统计学与R读书笔记(第六版)

徐俊晓

辛卯 兔年 十二月初八
(西元 2012年01月01日)

Contents

版权声明

警告

本文档是一个非正式的阅读笔记. 大部分内容来自其它资料. 虽然尽量注明参考文献与出处, 但是并未严格一一标明来源.

文档的R引用和实现大部分模仿R的帮助与其它资料, 很多代码原封不动的引用, 并未改动, 仅仅在适当的地方加入了注释. 对于个人撰写的部分, 会包含很多错误与不足之处. 敬请批评指正.

由于时间和精力的关系, 很多部分的内容是不完整的, 但是绝大部分都有其参考文献和出处. 强烈建议与其它正式资料一起阅读. 本人不对任何由此文档引发的后果负责.

温馨提示: 阅读原著有益身心健康, 二手资料易消化不良.

致谢

感谢所有对R的发展作出贡献的人[44]

感谢latex及其发明者: knuth

感谢我所参考过的和将要参考的所有资料的作者和编者.

第六版序

=======增加=======

《R基础》: 《数据库接口–RMySQL》 《推荐: R常见问题解答-153分钟学会R-Liu-FAQ》

《杂项》: 《Bootstrapping介绍》

《贝叶斯方法》: 《附: 一个例子的Bayes方法(不全)》

《非参数统计》: 《Randomization test of independence(permutation test)》

《线性模型》增加了下面几个分析的简单解释和例子:

- 《CFA 分析(Configural 频率分析)》
- 《关联分析(Correspondence Analysis)》
- 《通径分析》
- 《结构方程模型(SEM)》

=======修改=======

《回归与方差分析》: 拆分为《方差分析》和《线性模型》两个部分, 顺序和标题做了一定的调整, 条理更加清晰。

将原来《数据挖掘与机器学习》部分的《主成分分析(PCA)》《因子分析》《典型相关分析》放在《线性模型》部分.

《R基础》: 《绘图》部分添加了几个例子, 部分顺序和标题做了调整

将《杂项》部分的《图算法》和《信息理论》分别独立为一部分

其它小的修改: 略.....

徐俊晓

2012.01.01

辛卯兔年十二月初八

第五版序

增加和修改的部分比较杂, 在此不一一列出, 敬请见谅.

下面是不完全列表.

1. R基础部分将数据类型,读写与操作分两部分. 绘图部分添加了几个例子, 但是仍然不全面. 请参考其他资料.

2. 非参数统计提到试验设计之前

3. 回归与方差分析的修改比较多. 但是仍然不完全.

4. 谱分析中 sound 部分独立出来并添加声音数据的修改.

5. Bayes方法独立为一部分并增加了部分内容.

6. 杂项增加了对动态规划的介绍.

徐俊晓

2010.09.20

庚寅虎年八月十三

第四版序

增加及修改

- R基础
  - 增加了多元数据操作
  - 增加了S4类的介绍与例子, 包括定义类, 创建实例, 设置函数等操作.
- 数学部分
  - 增加"拉格朗日乘数"
  - 增加"空间几何", 包括坐标系旋转, 距离, 三角形等.
- 基本统计部分
  - 增加了"数据类型的划分"
  - "奇异值的处理"
- 回归及方差分析
  - "总结" 放到这部分的第一章了.
  - 广义线性回归部分结构及部分内容做了修改.
  - 增加了"偏最小二乘回归方法及其应用"
  - 注意factor的使用, 在一般情况下其方差分析结果自由度和方差显著不同.

- 杂项增加了
  - "信息熵与信息理论"
  - 图算法, 包括graph包(基本图方法), RBGL包(图算法), Rgraphviz包(图渲染)
- 增加了试验设计与部分, 包括分
  - 完全区组设计
  - 不完全区组设计
  - 正交设计.
- 增加了数据挖掘—机器学习部分. 包括
  - R包介绍
  - 基本概念
  - 基本算法描述
  - 树模型: 分类树, 回归树, 随机森林等
  - 核方法: svm, 各种核方法
  - 隐马尔可夫链(HMM)
  - 因子,聚类,主成分分析部分增加了"Fisher判别算法描述", 因为它们属于数据挖掘的线性方法, 所以转移到数据挖掘—机器学习部分.
- 增加了贝叶斯方法部分, 包括
  - 随机数的产生
  - 随机采样的几个方法: gibbs, 均匀采样, 马尔可夫链蒙特卡洛(MCMC)采样, 蒙特卡洛方法计算积分,
  - 贝叶斯方法的几个例子

徐俊晓

2009.06.27

第三版序

一切有为法, 如梦幻泡影, 如露亦如电, 应作如是观

若人言, 如来有所说法, 即为谤佛, 不能解我所说故

——摘自《金刚般若波罗密经》

演说：释迦牟尼

记录：阿难等（尊者）

翻译：鸠摩罗什（东晋后秦高僧）

般若波罗密, 即智慧到彼岸

增加

- 增加了"时间序列"部分. 包括
  - AR
  - MA
  - ARMA
  - ARIMA
  - ARCH
  - 谱分析
  - 小波分析
- 重写了"回归与方差分析" 部分
- 流行病学部分增加了筛选检验的一般概念和ROC曲线
- "基本统计分析" 部分"估计" 增加了"矩法" 和"极大似然法"

- ”杂项” 部分增加了 ”马尔可夫链与生物学”
- ”R基础与数学运算” 部分增加 ”运算符号” 与 ”复数基本运算” ”方程式求根” ”优化”

删除

- 删除 ”使用anova()比较多个模型”(可能解释有错误)

修改

- 对 ”R基础与数学运算”, ”基本统计分析”, ”回归与方差分析” 部分的结构顺序, 章节题目等做了较大变动
- 修正了 ”基本统计分析” 中 ”R的统计模型概述” 的公式格式错误, 并转移到 ”回归和方差分析”
- 修正了 ”方差不等的独立样本t检验” d'近似公式(原来有误)
- 修正了若干小错误和格式错误

徐俊晓

2008.12.28

第二版序

本次增加部分主要参考《统计建模与R软件》[21]. 部分参考 ”生物数学”[11]. 其它资源见正文.

增加

- "R 基础" 部分增加了 "数组与矩阵运算", "在 python 中调用 R(rpy2)".
- "回归与方差分析" 部分增加了 "逐步回归", "回归诊断"
- 增加了 "判别,聚类,因子分析等" 部分

改变

- 将 "绘图" 部分转入到 "R 基础".
- "广义线性 (Generalized Linear)模型" 和 "非线性回归与非线性最小平方" 两章补充了一点内容, 放到回归与方差分析(原线性回归与方差分析)
- "数据变换" 放入 "基本统计分析" 部分
- 修正了若干格式问题.

徐俊晓

2008.11.28

序

这是我学习生物统计学和R的笔记. 并不是一个系统介绍概率论与统计学和R应用的东西, 开始只是把用到的R的相关东西记下来, 以免忘记。后来看看记的还不少, 又想系统学习统计学, 就整理了一下。所以如果知道问题的解决方法, 直接看命令的用法就可以了。公式什么的是为了参考方便加入的, 随意性很强, 对希望系统了解的读者说声抱歉。

全部笔记除了第二部分“R基础”外，统计学部分主要参考 Bernard Rosner 的《生物统计学基础（Fundamentals of Biostatistics）》第五版 [14]。

孙尚拱先生说：我们的医学统计教师及流行病学教师们，如都能认真地阅读此书，我国的医学统计教学及科研水平必定会有大的提高。阅毕此书，深有同感。

由于本人比较懒散，故有的内容记录详细, 有的则简单。绝大部分内容是从参考文献得来, 开始不太在意参考文献的记录, 后来尽量加入了参考文献出处. 若没有注明, 请原文献作者谅解.

图：由于latex水平比较差(且本人很懒)，所有的图都没有放上来。

由于本人水平所限，其中肯定很多错误与不足的地方, 希望大家批评指正。尤其统计学的高级部分，象多元线性回归，广义线性回归（logistic回归，poisson回归，负二项回归），多元数据分析（因子分析，主成分分析，判别分析，聚类分析，典型相关分析）等部分更是似懂非懂，心有余而力不足，希望大家阅读的时候注意鉴别，如果有机会，以后可能会补充上述内容。笔记中TODO标记大多属于这种情况.

正当此笔记基本完成时, 看到一本书: 《统计建模与R软件》，薛毅，陈立萍编著，清华大学出版社。心想, 我想做的已经有人完成了, 而且非常之好, 作者水平也不是我所能比的, 后悔没有早点看到此书. 不过, 如果早看到了, 估计也就没有这篇笔记了.

愈写愈觉得自己所知其实有如沧海一粟, 不禁心生望洋之叹. 希望大家能够从这个笔记有所收获。祝大家学习进步。

徐俊晓

2008.10.01

Part I

R基础

此部分是R中的数据结构，语法等语言问题的描述，主要是平时遇到问题的一个汇总，虽然后来经过整理和添加，但是并不是一个系统介绍R的部分。若想系统了解R的语法和其它用法，请读者参考R网站的其它文档，这方面的文档是比较多的。R自带的帮助也是很不错的。主要参考了《simpleR》《R语言简介》《R for beginners》中文版，这几个都不错。

Chapter 1

环境相关

1.1 概述

R 的网站: http://cran.r-project.org/

进入网站, 点击左边的 Task Views, 浏览你需要的功能在哪个包里可以找到

Documentation 下面好多资料供参考.

安装位置在 /usr/share/R/ , 文档也在下面. 不过只是base的.

使用 google 的高级搜索, 站内搜索会更好

1.2 寻求帮助

```
> ?mean
> help(mean)
> help.search("mean")
> apropos(mean) # 或者 apropos("mean")
[1] "kmeans"         "weighted.mean"   "mean"           "mean.data.frame"
[5] "mean.Date"      "mean.default"    "mean.difftime"   "mean.POSIXct"
```

```
[9] "mean.POSIXlt"
```

1.3 管理R包

1.3.1 查看所有可用的包

使用 library()

1.3.2 查看某个包的信息

help(package="xxx")

1.3.3 查看当前调入内存的包

```
> search()
 [1] ".GlobalEnv"          "package:HSAUR"       "package:scatterplot3d"
 [4] "package:MASS"        "package:lattice"      "package:stats"
 [7] "package:graphics"     "package:grDevices"    "package:utils"
[10] "package:datasets"     "package:methods"      "Autoloads"
[13] "package:base"
```

1.3.4 载入需要的包

```
library(XXX) # XXX为包的名称
detach( "package:pkg" ) # library 的逆向操作
```

1.3.5 安装, 删除非二进制包

具体参考帮助

```
# 在线安装, 会提示选择 repos
install.packages("JGR",dep=TRUE)

# 本地安装两个包, XML_0.99-5.tar.gz, RSPerl_0.8-0.tar.gz
install.packages(
   c("XML_0.99-5.tar.gz",
   "../../Interfaces/Perl/RSPerl_0.8-0.tar.gz"),
   repos = NULL,
   configure.args = c(XML = '--with-xml-config=xml-config',
               RSPerl = "--with-modules='IO Fcntl'"))
```

有时候编译源代码需要 gcc, g++, gfortran 等编译器, 请单独安装.

一般的编译若提示 can not find -lblas 或 -llapack , 在系统shell下安装 libblas-dev 和 liblapack-dev 包

coin 包, 需要预先安装 refblas3 refblas3-dev 等库

shell 命令外部手工安装, 例如下载了 rgl包 rgl_0.81.tar.gz, 输入命令

```
sudo R CMD INSTALL rgl\_0.81.tar.gz

# windows外部安装包 Rgraphviz.
E:\biocbld\bbs-2.1-bioc\R\bin\R.exe CMD INSTALL --build --library=Rgraphviz.build
```

删除: remove.packages(utils)

1.3.6 升级更新包

直接下载R的最新版本安装即可升级到新版本.

update.packages() 更新所有已经安装的包. 比较现有的包的版本和source里面的包的版本, 如果发现新的, 下载并更新.

available.packages() 可用的更新.

download.packages(pkgs,...) 下载指定的包

packageStatus() 返回可更新信息

```
                           ok upgrade unavailable
  /usr/local/lib/R/site-library 41      70          0
  /usr/local/lib/R/library      22      5          0

Number of available packages (each package/bundle counted only once):

                                      installed not installed
  http://cran.cnr.Berkeley.edu/src/contrib      121         1642
```

1.4 环境变量与设置

1.4.1 查看当前环境下的变量

1.4.2 数字打印位数

options 设置或使用 print 参数

```
> old.digits = options("digits") # 保存默认打印字符长度 7
> options(digits=3)

> print(x,digits=2)
```

1.4.3 环境设置

详细见 *options*()

默认环境变量

```
‘add.smooth’          ‘TRUE’
‘check.bounds’        ‘FALSE’
‘continue’           ‘"+ "’
‘digits’             ‘7’
‘echo’               ‘TRUE’
‘encoding’           ‘"native.enc"’
‘error’              ‘NULL’
‘expressions’         ‘5000’
‘keep.source’         ‘interactive()’
‘keep.source.pkgs’    ‘FALSE’
‘max.print’          ‘99999’
‘OutDec’             ‘"."’
‘prompt’             ‘"> "’
‘scipen’             ‘0’
‘show.error.messages’ ‘TRUE’
‘timeout’            ‘60’
‘verbose’            ‘FALSE’
‘warn’               ‘0’
‘warning.length’      ‘1000’
‘width’              ‘80’
```

1.5 运行系统命令与R脚本及其命令行参数

R里面调用系统命令使用 system() 函数

```
system("ls x*")
```

如果需要保存输出结果为R对象, 加入参数intern=T

```
files <- system("ls x*",intern=T)
```

windows下运行脚本, 假设要运行的R脚本文件为 miRNA.r

```
C:\R\R-2.9.1\\bin\Rcmd.exe␣BATCH␣miRNA.r
␣␣␣␣
```

shell下运行R脚本.

-e 选项后面直接要执行的R命令

```
xjx@xjx-laptop:~$ Rscript -e "mean(1:10)"
[1] 5.5
```

如果R命令保存在文件里面, 例如 tmp.R 的内容为

```
x=1:10
y=mean(x)
print(y)
```

在shell下执行

```
xjx@xjx-laptop:~$ Rscript tmp.R
[1] 5.5
```

其它选项:
args: 是脚本传递的参数, 没测试
--save 保存工作空间
--restore 开始的时候载入当前目录的工作空间

在生成的Rscript文件的R程序中加入如下命令:

```
args <- commandArgs(TRUE)
```

通过上面的命令, 可以将命令行参数传递给args, 随后在R程序中就可以通过调用args这个变量来调用输入的命令行参数了,

Rscript脚本运行的命令如下:

```
Rscript foo.R arg1 arg2
```

其中foo.R是你创建的Rscript文件，后面的arg1和arg2是你输入的命令行参数

参考资料见R的在线文档：
http://cran.r-project.org/doc/manuals/R-intro.html#Scripting-with-R

1.6 内存管理

```
#␣启动时候管理内存，linux
R␣--max-vsize=2024M
#␣启动时候管理内存，windows
Rgui␣--min-vsize=10M␣--max-vsize=100M␣--min-nsize=500k␣--max-nsize=1M

#␣启动后修改最大内存与查询内存信息
help(memory.size)
memory.size(max␣=␣FALSE)
memory.limit(size␣=␣NA)
memory.limit()
memory.profile()
␣␣␣␣␣␣␣␣
```

1.7 R启动时调用的文件和函数

位置初始化文件的路径可以通过环境变量 R_PROFILE 设置。若 R_PROFILE 未设置, 则默认为R安装目录下面的子目录 etc 中的 Rprofile.site. 此文件包含执行 R 时的一些自动命令.

.Rprofile 文件允许用户定制它们的工作空间，设置不同的起始命令。此文件可以放在任何目录下。如果R 在该目录下面被调用，这个文件就会被载入。如果在起始目录中没有 .Rprofile, R 会在用户主目录下面搜索 .Rprofile 文件并且调用它(如果它存在的话)。

另外一个可以配置的文件是 .RData

.First() 函数：.Rprofile，.RData 文件中的函数，此函数可以定义自己的设置。例如：

```
> .First <- function() {
  options(prompt="$ ", continue="+\t") # $ 是提示符
  options(digits=5, length=999) # 定制数值和输出格式
  x11() # 定制图形环境
  par(pch = "+") # 定制数据点的标示符
  source(file.path(Sys.getenv("HOME"), "R", "mystuff.R")) # 个人编写的函数
  library(MASS) # 导入包
}
```

类似的是，如果定义了函数.Last()，它(常常)会在对话结束时执行。一个例子就是

```
> .Last <- function() {
# 一个小的安全措施。
  graphics.off()
# 该吃午饭了？
  cat(paste(date(),"\nAdios\n"))
}
```

1.8 推荐: R常见问题解答-153分钟学会R-Liu-FAQ

熟习基本操作后, 里面有对数据操作, 数学运算, 日期时间, 绘图, 统计模型等的各种技巧总结.

最后还包涵了一个 latex 嵌入 R 代码与结果的 Sweave 的介绍.

Chapter 2

类和泛型函数

一个对象的类决定了它会如何被一个泛型函数处理。相反，一个泛型函数由参数自身类的种类来决定完成特定工作或者事务的。如果参数缺乏任何类属性，或者在该问题中有一个不能被任何泛型函数处理的类，泛型函数会有一种默认的处理方式。

2.1 S3类

参考文献[38] 2.3.5. 有详细的描述.

中文版 R-lang http://www.biosino.org/R/R-doc/R-lang/

S3类是R内核带的. 把一个list的属性class赋值, 其它泛型函数在接受此参数的时候首先查看其类, 然后调用合式的方法.

```
h <- list(a=rnorm(3),b="This shouldn't print")
class(h) <- "myclass"
print.myclass<-function(x){cat("A is:",x$a,"\n")}
print(h)

A is: -0.710968 -1.611896 0.6219214
```

2.1.1 查看类可用的泛型函数

可以用函数methods() 得到当前对某个类对象可用的泛型函数列表：

```
methods(class="data.frame")
```

2.1.2 查看泛型函数可处理的类

相反，一个泛型函数可以处理的类同样很多。例如，plot() 有默认的方法和变量处理对象类"data.frame"，"density"，"factor"，等等。一个完整的列表同样可以通过函数methods() 得到：

```
methods(plot)
```

2.1.3 查看泛型函数代码

许多泛形函数的函数主体部分非常的短，如

```
> coef
function (object, ...)
UseMethod("coef")
```

UseMethod 的出现暗示着这是一个泛形函数。为了查看那些方法可以使用，我们可以使用函数methods()

```
> methods(coef)
[1] coef.aov*        coef.Arima*      coef.default* coef.listof*
[5] coef.nls*        coef.summary.nls*
```

```
   Non-visible functions are asterisked
```

这个例子中有六个方法，不过其中任何一个都不能简单地通过键入名字来查看。我们可以通过下面两种方法查看这种方法

```
> getAnywhere("coef.aov")
A single object matching 'coef.aov' was found
It was found in the following places
  registered S3 method for coef from namespace stats
  namespace:stats
with value

function (object, ...)
{
    z <- object$coef
    z[!is.na(z)]
}

> getS3method("coef", "aov")
function (object, ...)
{
    z <- object$coef
    z[!is.na(z)]
}
```

2.1.4 编写自己的类和泛型函数

下面是一个例子

```
# 编写函数
> xpos
function(x, ...)
    UseMethod("xpos")
```

```
> xpos.xypoint <- function(x) x$x
> xpos.rthetapoint <- function(x) x$r * cos(x$theta)

> xpos
function(x, ...)
   UseMethod("xpos")

# 改变数据的类
> x=list(x=c(1,2))
> x
$x
[1] 1 2

> x$x
[1] 1 2
> class(x)="xypoint"
> x
$x
[1] 1 2

attr(,"class")
[1] "xypoint"

# 调用泛型函数
> xpos(x)
[1] 1 2
```

2.1.5 修改函数

若需要修改函数 mean, 这样

```
edit(mean)
␣␣␣␣
```

或直接赋值也可以

2.2 S4 class

参考文献: 作者: R-用户, *R S4*面向对象编程 (http://rbbs.biosino.org/Rbbs/posts/list

参考文献: Martin Morgan, Robert Gentleman *Lecture: S4 classes and methods* 14 February, 2008

参考各个函数的帮助.

S4类是近期加入R的, 由包methods实现, R已经自带. 一般包含数据和函数, 类似其它语言的面向对象的语法.

setClass 创建新的类. new()函数创建其对象. 其属性使用"@"引用.

2.2.1 一些名词使用的说明

slot, 变量: 类内部包含的数据

对象, 实例: 类的一个实现.

查看一个类的信息, 例如类 genind 使用

```
class?genind
```

2.2.2 setClass(): 定义新类

```
setClass(Class, representation, prototype, contains=character(),
        validity, access, where, version, sealed, package)
```

```
## A simple class with two slots
# 定义一个简单的类, 名称为 track.
# 参数 representation 定义变量(slot). 带有2个变量(slot): x,y
    setClass("track",
```

```
          representation(x="numeric", y="numeric"))

## A class extending the previous, adding one more slot
# 定义新类 trackCurve.
# 参数 contains 说明继承自 track 类. 即包含变量 x,y, 另外还
包含变量 smooth
    setClass("trackCurve",
       representation(smooth = "numeric"),
       contains = "track")

## A class similar to "trackCurve", but with different structure
## allowing matrices for the "y" and "smooth" slots
# 定义新类 trackMultiCurve.
# 其中数据结构 y, smooth 为矩阵.
# 参数 prototype 为变量的初始化值. 可以省略 x=numeric()
    setClass("trackMultiCurve",
           representation(x="numeric", y="matrix", smooth="matrix"),
           prototype = list(x=numeric(), y=matrix(0,0,0),
                        smooth= matrix(0,0,0)))
##
## Suppose we want trackMultiCurve to be like trackCurve when there's
## only one column.
## First, the wrong way.
# 我们希望当只有一列的时候, trackMultiCurve 与 trackCurve 行
为类似.
# 下面是错误用法. 需要显式转换 matrix 到 numeric
    try(setIs("trackMultiCurve", "trackCurve",
       test = function(obj) {ncol(slot(obj, "y")) == 1}))

## Why didn't that work?  You can only override the slots "x", "y",
## and "smooth" if you provide an explicit coerce function to correct
## any inconsistencies:
# 下面是正确的用法.
# 参数 coerce 为一个函数, 当 test =TRUE, 就把 class1 的对
象(实例)定义为 class2.
    setIs("trackMultiCurve", "trackCurve",
      test = function(obj) {ncol(slot(obj, "y")) == 1},
      coerce = function(obj) {
        new("trackCurve",
           x = slot(obj, "x"),
           y = as.numeric(slot(obj,"y")),
```

```
            smooth = as.numeric(slot(obj, "smooth")))
})

## A class that extends the built-in data type "numeric"
# 创 建 新 类 numWithId, 继 承 内 部 类 numeric. 使 得 其 有 一
个 id 标 志.
    setClass("numWithId", representation(id = "character"),
            contains = "numeric")

# 创 建 numWithId 类 的 一 个 对 象. 使 用 1:3 初 始 化 值, id="An Example"
    new("numWithId", 1:3, id = "An Example")

## inherit from reference object of type "environment"
# 继 承 内 部 类 environment
    setClass("stampedEnv", contains = "environment",
         representation(update = "POSIXct"))

# 创 建 实 例.
    e1 <- new("stampedEnv", new.env(), update = Sys.time())

    setMethod(f, signature=character(), definition,
            where = topenv(parent.frame()),
            valueClass = NULL, sealed = FALSE)

# 创 建 方 法.
    setMethod("[[<-", c("stampedEnv", "character", "missing"),
      function(x, i, j, ..., value) {
          ev <- as(x, "environment")
          ev[[i]] <- value  #update the object in the environment
          x@update <- Sys.time() # and the update time
          x})

    e1[["noise"]] <- rnorm(10)
```

2.2.3 getClass(): 查看类定义和继承情况

获得其继承和被继承关系. 内含的变量(slot), 但是不能获得能够使用它的函数情况.

R基本数据结构和类型都有S4类与之对应, 可以用getClass查看它们的定义情况。

```
> getClass("track")
Class \track" [in ".GlobalEnv"] 类名称, 及其所在空间

Slots: 变量

Name:       x      y
Class: numeric numeric

Known Subclasses: 子类
Class "trackCurve", directly
Class "trackMultiCurve", by class "trackCurve", distance 2,
       with explicit test and coerce


# numeric 类的情况
> getClass("numeric")
Class \numeric" [package "methods"]

No Slots, prototype of class "numeric"

Extends: "vector" # 父类

Known Subclasses: # 子类
Class "integer", directly
Class "numWithId", from data part
Class "ordered", by class "integer", distance 3


# vector 类的情况
> getClass("vector")
Virtual Class \vector" [package "methods"] # 虚类
```

```
No Slots, prototype of class "NULL"

Known Subclasses:
Class "logical", directly
Class "numeric", directly
Class "character", directly
Class "complex", directly
Class "integer", directly
Class "raw", directly
Class "expression", directly
Class "list", directly
Class "structure", directly, with explicit coerce
Class "array", by class "structure", distance 2, with explicit coerce
Class "matrix", by class "array", distance 3, with explicit coerce
Class "signature", by class "character", distance 2
Class "ObjectsWithPackage", by class "character", distance 2
Class "mts", by class "matrix", distance 4, with explicit coerce
Class "ordered", by class "factor", distance 3
Class "numWithId", by class "numeric", distance 2
```

2.2.4 new(): 创建类的实例(对象)与初始化

```
new(Class, ...)

initialize(.Object, ...)
```

```
# 创建track类的一个实例 t1, 同时初始化其变量 x,y
> t1 <- new("track", x = 1:10, y = 1:15)
> t1
An object of class \track"
Slot "x":
 [1]  1  2  3  4  5  6  7  8  9 10

Slot "y":
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
```

```
# a new object including an object from a superclass, plus a slot
# 创建 trackCurve 的实例.
# 因为从 track 继承来, 可以使用 t1 的数据 x,y, 同时设置变量 smooth
> t2 <- new("trackCurve", t1, smooth = 1:20)
> t2
An object of class \trackCurve"
Slot "smooth":
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20

Slot "x":
 [1]  1  2  3  4  5  6  7  8  9 10

Slot "y":
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15

### define a method for initialize, to ensure that new objects have
### equal-length x and y slots.
# 定义 track 类的初始化函数, 保证 x,y 的长度相同.
setMethod("initialize",
       "track",
       function(.Object, x = numeric(0), y = numeric(0)) {
          if(nargs() > 1) {
          if(length(x) != length(y))
             stop("specified x and y of different lengths")
          .Object@x <- x
          .Object@y <- y
       }
       .Object
})

# 此时其子类 trackCurve 这样使用 new() 函数会出错. 只能另外定义自己的初始化函数.
# 原因下面解释.
> t2 <- new("trackCurve", t1, smooth = 1:20)
错误于.local(.Object, ...) : 变元((smooth = 1:20)) 没有用

## a better way to implement the previous initialize method.
## Why?  By using callNextMethod to call the default initialize method
## we don't inhibit classes that extend "track" from using the general
```

```
## form of the new() function.  In the previous version, they could only
## use x and y as arguments to new, unless they wrote their own
## initialize method.
# 这是个更好的方法定义初始化函数.
# 使用 callNextMethod() 调用其默认初始化函数.
# 我们不禁止其子类使用通用的 new() 函数.
# 在前一个版本, 其子类只能使用 x,y 作为 new() 的参数, 除非
其子类有自己的初始化函数.
    setMethod("initialize", "track", function(.Object, ...) {
        .Object <- callNextMethod()
        if(length(.Object@x) != length(.Object@y))
         stop("specified x and y of different lengths")
        .Object
})

# 此时其子类 trackCurve 这样使用 new() 函数就返回正确的结
果.
> t2 <- new("trackCurve", t1, smooth = 1:20)
错误于initialize(value, ...) : specified x and y of different lengths

> t1@x<-1:10
> t1@y<-1:10
> t2 <- new("trackCurve", t1, smooth = 1:20)
> t2
An object of class \trackCurve"
Slot "smooth":
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20

Slot "x":
 [1]  1  2  3  4  5  6  7  8  9 10

Slot "y":
 [1]  1  2  3  4  5  6  7  8  9 10
```

2.2.5 setMethod()—getMethods(): 定义和查看使用新类的函数方法

setMethod 的用法

```
setMethod(f, signature=character(), definition,
        where = topenv(parent.frame()),
        valueClass = NULL, sealed = FALSE)

removeMethod(f, signature, where)
```

例子

```
# 定义类 track
setClass("track",
      representation(x="numeric", y="numeric"))

# 创建 track 类的一个实例
t=new("track")
t@x=1:10
t@y=1:15

# 定义函数 length, 参数为 track 类. 具体实现为
setMethod(f="length", signature="track", definition=function(x)length(x@y))
length(t)

# 可以事先定义一个函数(或使用已经存在的函数)
length.track<-function(x){ length(x@y)}
setMethod(f="length", signature="track", definition=length.track)
length(t)

# 结果
> length(t)
[1] 15

#==================================
# 上面的定义函数的方法有一个问题, 就是参数可以
是 ANY, 有时候对参数检查不利.
```

```
# 可以使用 setGeneric() 来阻止参数类型为 ANY 的函数使用此类作为参数
act <- function(obj, para) {}
setGeneric("act", useDefault = F)
setMethod("act", signature(obj = "track", para = "numeric"),
function (obj, para) obj@x + obj@y + para)

# 上面可以合并为一个命令
setGeneric("act", function (obj, para) standardGeneric("act"))
```

2.2.6 查看函数的参数的类与类绑定的情况

```
> showMethods(length)
Function: length (package base)
x="track"

# 使用 getMethods 可以得到更多信息
> getMethods("length")
An object of class \MethodsList"
Slot "methods":
$ANY
function (x)  .Primitive("length")

$track
Method Definition:

function (x)
{
   length(x@y)
}

Signatures:
       x
target  "track"
defined "track"


Slot "argument":
```

```
x

Slot "allMethods":
$ANY
function (x)  .Primitive("length")

$track
Method Definition:

function (x)
{
   length(x@y)
}

Signatures:
       x
target  "track"
defined "track"
```

Chapter 3

编写自己的函数

一些比较杂的东西可能在[46] programming whith R, Dirty thricks

3.1 特殊符号函数

还可以设计自己的符号函数

```
> "%w/o%" <- function(x,y) x[!x %in% y]
> (1:10) %w/o% c(3,7,12)
[1] 1 2 4 5 6 8 9 10

> x=1:10
> y=1:10
> z=1:10
> "%i%" <- intersect
> intersect(x,y)   # Only two arguments
 [1]  1  2  3  4  5  6  7  8  9 10
> intersect( intersect(x,y), z )
 [1]  1  2  3  4  5  6  7  8  9 10
> x %i% y %i% z
 [1]  1  2  3  4  5  6  7  8  9 10
```

3.2 异常

stop("something") warning("something")

3.3 字符串表达式与求值

parse(), expression()

```
# 解析表达式(默认第一个参数为文件, 字符串需使用text参数指定)
# 返回表达式列表(list), 但是不求值
> parse(text="0==1")
expression(0 == 1)
# 求字符串代表的表达式的值
> eval(parse(text="0==1"))
[1] FALSE
```

表达式也可以用在绘图中, 会出现数学符号, 而不是字符串

```
x <- seq(0,4, length=100)
y <- sqrt(x)
plot(y~x, type="l", lwd=3, main=expression(y == sqrt(x)))
```

3.4 deparse(), substitute()

deparse() 返回函数代码字符串. substitute() 将变量替换为其值.

```
> temp<-function(x) {cat(deparse(substitute(x)))}
> x=1:10
```

```
> deparse(sum)
[1] ".Primitive(\"sum\")"
> deparse(temp)
[1] "function (x) "                 "{"
[3] "    cat(deparse(substitute(x)))" "}"
> temp("x")
"x">
> temp(x)
x>
```

3.5 stop和warning, 警告级别

例子来自 [46] 2.17.11

```
# 参数必须为长度>1的向量, 否则报错
do.it <- function (x) {
  if( !is.numeric(x) )
    stop("Expecting a NUMERIC vector!")
  if( !is.vector(x) )
    stop("Expecting a numeric VECTOR!")
  if( length(x)<2 )
    stop("Expecting a numeric vector of length at least 2")
  return("Well done.")
}
```

```
>  do.it("abc")
错误于do.it("abc") : Expecting a NUMERIC vector!
>  do.it(3)
错误于do.it(3) : Expecting a numeric vector of length at least 2
>  do.it(data.frame(a=1:3,b=3:1))
错误于do.it(data.frame(a = 1:3, b = 3:1)) :
  Expecting a NUMERIC vector!
>  do.it(matrix(1:4,nc=2,nr=2))
错误于do.it(matrix(1:4, nc = 2, nr = 2)) : Expecting a numeric VECTOR!
>  do.it(1:26)
[1] "Well done."
```

```
options(warn=-1) # 不打印警告
options(warn=0) # 直到最后才打印警告
options(warn=1) # 当警告发生时打印
options(warn=2) # 把警告当做错误
```

3.6 environment, new.env(), assign(), get()

```
x <- new.env(hash=T)
assign("foo", 3, env=x)
assign("bar", list("a"=3, "b"=list("c"=1, d="foo")), env=x)
assign("baz", data.frame(rnorm(10),rnorm(10)), env=x)
> ls(env=x)
[1] "bar" "baz" "foo"
> x
<environment: 0x832d470>
> get("foo")
错误于get("foo") : 找不到"foo"这个变数
> get("foo",env=x)
[1] 3
```

下面是environment在线的例子

```
> ?environment

> f <- function() "top level function"
>
>     ##-- all three give the same:
>     environment()
<environment: R_GlobalEnv>
>     environment(f)
<environment: R_GlobalEnv>
>     .GlobalEnv
<environment: R_GlobalEnv>
>
>     ls(envir=environment(stats::approxfun(1:2,1:2, method="const")))
```

```
[1] "f"     "method" "n"     "ux"    "x"     "y"     "yleft"  "yright"
>
>     is.environment(.GlobalEnv) # TRUE
[1] TRUE
>
>     e1 <- new.env(parent = baseenv())  # this one has enclosure package:base.
>     e2 <- new.env(parent = e1)
>     assign("a", 3, envir=e1)
>     ls(e1)
[1] "a"
>     ls(e2)
character(0)
>     exists("a", envir=e2)   # this succeeds by inheritance
[1] TRUE
>     exists("a", envir=e2, inherits = FALSE)
[1] FALSE
>     exists("+", envir=e2)   # this succeeds by inheritance
[1] TRUE
```

```
> foo<<-"new foo"
> get("foo")
[1] "new foo"
> get("foo",env=x)
[1] 3
```

3.7 测试运行时间

将函数包含在 system.time() 内, 返回运行时间

```
> system.time(for(i in 1:100) mad(runif(1000)))
 用户  系统  流逝
0.196 0.003 0.241
```

Chapter 4

数据类型

更多参考《R导论》

R操作的实体在技术上来说就是对象(object).

4.1 原子类型

R的对象类型包括数值型（numeric），复数型（complex），逻辑型（logical），字符型（character）和原味型（raw）.

4.2 NA

参考《statistics with R》

4.3 向量

向量必须保证它的所有元素是一样的模式.

向量必须明确属于逻辑型，数值型，复数型，字符型或者原味型。(这里有一个特定的例外是值为"NA"的元素. 实际上NA有好几种类型).

空向量也有自己的模式.

向量对象的类型的包括: 实数, 复数, 逻辑, 字符串. 它们是原子(atomic), 即元素类型一样.

4.4 因子

一个因子不仅包括分类变量本身还包括变量不同的可能水平（即使它们在数据中不出现）。因子函数factor用下面的选项创建一个因子：

factor及ordered函数在统计模型中特别有用. 例如将0,1改变为'y', 'n' 也很方便.

```
factor(x, levels = sort(unique(x), na.last = TRUE),
     labels = levels, exclude = NA, ordered = is.ordered(x))
```

levels用来指定因子可能的水平（缺省值是向量x中互异的值）；labels用来指定水平的名字；exclude表示从向量x中剔除的水平值；ordered是一个逻辑型选项用来指定因子的水平是否有次序。

函数tapply() 将一个功能函数（这里是mean()）用于第二个参数（这里是o）定义于第一个参数（这里是x）上得到的所有组(以factor或ordered决定)

注意，当第二个参数不是因子时，函数tapply() 同样有效，如tapply(x, state)。这对一些其他函数也是有效，因为必要时R会用as.factor() 把参数强制转换成因子。

```
> x=rbinom(n=10,size=2,p=c(0.2,0.3,0.5))
```

```
> x
[1] 1 1 2 0 2 0 0 1 1 0

> f=factor(x)
> f
[1] 1 1 2 0 2 0 0 1 1 0
Levels: 0 1 2

> factor(x,levels=0:3)
 [1] 1 1 2 0 2 0 0 1 1 0
Levels: 0 1 2 3

> factor(x,labels=c('a','b','c'))
 [1] b b c a c a a b b a
Levels: a b c

> t=table(x)
> t
x
0 1 2
4 4 2

> o=ordered(x)
> o
[1] 1 1 2 0 2 0 0 1 1 0
Levels: 0 < 1 < 2

> tapply(x,o,mean)
0 1 2
0 1 2
```

4.5 列表(list)

R的列表（list）是一个以对象的有序集合构成的对象。列表中包含的对象又称为它的分量（components）. 每个分量的长度和类型可以不同.

列表被认为是一种“递归”结构而不是原子结构, 因为它们的元素可以以它们各自的方式单独列出.

由于是递归的, 所以在产生长的列表的时候, 使用

```
vector("list",n)
```

可以初始化元素数为n的列表以加快速度.

分量可以是不同的模式或类型, 如一个列表可以同时包括数值向量, 逻辑向量, 矩阵, 复向量, 字符数组, 函数等等。下面的例子演示怎么创建一个列表与查看其信息:

```
> Lst <- list(name="Fred", wife="Mary", no.children=3,
            child.ages=c(4,7,9))
# 查看信息
> str(Lst)
List of 4
 $ name      : chr "Fred"
 $ wife      : chr "Mary"
 $ no.children: num 3
 $ child.ages : num [1:3] 4 7 9

> Lst
$name
[1] "Fred"

$wife
[1] "Mary"

$no.children
[1] 3

$child.ages
[1] 4 7 9
```

获取分量

Lst$name 和Lst[[1]] 返回结果都是"Fred",
Lst$wife 和Lst[[2]] 返回的则是"Mary",
而Lst$child.ages[1] 和Lst[[4]][1] 返回一样的数字4。

```
> Lst$name
[1] "Fred"
> Lst[1]
$name
[1] "Fred"
> Lst$child.ages[1]
[1] 4
> Lst[4]
$child.ages
[1] 4 7 9

> Lst[4][1]
$child.ages
[1] 4 7 9

> Lst[[4]][1]
[1] 4
```

这里特别要注意一下Lst[[1]] 和Lst[1] 的差别。[[. . .]] 是用来选择单个元素的操作符，而[. . .] 是一个更为一般的下标操作符。因此前者得到的是列表Lst 中的第一个对象, 并且含有分量名字的命名列表（named list）中的分量名字会被排除在外的1 。后者得到的则是列表Lst 中仅仅由第一个元素构成的子列表。如果是命名列表，分量名字会传给子列表的。

4.6 数据框–data.frame

数据框（data frame）也是列表, 是一个属于"data.frame" 类的列表。不过，对于可能属于数据框的列表对象有一些限制条件.

分量必须是向量(数值, 字符, 逻辑)，因子，数值矩阵，列表或者其他数据框; 每列的行数必须相等.

数据框常常会被看作是一个由不同模式和属性的列构成的矩阵。它能以矩阵形式出现，行列可以通过矩阵的索引习惯访问。

4.7 数组(array)及维度命名

数组可以看作是带有多个下标类型相同的元素集合，如数值型, 是矩阵的推广. R 有一些简单的工具创建和处理数组，特别是矩阵。

向量只有在定义了 dim 属性后才能作为数组在R 中使用。假定，z是一个含1500个元素的向量。那么

```
> dim(z) <- c(3,5,100)
```

对dim 属性的赋值使得z向量成一个3维的3*5*100的数组。

```
> z[1,] # z 的第一行
> z[,1] # z 的第一列
> z[1:3,] # z 的第1:3行
> z[2*(1:3)-1,] # z 的1,3,5行. 括号可以不加. z[2*1:3-1,]

# 命名的顺序总是行,列,第三维,..., 每一维还可以有一个总名字, 也可以没有
Rabbits <-array(
      c( 0, 0, 6, 5,
         3, 0, 3, 6,
         6, 2, 0, 4,
         5, 6, 1, 0,
         2, 5, 0, 0),
   dim = c(2, 2, 5),
   dimnames = list(
      Delay = c("None", "1.5h"),
      Response = c("Cured", "Died"),
      Penicillin.Level = c("1/8", "1/4", "1/2", "1", "4")))
```

```
> Rabbits
, , Penicillin.Level = 1/8

     Response
Delay  Cured Died
  None    0    6
  1.5h    0    5

, , Penicillin.Level = 1/4

     Response
Delay  Cured Died
  None    3    3
  1.5h    0    6

, , Penicillin.Level = 1/2

     Response
Delay  Cured Died
  None    6    0
  1.5h    2    4

, , Penicillin.Level = 1

     Response
Delay  Cured Died
  None    5    1
  1.5h    6    0

, , Penicillin.Level = 4

     Response
Delay  Cured Died
  None    2    0
  1.5h    5    0
```

4.8 矩阵

矩阵 (matrix) 是一个双下标(2维)的数组. 但是, 它非常的重要, 以至于需要单独讨论。

R 包括许多只对矩阵操作的操作符和函数。

命名与数组array()一样.

矩阵的下标顺序是先第一列, 然后第二列, 等等. 例如

```
d<-matrix(c(1,2,3,4,5,6,7,8,9),nc=3)
```

```
> d
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> d[1:5]
[1] 1 2 3 4 5
```

4.9 字符串及相关操作

字符串比较重要, 所以单独讨论.

针对字符串的函数有 print, paste, cat, nchar, strsplit, regexpr, grep, gsub, sub 等.

```
> seq="GGGGCGAAACCGAGACTCTCAAATGACTTTTCTGA"
> seq=strsplit(seq,"")
> seq
[[1]]
 [1] "G" "G" "G" "G" "C" "G" "A" "A" "A" "C" "C" "G" "A" "G" "A" "C" "T" "C" "T"
[20] "C" "A" "A" "A" "T" "G" "A" "C" "T" "T" "T" "T" "C" "T" "G" "A"
```

```
> seq[[1]]=="g"|seq[[1]]=="G"
 [1]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE
[13] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[25]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE

> s<-c("a","b")
> s
[1] "a" "b"
> paste(s)
[1] "a" "b"
> paste(s,collapse="+")
[1] "a+b"
```

下列函数参考具体帮助

- chartr(old, new, x): 将字符串x中的old换成new
- tolower(x), toupper(x): 大小写变换
- paste, cat: 合并字符串, 用法稍微不同
- nchar: 有多少字母
- substr(x, start, stop): 提取指定位置之间的子字符串, 或将其赋新值
- strsplit: 使用指定的分隔符分隔字符串
- gsub, sub: 使用指定字符串替换子字符串, sub只替换第一个
- regexpr, grep: 在字符串中查找指定字符串(正则表达式)位置

```
>␣a1="....."
>␣a2=".(())."
>␣a3="...()"
>␣a4="()..."
>␣a5="(())"
```

```
#␣查 询 开 始 部 分
>␣grep("^\\.+",a1)
[1]␣1
>␣grep("^\\.+",a2)
[1]␣1
>␣grep("^\\.+",a4)
integer(0)
>␣grep("^\\.+",a5)
integer(0)
#查 询 末 尾 部 分
>␣grep("\\.+$",a1)
[1]␣1
>␣grep("\\.+$",a3)
integer(0)
>␣grep("\\.+$",a4)
[1]␣1

#␣gregexpr␣的 用 法$
>␣gregexpr("^\\.+",a1)[[1]]
[1]␣1
attr(,"match.length")
[1]␣5
>␣gregexpr("^\\.+",a2)[[1]]
[1]␣1
attr(,"match.length")
[1]␣1
>␣gregexpr("^\\.+",a4)[[1]]
[1]␣-1
attr(,"match.length")
[1]␣-1
>␣gregexpr("^\\.+",a5)[[1]]
[1]␣-1
attr(,"match.length")
[1]␣-1


␣␣␣␣
```

4.10 分数

MASS fraction 函数可以将小数转换为近似的分数，例如

```
>␣fractions(0.291667)
[1]␣7/24
>␣fractions(0.333)
[1]␣333/1000
>␣fractions(0.333333333333)
[1]␣1/3
>␣fractions(pi)
[1]␣4272943/1360120
␣␣␣␣
```

法里序列（farey␣sequence）也是考虑这类问题的一个角度。如果给定法里序列的␣n␣足够大，那么我们必定能够将逼近出一个和小数相等的分数Fi[j]。

法里序列␣Fi␣（i=1␣到␣n）：

```
␣␣␣␣F1␣=␣{01,␣11}
␣␣␣␣F2␣=␣{01,␣12,␣11}
␣␣␣␣F3␣=␣{01,␣13,␣12,␣23,␣11}
␣␣␣␣F4␣=␣{01,␣14,␣13,␣12,␣23,␣34,␣11}
␣␣␣␣F5␣=␣{01,␣15,␣14,␣13,␣25,␣12,␣35,␣23,␣34,␣45,␣11}
␣␣␣␣F6␣=␣{01,␣16,␣15,␣14,␣13,␣25,␣12,␣35,␣23,␣34,␣45,␣56,␣11}
␣␣␣␣F7␣=␣{01,␣17,␣16,␣15,␣14,␣27,␣13,␣25,␣37,␣12,␣47,␣35,␣23,␣57,␣34,␣45,␣56,␣67
␣␣␣␣F8␣=␣{01,␣18,␣17,␣16,␣15,␣14,␣27,␣13,␣38,␣25,␣37,␣12,␣47,␣35,␣58,␣23,␣57,␣34
```

但这个过程会比较麻烦，F1000␣已经达到300927␣个数字。

```
␣␣␣␣
```

4.11 日期和时间

4.11.1 当前日期和时间

当前日期和时间, 返回 Date 和 DateTimeClasses 对象.

```
> date()
[1] "Thu Jul 15 19:02:56 2010"
> nchar(date()) == 24
[1] TRUE
> class(date())
[1] "character"

> Sys.time()
[1] "2010-07-15 19:04:19 CST"
> class(Sys.time())
[1] "POSIXt"  "POSIXct"

> Sys.Date()
[1] "2010-07-15"
> class(Sys.Date())
[1] "Date"
```

4.11.2 DateTimeClasses

包括两个类. "POSIXlt" "POSIXct"

POSIXct: 表示从 1970 年开始到现在的秒数.

POSIXlt: 为 list, 包含

- sec 0–61: seconds
- min 0–59: minutes
- hour 0–23: hours

- mday 1–31: day of the month
- mon 0–11: months after the first of the year.
- year Years since 1900.
- wday 0–6 day of the week, starting on Sunday.
- yday 0–365: day of the year.
- isdst Daylight savings time flag. Positive if in force, zero if not, negative if unknown.

```
> s='08:10:00'
> z=strptime(s,'%H:%M:%S')
> z
[1] "2010-07-15 08:10:00"
> class(z)
[1] "POSIXt" "POSIXlt"

> z$
z$hour   z$isdst  z$mday   z$min    z$mon    z$sec    z$wday   z$yday   z$year
> z$sec
[1] 0
> z$year
[1] 110
```

4.11.3 格式: format 参数的书写

参考 ?strptime

默认 ”

4.11.4 时区问题

带时区的转换是个比较难的问题. 参考 ?strptime

4.11.5 字符串转换为日期时间

字符串转换为日期时间的函数有 format 和 as.character

format 将 Date POSIXlt POSIXct 转换为指定格式的字符串.

as.character 转换为字符串

```
> s='08:10:00'
> z=strptime(s,'%H:%M:%S')
> z
[1] "2010-07-15 08:10:00"
> class(z)
[1] "POSIXt" "POSIXlt"

> x=format.POSIXlt(z,format="%Y-%m-%d")
> x
[1] "2010-07-15"

> as.character(z)
[1] "2010-07-15 08:10:00"
```

4.11.6 字符串转换为日期时间

strptime 将字符串转换为 "POSIXlt".

strftime 是 format.POSIXlt 的一个 wrapper.

```
> w=strptime("09:10:00",'%H:%M:%S') # 默认转换为 DateTimecleses
> w
[1] "2010-07-15 09:10:00"

# 将 DateTimecleses 转换为 Date
> as.Date(w)
[1] "2010-07-15"
```

还有两个转换函数:

ISOdatetime(year, month, day, hour, min, sec, tz = "")

ISOdate(year, month, day, hour = 12, min = 0, sec = 0, tz = "GMT")

4.11.7 时间差异

units = c("auto", "secs", "mins", "hours","days", "weeks")

```
> difftime(w,z,units="sec")
Time difference of 3600 secs

> as.double(difftime(w,z,units="sec"))
[1] 3600

> a=difftime(w,z,units="sec")
> b=difftime(z1,z,units="sec")
> a
Time difference of 3600 secs
> b
Time difference of 300 secs
> a-b
Time difference of 3300 secs
```

4.11.8 绘制日期时间

plot.Date and hist.Date for plotting.

Chapter 5

数据的读写与操作

5.1 查看数据

查看当前环境下的数据 ls()

删除数据 rm()

查看所有预先提供的数据: data()

查看base包所有预先提供的数据，使用 data(package="base")

载入数据，使用data('dataset name')，引号可以不加

5.2 读写

5.2.1 简单数据编辑器

如果X是一个矩阵，命令data.entry(X)将打开一个图形编辑器并且可以通过点击适当的单元格修改数值或者添加新的行或列.

5.2.2 导入 Execl 格式

(参考 R-data.pdf) 如果能够避免尽量避免直接导入. 因为xls格式复杂, 还可以包含很多数据表.

把xls保存为csv或其他格式(可以使用openoffice 或 gnumeric), 使用 read.delim2 或 read.csv2 导入.

例如, hormone.xls 使用分隔符 "," 保存为 hormone.csv

row.names=1 提示第一列为 rownames

```
d=read.csv2("hormone.csv",sep=",")
d=read.csv('data.csv',head=T,row.names=1)
```

目前Excel有个插件 RExcel, 可以到这个网http://rcom.univie.ac.at/站下载, 然后在电子表格上使用R语言中的函数。

5.2.3 好用的剪切板

使用表格工具, 例如windows的excel, linux下 的openoffice-spreadsheet, gnumeric [1] 等打开数据, 复制要选取的部分, 然后在R控制台下输入

```
data<-read.table(file="clipboard",sep="\t",header=T)

data<-read.table(
file="clipboard",sep="\t",header=F,
colClasses = "numeric",na.strings = "-")
```

分隔使用tab

写入数据到剪切板(windows),

[1]有时出现读取失败. 错误: readTableHeader在读取'clipboard'时遇到了不完全的最后一行

```
write.table(data,file="clipboard",sep="\t",col.names=NA)
```

linux用户需要设置选项'pipe("xclip -i", "w")', 没有成功. 参考help(file)

5.2.4 scan()函数-读取大数据

打开很大的数据建议使用scan函数, 因为可以设定数据类型, 而不是读取完毕才检查数据类型的一致性.

例如文件名称为"filename",默认当前路径(即你运行R的路径), 是",",分隔的数据, 想读入364行, 每行有5个数, 因为scan()是按行读取的, 矩阵的顺序是列为先的, 所以应该象下面, 先将矩阵设置为5行, 364列, 然后转置即可得到想要的结果

```
data<-t(matrix(scan("fileName",sep=','), 5, 364))
```

5.2.5 导出/保存

5.2.6 向文件写入数据

write.table 可以在文件中写入一个对象, 一般是写一个数据框, 也可以是其他数据类型, 向量, 矩阵...

write(x, file="data.txt") 简单的的将对象写入文件. 选项 append 缺省为删除已存在的数据. 默认5列, 字符为1列. 需要修改使用 ncol选项.

5.2.7 保存为R格式

要记录一组任意数据类型的对象, 我们可以使用命令save(x, y, z, file= "xyz.RData"). 可以使用选项ASCII=TRUE使得

数据在不同的机器之间更简易转移. 数据（用R的术语来说叫做工作空间）.

函数save.image()是save(list =ls(all=TRUE) file=".RData")的一个简捷方式.

可以在使用load("xyz. RData")之后被加载到内存中.

5.2.8 重定向输出

```
> sink("record.lis")
```

将输出重定向到文件 "record.lis".

```
> sink()
```

让你的输出流重新定向到控制台。

5.2.9 其它格式(SPSS, SAS, Stata and minitab)

包 foreign 提供读取SPSS, SAS, Stata and minitab格式的数据

```
>  library(foreign)
> search()
 [1] ".GlobalEnv"        "package:foreign"   "package:nlme"
 [4] "package:stats"     "package:graphics"  "package:grDevices"
 [7] "package:utils"     "package:datasets"  "package:methods"
[10] "Autoloads"         "package:base"
> help(read.spss)
```

5.2.10 latex

library(xtable) 可以把矩阵, data.frame 等变换为 latex 格式.

5.3 基本操作

5.3.1 产生序列

```
> x <- 1:9 # 初始化
[1] 1 2 3 4 5 6 7 8 9
> x <- seq(1,10,by=0.1)
```

5.3.2 where are they?

```
> x==3 # where are they?
[1] FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
```

5.3.3 what are they?

```
> which(x==3) # what are they?
[1] 3
```

5.3.4 各种计数

```
> length(x) # how many elements?
[1] 9
> sum(x>3)
[1] 7
```

```
> x>3
[1] FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
> sum(x>7|x<3)
[1] 5
```

5.3.5 反转序列

```
> p <- rev(x) # reverse element
> p
[1] 9 8 7 6 5 4 3 2 1
> p[x==3] # logical extraction. Very useful
[1] 7
> x[x>4]
> x = c(45,43,46,48,51,46,50,47,46,45)
```

5.3.6 取得变量的一部分

```
> x=rnorm(100)
> y=x[x<1] # y 为 x < 1 的值

# 返回除第1,2,3个元素的其它元素, 即相当于删除1,2,3个元素
> x[-c(1,2,3)]
```

5.3.7 删除变量

```
> rm(x) # x=NULL 不删除, 值为NULL
```

5.3.8 过滤缺失值(missing values)

如果有缺失值, 则使用下述方法过滤

```
> x[!is.na(x)]
```

5.3.9 apply 的用法

apply 是针对矩阵的, 返回向量. 参数 margin指明函数应用于行,列. margin=1 为行, =2为列

lapply 返回的是 list, 需要用 [[ ]] 下标的方法取得结果

tapply 看了看帮助, 应该是针对基本数据结构, 对其指定的因子水平(INDEX参数)执行FUN函数. 看例子应该比较明显.

rapply 是 lapply 的递归执行

apply 第二个参数为 1, 作用于行, 2 则作用于列. c(1,2) 作用于每个值

```
# 两个向量 euclidean 距离
dist.euclidean <- function(x,y){
   res <- sqrt(sum((x-y)^2))
   res
}

# x1,x2 为点的坐标, g 为其所属类别
 X=data.frame(
   x1=c(4,1,3,3,7,4,6,5,3,6,4,4,5,7,5,10,7,4,9,5,8,6,7,8),
   x2=c(3,3,3,7,4,1,5,6,7,2,6,4,8,8,6,5,6,10,7,4,5,6,4,8),
   g=c(rep(1,10),rep(2,14)) )
> X
  x1 x2 g
1  4  3 1
2  1  3 1
```

```
3   3  3 1
4   3  7 1
5   7  4 1
6   4  1 1
7   6  5 1
8   5  6 1
9   3  7 1
10  6  2 1
11  4  6 2
12  4  4 2
13  5  8 2
14  7  8 2
15  5  6 2
16 10  5 2
17  7  6 2
18  4 10 2
19  9  7 2
20  5  4 2
21  8  5 2
22  6  6 2
23  7  4 2
24  8  8 2

# 下面画图看看
# red 为类1, blue 为类2.
> plot(x2~x1,col=c("red","blue")[g],data=X)
# 新样本(6,5)为 "*" 标记
> points(6,5,pch=8,cex=3)


# 计算新样本 (6,5) 与 X 中坐标 (x1, x2) 的距离的平方
> d<-apply(X[,1:2],1,dist.euclidean,y=c(6,5))^2
> d
 [1]  8 29 13 13  2 20  0  2 13  9  5  5 10 10  2 16  2 29 13  2  4  1  2 13

# 联合起来
> d1<-cbind(d,X$g)
> d1
      d
 [1,]  8 1
 [2,] 29 1
```

```
 [3,] 13 1
 [4,] 13 1
 [5,]  2 1
 [6,] 20 1
 [7,]  0 1
 [8,]  2 1
 [9,] 13 1
[10,]  9 1
[11,]  5 2
[12,]  5 2
[13,] 10 2
[14,] 10 2
[15,]  2 2
[16,] 16 2
[17,]  2 2
[18,] 29 2
[19,] 13 2
[20,]  2 2
[21,]  4 2
[22,]  1 2
[23,]  2 2
[24,] 13 2

# 按照距离排序
> o<-order(d1[,1])
> d2<-d1[o,]
> d2
      [,1] [,2]
 [1,]    0    1
 [2,]    1    2
 [3,]    2    1
 [4,]    2    1
 [5,]    2    2
 [6,]    2    2
 [7,]    2    2
 [8,]    2    2
 [9,]    4    2
[10,]    5    2
[11,]    5    2
[12,]    8    1
[13,]    9    1
```

```
[14,]   10    2
[15,]   10    2
[16,]   13    1
[17,]   13    1
[18,]   13    1
[19,]   13    2
[20,]   13    2
[21,]   16    2
[22,]   20    1
[23,]   29    1
[24,]   29    2
```

5.3.10 attach 的用法

```
> attach(x) # x 包含的向量纳入搜索空间，可以直接使用了
> x1
[1] 1 2 3 4 5 6 7 8 9
> detach(x)
> x1
错误: 找不到这个目标对象"x1"
```

5.3.11 总结

```
how many elements?                 length(x)
ith element                       x[2] (i = 2)
all but ith element                x[-2] (i = 2)
?rst k elements                   x[1:5] (k = 5)
last k elements                   x[(length(x)-5):length(x)] (k = 5)
speci?c elements.                  x[c(1,3,5)] (First, 3rd and 5th)
all greater than some value         x[x>3] (the value is 3)
bigger than or less than some values x[ x< -2 | x > 2]
which indices are largest           which(x == max(x))
```

5.3.12 两个数据操作

```
> x = c(1,3,5,7,9)
> y = c(2,3,5,7,11,13)
> x+1
[1]  2  4  6  8 10
> y*2
[1]  4  6 10 14 22 26
> y[-3] # 去掉第三个
[1]  2  3  7 11 13
> y[x]
[1]  2  5 11 NA NA

> x = 1:10
> y=1:3
> y[4]=NA
> y
[1]  1  2  3 NA
> x[y]
[1]  1  2  3 NA
```

5.4 使用data.frame

数据框（data frame）也是列表, 是一个属于"data.frame" 类的列表。不过，对于可能属于数据框的列表对象有一些限制条件.

分量必须是向量(数值, 字符, 逻辑)，因子，数值矩阵，列表或者其他数据框; 每列的行数必须相等.

数据框常常会被看作是一个由不同模式和属性的列构成的矩阵。它能以矩阵形式出现，行列可以通过矩阵的索引习惯访问。

5.4.1 产生 data.frame

```
> weight = c(150, 135, 210, 140)
> height = c(65, 61, 70, 65)
> gender = c("Fe","Fe","M","Fe")
> study = data.frame(weight,height,gender)
```

5.4.2 行列的变量名称

列的名称可以在赋值的时候指定, 也可以用下面的方法

```
>  study = data.frame(w=weight,h=height,g=gender)
>  row.names(study)<-c("Mary","Alice","Bob","Judy")
> names(study) <- c("wei","hei","gen")
```

5.4.3 取得数据的各种方法

取得行, 列的数据

```
> study[,"wei"]
[1] 150 135 210 140
> study[,1:2]
   wei hei
Mary  150  65
Alice 135  61
Bob   210  70
Judy  140  65
> study["Mary",]
   wei hei gen
Mary 150  65  Fe
> study["Mary","wei"]
[1] 150
```

使用 $ 符号

```
> study$wei
[1] 150 135 210 140
```

使用名称及缩写

```
> study[["wei"]]
[1] 150 135 210 140
> study["wei"]
   wei
Mary  150
Alice 135
Bob   210
Judy  140
> study["w"]
错误在"[.data.frame"(study, "w") : 选择了未定义的列
> study[["w"]]
[1] 150 135 210 140
```

使用 index(下标)

```
> study[1]
   wei
Mary  150
Alice 135
Bob   210
Judy  140
> study[[1]]
[1] 150 135 210 140
```

5.4.4 条件取得数据

```
> study[study$gen=="Fe",]
```

```
   wei hei gen
Mary  150  65  Fe
Alice 135  61  Fe
Judy  140  65  Fe
```

5.4.5 使用 stack 与 unstack

stack 是把一个 data.frame 连接成为两列, 一列为数据, 另外一列为数据原来的列名称.

unstack 相反, 把一列数据按照因子(水平)分离为不同的列. 如果每列数量相等, 则强制为 data.frame, 否则为 list. 默认第一列为数据, 第二列为因子. 如果不是这个顺序的话, 需要 form 参数(模型)

```
> l <- list()
> x <- c("y","n")
> i <- sample(x,10,replace=TRUE)
> i
[1] "y" "n" "y" "y" "y" "n" "n" "y" "y" "n"
> a <- rep("y",5)
> b <- rep("n",5)
> c <- c(a,b)
> l <- list()
> l$ind <- i
> l$val <- rnorm(10)
> unstack(l,form=val~ind) # 注意用法, 第二个参数为 form, val~ind 也可以为 "val~ind"
$n
[1]  0.424591771  0.004047361 -1.208147843 -0.516055218

$y
[1] -0.0708544  0.5732878 -0.6390650 -0.6262143 -0.1372453  0.2929985

> l$ind <- c
> unstack(l,form=val~ind)
          n         y
```

```
1  0.004047361 -0.0708544
2 -1.208147843  0.4245918
3 -0.137245323  0.5732878
4  0.292998458 -0.6390650
5 -0.516055218 -0.6262143
```

如果顺序为默认的话, 不需要form参数

```
> l1 <- list()
> l1$val <- l$val
> l1$ind <- l$ind
> unstack(l1)
错误在inherits(object, "formula") : 缺少变元"form",也没有缺省值
> unstack(data.frame(l1))
          n        y
1  0.004047361 -0.0708544
2 -1.208147843  0.4245918
3 -0.137245323  0.5732878
4  0.292998458 -0.6390650
5 -0.516055218 -0.6262143
```

绘图(boxplot时候, 需要把list 转换为 data.frame, 即使赋值后也是如此)

```
> boxplot(unstack(data.frame(l1)))
> boxplot(l$val~l$ind)
错误在model.frame(formula, rownames, variables, varnames, extras, extranames,  :
      变数种类不对
> boxplot(l1$val~l1$ind)
错误在model.frame(formula, rownames, variables, varnames, extras, extranames,  :
      变数种类不对

> d <- data.frame(l1)
> boxplot(d$val~d$ind)
```

```
> x <-d$val
> y<-d$ind
> boxplot(x~y)
> x <-l$val
> y<-l$ind
> boxplot(x~y)
错误在model.frame(formula, rownames, variables, varnames, extras, extranames,  :
      变数种类不对
```

5.4.6 删除某列

```
> x=data.frame(a=c(1:3),b=c(2:4))
> x
  a b
1 1 2
2 2 3
3 3 4

# 下面方法不能同时删除多列, 且不能删除变量, 列全部删除后其值为NULL.
# 若想删除变量, 使用 rm, remove
# 如果想删除多列, 应该从后往前删除, 否则前面的删除后, 后面的会前移.
# 或者直接使用列名称删除, 保险一点
> x[1]<-NULL
> x
  b
1 2
2 3
3 4
> x[1]<-NULL
> x
NULL data frame with 3 rows
```

5.5 多元数据操作

本部分来自参考文献[41]《Multilevel Modeling in R》的翻译.

主要使用的包为: base, nlme, 数据来自 multilevel 包

看看数据

```
> library(multilevel)
> data(package="multilevel")
> data(cohesion)
> cohesion
    UNIT PLATOON COH01 COH02 COH03 COH04 COH05
1  1044B     1ST     4     5     5     5     5
2  1044B     1ST     3    NA     5     5     5
3  1044B     1ST     2     3     3     3     3
4  1044B     2ND     3     4     3     4     4
5  1044B     2ND     4     4     3     4     4
6  1044B     2ND     3     3     2     2     1
7  1044C     1ST     3     3     3     3     3
8  1044C     1ST     3     1     4     3     4
9  1044C     2ND     3     3     3     3     3
10 1044C     2ND     2     2     2     3     2
11 1044C     2ND     1     1     1     3     3
```

5.5.1 数据整合(merge)

例如有另外一个变量是platoon的大小, 我们想合并到数据cohesion中, 使用merge()函数

```
> group.size<-data.frame(UNIT=c("1044B","1044B","1044C","1044C"),
    PLATOON=c("1ST","2ND","1ST","2ND"),PSIZE=c(3,3,2,3))
> group.size
   UNIT PLATOON PSIZE
1 1044B     1ST     3
2 1044B     2ND     3
```

```
3 1044C    1ST    2
4 1044C    2ND    3
# 合并依据"UNIT","PLATOON"
>  new.cohesion<-merge(cohesion,group.size,by=c("UNIT","PLATOON"))
>  new.cohesion
   UNIT PLATOON COH01 COH02 COH03 COH04 COH05 PSIZE
1  1044B    1ST    4    5    5    5    5    3
2  1044B    1ST    3   NA    5    5    5    3
3  1044B    1ST    2    3    3    3    3    3
4  1044B    2ND    3    4    3    4    4    3
5  1044B    2ND    4    4    3    4    4    3
6  1044B    2ND    3    3    2    2    1    3
7  1044C    1ST    3    3    3    3    3    2
8  1044C    1ST    3    1    4    3    4    2
9  1044C    2ND    3    3    3    3    3    3
10 1044C    2ND    2    2    2    3    2    3
11 1044C    2ND    1    1    1    3    3    3
```

5.5.2 合计(aggregate)

按照分组情况合计, 使用函数aggregate().

```
# 将第3,4列按照UNIT和PLATOON的分组情况统计平均值
TEMP<-aggregate(cohesion[,3:4],
   list(cohesion$UNIT,cohesion$PLATOON),mean)
> TEMP
  Group.1 Group.2    COH01    COH02
1   1044B     1ST 3.000000       NA
2   1044C     1ST 3.000000 2.000000
3   1044B     2ND 3.333333 3.666667
4   1044C     2ND 2.000000 2.000000
# 按照UNIT合计, 并去除na数据
> TEMP<-aggregate(cohesion[,3:4],list(cohesion$UNIT),mean,na.rm=T)
> TEMP
  Group.1    COH01 COH02
1   1044B 3.166667   3.8
2   1044C 2.400000   2.0
```

5.5.3 按照合计情况再合并

合并统计结果

```
> names(TEMP)<-c("UNIT","PLATOON","G.COH01","G.COH02")
> final.cohesion<-merge(new.cohesion,TEMP,
   by=c("UNIT","PLATOON"))
> final.cohesion
   UNIT PLATOON COH01 COH02 COH03 COH04 COH05 PSIZE  G.COH01  G.COH02
1  1044B     1ST     4     5     5     5     5     3 3.000000       NA
2  1044B     1ST     3    NA     5     5     5     3 3.000000       NA
3  1044B     1ST     2     3     3     3     3     3 3.000000       NA
4  1044B     2ND     3     4     3     4     4     3 3.333333 3.666667
5  1044B     2ND     4     4     3     4     4     3 3.333333 3.666667
6  1044B     2ND     3     3     2     2     1     3 3.333333 3.666667
7  1044C     1ST     3     3     3     3     3     2 3.000000 2.000000
8  1044C     1ST     3     1     4     3     4     2 3.000000 2.000000
9  1044C     2ND     3     3     3     3     3     3 2.000000 2.000000
10 1044C     2ND     2     2     2     3     2     3 2.000000 2.000000
11 1044C     2ND     1     1     1     3     3     3 2.000000 2.000000
```

5.5.4 去掉重复(unique)

使用数据bhr2000, 详细解释见帮助.

```
> help(bhr2000)
> data(bhr2000,package="multilevel")#puts data in working environment
# GRP是分组情况
> names(bhr2000)
 [1] "GRP"   "AF06"  "AF07"  "AP12"  "AP17"  "AP33"  "AP34"  "AS14"  "AS15"
[10] "AS16"  "AS17"  "AS28"  "HRS"   "RELIG"
# 共有5400行数据
> nrow(bhr2000)
```

```
[1] 5400
# 看看有共多少组
> length(unique(bhr2000$GRP))
[1] 99

# 去掉重复
> a=data.frame(matrix(1:9,nc=3))
> a
  X1 X2 X3
1  1  4  7
2  2  5  8
3  3  6  9
> a<-rbind(a,a)
> a
  X1 X2 X3
1  1  4  7
2  2  5  8
3  3  6  9
4  1  4  7
5  2  5  8
6  3  6  9
> unique(a,MARGIN = 1)
  X1 X2 X3
1  1  4  7
2  2  5  8
3  3  6  9
```

5.6 排序

order返回其第一个参数的排序结果。如果有结（顺序相同的元素），根据其他参数决定结的排列顺序。

第一个向量的结使用第二个向量来排序，如果第二个向量还有结，使用第三个.....排序算法是稳定的（除非指定"method=quick"），所以任何最后未能排序的结都会保持其原来的顺序。

复数首先根据实数排序，然后根据虚数排序。

推荐使用默认的排序方法。参数'quick'排序快，但是会打乱结的顺序。参数'radix'只用于小于 100,000 的整数，但是非常快，且对结稳定，故对于排序因子非常有用。

参数 'partial' 是为了和 S 兼容。

注意因子总是根据其内码（levels）排序，并非你看到的打印顺序。

字符向量根据其字符编码的不同而顺序不同，有时候结果会让人吃惊。例如'en_US' 与 'C' 编码的字符的顺序就不同；爱沙尼亚语（Estonian）'Z' 在 'S' and 'T'之间；丹麦语（Danish）'aa'被看作一个字符，并且在'z'后；威尔士（Welsh）语中'ng'可能是一个字符，也可能不是：如果出现在'g'之后；等等。所以，不要猜测字符的顺序。请参考 'Comparison' 和 'locales'。获取和设置文字编码方式使用

```
␣␣␣␣␣Sys.getlocale(category␣=␣"LC_ALL")
␣␣␣␣␣Sys.setlocale(category␣=␣"LC_ALL",␣locale␣=␣"")

>␣Sys.getlocale(category␣=␣"LC_ALL")
[1]␣"LC_CTYPE=zh_CN.UTF-8;LC_NUMERIC=C;LC_TIME=zh_CN.UTF-8;LC_COLLATE=zh_CN.UTF-
␣␣␣␣


>␣x
[1]␣1␣1␣3␣2␣1␣1␣2␣3␣4␣3
>␣y
[1]␣9␣9␣8␣7␣6␣5␣4␣3␣2␣1
>␣z
[1]␣2␣1␣2␣3␣4␣5␣6␣7␣8␣9

#␣x有结␣1␣2␣3
#␣象我们希望的那样，要对x升序，相同的y降序，再相同的z升序排序（帮助的例子）
>␣d2<-data.frame(x,y,z)
>␣d2[␣order(x,␣-y,␣z)␣,]
␣␣x␣y␣z
```

```
2 1 9 1
1 1 9 2
5 1 6 4
6 1 5 5
4 2 7 3
7 2 4 6
3 3 8 2
8 3 3 7
10 3 1 9
9 4 2 8
# z使用内码顺序排序
> dd <- transform(data.frame(x,y,z),
+ z = factor(z, labels=LETTERS[9:1]))
> z
 [1] H I H G F E D C B A
Levels: I H G F E D C B A
> dd
   x y z
1  1 1 H
2  1 1 I
3  3 8 H
4  2 7 G
5  1 6 F
6  1 5 E
7  2 4 D
8  3 3 C
9  4 2 B
10 3 1 A
> dd[ order(x, -y, z) ,] # 实际上使用的是z的factor levels的顺序排序
   x y z
2  1 1 I
1  1 1 H
5  1 6 F
6  1 5 E
4  2 7 G
7  2 4 D
3  3 8 H
8  3 3 C
10 3 1 A
9  4 2 B
```

```
> d3 <- transform(data.frame(x,y,letters[1:10]))
> d3
   x y letters.1.10.
1  1 9                a
2  1 9                b
3  3 8                c
4  2 7                d
5  1 6                e
6  1 5                f
7  2 4                g
8  3 3                h
9  4 2                i
10 3 1                j
> order(d3) # 21-30按照字母顺序，数字比字母的顺序号小
 [1]  1  2  5  6 20  4  7 19  3  8 10 18  9 17 16 15 14 13 11 12 21 22 23 24 25
[26] 26 27 28 29 30

    
```

5.7 对象

R操作的实体在技术上来说就是对象(object).

R的对象类型包括数值型（numeric），复数型（complex），逻辑型（logical），字符型（character）和原味型（raw）.

5.7.1 对象的模式

一个对象的模式(mode)是该对象基本要素的类型. 所有对象都有的特征是长度(length).

空对象仍然有其模式.

```
> s=character()
> s
```

```
character(0)
> mode(s)
[1] "character"
> typeof(s)
[1] "character"

> e=numeric()
> e
numeric(0)
> mode(e)
[1] "numeric"
> typeof(e)
[1] "double"
```

5.7.2 对象函数

函数mode(object), typeof(object), length(object)可以用于任何数据对象以得到其模式和长度.

typeof是R自己独立的函数, 保留mode是为了和S兼容.

```
> x=1+2i
> x
[1] 1+2i
> mode(x)
[1] "complex"
> typeof(x)
[1] "complex"
> length(x)
[1] 1
```

5.7.3 获取和改变对象属性-类

attributes(object), str(object)

attr(object, name)

```
> attr(z, "dim") <- c(10,10) # 允 许 R 把z 当 作 一 个10×10 的 矩
阵 。

> x
 [1] 0 1 0 0 0 1 1 1 0 0

> y=table(x)
> y
x
0 1
6 4
> attributes(y)
$dim
[1] 2

$dimnames
$dimnames$x
[1] "0" "1"


$class
[1] "table"

> str(y)
 int [, 1:2] 6 4
 - attr(*, "dimnames")=List of 1
  ..$ x: chr [1:2] "0" "1"
 - attr(*, "class")= chr "table"
> y[,1]
错 误 在y[, 1] : 量 度 数 目 不 对
> y[1]
0
6
> y[2]
1
4
> class(y) # 确 定y的class
[1] "table"
```

```
> dim(y)
[1] 2
> dimnames(y)
$x
[1] "0" "1"

> dimnames(y)$x
[1] "0" "1"
```

5.7.4 模式转换

参考 help(as)

```
> as(x,"character")
 [1] "0" "1" "0" "0" "0" "1" "1" "1" "0" "0"
> as.character(x)
 [1] "0" "1" "0" "0" "0" "1" "1" "1" "0" "0"

> s=as.character(x)
> s
 [1] "0" "1" "0" "0" "0" "1" "1" "1" "0" "0"
> as.numeric(s)
 [1] 0 1 0 0 0 1 1 1 0 0
```

Chapter 6

绘图

参考文献[46] 中有丰富的图.

http://addictedtor.free.fr/graphiques/ 介绍有大量的R绘图功能。

由于cairo版本低(linux版本足够高的话一般没有这个问题), 绘制到设备(一般为图形文件), bmp, jpeg 都不能使用. 只能使用

```
png(type="cairo1")
```

例如

```
> png(type="cairo1",)
> x=1:10
> y=x
> plot(y~x)
> dev.off()
null device
        1
```

```
png(file="myplot.png", bg="transparent",type="cairo1")
    plot(1:10)
    rect(1, 5, 3, 7, col="white")
```

```
    dev.off()

## will make myplot1.jpeg and myplot2.jpeg
    jpeg(file="myplot%d.jpeg")
    example(rect)
    dev.off()
```

查看当前文件夹, 有一个png文件是此图形.

6.1 图形环境设置-par函数

参考 R-导论 图形工具 部分

6.1.1 设置margin大小

```
> op <- par(mar=c(3,4,2,2)+.1)
```

6.1.2 设置显示区域

```
> plot(-4:4, -4:4, type = "n")
```

6.1.3 绘制到文件

视系统支持情况, 可以绘制到pdf,ps,png,jpeg等格式的文件中.

```
> pdf("aa.pdf") # 打开绘图设备--一个pdf文件
> plot(c(1,2,3)) # 绘制
```

> dev.off() # 关闭当前打开的绘图设备, 然后可以到当前目录下看看有没有aa.pdf这个文件. 打开看看, 是不是你要绘制的图形?

6.2 坐标轴

6.2.1 轴和刻度

axis() 设置自己的坐标轴.

例如, 需要 y 轴按照 200 为单位显示

```
x=c(4.47,3.16,-2.24,-1.58,2.24,3.16,1.1,-1.1,0.77,1.73,3.81,2.25,-1.3,-1,-2.24,1
y=c(50,120,210,240,350,360,610,630,800,900,910,920,1100,1210,1300,1340,1350,1370
par(lab=c(7,15,12),las=1)
plot(x,y)
```

lab 前两个参数分别是x 和y 轴期望的刻度间隔数目。第三个参数刻度标记的字符长度（包括小数点）。这个参数设的太小会导致所有的标记变成一样的数字。

las 刻度标记的方向。0 表示总是平行于坐标轴, 1 表示总是水平, 2 表示总是垂直于坐标轴。

mgp=c(3,1,0) 三个坐标成分的位置。第一个参数是轴标签相对轴位置的距离, 以文本行作为参照单位的。第二个参数表示刻度标记的距离, 最后一个参数是轴位置到轴线的距离(常常是0)。正值表示在图形外, 负值表示在图形内。

tck=0.01 刻度的长度, 以画图区域大小的比率作为度量。当tck 比较小(小于0.5), x 和y 轴上的刻度强制大小一致。值为1时, 给出网格线。负值时刻度在图形外。tck=0.01 和mgp=c(1,-1.5,0) 表示内部刻度。

6.2.2 自定义坐标轴label

```
axis(side=1, 1:6, tcl=-0.2, labels=c("0.5h","1h","1.5h","2h","3h","4h"))
```

6.3 多图和多组数据

6.3.1 同时绘制多组数据

使用rbind与cbind连接再用plot绘制

```
> x=rnorm(10)
> y=rnorm(10)+3
> x1=rnorm(10)
> y1=rnorm(10)+5

> a=cbind(x,y)
> b=cbind(x1,y1)
> d=rbind(a,b)

> col=c(rep("red",10),rep("blue",10))

> plot(d,col=col)
```

6.3.2 points添加点

6.3.3 一页上绘制多个图

```
n <- 100
v <- .1
x1 <- rlnorm(n)
x2 <- rlnorm(n)
x3 <- rlnorm(n)
x4 <- x1 + x2 + x3 + v*rlnorm(n)
```

```
m2 <- cbind(x1,x2,x3,x4)

op<-par(mfrow=(c(2,2)))
for (i in 1:4){
   plot(m2[,i])
}
par(op)
```

6.3.4 在一幅图上添加另外一幅图

使用 par(fig=...,new=TRUE)

```
> n <- 1000
> x <- rnorm(n)
> qqnorm(x)
> qqline(x, col="red")
> op <- par(fig=c(.02,.5,.5,.98), new=TRUE)
> hist(x, probability=T,
+     col="light blue", xlab="", ylab="", main="", axes=F)
> lines(density(x), col="red", lwd=2)
> box()
> op
$fig
[1] 0 1 0 1

$new
[1] FALSE

> par(op)
```

6.4 文本相关

6.4.1 文字旋转

opt¡-par() par(srt=45) text(x=1,y=1,'xxxxxxxx') par(opt)

6.4.2 坐标轴文本及自定义标题文字大小

绘图时不要加xlab,ylab等, 绘制完成然后使用mtext添加

6.4.3 字体

font=2 整数是用来指定用于文中的字体类型。一般情况下，设备驱动设定的1 对应于纯文本，2 对应粗体，3 对应斜体，4 对应粗斜体，5 对应符号体(包括希腊字母)。

6.5 添加自定义图例

另外自定义图例

```
# 加入图例, 1,1.8是图例的位置, legend是文字, pch是对应的图形
legend(1,1.8,legend=c("Mock","Eai","GFPi"),pch=15:17)
```

6.6 lines

line type: lty; line width: lwd; color col;

```
>  X=c(10.0, 8.0, 13.0, 9.0, 11.0, 14.0, 6.0, 4.0, 12.0, 7.0, 5.0),
>  Y2=c(9.14,8.14, 8.74,8.77,9.26,8.10,6.13,3.10, 9.13,7.26,4.74),

> plot(Y2~X)

# 下面绘制曲线, 注意把顺序调好.
> o <- order(X)
X.o <- X[o]
> Y2.o<-Y2[o]
> lines(X.o,Y2.o,col="red")
```

6.7 boxplot 水平放置

加入参数 horizontal=TRUE

6.8 添加水平或垂直线

垂直线: abline(v=c(...),...)

水平线: abline(h=c(...),...)

```
> x <- rnorm(100)
> plot(1:100~sort(x))
> abline(v = quantile(x), col = "blue", lwd = 3, lty=2)
```

6.9 xy轴反转

```
> x <- rnorm(100)
> plot(1:100~sort(x))
```

6.10 rug-在一边加入显示密度的小短线

```
> x <- rnorm(100)
> plot(sort(x))
> rug(x,side=2)
```

6.11 绘制到x轴的垂直线

加入参数 type = ”h”

6.12 spline–平滑差值

```
x=c(0.001,0.01,0.1,1,10,100␣)
y=1:6+rnorm(6)
plot(y~log(x,10),xaxt="n")␣#␣x␣轴无坐标值
#␣spline␣使用多项式或样条函数差值产生平滑曲线的点.␣注意此处不能同拟合曲线混淆.
lines(spline(log(x,10),y))
```

6.13 curve–绘制函数曲线

用法: curve(expr, from, to, n = 101, add = FALSE, type = ”l”, ylab = NULL, log = NULL, xlim = NULL, ...)

绘制函数曲线, expr 为一个函数表达式.

用于添加曲线很方便.

```
> curve(sin(x),-10,10)
> curve(dnorm(x),-3,3)
```

6.14 平滑曲线(density)的绘制

可以选择平滑方法: ”’gaussian”’, ”’rectangular”’, ”’triangular”’, ”’epanechnikov”’, ”’biweight”’, ”’cosine”’ or ”’optcosine”’, with default ”’gaussian”

```
> data(faithful)
> attach(faithful)
> hist(eruptions,15,prob=T)
> lines(density(eruptions))
```

6.15 填充颜色

```
x=seq(-4,4,by=0.1)
y=dnorm(x)
x1=seq(3,4,by=0.1)
x1=x[(length(x)-20):length(x)]
y1=y[(length(x)-20):length(x)]
x2=c(x1,x1[length(x1):1])
y2=c(y1,rep(0,length(x1)))
plot(x,y,type='l')
polygon(x2,y2,col="red")
```

6.16 cex-绘制按照比例大小的图标

```
> x1=rnorm(100)
> x2=rnorm(100)
> x3=rnorm(100)
> m=cor(cbind(x1,x2,x3))
> m
           x1         x2         x3
x1  1.00000000 -0.01499516 0.24657311
x2 -0.01499516  1.00000000 0.07323174
x3  0.24657311  0.07323174 1.00000000
> class(m)
[1] "matrix"
> plot(col(m), row(m), cex=10*abs(m),xlim=c(0, dim(m)[2]+1),ylim=c(0, dim(m)[1]+1
> col(m)
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    1    2    3
[3,]    1    2    3
> row(m)
     [,1] [,2] [,3]
[1,]    1    1    1
[2,]    2    2    2
[3,]    3    3    3
```

6.17 同时绘制不同数据不同颜色的图

```
> n <- 100
> x <- runif(n)
> z <- ifelse(x>.5,1,0)
> y <- 2*z -x + .1*rnorm(n)
> plot( y~x, col=c('red','blue')[1+z] )
```

6.18 等高线图(contour)

将一个矩阵的等高线绘制出来. 不论矩阵的数据是什么. 参数 lev 指明要绘制哪些线

```
> z=matrix(rnorm(10000),100,100)
> dim(z)
[1] 100 100
> contour(z, lev=seq(0.1,0.5)
> contour(1:100,1:100,z) # 一样的, 只是改变了x,y轴的坐标表示
> contour(1:100,1:100,z,lev=c(0.1,0.5)) # 只绘制数据为0.1,0.5的线
> contour(1:100,1:100,z,lev=0.1) # 绘制0.1等高线
> contour(1:100,1:100,z,lev=2,add=T,col='red') # 增添0.5等高线(红色)
```

6.19 数学方程式

参考 plotmath

```
http://hosho.ees.hokudai.ac.jp/~kubo/Rdoc/library/grDevices/html/plotmath.html
```

parse()将字符串转换为未求值的表达式, 然后可以使用 eval 来计算.

```
parse(text="0==1")
eval(parse(text="0==1"))

> parse(text="0==1")
expression(0==1)
attr(,"srcfile")
```

```
<text>
> eval(parse(text="0==1"))
[1] FALSE
```

expression() 可以作为绘图的label表示出数学符号.

```
x=1:100
y=sqrt(x)
plot(y~x,main=expression(y==sqrt(x)))

z=log(x)
xtext=expression(paste(log[2], "(some text)"))
ytext=expression(paste(log[2], "(some text)"))
plot(z~x,xlab=xtext,ylab=ytext)

text(7,5,expression(log[2] (some text)))
text(4, 9, expression(hat(beta) == (X^t * X)^{-1} * X^t * y))
text(4, 8.4, expression(hat(beta) == (X^t * X)^{-1} * X^t * y)
text(4, 7, expression(bar(x) == sum(frac(x[i], n), i==1, n)))
```

6.19.1 语法和更多例子

```
Syntax   Meaning

x + y
x plus y
x - y  x minus y
x*y  juxtapose x and y
x/y  x forwardslash y
x %+-% y  x plus or minus y
x %/% y  x divided by y
x %*% y  x times y
x %.% y  x cdot y
x[i]  x subscript i
x^2  x superscript 2
paste(x, y, z)  juxtapose x, y, and z
sqrt(x)  square root of x
```

```
sqrt(x, y)  yth root of x
x == y  x equals y
x != y  x is not equal to y
x < y  x is less than y
x <= y  x is less than or equal to y
x > y  x is greater than y
x >= y  x is greater than or equal to y
x %~~% y  x is approximately equal to y
x %=~% y  x and y are congruent
x %==% y  x is defined as y
x %prop% y  x is proportional to y
plain(x)  draw x in normal font
bold(x)  draw x in bold font
italic(x)  draw x in italic font
bolditalic(x)  draw x in bolditalic font
symbol(x)  draw x in symbol font
list(x, y, z)  comma-separated list
...  ellipsis (height varies)
cdots  ellipsis (vertically centred)
ldots  ellipsis (at baseline)
x %subset% y  x is a proper subset of y
x %subseteq% y  x is a subset of y
x %notsubset% y  x is not a subset of y
x %supset% y  x is a proper superset of y
x %supseteq% y  x is a superset of y
x %in% y  x is an element of y
x %notin% y  x is not an element of y
hat(x)  x with a circumflex
tilde(x)  x with a tilde
dot(x)  x with a dot
ring(x)  x with a ring
bar(xy)  xy with bar
widehat(xy)  xy with a wide circumflex
widetilde(xy)  xy with a wide tilde
x %<->% y  x double-arrow y
x %->% y  x right-arrow y
x %<-% y  x left-arrow y
x %up% y  x up-arrow y
x %down% y  x down-arrow y
x %<=>% y  x is equivalent to y
x %=>% y  x implies y
```

```
x %<=% y  y implies x
x %dblup% y  x double-up-arrow y
x %dbldown% y  x double-down-arrow y
alpha { omega  Greek symbols
Alpha { Omega  uppercase Greek symbols
theta1, phi1, sigma1, omega1  cursive Greek symbols
Upsilon1  capital upsilon with hook
aleph  first letter of Hebrew alphabet
infinity  infinity symbol
partialdiff  partial differential symbol
nabla  nabla, gradient symbol
32*degree  32 degrees
60*minute  60 minutes of angle
30*second  30 seconds of angle
displaystyle(x)  draw x in normal size (extra spacing)
textstyle(x)  draw x in normal size
scriptstyle(x)  draw x in small size
scriptscriptstyle(x)  draw x in very small size
underline(x)  draw x underlined
x ~~ y  put extra space between x and y
x + phantom(0) + y  leave gap for "0", but don't draw it
x + over(1, phantom(0))  leave vertical gap for "0" (don't draw)
frac(x, y)  x over y
over(x, y)  x over y
atop(x, y)  x over y (no horizontal bar)
sum(x[i], i==1, n)  sum x[i] for i equals 1 to n
prod(plain(P)(X==x), x)  product of P(X=x) for all values of x
integral(f(x)*dx, a, b)  definite integral of f(x) wrt x
union(A[i], i==1, n)  union of A[i] for i equals 1 to n
intersect(A[i], i==1, n)  intersection of A[i]
lim(f(x), x %->% 0)  limit of f(x) as x tends to 0
min(g(x), x > 0)  minimum of g(x) for x greater than 0
inf(S)  infimum of S
sup(S)  supremum of S
x^y + z  normal operator precedence
x^(y + z)  visible grouping of operands
x^{y + z}  invisible grouping of operands
group("(",list(a, b),"]")  specify left and right delimiters
bgroup("(",atop(x,y),")")  use scalable delimiters
group(lceil, x, rceil)  special delimiters
```

更多例子

```
require(graphics)

x <- seq(-4, 4, len = 101)
y <- cbind(sin(x), cos(x))
matplot(x, y, type = "l", xaxt = "n",
       main = expression(paste(plain(sin) * phi, "  and  ",
                               plain(cos) * phi)),
       ylab = expression("sin" * phi, "cos" * phi), # only 1st is taken
       xlab = expression(paste("Phase Angle ", phi)),
       col.main = "blue")
axis(1, at = c(-pi, -pi/2, 0, pi/2, pi),
    labels = expression(-pi, -pi/2, 0, pi/2, pi))

## How to combine "math" and numeric variables :
plot(1:10, type="n", xlab="", ylab="", main = "plot math & numbers")
theta <- 1.23 ; mtext(bquote(hat(theta) == .(theta)))
for(i in 2:9)
   text(i,i+1, substitute(list(xi,eta) == group("(",list(x,y),")"),
                          list(x=i, y=i+1)))
## note that both of these use calls rather than expressions.

plot(1:10, 1:10)
text(4, 9, expression(hat(beta) == (X^t * X)^{-1} * X^t * y))
text(4, 8.4, "expression(hat(beta) == (X^t * X)^{-1} * X^t * y)",
    cex = .8)
text(4, 7, expression(bar(x) == sum(frac(x[i], n), i==1, n)))
text(4, 6.4, "expression(bar(x) == sum(frac(x[i], n), i==1, n))",
    cex = .8)
text(8, 5, expression(paste(frac(1, sigma*sqrt(2*pi)), " ",
                            plain(e)^{frac(-(x-mu)^2, 2*sigma^2)})),
    cex = 1.2)

## some other useful symbols
plot.new(); plot.window(c(0,4), c(15,1))
text(1, 1, "universal", adj=0); text(2.5, 1,  "\\042")
text(3, 1, expression(symbol("\042")))
text(1, 2, "existential", adj=0); text(2.5, 2,  "\\044")
text(3, 2, expression(symbol("\044")))
```

```
text(1, 3, "suchthat", adj=0); text(2.5, 3,  "\\047")
text(3, 3, expression(symbol("\047")))
text(1, 4, "therefore", adj=0); text(2.5, 4,  "\\134")
text(3, 4, expression(symbol("\134")))
text(1, 5, "perpendicular", adj=0); text(2.5, 5,  "\\136")
text(3, 5, expression(symbol("\136")))
text(1, 6, "circlemultiply", adj=0); text(2.5, 6,  "\\304")
text(3, 6, expression(symbol("\304")))
text(1, 7, "circleplus", adj=0); text(2.5, 7,  "\\305")
text(3, 7, expression(symbol("\305")))
text(1, 8, "emptyset", adj=0); text(2.5, 8,  "\\306")
text(3, 8, expression(symbol("\306")))
text(1, 9, "angle", adj=0); text(2.5, 9,  "\\320")
text(3, 9, expression(symbol("\320")))
text(1, 10, "leftangle", adj=0); text(2.5, 10,  "\\341")
text(3, 10, expression(symbol("\341")))
text(1, 11, "rightangle", adj=0); text(2.5, 11,  "\\361")
text(3, 11, expression(symbol("\361")))
```

6.20 3D-绘图

rgl 程序包: 绘制 3D 图形必备. 其它仅做参考.

自带曲面函数: persp()

其它: misc3d

下面是绘制一个内核为红色球, 外面有一个大浅黄色球, 大球残缺1/4, 露出内核的图. 就像通常看的地球内部构造.

```
library(rgl)

# 绘制内核红色球
Sigma <- matrix(c(1,0,0,0,1,0,0,0,1), 3,3)
Mean <- c(0,0,0)
open3d()
plot3d( ellipse3d(Sigma/2, centre=Mean), col="red", alpha=1, add = TRUE)
```

```
# 绘制缺少1/4的大球
lat <- matrix(seq(90,-90, len=50)*pi/180, 50, 50, byrow=TRUE)
long <- matrix(seq(-90, 180, len=50)*pi/180, 50, 50)
r <- 10 # radius of it
x <- r*cos(lat)*cos(long)
y <- r*cos(lat)*sin(long)
z <- r*sin(lat)

persp3d(x, y, z, col="yellow",
      specular="black", axes=FALSE, box=FALSE, xlab="", ylab="", zlab="",
#normal_x=x, normal_y=y, normal_z=z,
      alpha=0.4,add=TRUE)

# 添加直径线
l=10
a=c(0,0)
b=c(0,0)
c=c(-l,l)
plot3d(a,b,c,col="black",add=TRUE,type="l")
```

6.21 箭头

```
library(diagram)
?curvedarrow
␣␣␣␣
```

6.22 热图(heatmap)

```
d=read.csv('data.csv',head=T,row.names=1)
x␣␣<-␣as.matrix(d)
rc␣<-␣rainbow(nrow(x),␣start=0,␣end=1)
cc␣<-␣rainbow(ncol(x),␣start=0,␣end=1)
col<-rainbow(256)
hv␣<-␣heatmap(x,␣col␣=␣col,␣scale="column",
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣RowSideColors␣=␣rc,␣ColSideColors␣=␣cc,␣margins=c(5,10),
```

```
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣xlab␣=␣"specification␣variables",␣ylab=␣"Car␣Models",
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣main␣=␣"heatmap")
```

6.23 venn 图

使用 gplots 包函数 venn(). 不能加颜色.

```
> d[1:10,]
         Tp Cr Cm OZ
OG2_1000:  0  1  0  0
OG2_1001:  0  1  0  0
OG2_1002:  0  1  0  0
OG2_1003:  1  1  1  1
OG2_1004:  0  1  0  1
OG2_1005:  1  0  0  0
OG2_1006:  1  1  0  0
OG2_1007:  1  1  0  0
OG2_1008:  1  1  1  1
OG2_1009:  1  1  1  1

library(gplots)
venn(d)
```

??? eVenn 包也可以, 但是参数path_list 不知道什么意思. 只有一个函数 evenn(). 绘图可以自动加不同颜色. 交集的个数也自动显示. 例子最漂亮.

venneuler 也可以加颜色,但是不能显示交集的个数.

Chapter 7

数据库接口–RMySQL

7.1 DBI

数据库基本接口，其他数据库几乎都用到

7.2 RMySQL

在表的大小不超过内存范围和没有特殊查寻需求的情况下，建议使用R的多元数据操作函数，例如unique, merge, aggregate 等. 参考多元数据操作 5.5.

具体使用 ?RMySQL 来查看，下面是帮助.

1. 连结到数据库

```
con <- dbConnect(MySQL(), group = "lasers")
con2 <- dbConnect(MySQL(), user="opto", password="pure-light",
            dbname="lasers", host="merced")
```

2. 列出表和列名称 (List tables and fields in a table)

```
dbListTables(con)
dbListFields(con, "table\_name")
```

3. 导 入,导 出 表 到 data.frames. 事 先 表 若 不 存 在. 会 自 动 创 建 表. overwrite=True 则 覆 盖 原 来 的 表. (Import and export data.frames)

```
d <- dbReadTable(con, "WL")
dbWriteTable(con, "WL2", a.data.frame)       ## table from a data.frame
dbWriteTable(con, "test2", "~/data/test2.csv") ## table from a file
```

具 体 使 用 见 ?dbWriteTable 其 它 常 用 参 数 还 有

```
‘header=’, ‘row.names=’,
‘col.names=’, ‘sep=’, ‘eol=’, ‘field.types=’,
‘skip=’, and ‘quote=’
```

4. 执 行SQL命 令, 并 将 结 果 返 回 给 data.frame (Run an arbitrary SQL statement and extract all its output (returns a data.frame))

```
dbGetQuery(con, "select count(*) from a\_table")
dbGetQuery(con, "select * from a\_table")
```

5. 执 行SQL命 令, 并 将 结 果 返 回 给 result set. (Run an SQL statement and extract its output in pieces (returns a result set))

```
rs <- dbSendQuery(con, "select * from WL where width\_nm between 0.5 and 1")
d1 <- fetch(rs, n = 10000)
d2 <- fetch(rs, n = -1)
```

6. 执 行 多 个SQL语 句, 处 理 结 果 集. (Run multiple SQL statements and process the various result sets (note the ‘client.flag’ value in the ‘dbConnect’ call))

```
con <- dbConnection(MySQL(), dbname = "rs-dbi",
       client.flag = CLIENT\_MULTI\_STATEMENTS)
script  <- paste("select * from WL where width\_nm between 0.5 and 1"
             "select * from lasers\_id  where id LIKE 'AL100
             sep = ";")
rs1 <- dbSendQuery(con, script)
d1 <- fetch(rs1, n = -1)
if(dbMoreResults(con)){
rs2 <- dbNextResult(con)
d2 <- fetch(rs2, n=-1)
}
```

7. 获取元信息. (Get meta-information on a connection (thread-id, etc.))

```
summary(MySQL(), verbose = TRUE)
summary(con, verbose = TRUE)
summary(rs, verbose = TRUE)
dbListConnections(MySQL())
dbListResultSets(con)
dbHasCompleted(rs)
```

8. 关闭连结. (Close connections)

```
dbDisconnect(con)
dbDisconnect(con2)
```

下面是几个例子。

```
> library(RMySQL) # will load DBI as well
## 打开一个MySQL数据库的连接

> summary(MySQL(), verbose = TRUE)
<MySQLDriver:(4616)>
  Driver name:  MySQL
  Max  connections: 16
```

```
  Conn. processed: 0
  Default records per fetch: 500
  DBI API version:  0.2-5
  MySQL client version:  5.1.58
  Open connections: 0

> con <- dbConnect(dbDriver("MySQL"), dbname = "taxonomy",user='xjx', password='11

> summary(con, verbose = TRUE)
<MySQLConnection:(9191,0)>
  User: xjx
  Host: localhost
  Dbname: taxonomy
  Connection type: Localhost via UNIX socket
  MySQL server version:  5.1.58-1ubuntu1
  MySQL client version:  5.1.58
  MySQL protocol version:  10
  MySQL server thread id:  43
  No resultSet available

> con <- dbConnect(dbDriver("MySQL"), dbname = "test",user='xxx', password='111111
## 列出数据库中表
> dbListTables(con)
## 把一个数据框导入到数据库，删除任何已经存在的拷贝
> data(USArrests)
> dbWriteTable(con, "arrests", USArrests, overwrite = TRUE)
TRUE
> dbListTables(con)
[1] "arrests"
## 获得整个表
> dbReadTable(con, "arrests")
          Murder Assault UrbanPop Rape
Alabama     13.2     236       58 21.2
Alaska      10.0     263       48 44.5
Arizona      8.1     294       80 31.0
Arkansas     8.8     190       50 19.5
...
## 从导入的表中查询
> dbGetQuery(con, paste("select row_names, Murder from arrests",
                        "where Rape > 30 order by Murder"))
row_names Murder
```

```
1   Colorado    7.9
2    Arizona    8.1
3 California    9.0
4     Alaska   10.0
5 New Mexico   11.4
6   Michigan   12.1
7     Nevada   12.2
8    Florida   15.4
# 删除表
> dbRemoveTable(con, "arrests")
> dbDisconnect(con)
```

Chapter 8

在 python 中调用 R (rpy2)

安装 rpy2

```
Web: http://rpy.sourceforge.net
```

rpy2 与 rpy1.x 使用方法有点不同. 详细参考见网站 user guide

8.1 introduction

下面是如何从 python 中调用 R 命令的例子. (网站 user guide 之 introduction 的部分)

```
# 导入
import rpy2.robjects as robjects
from  rpy2.robjects import r

# 使用常量
In [16]: r['pi']
Out[16]: 3.14159265358979
```

```
In [17]: r('pi')
Out[17]: 3.14159265358979

# 返回的是 tuple, 获取必须使用下标.
# python 里的 add 函数对应 R 里的 c() 函数
In [18]: r('pi')+2
Out[18]: c(3.14159265358979, 2)

In [19]: r('pi')[0]+2
Out[19]: 5.1415926535897931

# 定义和使用函数
In [21]: r('''f <- function(r) { 2 * pi * r }''')
Out[21]:
function (r)
{
   2 * pi * r
}

In [3]: r('f')
Out[3]:
function (r)
{
   2 * pi * r
}

In [23]: r('f(3)') # r['f(3)'] 是错误的, 应该使用 r['f'](3)
Out[23]: 18.8495559215388

In [4]: r['f']
Out[4]:
function (r)
{
   2 * pi * r
}


# 执行一个字符串
In [9]: letters = r['letters']

In [10]: letters
```

```
Out[10]:
c("a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l",
"m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y",
"z")

In [11]: type(letters)
Out[11]: <class 'rpy2.robjects.RVector'>

In [12]: rcode = 'paste(%s, collapse="-")' %(repr(letters))

In [13]: rcode
Out[13]: 'paste(c("a", "b", "c", "d", "e", "f", "g", "h", "i", "j",
   "k", "l", \n"m", "n", "o", "p", "q", "r", "s", "t", "u", "v",
   "w", "x", "y", \n"z"), collapse="-")'

In [14]: type(rcode)
Out[14]: <type 'str'>

In [15]: r(rcode)
Out[15]: "a-b-c-d-e-f-g-h-i-j-k-l-m-n-o-p-q-r-s-t-u-v-w-x-y-z"


# 创建 rpy2 对象
In [18]: import rpy2.robjects as robjects

In [19]: robjects.StrVector(['abc', 'def'])
Out[19]: c("abc", "def")

In [20]: robjects.IntVector([1, 2, 3])
Out[20]: 1:3

In [21]: robjects.FloatVector([1.1, 2.2, 3.3])
Out[21]: c(1.1, 2.2, 3.3)

In [22]: type(robjects.FloatVector([1.1, 2.2, 3.3]))
Out[22]: <class 'rpy2.robjects.FloatVector'>

# 直接使用函数
In [35]: r.f(3)
Out[35]: 18.8495559215388
```

```
In [36]: r.sum(r.c(1,2,3))
Out[36]: 6L

# 间接使用函数
In [26]: m = robjects.r['matrix'](v, nrow = 2)

In [27]: m
Out[27]: structure(c(1.1, 2.2, 3.3, 4.4, 5.5, 6.6), .Dim = 2:3)

In [28]: type(m)
Out[28]: <class 'rpy2.robjects.RArray'>
# 函数也可以这样使用
In [29]: m = robjects.r('matrix')(v, nrow = 2)

In [30]: m
Out[30]: structure(c(1.1, 2.2, 3.3, 4.4, 5.5, 6.6), .Dim = 2:3)

# 调用函数对象
In [31]: rsum = robjects.r['sum']

In [32]: rsum(robjects.IntVector([1,2,3]))
Out[32]: 6L

In [33]: rsort = robjects.r['sort']
# 可以使用参数
In [34]: rsort(robjects.IntVector([1,2,3]), decreasing=True)
Out[34]: c(3L, 2L, 1L)

# 下面可以当做例程来使用
import rpy2.robjects as robjects
r = robjects.r

import array

x = array.array('i', range(10))
y = r.rnorm(10)

r.X11()

r.layout(r.matrix(array.array('i', [1,2,3,2]), nrow=2, ncol=2))
r.plot(r.runif(10), y, xlab="runif", ylab="foo/bar", col="red")
```

```
kwargs = {'ylab':"foo/bar", 'type':"b", 'col':"blue", 'log':"x"}
r.plot(x, y, **kwargs)

# s4 类
import rpy2.robjects as robjects
import array

r = robjects.r

r.setClass("Track",
         r.representation(x="numeric", y="numeric"))

a = r.new("Track", x=0, y=1)

a.x

# 下面代码未经测试
# 下面是一个线性回归的例子
# The R 代码:
ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)
trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
group <- gl(2, 10, 20, labels = c("Ctl","Trt"))
weight <- c(ctl, trt)

anova(lm.D9 <- lm(weight ~ group))

summary(lm.D90 <- lm(weight ~ group - 1))# omitting intercept

# 使用 rpy2.robjects
import rpy2.robjects as robjects

r = robjects.r

ctl = robjects.FloatVector([4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14])
trt = robjects.FloatVector([4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69])
group = r.gl(2, 10, 20, labels = ["Ctl","Trt"])
weight = ctl + trt

robjects.globalEnv["weight"] = weight
robjects.globalEnv["group"] = group
```

```
lm_D9 = r.lm("weight ~ group")
print(r.anova(lm_D9))

lm_D90 = r.lm("weight ~ group - 1")
print(r.summary(lm_D90))
```

8.2 把 python 数据转换为 R 可用的数据

robjects 有几个函数执行转换, 下面是常用的几个

```
BoolVector
FloatVector
IntVector
RArray
RDataFrame
RFormula
RFunction
RMatrix
RVector
StrVector
```

下面是几个例子 (RFormula 的用法参考下一节线性回归的例子)

```
# 导入
import rpy2.robjects as robjects
from  rpy2.robjects import r

In [3]: robjects.IntVector(range(10))
Out[3]: 0:9

In [4]: s="ATGCCCGTTAAAGGGTT"

In [5]: robjects.StrVector(s)
```

```
Out[5]:
c("A", "T", "G", "C", "C", "C", "G", "T", "T", "A", "A", "A",
"G", "G", "G", "T", "T")
```

8.3 执行 R 运算

定义和使用函数对象

```
# 导入
import rpy2.robjects as robjects
from  rpy2.robjects import r

fun=r('''f <- function(r) { 2 * pi * r }''')
fun(3)
r['f'](3)
r('f(3))

rnorm = robjects.r.rnorm
rnorm(100)
rnorm(100,mean=1)
```

两种方法运行 R 函数

```
# 直接使用字符串
r("t.test(c(1,2,3))")

# 使用参数
r['t.test'](r.c(1,2,3))

# 参数可以事先准备好
x=robjects.IntVector([1,2,3])
r['t.test'](x,mu=1)
```

下面是一个线性回归的例子

```
z=r('rnorm(100)')
l=len(z)
Z=z.r
N=robjects.IntVector(range(0,l))
res=r.lm("Z~N-1") # r('lm(Z~N-1)') 也可以

# 使用 RFormula
fmla = robjects.RFormula('Z~N-1')
env = fmla.getenvironment()
env['Z']=z # 不需要将 Z = z.r
env['N']=robjects.IntVector(range(0,l))
fit = robjects.r.lm(fmla)
```

8.4 将 R 结果提取到 python

subset 函数为 R 对象中提取函数. getnames() 查看 R 对象里有什么可以提取的 names.

```
# 接上面线性回归的例子
# 查看 R 对象里有什么可以提取的 names
res.getnames()
# Out[229]:
# c("coefficients", "residuals", "effects", "rank", "fitted.values",
   "assign", "qr", "df.residual", "xlevels", "call", "terms", "model"
   )

res.subset("coefficients")[0][0]
# Out[218]: 0.35289430254791571

len(res.subset("coefficients"))
# Out[232]: 1
```

Part II

基本数学计算

Chapter 9

数值计算

参考 [21] 第二章与 [46]

9.1 运算符号

- 基本符号

```
+*/-^
< <= > >= == !=
```

- 布尔运算符号

```
!,| &  - (可以 && 代替 &, -- 代替 -)
```

- 模余

```
%% 余数
%/% 商(Euclidian division )
> 12%/%3
[1] 4
> 12%%3
[1] 0
```

```
> 12.1%%3
[1] 0.1
> 12.1%/%3
[1] 4
> 12.1%/%2.2
[1] 5
# 5*-2+1=-9
> -9%%5
[1] 1
> -9%/%5
[1] -2
```

- 集合判断

```
%in%
> 17 %in% 1:100
[1] TRUE
> 17.1 %in% 1:100
[1] FALSE
```

帮助

```
?"+"
?"<"
?"<-"
?"!"
?"["
?Syntax
?kronecker
?match
library(methods)
?slot
```

还可以设计自己的符号

```
> "%w/o%" <- function(x,y) x[!x %in% y]
```

```
> (1:10) %w/o% c(3,7,12)
[1] 1 2 4 5 6 8 9 10
```

9.2 复数基本运算

- 'Re': 取实部
- 'Im': 取虚部
- 'Mod': 求模
- 'Arg': 角度
- 'Conj': 共轭

9.3 四则运算

形状不一样的数组也可以运算, 一般是把短向量循环使用(谨慎!).

```
> A <- matrix(1:6, nrow=2, byrow=T); A
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
>  B <- matrix(1:6, nrow=2); B
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
>  C <- matrix(c(1,2,2,3,3,4), nrow=2); C
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    2    3    4
> D <- 2*C+A/B; D
     [,1]     [,2] [,3]
[1,]    3 4.666667  6.6
[2,]    6 7.250000  9.0
```

9.4 集合运算

集合函数

```
union(x, y)        # 并集
intersect(x, y)    # 交集
setdiff(x, y)      # 差集
setequal(x, y)     # 集合是否相等

is.element(el, set) # el 是否 set 的元素, 同 %in% 运算符
```

例子

```
> x=c(1:5)
> y=c(3:8)
> x
[1] 1 2 3 4 5
> y
[1] 3 4 5 6 7 8

> x %in% y # 包含
[1] FALSE FALSE  TRUE  TRUE  TRUE
> is.element(x,y) # 同 %in%
[1] FALSE FALSE  TRUE  TRUE  TRUE
> union(x,y) # 并集
[1] 1 2 3 4 5 6 7 8
> intersect(x,y) # 交集
[1] 3 4 5
> setdiff(x,y) # 差集
[1] 1 2
> setequal(x,y)
[1] FALSE
```

9.5 插值

函数: spline

下面同时展示一个过拟合的例子

```
> n <- 10
> x <- seq(0,1,length=n)
> y <- 1-2*x+.3*rnorm(n)
> plot(spline(x, y, n = 10*n), col = 'red', type='l', lwd=3)
> points(y~x, pch=16, lwd=3, cex=2)
> abline(lm(y~x))
> title(main='Overfit')
```

9.6 排列组合

choose(n,k) 组合数 combn(n,k) 列出所有组合

9.7 积分

由于积分的复杂性, integrate() 有时候会出错. 一般用法为

```
integrate(f, lower, upper, ..., subdivisions=100,
            rel.tol = .Machine$double.eps^0.25, abs.tol = rel.tol,
            stop.on.error = TRUE, keep.xy = FALSE, aux = NULL)
```

'adapt' in the 'adapt' package on CRAN, for multivariate integration.

下面是几个例子.

```
> integrate(dnorm, -1.96, 1.96)
```

```
0.9500042 with absolute error < 1.0e-11

> integrate(dnorm, -Inf, Inf)
1 with absolute error < 9.4e-05

> integrand <- function(x) {1/((x+1)*sqrt(x))}
> integrate(integrand, lower = 0, upper = Inf)
3.141593 with absolute error < 2.7e-05

> integrate(integrand, lower = 0, upper = 10)
2.529038 with absolute error < 3e-04

> integrate(integrand, lower = 0, upper = 100000)
3.135268 with absolute error < 4.2e-07

> integrate(integrand, lower = 0, upper = 1000000, stop.on.error = FALSE)
failed with message 'the integral is probably divergent'

## integrate can fail if misused
    integrate(dnorm,0,2)
    integrate(dnorm,0,20)
    integrate(dnorm,0,200)
    integrate(dnorm,0,2000)
    integrate(dnorm,0,20000) ## fails on many systems
    integrate(dnorm,0,Inf)   ## works
```

9.8 求解方程式

参考数值方法的介绍 http://cran.r-project.org/web/views/Optimization.html

参考 [53] chapter 8

线性方程组求解见矩阵运算.

9.8.1 一元(非线性)方程式求根

求一个方程式的多个根使用: rootSolve 包 root.all(), 见下面

我们想求方程式

$$y = cos(x) - 2x \quad (or \quad cos(x) = 2x)$$

的根, 即使得$y = 0$的x的值.

先画图看看总是不错的.

```
curve(cos(x)-2*x,-10,10)
abline(h=0,lty=2)
```

看到确实有x值使得方程式为零.

下面使用函数 uniroot() 求根. 用法

```
uniroot(f, interval, ...,
        lower = min(interval), upper = max(interval),
        f.lower = f(lower, ...), f.upper = f(upper, ...),
        tol = .Machine$double.eps^0.25, maxiter = 1000)
```

- f: 方程式, 其第一个参数未知. 求使得方程式f值为零的第一个参数的值.
- interval: 根的搜索范围的结束点
- ...: f的其它参数的值
- tol: 需要的精确度
- f.lower, f.upper: the same as 'f(upper)' and 'f(lower)', 为了减少计算量传递的参数.

```
> u=uniroot(f = function(x) cos(x)-2*x, interval=c(-10,10)); u
$root # 根
[1] 0.4501686

$f.root # 在根处的方程式的值
[1] 3.655945e-05

$iter # 迭代次数
[1] 5

$estim.prec # 根的精确度
[1] 6.103516e-05

> r=u$root; cos(r)-2*r # 手工计算在根处的方程式的值
[1] 3.655945e-05

# 下面是函数帮助的例子
> f <- function (x,a) x - a
> str(xmin <- uniroot(f, c(0, 1), tol = 0.0001, a = 1/3))
List of 4
 $ root     : num 0.333
 $ f.root   : num -5.55e-17
 $ iter     : int 2
 $ estim.prec: num 5e-05
```

想计算$1000 = y * (3 + x) * (1 + y)^4$, 未知数是y, x从$1 - 100$变动. 我们绘出根与x的关系

```
eq<-function(y,x){
   return (1000-y*(3+x)*(1+y)^4)
}

r=rep(0,100)
x=1:100
for (i in x){
   r[i]<-uniroot(eq, c(-100,100),x=i)$root
}
plot(r~x)
```

9.8.2 多个根

求一个方程式的多个根使用: rootSolve 包 uniroot.all

```
>␣uniroot.all(f␣=␣function(x)␣x^2-1,␣interval=c(-10,10))
[1]␣-1␣␣1
␣␣␣␣
```

9.8.3 多元(非线性)方程组

非线性微分方程求根: rootSolve 包 multiroot() 求解n个(非线性)方程组的n个根.

下面是 multiroot()帮助的例子. 更具体见帮助文件.

```
> model <- function(x) {
   c(F1=x[1]^2+ x[2]^2 -1,F2=x[1]^2- x[2]^2 +0.5)}
> (ss<-multiroot(f=model,start=c(1,1)))
$root
[1] 0.5000000 0.8660254

$f.root
        F1         F2
2.323138e-08 2.323308e-08

$iter
[1] 5

$estim.precis
[1] 2.323223e-08
# 代入原方程组
> model(ss$root)
        F1         F2
2.323138e-08 2.323308e-08

# 3个方程式2个根
model <- function(x) {
```

```
   c(F1= x[1] + x[2] + x[3]^2 - 12,
      F2= x[1]^2 - x[2] + x[3] - 2,
      F3= 2 * x[1] - x[2]^2 + x[3] - 1 )}
# first solution
(ss<-multiroot(model,c(1,1,1),useFortran=FALSE))
(ss<-multiroot(f=model,start=c(1,1,1)))
# second solution; use different start values
(ss<-multiroot(model,c(0,0,0)))
model(ss$root)

# 还可以求解矩阵
f2<-function(x)
{
 X<-matrix(nr=5,x)
 X %*% X %*% X -matrix(nr=5,data=1:25,byrow=TRUE)
}
x<-multiroot(f2, start= 1:25 )$root
X<-matrix(nr=5,x)
X%*%X%*%X
```

9.9 优化(求极值)

最优方法是使得目标函数极大或极小. 对于极大的问题, 可以对目标函数$-f(x)$求极小. 最优化问题有些书籍称为极值问题或数学规划问题.

参考文献 [17] 第十三章对最优化方法有一个很好的描述.

参考数值方法的介绍 http://cran.r-project.org/web/views/Optimization.html

9.9.1 optimize()函数

函数 optimize() 求得一个函数在指定区间的极值. 使用 golden section search(黄金分割搜索) 和 successive parabolic interpolation(连续抛物线插值). 收敛速度不比使用 Fibonacci search 慢多少. 更

多详细解释见帮助, 另外参考 ”极大似然法”21.2.

用法为

```
optimize(f = , interval = ,  ..., lower = min(interval),
           upper = max(interval), maximum = FALSE,
           tol = .Machine$double.eps^0.25)
```

返回

- minimum(maximum): 函数取得最大(最小)值时自变量x的值
- objective: 函数的极大(极小)值

下面是帮助中的例子

```
f <- function (x,a) (x-a)^2
xmin <- optimize(f, c(0, 1), tol = 0.0001, a = 1/3)
> xmin
$minimum
[1] 0.3333333

$objective
[1] 0

# 不赋值的话, 可以看到函数自变量取值的情况
optimize(function(x) x^2*(print(x)-1), lower=0, upper=10)

# 函数中有部分常数值, 计算区间取的不够大的话, 就会犯错
误
f  <- function(x) ifelse(x > -1, ifelse(x < 4, exp(-1/abs(x - 1)), 10), 10)
fp <- function(x) { print(x); f(x) }

plot(f, -2,5, ylim = 0:1, col = 2)

# 虽然函数极小值在(1,0)附近, 但是区间不够的话收敛到错误
的地方
```

```
optimize(fp, c(-4, 20))# doesn't see the minimum
# 这个就正确了
optimize(fp, c(-7, 20))# ok
```

9.9.2 nlm()函数

nlm() 函数使用 Newton-type 算法求最小值. 参考 36 章非线性回归与非线性最小平方. 《R导论》[25](page 73)中统计模型部分中有一个广义线性模型对广义线性模型有很好的描述, 请参考之。

求无约束求化问题(Rosenbrock函数, 或橡胶函数)

$$min \quad f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

的极小点.

```
# 写出目标函数
obj<-function(x){
  f<-c(10*(x[2]-x[1]^2), 1-x[1])
  sum(f^2)
}

# 初始值设置
x0=c(-1.2,1)
# 求解
nlm(obj,x0)
> nlm(obj,x0)
$minimum # 极小值
[1] 3.973766e-12

$estimate # 极小值对应的x点
[1] 0.999998 0.999996

$gradient # 极小值处的梯度
[1] -6.539277e-07  3.335997e-07

$code # 成功与否
```

```
[1] 1

$iterations # 迭代次数
[1] 23
```

9.9.3 其它函数

BB 包求解非线性系统方程组, 并根据约束优化.

R的非线性优化程序是optim(), nlm() 和nlminb().

函数 optim() 是一个广泛意义的优化函数.

函数 nls(): 非线性模型参数的最小平方估计 (Determine the nonlinear (weighted) least-squares estimates of the parameters of a nonlinear model). 使用请参考 36 章非线性回归与非线性最小平方

9.10 拉格朗日乘数(Lagrange Multipliers)

参考 http://zh.wikipedia.org/wiki/拉格朗日乘数

在数学最优化问题中, 拉格朗日乘数（以约瑟夫·路易斯·拉格朗日命名）是一种寻找变量受一个或多个限制的多元方程的极值的方法。这种方法将一个有n 变量与 k 约束的问题转换为一个更易解的 $n+k$ 个变量的方程组，其变量不受任何约束。这种方法引入了一种新的标量未知数，即拉格朗日乘数：约束方程的斜率（gradient）的线性组合里每个向量的系数。

此方法的证明牵涉到偏微分，全微分或链法，从而找到能让设出的隐函数的微分为零的未知数的值。

9.10.1 介绍

先看一个二维的例子：假设有方程：$f(x,y)$，要求其最大值，且

$$g\left(x,y\right)=c$$

c 为常数。对不同d_n的值，不难想象出

$$f\left(x,y\right)=d_n$$

的等高线。而方程g的等高线正好是$g(x,y)=c$。想象我们沿着$g=c$的等高线走；因为大部分情况下f和g的等高线不会重合，但在有解的情况下，这两条线会相交。想象此时我们移动$g=c$上的点，因为f是连续的方程，我们因此能走到$f\left(x,y\right)=d_n$更高或更低的等高线上，也就是说d_n可以变大或变小。只有当$g=c$和$f\left(x,y\right)=d_n$相切，也就是说，此时，我们正同时沿着$g=c$和$f\left(x,y\right)=d_n$走。这种情况下，会出现极值或鞍点。

气象图中就很常出现这样的例子，当温度和气压两列等高线同时出现的时候，切点就意味着约束极值的存在。

用向量的形式来表达的话，我们说相切的性质在此意味着f和g的斜率在某点上平行。此时引入一个未知标量λ，并求解：

$$\nabla\Big[f\left(x,y\right)+\lambda\left(g\left(x,y\right)-c\right)\Big]=0$$

且 ‘0. 一旦求出λ的值，将其套入下式，易求在无约束极值和极值所对应的点。

$$F\left(x,y\right)=f\left(x,y\right)+\lambda\left(g\left(x,y\right)-c\right)$$

新方程$F(x,y)$在达到极值时与$f(x,y)$相等，因为$F(x,y)$达到极值时$g(x,y)$`c总等于零。

9.10.2 拉格朗日乘数的运用方法

如f定义为在R^n上的方程，约束为$g_k(x) = c$（或将约束左移得到$g_k(x)`c = 0\Psi$。定义拉格朗日为

$$\Lambda(\mathbf{x}, \boldsymbol{\lambda}) = f + \sum_k \lambda_k g_k.$$

注意极值的条件和约束现在就都被记录到一个式子里了：

$$\nabla_{\mathbf{x}}\Lambda = 0 \Leftrightarrow \nabla_{\mathbf{x}} f = -\sum_k \lambda_k \nabla_{\mathbf{x}} g_k$$

和

$$\nabla_{\lambda}\Lambda = 0 \Leftrightarrow g_k = c.$$

拉格朗日乘数常被用作表达最大增长值。原因是从式子：

$$\frac{\partial \Lambda}{\partial g_k} = \lambda_k.$$

中我们可以看出$_k$是当方程在被约束条件下，能够达到的最大增长率。拉格朗日力学就使用到这个原理。

拉格朗日乘数法在Karush-Kuhn-Tucker最化件被推广。

9.10.3 例子

很简单的例子

求此方程的最大值：

$$f(x, y) = x^2 y$$

同时未知数满足

$$x^2 + y^2 = 1$$

因为只有一个未知数的限制条件，我们只需要用一个乘数λ.

$$g(x,y) = x2 + y2`1$$
$$(x,y,) = f(x,y) + g(x,y) = x2y + (x2 + y2`1)$$

将所有Φ方程的偏微分设为零，得到一个方程组，最大值是以下方程组的解中的一个：

$$2xy + 2x = 0$$
$$x^2 + 2y = 0$$
$$x^2 + y^{2`}1 = 0$$

另一个例子

求此离散分布的最大熵：

$$f(p_1, p_2, \ldots, p_n) = -\sum_{k=1}^{n} p_k \log_2 p_k.$$

所有概率的总和是1，因此我们得到的约束是$g(p) = 1$ 即

$$g(p_1, p_2, \ldots, p_n) = \sum_{k=1}^{n} p_k = 1.$$

可以使用拉格朗日乘数找到最高熵（概率的函数）。对于所有的k 从1 到 n, 要求

$$\frac{\partial}{\partial p_k}(f + \lambda(g-1)) = 0,$$

由此得到

$$\frac{\partial}{\partial p_k}\left(-\sum_{k=1}^{n} p_k \log_2 p_k + \lambda(\sum_{k=1}^{n} p_k - 1)\right) = 0.$$

计算出这n个等式的微分，我们得到：

$$-\left(\frac{1}{\ln 2}+\log_2 p_k\right)+\lambda=0.$$

这说明p_i都相等(因为它们都只是 λ 的函数). 解出约束 $\jmath kp_k=1$, 得到

$$p_k=\frac{1}{n}.$$

因此，使用均匀分布可得到最大熵的值。

9.10.4 经济学

约束最优化在经济学占有很重要的地位。例如一个消费者的选择问题可以被视为一个求效用方程在预算约束下的最大值问题。拉格朗日乘数在经济学中被解释为影子价格，设定在某种约束下，在这里即收入的边际效用。

拉格朗日乘数就是效用函数在最优解出对收入的偏导数，也就是在最优解处增加一个单位收入带来的效用增加，或者说在最优解处有效用衡量收入的价值，称之为收入的边际效用。

在企业生产问题中，拉格朗日乘数用来衡量要素投入变动所带来的收入变动，$du/dm=$,u表示效用函数或生产函数，m表示收入或要素投入。

在具体数学推导中还可以运用包络定理的内容。

Chapter 10

空间几何

10.1 坐标系旋转

$$X' = x * cos(n) + y * sin(n)$$
$$Y' = -x * sin(n) + y * cos(n)$$

n是旋转的角度。将原坐标系逆时针旋转角度n后, 形成新的坐标系. X'和Y'为新坐标系下点的坐标. 而x和y为该点在原来坐标系下的坐标.

等价于坐标点顺时针旋转后在原坐标系的坐标.

```
# 计算坐标系逆时针旋转后的新坐标, 注意不包括平移.
# 等价于坐标点顺时针旋转后在原坐标系的坐标.
# 假设以(1,2)为中心的旋转, 那么旋转时需先x-1, y-2, 新坐标需要x'+1, y'+2
new.pos<-function(x,y,angle){
   n=angle*3.141592653589793/180
   x1=x*cos(n)+y*sin(n)
   y1=-x*sin(n)+y*cos(n)
   res=c(x1,y1)
   res
}
```

10.2 两点的直线方程

方程形式为

```
ax+by+c=0
```

三维的点为平面, 形式为

```
ax+by+cz+d=0
```

```
line.coef<-function(x1,y1,x2,y2){
   dx=x2-x1
   dy=y2-y1
   if(dx==0){
      a=1 # 固定 a=1
      b=0
      c=-x1
      return(c(a,b,c))
   }
   if(dy==0){
      a=0
      b=1 # 固定 b=1
      c=-y1
      return(c(a,b,c))
   }
   a=1 # 固定 a=1
   b=-dx/dy
   c=-(x1+b*y1)
   return(c(a,b,c))
}
```

10.3 距离

参考 http://zh.wikipedia.org/wiki/距离

另外参考 距离系数15

10.3.1 两点间的距离

即两个点之间的线段的长度。二维距离：

$$d = \sqrt{(\Delta x)^2 + (\Delta y)^2}$$

三维距离：

$$d = \sqrt{(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2}$$

R的例子

```
> x <- c(0, 0, 1, 1, 1, 1)
> y <- c(1, 0, 1, 1, 0, 1)
> dist(rbind(x,y)) # 每行为一个点的坐标, 默认为欧氏距离
         x
y 1.414214

# 手工计算
> sqrt(sum((x-y)^2))
[1] 1.414214
```

10.3.2 点到直线的距离

点和直线的距离是点到直线的垂直线段的长度.

若在平面坐标几何上的直线定义为 $ax+by+c=0$，点的座标为 $(x0,y0)$，则它们之间的距离为：

$$d=\left|\frac{ax_0+by_0+c}{\sqrt{a^2+b^2}}\right|$$

10.3.3 异面直线间的距离

设两直线的方程分别为：

$$\frac{x-x_1}{L_1}=\frac{y-y_1}{M_1}=\frac{z-z_1}{N_1}$$

$$\frac{x-x_2}{L_2}=\frac{y-y_2}{M_2}=\frac{z-z_2}{N_2}$$

则，该两直线间的距离

$$d=\frac{\begin{vmatrix}x_2-x_1 & y_2-y_1 & z_2-z_1\\ L_1 & M_1 & N_1\\ L_2 & M_2 & N_2\end{vmatrix}}{\sqrt{\begin{vmatrix}M_1 & N_1\\ M_2 & N_2\end{vmatrix}^2+\begin{vmatrix}N_1 & L_1\\ N_2 & L_2\end{vmatrix}^2+\begin{vmatrix}L_1 & M_1\\ L_2 & M_2\end{vmatrix}^2}}$$

10.3.4 点到平面的距离

若点坐标为$(x0,y0,z0)$,平面为$Ax+By+Cz+D=0$,则点到平面的距离为:

$$d=\left|\frac{Ax_0+By_0+Cz_0+D}{\sqrt{A^2+B^2+C^2}}\right|$$

推广到超平面, 若R^n空间中点坐标为$x=(x_1,\cdots,x_n)\in R^n$, R^n空间中的超平面可以使用一个系数向量w和平移(偏置)b表示

为$< w, b >$, 即$f(x) = wx + b = 0$, $f(x) = 0$就是R^n空间中的一个超平面. 某点与此超平面的距离为

$$d = \left| \frac{f(x)}{||w||} f(x) = \frac{wx + b}{\sqrt{w_1^2 + \cdots + w_n^2}} \right|$$

10.3.5 两平行直线

若直线分为 $ax + by + c1 = 0$, 和 $ax + by + c2 = 0$, 则它们之间的距离为:

$$d = \left| \frac{c_1 - c_2}{\sqrt{a^2 + b^2}} \right|$$

10.3.6 两平行平面间的距离

若两平为 $Ax + By + Cz + D1 = 0$, 和 $Ax + By + Cz + D2 = 0$, 则他们之间的距离为:

$$d = \left| \frac{D_1 - D_2}{\sqrt{A^2 + B^2 + C^2}} \right|$$

10.3.7 范数

设在$\mathbb{R}^m$空间有两点， $p = (p1, p2, ..., pm)$fi$q = (q1, q2, ..., qm)$， 不同

的范数都是一种距离：

1-阶范数 $= \sum |x_i - y_i|$

2-阶范数 $= \left(\sum |p_i - q_i|^2 \right)^{\frac{1}{2}}$

n-阶范数 $= \left(\sum |p_i - q_i|^n \right)^{\frac{1}{n}}$

无穷大阶范数 $= t$阶范数的极限，即 n 趋向无穷大 $\lim_{n \to \infty} \left(\sum |p_i - q_i|^n \right)^{\frac{1}{n}} = max |$

10.4 三角形

参考 http://zh.wikipedia.org/wiki/三角形

10.4.1 基本概念

中线：三角形一边中点与这边所对定点的连线段。

高线：从三角形一个顶点向它的对边所作的垂线段。

角平分线：平分三角形一角、一个端点在这一角的对边上的线段。

10.4.2 定理

三角不等式

三角形两边之和大于第三边，两边之差的绝对值小于第三边。如果两者相等，则是退化三角形。

三角形任意一个外角大于不相邻的一个内角。

勾股定理

（又称毕氏定理或毕达哥拉斯定理）及其勾股逆定理：

设直角三角形ABC的三顶点A、B、C所对的三边分别为a、b、c，则 $a^2+b^2=c^2$ 当角$C=90$。

正弦定理

（R为三角形外接圆半径）：

$$\frac{a}{\sin(\alpha)}=\frac{b}{\sin(\beta)}=\frac{c}{\sin(\gamma)}=2R$$

余弦定理

$$a^2=b^2+c^2-2bc\cdot\cos(\alpha)$$
$$b^2=a^2+c^2-2ac\cdot\cos(\beta)$$
$$c^2=a^2+b^2-2ab\cdot\cos(\gamma)$$

10.4.3 角度

三角形两只内角之和，等于剩下的一只的外角。

在欧几里德平面内，三角形的内角和等于180°。

10.4.4 分类

锐角、钝角三角形

钝角三角形是其中一角为钝角（大于90°）的三角形，其

余两角均小于90°。

锐角三角形的所有内角均为锐角（小于90°）。

直角三角形

有一个角是直角（90°）的三角形为直角三角形。成直角的两条边称为直角边（cathetus），直角所对的边是斜边（hypotenuse）；或最长的边称为弦，，底部的一边称作勾（又作句），另一边称为股。

可以透过不同角度的直角三角形各边的比求得锐角三角函数。

等边三角形

等边三角形（又称正三角形），为三边相等的三角形。其三个内角相等，均为60°。它是锐角三角形的一种。设其边长是a，则其面积公式为

$$\frac{\sqrt{3}}{4}a^2$$

等边三角形是正四面体、正八面体和正二十面体这三个正多面体面的形状。六个等边三角形可以拼成一个正六边形。

等腰三角形

等腰三角形是三条边中有两条边相等（或是其中两只内角相等）的三角形。等腰三角形中的两条相等的边被称为腰，而另一条边被称为底边，两条腰交叉组成的那个点被称为顶点，它们组成的角被称为顶角。等腰三角形的重心、中心和垂心都位于顶点向底边的垂线上。

等腰三角形的底的垂直平分线，刚好又是对应角的角平分线。

等边三角形是等腰三角形的一个特殊形式。

等腰直角三角形只有一种形状，其中两个角为45度。

退化三角形

退化三角形的面积为零。这种三角形通常只有几类：如果一个三角形内的三只角的角度分别为 $(180, 0, 0)$ 或 $(90, 90, 0)$，则它是一个退化三角形。

另外，如果一个三角形的其中一条边等于其余两条边之和，或者其中一条边为零，都可以称为退化三角形。

一般来说，这些三角形都不被认定为三角形，因此有人认为退化三角形并非三角形的一种；这是由于它介乎于三角不等式之间，在一些资料中已否定了其中一条边等于其余两条边的情况。

10.4.5 特性

三角形具有稳定性：当三角形的三边确定后，它的形状、大小就不会改变。

10.4.6 面积

已知两边及其夹角

设a、b为所知的两边，C为该夹角，三角形面积

$$S = \frac{1}{2}ab\sin C$$

已知底和高

$$S = \frac{1}{2}bh$$

即底×高÷2。因为两个相同的三角形叠合可成平行四边形。

已知三边长

希罗公式（又称海伦公式）：设p等于三角形三边和的一半：

$$p = \frac{a+b+c}{2}$$

则

$$S = \sqrt{p\left(p-a\right)\left(p-b\right)\left(p-c\right)}$$

化简后就是：

$$S = \frac{1}{4}\sqrt{\left(a+b+c\right)\left(a+b-c\right)\left(a+c-b\right)\left(b+c-a\right)}$$

秦九韶亦求过类似的公式，称为三斜求积法：

$$\sqrt{\frac{1}{4}(c^2a^2 - (\frac{c^2+a^2-b^2}{2})^2)}$$

基于希罗公式在三角形拥有非常小的角度时并不数值稳定，有一个变化的计法。设$a >= b >= c$，三角形面积为

$$\frac{1}{4}\sqrt{(a+(b+c))(c-(a-b))(c+(a-b))(a+(b-c))}$$

在坐标系中已知三顶点坐标

由$(x1,y1),(x2,y2),(x3,y3)$三个顶点构成的三角形，其面积为：

$$\frac{1}{2}\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$$

任三角形外心和内心半径算面积法

假设已知三角形面积为x，三边边长分别为a.b.c，s为三角形周长$(a+b+c)$

内心半径(r)：

$$x = \frac{1}{2}sr$$

外心半径(R)：

$$x = \frac{abc}{4R}$$

10.4.7 其他三角形有关的定理

拿破仑三角形 费马点 欧拉线 梅涅劳斯定理

10.4.8 三角形的五心

内心

三个内角的角平分线的交点 三角形内切圆的圆心

外心

三条边的垂直平分线的交点 三角形外接圆的圆心

垂心

三条高的交点

重心

三条中线的交点 被交点划分的线段比例为1:2 (靠近角的一段较长)

旁心

外角的角平分线的交点有三个，为三角形某一边上的旁切圆的圆心

10.5 三角函数

参考 http://zh.wikipedia.org/wiki/三角函

10.6 凸包

创建者：阳光可可豆333 http://baike.baidu.com/view/707209.htm

10.6.1 概念

1.1 点集Q的凸包(convex hull)是指一个最小凸多边形，满足Q中的点或者在多边形边上或者在其内。下图中由红色线段表示的多边形就是点集Q=p0,p1,...p12的凸包。

1.2 一组平面上的点，求一个包含所有点的最小的凸多边形，这就是凸包问题了。这可以形象地想成这样：在地上放置一些不可移动的木桩，用一根绳子把他们尽量紧地圈起来，这就是凸包了。

10.6.2 平面凸包的求法

2.1 凸包最常用的凸包算法是Graham扫描法和Jarvis步进法。

对于一个有三个或以上点的点集Q，过程如下：

计算点集最右边的点为凸包的顶点的起点，如上图的P3点。

```
Do
For i = 0 To 总顶点数
 计算有向向量P3->Pi
If 其余顶点全部在有向向量P3->Pi的左侧或右侧，则Pi点为凸包的下一顶点
Pi点加入凸包列表
GoTo 1
End If
Next
Exit Do
1:
Loop
```

此过程执行后，点按极角自动顺时针或逆时针排序，只需要按任意两点的次序就可以了。而左侧或右侧的判断可以用前述的矢量点积性质实现。

2.2 求凸包有很多方法，不过最适合OI的估计还是Graham's Scan这个方法了。它的大致方法是这样的：首先，找到所有点中最左边的（y坐标最小的），如果y坐标相同，找x坐标最小的；以这个点为基准求所有点的极角（atan2(y-y0,x-x0)），并按照极角对这些点排序，前述基准点在最前面，设这些点为P[0]..P[n-1]；建立一个栈，初始时P[0]、P[1]、P[2]进栈，对于P[3..n-1]的每个点，若栈顶的两个点与它不构成"向左转"的关系，则将栈顶的点出栈，直至没有点需要出栈以后将当前点进栈；所有点处理完之后栈中保存的点就是凸包了。

如何判断A、B、C构成的关系不是向左转呢？如果b-a与c-a的叉乘小于0就不是。a与b的叉乘就是a.x*b.y-a.y*b.x。

上面的这个Graham的实现比我原来按照USACO里的课文写得简单多了，主要是它通过简单的预处理保证了P[0]、P[1]以及P[n-1]肯定是凸包里的点，这样就可以避免在凸包"绕回来"的时候繁杂的处理

10.6.3 例子: geometry包

下面定义x为4个点, 实际上是边长为2的正方形的四个顶点. 2维时, area为凸包的周长(2维). vol为面积. 3维时, area为凸包的表面积. vol为体积. 当维数增加时依次类推.

```
> x=matrix(c(0,2,2,0,0,0,2,2),nc=2)
> x
     [,1] [,2]
[1,]    0    0
[2,]    2    0
[3,]    2    2
[4,]    0    2
> library(geometry)
> convhulln(x,option="FA")
$hull
     [,1] [,2]
[1,]    2    1
[2,]    3    2
```

```
[3,]   4   1
[4,]   4   3

$area
[1] 8

$vol
[1] 4
```

Chapter 11

向量代数

11.1 向量概念

11.1.1 数量

只有大小没有方向的量, 例如质量,体积,面积,温度,时间等, 叫做数量.

11.1.2 向量

不仅有大小, 还有方向, 如速度, 力, 位移等, 叫做向量.

11.1.3 自由向量

许多问题中, 只研究向量的大小和方向, 不考虑始点位置, 称为自由向量.

11.1.4 向量相等

自由向量中, 所谓两个向量相等, 指两个向量大小相等, 互相平行且指向相同, 即平移后能够完全重合.

11.1.5 向量的模

向量的大小叫做向量的模.

11.1.6 单位向量

模为1的向量称为单位向量.

11.1.7 零向量

模等于0的向量称为零向量. 零向量的方向可以看作任意的.

11.1.8 向径

直角坐标系中, 以坐标原点o为始点, 向一个点M引向量$O\vec{M}$, 这个向量称为点M对O的向径, 常用粗体$\boldsymbol{r}$表示.

11.2 向量加法

设$a = O\vec{A}, b = vecOB$, 以A,B为边做平行四边形, O的对角为C, $c = O\vec{C}$, 那么

$$a + b = c = O\vec{C}$$

叫做向量加法的三角形法则.

三角形法则可以推广到任意有限个向量的和.

11.3 向量在轴上的投影

11.3.1 两个向量的夹角

两个非零向量a,b, 交于一点S, 如果不相交, 可以平移其中一个, 使其相交. 把其中一个向量绕S在两个向量决定的平面上旋转, 使得其正方向与另外一个向量的正方向重合, 这样得到的一个旋转角度$\varphi(0 < \varphi < \pi)$称为向量a,b之间的夹角. 若a,b平行, 规定其夹角为0.

11.3.2 向量的投影

向量$vecAB$在轴u上的投影等于向量的模乘以轴与向量间的夹角φ的余弦

$$Prj_u\vec{AB} = |\vec{AB}|\cos\varphi$$

定理: 有限个向量的和在轴上的投影等于各个向量在该轴上的投影的和.

11.3.3 模的坐标表示

设点M的坐标为x,y,z. 则向量$a = \vec{OM}$的模

$$|a| = \sqrt{x^2 + y^2 + z^2}$$

11.3.4 方向余弦

接上面, 又设a与三个坐标轴的夹角分别为α, β, γ, 由投影定理得

$$x = |a|\cos\alpha$$
$$y = |a|\cos\beta$$
$$z = |a|\cos\gamma$$

从而

$$\cos\alpha = \frac{x}{|a|}$$

$$\cos\beta = \frac{y}{|a|}$$

$$\cos\gamma = \frac{z}{|a|}$$

$\cos\alpha, \cos\beta, \cos\gamma$叫做a的方向余弦. α, β, γ叫做向量a的方向角.

把上面三个等式平方后相加得

$$\cos^2\alpha + \cos^2\beta + \cos^2\gamma = \frac{x^2+y^2+z^2}{|a|} = 1$$

即, 任何向量的方向余弦的平方和为1.

11.4 两个向量的数量积(点积,内积)

11.4.1 定义

两个向量a,b的模与它们夹角θ的余弦的积, 叫做a,b的数量积, 记作

$$a \cdot b = |a| \cdot |b| \cos\theta$$

物理上, 可以把a看作力, b为物体位移, θ为力与位移的夹角, 其数量积则是力做的功.

数量积也称为"点积","内积". 数量积是一个数值, 没有方向.(向量积是有方向的)

由投影定理得知, $|b|\cos\theta$为b在方向a上的投影, 记作$Prj_a b$, 有

$$a \cdot b = |a| Prj_a b$$

$$a \cdot b = |b| Prj_b a$$

即, 两个向量的数量积等于其中一个向量的模和另一个向量在此向量上投影的积.

11.4.2 推论

$a \cdot a = |a|^2$

两个非零向量互相垂直的充要条件是 $a \cdot b = 0$

11.4.3 数量积的坐标表示

设向量a坐标为a_x, a_y, a_z,向量b的坐标为b_x, b_y, b_z,三个坐标轴单位向量为i, j, k. 即

$$a = a_x i a_y j a_z k$$
$$b = b_x i b_y j b_z k$$

根据数量积的运算规律

$$a \cdot b = (a_x i a_y j a_z k) \cdot (b_x i b_y j b_z k) = \cdots$$

由于$i \cdot j = j \cdot k = i \cdot k = 0, i \cdot i = j \cdot j = k \cdot k = 1$, 代入上面有

$$a \cdot b = a_x b_x + a_y b_y a_z b_z$$

即数量积的坐标表示.

当a,b非零, 有

$$\cos\theta = \frac{a \cdot b}{|a||b|}$$

坐标表示为

$$\cos\theta = \frac{a_x b_x + a_y b_y a_z b_z}{\sqrt{a_x^2 + a_y^2 + a_z^2}\sqrt{b_x^2 + b_y^2 b_z^2}}$$

11.4.4 向量垂直的充要条件

a,b垂直的充要条件为

$$a_x b_x + a_y b_y a_z b_z = 0$$

11.4.5 计算函数

很容易编写夹角计算函数

```
theta<-function(x,y){
   r<-acos(sum(x*y)/sqrt(sum(x^2)*sum(y^2)))
   r
}
> x=c(5,2,5)
> y=c(2,-1,2)
> theta(x,y) # 弧度
[1] 0.6154797
```

11.5 两个向量的向量积(矢量积,叉积,外积)

11.5.1 定义

设向量c由两个向量a,b按照下面的规则给出,

1. c的模$|c| = |a| \cdot |b| \cdot \sin\theta$, 其中$\theta$是a,b的夹角
2. c垂直于a,b确定的平面. 指向使得a,b,c符合右手法则.

那么c叫做a,b的向量积, 记作ab.

c的模相当于a,b构成的平行四边形的面积.

物理上, 设O为杠杆的支点, $\vec{OP}$为杠杆, 力F作用于P点, 与$\vec{OP}$的夹角为θ, 力F对O的力矩为向量M, 则

$$|M| = |\vec{OP}||F|\sin\theta$$

M的方向符合右手法则.

11.5.2 推论

$aa = 0$

对于非零向量a, b平行的充要条件为

$$ab = 0$$

运算规律

1. $ba = -ab$
2. $\lambda ab = a(\lambda b)$
3. $(a + b)c = ac + bc$

11.5.3 坐标形式

设

$$a = a_x i a_y j a_z k$$
$$b = b_x i b_y j b_z k$$

根据

$$ii = jj = kk = 0$$
$$ij = k, jk = i, ki = j$$
$$ji = -k, kj = -i, ik = -j$$

经过运算得到

$$\begin{aligned} ab &= (a_y b_z - a_z b_y)i + (a_z b_x - a_x b_z)j + (a_x b_y - a_y b_x)k \\ &= \begin{vmatrix} i & j & k \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{vmatrix} \end{aligned}$$

11.5.4 向量平行的充要条件

两个向量互相平行相当于

$$a_y b_z - a_z b_y = 0,\ a_z b_x - a_x b_z = 0\ a_x b_y - a_y b_x = 0$$

$$or\quad \frac{a_x}{b_x} = \frac{a_y}{b_y} = \frac{a_z}{b_z}$$

11.5.5 为什么力矩垂直于力和力臂确定的平面

这要从角速度方向的定义说起, 角速度是矢量, 但它的方向和力, 速度, 电场等物理量方向的定义不同. 因为物体转动时, 每个质点的线速度方向可能不同. 而如果简单的说顺时针和逆时针, 这也不行, 因为这是相对的. 正面看是顺时针, 背面看就成了逆时针. 所以规定角速度方向是垂直于转动平面, 并遵循右手定则. 如果有一个圆盘在纸上顺时针转动, 则它的角速度方向是垂直于纸面向里的.

现在说力矩的方向, 因为力矩的效应是使物体产生转动或具有转动趋势. 所以它的方向也该是垂直于纸面, 并遵循右手定则.

11.5.6 计算函数

很容易编写向量积计算函数

```
cross.prod<-function(x,y){
   r<-c(x[2]*y[3]-x[3]*y[2],x[3]*y[1]-x[1]*y[3],x[1]*y[2]-x[2]*y[1])
   r
}
> x=c(-3,4,-6)
> y=c(-2,3,-1)
> cross.prod(x,y)
[1] 14  9 -1
```

11.6 例子: 求两个向量的夹角

点 O,A,B, 两 个 向 量OA=v1=(x1,y1,z1), OB=v2=(x2,y2,z2), 那 么 OA,OB 之 间 的 夹 角 的 余 弦 为

```
cos(theta)=sum(v1*v2)/sqrt(sum(v1^2)*sum(v2^2))
theta=acos( cos(theta) )
```

例 如

```
> v1=c(1,2,3)
> v2=c(2,7,4)
> sum(v1*v2) # 点 积
> cos_value=sum(v1*v2)/sqrt(sum(v1^2)*sum(v2^2))
> cos_value
[1] 0.9008852
> acos(cos_value) # 角 度
[1] 0.4489917
```

Chapter 12

矩阵运算

12.1 构造Hilbert矩阵

Matrix包有函数Hilbert()可以产生n阶对称Hilbert矩阵. Hilbert矩阵的阶数n较大的时候是病态的, 故经常用来测试数值方法程序.

```
# 手工计算
n<-4; x<-array(0, dim=c(n,n))
for (i in 1:n){
   for (j in 1:n){
      x[i,j]<-1/(i+j-1)}}
> x
        [,1]      [,2]      [,3]      [,4]
[1,] 1.0000000 0.5000000 0.3333333 0.2500000
[2,] 0.5000000 0.3333333 0.2500000 0.2000000
[3,] 0.3333333 0.2500000 0.2000000 0.1666667
[4,] 0.2500000 0.2000000 0.1666667 0.1428571

# 使用函数 Hilbert()
library(Matrix)
> Hilbert(3)
3 x 3 Matrix of class "dpoMatrix"
        [,1]      [,2]      [,3]
```

```
[1,] 1.0000000 0.5000000 0.3333333
[2,] 0.5000000 0.3333333 0.2500000
[3,] 0.3333333 0.2500000 0.2000000
```

12.2 范数

向量w的p范数为

$$||w||_p = \sqrt[p]{w_1^p + \cdots + w_n^p}$$

$p = 2$时为传统的长度度量. 当不指明p时, 经常意味着不关心是几范数

```
> x=1:10
# x 的 3 范数, 即3次方之和, 然后开3次方
> sum(x^3)^(1/3)
[1] 14.46245
```

12.3 矩阵转置

使用函数 t()

```
> A <- matrix(1:6, nrow=2, byrow=T); A
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
> t(A)
     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
```

12.4 上下三角矩阵

base 包的函数如下

```
> x=matrix(1:20,c(4,5))
> x
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20
> upper.tri(x, diag = FALSE)
      [,1]  [,2]  [,3]  [,4] [,5]
[1,] FALSE  TRUE  TRUE  TRUE TRUE
[2,] FALSE FALSE  TRUE  TRUE TRUE
[3,] FALSE FALSE FALSE  TRUE TRUE
[4,] FALSE FALSE FALSE FALSE TRUE
> x[upper.tri(x)]
 [1]  5  9 10 13 14 15 17 18 19 20

# 下三角矩阵
> lower.tri(x)
```

spam 包的函数功能要多一些

```
> y=matrix(1:20,c(4,5))
> y1=as.spam(y)
> upper.tri(y1,diag=T)
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    1    1    1    1
[2,]    0    1    1    1    1
[3,]    0    0    1    1    1
[4,]    0    0    0    1    1
Class 'spam'
> y1
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
```

```
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20
Class 'spam'
> lower.tri(y1,diag=T)

# 获取
> y1[lower.tri(y1,diag=T)]
     [,1] [,2] [,3] [,4]
[1,]    1    0    0    0
[2,]    2    6    0    0
[3,]    3    7   11    0
[4,]    4    8   12   16
Class 'spam'
```

12.5 行列式的值

```
> det(A)
错误于determinant.matrix(x, logarithm = TRUE, ...) :
  'x'必需是正方形矩阵
> det(A[1:2,1:2])
[1] -3
```

12.6 内积与外积

内积(点积)可以使用

```
> x <- 1:5; y <- 2*1:5
> x
[1] 1 2 3 4 5
> y
[1]  2  4  6  8 10
```

```
# 向量内积
> x %*% y
     [,1]
[1,]  110

# %*% 符号是通常意义下的矩阵乘
# crossprod() 是内积函数, 执行 t(x) %*% y
> crossprod(x,y)
     [,1]
[1,]  110

# 矩阵内积
> A <- matrix(1:6, nrow=2, byrow=T); A
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
>  B <- matrix(1:6, nrow=2); B
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6

> A %*% B
错误于A %*% B : 非整合变元

> t(A) %*% B
     [,1] [,2] [,3]
[1,]    9   19   29
[2,]   12   26   40
[3,]   15   33   51

> crossprod(A,B)
     [,1] [,2] [,3]
[1,]    9   19   29
[2,]   12   26   40
[3,]   15   33   51

# tcrossprod(x,y) 是外积, 执行 x %*% t(y), 或 x %o% y 或 outer(x,y)
> tcrossprod(A,B)
     [,1] [,2]
[1,]   22   28
[2,]   49   64
```

```
> tcrossprod(x,y)
     [,1] [,2] [,3] [,4] [,5]
[1,]    2    4    6    8   10
[2,]    4    8   12   16   20
[3,]    6   12   18   24   30
[4,]    8   16   24   32   40
[5,]   10   20   30   40   50

> x %o% y
     [,1] [,2] [,3] [,4] [,5]
[1,]    2    4    6    8   10
[2,]    4    8   12   16   20
[3,]    6   12   18   24   30
[4,]    8   16   24   32   40
[5,]   10   20   30   40   50

> x %*% t(y)
     [,1] [,2] [,3] [,4] [,5]
[1,]    2    4    6    8   10
[2,]    4    8   12   16   20
[3,]    6   12   18   24   30
[4,]    8   16   24   32   40
[5,]   10   20   30   40   50

> outer(x,y)
     [,1] [,2] [,3] [,4] [,5]
[1,]    2    4    6    8   10
[2,]    4    8   12   16   20
[3,]    6   12   18   24   30
[4,]    8   16   24   32   40
[5,]   10   20   30   40   50
```

函数 outer() 用法为

```
outer(X, Y, fun = "*", ...)
```

fun 是外积运算的函数, 做三维曲面时非常有用

12.7 对角矩阵与取对角

```
# 当参数为向量时, 产生对角矩阵
> v<-c(1,4,5)
> diag(v)
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    4    0
[3,]    0    0    5

# 当参数为矩阵时, 取对角元素
> A <- matrix(1:6, nrow=2, byrow=T); A
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
> diag(A)
[1] 1 5
```

12.8 解线性方程组和求矩阵的逆矩阵

求解线性方程组 $Ax = b$, 使用命令 solve(A,b). 求A的逆, 使用命令 solve(A), 实际上把b看作单位矩阵, 结果就是A的逆.

```
> A <- t(array(c(1:8, 10),dim=c(3,3))); A
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8   10
> b <- c(1,1,1)
# 解方程组
> x <- solve(A,b); x
[1] -1.000000e-00  1.000000e-00  3.806634e-16

# 求逆矩阵
> B <- solve(A); B
          [,1]     [,2] [,3]
```

```
[1,] -0.6666667 -1.333333    1
[2,] -0.6666667  3.666667   -2
[3,]  1.0000000 -2.000000    1
```

12.9 求矩阵的特征值与特征向量

```
> A <- t(array(c(1:8, 10),dim=c(3,3))); A
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8   10

# values 为特征值, vectors 的列为对应的特征向量

# 非对称矩阵的特征值与特征向量
> eigen(A)
$values
[1] 16.7074933 -0.9057402  0.1982469

$vectors
          [,1]       [,2]       [,3]
[1,] -0.2235134 -0.8658458  0.2782965
[2,] -0.5039456  0.0856512 -0.8318468
[3,] -0.8343144  0.4929249  0.4801895

# 对称矩阵的特征值与特征向量
> eigen(crossprod(A))
$values
[1] 303.19533618   0.76590739   0.03875643

$vectors
          [,1]         [,2]       [,3]
[1,] -0.4646675  0.833286355  0.2995295
[2,] -0.5537546 -0.009499485 -0.8326258
[3,] -0.6909703 -0.552759994  0.4658502
```

12.10 矩阵分解

12.10.1 三角分解法(LU)

三角分解法是将原正方 (square) 矩阵分解成一个上三角形矩阵 或是排列(permuted) 的上三角形矩阵(L)和一个下三角形矩阵(U), 这样的分解法又称为LU分解法。

$$A = LU$$

它的用途主要在简化一个大矩阵的行列式值的计算过程, 求反矩阵, 和求解联立方程组。

不过要注意这种分解法所得到的上下三角形矩阵并非唯一, 还可找到数个不同的一对上下三角形矩阵, 此两三角形矩阵相乘也会得到原矩阵。

```
> library(Matrix)
> x=matrix(rnorm(9),c(3,3)); x
          [,1]       [,2]       [,3]
[1,] -0.6334882 -0.3915563  0.4906192
[2,]  0.4591368  0.5246114  0.6949097
[3,] -0.4435543 -1.5035618 -0.0191876

# 根据例子, 需要将矩阵转换为 CsparseMatrix 类. why???
> lu(x)
错误于function (classes, fdef, mtable) :
  unable to find an inherited method for function "lu", for signature "matrix"

> A = as(x,"CsparseMatrix")
> p=lu(A)
# 结果是 'MatrixFactorization' of Formal class 'sparseLU'
> p
'MatrixFactorization' of Formal class 'sparseLU' [package "Matrix"] with 5 slots
  ..@ L  :Formal class 'dtCMatrix' [package "Matrix"] with 7 slots
  .. .. ..@ i      : int [1:6] 0 1 2 1 2 2
  .. .. ..@ p      : int [1:4] 0 3 5 6
  .. .. ..@ Dim    : int [1:2] 3 3
```

```
  .. .. ..@ Dimnames:List of 2
  .. .. .. ..$ : NULL
  .. .. .. ..$ : NULL
  .. .. ..@ x      : num [1:6]  1.000  0.700 -0.725  1.000 -0.196 ...
  .. .. ..@ uplo   : chr "L"
  .. .. ..@ diag   : chr "N"
  ..@ U  :Formal class 'dtCMatrix' [package "Matrix"] with 7 slots
  .. .. ..@ i      : int [1:6] 0 0 1 0 1 2
  .. .. ..@ p      : int [1:4] 0 1 3 6
  .. .. ..@ Dim    : int [1:2] 3 3
  .. .. ..@ Dimnames:List of 2
  .. .. .. ..$ : NULL
  .. .. .. ..$ : NULL
  .. .. ..@ x      : num [1:6] -0.633 -0.392 -1.229  0.491 -0.363 ...
  .. .. ..@ uplo   : chr "U"
  .. .. ..@ diag   : chr "N"
  ..@ p  : int [1:3] 0 2 1
  ..@ q  : int [1:3] 0 1 2
  ..@ Dim: int(0)

> p@L
3 x 3 sparse Matrix of class "dtCMatrix"

[1,]  1.0000000  .         .
[2,]  0.7001776  1.0000000 .
[3,] -0.7247755 -0.1958845 1

> p@U
3 x 3 sparse Matrix of class "dtCMatrix"

[1,] -0.6334882 -0.3915563  0.4906192
[2,]  .         -1.2294029 -0.3627082
[3,]  .          .          0.9794496

# L*U 既得原来的矩阵, 行顺序可能不同
> p@L %*% p@U
3 x 3 sparse Matrix of class "dgCMatrix"

[1,] -0.6334882 -0.3915563  0.4906192
[2,] -0.4435543 -1.5035618 -0.0191876
[3,]  0.4591368  0.5246114  0.6949097
```

```
> A
3 x 3 sparse Matrix of class "dgCMatrix"

[1,] -0.6334882 -0.3915563  0.4906192
[2,]  0.4591368  0.5246114  0.6949097
[3,] -0.4435543 -1.5035618 -0.0191876
```

12.10.2 QR分解

QR分解法是将矩阵分解成一个正规正交矩阵(Q)与上三角形矩阵(R)。

$$A = QR$$

正规正交矩阵Q满足的条件

$$QQ^T = I$$

所以称为QR分解法与此正规正交矩阵的通用符号Q有关。

类似的，我们可以定义 A 的 QL, RQ 和 LQ 分解。

更一般的，我们可以因数分解复数 mn 矩阵(有着 $m >= n$)为 mn 酉矩阵 (在 Q*Q = I 的意义上)和 nn 上三角矩阵的乘积。

如果 A 是非奇异的，则这个因数分解是唯一，当我们要求 R 的对角是正数的时候。

QR分解的实际计算有很多方法，例如Givens旋转、Householder变换，以及Gram-Schmidt正交化等等。每一种方法都有其优点和不足。

设 X 为 $n * p$ 矩阵, 可以求得正交矩阵Q, 使得$Q^T X$在主对角线以下为 0. $n >= p$ 时

$$Q^T X = \begin{bmatrix} R \\ 0 \end{bmatrix}$$

其中 R 为上三角矩阵.

将 Q 分割为(Q_1, Q_2), Q_1 有 P 行, 则 $Q^T = Q^{-1}$(正交矩阵的特性 $QQ^T = I$)

$$X = Q \begin{bmatrix} R \\ 0 \end{bmatrix} = [Q_1, Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix} = Q_1 R$$

若 X 有 p 秩(rank), 则由 X 行向量形成的空间可以找到一个正交投影 (orthogonal projection) 矩阵 P

$$P = X(X^T X)^{-1} X^T = Q_1 R(R^T Q_1^T R Q_1)^{-1} R^T Q_1^T = Q_1 Q_1^T$$
$$(Q^T Q = I -> Q_1^T Q_1 = I)$$

另外有矩阵$P_x = Q_2 Q_2^T$为对X垂直方向的投影.

```
> x
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20
> q=qr(x)

# 其中 $qr矩阵上三角为QR分解的R矩阵,
# 下三角为正交矩阵Q的部分信息, 使用压缩存储方法(DQRDC and DGEQP3 differs).
# $qraux 为Q的附加信息.
> q
$qr
          [,1]       [,2]         [,3]         [,4]         [,5]
[1,] -5.4772256 -12.7801930 -2.008316e+01 -2.738613e+01 -3.468910e+01
[2,]  0.3651484  -3.2659863 -6.531973e+00 -9.797959e+00 -1.306395e+01
[3,]  0.5477226  -0.3781696  2.641083e-15  2.056562e-15  5.493622e-15
[4,]  0.7302967  -0.9124744  8.583032e-01 -2.111449e-16  6.562532e-16

$rank
[1] 2

$qraux
```

```
[1] 1.182574e+00 1.156135e+00 1.513143e+00 2.111449e-16 6.562532e-16

$pivot
[1] 1 2 3 4 5

attr(,"class")
[1] "qr"

# $qr的下三角信息结合 $qraux 解压缩为 Q 矩阵
> Q=qr.Q(q); Q
          [,1]         [,2]      [,3]       [,4]
[1,] -0.1825742 -8.164966e-01 -0.4000874 -0.37407225
[2,] -0.3651484 -4.082483e-01  0.2546329  0.79697056
[3,] -0.5477226 -6.163689e-17  0.6909965 -0.47172438
[4,] -0.7302967  4.082483e-01 -0.5455419  0.04882607

# $qr 的上三角矩阵
> R=qr.R(q); R
         [,1]      [,2]         [,3]         [,4]         [,5]
[1,] -5.477226 -12.780193 -2.008316e+01 -2.738613e+01 -3.468910e+01
[2,]  0.000000  -3.265986 -6.531973e+00 -9.797959e+00 -1.306395e+01
[3,]  0.000000   0.000000  2.641083e-15  2.056562e-15  5.493622e-15
[4,]  0.000000   0.000000  0.000000e+00 -2.111449e-16  6.562532e-16
> qr.X(q)
    [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
[3,]    3    7   11   15
[4,]    4    8   12   16

# 重构 x
> Q%*%R
    [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20
```

12.10.3 奇异值分解(svd)

奇异值分解 (sigular value decomposition,SVD) 是另一种正交矩阵分解法; SVD是最可靠的分解法, 但是它比QR 分解法要花上近十倍的计算时间。和QR分解法相同, 原矩阵A不必为正方矩阵。使用SVD分解法的用途是解最小平方误差法和数据压缩。

$$A = UDV^T$$

其中, 其中U和V代表二个相互正交矩阵. D 为对角矩阵, 即 A 的奇异值.

```
> A <- t(array(c(1:8, 10),dim=c(3,3))); A
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8   10

> s=svd(A); s
$d
[1] 17.4125052  0.8751614  0.1968665

$u
          [,1]       [,2]      [,3]
[1,] -0.2093373  0.96438514  0.1616762
[2,] -0.5038485  0.03532145 -0.8630696
[3,] -0.8380421 -0.26213299  0.4785099

$v
          [,1]        [,2]      [,3]
[1,] -0.4646675 -0.833286355  0.2995295
[2,] -0.5537546  0.009499485 -0.8326258
[3,] -0.6909703  0.552759994  0.4658502

> s$u %*% diag(s$d) %*% t(s$v)
     [,1] [,2] [,3]
[1,]    1    2    3
```

```
[2,]    4    5    6
[3,]    7    8   10
```

12.10.4 谱分解

设Q可分解为

$$Q = U\Lambda U^{-1}$$

其中U是非奇异矩阵, 但不必是对称的. svd分解中分解出的两个矩阵是正交的(即对称且乘积为I). $\Lambda = diag(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$是对角矩阵. 那么很明显

$$Q^2 = (U\Lambda U^{-1})(U\Lambda U^{-1}) = U\Lambda^2 U^{-1}$$
$$Q^m = U\Lambda^m U^{-1} = Udiag(\lambda_1^m, \lambda_2^m, \lambda_3^m, \lambda_4^m)U^{-1}$$

λ是Q的特征值, U的列是Q的左特征向量, U^{-1}的行是Q的右特征向量.

上面的分解Q的方法也称为Q的谱分解(spectral decomposition).

```
> x=cbind(c(1,-1),c(-1,1)); x
     [,1] [,2]
[1,]    1   -1
[2,]   -1    1
> y=eigen(x)
> a<-y$values
> u<-y$vectors #
> u%*%diag(a)%*%solve(u)
     [,1] [,2]
[1,]    1   -1
[2,]   -1    1
```

12.11 最小二乘法与QR分解

12.11.1 原理

假设求解一个最小二乘法问题. X 为 $n*p$ 矩阵

$$\rho^2 =\| y - Xb \|^2= min$$

若 X 为行向量间线性独立的矩阵, 则

$$\begin{aligned} X^TXb &= X^Ty \\ b &= (X^TX)^{-1}X^Ty \\ &= (R^TQ_1^TQ_1R)^{-1}R^TQ_1^Ty \\ &= R^{-1}R^{-T}R^TQ_1^Ty \\ &= R^{-1}Q_1^Ty \end{aligned}$$

设

$$z = Q_1^Ty$$

则解$Rb = z$系统即可求得b. 残差向量$r = y - Xb$为y向量投影到X矩阵行向量垂直方向的分量. 由前面

$$r = P_xy = Q_2Q_2^Ty$$

令 $s = Q_2^Ty, r = Q_2s$, 则

$$\rho^2 =\| r \|^2=\| Q_2s \|^2=\| s \|^2$$

对于原来的问题可以删减而得到一个部分系统

$$\rho_1^2 =\| y - X_1^{(1)} \|^2$$

求其最小值

$$b^{(1)} = R_{11}^{-1}Q_1^{(1)T}y \equiv R_{11}^{-1}z$$

$$Q_1 = (Q_1^{(1)}, Q_2^{(1)})$$

相同地

$$z^T = (z_1^T, z_2^T)$$

残差平方和

$$\rho_1^2 = \| Q_2^T y \|^2 + \| Q_2^{(1)T} y \|^2 \equiv \| s \|^2 + \| z_2 \|^2$$

因此, QR因子可以解最小二乘法删去任意组末段行向量的问题.

12.11.2 lsfit()

函数 lsfit() 解最小二乘估计问题中的b向量($coefficients). 下面是一个例子.

```
> x<-c(0.0, 0.2, 0.4, 0.6, 0.8)
> y<-c(0.9, 1.9, 2.8, 3.3, 4.2)
> l <- lsfit(x, y)
> l
$coefficients
Intercept         X
    1.02      4.00

$residuals
[1] -0.12  0.08  0.18 -0.12 -0.02

$intercept
[1] TRUE

$qr
$qt
[1] -5.85849810  2.52982213  0.23749843 -0.02946714  0.10356728

$qr
     Intercept         X
[1,] -2.2360680 -0.8944272
[2,]  0.4472136  0.6324555
[3,]  0.4472136 -0.1954395
```

```
[4,]  0.4472136 -0.5116673
[5,]  0.4472136 -0.8278950

$qraux
[1] 1.447214 1.120788

$rank
[1] 2

$pivot
[1] 1 2

$tol
[1] 1e-07

attr(,"class")
[1] "qr"
```

12.11.3 QR分解

如果使用QR分解, 输入矩阵需要加入一列1元素. 结果与lsfit里的一样.

```
> X<-matrix(c(rep(1,5), x), ncol=2)
> X
     [,1] [,2]
[1,]    1  0.0
[2,]    1  0.2
[3,]    1  0.4
[4,]    1  0.6
[5,]    1  0.8
> qr(X)
$qr
          [,1]       [,2]
[1,] -2.2360680 -0.8944272
[2,]  0.4472136  0.6324555
[3,]  0.4472136 -0.1954395
```

```
[4,]  0.4472136 -0.5116673
[5,]  0.4472136 -0.8278950

$rank
[1] 2

$qraux
[1] 1.447214 1.120788

$pivot
[1] 1 2

attr(,"class")
[1] "qr"
```

12.12 矩阵指数

根据定义矩阵A的指数为矩阵的无穷泰勒展开

$$e^A = I + A + A^2/2! + A^3/3! + \cdots$$

包Matrix的函数expm计算矩阵的指数，使用带有3步前提条件（preconditioning）的Ward's diagonal Pade逼近。此函数来自Octive函数，并对一个小bug做了修正。

包ape的函数matexpo也计算矩阵指数，使用一个特别的矩阵分解方法。下面是函数expm的例子

```
> example(expm)

expm> (m1 <- Matrix(c(1,0,1,1), nc = 2))
2 x 2 Matrix of class "dtrMatrix"
     [,1] [,2]
[1,]    1    1
[2,]    .    1
```

```
expm> (e1 <- expm(m1)) ; e <- exp(1)
2 x 2 Matrix of class "dtrMatrix"
    [,1]    [,2]
[1,] 2.718282 2.718282
[2,]       . 2.718282

expm> stopifnot(all.equal(e1@x, c(e,0,e,e), tol = 1e-15))

expm> (m2 <- Matrix(c(-49, -64, 24, 31), nc = 2))
2 x 2 Matrix of class "dgeMatrix"
    [,1] [,2]
[1,]  -49   24
[2,]  -64   31

expm> (e2 <- expm(m2))
2 x 2 Matrix of class "dgeMatrix"
         [,1]     [,2]
[1,] -0.7357588 0.5518191
[2,] -1.4715176 1.1036382

expm> (m3 <- Matrix(cbind(0,rbind(6*diag(3),0))))# sparse!
4 x 4 sparse Matrix of class "dtCMatrix"

[1,] . 6 . .
[2,] . . 6 .
[3,] . . . 6
[4,] . . . .

expm> (e3 <- expm(m3)) # upper triangular
4 x 4 Matrix of class "dtrMatrix"
    [,1] [,2] [,3] [,4]
[1,]    1    6   18   36
[2,]    .    1    6   18
[3,]    .    .    1    6
[4,]    .    .    .    1

> str(e1)
Formal class 'dtrMatrix' [package "Matrix"] with 5 slots
  ..@ x      : num [1:4] 2.72 0 2.72 2.72
  ..@ Dim    : int [1:2] 2 2
  ..@ Dimnames:List of 2
```

```
.. ..$ : NULL
.. ..$ : NULL
..@ uplo   : chr "U"
..@ diag   : chr "N"
```

Chapter 13

数据的中心化和标准化

13.1 数据挖掘中的变换

数据变换将数据转换或统一成适合于挖掘的形式。数据变换可能涉及如下内容：

- 光滑：去掉数据中的噪声。这种技术包括分箱、回归和聚类。
- 聚集：对数据进行汇总或聚集。例如，可以聚集日销售数据，计算月和年销售量。通常，这一步用来为多粒度数据分析构造数据立方体。
- 数据泛化：使用概念分层，用高层概念替换低层或“原始”数据。例如，分类的属性，如街道，可以泛化为较高层的概念，如城市或国家。类似地，数值属性如年龄，可以映射到较高层概念如青年、中年和老年。
- 规范化：将属性数据按比例缩放，使之落入一个小的特定区间，如－1.0～1.0或0.0～1.0。

13.2 标准化

标准化也叫做 z-score 规范化（零均值规范化）.

$$x'_{ij} = \frac{x_{ij} - \bar{x_j}}{s_j}, \quad i = 1, \cdots, n \quad j = 1, \cdots, p$$

其中

$$\bar{x_j} = \frac{1}{n}\sum_{k=1}^{n} x_{kj}$$

$$s_j = \frac{1}{n-1}\sum_{k=1}^{n}(x_{kj} - \bar{x_j})^2$$

变换后均值为 0, 方差为 1.

记

$$X' = \frac{X - E(X)}{\sqrt{D(X)}}$$

为标准化的随机变量,显然

$$E(X') = 0, D(X') = 1$$

这就是标准化的理由.

函数scale()执行中心化和标准化. 若center为数字或向量, x减去center. center=TRUE 则减去x的平均值, 即center=mean(x). scale=TRUE, 则为x中心化后除以根方差, 若scale为与x等长的向量, 则x除以scale每个值.

```
scale(x, center = TRUE, scale = TRUE)
x=1:10
# 相当于 scale(x,center=TRUE,scale=TRUE), 即标准化
# scale(x, center=mean(x),scale=sqrt(sum((x-center)^2)/(length(x)-1))
> scale(x)
```

```
          [,1]
 [1,] -1.4863011
 [2,] -1.1560120
 [3,] -0.8257228
 [4,] -0.4954337
 [5,] -0.1651446
 [6,]  0.1651446
 [7,]  0.4954337
 [8,]  0.8257228
 [9,]  1.1560120
[10,]  1.4863011
attr(,"scaled:center")
[1] 5.5
attr(,"scaled:scale")
[1] 3.027650

> y=scale(x)
> mean(y)
[1] 0
> var(y)
     [,1]
[1,]    1
> sd(y)
[1] 1

sqrt(sum(x^2)/(length(x)-1)) #  6.540472

# center
> c=2
# scale
> sqrt(sum((x-c)^2)/(length(x)-1))
[1] 4.772607
> scale(x,c)
          [,1]
 [1,] -0.2095291
 [2,]  0.0000000
 [3,]  0.2095291
 [4,]  0.4190582
 [5,]  0.6285873
 [6,]  0.8381164
 [7,]  1.0476454
```

```
 [8,]  1.2571745
 [9,]  1.4667036
[10,]  1.6762327
attr(,"scaled:center")
[1] 2
attr(,"scaled:scale")
[1] 4.772607
```

```
> x=1:10
> mean(x)
[1] 5.5
> var(x)
[1] 9.166667

# 手工计算
> x.zscore=(x-mean(x))/sd(x)
> mean(x.zscore)
[1] 0
> sd(x.zscore)
[1] 1
> x.zscore
 [1] -1.4863011 -1.1560120 -0.8257228 -0.4954337 -0.1651446  0.1651446
 [7]  0.4954337  0.8257228  1.1560120  1.4863011

# 标准化
> y=scale(x);y
           [,1]
 [1,] -1.4863011
 [2,] -1.1560120
 [3,] -0.8257228
 [4,] -0.4954337
 [5,] -0.1651446
 [6,]  0.1651446
 [7,]  0.4954337
 [8,]  0.8257228
 [9,]  1.1560120
[10,]  1.4863011
attr(,"scaled:center")
[1] 5.5
```

```
attr(,"scaled:scale")
[1] 3.027650
> mean(y)
[1] 0
> var(y)
     [,1]
[1,]    1

# center 相当于 mean(x)
> y=scale(x,center=0,scale=F);y
      [,1]
 [1,]    1
 [2,]    2
 [3,]    3
 [4,]    4
 [5,]    5
 [6,]    6
 [7,]    7
 [8,]    8
 [9,]    9
[10,]   10
attr(,"scaled:center")
[1] 0
```

13.3 中心化

n 个样本的 p 维向量中心化为

$$x'_{ij} = x_{ij} - \bar{x_j}, \quad i = 1, \cdots, n \quad j = 1, \cdots, p$$

其中

$$\bar{x_j} = \frac{1}{n} \sum_{k=1}^{n} x_{kj}$$

变换后均值为 0, 方差矩阵不变.

```
> x=1:10
> mean(x)
[1] 5.5
> var(x)
[1] 9.166667

# 中心化
> y=scale(x,scale=F);y
     [,1]
 [1,] -4.5
 [2,] -3.5
 [3,] -2.5
 [4,] -1.5
 [5,] -0.5
 [6,]  0.5
 [7,]  1.5
 [8,]  2.5
 [9,]  3.5
[10,]  4.5
attr(,"scaled:center")
[1] 5.5
> mean(y)
[1] 0
> var(y)
       [,1]
[1,] 9.166667
```

13.4 极差正规化(最小-最大规范化)

最小-最大规范化对原始数据进行线性变换。假定minA和maxA分别为数据A的最小值和最大值。最小-最大规范化通过计算

$$x' = \frac{x - min_A}{max_A - min_A}(newmax_A - newmin_A) + newmin_A$$

将A的值x映射到区间$[newmin_A, newmax_A]$.

映射到区间 $[0,1]$ 称为极差正规化

最小-最大规范化保持原始数据值之间的联系。如果今后的输入落在A的原始数据值域之外，该方法将面临“越界”错误。下面的例子把x映射到$[0,1]$之间

```
> x=1:10
> x1=(x-min(x))/(max(x)-min(x)) *(1-0)+0
> x1
 [1] 0.0000000 0.1111111 0.2222222 0.3333333 0.4444444 0.5555556 0.6666667
 [8] 0.7777778 0.8888889 1.0000000
```

13.5 极差标准化

$$x' = \frac{x - mean(x)}{max_X - min_X}$$

变换后均值为 0, 极差为 1

13.6 小数定标规范化

小数定标规范化通过移动属性A的小数点位置进行规范化。小数点的移动位数依赖于A的最大绝对值. 由下式计算：是使得$Max(|v'|) < 1$的最小整数。假定A的取值由-986～917。A的最大绝对值为986。使用小数定标规范化，用1 000（即i = 3）除每个值，这样，−986规范化为 −0.986，而917被规范化为0.917。

```
> x=rnorm(10)*1000
> x
 [1]   687.82463  -168.41964   -56.08794  -880.85248  -910.98267  1882.82441
```

```
 [7]  -978.97664   736.98754 -1723.98835  -384.87254
> i=ceiling(log(max(abs(x)),10))  # 小数定标的指数
> i
[1] 4
> x/10^i
 [1]  0.068782463 -0.016841964 -0.005608794 -0.088085248 -0.091098267
 [6]  0.188282441 -0.097897664  0.073698754 -0.172398835 -0.038487254
```

注意，规范化将原来的数据改变，特别是上面的后两种方法。有必要保留规范化参数（如均值和标准差，如果使用z-score规范化），以便将来的数据可以用一致的方式规范化。

13.7 正则化(normalize)

变量除以它的范数, 使平方和等于 1.

```
> x=1:10
> x
 [1]  1  2  3  4  5  6  7  8  9 10
> x.nor=x/sqrt(sum(x^2))
> x.nor
 [1] 0.05096472 0.10192944 0.15289416 0.20385888 0.25482360 0.30578831
 [7] 0.35675303 0.40771775 0.45868247 0.50964719

# 平方和等于 1.
> sum(x.nor^2)
[1] 1

# 和与方差皆未知
> sum(x.nor)
[1] 2.803060
> sd(x.nor)
[1] 0.1543033
```

Chapter 14

数据正态化变换

数据变换的目的大概有三种

1. 稳定方差
2. 直线化
3. 使分布正态或接近正态

如果一个变换 $y = f(x)$ 是x的线性函数, 则不影响分析. 但是, 如果是非线性函数, 则y就会表现的和x完全不同, 包括分布方差及数据间的关系.

14.1 误差传播公式(delta方法)–随机变量函数的方差

14.1.1 误差传播公式

参考文献 [17] p67 3.9节.

设n个直接测量的量可以由n维随机向量$X = (X_1, \cdots, X_n)^T$的各分量表示, 其测量误差由X的协方差矩阵表示. 现在要求X的函

数$Y = Y(X)$(间接测量的)方差.

设X的方差是小的, 这时有

$$E[Y(X)] \sim Y(\mu)$$

其中$\mu = \mu_1, \cdots, \mu_n,\ \mu_i$是X的期望. 则Y的方差为

$$V(Y) = E[Y - E(Y(X))]^2 \sim E[Y - E(Y(\mu))]^2$$

将Y在μ附近做泰勒展开有

$$Y = Y(X) \sim Y(\mu) + \sum_{i=1}^{n}(X_i - \mu_i)\frac{\partial Y}{\partial X_i} + o(n)$$

$o(n)$为高次项. 略去高次项代入方差公式得

$$\begin{aligned} V(Y) &\sim \sum_{i=1}^{n}\sum_{j=1}^{n}(\frac{\partial Y}{\partial X_i}\frac{\partial Y}{\partial X_j})_{X=\mu}E[(X_i - \mu_i)(X_j - \mu_j)] \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}(\frac{\partial Y}{\partial X_i}\frac{\partial Y}{\partial X_j})_{X=\mu}V_{ij}(X) \end{aligned}$$

由于X的协方差矩阵$V(X)$是对称的, 上式也可以写作

$$\begin{aligned} V(Y) \sim & \sum_{i=1}^{n}(\frac{\partial Y}{\partial X_i})^2_{X=\mu}V_{ii}(X) \\ & +2\sum_{i<j,j=2}^{n}(\frac{\partial Y}{\partial X_i}\frac{\partial Y}{\partial X_j})_{X=\mu}V_{ij}(X) \end{aligned}$$

这就是误差传播公式. 一般情况下, 它只是近似正确, 因为略去了高阶项. 但是当Y为X的线性函数的时候, 泰勒展开1阶以上的项都为0, 故此时误差传播公式严格正确.

若X_i之间相互独立, 那么$V(X)$非对角项皆为0, 上式变为

$$V(Y) \sim \sum_{i=1}^{n}(\frac{\partial Y}{\partial X_i})^2_{X=\mu}V_{ii}(X) = \sum_{i=1}^{n}(\frac{\partial Y}{\partial X_i})^2_{X=\mu}\sigma_i^2(X_i)$$

即函数的方差为各变量方差的线性和.

更一般的情况, 略...

14.1.2 delta 近似方法

若x,y的非线性函数分别是$f(x).f(x,y)$, 且σ_x^2, σ_y^2已知, 当x,y渐近正态分布且n_x, n_y较大时有

$$var[f(x)] \approx (\frac{df}{dx})^2 var(x)$$

$$var[f(x,y)] \approx (f'(x)^2 var(x) + (f'(y)^2 var(y) + 2(f'(x)(f'(y)cov(x,y)$$

这就是著名的 delta 方法. ([14] Page 556. [15] 第二章)

假定原变量为x, 应用变换 $y = f(x)$, 当x变异系数较小时, 应用第一个式子有

$$var(y) \approx (f'(x)^2 var(x)$$

欲使$var(y)$为常数c, 则应使

$$f'(x) = \frac{c}{\sqrt{var(x)}} = \frac{c}{s}$$

此时可以求得变换

$$y = f(x)$$

其中$var(y) = c$为常数

14.1.3 几种情况下的误差传播公式-加减

参考文献 [17] 3.9节 p70

其中 $U = U(X, Y), u = u(x, y)$

$$U = aX \pm bY$$
$$\sigma^2(U) = a^2\sigma_X^2 + b^2\sigma_Y^2 \pm 2abcov(X, Y)$$

14.1.4 几种情况下的误差传播公式-乘

参考文献 [17] 3.9节 p70

$$U = \pm aXY$$
$$\sigma^2(U)/u^2 = \sigma_X^2/x^2 + \sigma_Y^2/y^2 + 2cov(X, Y)/(xy)$$

14.1.5 几种情况下的误差传播公式-除

参考文献 [17] 3.9节 p70

$$U = \pm aX/Y$$
$$\sigma^2(U)/u^2 = \sigma_X^2/x^2 + \sigma_Y^2/y^2 - 2cov(X, Y)/(xy)$$

14.1.6 几种情况下的误差传播公式-乘幂

参考文献 [17] 3.9节 p70

$$U = aX^{\pm b}$$
$$\sigma(U)/u = b\sigma_X/x$$

14.1.7 几种情况下的误差传播公式-指数1

参考文献 [17] 3.9节 p70

$$U = ae^{\pm bX}$$
$$\sigma(U)/u = b\sigma_X$$

$$U = a \pm bX = (e^{\ln a})^{\pm bX} = e^{\pm (b \ln a) X}$$
$$\sigma(U)/u = b \ln a \sigma_X$$

14.1.8 几种情况下的误差传播公式-对数

参考文献 [17] 3.9节 p70

$$U = a \ln(\pm bX)$$
$$\sigma(U) = a\sigma_X / X$$

14.2 Box-Cox变换

14.2.1 茆诗松的定义

[1]Box与Cox(1964)从实际数据出发提出了一个很有效的变换, 把常用变换作为其特例包含其中, 称为Box-Cox变换. 变换如下

$$y = \begin{cases} x^k & if k \neq 0 \\ lnx & if k = 0 \end{cases}$$

Box-Cox变换有如下特点

- 可以改变分布形状, 使之正态分布, 至少是对称的
- 当$x >= 0$, 能够保持数据的大小次序
- 对变换结果可以有很好的解释
  - k=2 为平方变换
  - k=1 为恒等变换
  - k=0.5 平方根变换

[1]茆诗松. 试验设计. Page 60

  - k=0 对数变换
  - k=-0.5 平方根倒数变换
  - k=-1 倒数变换
- 变换是对k连续的
- 注意: 当$x_{max}/x_{min} > 2$时, 特别有效. $x_{max}/x_{min} <= 2$时无效.

关键是寻找k值, 使变换后的数据正态分布. Montgomery在他的书[2]中提出, k的极大似然估计就是使$y_1, y_2, \cdots, y_n$的偏差平方和$Q(k) = \sum(y_i - \bar{y})^2$达到最小的k值. 可以画出Q(k)的曲线, 读出Q(k)最小的k值即可. 也可以选择10-20个k值, 选择Q(k)最小的k值. 若需要进一步精确估计, 则使用精确网络进一步迭代.

14.2.2 R的定义

经过查询R和百度, 发现定义与茆诗松的描述稍微不同. 变换为

$$y = \begin{cases} (x^k - 1)/k & if k \neq 0 \\ lnx & if k = 0 \end{cases}$$

下面是R的一个例子, 使用最大似然法. box.cox.powers 使用最大似然法按照上面的公式计算指数值. 之后使用box.cox函数得到变换结果.

```
library(car)
attach(Prestige)
# 两个变量会计算多元正态分布
box.cox.powers(cbind(income, education))
par=matrix(c(1,2))
plot(income, education)
plot(box.cox(income, .26), box.cox(education, .42))
# 单变量会直接转换为正态分布
box.cox.powers(income)
```

[2]Montgomery. 实验设计与分析(第三版). 1998.

```
qq.plot(income) # car 包的绘图函数
qq.plot(income^.18)
```

还有一个扩展形式, 其a值比较明显, 还是估计k值的问题.

$$y = \begin{cases} ((x+a)^k - 1)/k & if k \neq 0 \\ lnx & if k = 0 \end{cases}$$

在Box和Cox论文中采用了两种方法，其一是最大似然估计，其二是Bayes方法。

14.3 稳定方差的变换

14.3.1 对数变换-方差正比于自变量的平方

当$var(x) \propto x^2$, 即x增大, $var(x)$也增大, 那么有

$$s_x = kx$$

此时x的变异系数$cv = s/\bar{x} = c$为常数.

将$s = kx$带入下式

$$f'(x) = \frac{c}{\sqrt{var(x)}} = \frac{c}{s}$$

合并常数项后有

$$f'(x) = \frac{c}{x} \Longrightarrow f(x) = lg(x)$$

即$y = lg(x)$使方差稳定.

使用的时候, 注意: $x_i <= 0$时不能使用. 将$x_i <= 0$替换为$-x_i$即可. 若有$x = 0$, 常常用$lg(x+1)$代替$lg(x)$, $x = 0$时, $y = 0$.

```
> rep(1:10,10)
  [1]  1  2  3  4  5  6  7  8  9 10  1  2  3  4  5  6  7  8  9 10  1  2  3  4  5
 [26]  6  7  8  9 10  1  2  3  4  5  6  7  8  9 10  1  2  3  4  5  6  7  8  9 10
 [51]  1  2  3  4  5  6  7  8  9 10  1  2  3  4  5  6  7  8  9 10  1  2  3  4  5
 [76]  6  7  8  9 10  1  2  3  4  5  6  7  8  9 10  1  2  3  4  5  6  7  8  9 10
> matrix(rep(1:10,10),nc=10) # nr =10也一样
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]    1    1    1    1    1    1    1    1    1     1
 [2,]    2    2    2    2    2    2    2    2    2     2
 [3,]    3    3    3    3    3    3    3    3    3     3
 [4,]    4    4    4    4    4    4    4    4    4     4
 [5,]    5    5    5    5    5    5    5    5    5     5
 [6,]    6    6    6    6    6    6    6    6    6     6
 [7,]    7    7    7    7    7    7    7    7    7     7
 [8,]    8    8    8    8    8    8    8    8    8     8
 [9,]    9    9    9    9    9    9    9    9    9     9
[10,]   10   10   10   10   10   10   10   10   10    10
> c(t(matrix(rep(1:10,10),nc=10)))
  [1]  1  1  1  1  1  1  1  1  1  1  2  2  2  2  2  2  2  2  2  2  3  3  3  3  3
 [26]  3  3  3  3  3  4  4  4  4  4  4  4  4  4  4  4  5  5  5  5  5  5  5  5  5  5
 [51]  6  6  6  6  6  6  6  6  6  6  7  7  7  7  7  7  7  7  7  7  8  8  8  8  8
 [76]  8  8  8  8  8  9  9  9  9  9  9  9  9  9  9 10 10 10 10 10 10 10 10 10 10
> s=c(t(matrix(rep(1:10,10),nc=10)))
> x=rnorm(100)
> plot(y<-x*s) # y的方差与x呈正比
> plot(log(y)+10,ylim=c(0,15)) # 查看log变换的y方差基本一致
```

14.3.2 平方根变换-方差正比于自变量

若$var(x) \propto x$, 则有

$$f'(x) = \frac{c}{\sqrt{x}} \Longrightarrow f(x) = \sqrt{x}$$

例如 $var(x) = kx \Longrightarrow var(y) = [(\sqrt{x})']^2(kx) = k/4$, 可见方差稳定.

当x绝对值较小时,常用$y = \sqrt{x+1}$或$y = \sqrt{x} + \sqrt{x+1}$, 效果好于$y = \sqrt{x}$.

```
> s1=c(t(matrix(rep((1:20)^0.5,10),nc=10)))
> x1=rnorm(200)
> plot(s1)
> plot(y1<-s1*x1) # 方差随x变大
> plot(sqrt(y1)) # 看到方差已经基本一致
```

14.3.3 反正弦变换(角变换)-百分率表示的数据

应用于以百分率表示的数据. 即在n次试验中, 成功k次, 则有

$$p = \frac{k}{n}$$

$$var(p) = p(1-p)/n$$

即$var(p)$与p有关. 我们有

$$f'(p) = \frac{c}{\sqrt{p(1-p)}} \Longrightarrow y = sin^{-1}\sqrt{p}$$

可以验证

$$var(y) = (\frac{c}{\sqrt{p(1-p)}})^2 \frac{p(1-p)}{n} = \frac{c^2}{4n}$$

注意, 当多组样本数不同时, 使用各自的样本数加权.

14.3.4 倒数变换-方差正比于自变量4次方

若$var(x) \propto x^4$,即$s = kx^2$, x增加时, var(x)增加很快. 我们有

$$f'(x) = \frac{c}{x^2} \Longrightarrow y = \frac{1}{x}$$

这样, x增加到一定程度后y的减小微不足道. 验证

$$var(y) = (\frac{c}{x^2})^2 kx^4 = kc^2$$

常常用于质反应时间为指标的数据

14.4 量反应直线化

某些数据上凸或下凹, 则使用多种变换尺度(metameter)使之直线化. 理想的变换应该使

1. y与x呈直线关系
2. var(y) 稳定
3. 直线斜率较大

14.4.1 对数变换

又分为两种, 单对数变换:

$$y = a + b * lg(x)$$

双对数变换:

$$lg(y) = a + b * lg(x)$$

14.4.2 平方根变换

当对数变换后仍然开始较陡峭, 以后平坦(变换前适合二次拟合), 则使用下面的变换:

$$\sqrt{y} = a + b * lg(x)$$

变换后方差也较变换前稳定

14.4.3 倒数变换

单倒数变换.

下面是一个蛋白质与底物反应的例子. 设x为游离蛋白质浓度, y为游离底物浓度, m为结合物浓度. X=x+m为总蛋白质浓度, Y=y+m为总底物浓度. 已知X,Y, 可以测得y,m,但x不易测定. 根据质量作用定律

$$\frac{xy}{m} = k$$

此处k为常数. k大,说明容易结合, 否则不易结合. 欲得到x或y与k的关系(视测量难度选取容易的). 测得$(m_1, y_1), \cdots, (m_n, y_n)$. 以此做曲线拟合. 由上式

$$\frac{x}{k} = \frac{m}{y} \Longrightarrow \frac{X-m}{k} = \frac{m}{y} \Longrightarrow \frac{m}{y} = \frac{m}{-k} + \frac{X}{k}$$

可以看到$\frac{m}{y}$与m呈直线关系. 使用$\frac{m}{y}$与m作图可以得到一条直线.

双倒数作图法(double reciprocal, Linewaver-Burk plot) 药物动力学常常符合此种情况.

设x为药物浓度, y为效应, k为解离常数, $a = y_{max}$为内在活性. 根据Ariens学说有

$$y = \frac{ax}{x+k} \Longrightarrow \frac{1}{y} = \frac{1}{a} + \frac{k}{a}\frac{1}{x}$$

即$1/y, 1/x$呈线性关系. 此时由数据线性回归即可求得k.

14.5 质反应直线化

质反应是反应特定反应的有无,死活等离散数据的反应. 反应特点一般呈S曲线. 其成功概率

$$p \sim N(\hat{p}, \hat{p}\hat{q}/n)$$

标准正态偏离定义为

$$z_p = \frac{p - \hat{p}}{\sqrt{\hat{p}\hat{q}/n}}$$

14.5.1 probit变换(概率单位变换)

probit变换又叫做概率单位变换(probability unit). probit定义为$y = 5 + z$, 可以使S曲线直线化. 但是$var(y) \neq c$常数. 故不适合最小二乘法. 常常使用最大似然法(maximum likelyhood, ML)

14.5.2 角变换

$y = sin^{-1}_{\sqrt{p}}$可以使S曲线直线化. 当$p \in [0, 1]$时, $y \in [0, \pi/2]$. 即p等距离变化时, y的两端变化大, 中间小, 使S曲线拉直.

$$var(y) = \frac{820.7}{n(90 * 2/\pi)^2}$$

故n不变时, 比probit变换易于分析.

14.5.3 logit变换

p的logit变换定义为

$$y = ln\frac{p}{1-p} \quad or \quad y = \frac{1}{2} + ln\frac{p}{1-p}$$

其效果与probit相似. 当$p = 0, 1$时,$y = -\infty, \infty$. 故修正为

$$y = ln\frac{r + 1/2}{n - r + 1/2}$$

其中$r = np$.

14.6 相关系数的正态化变换–Fisher变换(Z变换)

参考回归部分 chapter 28章 section 29.2节

14.7 总结

很多右偏数据可以正态化.

对数变换后呈正态分布, 又称对数正态分布, 方差稳定.

不太严重的右偏, 使用平方根变换

严重右偏, 倒数变换

Chapter 15

距离系数

参考 [11] 3.1 距离系数

15.1 基本性质

距离系数一般应该满足下面三个基本性质

1. $d_{AB} >= 0$, 当且仅当 $A = B$ 时成立
2. $d_{AB} = d_{BA}$
3. $d_{AB} <= d_{AC} + d_{CB}$ (三角不等式)

有时候第三条修改为

- $d_{AB} <= max(d_{AC}, d_{CB})$

比原来的三角不等式要强, 因为 $max(d_{AC}, d_{CB}) <= d_{AC} + d_{CB}$

R 函数计算各种距离, 包括 ”’euclidean”’, ”’maximum”’, ”’manhattan”’, ”’canberra”’,”’binary”’ or ”’minkowski”’.

注意: 各个系数如果求和之后除以 p 再进行开方运算, 就变成平均 XX 距离系数. 例如平均欧氏距离变为

$$d(x,y)=\sqrt{\frac{1}{n}[(x_1-y_1)^2+\cdots+(x_p-y_p)^2]}=[\frac{1}{n}\sum_{i=1}^{p}(x_i-y_i)^2]^{\frac{1}{2}}$$

但是 R 并没有平均距离的函数.

15.2 绝对距离(曼哈顿距离, absolute distance)

$$d(x,y)=\sum_{i=1}^{p}|x_i-y_i|$$

```
> x=matrix(c(0,1,2,3,0,1,2,3),ncol=2)
> x
     [,1] [,2]
[1,]    0    0
[2,]    1    1
[3,]    2    2
[4,]    3    3

> dist(x,diag=T,method="manhattan")
  1 2 3 4
1 0
2 2 0
3 4 2 0
4 6 4 2 0
```

15.3 欧氏距离(Euclidean distance)

p 维空间的两点 $x=(x_1,\cdots,x_p)^T, y=(y_1,\cdots,y_p)^T$, 其欧氏距离系数为

$$d(x,y)=\sqrt{(x_1-y_1)^2+\cdots+(x_p-y_p)^2}=[\sum_{i=1}^{p}(x_i-y_i)^2]^{\frac{1}{2}}$$

```
# 默认计算欧氏距离系数
> x=matrix(c(0,1,2,3,0,1,2,3),ncol=2)
> dist(x,method = "euclidean", diag=T, upper = FALSE)
       1        2        3        4
1 0.000000
2 1.414214 0.000000
3 2.828427 1.414214 0.000000
4 4.242641 2.828427 1.414214 0.000000
```

15.4 Minkowski 距离(明氏距离)

$$d(x,y)=\sqrt[r]{|x_1-y_1|^r+\cdots+|x_p-y_p|^r}=[\sum_{i=1}^{p}|x_i-y_i|^r]^{\frac{1}{r}}$$

其中$r>0$. 这个系数常常被化学分类学使用, 比较两个薄层层析的差异. r 充分小时, 对较小的差异敏感, 故适合差异较小的分类单位之间建立相似性比较.

$r=1$ 时转化为 曼哈顿距离, $r=2$ 时转化为欧氏距离.

```
> x=matrix(c(0,1,2,3,0,1,2,3),ncol=2)
> dist(x,diag=T,method="minkowski",p=0.5)
  1  2  3  4
```

```
1  0
2  4  0
3  8  4  0
4 12  8  4  0
> dist(x,diag=T,method="minkowski",p=3)
         1        2        3        4
1 0.000000
2 1.259921 0.000000
3 2.519842 1.259921 0.000000
4 3.779763 2.519842 1.259921 0.000000
```

15.5 Chebyshev 距离

$$d_{ij}(\infty) = max_{1<=k<=p}|x_{ik} - x_{jk}|$$

是 Minkowski 距离 $r \to \infty$ 时的情况

dist() 函数 method=”maximum” 是计算 Chebyshev 距离.

```
> x=matrix(c(0,1,2,3,0,1,2,3),ncol=2)
> x
     [,1] [,2]
[1,]    0    0
[2,]    1    1
[3,]    2    2
[4,]    3    3
> dist(x,diag=T,method="maximum")
  1 2 3 4
1 0
2 1 0
3 2 1 0
4 3 2 1 0
```

15.6 Canberra 距离

实际上是 Lance 距离的扩展, 不要求 $x_{ij} > 0$

$$d(x,y) = \sum_{i=1}^{p} \frac{|x_i - y_i|}{|x_i + y_i|}$$

```
> x=matrix(c(0,1,2,3,0,1,2,3),ncol=2)
> dist(x,diag=T,method="canberra")
        1        2        3        4
1 0.0000000
2 2.0000000 0.0000000
3 2.0000000 0.6666667 0.0000000
4 2.0000000 1.0000000 0.4000000 0.0000000
```

15.7 分离系数

与 Canberra 距离系数类似

$$d(x,y) = [\sum_{i=1}^{p} (\frac{x_i - y_i}{x_i + y_i})^2]^{\frac{1}{2}}$$

15.8 Lance 和 Williams 距离

实际上是 Canberra 距离的特殊形式.

$$d_{ij}(L) = \sum_{k=1}^{p} \frac{|x_{ik} - x_{jk}|}{x_{ik} + x_{jk}}$$

其中

$$\begin{aligned} x_{ij} &> 0 \\ i &= 1, 2, \cdots, n \\ j &= 1, \cdots, p \end{aligned}$$

用法使用 method="canberra" 即可.

15.9 Mahalanobis distance(马氏距离)

参考

- [21] 8.1 判别分析
- http://en.wikipedia.org/wiki/Mahalanobis_distance

设总体 $X = [x_{ij}]_{n*p}$ 为 p 维空间中的 n 个点, 均值为 $\mu = (\mu_1, \mu_2, \cdots, \mu_p)^T$), 协方差矩阵 $\sum$ 为 $p*p$ 的方阵. 则 p 维空间中一个样本点 $x = (x_1, \cdots, x_p)^T$ 与总体 X 的 Mahalanobis 距离为

$$d(x, X) = \sqrt{(x-\mu)^T \sum^{-1} (x-\mu)}$$

其中 $\sum^{-1}$ 为 $\sum$ 的逆矩阵.

实际上是对 x 标准化.

总体内两个点 x,y (即服从均值 μ, 协方差矩阵方差 $\sum$) 之间的 Mahalanobis 距离定义为

$$d(x, y) = \sqrt{(x-y)^T \sum^{-1} (x-y)}$$

实际上是标准化了的 x,y 之间的距离.

R 函数 mahalanobis() 计算 Mahalanobis 距离. 但是未开方用法为

```
mahalanobis(x, center, cov, inverted=FALSE, ...)
```

其中

- x 为 p 维的向量(一个点)或 p 列的矩阵(多个点).
- center 为 p 维向量, 代表总体均值. 如果给出的不是均值, 而是另外一个 p 维向量 y, 则函数计算的就是 x,y之间的 Mahalanobis 距离
- cov 代表 $p*p$ 的协方差矩阵
- inverted=TRUE 代表给出的协方差矩阵已经求逆. 否则函数会计算其逆.

返回 x 的每个点与 X(均值) 的 Mahalanobis 距离或 x,y 之间的 Mahalanobis 距离(未开方的)

考虑一维的例子, 实际上就是 x 的标准化 (函数计算的是未开方的结果)

```
> X=c(1:10)
> X
 [1]  1  2  3  4  5  6  7  8  9 10
> mu=mean(X); mu
[1] 5.5
> cov=var(X); cov
[1] 9.166667

> dist.mahalanobis(0,mu,cov)
    [,1]
[1,]  3.3
> mahalanobis(0,mu,cov)
[1] 3.3
# 实际上是标准化
```

```
> (0-mu)*(1/cov)*(0-mu)
[1] 3.3
> (0-mu)^2/cov
[1] 3.3

# 下面为两个总体, 查看样本 x=15 与两个总体的 Mahalanobis 距离
> X1=c(1:10)
> X2=c(11:20)
> mu1=mean(X1)
> mu2=mean(X2)
> cov1=var(X1)
> cov2=var(X2)

> mahalanobis(15,mu1,cov1)
[1] 9.845455
> mahalanobis(15,mu2,cov2)
[1] 0.02727273
```

下面是二维的例子 (函数计算的是未开方的结果)

```
> X=matrix(c(1:10,1:5,10:6),ncol=2)
> X
      [,1] [,2]
 [1,]    1    1
 [2,]    2    2
 [3,]    3    3
 [4,]    4    4
 [5,]    5    5
 [6,]    6   10
 [7,]    7    9
 [8,]    8    8
 [9,]    9    7
[10,]   10    6

> a=colMeans(X)[1]; a
[1] 5.5
> b=colMeans(X)[2]; b
```

```
[1] 5.5

# 绘制X
> plot(X)
# 均值点 (center)
> points(a,b,col='red')

# 编制点 x 与总体 X 的距离函数
# 如果 mu 给出的不是均值, 而是另外一个 p 维向量 y,
# 则函数计算的就是 x,y之间的 Mahalanobis 距离
dist.mahalanobis<-function(x,mu,cov){
   r <- (x-mu)%*%solve(cov)%*%(x-mu)
   r
}

# 以 X 为总体, 计算均值与协方差矩阵
> mu=colMeans(X)
> cov=cov(X)
# 计算点 (0,0) 与 X 的距离
> dist.mahalanobis(c(0,0),mu,cov)
       [,1]
[1,] 3.755172
# 计算点 (1,1) 与 X 的距离
> dist.mahalanobis(c(1,1),mu,cov)
       [,1]
[1,] 2.513793

# 下面使用 R 中的函数计算
> x=matrix(c(0,1,0,1),nrow=2)
> x
    [,1] [,2]
[1,]    0    0
[2,]    1    1

> mahalanobis(x,mu,cov)
[1] 3.755172 2.513793

# 计算 x,y之间的距离
> a=c(0,0)
> b=c(1,1)
> mahalanobis(a,b,cov)
```

```
[1] 0.1241379
> dist.mahalanobis(a,b,cov)
        [,1]
[1,] 0.1241379
```

15.10 二值定性距离

两个 p 维向量 X_i, X_j 元素是二值数据时, 设 0 代表无, 1 代表有. 两个样本都有 p 个值. 第 k 个都是 0, 称在第 k 个值 0-0 配对; 第 k 个都是 1, 称在第 k 个值 1-1 配对; 若第 k 个不一样, 称在第 k 个值不配对.

记 m_0, m_1 分别为 0-0 配对和 1-1 配对的个数, m_2 为不配对的个数. 显然有

$$m_0 + m_1 + m_2 = p$$

两个样本的距离可以定义为

$$d_{ij} = \frac{m_2}{m_1 + m_2}$$

dist() 函数 method="binary" 即计算二值定性距离. 值为非零作为 "on", 值为零的作为 "off" 对待.

下面例子中, 不配对有2个, 1-1 配对有3个故距离为

$$d = 2/(2+3) = 0.4$$

```
> x <- c(0, 0, 1, 1, 1, 1)
> y <- c(1, 0, 1, 1, 0, 1)

> dist(rbind(x,y), method= "binary")
   x
y 0.4
```

Chapter 16

相似系数

参考 [11] 第 3 章

设 r_{ij} 为变量 X_i, X_j 之间的相似系数. 一般要求

- $r_{ij} = \pm 1$ 当且仅当 $X_i = aX_j (a \neq 0)$
- $|r_{ij}| <= 1$ 对一切 i,j 成立
- $r_{ij} = r_{ji}$ 对一切i,j成立

$|r_{ij}|$ 越接近 1, 表示关系越密切, 越接近 0, 关系越疏远.

16.1 角余弦系数

变量 X_i, X_j 的角余弦系数(coefficient of cosine of included angle)定义为

$$\begin{aligned} r_{ij} &= \frac{\sum_{k=1}^{n} x_{ki} x_{kj}}{\sqrt{(\sum_{k=1}^{n} x_{ki}^2)(\sum_{k=1}^{n} x_{kj}^2)}} \\ &= \frac{X_i X_j^T}{\| X_i \| \| X_j \|} \end{aligned}$$

实际上是未标准化的相关系数. 两个变量正交时, $r=0$. 完全相似时, $r=\pm 1$

设两个变量的夹角为 θ, 则

$$cos\theta = \frac{X_i X_j^T}{\parallel X_i \parallel \parallel X_j \parallel} = r_{ij}$$

16.2 相关系数

最常用的相关系数就是 Pearson 乘积矩关联系数. 实际上是标准化的角余弦系数. 也是中心化与标准化后的协方差. 协方差见20部分.

$$r_{ij} = \frac{\sum_{k=1}^{n}(x_{ki}-\bar{x_i})(x_{kj}-\bar{x_j})}{\sqrt{(\sum_{k=1}^{n}(x_{ki}-\bar{x_i})^2)(\sum_{k=1}^{n}(x_{kj}-\bar{x_j})^2)}}$$

```
> x1=1:10
> x2=11:20

===============
# 角余弦系数

> a=sum(x1*x2)
> b=sqrt(sum(x1^2)*sum(x2^2))
> r=a/b; r
[1] 0.9559123

# x3 与 x1 完全相似
> x3=2*x1
> a=sum(x1*x3)
> b=sqrt(sum(x1^2)*sum(x3^2))
> r=a/b; r
[1] 1

# 负相关
```

```
> x4=-2*x1
> a=sum(x1*x4)
> b=sqrt(sum(x1^2)*sum(x4^2))
> r=a/b; r
[1] -1

===============
# 相关系数
> cor(x1,x2)
[1] 1
> cor(x1,x3)
[1] 1
> cor(x1,x4)
[1] -1
```

16.3 联合系数(assosiation coefficient, confusion matrix)

设两个 n 维向量 X_i, X_j 元素是离散数据(二值或多值数据), 联合系数是它们之间一致性度量的函数. 大部分情况以二值数据出现, 这里假定取二值数据 0,1.

第 k 个元素的匹配有四种情况: 0-0 匹配和 1-1 匹配, 0-1 不匹配, 1-0 不匹配. 下表为各种匹配的个数, 明显 $a+b+c+d=n$

	1	0
1	a	b
0	c	d

```
> x <- c(0, 0, 1, 1, 1, 1)
> y <- c(1, 0, 1, 1, 0, 1)
> x==0 & y==0
[1] FALSE  TRUE FALSE FALSE FALSE FALSE
```

```
# a,b,c,d 各种匹配的个数
> a=sum(x==0 & y==0); a
[1] 1
> d=sum(x==1 & y==1); d
[1] 3
> b=sum(x==1 & y==0); b
[1] 1
> c=sum(x==0 & y==1); c
[1] 1
```

最简单的考虑就是计算匹配一致的个数占总个数的百分比(下表第 6 个公式)

$$S = \frac{a+d}{n}$$

16.4 各种系数列表

下面是各种系数的列表注释: $A = \sqrt{(a+b)(a+c)}, D = \sqrt{(d+b)(d+c)}$

联合系数的选择没有同一的标准. 大部分的联合系数对 a 强调, 忽视 d. 徐克学等(1989)[11] (page 98) 设计了联合系数的普遍公式.

公式: 略

编号	公式	作者/系数名称	范围
1	$\frac{a}{n}$	Russell and Rao, 1940	$[0,1]$
2	$\frac{a}{a+2(b+c)}$	Sokal and Sneath, 1963	$[0,1]$
3	$\frac{a}{a+b+c}$	Jaccard, 1908	$[0,1]$
4	$\frac{a}{2a+b+c}$	Czekanowski, 1913	$[0,1]$
5	$\frac{a+d}{n+b+c}$	Rogers and Tanimoto, 1960	$[0,1]$
6	$\frac{a+d}{n}$	Simple Matching	$[0,1]$
7	$\frac{2(a+d)}{n+a+d}$	Sokal and Sneath, 1963	$[0,1]$
8	$\frac{ad}{ad+bc}$	Unnamed coefficient	$[0,1]$
9	$\frac{2a}{2a+ab+ac+bc}$	Unnamed coefficient	$[0,1]$
10	$\frac{a}{2}(\frac{1}{a+b}+\frac{1}{a+c})$	Kulczynski, 1927	$[0,1]$
11	$\frac{1}{4}(\frac{a}{a+b}+\frac{a}{a+c}+\frac{d}{d+b}+\frac{d}{d+c})$	Sokal and Sneath, 1963	$[0,1]$
12	$\frac{a}{A}$	Ochiai, 1957	$[0,1]$
13	$\frac{ad}{AD}$	Sokal and Sneath, 1963	$[0,1]$
14	$\frac{AD+ad-bc}{2AD}$	Unnamed coefficient	$[0,1]$
15	$\frac{ad-bc}{AD}$	correlation coefficient, Guifford,1942	$[-1,1]$
16	$\frac{a^2-bc}{A^2}$	McConnaughy, 1964	$[-1,1]$
17	$\frac{a+d-b-c}{n}$	Hamann, 1961	$[-1,1]$
18	$\frac{ad-bc}{ad+bc}$	Yule and Kendall, 1950	$[-1,1]$
19	$\frac{a+d}{b+c}$	Sokal and Sneath, 1963	$[0,\infty)$
20	$\frac{a}{b+c}$	Kulczynski, 1927	$[0,\infty)$
21	$\frac{2a}{ab+ac+bc}$	Sneath and Sokal, 1973	$[0,\infty)$
22	$\frac{b+c}{2a+b+c}$	Watson et. al., 1966	$[0,1]$
23	$\frac{b+c}{n}$	Euclidean Distance	$[0,1]$
22	$\frac{a}{A}-\frac{1}{2\sqrt{a+b}}$	Fager and McGowan, 1963	$(-\infty,1]$

Part III

基本统计分析

“基本统计分析”参考文献除了[14]，R部分主要参考了《simpleR》《Statistics with R》等。

Chapter 17

数据类型的划分

此处的数据类型并不是R中的数据基本类型和其组织结构(数值型，复数型，逻辑型，字符型和原味型(raw), 以及向量, 矩阵等), 而是数据最基本的度量性质. 包括3种.

非参数统计假设数据是基数尺度或有序尺度. 如果基数尺度数据样本很小或分布形状未知, 中心极限定理似乎又不适用, 则使用非参数方法最有效.

17.1 基数数据(cardinal data)

可以使用某种尺度测出任何两个数据的距离. 基数数据(包括区间尺度数据和比例尺度数据)的均值和标准差都是有意义的.

例如, 体重是基数数据, 差异6磅是差异3磅的2倍.

17.1.1 区间尺度数据(interval scale data)

对于基数数据, 如果零点是任意的(零点意义不明确), 称为区间尺度数据. 其比值可能没有意义.

例如: 体温是区间尺度数据, 因为它的零点不固定, 华氏和摄氏中, 其零点的意义是不同的. 华氏和摄氏温度的比值没有意义.

17.1.2 比例尺度数据(ratio scale data)

零点固定(零点意义明确), 称为比例尺度数据. 任何两个数据的比值是有意义的.

例如: 血压和体重, 身高是比例尺度数据. A的体重比B多10%.

17.2 有序数据(ordinal data)

可以排列次序, 比较大小, 但是没有指定的数值, 通常的算数运算没有意义. 计算均值和标准差通常是不合适的.

例如: 视敏度虽然有值表示, 但是只是表示相对大小, 不能进行数学运算. 视力也一样. 病情的严重程度也可以用1,2,3,4等来表示, 但是运算同样没有意义.

17.3 名义尺度数据(nominal scale data)

不同的数值代表的是类型, 而类型是没有次序的, 当然数学运算也是没有意义的.

例如: 可以使用一列数字代表各种不同的死亡原因, 药物类型, 工作种类等. 但是它们是没有次序的.

Chapter 18

描述性统计

18.1 探索性分析

也可以叫做经验性数据分析. 目的是看一看数据适合哪一种统计模型. 对于单变量数据, 我们可以看看它的分布是否正态, 尾部偏大还是偏小, 对称还是偏态.

主要的工具就是图形工具.

- barplots for categorical data(类型数据)
- histogram, dot plots, stem and leaf plots to see the shape of numerical distributions
- boxplots to see summaries of a numerical distribution, useful in comparing distributions and identifying long and short-tailed distributions.
- normal probability plots To see if data is approximately normal

18.2 样本特征数

若非特别说明, 数据使用这个

```
> x=exp(seq(-1,3,by=0.1))
> x
 [1]  0.3678794  0.4065697  0.4493290  0.4965853  0.5488116  0.6065307
 [7]  0.6703200  0.7408182  0.8187308  0.9048374  1.0000000  1.1051709
[13]  1.2214028  1.3498588  1.4918247  1.6487213  1.8221188  2.0137527
[19]  2.2255409  2.4596031  2.7182818  3.0041660  3.3201169  3.6692967
[25]  4.0552000  4.4816891  4.9530324  5.4739474  6.0496475  6.6858944
[31]  7.3890561  8.1661699  9.0250135  9.9741825 11.0231764 12.1824940
[37] 13.4637380 14.8797317 16.4446468 18.1741454 20.0855369
> plot(x)
```

18.2.1 方差

若$E(X-E(X))^2$存在, 则称之为X的方差, 并记为

$$D(X)=E(X-E(X))^2$$

根据期望的定义, $D(X)=E(X^2)-(E(X))^2$.

1. $D(X)=0 \Longleftrightarrow p(X=C)=1$ 即X为常数
2. $D(aX)=a^2D(X)$
3. 若$a \neq E(X)$则$E(X-a)^2 > D(X)=E(X-E(X))^2$
4. 若X和Y相互独立并且都有有限方差,则

$$D(X+Y)=D(X)+D(Y)$$

(柯西-施瓦茨不等式)

$$E(XY)^2 <= E(X^2)E(Y^2)$$

(这个不等式的证明方法有很多)

```
> var(x)
[1] 29.35325
```

18.2.2 标准差

标准差称$\sqrt{D(X)}$为根方差或标准差

```
> sd(x)
[1] 5.417864
# 可以手工计算验证一下
> sqrt((sum(x^2)-(sum(x))^2/length(x))/(length(x)-1))
[1] 5.417864
```

18.2.3 最大最小值

```
> max(x)
[1] 20.08554
> min(x)
[1] 0.3678794
```

18.2.4 累积最大最小值

```
> cummax(x)
 [1]  0.3678794  0.4065697  0.4493290  0.4965853  0.5488116  0.6065307
 [7]  0.6703200  0.7408182  0.8187308  0.9048374  1.0000000  1.1051709
[13]  1.2214028  1.3498588  1.4918247  1.6487213  1.8221188  2.0137527
[19]  2.2255409  2.4596031  2.7182818  3.0041660  3.3201169  3.6692967
[25]  4.0552000  4.4816891  4.9530324  5.4739474  6.0496475  6.6858944
[31]  7.3890561  8.1661699  9.0250135  9.9741825 11.0231764 12.1824940
[37] 13.4637380 14.8797317 16.4446468 18.1741454 20.0855369
> cummin(x)
 [1] 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794
 [8] 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794
[15] 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794
[22] 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794
[29] 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794
[36] 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794
```

18.2.5 差分

```
> diff(x)
 [1] 0.03869022 0.04275930 0.04725634 0.05222633 0.05771902 0.06378939
 [7] 0.07049817 0.07791253 0.08610666 0.09516258 0.10517092 0.11623184
[13] 0.12845605 0.14196589 0.15689657 0.17339753 0.19163391 0.21178822
[19] 0.23406218 0.25867872 0.28588420 0.31595090 0.34917974 0.38590330
[25] 0.42648910 0.47134335 0.52091497 0.57570007 0.63624698 0.70316166
[31] 0.77711381 0.85884359 0.94916896 1.04899393 1.15931758 1.28124407
[37] 1.41599369 1.56491505 1.72949860 1.91139155
```

18.2.6 平均值

计算平均值的时候，无论 x 是多少维的，都计算所有的 x 的值

```
> y=array(1:20,dim=c(4,5))
> y
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20
> mean(y)
[1] 10.5
> colMeans(y) # 行均值
[1]  2.5  6.5 10.5 14.5 18.5
> rowMeans(y) # 列均值
[1]  9 10 11 12
```

18.2.7 中位数

```
> median(x)
[1] 2.718282
```

18.2.8 众数

```
> y=c(1,1,2,2,2,3,4)*2
> y
[1] 2 2 4 4 4 6 8
> table(y)
y
2 4 6 8
2 3 1 1
> max(table(y)) # 众数出现的次数
[1] 3
> table(y)==max(table(y))
y
   2    4    6    8
FALSE  TRUE FALSE FALSE
> which(table(y)==max(table(y))) # 众数在table(y)第几个? 第2个
4
2
```

18.2.9 偏斜度(skewness)

moments 包里面有 skewness 和 kurtosis 函数。自己编一个很简单，由偏斜度公式

$$m_3 = \frac{\sum(x-\bar{x})^3}{n}$$

编写函数(偏斜度):

```
skewness<-function(x){
   sum(((x-mean(x))^3))/length(x)
}
# 计 算 结 果
> skewness(x)
[1] 197.8397
```

18.2.10 峭 度(kurtosis)

4阶 中 心 距

$$m_4 = \frac{\sum(x-\bar{x})^4}{n}$$

2阶 中 心 距

$$m_2 = \frac{\sum(x-\bar{x})^2}{n}$$

峭 度

$$g2 = \frac{m^4}{m2^2} - 3$$

编 写 函 数(峭 度):

```
kurtosis<-function(x){
   a=mean(x)
   n=length(x)
   m4=sum((x-a)^4)/n
   m2=sum((x-a)^2)/n
   kurt=m4/m2^2 -3
   kurt
```

```
}
# 计 算 结 果
> kurtosis(x)
[1] 0.6260693
```

18.2.11 变 异 系 数(coefficient of variability)

公 式

$$CV = \frac{sd(x)}{\bar{x}}$$

编 写 函 数(变 异 系 数)

```
CV<-function(x){
   sd(x)/mean(x)
}
> CV(x)
[1] 1.070169
```

18.2.12 异 常(极 端)值

异 常 值:

```
x >上 百 分 位 数+1.5×（ 上 百 分 位 数-下 百 分 位 数）
x <下 百 分 位 数-1.5×（ 上 百 分 位 数-下 百 分 位 数）
```

极 端 异 常 值:

```
x >上 百 分 位 数+3×（ 上 百 分 位 数-下 百 分 位 数）
x <下 百 分 位 数-3×（ 上 百 分 位 数-下 百 分 位 数）
```

```
# 分位数
> q=quantile(x,c(.25,.75)); q
    25%     75%
1.000000 7.389056

# 异常值下侧界限, 故x没有下侧异常值
> out.low=q[1]-1.5*(q[2]-q[1]);out.low
     25%
-8.583584

# 异常值上侧界限, x有上侧异常值
> out.upper=q[1]+1.5*(q[2]-q[1]);out.upper
    25%
10.58358

# 绘图来查看, 可以看到x的上侧异常值
> boxplot(x)
```

18.3 离散数据(Categorical data)

18.3.1 列表:table()

table 可以作用于单个因子, 及多个因子. 2因子的会产生2维频数分布表, 相应k因子会产生k维频数分布表.

```
> x
[1] 1 1 2 0 2 0 0 1 1 0
> y=sample(c('y','n'),10,replace=TRUE)
> y
[1] "n" "y" "y" "y" "y" "y" "n" "y" "n" "y"

> table(x)
x
0 1 2
4 4 2
> table(y)
```

```
y
n y
3 7
> table(x,y)
y
x  n y
0 1 3
1 2 2
2 0 2

> x=c("Yes","No","No","Yes","Yes")
> table(x)
x
No Yes
2  3
> y=1:9
> table(y)
y
1 2 3 4 5 6 7 8 9
1 1 1 1 1 1 1 1 1
```

18.3.2 factor()函数

```
> factor(x)
[1] Yes No  No  Yes Yes
Levels: No Yes
> factor(y)
[1] 1 2 3 4 5 6 7 8 9
Levels: 1 2 3 4 5 6 7 8 9
> table(x)/length(x)
x
No Yes
0.4 0.6
```

18.3.3 gl()函数

gl() 函数可以方便的产生因子, 一般用法为

```
gl(n, k, length = n*k, labels = 1:n, ordered = FALSE)
```

```
> gl(3,5)
 [1] 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3
Levels: 1 2 3

> gl(3,1,15)
 [1] 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3
Levels: 1 2 3
```

18.3.4 条形图, 饼图

绘制因子频率(factor)

```
> b = scan()
1:  3 4 1 1 3 4 3 3 1 3 2 1 2 1 2 3 2 3 1 1 1 1 4 3 1
26:
Read 25 items
> barplot(b) # not correct
> barplot(table(b)) # right
> barplot(table(b)/length(b))
> b.count=table(b) # 存于一个变量中
> pie(b.count)
> names(b.count)=c("a","b","c") # 命名
> pie(b.count)
> pie(b.count,col=c("purple","green","cyan","white")) # 改变颜
色
```

18.3.5 折线图

```
# 好象需要强制转换一下
> x=as.numeric(t)
> lines(x)
```

18.4 连续数据(numerical data)

```
> s = scan() # 工资
1: 12 .4 5 2 50 8 3 1 4 0.25
11:
Read 10 items
> s
[1] 12.00  0.40  5.00  2.00 50.00  8.00  3.00  1.00  4.00  0.25
```

18.4.1 fivenum

最小, 0.25 , 0.5 0.75, 最大的 5 个数

```
> fivenum(s) # min, lower hinge, Median, upper hinge, max
[1]  0.25  1.00  3.50  8.00 50.00
```

18.4.2 summary

```
> summary(s)
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.250   1.250   3.500   8.565   7.250  50.000
```

18.4.3 分位数

最小的值为 0, 最大的为 1, %u 为 min+(max-min)*u

```
> quantile(s) # 分位数
0%   25%   50%   75%  100%
0.25  1.25  3.50  7.25 50.00
> quantile(s,.25) # 分位数
25%
1.25
> quantile(s,c(.25,.75))
25%  75%
1.25 7.25
> sort(s)
[1]  0.25  0.40  1.00  2.00  3.00  4.00  5.00  8.00 12.00 50.00
```

18.4.4 条件性测量

```
> mean(s,trim=1/10)
[1] 4.425
> mean(s,trim=2/10)
[1] 3.833333
> IQR(s) #  interquartile range is the difference of the 3rd and 1st quartile.
[1] 6
```

18.4.5 茎叶图

```
> stem(s)

The decimal point is 1 digit(s) to the right of the |

0 | 00123458
1 | 2
2 |
```

```
3 |
4 |
5 | 0
```

18.4.6 直方图

```
> x=scan()
1: 29.6 28.2 19.6 13.7 13.0 7.8 3.4 2.0 1.9 1.0 0.7 0.4 0.4 0.3
15: 0.3 0.3 0.3 0.3 0.2 0.2 0.2 0.1 0.1 0.1 0.1 0.1
27:
Read 26 items
> a=hist(x) # 频率
> hist(x,probability=TRUE) # 密度

# 获得额外信息--频数、频率、组值、组限、中值等
> str(a)
List of 7
$ breaks    : num [1:13] -3 -2.5 -2 -1.5 -1 -0.5 0 0.5 1 1.5 ...
$ counts    : int [1:12] 0 1 1 1 3 4 6 2 2 0 ...
$ intensities: num [1:12] 0.0 0.1 0.1 0.1 0.3 ...
$ density   : num [1:12] 0.0 0.1 0.1 0.1 0.3 ...
$ mids      : num [1:12] -2.75 -2.25 -1.75 -1.25 -0.75 -0.25 0.25 0.75 1.25 1.75 ..
$ xname     : chr "x"
$ equidist  : logi TRUE
- attr(*, "class")= chr "histogram"
```

18.4.7 盒形图

```
> boxplot(x)
```

18.4.8 折线图

```
# 添加折线图
> a=hist(x,breaks=seq(-3,3,by=0.5))
> lines( c(min(a$breaks),a$mids,max(a$breaks)),c(0,a$counts,0),type='l' )
```

18.4.9 区间分割–cut函数

把每个数据归属于某一类, 或某一区间

```
> sals = c(12, .4, 5, 2, 50, 8, 3, 1, 4, .25) # enter data
> cats = cut(sals,breaks=c(0,1,5,max(sals))) # specify the breaks
> cats
[1] (5,50] (0,1]  (1,5]  (1,5]  (5,50] (5,50] (1,5]  (0,1]  (1,5]  (0,1]
Levels: (0,1] (1,5] (5,50]
```

改变水平标签

```
> levels(cats) = c("a","b","c") #
> cats
[1] c a b b c c b a b a
Levels: a b c
> cats[1]
[1] c
Levels: a b c
> table(cats)
cats
a b c
3 4 3
```

绘图

```
> barplot(table(cats))
```

```
# 错误, must be numeric
> hist(cats)
```

18.5 几个例子

18.5.1 类型数据 vs. 类型数据

一个抽烟–学习时间的例子, Problem: 验证一项假定，抽烟的学生学习的时间少，抽样了10个人。

```
> x$smokes=c("Y","N","N","Y","N","Y","Y","Y","N","Y") # 抽烟与否
> x$study =  c(1,2,2,3,3,1,2,1,3,2) # 每天学习时间
> table(x)
   study
smokes 1 2 3
   N 0 2 2
   Y 3 2 1
> tmp = table(x)
> tmp
   study
smokes 1 2 3
   N 0 2 2
   Y 3 2 1
> str(tmp)
int [1:2, 1:3] 0 3 2 2 2 1
- attr(*, "dimnames")=List of 2
..$ smokes: chr [1:2] "N" "Y"
..$ study : chr [1:3] "1" "2" "3"
- attr(*, "class")= chr "table"
> old.digits = options("digits") # 保存默认打印字符长度 7
> options(digits=3)
```

prop.table 相当于

```
> tmp[1,1:3]/sum(tmp[1,1:3])
> tmp[2,1:3]/sum(tmp[2,1:3])
```

```
> prop.table(tmp,1) # 1 为按行，2 为列
   study
smokes    1    2    3
   N 0.000 0.500 0.500
   Y 0.500 0.333 0.167
> options(digits=7) # 还原打印字符位数

# 下面绘制条形图
> smokes=factor(smokes)
> smokes
[1] Y N N Y N Y Y Y N Y
Levels: N Y
> barplot(table(smokes,amount),
+ beside=TRUE,                # put beside not stacked
+ legend.text=T)   # add legend

# 只加图例
> barplot(table(amount,smokes),legend.text=T)
> barplot(table(smokes,amount), legend.text=T)

# 更改图例文字
> barplot(table(amount,smokes),main="table(amount,smokes)",
+ beside=TRUE,
+ legend.text=c("less than 5","5-10","more than 10"))
```

18.5.2 类型数据 vs. 连续数据

有实验组为x, 对照组为y, 画出盒型图来对照是一个不错的开始.(2 种方法)

```
> x = c(5, 5, 5, 13, 7, 11, 11, 9, 8, 9)
> y = c(11, 8, 4, 5, 9, 5, 10, 5, 4, 10)
> boxplot(x,y)

# 或者也可以这样
> num = scan()
1:  5 5 5 13 7 11 11 9 8 9 11 8 4 5 9 5 10 5 4 10
```

```
21:
Read 20 items
> cat = scan()
1: 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2
21:
Read 20 items
> boxplot(num~cat)
```

18.5.3 连续数据 vs. 连续数据

常用且比较简单. 最常用的是散点图(plot(x,y)).

Chapter 19

概率分布与统计函数表

参考 r cran task view: distribution 里有其它分布的函数与包的介绍, 包括多元正态分布, 多元t分布等.

19.1 R的统计函数表

在统计学中，产生随机数据是很有用的，R可以产生多种不同分布的随机数序列。这些分布函数的形式为rfunc(n,p1,p2,...)，其中func指概率分布函数，n为生成数据的个数，p1, p2, . . . 是分布的参数数值。大多数这种统计函数都有相似的形式，只需用d、p或者q去替代r，比如密度函数(dfunc(x,...))，累计概率密度函数（也即分布函数）(pfunc(x,...))和分位数函数(qfunc (p, ...)，0 ¡ p ¡ 1)。最后两个函数序列可以用来求统计假设检验中P值或临界值。

概率分布	R 对应的名字	附加参数
β分布	beta	shape1, shape2, ncp
二项式分布	binom	size, prob
Cauchy 分布	cauchy	location, scale
卡方分布	chisq	df, ncp
指数分布	exp	rate
F分布	f	df1, df1, ncp

γ 分布	gamma	shape, scale
几何分布	geom	prob
超几何分布	hyper	m, n, k
对数正态分布	lnorm	meanlog, sdlog
logistic 分布	logis	location, scale
负二项式分布	nbinom	size, prob
正态分布	norm	mean, sd
Poisson 分布	pois	lambda
t 分布	t	df, ncp
均匀分布	unif	min, max
Weibull 分布	weibull	shape, scale
Wilcoxon 分布	wilcox	m, n

19.2 各种分布的关系图

19.3 简单抽样

更复杂的抽样使用 MCMC.

重复和不重复的采样（放回和非放回的）

```
sample(x, size, replace = FALSE, prob = NULL)
> x <- 1:100
> sample(x,10)
[1] 96 60 86 43 30 81 26 24 94 28
> y <- 1:6  # 掷骰子
> sample(y,4,replace=TRUE)
[1] 5 1 5 1
```

19.3.1 放回式抽样

```
sample(x, size, replace = FALSE, prob = NULL)
```

```
> x=c('y','n')
> sample(x,10,replace=TRUE)
[1] "y" "n" "y" "n" "y" "n" "n" "y" "n" "n"
> y=c(1,2)
> sample(y,10,replace=TRUE)
[1] 1 1 2 1 1 1 1 1 1 2
```

19.3.2 非放回式抽样

```
> x=1:9
> sample(x,3)
[1] 4 3 1
# 只有一个参数时，相当于 shuffle
> sample(x)
[1] 6 8 4 7 2 1 5 3 9
# replace=TRUE 时，采样数目 = length(x)
> sample(x,replace=TRUE)
[1] 9 7 5 8 8 2 1 9 8
```

19.4 退化分布(单点分布)

当随机变量只取常数值时, 即

$$P(X(\omega) = c) \equiv 1$$

为退化分布, 其分布函数为

$$F(X) = \begin{cases} 0, & x < c \\ 1, & x >= c \end{cases} \tag{19.1}$$

或

$$F(X - c) = \begin{cases} 0, & x <= c \\ 1, & x > c \end{cases} \tag{19.2}$$

期望

$$E(X) = \sum_{\infty}^{\infty} xp(x) = c$$

方差

$$D(X) = E(X^2) - (E(X)^2 = c^2 - c^2 = 0$$

19.5 贝努里分布（Bernoulli distribution）

一次试验中只有两个结果$\Omega = \{A, \bar{A}\}$, 这种试验称为贝努里试验. 其中

$$P(A) = p, \qquad P(B) = 1 - p = q$$

记X为事件A出现的次数, 则

$$X = \begin{cases} 0, & X\text{不出现} \\ 1, & X\text{出现} \end{cases} \tag{19.3}$$

概率取值为

$$\begin{cases} P(X = 1) = q \\ P(X = 0) = p \end{cases} \tag{19.4}$$

那么我们有

$$P(X) = \begin{cases} p, & X = 1 \\ q, & X = 0 \\ 0. & X = \text{其它} \end{cases} \tag{19.5}$$

期望

$$E(X) = 0 * q + 1 * p = p$$

方差

$$D(X) = E(X^2) - (E(X)^2 = p - p^2 = pq$$

(考虑一下X的取值变为A出现为2, 否则为0, 期望和方差会是什么? [1])

下面考虑贝努里分布的母函数

$$g(z) = qz^0 + pz^1 = q + pz$$

那么期望和方差可以由下面得到

$$\begin{aligned} E(X) &= g'(1) = p \\ E(X^2) &= g''(1) + g'(1) = p \\ D(X) &= p - p^2 = pq \end{aligned}$$

产生200个贝努里分布的随机数, 其中-1出现的概率为0.2, 1出现的概率为0.8

```
> n <- 200
> x <- sample(c(-1,1), n, replace=T, prob=c(.2,.8))
> plot(cumsum(x),type='l')
```

19.6 二项分布

19.6.1 理论

在n重贝努里试验中, 记k为A出现的次数, 则k的取值为$0, 1, 2, \cdots, n$.

记A_i为第i次试验中出现事件A, $\overline{A_i}$为第i次试验中A不出现. 若记B_k为n重贝努里试验中, A出现k次这一事件, 则

$$B_k = (A_1 \cdots A_k \overline{A_{k+1}} \cdots \overline{A_n}) + (\cdots) + (\overline{A_1} \cdots \overline{A_{n-k}} A_{n-k+1} \cdots \overline{A_n})$$

右边一共有$\binom{n}{k}$项, 且两两互不相容. 由独立性得出

$$P(A_1 \cdots A_k \overline{A_{k+1}} \cdots \overline{A_n}) = P(A_1) \cdots P(A_k) P(\overline{A_{k+1}}) \cdots P(\overline{A_n})$$

[1] 答案为$E(X) = 2p, \quad E(X^2) = 4pq$

利用概率的加法定理得

$$P(B_k) = \binom{n}{k} p^k q^{n-k}$$

我们常常把此概率记为

$$B(k; n, p) = \binom{n}{k} p^k q^{n-k}$$

期望 [2]

$$E(X) = \sum_{k=0}^{n} P(B_k) = np$$

方差 [3]

$$D(X) = E(X^2) - (E(X)^2 = p - p^2 = npq$$

下面考虑二项分布的母函数

$$g(z) = \sum_{k=0}^{n} P(X = k) z^k = (pz + q)^n$$

也可以这样考虑, 记$X = X_1 + X_2 + \cdots + X_n$, 其中$X_i$为第i次贝努里试验. 由于$X_i$相互独立, 则二项分布的母函数可以由$g(z) = q + pz$的n次方给出

$$g(z) = (pz + q)^n = \sum_{k=0}^{n} \binom{n}{k} q^{n-k} p^k z^k$$

[2]小提示: 可以直接计算, 也可以使用独立随机变量的和的期望等于期望的和的性质来计算. 后者更简单一点. 还有一种有点技巧但容易公式化的方法. 母函数的方法最简单

[3]小提示: 同期望一样, 也可以使用几种不同的方法

由母函数的定义知, z^k的系数$\binom{n}{k}q^{n-k}p^k = P(X=k)$

那么期望和方差可以由下面得到

$$E(X) = g'(1) = np$$
$$E(X^2) = g''(1) + g'(1) = n^2p^2 - np^2 + np$$
$$D(X) = npq$$

19.6.2 产生二项分布随机数

rbinom(n, size, prob)

n为产生的随机数的个数(可以大于size), prob为单点分布(Bonulli分布)的成功的概率. size为二项分布的试验次数, 成功x的概率为:

```
p(x) = choose(n,x) p^x (1-p)^(n-x)
> rbinom(5,10,0.5)
[1] 7 4 7 5 5
> rbinom(5,10,0.1)
[1] 1 2 3 0 1

> dbinom(5,10,p=0.5)
[1] 0.2460938
```

当size取大于1的值时, 结果似乎会产生0,1,2,...,size的正态分布

```
> x=rbinom(10000,9,0.5)
> table(x)/length(x)
x
   0     1     2     3     4     5     6     7     8     9
0.0028 0.0203 0.0704 0.1605 0.2541 0.2401 0.1623 0.0701 0.0176 0.0018
> table(rbinom(10000,10,0.3))

0    1    2    3    4    5    6    7    8
261 1272 2327 2655 1987 1033  355   91   19
```

设100次试验，A发生的概率为0.3,A发生20次的概率为：

```
> dbinom(20, 100, 0.3)
[1] 0.007575645
```

A发生20 <=<= 60次的概率为：

```
> sum(dbinom(20:60, 100, 0.3))
[1] 0.9911128
```

其他

```
> dbinom(1,2,0.5)
[1] 0.5
> dbinom(0,2,0.5)
[1] 0.25
> dbinom(2,2,0.5)
[1] 0.25
```

19.6.3 累积概率密度函数及图

```
> pbinom(60,100,0.5)-pbinom(39,100,0.5)
[1] 0.9647998
> pbinom(6,10,0.5)-pbinom(3,10,0.5)
[1] 0.65625
# 也可以使用下面
> sum(dbinom(40:60, 100, 0.5))
[1] 0.9647998
> sum(dbinom(4:6, 10, 0.5))
[1] 0.65625
```

最后画出密度和累积密度的图

```
> plot(dbinom(0:100,100,0.5))
> plot(pbinom(0:100,100,0.5))
```

19.6.4 指定累积概率的q值

求成功概率为0.2, 总次数为10, 指定累积概率为0.5的试验次数为

```
> qbinom(p=0.5,size=10,prob=0.2)
[1] 2
# 检验
> pbinom(q=2,size=10,prob=0.2)
[1] 0.6777995
> pbinom(q=1,size=10,prob=0.2)
[1] 0.3758096
> pbinom(q=3,size=10,prob=0.2)
[1] 0.8791261
```

19.7 几何分布

19.7.1 性质

在n重贝努里试验中, 设A的第一次出现是在第k次试验, 记此事件为W_k, 则

$$W_k = \overline{A_1 A_2} \cdots \overline{A_{k-1}} A_k$$

$$P(W_k) = P(\overline{A_1})P(\overline{A_2}) \cdots P(\overline{A_{k-1}} A_k) = q^{k-1}p$$

记

$$g(k;p) = q^{k-1}p, \quad k = 0, 1, 2, \cdots$$

$g(k;p)$是几何级数的一般项, 因此上式称为几何分布.

验证

$$\sum_{k=1}^{\infty} g(k;p) = \frac{1}{1-q}p = 1$$

期望[4]

$$E(X) = \sum_{k=1}^{\infty} kg(k;p) = \frac{1}{p}$$

而

$$E(X^2) = \sum_{k=1}^{\infty} k^2 g(k;p) = \frac{1+q}{p^2}$$

则方差

$$D(X) = \frac{q}{p^2}$$

母函数

$$g(z) = \sum_{k=0}^{n} P(X=k)z^k = \frac{pz}{1-qz}$$

期望

$$E(X) = g'(1) = \frac{1}{p}$$

[4]虽然有一点点复杂, 但是鼓励你尝试一下

$$g''(1) = \frac{2q}{p^2}$$

方差

$$D(X) = \frac{q}{p^2}$$

偏度

$$\gamma_1 = (2-p)/(1-p)^{1/2}$$

峰度

$$\gamma_2 = (p^2 - 6p + 6)/(1-p)$$

19.7.2 无记忆性

无记忆性: 假设在前m次贝努里试验中没有出现事件A, 那么在此后的贝努里试验中, 事件A首次出现的概率仍然服从几何分布, 与前面的试验次数m无关.

19.7.3 指数分布近似

如果实验次数足够大, 即 p 足够小, 则几何分布可以近似为指数分布

$$g(k;p) = \frac{1}{N} e^{-\frac{k-1}{N}} \approx p * e^{-p(k-1)}$$

19.8 负二项分布(巴斯卡分布)

负二项分布也称为巴斯卡分布(Pascal). 考虑重复独立的贝努里试验. 在第r次试验中事件A出现第k次, 则随机变量r服从负二项分布. $k = 1$的负二项分布即是几何分布.

19.8.1 性质

分布概率

$$P_k(r,p) = \binom{r-1}{k-1} p^k (1-p)^{r-k}$$

均值

$$E(r) = k/p$$

方差

$$V(r) = k(1-p)/p^2$$

偏度

$$\gamma_1 = (2-p)/\sqrt{k(1-p)}$$

峰度

$$\gamma_2 = (p^2 - 6^p + 6)/k(1-p)$$

概率母函数

$$G(Z) = [\frac{pZ}{1-(1-p)Z}]^k$$

19.8.2 推导

接着几何分布考虑, 若$T_1, T_2, \cdots, T_n$每个以几何分布的母函数为母函数(回忆一下母函数与分布函数互相唯一确定), 也就是每个都是几何分布(等待第一次成功的次数的随机变量).

记$S_n = T_1 + T_2 + \cdots + T_n$, 则$S_n$为第n次成功的等待时间(1次算一个单位时间的话).

我们先来推导两个式子.

第一个式子

$$\frac{1}{1-x} = 1 + x + x^2 + \cdots$$

$$(\frac{1}{1-x})' = \frac{1}{(1-x)^2} = 1 + 2x + 3x^2 + \cdots$$

$$(\frac{1}{1-x})'' = \frac{2!}{(1-x)^3} = 2! + 32x + 43x^2 + \cdots$$

$$(\frac{1}{1-x})^{(n)} = \frac{(n-1)!}{(1-x)^n} = (n-1)! + \frac{n!}{1!}x + \frac{(n+1)!}{2!}x^2 + \cdots + \frac{(n+j-1)!}{j!}x^j + \cdots$$

两边同除以$(n-1)!$, 由归纳法得

$$\frac{1}{(1-x)^n} = \sum_{j=0}^{\infty} \binom{n-1+j}{j} x^j$$

第二个式子–负二项分布(牛顿二项分布的推广)

$$\begin{aligned}
\binom{-n}{j} &= \tfrac{n(n+1)\cdots(n+j-1)}{j!}(-1)^j \\
&= \tfrac{(n-1+j)!}{j!(n-1)!}(-1)^j \\
&= \tbinom{n-1+j}{j}(-1)^j \\
&= \tbinom{n-1+j}{n-1}(-1)^j
\end{aligned}$$

由这两个式子得

$$\frac{1}{(1-x)^n}=\sum_{j=0}^{\infty}\binom{-n}{j}(-1)^j x^j$$

下面我们来看S_n, 由于T_i相互独立, 则S_n的母函数由下式给出(把上式代入)

$$g(z)^n=(\frac{pz}{1-qz})^n=(pz)^n\sum_{j=0}^{\infty}\binom{-n}{j}(-1)^j(qz)^j=\sum_{j=0}^{\infty}\binom{n+j-1}{n-1}p^nq^jz^{n+j}$$

设$k=n+j$, 则

$$g(z)^n=\sum_{k=n}^{\infty}\binom{k-1}{n-1}p^nq^{k-n}z^k$$

根据母函数的定义, 第n次成功出现在第$n+j$次试验的概率为

$$P(S_n=n+j)=\binom{n+j-1}{n-1}p^nq^j$$

下面的等式也是成立的

$$P(S_n=n+j)=\binom{n+j-1}{j}p^nq^j=\binom{-n}{j}p^n(-q)^j$$

由上式给出的分布叫做负二项分布.

我们再来看

$$\begin{aligned}\frac{g(z)}{z} &= \sum_{j=1}^{\infty}\frac{q^{j-1}pz^j}{z}\\ &= \sum_{k=0}^{\infty}q^kpz^k\end{aligned}$$

观察z^k的系数为T_i-1即第一次成功前失败的次数. 那么

$$\begin{aligned}(\frac{g(z)}{z})^n &= (\frac{p}{1-qz})^n\\ &= \sum_{k=0}^{\infty}\binom{n+k-1}{k}p^n(qz)^k\end{aligned}$$

就是S_n-n的母函数, 即第n次成功前失败的次数.

另外可以这样考虑, 若第n次成功发生在第$n+j$次试验, 当且仅当$n+j-1$次试验中成功$n-1$次, 失败j次, 且第$n+j$次成功, 故有

$$\begin{aligned}P(S_n=n+j) &= \binom{n+j-1}{n-1}p^{n-1}q^jp=\binom{n+j-1}{j}\\ p^nq^j &= \binom{-n}{j}p^n(-q)^j\end{aligned}$$

19.9 超几何分布(Hypergeometric distribution)及其推广

19.9.1 超几何分布

N个元素, 其中a个元素为成功, 其余为失败. 做不放回n次抽样. 这n次抽样中包含r次成功(相应$n-r$次失败)的概率称为超几何分布.

分布概率

$$P(r,N,n,a) = \binom{N-a}{n-r}\binom{a}{r}/\binom{N}{n}, \qquad r = 0,1,2,\cdots,min(a,n)$$

均值

$$E(r) = \frac{na}{N}$$

方差

$$V(r) = \frac{N-n}{N-1}\frac{na}{N}(1-\frac{a}{N})$$

```
# 参数的意义
# rhyper(nn, m, n, k)
# m: 白球的数目. n: 黑球的数目. k: 抽出球的数目. nn: 观察的次数.
> rhyper(10,15,5,5)
[1] 3 3 3 3 5 2 5 4 3 4
```

下面是一个例子，从13000中抽一组基因，934个，再抽一组1000个，然后他们的交集是130，这是否正常？

此模型可以看做13000个球中黑球934个，其余为白球。从中抽取1000个球，得到了黑球130个，我们希望知道这是否正常？

这是一个超几何分布。

我们应该考虑抽取1000个球得到的球的数目>= 130的概率，如果这个概率< 0.05，我们就认为它不正常，否则它就是正常的。

这个概率为

```
> 1-phyper(130, 934, 13000-934, 1000)
```

```
[1] 3.871348e-12
```

```
3.871348e-12 <<0.05
```

所以我们说出现130个黑球及其以上的概率是很小的，也就是说，这是不正常的。

还可以绘制一下抽取1000个球得到不同数目黑球的概率（大部分在30-120之间）

```
plot(dhyper(30:120, 934, 13000-934, 1000),t='l')
```

19.9.2 推广的超几何分布

设N个元素可以分为k种事件A_i, 属于事件A_i的事件个数有a_i个. 对N个元素做n次不放回抽样, 事件A_i出现的次数为随机变量.它服从推广的超几何分布.

$$P(r;N,n;a)=\prod_{i=1}^{k}\binom{a_i}{r_i}/\binom{N}{n},\qquad r_i=0,1,2,\cdots,min(a_i,n)$$

其中r,a为向量.

$$\sum_{i=1}^{k}r_i=n,\quad \sum_{i=1}^{k}a_i=N,$$

当$n<<N$, 推广的超几何分布近似于$p_i=\frac{a_i}{N}$的多项分布.

下面是一个例子. 设10个人的血型为O型3个, A型4个, B型3个. 随机轴5人, 问得到O型1人, A,B型各2人的概率.

本例中, $N=10,n=5,r_1=1,r_2=2,r_3=2,a_1=3,a_2=4,a_3=3$. 代入上面的公式中有

$$P(1,2,2;10,5;3,4,3)=\frac{\binom{3}{1}\binom{4}{2}\binom{3}{2}}{\binom{10}{5}}=3/14$$

19.10 泊松分布

19.10.1 产生泊松分布随机数

rpois(n, lambda) n 为要产生随机数的个数, lambda 为 poisson 的参数.

19.10.2 期望和方差

具有参数 lambda 的泊松分布的均值和期望均为 lambda.

19.10.3 密度-累积概率密度函数

```
ppois(q, lambda, lower.tail = TRUE, log.p = FALSE)
dpois(x, lambda, log = FALSE)
```

```
> x=1:10
> dpois(x,3)
 [1] 0.1493612051 0.2240418077 0.2240418077 0.1680313557 0.1008188134
 [6] 0.0504094067 0.0216040315 0.0081015118 0.0027005039 0.0008101512
> ppois(x,3)
 [1] 0.1991483 0.4231901 0.6472319 0.8152632 0.9160821 0.9664915 0.9880955
 [8] 0.9961970 0.9988975 0.9997077
```

19.10.4 指定累积概率的q值

```
qpois(p, lambda, lower.tail = TRUE, log
.p = FALSE)
```

```
> x=seq(0,1,0.1)
> x
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
```

```
> qpois(x,3)
 [1]   0   1   2   2   2   3   3   4   4   5 Inf
```

19.11 均匀分布

密度函数

$$p(x) = \begin{cases} 1/(b-a) & a <= x <= b \\ 0 & x < a \quad or \quad x > b \end{cases}$$

分布函数

$$F(x) = \int_{-\infty}^{x} p(y)dy = \begin{cases} 0 & x <= a \\ (x-a)/(b-a) & a < x <= b \\ 1 & x > b \end{cases}$$

其它[5]

$$E(X) = \int_{-\infty}^{\infty} xp(x)dx = \frac{a+b}{2}$$

$$E(X^2) = \int_{-\infty}^{\infty} x^2 p(x)dx = \frac{a^2+ab+b^2}{3}$$

$$D(X) = \frac{(b-a)^2}{12}$$

使用 sample 模拟

[5]在几乎任何概率论教科书上都可以找到推导, 并且它们很简单

```
> sample(1:10, 20, replace=T)
[1]  7 10 10  4  6  8  6  6  4  8  1  3  9 10  9  8  3  4 10 10
```

19.12 指数分布

19.12.1 定义

符合下述密度函数

$$p(x) = \begin{cases} ae^{-ax} & x >= \\ 0 & x < 0 \end{cases}$$

和累积分布函数

$$F(x) = \begin{cases} 1 - e^{-ax} & x >= \\ 0 & x < 0 \end{cases}$$

的分布称为指数分布.

均值

$$E(X) = \frac{1}{a}$$

方差

$$V(X) = \frac{1}{a^2}$$

偏度

$$\gamma_1 = 2$$

峰度

$$\gamma_2 = 6$$

k阶 中 心 矩

$$\mu_k = E[x - E(X)]^k = \frac{k!}{a^k}\sum_{i=0}^{k}(-1)^{k-i}\frac{1}{(k-i)!}$$

特 征 函 数

$$\varphi(t) = (1 - \frac{it}{a})^{-i}$$

19.12.2 无 记 忆 性

指 数 分 布 有 类 似 几 何 分 布 的"无 记 忆 性", 即

$$p(x > s + t|x > s) = \frac{p(x > s + t)}{p(x > s)} = \frac{e^{-a(s+t)}}{e^{-as}} = e^{-at} = p(x > t)$$

指 数 分 布 是 唯 一 具 有 此 性 质 的 连 续 分 布.

(可 以 这 样 理 解, 已 知 寿 命 长 于s年, 则 再 活t年 的 概 率 与 年 龄s无 关.)

19.12.3 与 泊 松 分 布 的 关 系

记X(t)为 参 数at的 泊 松 分 布(过 程), 则

$$p(X(t) = k) = \frac{e^{-at}(at)^k}{k!}$$

当k=0时

$$p(X(t) = 0) = e^{-at} \sim \text{指 数 分 布}$$

19.13 伽马分布(Gamma distribution)

19.13.1 特征

随机变量X的密度函数为

$$f(x;\alpha,\beta)=\frac{\beta^{\alpha}}{\Gamma(\alpha)}x^{\alpha-1}e^{-\beta x}$$

其中α,β为正常数, 称X服从参数为α,β的伽马分布.

期望

$$E(X)=\alpha/\beta$$

方差

$$V(X)=\alpha/\beta^2$$

偏度

$$\gamma_1=\frac{2}{sqrt\alpha}$$

峰度

$$\gamma_2=\frac{6}{\alpha}$$

特征函数

$$\varphi(t)=(1-\frac{it}{\beta})^{-\alpha}$$

当$\alpha <= 1$,函数单调下降, $\alpha > 1$,概率密度为单峰函数, 极大值在$x=(\alpha-1)/\beta$处.

19.13.2 Gamma 函数

$\Gamma(\alpha)$的表达式为

$$\Gamma(\alpha) = \int_0^\infty y^{\alpha-1}e^{-y}dy, \quad \alpha > 0$$

有以下性质

$$\Gamma(1) = 1, \qquad \Gamma(1/2) = \sqrt{\pi}$$
$$\Gamma(\alpha) = (\alpha-1)\Gamma(\alpha-1)$$
$$\Gamma(n) = (n-1)!, \qquad n\text{为正整数}$$
$$\Gamma(n+1/2) = \frac{(2n-1)!!}{2^n}\sqrt{\pi}$$

```
#␣小技巧，应用gamma函数与阶乘的关系,␣lgamma()为gamma函数的对数
>␣choose(20,10)
[1]␣184756
>␣exp(lgamma(21)-lgamma(11)-lgamma(11))␣#␣此式相当于choose(20,10)
[1]␣184756
␣␣␣␣
```

与beta函数的关系

$$B(p,q) = \frac{\gamma(p)\gamma(q)}{\gamma(p+q)} \tag{19.6}$$

19.13.3 与指数分布,卡方分布,厄兰分布的关系

当$\alpha = 1$, 伽马分布变为指数分布.

参数$\alpha/2, \beta 1/2$, 其中v为正整数的伽马分布即为自由度为v的卡方χ^2分布.

α为正整数的伽马分布称为厄兰分布(Erlangian distributions). 概率密度为

$$f(t;k,\lambda)=\frac{\lambda^k}{(k-1)!}t^{k-1}e^{-\lambda t},\qquad \lambda>0,\ k=1,2,\cdots,\ 0<=t<\infty$$

容易得到厄兰分布的均值和方差为

$$E(t)=k\lambda^{-1},\qquad V(t)=k\lambda^{-2}$$

厄兰分布可以从泊松分布推导出来. 因而可以描述泊松随机过程.

19.13.4 厄兰分布的推导

若X(t)是服从参数为at的泊松分布(过程). 记τ_r为第r个跳跃发生的时刻(第r个例子到来的时刻). 则

$$\{\tau_r<t\}\Longleftrightarrow\{X(t)>=r\}$$

即第r个跳跃发生在时刻t之前, 也就是t时刻之前发生至少r次跳跃. 我们以F(x)记τ_r的分布函数, 则有

$$F(t)=p(\tau_r<t)=p(X(t)>=r)=1-\sum_{k=0}^{r-1}\frac{(at)^ke^{-at}}{k!}$$

那么[6]

$$p(t)=F'(t)=\frac{a^rt^{r-1}e^{-at}}{(r-1)!}=\frac{a^rt^{r-1}e^{-at}}{\Gamma(r)}$$

[6]中间的推导只有一点点的烦琐. 鼓励大家推导出来以增加信心

称

$$p(x) = \begin{cases} \frac{a^r x^{r-1} e^{-ax}}{\Gamma(r)} & x > 0 \\ 0 & x <= 0 \end{cases}$$

的分布为Γ-分布. 其中a¿0, r¿0为参数.

泊松过程的第r个跳跃发生的时刻服从Γ-分布.

r=1时, Γ-分布变为指数分布

r=正整数时, Γ-分布为r个服从指数分布的随机变量之和的分布, 与负二项分布类似.

19.13.5 一些物理现象与Gamma分布的关系

来自 http://www.moon-soft.com/program/bbs/readelite80182.htm

参考改造后的熵——张学文一文 (http://www.moon-soft.com/program/bbs/docelite80144.htm)

张学文 中国气象局,乌鲁木齐沙漠气象研究所

本讲对上次介绍的Gamma 分布再做一些说明。1. Gamma分布兼有指数分布和幂分布的特点。从Gamma分布公式看，当b为零时，它就变成了前面介绍过的幂分布。当n=1时，它变成了前面介绍的指数分布。而它的分布函数可以视为前两种分布的乘积。幂分布与指数分布在变量值很小时其概率值很大，但是它们组成的Gamma的最大值却不在变量最小时而是有一个峰值比较居中。这可能是我们主观所没有料到的。2. 利用概率知识，我们还可以就一个服从指数分布的变量的n个合计值的概率分布问题做研究，而且可以得到这个合计值应当服从Gamma分布。其中的n也就是Ga mma分布中的n。这从另外一个侧面说明：变量的代数平均值固定、变量的出现概率对应的复杂程度（熵）最大，那么n个变量的合计值应当服从Gamma分布。这也是形成Gamma分布的物理原因的另外一种思考途径（它代替了几何平均值固定的约束条件，证明这个结论要对一种卷积积分，这里不谈了）。3. 一个地方

的一次降雨的雨量是很不规则的，于是雨量分布对应的复杂程度应当最大化。我们可以想到的约束就是它的代数平均值应当固定（对应与当地气候在一定时段内不变化）。利用这两点我们前面就推出其雨量的概率分布是指数分布（见斩乱麻问题）。根据上一段的介绍，两场雨的合计值的概率就应当是n=2的G amma分布。三场雨的合计值是n=3的Gamma分布。如果一个月大约有4场雨，那么其月雨量的概率分布就应当与n=4的Gamma分布很接近。而气象上的统计也确实证明的雨量比较多的地方，其月雨量服从Gamma分布。有了我们这些认识对气象要素的概率分布的理解就深了一个层次。4. 水文上也对河水流量等广泛使用皮尔逊Ⅲ型分布，即Gamma分布。为什么这种分布符合水文实际？在气候不变时（一个长时期），流量的代数平均值不变是个合理的假设。变量的几何平均值不变对应的是变量的相对变化的平均值固定，这符合水文现象的特点。于是对于水文现象，它满足代数平均值和几何平均值分别固定的假设是合理的。再加上复杂程度最大化（用最复杂原理），就如上一讲那样，我们自然得到了一个服从Gamma分布的结论。5. 简而言之，一个广义集合（如一批水文观测数据），如其代数平均值和几何平均值应当是受约束的（有固定值的），当承认其复杂程度应当最大，其分布函数就应当是服从Gamma分布。当我们证实一批资料符合Gamma分布时我们高兴，当我们用最复杂原理配合代数平均值几何平均值合理地说明了它也就是应当服从Gamm a分布时，我们的工作就从现象（经验方程。唯象方程）向理论深入一步。6. 现在的统计书介绍Gamma分布的也不少，但是利用熵原理（最复杂原理）说明它的物理背景的几乎没有（也许我见识少）。我认为应当把这一层认识介绍给统计教科书的作者。

19.14 Beta分布

来自 http://hi.baidu.com/msingle/blog/item/36bb24df31f2bb1b4954033b.html

概率论中还有一种称为贝塔（β，beta）分布的概率密度分

布函数。它的数学形式是

$$f(x) = \frac{1}{B(p,q)} x^{p-1}(1-x)^{q-1} \qquad 0 < x < 1, p > 0, q > 0 \tag{19.7}$$

这里的变量x仅能出现于0到1之间，p，q是两个大于0的参数。B(p,q) 的含义是

$$B(p,q) = \int_0^1 x^{p-1}(1-x)^{q-1} dx \tag{19.8}$$

它与Γ函数，有如下关系

$$B(p,q) = \frac{\gamma(p)\gamma(q)}{\gamma(p+q)} \tag{19.9}$$

而我们介绍过的阶乘符号！与Γ的关系是

n!= Γ(n+1)

所以贝塔分布也可以写为

$$f(x) = \frac{(m+n+1)!}{m!n!} x^m (1-x)^n \qquad 0 < x < 1, p > 0, q > 0 \tag{19.10}$$

beta分布的均值为$m = a/(a+b)$，方差为$v = m(1-m)/(a+b+1)$。对于一般用户，可能估计a，b比较困难。但是，一般用户会给出两个百分位点的估计，一个为50%的p值小于0.3, 即p的中位数为0.3, 90%的p值会小于0.5. 即p的90百分位点为0.5.那么使用下面的函数估计参数a，b（LearnBayes包）

```
quantile2=list(p=.9,x=.5)
quantile1=list(p=.5,x=.3)
beta.select(quantile1,quantile2)

>␣␣beta.select(quantile1,quantile2)
[1]␣3.26␣7.19
␣␣␣␣
```

beta.select()函数使用迭代的方法逼进给定两个百分位点的beta参数a，b。

现在考虑从最复杂原理加适当的约束条件推求这个概率密度分布函数的问题。根据过去的经验，容易看出它可能是下面两个约束条件与最复杂原理的应用结果。

变量x的对数的平均值为固定值（等价于几何平均值为常数）：

$$u = \int_0^1 (\ln x) f(x) dx \tag{19.11}$$

（1-x）的对数的平均值也是固定之值：

$$v = \int_0^1 [\ln(1-x)] f(x) dx \tag{19.12}$$

作为概率密度，当然还有

$$1 = \int_0^1 f(x) dx \tag{19.13}$$

根据上面的三个约束公式和最复杂原理，利用拉哥朗日方法，构造的F函数是

$$F = \int_0^1 -f \ln f dx + C_1[(\int_0^1 f dx) - 1] + C_2[(\int_0^1 \ln x f dx) - u] C_3[(\int_0^1 \ln(1-x) f dx - v] \tag{19.14}$$

求F对未知的概率密度f的偏微商，并且令它等于0（利用了最复杂原理），我们得到

$$f(x) = [exp(C_1 - 1)]x^{C_2}(1-x)^{C_3} \tag{19.15}$$

利用分布函数的积分应当等于1的约束和积分知识我们得到

$$exp(C_1 - 1) = \frac{1}{B(C_2 + 1, C_3 + 1)} \tag{19.16}$$

所以分布函数可以写为

$$f(x) = \frac{1}{B(C_2 + 1, C_3 + 1)}x^{C_2}(1-x)^{C_3} \tag{19.17}$$

显然，这个公式的外型已经与贝塔分布一致了。余下的问题是利用关于u，v的约束公式可以求出C2，C3。使这个公式通过u，v来表示。由于u，v与C2，C3的关系比较复杂，我们没有得到具体的关系式。但是概率密度分布函数的形状与概率论中的贝塔分布一致就已经达到了我们的目的：界于0-1之间的变量的两种几何平均值固定和最复杂原理相结合可能是一些贝塔分布形成的原因。

贝塔分布中的变量x的变化范围仅能在0到1之间，而且（1-x）与x有对称性，这是重要的特点。图18.5给出了p=3,q=6时的贝塔分布函数的形状。

图18.5贝塔分布的曲线形状

空气中含有的气体状态的水分。表示这种水分的一种办法就是相对湿度。即现在的含水量与空气的最大含水量（饱和含水量）的比值。我们听到的天气预告用语中就经常使用相对湿度这个名词。

相对湿度的值显然仅能出现于0到1之间（经常用百分比表示）。而空气为什么出现某个相对湿度显然具有随机性（可以利用最复杂原理），这些提示我们空气的相对湿度可能符合贝塔分布。

马淑红等人完成的“塔里木气候极值及其在油田工程设计中的应用”研究中[13]（同名的书由气象出版社于1995年出版见138-142页），刘绍民等人分析了冬季塔里木盆地的日最大相对湿度和夏季日最小相对湿度。证实它们都符合贝塔分布。

19.15 正态分布

推导泊松分布的时候总是感觉有点不太正常(还记得泊松分布的条件吗?), 而且还有计算二项分布值的广泛需要.例如: n=100, p=0.5, k=50时$B(k;n,p)$的值到底是多少?

下面我们将一步一步推导出正态分布的表达式(如果时间允许的话)

19.15.1 Stirling 公式

Stirling 公式为阶乘的近似计算公式 [7]

$$\chi(n) = (e/n)^n\sqrt{2\pi n}e^{\omega(n)} = n! \qquad (\ 1/(12(n+1/2)) < \omega < 1/12n\ )$$

19.15.2 从二项分布到正态分布

- 首先推导当$n \to \infty$时二项系数的值趋于0
- 其次证明当$n \to \infty$时, 对于固定的区间, 二项分布的概率值之和为0
- 再次设$0 < p < 1$, $q = 1 - p$, 且

$$x = \frac{k - np}{\sqrt{npq}} \qquad 0 <= k <= n$$

[7]推导见《数学分析原理》第二卷第一分册 52页. 作者: 格.马.菲赫金哥尔茨. 译者: 丁寿田

设A是一个任意而固定的正常数. 于是在满足$|x| <= A$的k的范围内, 我们有

$$\binom{n}{k} p^k q^{n-k} \approx \frac{1}{\sqrt{2\pi npq} e^{-x^2/2}}$$

且收敛是一致的.

- 再次 (棣莫佛-拉普拉斯定理)对任意两个常数a和b, 我们有

$$\lim_{n\to\infty} P(a < \frac{S_n - np}{\sqrt{npq}} <= b) = \frac{1}{\sqrt{2\pi}} \int_a^b e^{-x^2/2} dx$$

19.15.3 定义

以下面的函数

$$F(x) = \frac{1}{\sqrt{2\pi}} \int_a^b e^{-x^2/2} dx$$

作为分布函数的概率分布称做标准正态分布. 概率密度函数显然就为

$$p(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$

正态分布函数为

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_a^b e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} dx$$

的概率分布称为正态分布。其中σ为方差，μ为平均值。

下面来验证一下[8]

$$\int_{-\infty}^{\infty} p(x) dx = 1$$

[8]在菲赫金哥尔茨的《数学分析原理》中至少提供了4中方法来得到这个非正常积分的结果

设随机变量

$$X_j \sim N(\mu_j, \sigma_j^2) \quad j = 1, 2, \cdots, n$$

其中μ_i为均值, σ_j^2为方差. 则

$$X_1 + X_2 + \cdots + X_n \sim N(\sum_{j=1}^{n} \mu_j, \sum_{j=1}^{n} \sigma_j^2)$$

19.15.4 转换非标准正态分布到标准正态分布

具有均值为μ标准差为σ的正态分布变量 x, 可以使用下面的公式变换为标准正态分布

$$Z = \frac{x - \mu}{\sigma}$$

19.15.5 例子

```
# 产生正态分布随机数
rnorm(n, mean=0, sd=1)

> rnorm(10,0,1)
 [1]  0.9944192 -0.1384374 -0.8876501  1.0416947 -0.3217919 -0.8546145
 [7] -2.0329649 -0.5276146  0.1380986 -0.8563042


# 密度-累积概率密度函数
dnorm(x, mean=0, sd=1, log = FALSE)
pnorm(q, mean=0, sd=1, lower.tail = TRUE, log.p = FALSE)
```

```
> dnorm(x)
 [1] 0.2419707 0.2660852 0.2896916 0.3122539 0.3332246 0.3520653 0.3682701
 [8] 0.3813878 0.3910427 0.3969525 0.3989423 0.3969525 0.3910427 0.3813878
[15] 0.3682701 0.3520653 0.3332246 0.3122539 0.2896916 0.2660852 0.2419707
> pnorm(x)
 [1] 0.1586553 0.1840601 0.2118554 0.2419637 0.2742531 0.3085375 0.3445783
 [8] 0.3820886 0.4207403 0.4601722 0.5000000 0.5398278 0.5792597 0.6179114
[15] 0.6554217 0.6914625 0.7257469 0.7580363 0.7881446 0.8159399 0.8413447
```

```
# 指定累积概率的q值
qnorm(p, mean=0, sd=1, lower.tail = TRUE, log.p = FALSE)
```

```
> x=seq(0,1,0.1)
> qnorm(x)
[1]      -Inf -1.2815516 -0.8416212 -0.5244005 -0.2533471  0.0000000
[7]  0.2533471  0.5244005  0.8416212  1.2815516       Inf
```

19.16 t分布

19.16.1 产生t分布的随机数

产生10个自由度为5的t分布随机数.

```
> rt(n=10,df=5)
[1]  0.7965116  0.9019405  0.2392244  0.3129466 -0.2910085 -1.2970800
[7]  1.4356046  0.1165443  0.9069540  0.3450907
```

19.16.2 密度-累积概率密度函数

```
dt(x, df, ncp=0, log = FALSE)
pt(q, df, ncp=0, lower.tail = TRUE, log.p = FALSE)
```

```
> x=-5:5
> x
[1] -5 -4 -3 -2 -1  0  1  2  3  4  5
> dt(x,df=20)
[1] 0.0000789891 0.0008224743 0.0079637866 0.0580872152 0.2360456491
[6] 0.3939885857 0.2360456491 0.0580872152 0.0079637866 0.0008224743
[11] 0.0000789891
> pt(x,df=20)
[1] 3.436514e-05 3.517616e-04 3.537949e-03 2.963277e-02 1.646283e-01
[6] 5.000000e-01 8.353717e-01 9.703672e-01 9.964621e-01 9.996482e-01
[11] 9.999656e-01
```

19.16.3 指定累积概率的q值

产生累积概率为0.025, 0.975的自由度为20的t分布的值

```
> qt(p=0.025,df=20)
[1] -2.085963
> qt(p=0.975,df=20)
[1] 2.085963
```

19.17 χ^2分布

19.17.1 产生χ^2分布的随机数

产生10个自由度为20的χ^2分布的随机数

```
> rchisq(n=10,df=20)
[1] 13.26240 20.74800 17.96519 14.57688 16.04691 28.31448 16.28799 32.64230
[9] 13.38085 15.97800
```

19.17.2 密度-累积概率密度函数

```
dchisq(x, df, ncp=0, log = FALSE)
pchisq(q, df, ncp=0, lower.tail = TRUE
, log.p = FALSE)

> x=0:10
> x
[1]  0  1  2  3  4  5  6  7  8  9 10
> dchisq(x,df=5)
[1] 0.00000000 0.08065691 0.13836917 0.15418033 0.14397591 0.12204152
[7] 0.09730435 0.07437127 0.05511196 0.03988664 0.02833456
> pchisq(x,df=5)
[1] 0.00000000 0.03743423 0.15085496 0.30001416 0.45058405 0.58411981
[7] 0.69378108 0.77935969 0.84376437 0.89093584 0.92476475
```

19.17.3 指定累积概率的q值

```
> qchisq(p=0.025,df=5)
[1] 0.8312116
> qchisq(p=0.975,df=5)
[1] 12.83250
```

19.18 二项分布,泊松分布,正态分布的关系

19.19 正态分布与卡方分布,t分布,F分布的关系

1. 若 $\xi \sim N(0,1)$, 则

$$\eta = \xi^2 \sim \chi^2(1)$$

2. 若 $\xi \sim \chi^2(k), \eta \sim \chi^2(j)$, 且 ξ, η 相 互 独 立, 则

$$\eta + \xi \sim \chi^2(k+j)$$

推 论:

- 若 $\xi_i (i = 1, \cdots, n)$相 互 独 立, 且$\xi_i \sim \chi^2(k_i)$, 则

$$\xi = \sum_{i=1}^{n} \xi_i \sim \chi^2(\sum_{i=1}^{n} k_i)$$

- 若 $\xi_i (i = 1, \cdots, n)$相 互 独 立, 且$\xi_i \sim N(0,1)$, 则

$$\xi = \sum_{i=1}^{n} \xi_i \sim \chi^2(n)$$

- 若 ξ_1, ξ_2相 互 独 立, 且$\xi_1 \sim \chi^2(k), \xi_2 \sim \chi^2(j)$, 则

$$\xi_1 - \xi_2 \sim \chi^2(k-j)$$

3. 若$\xi \sim N(0,1), \eta \sim \chi^2(k)$, 且 相 互 独 立, 则

$$\xi / \sqrt{\frac{\eta}{k}} \sim t(k)$$

4. 若$\xi \sim \chi^2(k), \eta \sim \chi^2(j)$, 且 相 互 独 立, 则

$$\frac{\xi}{k} / \frac{\eta}{j} \sim F(k,j)$$

19.20 柯 西 分 布

若 随 机 变 量 概 率 密 度 为

$$f(x) = \frac{1}{\pi} \frac{1}{1+x^2}, \qquad -\infty < x < \infty$$

称x服从柯西分布或布雷特-维格纳(Breit-Wigner)分布. 特征函数为

$$\varphi(x) = e^{-|t|}$$

严格意义的柯西分布的各阶矩都是发散的, 因为

$$f(x) = \lim_{L\to\infty} \int_{-L}^{L} x^k \frac{1}{\pi} \frac{1}{1+x^2}, \qquad -\infty < x < \infty$$

不存在.

实际测定柯西分布的时候, 它的值域总是有限的. 因此, 可以将随机变量X的值域取为$[-L, L]$, 此时, 柯西分布的归一化概率密度为

$$f'(x) = \frac{1}{2arctanL\pi} \frac{1}{1+x^2}, \qquad -L < x < L$$

从分布的对称性立即可知, $f'(x)$所有奇次阶原点矩为0. 特别$E(x) = 0$. 方差为

$$V(X) = \frac{L}{arctanL} - 1$$

加法定理: 若M1,M2为两个独立的随机变量, 服从柯西分布, 中心值为M_{01}, M_{02}, 半峰宽为Γ_1, Γ_2. 那么$M = M1 + M2$的特征函数为

$$\varphi(M) = \varphi_{M1}(t)\varphi_{M2}(t) = e^{-i(M_{01}+M_{02})t-(\Gamma_1+\Gamma_2)|t|}$$

显然, M也服从柯西分布, 其中心值为

$$M_0 = M_{01} + M02$$

半峰宽为

$$\Gamma = \Gamma_1 + \Gamma_2$$

可以推广到多个柯西分布随机变量之和的情况.

19.21 Dirichlet分布

参考 http://en.wikipedia.org/wiki/Dirichlet_distribution 有详细描述

Dirichlet分布是一族连续多维分布. 是多维 beta distribution 的推广, 也是 Bayesian statistics 中 categorical distribution 和 multinomial distribution 的共轭分布. 即它的概率密度函数是 K 个对等事件(rival events)为x_i, 每个事件观察到 $\alpha_i - 1$ 次的概率.

概率密度函数为

$$f(x_1, \ldots, x_{K-1}; \alpha_1, \ldots, \alpha_K) = \frac{1}{\mathrm{B}(\alpha)} \prod_{i=1}^{K} x_i^{\alpha_i - 1}$$

所有$x_1, ..., x_{K-1} > 0$, 且$x_1 + ... + x_{K-1} < 1$, $x_K = 1–x_1–...–x_{K-1}$. 概率密度在 (K 1) 维空间(边长为1的超立方体)之外为0.

归一化常数为 multinomial beta function, 可以表示为 gamma function:

$$\mathrm{B}(\alpha) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}, \qquad \alpha = (\alpha_1, \ldots, \alpha_K)$$

性质, 与其它分布的关系: 略...

Chapter 20

相关与协方差

参考 [21] 3,4 多元数据的数据特征与相关分析

记 $x = x_1, x_2, \cdots, x_n$. $y = y_1, y_2, \cdots, y_n$.

20.1 协方差

$$s_{xy} = \frac{1}{n-1}\sum(x_i - \bar{x})(y_i - \bar{y})$$

20.2 协方差矩阵

$$S = \begin{bmatrix} s_{xx} & s_{xy} \\ s_{xy} & s_{yy} \end{bmatrix}$$

20.3 相关系数

相关系数实际上是中心化与标准化后的协方差

$$r = \frac{s_{xy}}{\sqrt{s_{xx}}\sqrt{s_{yy}}}$$

```
ore<-data.frame(
    x=c(67, 54, 72, 64, 39, 22, 58, 43, 46, 34),
    y=c(24, 15, 23, 19, 16, 11, 20, 16, 17, 13)
)

# 相关矩阵
> cor(ore)
        x       y
x 1.0000000 0.9202595
y 0.9202595 1.0000000

# 协方差矩阵
> cov(ore)
       x       y
x 252.7667 60.60000
y  60.6000 17.15556
```

20.4 相关系数的区间估计

可以证明, 当样本充分大, 样本相关总体也相关. 但是样本比较少时, 无法得到可靠的结论. 问题是, 样本个数 n 取多少才能保证总体也相关?

Ruben 给出了总体相关系数的区间估计的近似逼近公式. 设

n 为样本个数, r 为样本相关系数, $u = z_{\alpha/2}$, 则计算

$$\begin{aligned} r^* &= \frac{r}{\sqrt{1-r^2}} \\ a &= 2n-3-u^2 \\ b &= r^*\sqrt{(2n-3)(2n-5)} \\ c &= (2n-5-u^2)r^{*2}-2u^2 \end{aligned}$$

求方程 $ay^2 - 2by + c = 0$ 的根

$$y_1 = \frac{b-\sqrt{b^2-ac}}{a}$$

$$y_2 = \frac{b+\sqrt{b^2-ac}}{a}$$

则 $1-\alpha$ 的双侧置信区间为

$$L = \frac{y_1}{1+y_1^2}$$

$$U = \frac{y_2}{1+y_2^2}$$

下面是一个例子. $n = 6$ 时即使 $r = 0.8$ 也不可靠. $n = 25$ 则总体可以是相关的.

```
ruben.test <- function(n, r, alpha=0.05){
  u <- qnorm(1-alpha/2)
  r_star <- r/sqrt(1-r^2)
  a <- 2*n-3-u^2
  b <- r_star*sqrt((2*n-3)*(2*n-5))
  c <- (2*n-5-u^2)*r_star^2-2*u^2
  y1 <- (b-sqrt(b^2-a*c))/a
  y2 <- (b+sqrt(b^2-a*c))/a
  data.frame(n = n, r = r, conf = 1-alpha,
    L = y1/sqrt(1+y1^2), U = y2/sqrt(1+y2^2))
}
```

```
# n=6, r=0.8
> ruben.test(n=6,r=0.8)
  n   r conf          L         U
1 6 0.8 0.95 -0.09503772 0.9727884

# n=25, r=0.7
> ruben.test(n=25,r=0.7)
   n   r conf         L         U
1 25 0.7 0.95 0.4108176 0.8535657
```

相关系数置信区间的方法还有 David(1954) 提出的图表法, Kendall 与 Stuart (1961) 提出的 Fisher 逼近法等.

最有效的方法是做总体的相关性检验. 可以证明, 当 $(X,Y)^T$ 为二元正态总体, 且 $\rho(X,Y)=0$ 时

$$t=\frac{r_{xy}\sqrt{n-2}}{\sqrt{1-r_{xy}^2}}$$

服从自由度为 $n-2$ 的 t 分布. 由于相关系数 r_{xy} 称为 Pearson 相关系数, 故此检验称为 Pearson 相关检验.

其它还有 Spearman 秩检验和 Kendall 秩检验.

R 函数 cor.test() 可以进行 Pearson, Spearman 秩检验和 Kendall 秩检验三种方法.

```
> attach(ore)
> cor.test(x,y,method='pearson')

        Pearson's product-moment correlation

data:  x and y
t = 6.6518, df = 8, p-value = 0.0001605
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.6910290 0.9813009
sample estimates:
```

```
     cor
0.9202595
```

20.5 各种相关的检验

非参数检验很多时候都在讨论各种相关性的度量与检验. 包括二项比例, 列联表, 秩, 多个样本等等的相关性分析.

另外参考回归部分 chapter 28 section 29.2, 讨论回归系数的相关性

R默认的已经有很多函数做相关性的度量及检验.

coin 包含了很多的相关性检验的函数, 可以参考, help(pac="coin") .

Chapter 21

点估计与区间估计

参考文献 [6] 第七章

参考文献 [21] 第四章

参数估计有点估计和区间估计两方面的问题. 点估计有矩法, 极大似然法, 贝叶斯估计, 最小二乘估计等.

非参数估计问题例如随机变量$\xi\eta$之间有一定的相关性, 试问在什么准则下, 由一个对另外一个的预测为最佳.

21.1 矩法

英国统计学家 K. Pearson 引入的矩法是较早的参数点估计的方法.

矩法是最古老的点估计方法. 不要求知道分布函数. 但是要求随机变量的原点矩存在. 否则就不能估计了. 利用矩法估计均值和方差, 等价于用样本的一阶原点矩估计均值, 二阶中心矩估计方差. 由于矩与分布函数无关, 那么矩法还没有充分利用分布函数对参数提供的信息, 因此一般不是有效或充分的估计量. 虽然缺乏理论上的最优性质, 但是由于其简单易行, 在实际问题中仍然广泛使用.

21.1.1 一般描述

设总体X的分布函数$F(x;\theta_1,\cdots,\theta_m$中有m个未知参数. 假设总体的m阶原点矩存在. n个样本$x_1,\cdots,x_n$令总体的k阶原点矩等于样本的k阶原点矩, 即

$$E(X)=\frac{1}{n}\sum_{i=1}^{n}x_i$$

$$E(X^2)=\frac{1}{n}\sum_{i=1}^{n}x_i^2$$

$$\cdots$$

$$E(X^m)=\frac{1}{n}\sum_{i=1}^{n}x_i^m$$

解此方程组得到$\hat{\theta}_1,\cdots,\hat{\theta}_m$, 并使用$\hat{\theta}_k$作为参数$\theta_k$的估计, 则称$\hat{\theta}_k$为参数$\theta_k$的矩法估计量.

21.1.2 估计均值与方差

更一般的提法为: 利用样本的数字特征作为总体的数字特征的估计. 例如, 无论总体服从什么分布, 其均值和方差分别为$E(X)=\mu, E[(X-E(X))^2]=\sigma^2$. 使用矩法估计均值和方差. 列出方程组

$$\begin{aligned} E(X) &= \mu=\frac{1}{n}\sum_{i=1}^{n}x_i=\bar{x} \\ E(X^2) &= Var(X)+[E(X)]^2=\sigma^2+\mu^2=\frac{1}{n}\sum_{i=1}^{n}x_i^2 \end{aligned}$$

解得均值与方差的矩法点估计

$$\begin{aligned} \hat{\mu} &= \bar{x} \\ \hat{\sigma}^2 &= \frac{1}{n}\sum_{i=1}^{n}x_i^2-\bar{x}^2=\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2 \end{aligned}$$

注意, 方差的矩估计不等于样本方差S^2, 而是

$$\hat{sigma}^2 = \frac{n-1}{n}S^2$$

21.1.3 例1: 贝努里分布

求贝努里分布(两点分布, 硬币实验)参数p的矩法估计量.

设随机变量X服从贝努里分布, 成功$X=1$, 失败$X=0$. $E(X)=p$. 设试验n次, 成功m次. 则p的矩法估计为

$$\hat{p} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{m}{n}$$

即使用成功次数出现的频率作为概率p的估计.

21.1.4 例2: 均匀分布

设随机变量X服从$[0,\theta]$的均匀分布, 现有n个样本$x_1,\cdots,x_n$. 试估计参数θ.

均匀分布的一阶矩(均值)为$\theta/2$, 故其估计为

$$E(X) = \frac{\theta}{2} = \bar{x} \Longrightarrow \hat{\theta} = 2\bar{x}$$

21.1.5 例3: 均匀分布

设随机变量X服从$[\theta_1,\theta_2]$的均匀分布, 现有n个样本$x_1,\cdots,x_n$. 试估计参数θ_1,θ_2.

我们使用一阶原点矩估计均值, 二阶耶酥教估计方差, 即

$$\begin{aligned} E(X) &= \frac{theta_1+\theta_2}{2}=\bar{x} \\ Var(X) &= \frac{(theta_2-\theta_1)^2}{12}=\frac{1}{n}\sum_{i=1}^{n}(x_1-\bar{x})^2=S^2 \end{aligned}$$

解方程组得

$$\begin{aligned} \hat{\theta}_1 &= \bar{x}-\sqrt{3}S \\ \hat{\theta}_2 &= \bar{x}+\sqrt{3}S \end{aligned}$$

我们使用rootSolve 包的函数 multiroot() 解此方程组

```
x=c(4, 5, 2, 9, 5, 1, 6, 4, 6, 2)
m1=mean(x) # 均值
m2=sum((x-mean(x))^2)/10 # 方差
# x=[theta_1, theta_2]
model <- function(x,m1,m2){
   c(F1= x[1]+x[2]-2*m1,
      F2= (x[2] - x[1])^2/12 - m2)}
# 求解
> multiroot(f=model,start=c(0,10),m1=m1,m2=m2)
$root #  theta_1  theta_2
[1] 0.5115551 8.2884449

$f.root
         F1          F2
-1.713101e-10  1.205959e-06

$iter
[1] 4

$estim.precis
[1] 6.030653e-07

# 按照公式计算的 theta_1 theta_2
```

```
> m1-sqrt(3*m2)
[1] 0.5115556
> m1+sqrt(3*m2)
[1] 8.288444
```

21.1.6 例4: 二项分布

设总体服从二项分布$B(N,p)$, N,p为未知参数. 均值(一阶原点矩)为$M1 = N*p$,方差(二阶中心矩)为$M2 = N*p*(1-p)$. 建立方程组

$$\begin{aligned} F1 &= Np - M1 = 0 \\ F2 &= Np(1-p) - M2 = 0 \end{aligned}$$

解析结果为

$$N = \frac{M1^2}{M1 - M2}, \quad p = \frac{M1 - M2}{M1}$$

```
# N=20,p=0.7, 试验次数n=100
x<-rbinom(100, 20, 0.7);
m1=mean(x)
m2=sum((x-mean(x))^2)/100
> m1
[1] 13.84
> m2
[1] 4.8544

# 先给出解析计算的结果
> N=m1^2/(m1-m2); N
[1] 21.31695
> p=(m1-m2)/m1; p
[1] 0.6492486

# 下面使用 multiroot() 函数计算
# x=[N,p]
model <- function(x,m1,m2){
```

```
   c(F1= x[1]*x[2]-m1,
      F2= x[1]*x[2]*(1-x[2])- m2)}
multiroot(f=model,start=c(20,1),m1=m1,m2=m2)
# 下面是结果
$root
[1] 21.3169515  0.6492486

$f.root
         F1          F2
 1.205192e-08 -3.955911e-08

$iter
[1] 5

$estim.precis
[1] 2.580551e-08
```

21.2 极大似然法(MLE)

极大似然估计(Maximum likelyhood estimation, MLE)是Fisher1912年提出的应用非常广泛的参数估计方法, 其思想始于Gauss的误差理论. 它充分利用了分布函数的信息, 克服了矩法的某些不足.

21.2.1 极大似然原理

下面是一个摸球的例子. (参考文献[6] 7.1). 一个布袋里面有黑球和白球. 我们要估计它们的比例是1/4还是3/4. 现在有放回的抽取了3个球, 其中黑球的个数记为x. 我们就要通过黑球的数目来判断$p = 1/4$还是$p = 3/4$. 下面是$p = 1/4$和$p = 3/4$出现黑球个数的概率从表中确定, 当$x = 0, 1$时, $p = 1/4$, 当$x = 2, 3$时, $p = 3/4$.

一般的说, 我们把参数θ看作未知参数. 观察值是随机变量的一次实现. 不同的θ对应于观察值出现的概率不同. 既然出现了

Table 21.1: 不同参数下黑球出现个数的概率

x	0	1	2	3
P(x;3/4)	1/64	9/64	27/64	27/64
P(x;1/4)	27/64	27/64	9/64	1/64

观察值, 我们认为如果某个参数应该是使得此观察值出现的概率比其它参数时观察值出现的概率要大, 那么这个参数应该就是此观察值对应的参数. 这就是极大似然原理.

21.2.2 似然函数

记概率密度函数(离散时为分布律)为$f(x;\theta)$, 观察值$x = x_1, \cdots, x_n$. 称下面的函数

$$L(\theta; x) = L(\theta_1, \cdots, \theta_l; x_1, \cdots, x_n) = \prod_{i=1}^{n} f(x_i; \theta)$$

为参数θ的似然函数(likelyhood function).

显然, 样本固定时, $L(\theta; x)$是θ的函数, 若θ固定, 则$L(\theta; x)$就是样本的联合概率密度函数(离散的时候为联合分布律)

21.2.3 极大似然估计(MLE)

使得$L(\theta; x)$最大的一个(一组)θ值称为参数θ的极大似然估计(MLE), 即

$$L(\hat{\theta}; x) = max(L(\theta; x)),$$

称$\hat{\theta}$为参数的极大似然估计量.

21.2.4 似然方程的求解

由极值的一阶必要条件, 似然函数$L(\theta; x)$对参数偏导得似然

方程(likelyhood equation)

$$\frac{\partial L(\theta;x)}{\partial \theta_i}=0,\quad i=1,\cdots,l$$

连乘形式计算不方便,取对数得等价形式,对数似然方程(loglikelyhood equation)

$$\frac{\partial \ln L(\theta;x)}{\partial \theta_i}=0,\quad i=1,\cdots,l$$

严格讲,极大似然估计一定是似然方程或对数似然方程的解,但是似然方程或对数似然方程对参数的二阶Hessen矩阵负定,则似然方程或对数似然方程的解才是极大似然估计.

21.2.5 例1: 正态分布

设X服从正态分布$N(\mu,\sigma^2)$. $x=x_1,\cdots,x_n$为来自总体的一组样本. 试用极大似然法估计参数μ,σ^2.

似然函数为

$$L(\mu,\sigma^2;x)=\prod_{i=1}^{n}f(x_i;\mu,\sigma^2)=(2\pi\sigma^2)^{-n/2}exp[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2]$$

对数似然函数为

$$\ln L(\mu,\sigma^2;x)=-\frac{n}{2}\ln(2\pi\sigma^2)-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2$$

求偏导得到对数似然方程

$$\begin{aligned}\frac{\partial \ln L(\mu,\sigma^2;x)}{\partial \mu} &= -\frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2=0\\ \frac{\partial \ln L(\mu,\sigma^2;x)}{\partial \sigma^2} &= -\frac{n}{2\sigma^2}+\frac{1}{2\sigma^4}\sum_{i=1}^{n}(x_i-\mu)^2=0\end{aligned}$$

解此似然方程组得到[1]

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i = \bar{x}, \quad \sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

验证对数似然函数的二阶Hesse矩阵为负定, 故此估计就是似然方程的极大值点, 与矩法的一阶二阶矩估计是一致的.

```
x=rnorm(10)

# multiroot()函数计算
# e[1]=\mu, e[2]=\sigma, x=样本
model <- function(e,x){
   n=length(x)
   c(F1= sum(x-e[1]),
      F2= -n/e[2] + sum((x-e[1])^2)/e[2]^3)}
> multiroot(f=model,start=c(0,1),x=x)
$root
[1] 0.1273094 1.1256564

$f.root
        F1         F2
5.551115e-17 1.394105e-08

$iter
[1] 5

$estim.precis
[1] 6.970523e-09
# 公式计算
> mean(x)
```

[1]第二个方程也可以为

$$\frac{\partial \ln L(\mu,\sigma^2;x)}{\partial \sigma} = -\frac{n}{\sigma} + \frac{1}{\sigma^3}\sum_{i=1}^{n}(x_i-\mu)^2 = 0$$

```
[1] 0.1273094
> sum((x-mean(x))^2)/10
[1] 1.267102
```

21.2.6 例2: 指数分布

设总体X服从指数分布, 密度函数为

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, & x >= 0 \\ 0, & x < 0 \end{cases}$$

$x = x_1, \cdots, x_n$为来自总体的一组样本. 估计参数λ.

$$\ln L(\lambda; x) = n \ln \lambda - \lambda \sum_{i=1}^{n} x_i$$

取导数

$$\frac{\partial \ln L(\lambda; x)}{\partial \lambda} = \frac{n}{\lambda} - \sum_{i=1}^{n} x_i$$

解得

$$\lambda = \frac{n}{\sum_{i=1}^{n} x_i}$$

二阶导数(对应Hesse矩阵)$-\frac{n}{\lambda^2} < 0$, 故此估计为极大点.

21.2.7 例3: 均匀分布

当参数空间(可能的值)为开区域, 此时似然方程组解的方法不适用.

设总体X服从区间$[a,b]$的均匀分布. $x=x_1,\cdots,x_n$为来自总体的一组样本. 估计参数a,b.

似然函数为

$$L(a,b;x)=\begin{cases}\frac{1}{(b-a)^n}, & a<=x_i<=b, \quad i=1,\cdots,n\\ 0, & others\end{cases}$$

显然, $L(a,b;x)$不是a,b的连续函数, 其似然方程为

$$\begin{aligned}\frac{\partial \ln L(a,b;x)}{\partial a} &= \frac{n}{b-a}=0\\ \frac{\partial \ln L(a,b;x)}{\partial b} &= \frac{-n}{b-a}=0\end{aligned}$$

因此不能求解.

应该从极大似然估计的定义出发来求$L(a,b;x)$的最大值. 要$L(a,b;x)$达到最大, 那么$b-a$应该尽可能的小, 但是a不能大于$min(x)$, b不能小于$max(x)$. 因此a,b的极大似然估计为

$$\hat{a}=min(x), \quad \hat{b}=max(x)$$

21.2.8 例4: 钓鱼问题

在鱼塘钓出r条鱼, 做上记号, 然后再钓出s条, 发现有x条有标记. 试估计鱼塘所有的鱼有多少?

第二次钓出的鱼的条数X服从超几何分布

$$P(X=x)=\frac{C_r^x C_{N-r}^{s-x}}{C_N^s}$$

似然函数为

$$L(N;x)=P(X=x)$$

直接对似然函数求导相当困难, 那么考虑似然函数的比值

$$g(x;N)=\frac{L(N;x)}{L(N-1;x)}=\frac{(N-s)(N-r)}{N(N-r-s+x)}=\frac{N^2-(r+s)N+rs}{N^2-(r+s)N+xN}$$

当$rs>xN$时 有$g(x;N)>1$, $rs<xN$时 有$g(x;N)<1$, 即似然函数$L(N;x)$在$N=\frac{rs}{x}$附近达到最大. 即N的极大似然估计为

$$\hat{N}=[\frac{rs}{x}],\quad []\text{表示取整数}$$

21.2.9 例5: Cauchy分布(数值方法)

设总体X服从Cauchy分布, 密度函数为

$$f(x;\theta)=\frac{1}{\pi[1+(x-\theta)^2]},\quad -\infty<x<\infty$$

$x=x_1,\cdots,x_n$为来自总体的一组样本. 估计参数θ.

Cauchy分布的似然函数为

$$L(\theta;x)=\prod_{i=1}^{n}f(x_i;\theta)=\frac{1}{\pi^n}\prod_{i=1}^{n}\frac{1}{\pi[1+(x_i-\theta)^2]}$$

求导得到对数似然方程为

$$\sum_{i=1}^{n}\frac{x_i-\theta}{1+(x_i-\theta)^2}=0$$

求对数似然方程的解析解是困难的, 考虑使用数值方法.

使用 uniroot() 函数

```
# 参数为1的cauchy分布
```

```
x=rcauchy(100,1)
f<-function(p) sum((x-p)/(1+(x-p)^2))
out<-uniroot(f,c(0,5))
> out
$root
[1] 0.7481134

$f.root
[1] 0.0001692195

$iter
[1] 5

$estim.prec
[1] 6.103516e-05
```

使用 optmize() 函数

```
loglike<-function(p)sum(log(1+(x-p)^2))
> optimize(loglike,c(0,5))
$minimum
[1] 0.7481312

$objective
[1] 129.1854
```

21.3 TODO: 最小二乘法

参考文献[17] 第九章

长时间来, 最小二乘法是最广泛使用的参数估计方法之一. 与极大似然法不同, 最小二乘法得到的估计量并没有一般意义上的最优性质或者是渐近的最优性质. 但是对于线性模型, 即观测值所服从的分布与待估计参数具有线性关系这一类经常遇到的重要问题, 最小二乘法具有突出的优点, 最小二乘估

计量是达到最小方差界的无偏估计量,并且这一性质与子样容量无关. 同时, 最小二乘估计量又与观测服从的分布无关, 因而, 当总体分布的函数形式并不严格知道, 无法进行极大似然法估计时, 运用最小二乘法是十分方便的.

21.3.1 最小二乘原理

21.4 均值估计

21.4.1 点估计

总体均值μ的最小方差无偏估计为样本的均值 $\bar{x}$.

21.4.2 均值的标准误

均值标准误差的估计量是 $s/\sqrt{n}$ –样本均值集合的标准差. 实际上, 总体方差常常未知. 后面会看到, σ^2 的合理估计是 s^2.

21.4.3 均值的区间估计–总体方差已知

总体方差已知时, 均值为正态分布.

我们常常希望得到均值的严格似乎合理的区间估计. 下面的区间估计仅当未知分布是正态分布才是正确的. 若不是正态分布, 则只能近似成立.

若$\bar{x} \sim N(\mu, \sigma^2/n)$, 那么把 $\bar{x}$ 写为标准形式, 即

$$z = \frac{\bar{x} - \mu}{\sigma/\sqrt{n}}$$

则 z 应该是标准正态分布. 当重复抽样时, 95% 的 z 值落入 -1.96 到 1.96 之间. 但是 σ 在实际中很少知道.

```
# x 为均值 5, 方差 1 的总体中抽取的 10 个样本
> x=rnorm(10,5)
> x
 [1] 4.927264 4.067237 6.136822 5.722123 6.286754 3.266601 4.443779 3.630787
 [9] 4.874269 3.748306
# z 值为 qnorm(0.025)=-1.959964, qnorm(0.975)=1.959964
> mean(x)+qnorm(0.025)*1/sqrt(10)
[1] 4.090599
> mean(x)+qnorm(0.975)*1/sqrt(10)
[1] 5.330189
```

21.4.4 均值的区间估计–总体方差未知

总体方差未知时, 均值为t分布.

当 σ 未知时, 合理的估计是用样本的标准差 s 估计 σ 而用代替后计算的 z 来构建置信区间. 问题是, 此时的 z 已经不是正态分布了. 此时的 z 的分布是 t 分布.

正态分布中均数的置信区间具未知方差的正态分布的均值 μ 的 $100\% * (1-\alpha)$ 置信区间 (confidence interval, CI) 可以写成

$$(\bar{x} - t_{n-1,1-\alpha/2}s/\sqrt{n}, \bar{x} + t_{n-1,1-\alpha/2}s/\sqrt{n})$$

样本均值为3, 标准差为5, 样本量为20的均值的95%的置信区间为

```
> 3+qt(p=0.025,df=20)*5/sqrt(20)
[1] 0.667822
> 3+qt(p=0.975,df=20)*5/sqrt(20)
[1] 5.332178
```

21.5 方差估计

21.5.1 点估计

按照公式即可

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x}^2)$$

```
> x
 [1] -5 -4 -3 -2 -1  0  1  2  3  4  5
> var(x)
[1] 11
> sum((x-mean(x))^2)/(length(x)-1)
[1] 11
```

21.5.2 区间估计

σ^2 的 $100\% * (1-\alpha)$ 置信区间为

$$[\frac{(n-1)s^2}{\chi^2_{n-1,\alpha/2}}, \frac{(n-1)s^2}{\chi^2_{n-1,1-\alpha/2}}]$$

```
> x
 [1] -5 -4 -3 -2 -1  0  1  2  3  4  5
> (10-1)*var(x)/qchisq(0.025,10-1)
[1] 36.66138
> (10-1)*var(x)/qchisq(0.975,10-1)
[1] 5.20429
```

21.6 二项分布的估计

21.6.1 参数p及标准误差的点估计

记 x 是二项随机变量, 其参数为 n 及 p, p 的无偏估计为事件中的样本比例 $\hat{p}$, 标准误差 $\sqrt{pq/n}$ 的精确估计为 $\sqrt{\hat{p}\hat{q}/n}$.

```
> x=rbinom(10,1,0.5)
> x
 [1] 1 1 0 1 1 1 0 0 1 0
> t=table(x)
> t
x
0 1
4 6
> t['1']/length(x) # 此即为p的点估计, 还可以使用binom.test(table(x))得到.
  1
0.6
> sqrt(t['1']*t['0']/length(x)) # 此为标准误差的点估计
      1
1.549193
```

21.6.2 p的区间估计

```
> binom.test(table(x))

        Exact binomial test

data:  table(x)
number of successes = 4, number of trials = 10, p-value = 0.7539
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.1215523 0.7376219
sample estimates:
```

```
probability of success
               0.4

> b=binom.test(table(x))
> str(b)
List of 9
 $ statistic  : Named int 4
  ..- attr(*, "names")= chr "number of successes"
 $ parameter  : Named int 10
  ..- attr(*, "names")= chr "number of trials"
 $ p.value    : Named num 0.754
  ..- attr(*, "names")= chr "0"
 $ conf.int   : atomic [1:2] 0.122 0.738
  ..- attr(*, "conf.level")= num 0.95
 $ estimate   : Named num 0.4
  ..- attr(*, "names")= chr "probability of success"
 $ null.value : Named num 0.5
  ..- attr(*, "names")= chr "probability of success"
 $ alternative: chr "two.sided"
 $ method     : chr "Exact binomial test"
 $ data.name  : chr "table(x)"
 - attr(*, "class")= chr "htest"
```

Chapter 22

假设检验

22.1 各种情况使用的方法

Aim	Parametric tests	Non-parametric tests
compare two means	Student's T test	Wilcoxon's U test
compare more than two means	Anova (analysis of variance)	Kruskal--Wallis test
Compare two variances	Fisher's F test	Ansari-Bradley or Mood test
Comparing more than	Bartlett test	Fligner test

22.2 如何检验一个分布为指定分布

参考第 49 章

22.3 单样本假设检验

22.3.1 方差未知的正态分布均值的单样本检验

前提条件–数据为正态分布, 使用t.test(). 若数据非正态分布, 应该使用 Wilcoxon's U test（见非参数检验）.

```
> x=rnorm(200)
> t.test(x)

       One Sample t-test

data:  x
t = -1.1695, df = 199, p-value = 0.2436
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.21865305  0.05585082
sample estimates:
  mean of x
-0.08140112
```

可以看一下p值的分布, 若零假设成立, p值在[0,1]之间为均匀分布

```
> p <- c()
> for (i in 1:1000) {
+   x <- rnorm(200)
+   p <- append(p, t.test(x)$p.value)
+ }
> hist(p, col='light blue')
```

22.3.2 数据非正态时的情况

数据非正态时需要做转换使其变为正态分布, 或使用非参数检验.

数据为均匀分布时, 会出现下面的情况.

```
> N <- 1000
> n <- 3
> v <- vector()
> for (i in 1:N) {
+   x <- runif(n, min=-1, max=1)
+   r <- t.test(x)$conf.int
+   v <- append(v, r[1]<0 & r[2]>0)
+ }
> sum(v)/N
[1] 0.919
```

数据正态分布时,

```
> N <- 1000
> n <- 100
> v <- vector()
> for (i in 1:N) {
+   x <- rnorm(n, sd=1/sqrt(3))
+   r <- t.test(x)$conf.int
+   v <- append(v, r[1]<0 & r[2]>0)
+ }
> sum(v)/N
[1] 0.947
```

可以看到, 将均匀分布作为正态分布时其置信区间的概率不是0.95而是0.92. 这增大了2型错误的概率.

但是样本量很大时, 误差就不明显了

```
> N <- 1000
> n <- 100
> v <- vector()
> for (i in 1:N) {
+   x <- runif(n, min=-1, max=1)
+   v <- append(v, t.test(x)$p.value)
+ }
> sum(v>.05)/N
[1] 0.957
```

22.3.3 方差已知的正态分布均值的单样本检验

此时使用z检验.

某些研究中, 根据过去的资料翻查可能方差是知道的. 在这种情况下, 检验统计量 t 可以由 z 代替, 临界值也由相应的标准正态分布的临界值代替. 其中

$$z = \frac{\bar{x} - \mu}{\sigma/\sqrt{n}}$$

其它的计算完全类似于方差未知时的 t 检验, 不论是单侧还是双侧.

下面例子假设总体方差为1. 检验其零假设为0. 样本量为100

```
> x=rnorm(100)
> z=(mean(x)-0)/(1/sqrt(100))
> z
[1] 2.005832
> pnorm(z)
[1] 0.9775629
```

22.3.4 功效与样本量

参考 power.t.test()

```
power.t.test(n = NULL, delta = NULL, sd = 1, sig.level = 0.05,
             power = NULL,
             type = c("two.sample", "one.sample", "paired"),
             alternative = c("two.sided", "one.sided"),
             strict = FALSE)
```

```
> power.t.test(n = 20, delta = 1) #已知样本量, 求功效

    Two-sample t test power calculation

             n = 20
         delta = 1
            sd = 1
     sig.level = 0.05
         power = 0.8689528
   alternative = two.sided

 NOTE: n is number in *each* group
```

```
> power.t.test(power=0.8, delta = 1)#已知功效, 求样本量

    Two-sample t test power calculation

             n = 16.71477
         delta = 1
            sd = 1
     sig.level = 0.05
         power = 0.8
   alternative = two.sided

 NOTE: n is number in *each* group
```

22.3.5 方差的区间估计及检验–卡方检验

R 中没有chisq.var.test()

在方差的置信区间估计及检验中, 正态条件特别重要. 若样本不满足正态性, 则临界值p-值及置信区间都不是有效的.

欲检验

$$H_0 : \sigma^2 = \sigma_0^2 \quad vs. \quad H_1 : \sigma^2 \neq \sigma_0^2$$

计算检验统计量

$$X^2 = (n-1)s^2/\sigma_0^2 \sim \chi^2_{n-1}$$

如果 $X^2 < \chi^2_{n-1,\alpha/2}$ 或 $X^2 > \chi^2_{n-1,1-\alpha/2}$, 则拒绝 H_0 如果 $\chi^2_{n-1,\alpha/2} \leq X^2 \leq \chi^2_{n-1,1-\alpha/2}$, 则接受 H_0

p-值(双侧备择)

同上计算检验统计量 X^2 如果 $s^2 \leq \sigma_0^2$, 则 p-值=2*(χ^2_{n-1} 分布曲线下从左到 X^2 的面积)

如果 $s^2 > \sigma_0^2$, 则 p-值=2*(χ^2_{n-1} 分布曲线下从右到 X^2 的面积)

下面是一个例子. 由于$var(x) <= 1$, 则$p = 2 * pchisq(q = chi2, df = 99)$. 若$var(x) > 1$, 则$p = 1 - 2 * pchisq(q = chi2, df = 99)$. 单侧检验不用2倍

```
> x=rnorm(100) # 检验x的总体的方差是否为1
> var(x)
[1] 0.9344586

> chi2=(100-1)*var(x)/1 #计算检验统计量
> chi2
[1] 92.5114
```

```
> qchisq(df=99,p=0.025) # 区间下侧
[1] 73.36108
> qchisq(df=99,p=0.975) # 区间上侧
[1] 128.422

> p=2*pchisq(q=chi2,df=99)  # p值
> p
[1] 0.671611
```

22.4 方差齐性检验-F检验

两个样本的均值t检验之前, 需要判断其方差是否相同. 正态样本使用此F检验

22.4.1 F分布的特点

具自由度 d_1, d_2 的F分布的下侧第p个百分位点, 就是具有自由度为 d_12, d_1 的F分布的上侧第p个百分位点的倒数, 即

$$F_{d_1,d_2,p} = 1/F_{d_2,d_1,1-p}$$

22.4.2 F检验

使用t检验之前需要此检验.

参考var.test()

```
> x <- rnorm(50, mean = 0, sd = 2)
> y <- rnorm(30, mean = 1, sd = 1)

> var.test(x, y)  # 第一种用法

        F test to compare two variances
```

```
data:  x and y
F = 6.1786, num df = 49, denom df = 29, p-value = 1.516e-06
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 3.104259 11.624472
sample estimates:
ratio of variances
        6.178575

> var.test(lm(x ~ 1), lm(y ~ 1))  # 第二种用法. The same.

       F test to compare two variances

data:  lm(x ~ 1) and lm(y ~ 1)
F = 6.1786, num df = 49, denom df = 29, p-value = 1.516e-06
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 3.104259 11.624472
sample estimates:
ratio of variances
        6.178575
```

手工计算

```
> f=var(y)/var(x)
> f
[1] 0.1618496
> qf(0.025, 49,29)
[1] 0.5315144
> qf(0.975, 49,29)
[1] 1.990354
> f<qf(0.025,49,29)
[1] TRUE
```

22.4.3 多于2个正态样本的方差检验

参考 bartlett.test

22.4.4 2个非正态样本的方差检验

参考 ansari.test 或 mood.test , 它们是非参数检验

22.4.5 多于2个非正态样本

参考 fligner.test

22.5 两样本均值的t检验

样本需正态分布, 非正态分布的数据需要转换为正态分布或使用非参数检验

对于两个样本方差不一样的情况, p值保持正确, 但是功效下降的很快. 若数据看起来是正态分布但是方差不同, 最好对它们归一化处理(x/var(x), y/var(y))然后使用t检验, 这样比使用非参数检验要好.

22.5.1 t检验

参考t.test()

用法为: 默认为非配对, 方差不相等(paired = FALSE, var.equal = FALSE). 若只有一个样本, mu代表其被检验的均值, 若两个样本(x, y)则mu代表其均值之差.

```
t.test(x, y = NULL,
```

```
          alternative = c("two.sided", "less", "greater"),
          mu = 0, paired = FALSE, var.equal = FALSE,
          conf.level = 0.95, ...)

## S3 method for class 'formula':
t.test(formula, data, subset, na.action, ...)
```

第一种用法, 数据在一个向量里, 由group指明不同的组

```
> d=sleep
> d
   extra group
1    0.7     1
2   -1.6     1
3   -0.2     1
4   -1.2     1
5   -0.1     1
6    3.4     1
7    3.7     1
8    0.8     1
9    0.0     1
10   2.0     1
11   1.9     2
12   0.8     2
13   1.1     2
14   0.1     2
15  -0.1     2
16   4.4     2
17   5.5     2
18   1.6     2
19   4.6     2
20   3.4     2

> t.test(extra ~ group, data = sleep)

        Welch Two Sample t-test

data:  extra by group
t = -1.8608, df = 17.776, p-value = 0.0794
```

```
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.3654832  0.2054832
sample estimates:
mean in group 1 mean in group 2
         0.75          2.33
```

第二种用法, 普通的两个数据

```
> attach(d) #将d的数据 extra, group 纳入名称空间, 可以直接使用
> t.test(extra[group == 1], extra[group == 2])

       Welch Two Sample t-test

data:  extra[group == 1] and extra[group == 2]
t = -1.8608, df = 17.776, p-value = 0.0794
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.3654832  0.2054832
sample estimates:
mean of x mean of y
    0.75      2.33
```

22.5.2 功效与样本量

参考 power.t.test type=paired

不配对的样本量估计参考流行病部分两个均值的样本量估计. epicalc 包的函数为 n.for.2means

功效见 epicalc 包的函数 power.for.2means

Chapter 23

奇异值的处理

参考文献 [14] page 284, chapter 8.9 奇异值的处理

某些奇异值可能对研究结论有重要影响. 因此, 需要识别奇异值并将其排除在外, 或至少做有或无奇异值时候的统计判断.

23.1 极端学生化偏差(ESD)

常用的方法是以远离均值的标准差的倍数的多少来定量的描述奇异值. 一个样本中把这个统计量应用于最极端的观察值时, 称为"极端学生化偏差"(extreme studentized deviate, ESD). 定义为

$$ESD = max_{i=1,\cdots,n}|x_i - \bar{x}|/s$$

其中s为标准差.

ESD取多大才能列为奇异值? 回答是与样本量有关. 在正态分布下, 样本量为n而没有奇异值的时候, 我们希望最大的值应该近似对应第$100\%\frac{n}{n+1}$个百分位点. 即正态分布的样本量为64的样本, 上式为$100 * 64/65 = 98.5$g 百分位点, 其值为2.17, 即如果有奇异值, 则该值的ESD应该大于2.17.

附录10中的临界值依赖于样本量n和你定义的百分位点$1-\alpha$, 计算公式为

$$ESD_{n,1-\alpha}=\frac{t_{n-2,p}(n-1)}{\sqrt{n(n-2+t^2_{n-2,p})}},\quad p=1-[\alpha/(2n)]$$

23.2 ESD的单个奇异值法

对于不出现在附录10中的n值, 也可以使用下面的方法近似求出. 样本量增加时, 临界值也增加.

假设我们有样本服从正态分布, 感觉有奇异值, 则在I型错误α下, 检验

$$H_0:\ no\ extreme\ value\quad vs.\quad H_1:\ has\ one\ extreme\ value$$

计算统计量ESD,

$$ESD=max_{i=1,\cdots,n}|x_i-\bar{x}|/s$$

记取此ESD为最大的样本为$x^{(n)}$.

差附录表10中的临界值, 记为$ESD_{n,1-\alpha}$

如果$ESD>ESD_{n,1-\alpha}$, 拒绝零假设, 认为此$x^{(n)}$为奇异值, 否则这个样本中没有奇异值.

23.3 ESD求多个奇异值法

当有多个奇异值的时候, 上面的求单个奇异值的方法不妥, 因为其标准差s的计算有些问题, 可能不能发现多个奇异值.

为了解决此问题, 我们需要首先对数据中的奇异值的个数做出判断, 给出一个合理的上限. 这个上限的一个经验的取法([14] page 286)为 $min([n/10],5)$. 如果一个数据中有多于5个奇异值, 除非样本量很大, 这个样本分布可能是非正态的.

设样本中大多数是服从正态分布的, 但怀疑其中有k个奇异值, $k = min([n/10], 5)$, $[n/10]$为小于等于$n/10$的最大整数. 在I型错误α下, 检验

H_0 : 无奇异值 $\quad vs. \quad$ H_1 : 至少1个但不超过k个奇异值

首先, 计算全体样本的ESD统计量, 找到最大的ESD对应的样本值$x^{(n)}$, 其值记为$ESD^{(n)}$.

去除$x^{(n)}$, 在其它$n-1$个样本数据中重新计算均值, s和ESD. 标记最大的样本及ESD值为$x^{(n-1)}, ESD^{(n-1)}$.

继续k次, 直到有k个ESD值和对应的x值.

从附表中查找每个ESD对应的临界值$ESD_{n,1-\alpha}, ESD_{n-1,1-\alpha}, \cdots, ESD_{n-k+1,1-\alpha}$.

从最后一个ESD开始, 若$ESD^{(n-k+1)} > ESD_{n-k+1,1-\alpha}$, 则我们认为k个值都是奇异值, 若上式不成立, 但是有$ESD^{(n-k+2)} > ESD_{n-k+2,1-\alpha}$, 则我们认为有$k-1$个奇异值, 对应的x值为奇异值. $\cdots$ 如果一直到$ESD^{(n-1)} <= ESD_{n,1-\alpha}$, 那么数据中就没有奇异值.

除非真实的奇异值多于k个, 否则这个方法有很好的功效.

23.4 处理奇异值的方法

有几种方法处理奇异值, 一种是检测出奇异值, 然后对有奇异值和没有奇异值的情况下分别分析数据以便比较.

另外, 也可以不删除奇异值, 而是将其在分析中的作用减小, 这可以有多种方法. 一种是把连续数据转换为等级变量(例如, 大, 中, 小等), 再使用等级数据来分析, 不过此时一般是使用非参数方法.

还有其它方法是对重要参数, 比如均值使用稳健的估计量, 这项估计量受样本中的奇异值影响较小, 但是又不排除它们.

上面的方法功效都要比t检验低, 但是如果确实有奇异值, 则上面的方法的功效是好的.

一般, 没有一种方法可以适合所有数据, 对一个研究, 如果几种方法得到的结论一致, 自然可以增加结果的可靠性.

23.5 备忘: 异常值

备忘: 回忆异常值的定义, 和奇异值不同.

异常值:

x >上百分位数+1.5×（上百分位数-下百分位数）
x <下百分位数-1.5×（上百分位数-下百分位数）

极端异常值:

x >上百分位数+3×（上百分位数-下百分位数）
x <下百分位数-3×（上百分位数-下百分位数）

23.6 例子

23.6.1 boxplot

boxplot中的out值为异常值, 非此处的奇异值. 其结果中的stats为异常值临界点, 上下25%百分位点, 中位数.

注意, boxplot结果中的10为out值, 但是在下面的奇异值检验中($\alpha = 0.05$)并不是奇异值.

```
> s=c(1:5,10)
> y=boxplot(s)
> y
```

```
$stats
     [,1]
[1,]  1.0
[2,]  2.0
[3,]  3.5
[4,]  5.0
[5,]  5.0

$n
[1] 6

$conf
        [,1]
[1,] 1.564903
[2,] 5.435097

$out  # 注意, 此异常值在下面的检验中不是奇异值.
[1] 10

$group
[1] 1

$names
[1] "1"
```

23.6.2 奇异值检验

```
# 计算ESD临界值. 算法参考生物统计学基础附表10
esd.threshold<-function(n,alpha){
   p=1-alpha/(2*n)
   x<-qt(p,df=n-2)
   res<-x*(n-1)/sqrt(n*(n-2+x^2))
   res
}

# 计算样本的ESD
esd<-function(x){
   # 实际上的计算为 abs((x-mean(x, na.rm=T))/sd(x, na.rm=T))
```

```
    res<-abs(scale(x))
    res
}

# 检测单个奇异值, alpha=0.05

# 10在boxplot中为异常值out, 但是并不是奇异值
> s=c(1:5,10)
> esd.s<-esd(s)
> thr<-esd.threshold(length(s),0.05)
> which(esd.s>thr) # 没有奇异值
integer(0)

# 第一个值15为奇异值
> s=c(15,1:5)
> esd.s<-esd(s)
> thr<-esd.threshold(length(s),0.05)
> which(esd.s>thr)
[1] 1

# 检测多个奇异值
esd.test<-function(x,k,alpha=0.05){
    n<-length(x)
    tmp<-x
    e.thr<-c()
    e.max<-c()
    pos<-c()
    for (i in 1:k){
        t<-esd.threshold(n-i+1,alpha)
        e<-esd(tmp)
        a<-which(e==max(e,na.rm=T))
        e.max<-append(e.max,e[a])
        e.thr<-append(e.thr,rep(t,length(a)))
        pos<-append(pos,a)
        #cat("--------\n",a,e[a],t,"\n")
        tmp[a]<-NA
    }
    #print(data.frame(e.max,e.thr,pos))
    m=0
    for(i in k:1){
        if (e.max[i]>e.thr[i]){
```

```
            m=i
            break
        }
    }
    if(m>0){
        res<-data.frame(which=pos[1:m],value=x[pos[1:m]],
                    esd=e.max[1:m],esd.thr=e.thr[1:m])
    }
    else{ # 长度为0的data.frame
        res<-data.frame(which=NULL,value=NULL,
                    esd=NULL,esd.thr=NULL)
    }
    res
}

> x=exp(seq(0,5,by=0.2))
> esd.test(x,2,0.05)
  which    value     esd  esd.thr
1    26 148.4132 2.880596 2.840774
2    25 121.5104 2.826053 2.821681
> esd.test(x,3,0.05)
  which    value     esd  esd.thr
1    26 148.4132 2.880596 2.840774
2    25 121.5104 2.826053 2.821681
> esd.test(x,4,0.05)
  which    value     esd  esd.thr
1    26 148.4132 2.880596 2.840774
2    25 121.5104 2.826053 2.821681


> x=c(15,1:5)
>  k=max(min(floor(length(s)/10),5),1) # k=1
>  alpha=0.05
>  esd.test(x,k,alpha)
  which value     esd  esd.thr
1     1    15 1.961161 1.887145

> x=c(15,20,1:5)
>  k=2
>  alpha=0.05
>  esd.test(x,k,alpha)
```

```
  which value     esd  esd.thr
1     2    20 1.752729 2.019969
2     1    15 1.961161 1.887145
```

Part IV

方差分析

本部分参考文献除了[14], R部分主要参考《Statistics with R》，《Practical Regression and Anova using R》，其它《simpleR》《R语言简介》《R for beginners》等也有少量涉及。

Chapter 24

开始之前

参考《Practical Regression and Anova using R》

描述问题, 往往比解决更重要. (The formulation of a problem is often more essential than its solution which may be merely a matter of mathematical or experimental skill. —-Albert Einstein)

搜集数据. 理解数据如何搜集的非常重要.

初步分析. 非常重要. 看看数据大概的样子, 是否偏斜, 是否有错误数据, 异常值等等

最后才是回归或方差分析.

24.1 非正态数据的转换

开始分析之前, 需要使数据看起来为正态分布. 重要的是对称且没有极端值. 几乎所有情况下, 转换的同时也去除了残差问题, 例如方差不齐. 无一定的转换模式, 通常使用手工转换. 请参考 chapter 14 数据变换.

24.2 不能转换为正态数据的多重比较–Kurskal-Wallis检验

普通参数方法称为"单因素方差分析", 或有时候称为单因素F检验. 违反正态假设可能对F有一些影响, 但是某些非正态分布的数据(例如有极值)F检验的功效会比Kurskal-Wallis 检验小很多.

相对于F检验, Kurskal-Wallis 检验的A.R.E.从来不会小于0.864, 若是正态分布, A.R.E.=0.955. 均匀分布=1.0, 双指数分布=1.5.

总体非正态分布, 或根本是有序数据(例如, 得分), 那么应该使用非参数统计的 Kurskal-Wallis 检验48.3 对应正态数据的方差分析.

然后使用 Mann-Whitney 检验或 Kurskal-Wallis 做两两比较即可.

24.3 非正态的残差

如果残差非正态, 最小平方估计就不是最优的. 其它一些鲁棒的方法可能更好, 虽然可能是有偏的. 最坏的情况下, 所有的结果都是错的, 包括检验, 方差, 区间等. 参考回归诊断-残差及检验30.2

但是, 如果残差分布比正态紧密, 或样本量非常大, 则可以忽略这个问题.

可以使用 shapiro.test() 检验残差的正态性. 参考第 49 章: Kolmogorov-Smirnov 型统计量

可以使用 histograms, box-and-whiskers plots (boxplots), qqplot(quantile-quantile plots) 来查看残差.

```
> x <- runif(100)
> y <- 1 - 2*x + .3*exp(rnorm(100)-1) # 产生非均匀的数据
```

```
> r <- lm(y~x)
# 绘图
> boxplot(r$residuals, horizontal=T)
> hist(r$residuals, breaks=20, probability=T, col='light blue')
> lines(density(r$residuals), col='red', lwd=3)
> qqnorm(r$residuals) # normal qq plot
> qqline(r$residuals,col='red')
```

24.4 异质性噪声

noise(噪声) 随x不同而不同. 最简单的方法是对数据做变换. 可能的话, 寻找一个转换即可以使残差正态, 又可以使噪声同质. 参考回归诊断-残差及检验30.2

广义最小平方法允许对异质性噪声的数据做回归, 但是需要知道噪声的变化情况.

```
> x <- runif(100)
> y <- 1 - 2*x + .3*x*rnorm(100)
> r <- lm(y~x)
> n <- 10000
> xp <- sort(runif(n,))
> yp <- predict(r, data.frame(x=xp), interval="prediction")
> yr <- 1 - 2*xp + .3*xp*rnorm(n)
> plot(c(xp,x), c(yp[,1],y), pch='.') # 同时画出点和回归线
> lines(yp[,1]~xp) # 此线与上面的线是一个
> abline(r, col='red') # 此线与上面的线也是一个
> lines(xp, yp[,2], col='blue') # 下侧的区间线
> lines(xp, yp[,3], col='blue') # 上侧的区间线
> points(yr~xp, pch='.') # 散点
> points(y~x, col='orange', pch=16) # 散点
> points(y~x) # 同上的散点
> yp[1:10,] # 对 lm 的预测结果包括预测值, 上侧,下侧值
       fit       lwr      upr
1  0.9775679 0.6200168 1.335119
2  0.9772831 0.6197349 1.334831
3  0.9771233 0.6195768 1.334670
```

```
4  0.9765344 0.6189940 1.334075
5  0.9765169 0.6189766 1.334057
6  0.9762817 0.6187438 1.333820
7  0.9761578 0.6186212 1.333694
8  0.9761335 0.6185972 1.333670
9  0.9757138 0.6181818 1.333246
10 0.9755262 0.6179961 1.333056
```

24.5 决策树对回归的帮助

先看看决策树可能对回归有帮助. S-plus 是 tree, 但是 R 中推荐 rpart. 决策树似乎需要 factor(is.na(x1)) 对 x2.

```
library(rpart)
n <- 100
x1 <- rlnorm(n)
x2 <- rlnorm(n)
> r <- rpart(x1~x2)
> r
n=83 (17 observations deleted due to missing)

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

1) root 83 147.093300 1.3673040
   2) x2>=1.334109 20   4.386958 0.8184096 *
   3) x2< 1.334109 63 134.767700 1.5415560
   6) x2< 1.094479 53  94.693460 1.3232420
   12) x2>=0.9443127 8   1.695635 0.6366346 *
   13) x2< 0.9443127 45  88.555910 1.4453060
      26) x2< 0.8501714 38  47.850200 1.3111270
      52) x2< 0.3734032 14   4.773765 1.0510790 *
      53) x2>=0.3734032 24  41.577410 1.4628210
         106) x2>=0.5465227 16  21.178520 1.1821200 *
         107) x2< 0.5465227 8  16.616830 2.0242230 *
      27) x2>=0.8501714 7  36.307570 2.1737080 *
   7) x2>=1.094479 10  24.160210 2.6986210 *
```

```
> r <- rpart(factor(is.na(x1))~x2) # 看看 na 与其它值的分类
> r
n=92 (8 observations deleted due to missing)

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

1) root 92 9 FALSE (0.90217391 0.09782609) *
```

24.6 缺失数据的处理

简单的去除缺失并不太好. 当一个数据缺失, 需要同时去除其它的数据, 意味着一行数据被去除.

若数据缺失位置是随机的, 可以用平均值或中位数来代替.

但是很多时候, 缺失数据依情况而定. 例如, 收入调查往往缺失高收入的情况.

```
n <- 100
v <- .1
x1 <- rlnorm(n)
x2 <- rlnorm(n)
x3 <- rlnorm(n)
x4 <- x1 + x2 + x3 + v*rlnorm(n)
remove.higher.values <- function (x) {
n <- length(x)
ifelse( rbinom(n,1,(x-min(x))/(max(x)+1))==1 , NA, x)
}
x1 <- remove.higher.values(x1)
x2 <- remove.higher.values(x2)
x3 <- remove.higher.values(x3)
x4 <- remove.higher.values(x4)
m2 <- cbind(x1,x2,x3,x4)
pairs(m2, main="A few missing values")
```

24.7 极端值(outliers)–去除或缺失

可以手工去除, 或把它们当做缺失值处理. 参考 回归诊断-影响分析30.3

异常值:

x >上百分位数+1.5×（上百分位数-下百分位数）
x <下百分位数-1.5×（上百分位数-下百分位数）

极端异常值:

x >上百分位数+3×（上百分位数-下百分位数）
x <下百分位数-3×（上百分位数-下百分位数）

图来查看: boxplot, histogram, density, qqnorm 等

24.8 共线性的处理

24.8.1 例子–gls用法

下面是一个相关数据的例子

```
> n <- 100
> x <- runif(n)
> b <- rep(NA,n)
> b[1] <- 0
> for (i in 2:n) {
+   b[i] <- b[i-1] + .1*rnorm(1)
+ } # b 自身相关
> y <- 1-2*x+b[1:n]
```

```
> cor(x,y) # xy的相关系数
[1] -0.847911
> plot(x,y) # 绘图查看
> r <- lm(y~x)
> r

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)          x
   1.241      -1.790

> plot(r$res) # 查看残差是相关的
> cor.test(r$res[1:(n-1)], r$res[2:n]) # 检验残差的相关性

       Pearson's product-moment correlation

data:  r$res[1:(n - 1)] and r$res[2:n]
t = 26.2987, df = 97, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
0.9066943 0.9569760
sample estimates:
   cor
0.936483
```

下面是一个不相关的例子

```
> n <- 100
> x <- runif(n)
> b <- .1*rnorm(n+1) # b 自身不相关
> y <- 1-2*x+b[1:n] # xy不相关
> r <- lm(y~x)$res
> cor.test(r[1:(n-1)], r[2:n])

       Pearson's product-moment correlation

data:  r[1:(n - 1)] and r[2:n]
```

```
t = 0.0748, df = 97, p-value = 0.9405
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
-0.1901025  0.2046993
sample estimates:
       cor
0.007594353
```

在这种情况下, 可以使用 generalized least squares, 即 gls, 包含在 nlme 包里. 其中的 AR1 模型(自回归1模型)假设两个邻近的 error 是相关的.

$$e_{i+1} = r * e_i + f_i$$

其中 r 为 AR1 的系数, f_i 为独立变量.

```
> n <- 100
> x <- rnorm(n)
> e <- vector()
> e <- append(e, rnorm(1))
> for (i in 2:n) {
+   e <- append(e, .6 * e[i-1] + rnorm(1) )
+ } # e 为自相关的
> y <- 1 - 2*x + e
> i <- 1:n
> plot(y~x)

> library(nlme)
> g <- gls(y~x, correlation = corAR1(form= ~i))
# 绘图查看与 lm 的区别. 此处区别不大
> plot(y~x)
> abline(lm(y~x))
> abline(g, col='red')

> summary(g)
Generalized least squares fit by REML
Model: y ~ x
Data: NULL
   AIC     BIC   logLik
```

```
294.0995 304.4394 -143.0498

Correlation Structure: AR(1)
Formula: ~i
Parameter estimate(s):
    Phi
0.5901658

Coefficients:
              Value  Std.Error   t-value p-value
(Intercept)  0.7199012 0.24061996   2.99186  0.0035
x           -2.0964824 0.09715024 -21.57980  0.0000

Correlation:
(Intr)
x -0.003

Standardized residuals:
       Min         Q1        Med         Q3        Max
-2.754014254 -0.626080011 -0.001852298  0.780250206  1.672214722
```

24.8.2 多个线性相关的处理

数据中可能有多个变量相关,例如 X3 = X1+X2. 可以把每个变量(X_k)同其它变量($X_i's$)做回归, 然后检测R^2: 如果比较大(大于0.1), 则 Xk 可以表达为其它变量的线性组合.

为解决此问题, 我们可以去除强相关的变量, 但是小心你的解释会有问题. 很多时候相关是由于变量多而样本量少造成的, 此时可以增加样本量. 另外可以借助其它分析方法, 例如 “ridge regression” or SVM

另外, chapter 38 section 38.6 主成分回归也可以有效解决多个变量相关的问题, chapter 典型相关分析对多变量的相关的分析也很好

```
> n <- 100
```

```
> x <- rnorm(n) # x 为随机的
> x1 <- x+rnorm(n) # x1 由 x 而来, 与 x 相关
> x2 <- x+rnorm(n) # x2 由 x 而来, 与 x 相关
> x3 <- rnorm(n) # x3 也为随机的
> y <- x+x3
# 下面是它们表现出的关系. 注意 x1, x2 都与 x 相关, x3 随机
> summary(lm(x1~x2+x3))$r.squared # x1 与 x2,x3 线性相关
[1] 0.1726021
> summary(lm(x2~x1+x3))$r.squared # x2 与 x1,x3 线性相关
[1] 0.1337601
> summary(lm(x3~x1+x2))$r.squared # x3 独立于 x1,x2
[1] 0.07304001
# 可以推测, y 与 x,x1,x2,x3都相关
> summary(lm(y~x1))$r.squared
[1] 0.3182037
> summary(lm(y~x3))$r.squared
[1] 0.6227284
> summary(lm(y~x2))$r.squared
[1] 0.3027603
```

我们还可以查看估计系数的相关矩阵. 下面可以看出, x1 与 x3 是相关的. 注意系数的相关矩阵与变量间的相关系数不同.

```
> n <- 100
> v <- .1
> x <- rnorm(n)
> x1 <- x + v*rnorm(n)
> x2 <- rnorm(n)
> x3 <- x + v*rnorm(n)
> y <- x1+x2-x3 + rnorm(n)
> summary(lm(y~x1+x2+x3), correlation=T)$correlation
             (Intercept)         x1         x2         x3
(Intercept)  1.00000000 -0.08185033 -0.06912212  0.0907740
x1          -0.08185033  1.00000000 -0.20227089 -0.9903370
x2          -0.06912212 -0.20227089  1.00000000  0.2171248
x3           0.09077400 -0.99033697  0.21712483  1.0000000
> cor(cbind(x1,x2,x3)) # 变量间的相关系数
             x1         x2         x3
x1  1.00000000 -0.09422705  0.9900126
```

```
x2 -0.09422705  1.00000000 -0.1237598
x3  0.99001258 -0.12375983  1.0000000
# 下面画一个图使用圆圈的大小来表示相关系数的大小. 注意 cex 参数的用法
> m <- summary(lm(y~x1+x2+x3), correlation=T)$correlation
> m
            (Intercept)         x1         x2         x3
(Intercept)  1.00000000 -0.08185033 -0.06912212  0.0907740
x1          -0.08185033  1.00000000 -0.20227089 -0.9903370
x2          -0.06912212 -0.20227089  1.00000000  0.2171248
x3           0.09077400 -0.99033697  0.21712483  1.0000000
> class(m)
[1] "matrix"
> plot(col(m), row(m), cex=10*abs(m),
+     xlim=,
+     ylim=c(0, dim(m)[1]+1),
+ )
> col(m)
   [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    1    2    3    4
[3,]    1    2    3    4
[4,]    1    2    3    4
> row(m)
   [,1] [,2] [,3] [,4]
[1,]    1    1    1    1
[2,]    2    2    2    2
[3,]    3    3    3    3
[4,]    4    4    4    4
```

24.9 t检验和ANOVA的关系

两个组均值的比较使用t检验.

方差分析实际上是t检验的拓广. 多于两个组的时候就使用方差分析(analysis of variance, ANOVA).

24.10 什么时候使用协方差分析

(一) 协方差分析基本思想

不论是单因素方差分析还是多因素方差分析，控制因素都是可控的，其各个水平可以通过人为的努力得到控制和确定。

但在许多实际问题中，有些控制因素很难人为控制，但它们的不同水平确实对观测变量产生了较为显著的影响。

例如，在研究农作物产量问题时，如果仅考察不同施肥量、品种对农作物产量的影响，不考虑不同地块等因素而进行方差分析，显然是不全面的。因为事实上有些地块可能有利于农作物的生长，而另一些却不利于农作物的生长。不考虑这些因素进行分析可能会导致：即使不同的施肥量、不同品种农作物产量没有产生显著影响，但分析的结论却可能相反。

再例如，分析不同的饲料对生猪增重是否产生显著差异。如果单纯分析饲料的作用，而不考虑生猪各自不同的身体条件（如初始体重不同），那么得出的结论很可能是不准确的。因为体重增重的幅度在一定程度上是包含诸如初始体重等其他因素的影响的。

（二）协方差分析的原理

协方差分析将那些人为很难控制的控制因素作为协变量，并在排除协变量对观测变量影响的条件下，分析控制变量（可控）对观测变量的作用，从而更加准确地对控制因素进行评价。

协方差分析仍然沿承方差分析的基本思想，并在分析观测变量变差时，考虑了协变量的影响，人为观测变量的变动受四个方面的影响：即控制变量的独立作用、控制变量的交互作用、协变量的作用和随机因素的作用，并在扣除协变量的影响后，再分析控制变量的影响。

方差分析中的原假设是：协变量对观测变量的线性影响是不显著的；在协变量影响扣除的条件下，控制变量各水

平下观测变量的总体均值无显著差异，控制变量各水平对观测变量的效应同时为零。检验统计量仍采用F统计量，它们是各均方与随机因素引起的均方比。

（三）协方差分析的应用举例

为研究三种不同饲料对生猪体重增加的影响，将生猪随机分成三组各喂养不同的饲料，得到体重增加的数据。由于生猪体重的增加理论上会受到猪自身身体条件的影响，于是收集生猪喂养前体重的数据，作为自身身体条件的测量指标。

Chapter 25

R的统计模型概述

这些统计模型在较复杂的分析, 特别是回归和方差分析中应用广泛, 但是在其它例如因子分析等也会使用这里的模型.

R基本的屏幕输出是简洁的, 因此用户需要调用一些辅助函数来提取细节的结果信息。也就是说, 经常会联合使用多个函数来得到更全面详细的结果. 例如, 方差分析中一般会这样使用回归与方差分析函数lm()与anova(), 并进一步使用summary()函数来取得其详细的输出.

```
anova(lm(data~group))
summary(anova(lm(data~group)))
```

25.1 公式

假定y, x, x0, x1, x2, . . . 是数值变量, X是一个矩阵, 而A, B, C, . . . 是因子。下面的例子中, 左边给出公式, 右边给出该公式的统计模型的描述. 下面是一些例子

- `y~x`
  `y~1+x`

二者都反映了y 对x 的简单线性模型。第一个公式包含了一个隐式的截距项，而第二个则是一个显式的截距项。

- `y ~ 0+x`
  `y ~ -1 + x`
  `y ~ x-1`

  y 对x 过原点的简单线性模型(也就是说，没有截距项)。

- `log(y) ~ x1 + x2`

  y 的变换形式log(y) 对x1 和x2 进行的多重回归(有一个隐式的截距项)。

- `y ~ poly(x,2)`
  `y ~ 1 + x + I(x^2)`

  y 对x 的二次多项式回归。第一种是正交多项式（orthogonal polynomial），第二种则显式地注明各项的幂次。

- `y~A`

  y 的单因素方差分析模型，类别由A 决定。

- `y~A+x`

  y 的单因素协方差分析模型，类别由A 决定，协方差项为x。

- `y ~ A*B`
  `y ~ A + B + A:B`
  `y ~ B %in% A`
  `y ~ A|B`

  y 对A 和B 的非可加两因子方差分析模型（two factor non-additive model）。前两个公式表示相同的交叉分类设计（crossed classification），后两个公式表示相同的嵌套分类设计(nested classification)。抽象一点说，这四个公式指明同一个模型子空间。

- `y ~ (A + B + C)^2`
  `y ~ A*B*C - A:B:C`

  三因子实验。该模型包括一个主效应（main efects）和两个因子的交互效应（interactions）。这两个公式等价。

- `y~A*x`
  `y ~ A|x`
  `y ~ A|(1 + x) - 1`

  在A的各个水平独立拟合y对x的简单线性回归。三个公式的编码不一样。最后一个公式会对A各个水平分别估计截距项和斜率项的。

- `y ~ A*B + Error(C)`

  一个实验设计有两个处理因素A和B以及因子C决定的误差分层（error strata）。如在裂区实验设计（split plot experiment）中，所有区组（还包括子区组）都由因子C决定的。

25.2 符号总结

- `a+b`

  a和b的相加效应

- `a-b`

  包括a但排除b项。

- `X`

  如果X是一个矩阵, 这将反映各列的相加效应, 即 $X[,1]+X[,2]+...+X[,ncol(X)]$; 还可以通过索引向量选择特定列进行分析(如, $X[,2:4]$)

- `a:b`

  a 和b 的交互效应. a b 的张量积（tensor product）。如果两项都是因子，那么将产生“子类”因子(subclasses factor, 即因子交互作用)。

- `a*b`
  (等价于`a+b+a:b`)

  相加和交互效应

- `poly(a, n)`

  a的n价多项式

- `^n`
  `(a+b+c)^2` 等价于 `a+b+c+a:b+a:c+b:c`

  a+b 包含所有的直到n阶的交互作用

- `b %in% a`
  `a+a:b`
  `a|b`

  b 和 a 的嵌套分类设计

- `(a+b+c)^2-a:b`
  `a+b+c+a:c+b:c`

  去掉因子b的影响, 如:

- `y~x-1`
  `y~x+0`
  `0+y~x`

  表示通过原点的线性回归(等价于)

- `y~1`

  拟合一个没有因子影响的模型(仅仅是截距)

- `offset(...)`

  向模型中增加一个影响因子但不估计任何参数(如, offset(3*x))

注意，在常常用来封装函数参数的括弧中的操作符按普通的四则运算法则解释。I() 是一个恒等函数（identity function），它使得常规的算术运算符可以用在模型公式中。为了可以在公式中使用常规的运算符,

例如

```
y~x1+x2
```

表示模型

```
y = \beta_1 x1 + \beta_2 x2 + \alpha
```

而不是

```
y = \beta(x1 +  x2) + \alpha
```

公式

```
y~I(x1+x2)
```

就表示

```
y = \beta(x1 +  x2) + \alpha
```

还要特别注意模型公式仅仅指定了模型矩阵的列项，暗含了对参数项的指定。在某些情况下可能不是这样，如非线性模型的参数指定。

尽管细节是复杂的，R里面的模型公式在要求不是太离谱的情况下可以产生统计专家所期望的各种模型。提供模型公式的各种扩展特性是让R更灵活。例如，利用关联项而非主要效应的模型拟合常常会产生令人惊讶的结果，不过这些仅仅为统计专家们设计的。

25.3 注意: 添加factor

25.4 LRT

LRT(likelihood ratio test)是一个检测模型参数有关假设的强大的且通用的方法，但是限于似然法的范畴。它比较两个嵌套的假设。换句话说，一个假设是另外一个的特例。例如零假设为H_0，有p_0个参数，备则假设为H_1，是零假设的拓展，有p_1个参数

25.5 AIC(赤池信息量)准则

AIC准则即赤池信息量准则(Akaike' information criterion，AIC),是由赤池弘次(H. Akaike)在研究信息论特别是在解决时间序列定阶问题时提出来的。这是一个在统计分析特别是在统计模型的选择中有着广泛应用的准则。其显著特点之一是“吝啬原理(principle of parsimony)”的具体化。

定义为

AIC = - 2ln(模型的极大似然函数) + 2(模型的独立参数个数)

赤池建议,当欲从一组可供选择的模型中选择一个最佳模型时,AIC为最小的模型是最佳的。当两个模型之间存在着相当大的差异时,这个差异在右边第一项得到表现;而当两个模型间的差异几乎没有时,则第二项起作用,从而参数个数小的模型是好的模型.

25.6 BIC(贝叶斯信息量)准则

LRT和AIC准则经常会倾向于现在复杂的，参数多的模型，拒绝简单模型。贝叶斯信息量准则基于贝叶斯理论，对参数多的模型惩罚更严厉。定义为

$$BIC = -2l + p\log(n)$$

n为样本数，p为参数个数，l为对数极大似然值。同样BIC小的模型被认为比较好。

25.7 一些用于某些特殊回归和数据分析问题的工具

关于R中的回归函数的总结,参考R网站的文章《R FUNCTIONS FOR REGRESSION ANALYSIS》

关于非参数回归，参考R网站的文章《Nonparametric Regression》非参数回归包括局部多项式回归,平滑曲线简单回归,一般广义非参数回归(局部似然估计),累加模型等.

我们简单提一下R里面某些用于某些特殊回归和数据分析问题的工具.

- 混合模型（Mixed models）。用户捐献包nlme里面提供了函数lme() 和nlme()。这些函数可以用于混合效应模型（mixed-effects models）的线性和非线性回归。也就是说在线性和非线性回归中，一些系数受随机因素的影响。这些函数需要充分利用公式来描述模型。

- 局部近似回归(Local approximating regressions)。函数loess() 利用局部加权回归进行一个非参数回归。这种回归对显示一组凌乱数据的趋势和描述大数据集的整体情况非常有用。

  函数loess 和投影跟踪回归（projection pursuit regression）的代码一起放在标准包stats 中。

- 稳健回归(Robust regression)。有多个函数可以用于拟合回归模型，同时尽量不受数据中极端值的影响。在推荐包MASS 中的函数lqs为高稳健性的拟合提供了最新的算法。另外，稳健性较低但统计学上高效的方法同样可以在包MASS 中得到，如函数rlm。

- 累加模型(Additive models)。这种技术期望可以通过决定变量的平滑累加函数（smooth additive function）构建回归函数。一般来说，每个决定变量都有一个平滑累加函数。用户捐献的包acepack 里面的函数avas 和ace 以及包mda 里面的函数bruto 和mars 为这种技术提供了一些例子。这种技术的一个扩充是用户捐献包gam 和mgcv 里面实现的广义累加模型。

- 树型模型(Tree-based models)。除了利用外在的全局线性模型预测和解释数据，还可以利用树型模型递归地在决定性变量的判断点上将数据的分叉分开。这样做会把数据最终分成几个不同组，使得组内尽可能相似而组间尽可能差异。这样常常会得到一些其他数据分析方法不能产生的的信息。模型可以用一般的线性模型形式指定。该模型拟合函数是tree()，而且许多泛型函数，如plot() 和text() 都可以很好的用于树型模型拟合结果的图形显示。R 里面的树型模型函数可以通过用户捐献的包rpart 和tree 得到。

25.8 最大变量数

lm()函数的最大变量数可能为48个, 经测试, 超过48, 后面变量的系数显示为NA

Chapter 26

方差分析(ANOVA)

参考文献 [21] chapter 7.

参考文献 [14] chapter 12.

参考文献 [43] chapter 16.

参考百度百科介绍

http://baike.baidu.com/view/786804.htm?fr=ala0_1

26.1 多重比较的条件及检验

26.1.1 条件

1. 各组方差齐性, 即所有 i, j $\varepsilon_{i,j}$ 有相同的 σ^2.
2. 总体平均数为 0, 使样本平均数为总体平均数的无偏估计.
3. 服从正态分布. 这个要求对假设检验是必需的, 对参数估计不一定需要.

26.1.2 误差的正态性检验

Kolmogorov-Smirnov Test, 可以检验各种分布. 或专门检验正态分布的 shapiro.test(). 参考第 49 章

26.1.3 方差齐性检验

两个样本可以使用F检验. 多于两个使用Bartlett检验.

两个非正态样本使用 ansari.test 或 mood.test , 它们是非参数检验. 多于两个非正态样本参考 fligner.test.

bartlett.test 有两种用法.

```
> bartlett.test(list(rnorm(100),rnorm(100)+1,rnorm(100)+2))

        Bartlett test of homogeneity of variances

data:  list(rnorm(100), rnorm(100) + 1, rnorm(100) + 2)
Bartlett's K-squared = 2.7132, df = 2, p-value = 0.2575

> bartlett.test(c(rnorm(100),exp(rnorm(100))+1,rnorm(100)+2),
    g=c(rep(1,100),rep(2,100),rep(3,100)))

        Bartlett test of homogeneity of variances

data:  c(rnorm(100), exp(rnorm(100)) + 1, rnorm(100) + 2) and c(rep(1, 100), rep(2,
Bartlett's K-squared = 141.2364, df = 2, p-value < 2.2e-16
```

固定效应模型某个因素的水平是固定的, 例如, 死亡原因中的疾病种类

26.2 单因素方差分析-固定效应模型

单因素方差分析, 也叫做单因素方差分析(one-way nalysis of variance). 目的是比较多个均值是否相等.

26.2.1 数据描述

y_1	$\cdots$	y_k
y_{11}	$\cdots$	y_{k1}
$\cdots$	$\cdots$	$\cdots$
y_{1n_1}	$\cdots$	y_{kn_k}
$\bar{y}_1$	$\cdots$	$\bar{y}_k$

记总平均值为$\bar{y}$

26.2.2 模型

如果 Y 依赖于 X, 例如象下面 $Y = a0+a1*(X == 1)+a2*(X == 2) + a3 * (X == 3) + a4 * (X == 4)$

与 $Y = b0$ 比较. 即

$$y_{ij} = \mu + \alpha_i + e_{ij}$$

R的公式可以这样

```
y ~ x
```

26.2.3 平方和的分解

$$\sum_{i=1}^{k}\sum j=1^{n_i}(y_{ij}-\bar{y})^2 \;=\; \sum_{i=1}^{k}\sum j=1^{n_i}(y_{ij}-\bar{y}_i)^2+\sum_{i=1}^{k}\sum j=1^{n_i}(\bar{y}_i-\bar{y})^2$$

$$SS_T = SS_B + SS_W$$

其中

- SS_T: 总平方和(total)
- SS_B: 组间平方和(between)
- SS_W: 组内平方和(within)
- SS_B的自由度为 $k-1$
- SS_W的自由度为 $n-k$

那么组间平均平方和为

$$MS_B = SS_B/(k-1)$$

那么组内平均平方和为

$$MS_W = SS_W/(n-k)$$

26.2.4 方差分析表

Table 26.1: 单因素方差分析表

方差来源	自由度	平方和	均方	F值	p值
因素	$k-1$	SS_B	$MS_B = SS_B/(k-1)$	$F = MS_B/MS_W$	p
误差	$n-k$	SS_W	$MS_W = SS_W/(n-k)$		
总和	$n-1$	$SS_T = SS_B + SS_W$			

26.2.5 F检验

原理是: 如果组间的差异大于组内的差异, 拒绝零假设, 否则接受零假设.

注意: 不能够得知哪个组的均值有显著差异, 若需进一步知道, 使用多重检验.

检验为

$$H_0: 所有\alpha_i = 0 \quad vs. \quad H_1: 至少一个\alpha_i \neq 0$$

检验统计量为

$$F = MS_B/MS_W \sim F_{k-1,n-k} \quad (H_0下)$$

判断

$$F > F_{k-1,n-k,1-\alpha}, 拒绝零假设$$
$$F <= F_{k-1,n-k,1-\alpha}, 接受零假设$$

p-值为

$$p - value = P(F_{k-1,n-k} > F)$$

26.2.6 例子

使用 anova() 和 aov() 函数

下面是一个例子. y 被 x 分为 3 组, 比较 3 组 y 均值是否相同. 其中第一行为组间变量的信息, Df 为自由度, Sum Sq 为平方和SS, Mean Sq 为平均平方和SS

```
# 数据
n <- 30
x <- sample(LETTERS[1:3],n,replace=T, p=c(3,2,1)/6)
x <- factor(x)
y <- rnorm(n)
```

```
# 绘图
plot(y ~ x,
    col = 'pink',
    xlab = "", ylab = "",
    main = "Simple anova: y ~ x")

# F值小(p值大), 说明均值差异不显著.
# SS_B=0.3478, SS_W=28.368, df_B=2, df_W=27
# MS_B=0.1739, MS_W=1.0507
# F=MS_B/MS_W=0.1655
# p值=P(F_{2,27}>F)=0.8483
> anova(lm(y~x)) # anova 必须与lm联合使用.
Analysis of Variance Table

Response: y
          Df  Sum Sq Mean Sq F value Pr(>F)
x          2  0.3478  0.1739  0.1655 0.8483
Residuals 27 28.3680  1.0507

# summary(aov(y~x)) 与 anova(lm(y~x)) 的结果是一样的
> summary(aov(y~x)) # p值大, 说明均值差异不显著.
            Df  Sum Sq Mean Sq F value Pr(>F)
x            2  0.3478  0.1739  0.1655 0.8483
Residuals   27 28.3680  1.0507
```

下面是另外一个例子

```
x=rnorm(100)
y=rnorm(100)+1
z=rnorm(100)+2
data=c(x,y,z)
g=c(rep(0,100),rep(1,100),rep(2,100)) # 分组信息
> boxplot(data~g) # 画图看看
>
> bartlett.test(data~g) # 方差齐性检验

        Bartlett test of homogeneity of variances
```

```
data:  data by g
Bartlett's K-squared = 4.4351, df = 2, p-value = 0.1089

> summary(anova(lm(data~g)))
      Df           Sum Sq        Mean Sq          F value
 Min.   :  1.00   Min.   :195.0   Min.   :  1.059   Min.   :184.2
 1st Qu.: 75.25   1st Qu.:225.2   1st Qu.: 49.549   1st Qu.:184.2
 Median :149.50   Median :255.3   Median : 98.040   Median :184.2
 Mean   :149.50   Mean   :255.3   Mean   : 98.040   Mean   :184.2
 3rd Qu.:223.75   3rd Qu.:285.4   3rd Qu.:146.530   3rd Qu.:184.2
 Max.   :298.00   Max.   :315.6   Max.   :195.020   Max.   :184.2
                                                    NA's   :  1.0
    Pr(>F)
 Min.   :5.434e-33
 1st Qu.:5.434e-33
 Median :5.434e-33
 Mean   :5.434e-33
 3rd Qu.:5.434e-33
 Max.   :5.434e-33
 NA's   :1.000e+00
> summary(aov(data~g))
            Df  Sum Sq Mean Sq F value    Pr(>F)
g            1 195.020 195.020  184.15 < 2.2e-16 ***
Residuals  298 315.589   1.059
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

26.2.7 单向ANOVA与多重回归的关系

请参考文献 [14] 12.5.2, 及单因素协方差分析26.4

使用虚变量来表示各组的变量. 单向ANOVA与多重回归最后计算的结果是相同的

下面是例子

```
res<-lm(data~factor(g))
```

```
> res

Call:
lm(formula = data ~ factor(g))

Coefficients:
(Intercept)  factor(g)1  factor(g)2
   -0.2210      1.1945      2.3326

# 产生虚变量
> k= diag(length(coef(res)))[-1,]
> k
     [,1] [,2] [,3]
[1,]    0    1    0
[2,]    0    0    1
# 结果与线性模型一样
> library(multcomp)
> glht(res, linfct = k)

        General Linear Hypotheses

Linear Hypotheses:
      Estimate
1 == 0    1.195
2 == 0    2.333
```

26.3 均值的多重比较

当F检验拒绝零假设, 我们需要找到哪两个组的均值不同, 需要使用多重比较.

方法比较多, 原理都是调整临界值或置信水平, 减少假阳性或假阴性.

26.3.1 Studentized range (distribution)

http://en.wikipedia.org/wiki/Studentized_range

In statistics, the studentized range computed from a list x1, ..., xn of numbers is

$$\frac{\max\{\, x_1, \ldots, x_n \,\} - \min\{\, x_1, \ldots, x_n \,\}}{s},$$

where

$$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{x})^2,$$

is the sample variance and

$$\overline{x} = \frac{x_1 + \cdots + x_n}{n}$$

is the sample mean.

Generally, studentized means adjusted by dividing by an estimate of a population standard deviation; see also studentized residual. The concept is named after William Sealey Gosset, who wrote under the pseudonym "Student". The fact that the variance is a sample variance rather than the population variance, and thus something that differs from one random sample to the next, is essential to the definition, and complicates the problem of finding the probability distribution of any statistic that is studentized.

If X1, ..., Xn are independent identically distributed random variables that are normally distributed, the probability distribution of their studentized range is what is usually called the studentized range distribution.

This probability distribution is the same regardless of the expected value and standard deviation of the normal distribution from which the sample is drawn. This probability distribution has applications to hypothesis testing and multiple comparisons.

26.3.2 各种方法介绍

LSD 最小显著差数法（least significant difference），简称LSD法

LSR 最小显著极差法(least significant range)，简称LSR法, 基于极差分布(q 分布)的检验

(1) 新复极差测验（SSR法）

这种方法由邓肯（D.B.Duncan）氏于1955年首创，又称SSR（shortest significant ranges）测验，SSR法对不同秩次距的平均数极差采用不同的显著尺度，因在同一总体抽样时，平均数的极差值将随k的增加而增大，改进了LSD法测验中不够合理部分。SSR法是一种极差测验

（2）新复极差测验（q法）

这种方法与新复极差法相似，只是在计算最小显著极差时，是查附表7（q值表）。

LSRα,k,df＝qα,k,df×SE

Single-step procedures

* Tukey–Kramer method (Tukey's HSD) (1951) based on a studentized range distribution q * Scheffe method (1953)

Multi-step procedures based on Studentized range statistic

* Duncan's new multiple range test (1955) * The Nemenyi test is similar to Tukey's range test in ANOVA.

* The Bonferroni–Dunn test allows comparisons, controlling the family-wise error rate.[vague] * Student Newman-Keuls post-hoc ANOVA analysis

26.3.3 各种方法介绍2

http://www.jerrydallal.com/LHSP/mc.htm

http://www.statsdirect.com/help/analysis_of_variance/multi.htm

The following is a decision tree for selecting a multiple contrast method:

* pairwise o equal groups sizes: Tukey o unequal group sizes: Tukey-Kramer or Scheffé * not pairwise o with a control: Dunnett o planned: Bonferroni o not planned: Scheffé

Note that Bonferroni and Scheffé methods are completely general; they can be used for unplanned (a posteriori) or planned (a priori) multiple comparisons.

26.3.4 LSD法(最小显著性差异法)

LSD法(least significant difference), 称为最小显著性差异法.

原理为: 对指定两个组的数据进行t检验, 但是对方差的估计是利用全体数据的均方MS_W, 故检验统计量t的自由度变大.

注意: k个组的方差齐性检验相等时才能利用全体数据的均方MS_W, 否则只能做普通的t检验.

检验假设

$$H_0 : \alpha_i = \alpha_j \quad vs. \quad H_1 : \alpha_i \neq \alpha_j$$

检验统计量

$$t = \frac{\bar{y}_i - \bar{y}_j}{\sqrt{MS_W(\frac{1}{n_i} + \frac{1}{n_j})}} \sim t_{n-k}$$

对于双侧置信水平α有

$$|t| > t_{n-k,1-\alpha/2}, \quad reject \ H_0$$
$$|t| <= t_{n-k,1-\alpha/2}, \quad accept \ H_0$$

p值为

$$p = 2 * P(t_{n-k} > |t|)$$

对于只做一对比较, 显著性水平$\alpha = 0.05$是合适的, 但是如果做多对比较, 那么I型错误的概率会增加. 即假阳性增加. 下面是显著性水平增加的情况([43] 16.1.5)

Table 26.2: 多重比较I型错误概率(假阳性增加)

组数(k)	2	3	4	5	6
正常I型错误概率	5%	5%	5%	5%	5%
多重比较	5%	12.2%	20.3%	28.6%	36.6%

26.3.5 Bonferroni法-LSD法的修正

由于LSD法的假阳性增加的问题, 需要修正其置信水平α, 或等价的修正其检验统计量的阈值.

显著性水平的修正为

$$\alpha^* = \alpha / \binom{k}{2}$$

下面是其理由. 如果有k个组, 两两比较的数目为$c = \binom{k}{2}$. 记E为至少一个两组比较是显著的这一事件, $P(E)$有时候称为"实验性I型误差"(experiment-wise type I error). 需要决定α^*使得$P(E) = \alpha$

如果两两比较是独立的, 有

$$P(\bar{E}) = 1 - \alpha = (1 - \alpha^*)^c$$

当α^*很小的时候有[1]

$$1 - \alpha = (1 - \alpha^*)^c \approx 1 - c\alpha^* \Longrightarrow \alpha^* = \alpha / \binom{k}{2}$$

[1]展开略去高阶项

多重比较比普通的LSD法要严格, 即LSD显著的在多重比较中可能不显著.

应该指出, 通常的两两比较不会都是独立的, 故α^*的合适值一般要大于$\alpha/\binom{k}{2}$, 所以Bonferroni法是保守的.

一般, 在事先没有计划要对某些特定的组比较且k比较大的时候, 使用Bonferroni法, 在组数较小且仅仅对某些特定的组比较的时候(通常称草案分析)建议使用LSD法.

26.3.6 线性约束

参考文献 [14] 12.4.2

比LSD法更一般的是选取a个组和另外的b个组做比较.

下面是一个肺病的例子一般人群中, 轻度, 中度, 重度吸烟

组号	吸烟情况	样本量	肺功能(用力中期呼出量,FEF)
1	非吸烟	200	3.78
2	被动吸烟	200	3.30
3	非吸入吸烟(不把烟吸入)	50	3.32
4	轻度吸烟(1-10支/天)	200	3.23
5	中度吸烟(11-39支/天)	200	2.73
6	重度吸烟(40支以上/天)	200	2.59

的比例大概是$10\%, 70\%, 20\%$.

我们想比较吸烟的(包括轻度, 中度, 重度)和非吸烟的人群的肺功能差异. 对于此问题, 应该使用线性约束的估计及检验.

线性约束(linaer contrast): 指对某些组的均值做线性组合, 其系数之和应该为0. 即

$$\begin{aligned} L &= \sum_{i=1}^{k} c_i \bar{y}_i \\ \sum_{i=1}^{k} c_i &= 0 \end{aligned}$$

注意两个组之间的比较可以算做特例.

例如, 比较非吸烟的和被动吸烟的肺功能, 线性约束可以写为

$$L = \bar{y}_1 - \bar{y}_2, \quad \text{其中} \quad c_1 = 1, c_2 = -1$$

吸烟的(包括轻度, 中度, 重度)和非吸烟的人群的肺功能差异, 线性约束可以写为

$$L = \bar{y}_1 - 0.1\bar{y}_4 - 0.7\bar{y}_5 - 0.2\bar{y}_6$$

记μ_L为L的理论值, 即

$$\mu_l = c_1\alpha_1 + \cdots + c_k\alpha_k$$

因为$Var(\bar{y}_i) = MS_W/n_i$, 故

$$Var(L) = MS_W \sum_{i=1}^{k} c_i^2/n_i$$

那么假设检验为

$$H_0 : \mu_L = 0 \quad vs \quad H_1 : \mu_L \neq 0$$

检验统计量为

$$t = \frac{L}{\sqrt{Var(L)}} \sim t_{n-k}$$

对于双侧置信水平α有

$$\begin{aligned} |t| > t_{n-k,1-\alpha/2}, \quad & reject\ H_0 \\ |t| <= t_{n-k,1-\alpha/2}, \quad & accept\ H_0 \end{aligned}$$

p值为

$$p = 2 * P(t_{n-k} > |t|)$$

线性约束的其它用法: 当不同的组与某种特定的数量指标(例如, 药物剂量)对应时, 线性约束的系数可以取能够反映上述数量关系的值. 在不同组中样本量差别很大时, 特别有用. 因为小样本的组统计检验时常常出现不显著的结果, 但是其趋势常在某个方向上.

例如, 考察吸烟的(包括轻度, 中度, 重度)吸烟数量是否影响肺功能. 还要考察吸烟数量与肺功能的方向关系.

轻度吸烟, 我们取平均值$(1+10)/2 = 5.5$, 中度吸烟平均值$(11+39)/2 = 25$, 重度吸烟平均值取40代表(这是一个保守的估计), 检验

$$L = 5.5\bar{y}_4 + 25\bar{y}_5 + 40\bar{y}_6$$

为了使系数和为0, 将每个系数减去系数的平均值$(5.5+25+40)/3 = 23.5$, 约束变为

$$\begin{aligned} L &= (5.5-23.5)\bar{y}_4 + (25-23.5)\bar{y}_5 + (40-23.5)\bar{y}_6 \\ &= -18\bar{y}_4 + 1.5\bar{y}_5 + 16.5\bar{y}_6 \end{aligned}$$

这个约束表示: 3个组中每天吸烟量的增加数.

下面按照步骤检验即可. 设已知$MS_W = 0.636$, 那么

```
L=-18*3.23+1.5*2.73+16.5*2.59
s=sqrt(0.636*((-18)^2/200+1.5^2/200+16.5^2/200))
t=L/s
> t
[1] -8.198171
> pt(t,df=1044) # p值
[1] 3.552091e-16
```

26.3.7 scheffe法-线性约束的多重比较

如果某个线性约束不是事先计划好的, 那么应该使用线性约束的多重比较.

检验假设

$$H_0 : \mu_L = 0 \quad v.s. \quad H_1 : \mu_L \neq 0$$

此处

$$\begin{aligned} L &= sum_{i=1}^{k} c_i \bar{y}_i \\ \sum_{i=1}^{k} c_i &= 0 \\ \mu_L &= sum_{i=1}^{k} c_i \mu_i \end{aligned}$$

显著性水平为α

计算检验统计量

$$t = \frac{L}{\sqrt{Var(L)}} = \frac{L}{\sqrt{MS_W \sum_{i=1}^{k} c_i^2 / n_i}}$$

记临界值$a = \sqrt{(k-1)F_{k-1,n-k,1-\alpha}}$, 判断

$$\begin{aligned} |t| > a, &\quad reject\ H_0 \\ |t| <= a, &\quad accept\ H_0 \end{aligned}$$

scheffe法也可以用于两组之间的均值比较, 因为是线性约束的特例. 但是此情况下Bonferroni法要更可取, 因为当差异确实存在的时候, Bonferroni法比Scheffe法在显著性检验上更合适.

如果线性约束个数很少, 而且事先就指定要检验的约束, 则我们可以不使用线性约束的多重比较, 因为如果使用线性约束的多重比较, 在发现差异上就会比直接使用线性约束的功效小很多.

如果约束很多, 且不是在数据前指定检验, 则此scheffe法是合适的.

26.3.8 其它方法

Dunnett方法: 比较k个用药组与一个对照组的均值

Duncan法(Newman-Keuls检验): 多组均值的两两比较, 显著性的差别介于LSD与Tukey法之间.

Tukey法: 多组均值的两两比较, 比较严格.

26.3.9 p.adjust() 函数

p.adjust() 函数计算调整后的p值, 用法为

```
p.adjust(p, method = p.adjust.methods, n = length(p))

p.adjust.methods
# c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY",
#   "fdr", "none")
```

- 默认方法为"Holm"
- 参数p为一个p值向量
- n为比较的次数. 默认为p值的个数.

Bonferroni 法使用p值乘以比较的次数. "holm"法比Bonferroni法保守性稍微小一点. 前四个方法对阳性错误率(family wise error rate)的控制较严. 似乎没有理由使用非修正的Bonferroni法, 应该使用Holm法.

若假设检验是独立的, 或非负相关, 那么 Hochberg's and Hommel's methods 比较合适. Hommel方法比Hochberg方法要强, 但是差别很小, 且Hochberg方法计算速度快. "BH"法和"BY"法控制阴性率(false discovery rate)好一些,

下面是帮助的例子

```
> x <- rnorm(50, mean=c(rep(0,25),rep(3,25)))
>  p <- 2*pnorm( sort(-abs(x)))
>  round(p, 3)
 [1] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.001 0.001
[13] 0.002 0.002 0.002 0.002 0.003 0.003 0.005 0.006 0.012 0.020 0.023 0.035
[25] 0.048 0.096 0.103 0.108 0.141 0.191 0.208 0.220 0.261 0.288 0.333 0.399
[37] 0.409 0.452 0.496 0.572 0.577 0.581 0.588 0.598 0.646 0.744 0.776 0.846
[49] 0.868 0.985
> round(p.adjust(p), 3)
 [1] 0.000 0.000 0.001 0.002 0.003 0.005 0.007 0.007 0.013 0.019 0.053 0.054
[13] 0.064 0.067 0.070 0.077 0.087 0.087 0.165 0.195 0.346 0.594 0.632 0.951
[25] 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000
[37] 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000
[49] 1.000 1.000
>  round(p.adjust(p,"BH"), 3)
 [1] 0.000 0.000 0.000 0.000 0.001 0.001 0.001 0.001 0.002 0.002 0.006 0.006
[13] 0.006 0.006 0.006 0.007 0.007 0.007 0.014 0.016 0.027 0.047 0.049 0.073
[25] 0.095 0.184 0.190 0.193 0.242 0.318 0.335 0.343 0.395 0.423 0.475 0.552
[37] 0.552 0.594 0.636 0.679 0.679 0.679 0.679 0.679 0.718 0.809 0.826 0.882
[49] 0.886 0.985
```

26.3.10 pairwise.t.test()函数

计算多重比较, 使用p.adjust()里面的方法. 可以为"none".

```
x=rnorm(100)
y=rnorm(100)+1
z=rnorm(100)+2
data=c(x,y,z)
g=c(rep(0,100),rep(1,100),rep(2,100)) # 分组信息
> pairwise.t.test(data, g, p.adjust.method = "none")

        Pairwise comparisons using t tests with pooled SD

data:  data and g

  0       1
```

```
1 3.4e-12 -
2 < 2e-16 1.4e-09

P value adjustment method: none
> pairwise.t.test(data, g, p.adjust.method = "holm")

        Pairwise comparisons using t tests with pooled SD

data:  data and g

  0      1
1 6.7e-12 -
2 < 2e-16 1.4e-09

P value adjustment method: holm
```

26.3.11 TukyHSD法

http://en.wikipedia.org/wiki/Tukey

Assumptions of Tukey's test

1. The observations being tested are independent 2. The means are from normally distributed populations 3. There is equal variation across observations. (homoscedasticity)

计算Tukey Honest Significant Differences, 即计算置信水平下的均值差值的置信区间与p值.

```
# 使用上面的数据
> TukeyHSD(aov(data~factor(g)))
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = data ~ f)

$f
```

```
        diff      lwr     upr p adj
1-0 0.9355527 0.5922622 1.278843    0
2-0 1.9749450 1.6316544 2.318236    0
2-1 1.0393923 0.6961017 1.382683    0
```

26.3.12 S-N-K法(建议使用 Tukey test)

Newman-Keuls (also called Student-Newman-Keuls test)

Tukey test (also called Tukey-Kramer test)

算法请参考

`http://en.wikipedia.org/wiki/Post-hoc_analysis`

下面的解释来自 http://www.graphpad.com/faq/viewfaq.cfm?faq=1093

How do I decide between the Tukey and Newman-Keuls multiple comparison test? FAQ 1093

Both the Tukey test (also called Tukey-Kramer test) and the Newman-Keuls (also called Student-Newman-Keuls test) are used to compare all pairs of means following one-way ANOVA. Although these are called post tests, they can be performed regardless of the results of the overall ANOVA results.

The Newman-Keuls test has more power. This means it can find that a difference between two groups is 'statistically significant' in some cases where the Tukey test would conclude that the difference is 'not statistically significant'. But this extra power comes at a price. Although the whole point of multiple comparison post tests is to keep the chance of a Type I error in any comparison to be 5%, in fact the Newman-Keuls test doesn't do this1. In some cases, the chance of a Type I error can be greater than 5%. Another problem is because the Newman-Keuls test works in a sequential fashion, it can not produce 95% confidence intervals for each difference.

Because the Newman-Keuls test has two strikes against it (doesn't con-

trol error rate, doesn't generate confidence intervals) we recommend that you use the Tukey test instead.

1 MA Seaman, JR LEvin and RC Serlin, Psychological Bulletin 110:577-586, 1991.

26.4 单因素协方差分析(ANCOVA)

参考文献 [14] 12.5.3

参考 help(rp.ancova,pac="rpanel"), 交互单因素协方差分析

参考 help(sm.ancova,pac="sm"), 非参数单因素协方差分析

协方差分析将那些人为很难控制的控制因素作为协变量，并在排除协变量对观测变量影响的条件下，分析控制变量（可控）对观测变量的作用，从而更加准确地对控制因素进行评价。

在这里,我们想考察一个因素水平的差异是否对结果变量(正态分布)均值有显著影响,但是需要控制其它协变量(可以是连续,也可以是分类变量). ANCOVA(one way analysis-of-covariance model)是控制潜在的混杂变量的基础上去比较2组或多组的连续结果变量均值. 这个模型叫做单向协方差模型,也称作协方差分析模型(多重回归).

下面是y的单因素协方差分析公式，类别由A决定，协方差项为x。(统计模型一章有其它的模型25)

```
y~A+x
```

下面是一个虚拟的例子. 我们将年龄(age), 性别(sex)作为协变量, 考察用药与否(ctl)与血压(y)的关系. 其模型为

$$y = \alpha + \beta_1 ctl + \beta_2 sex + \beta_3 age + e$$

此模型考察的是控制年龄(age), 性别(sex)后用药与否(ctl)和血压

的关系. (注意我们将模型自变量的顺序改变后结果会不同, 有时候甚至相反)

可以看到, 控制age, sex后, ctl的影响是显著的, 对照(ctl=0)比用药(ctl=1)要低6.68个单位. 性别和年龄的影响是不显著的(p=0.72, p=0.97), 女性(sex=0) 比男性(sex=1)平均血压要低0.06个单位, 但是年龄每增加1,平均血压下降0.13个单位.

```
# 年龄
age=sample(c(10:20),100,replace=TRUE)
# 性别
sex=sample(c(1,2),100,replace=TRUE)
# 服药与否, 前50个未服药, 后50个服药
ctl=c(rep(0,50),rep(1,50))
# 血压, 假设服药组血压高
y=round(runif(100)*40+80,1); y[51:100]=y[51:100]+10

# 控制年龄(age), 性别(sex)后用药与否(ctl)和血压的关系.
# lm(y~ctl+(age+sex)) 与写法 lm(y~ctl+age+sex) 结果一样
> lm(y~ctl+(age+sex))

Call:
lm(formula = y ~ ctl + (age + sex))

Coefficients:
(Intercept)         ctl         age         sex
  105.92229     6.67843    -0.12829     0.06079

> summary(lm(y~ctl+(age+sex)))

Call:
lm(formula = y ~ ctl + (age + sex))

Residuals:
   Min     1Q  Median     3Q    Max
-23.559 -10.051   1.038  10.055  18.448

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 105.92229    6.32497  16.747  < 2e-16 ***
```

```
ctl          6.67843    2.32660   2.870  0.00504 **
age         -0.12829    0.35835  -0.358  0.72113
sex          0.06079    2.31394   0.026  0.97910
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.29 on 96 degrees of freedom
Multiple R-squared: 0.08218,    Adjusted R-squared: 0.0535
F-statistic: 2.865 on 3 and 96 DF,  p-value: 0.04066

# 这里给出了F值及其p值
> summary(aov(lm(y~ctl+(age+sex))))
          Df  Sum Sq Mean Sq F value   Pr(>F)
ctl        1  1079.1  1079.1  8.4673 0.004494 **
age        1    16.3    16.3  0.1276 0.721703
sex        1     0.1     0.1  0.0007 0.979097
Residuals  96 12234.8   127.4
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
# anova()函数的结果是一样的
> anova(lm(y~ctl2+age+sex))
Analysis of Variance Table

Response: y
         Df  Sum Sq Mean Sq F value   Pr(>F)
ctl2      1  1079.1  1079.1  8.4673 0.004494 **
age       1    16.3    16.3  0.1276 0.721703
sex       1     0.1     0.1  0.0007 0.979097
Residuals 96 12234.8   127.4
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

可以看到ctl(用药)的标记(0,1 还是 1,2)对分析结果无影响

```
> ctl2=ctl+1
> summary(lm(y~ctl2+age+sex))

Call:
```

```
lm(formula = y ~ ctl2 + age + sex)

Residuals:
   Min     1Q  Median     3Q    Max
-23.559 -10.051   1.038  10.055  18.448

Coefficients:
          Estimate Std. Error t value Pr(>|t|)
(Intercept) 99.24386    6.64577  14.933  < 2e-16 ***
ctl2         6.67843    2.32660   2.870  0.00504 **
age         -0.12829    0.35835  -0.358  0.72113
sex          0.06079    2.31394   0.026  0.97910
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.29 on 96 degrees of freedom
Multiple R-squared: 0.08218,    Adjusted R-squared: 0.0535
F-statistic: 2.865 on 3 and 96 DF,  p-value: 0.04066
```

改变顺序后结果相同(都是控制其它变量后的结果)

```
> lm(y~ctl+(age+sex))
Coefficients:
(Intercept)         ctl         age         sex
  105.92229     6.67843    -0.12829     0.06079

> lm(y~age + sex + ctl)

Call:
lm(formula = y ~ age + sex + ctl)

Coefficients:
(Intercept)         age         sex         ctl
  105.92229    -0.12829     0.06079     6.67843

> summary(lm(y~age + sex + ctl))

Call:
lm(formula = y ~ age + sex + ctl)
```

```
Residuals:
   Min     1Q  Median     3Q    Max
-23.559 -10.051   1.038  10.055  18.448

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 105.92229    6.32497  16.747  < 2e-16 ***
age          -0.12829    0.35835  -0.358  0.72113
sex           0.06079    2.31394   0.026  0.97910
ctl           6.67843    2.32660   2.870  0.00504 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.29 on 96 degrees of freedom
Multiple R-squared: 0.08218,    Adjusted R-squared: 0.0535
F-statistic: 2.865 on 3 and 96 DF,  p-value: 0.04066
```

26.5 两因素方差分析

多因素方差分析用来研究两个及两个以上控制变量是否对观测变量产生显著影响。这里，由于研究多个因素对观测变量的影响，因此称为多因素方差分析。多因素方差分析不仅能够分析多个因素对观测变量的独立影响，更能够分析多个控制因素的交互作用能否对观测变量的分布产生显著影响，进而最终找到利于观测变量的最优组合。

例如：分析不同品种、不同施肥量对农作物产量的影响时，可将农作物产量作为观测变量，品种和施肥量作为控制变量。利用多因素方差分析方法，研究不同品种、不同施肥量是如何影响农作物产量的，并进一步研究哪种品种与哪种水平的施肥量是提高农作物产量的最优组合。

两因素方差分析又称为 double anova, two-factor anova, two-way anova. 我们需要考察结果变量(正态分布)与两个因素的关系. (需要控制其它协变量的时候使用双向协方差分析)

26.5.1 无交互影响的双因素方差分析

如果根据经验或某种分析能够事先判断两因素之间没有交互影响,每组试验就不必重复.

```
因素 B
B1 B2 ...  Bn
------------------------------
因 A1 X11 X12 ...  X1n
素 A2 X21 X22 ... X2n
A .
Ar Xr1 Xr2 ... Xrn
------------------------------
```

离差平方和分解形式为

$$SS_T = SS_A + SS_B + SS_E$$

其中

$$\begin{aligned} SS_T &= \sum\sum(X_{ij} - \bar{X})^2 \\ SS_A &= \sum\sum(\bar{X}_{i.} - \bar{X})^2 = \sum n(\bar{X}_{i.} - \bar{X})^2 \\ SS_B &= \sum\sum(\bar{X}_{.j} - \bar{X})^2 = \sum r(\bar{X}_{.j} - \bar{X})^2 \\ SS_E &= \sum\sum(X_{ij} - \bar{X}_{i.} - \bar{X}_{.j} + \bar{X})^2 \end{aligned}$$

- SS_T 的自由度为 $n * r - 1$
- SS_A 的自由度为 $r - 1$
- SS_B 的自由度为 $n - 1$
- SS_E 的自由度为 $n * r - r - n - 1 = (r - 1) * (n - 1)$

对应的均方差为

- A: $MS_A = \frac{SS_A}{r-1}$
- B: $MS_B = \frac{SS_B}{n-1}$
- 误差 E: $MS_E = \frac{SS_E}{(r-1)*(n-1)}$

得到的因素A和B的影响是否显著的检验统计量为

$$F_A = \frac{MS_A}{MS_E} \sim F_{r-1,(r-1)(n-1)}$$

$$F_B = \frac{MS_B}{MS_E} \sim F_{n-1,(r-1)(n-1)}$$

26.5.2 有交互影响的双因素方差分析

如果根据经验或某种分析能够事先判断两因素之间没有交互影响,每组试验就不必重复.

```
因素 B
B1 B2 ...  Bn
---------------------------------------
因 A1 X111 X121 ...  X1n1
X112 X122 ... X1n2
. . ... .
X11m X12m X1nm

素 A2 X211 X221 ... X2n1
X212 X222 ... X2n2
. . ... .
X21m X22m X2nm

A .
Ar Xr11 Xr21 ... Xrn1
Xr12 Xr22 ... Xrn2
. . ... .
```

```
Xr1m Xr2m Xrnm

--------------------------------------
```

其中

$$\bar{X}_{ij.} = \frac{1}{m}\sum_{l=1}^{m} X_{ijl} \quad \bar{X}_{i..} = \frac{1}{nm}\sum_{j-1}^{n}\sum_{l=1}^{m} X_{ijl}$$

$$\bar{X}_{.j.} = \frac{1}{rm}\sum_{i=1}^{r}\sum_{l=1}^{m} X_{ijl} \qquad \bar{X} = \frac{1}{rnm}\sum\sum\sum X_{ijl}$$

离差平方和的分解形式

$$SS_T = SS_A + SS_B + SS_{AB} + SS_E$$

其中

$$\begin{aligned} SS_T &= \sum\sum\sum (X_{ijl} - \bar{X})^2 \\ SS_A &= nm\sum (\bar{X}_i - \bar{X})^2 \\ SS_B &= rm\sum (\bar{X}_{.j.} - \bar{X})^2 \\ SS_{AB} &= m\sum\sum (\bar{X}_{ij.} - \bar{X}_{i..} - \bar{X}_{.j.} + \bar{X})^2 \\ SS_E &= \sum\sum\sum (X_{ijl} - \bar{X}_{ij.})^2 \end{aligned}$$

自由度分别为 $rnm-1, r-1, n-1, (r-1)(n-1), rn(m-1)$

均方差为

- A: $MS_A = \frac{SS_A}{r-1}$
- B: $MS_B = \frac{SS_B}{n-1}$
- AB: $MS_{AB} = \frac{SS_{AB}}{(r-1)(n-1)}$

- 误差 E: $MS_E = \frac{SS_E}{rn(m-1)}$

检验统计量为

$$F_A = \frac{MS_A}{MS_E} \sim F_{r-1,rnm-rn}$$

$$F_B = \frac{MS_B}{MS_E} \sim F_{n-1,rnm-rn}$$

$$F_{AB} = \frac{MS_{AB}}{MS_E} \sim F_{(r-1)(n-1),rnm-rn}$$

26.5.3 例子

有交互的例子

```
dd = rnorm(27)
time<-rep(c('m','J1','J3'),9)
org <- rep(c(rep('ab',3),rep('ce',3),rep('body',3)),3)

# 正态性建检验
shapiro.test(dd)
hist(dd)
# 方差齐性检验
bartlett.test(dd,g=rep(1:3,each=9))
# 方差分析表
summary(aov(dd~time*org))
# 线性模型的分析
summary(lm(dd~time*org))
# 多重比较
TukeyHSD(aov(dd~factor(time)*factor(org)))
```

我们使用ANCOVA(单因素协方差中的例子)26.4, 考察性别(sex)及用药与否(ctl)与血压的关系. 统计模型为

$$y_{ijk} = a + b_i sex + c_j ctl + \gamma_{ij} + e_{ijk}$$

- a: 常数
- b_i: 常数, 代表性别的效应
- c_i: 常数, 代表用药与否的效应
- γ: 交互作用.

R公式可以这样(统计模型一章有其它的模型25)

```
y ~ A*B
y ~ A + B + A:B
y ~ B %in% A
y ~ A|B
```

y对A和B的非可加两因子方差分析模型（two factor non-additive model）。前两个公式表示相同的交叉分类设计（crossed classification），后两个公式表示相同的嵌套分类设计(nested classification)。抽象一点说，这四个公式指明同一个模型子空间。

下面是计算结果, 首先是方差分析表

```
# 列出平方和分解的值
> aov(y~ctl*sex)
Call:
   aov(formula = y ~ ctl * sex)

Terms:
                     ctl      sex   ctl:sex Residuals
Sum of Squares  1079.122    0.018    44.240 12206.861
Deg. of Freedom       1        1         1        96

Residual standard error: 11.27629
Estimated effects may be unbalanced

# 列出方差分析表, F及p值, 看到控制其它(sex和交互后)ctl的影响是显著的
```

```
> summary(aov(y~ctl*sex))
          Df  Sum Sq Mean Sq F value  Pr(>F)
ctl        1  1079.1  1079.1  8.4867 0.00445 **
sex        1 0.01849 0.01849  0.0001 0.99040
ctl:sex    1    44.2    44.2  0.3479 0.55668
Residuals  96 12206.9   127.2
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
# anova()的方差分析表, 与aov() 结果一样的
> anova(lm(y~ctl*sex))
Analysis of Variance Table

Response: y
         Df  Sum Sq Mean Sq F value  Pr(>F)
ctl       1  1079.1  1079.1  8.4867 0.00445 **
sex       1 0.01849 0.01849  0.0001 0.99040
ctl:sex   1    44.2    44.2  0.3479 0.55668
Residuals 96 12206.9   127.2
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

下面是线性模型的结果: 得到回归系数, 总体F值2.945, p: 0.03681, 说明有一个系数(截距)显著不为0. 虽然不显著, 但是控制了其它后, 男性(sex=1)比女性(sex=0)血压平均值要高1.34单位, 而服药组比不服药组血压平均高10.77单位

```
> summary(lm(y~sex*ctl))

Call:
lm(formula = y ~ sex * ctl)

Residuals:
   Min     1Q  Median     3Q    Max
-24.277  -9.841   1.245  10.219  17.503

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  102.187      4.893  20.883   <2e-16 ***
sex            1.345      3.213   0.419    0.676
```

```
ctl          10.772      7.495   1.437    0.154
sex:ctl      -2.726      4.622  -0.590    0.557
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.28 on 96 degrees of freedom
Multiple R-squared: 0.08427,    Adjusted R-squared: 0.05566
F-statistic: 2.945 on 3 and 96 DF,  p-value: 0.03681
```

26.6 两因素协方差分析

当结果变量(正态分布)可能与两个类型变量有关, 而同时需要控制一个或多个协变量(可以是连续或类型变量), 应该使用双向协方差分析.

双向协方差分析也可能表示成多重回归的特例.

这里, 我们仍然使用ANCOVA(单因素协方差中的例子)26.4, 考察性别(sex)及用药与否(ctl)与血压的关系, 但是把age作为协变量来控制. 可以看到age的影响是不显著的, $F = 0.1273, p = 0.722067$, 其系数为-0.12

```
# 方差分解情况
> aov(y~ctl*sex+age)
Call:
  aov(formula = y ~ ctl * sex + age)

Terms:
                    ctl      sex      age   ctl:sex Residuals
Sum of Squares  1079.122    0.018   16.333    43.172 12191.596
Deg. of Freedom       1        1        1        1       95

Residual standard error: 11.32840
Estimated effects may be unbalanced

# 方差分析表
> summary(aov(y~ctl*sex+age))
```

```
          Df  Sum Sq Mean Sq F value   Pr(>F)
ctl        1  1079.1  1079.1  8.4088 0.004638 **
sex        1 0.01849 0.01849  0.0001 0.990447
age        1    16.3    16.3  0.1273 0.722067
ctl:sex    1    43.2    43.2  0.3364 0.563284
Residuals 95 12191.6   128.3
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

# 多重回归系数
> summary(lm(y~sex*ctl+age))

Call:
lm(formula = y ~ sex * ctl + age)

Residuals:
   Min     1Q  Median     3Q    Max
-23.877  -9.876   1.457  10.238  17.546

Coefficients:
          Estimate Std. Error t value Pr(>|t|)
(Intercept) 103.9873     7.1703  14.502   <2e-16 ***
sex          1.3612     3.2278   0.422    0.674
ctl         10.8316     7.5316   1.438    0.154
age         -0.1240     0.3597  -0.345    0.731
sex:ctl     -2.6935     4.6439  -0.580    0.563
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.33 on 95 degrees of freedom
Multiple R-squared: 0.08542,    Adjusted R-squared: 0.04691
F-statistic: 2.218 on 4 and 95 DF,  p-value: 0.07278
```

26.7 随机效应模型

包nlme, 线性与非线性混合效应模型, 函数 lme()

包lme4, 线性混合效应模型, 函数 lmer()

26.7.1 问题描述

例如, 一项研究是研究激素与疾病的关系(护士卫生研究,参考文献[14] 12.8), 从5名月经后期的女性获得血样, 然后被分为两等份, 采用双盲的方式把血样送到实验室分析. 研究的目的是判断人与人之间的差异与一个人血样中的波动各有多大. 数据如下

```
# hormone 浓度
horm=c(25.5,30.4,11.1,15.0,8.0,8.1,20.7,16.9,5.8,8.4)
# 每个人重复2次
rep=rep(c(1,2),5)
# 5个人编号
per=rbind(1:5,1:5)[1:10]

blood=data.frame(horm=horm,rep=rep,per=per)
> blood
   horm rep per
1  25.5   1   1
2  30.4   2   1
3  11.1   1   2
4  15.0   2   2
5   8.0   1   3
6   8.1   2   3
7  20.7   1   4
8  16.9   2   4
9   5.8   1   5
10  8.4   2   5

# 两次重复的均值
m=matrix(horm,nr=2)
mean=colMeans(m)
> mean
[1] 27.95 13.05  8.05 18.80  7.10

# 两次重复的差值
```

```
delta=m[2,]-m[1,]
> delta
[1]  4.9  3.9  0.1 -3.8  2.6
```

可以看到, 对于激素平均值较大的人, 其差值也较大. 即重复测量的变异程度与该人的平均值大小有关, 我们将对数据取对数, 再做分析, 这样重复测量的标准差就会独立于取对数后的平均水平[2].

26.7.2 模型与假设检验

估计人与人之间的差异及人内部的差异时, 常使用下面的模型

$$y_{ij} = \mu + \alpha_i + e_{ij}$$

这个模型与固定效应模型是一样的, 只不过对它的解释不同. 其中

- y_{ij}: 第i个人的第j重复
- α_i: 人之间(组间)差异的随机变量, 常被认为服从正态分布$N(0, \sigma_A^2)$
- e_{ij}: 人内部(组内)差异的随机变量, 互相独立, 且独立于α, 服从正态分布$N(0, \sigma^2)$

这个方程常被称为随机效应(random-effect)单向方差分析模型.

第i,j个人的均值分别是$\mu + \alpha_i$, $\mu + \alpha_j$, 故每个人的均值是不同的, 其变异性的指标为σ_A^2. 第i个人多次重复的均值为$\mu + \alpha_i$, σ^2代表其变异性.

随机效应分析的一个重要目的是检验假设σ_A^2是否异于零, 即

$$H_0 : \sigma_A^2 = 0 \quad vs \quad H_1 : \sigma_A^2 > 0$$

[2]更多数据变换见14

零假设成立表明人与人之间没有差异, 所有差异来源于人内部的差异(波动, 也叫做噪声). 如果备择假设为真, 说明人与人之间, 或组之间有真实的差异.

26.7.3 几个公式

组内平均方差的期望为

$$E(MS_W) = \sigma^2$$

组间平均方差的期望为(平衡设计, 即每组重复数相同的时候)

$$\begin{aligned} E(MS_B) &= \sigma^2 + n\sigma_A^2 \\ n &= n_1 = n_2 = \cdots = n_k = \text{每个人的重复数} \end{aligned}$$

组间平均方差的期望为(非平衡设计, 即每组重复数不全相同的时候)

$$\begin{aligned} E(MS_B) &= \sigma^2 + n_0\sigma_A^2 \\ n_0 &= (\sum_{i=1}^{k} n_i - \sum_{i=1}^{k} n_i^2 / \sum_{i=1}^{k} n_i)/(k-1) \end{aligned}$$

显然如果$n = n_1 = n_2 = \cdots = n_k$, 即每个人的重复数相同, 那么

$$n_0 = [kn - kn^2)/(kn)]/(k-1) = (kn-n)/(k-1) = n$$

一般非平衡时, $n_0 < n$, 但是差异常常不大.

σ_A^2的无偏估计为

$$\hat{\sigma}_A^2 = E(\frac{MS_B - MS_W}{n}) = \frac{\sigma^2 + n\sigma_A^2 - \sigma^2}{n} = \sigma_A^2$$

在非平衡设计中, 只要使用n_0代替n即可.

26.7.4 F检验

我们可以使用与固定效应模型相同的检验统计量

$$F = MS_B/MS_W \sim F_{k-1,N-k} \quad (H_0\text{下,即}\sigma_A^2 = 0)$$

组内平均方差

$$MS_W = \sum_{i=1}^{k}\sum_{j=1}^{n_i}(y_{ij} - \bar{y}_i)^2/(N-k)$$

组间平均方差

$$MS_B = \sum_{i=1}^{k} n_i(\bar{y}_i - \bar{y})^2/(k-1)$$

其中

$$\begin{aligned} \bar{y}_i &= \sum_{j=1}^{n_i} y_{ij}/n_i \\ \bar{y} &= \sum_{i=1}^{k}\sum_{j=1}^{n_i} y_{ij}/N = \sum_{i=1}^{k} n_i\bar{y}_i/N \\ N &= n_1 + \cdots + n_k \end{aligned}$$

判断

$$if \quad F > F_{k-1,N-k,1-\alpha}, \quad reject\ H_0 if \quad F <= F_{k-1,N-k,1-\alpha}, \quad accept\ H_0$$

p值

$$p = P(F_{k-1,N-k} > F)$$

26.7.5 组内,组间平均方差的估计

估计组内方差

$$\hat{\sigma^2} = MS_W$$

估计组间方差(若小于0, 则令其等于0)

$$\hat{\sigma_A^2} = \frac{MS_B - MS_W}{n_0}$$

26.7.6 重复性研究中变异系数的估计

一般说来, 重复测量中变异系数< 20%是理想的, > 30%是不理想的. 重复测量中变异系数的定义为

$$CV = 100\% \frac{\sqrt{MS_W}}{mean\ of\ within\ group}$$

但是, 当方差随均值增加时, 更好的方法是使用下面的方法

- 对每个数取自然对数
- 计算MS_W
- $CV = 100\% \sqrt{MS_W}$

26.7.7 组内相关系数(ICC, 方差估计量分析,可靠性系数)

单向随机效应模型中, 同一个人两个重复之间的相关性称为组内相关系数(intraclass correlation coefficient, ICC), 记为ρ. 有多种方法估计组内相关系数, 最简单也是最普遍使用的方法是基于单向随机效应模型的.

$$\rho = \frac{\sigma_A^2}{\sigma^2 + \sigma_A^2}$$

点估计为

$$\hat{\rho} = max[\frac{\hat{\sigma}_A^2}{\hat{\sigma}^2 + \hat{\sigma}_A^2}, 0]$$

区间估计为

$$c1 = max[\frac{F/F_{k-1,N-k,1-\alpha/2} - 1}{n_0 + F/F_{k-1,N-k,1-\alpha/2} - 1}, 0]$$

$$c2 = max[\frac{F/F_{k-1,N-k,1-\alpha/2}}{n_0 + F/F_{k-1,N-k,1-\alpha/2}}, 0]$$

这个分析也叫做方差估计量分析(analysis-of-variance estimator).

组内相关系数也常常理解为可靠性的一个度量, 有时候也称为可靠性系数(reliability coefficient).

解释

- $\rho < 0.4$: 重复性很差
- $0.4 <= \rho < 0.75$: 重复性中等
- $\rho >= 0.75$: 重复性很好

包multilevel函数ICC1, 计算组内相关系数(重复性), ICC2计算组之间的可靠性(reliability).(详细用法参考27.11)

```
> library(multilevel)
> res=aov(horm~as.factor(per),blood)
> summary(res)
             Df  Sum Sq Mean Sq F value   Pr(>F)
as.factor(per)  4 2.65775 0.66444  22.146 0.002221 **
Residuals       5 0.15001 0.03000
---
```

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
# 组内(人内部)相关系数, 即可重复性很大. 即组内部的方差有91.36可以被
> ICC1(res)
[1] 0.9135923
# 组间(人之间)的相关系数也很大, 表示人之间的均值可以很好的区分
> ICC2(res)
[1] 0.9548453
```

26.7.8 例子

参考文献 [40] 10.1

nlme 包的 lme() 函数与 lme4 包的 lmer() 函数计算混合效应模型(固定+随机效应).

注意, 与SAS计算结果中的F值是不同的.

随机效应因素放在竖线后面, 写法见例子.

随机效应的结果主要看Random effects部分, nlme包的lme()函数给出了标准差, 但是可以容易的计算出方差.

- 组间变异 $\sigma_A^2 = 0.5632196^2 = 0.3172163$
- 组间平均方差 $MS_B = n_0\sigma_A^2 = 2 * 0.5632196^2 = 0.6344326$
- 组内变异 $\sigma^2 = 0.1732123^2 = 0.0300025$
- 组内平均变异就是 $MS_W = \sigma^2 = 0.0300025$
- 变异系数$100\%\sigma = 17.32123\%$
- 组内相关系数的点估计 $\rho = \frac{\sigma_A^2}{\sigma^2+\sigma_A^2} = 0.317/(0.317 + 0.030) = 0.914$
- 组内相关系数的区间估计, 其中$F = 99.27, qf(0.975, 4, 5) = 7.387886, qf(0.025, 4, 5) = 1.107$, 那么$c_1 = max[(99.27/7.39 - 1)/(2 +$

$(99.27/7.39-1)),0] = 0.86,\ c_2 = (99.27/1.107-1)/(2+(99.27/1.107-1)) = 0.9779432$ [3]

```
# hormone 浓度
horm=c(25.5,30.4,11.1,15.0,8.0,8.1,20.7,16.9,5.8,8.4)
# 每个人重复2次
rep=rep(c(1,2),5)
# 5个人编号
per=rbind(1:5,1:5)[1:10]
# 数据框内, hormone 水平取对数值
blood=data.frame(horm=log(horm),rep=rep,per=per)

library(nlme)
ll=lme(horm~1,random=~1|per,data=blood)
> summary((lme(horm~1,random=~1|per,data=blood)))
Linear mixed-effects model fit by REML
 Data: blood
      AIC      BIC    logLik
  14.67583 15.26751 -4.337916

Random effects:
 Formula: ~1 | per
       (Intercept)  Residual
StdDev:   0.5632196 0.1732123

Fixed effects: horm ~ 1
             Value Std.Error DF  t-value p-value
(Intercept) 2.568296 0.2577664  5 9.963656   2e-04

Standardized Within-Group Residuals:
      Min        Q1       Med        Q3       Max
-1.2321300 -0.4460533 -0.1258207  0.6986361  0.9061354

Number of Observations: 10
Number of Groups: 5

# 可以使用 VarCorr 得到方差与标准差, 结果与lmer()一样.
> VarCorr(ll)
per = pdLogChol(1)
```

[3]由于与SASF值计算不同, 区间也不同. SAS的$F = 22.15$

```
          Variance   StdDev
(Intercept) 0.31721636 0.5632196 # 组间变异性56%
Residual    0.03000249 0.1732123 # 组内变异性17%

# F值, 注意与SAS结果不同.
> anova(ll)
          numDF denDF  F-value p-value
(Intercept)     1     5 99.27445   2e-04

> coef(ll)
$per
  (Intercept)
1    3.292320
2    2.557985
3    2.107447
4    2.912448
5    1.971279
```

lme4包的函数lmer()同时给出了方差和标准差(Random effects部分下面的 Variance Std.Dev.) 如果固定因素多于一个, 还给出固定效应因素的协方差矩阵.

```
library(lme4)
> lmer(horm~1+(1|per),data=blood)
Linear mixed model fit by REML
Formula: horm ~ 1 + (1 | per)
   Data: blood
   AIC   BIC logLik deviance REMLdev
 14.68 15.58 -4.338    7.749   8.676
Random effects:        方差   标准差
 Groups   Name        Variance Std.Dev.
 per      (Intercept) 0.317209 0.56321
 Residual             0.030003 0.17321
Number of obs: 10, groups: per, 5

Fixed effects:
          Estimate Std. Error t value
(Intercept)   2.5683     0.2578   9.964
```

```
> coef(lmer(horm~1+(1|per),data=blood))
  (Intercept)
1    3.292321
2    2.557985
3    2.107446
4    2.912449
5    1.971278
```

Chapter 27

一致性(agreement)估计

本部分来自参考文献[41]《Multilevel Modeling in R》的翻译.

主要使用的包为: base, nlme, multilevel

其内容还有: 普通最小二乘法(Ordinary Least Square,简称OLS),OLS方法使用线性模型lm(), 和随机模型lme(), 参考回归与方差分析部分.

增长模型: 是Solow于1956年首次创立的, 用来说明储蓄、资本积累和增长之间的关系。自建立以来, 这一模型一直是分析以上三个变量关系的主要理论框架。参考文献《Multilevel Modeling in R》[41] chapter 4. 主要使用 lme() 计算, 6个步骤

27.1 Agreement(一致性相关系数, CCC)

包multilevel里有几个函数可以估计推测一致性指标(agreement indices). 函数为rwg, rwg.j, rwg.sim, rwg.j.sim,rwg.j.lindell, ad.m, ad.m.sim rgr.agree. 具体见函数帮助.

另外, 包agreement的函数lin.simulation()使用模拟方法计算两个方法的一致性(参考文献[30]). lin.simulation()函数中两个方

法X,Y对测量一致性相关系数(Concordance Correlation Coefficient, CCC) 是Pearson($\rho_{x,y}$)相关系数与精确度(C_b)的乘积, (详细见函数帮助及其参考文献Lin,1989, Lin2002文献给出了其它几个指标来度量一致性)

$$\begin{aligned} C_b &= 2\sigma_x\sigma_y/[\sigma_x^2+\sigma_y^2+(\mu_x-\mu_y)^2] \\ CCC &= \rho_{xy}C_b \end{aligned}$$

27.2 一致性度量

此处的一致性就是方差分析中随机效应模型中的组内相关系数. 其中r_{wg}度量单独数据组内一致性(within group agreement measure for single item measures, James et al. (1984), 具体文献见函数帮助), 默认期望的随机方差(Expected Random Variance)为2

$$rwg = 1-(ObservedGroupVariance/ExpectedRandomVariance)$$

一般, 一致性系数> 0.7比较好, 否则就比较差.

27.3 估计EV

期望随机方差(Expected Random Variance, EV)的估计是这样的(参考help(rwg)), 默认的$EV=2$是按照分5个等级计算的(例如, Strongly Disagree, Disagree, Neither, Agree, Strongly Agree), 如果不是5个组, 记A为组数, 那么A的方差基于矩形分布(rectangular distribution)

$$EV = (A^2-1)/12$$

27.4 例子

```
data(bhr2000,package="multilevel")
RWG.RELIG<-rwg(bhr2000$RELIG,bhr2000$GRP,ranvar=2)
# 共94组, 查看前10组
```

```
> RWG.RELIG[1:10,]
   grpid       rwg gsize
1      1 0.11046172    59
2      2 0.26363636    45
3      3 0.21818983    83
4      4 0.31923077    26
5      5 0.22064137    82
6      6 0.41875000    16
7      7 0.05882353    18
8      8 0.38333333    21
9      9 0.14838710    31
10    10 0.13865546    35
>  summary(RWG.RELIG)
     grpid        rwg            gsize
 1      : 1   Min.   :0.0000   Min.   :  8.00
 10     : 1   1st Qu.:0.1046   1st Qu.: 29.50
 11     : 1   Median :0.1899   Median : 45.00
 12     : 1   Mean   :0.1864   Mean   : 54.55
 13     : 1   3rd Qu.:0.2630   3rd Qu.: 72.50
 14     : 1   Max.   :0.4328   Max.   :188.00
 (Other):93
```

```
# 对rwg排序, 或查看直方图也比较有用
> sort(RWG.RELIG[,2])
> hist(RWG.RELIG[,2])
```

下面来估计工作时间的r_{wg}, 我们需要改变期望随机方差(expected random variance, EV). 工作时间被要求是11个等级(11-point item, 即按照工作时间分为11等级), 因此EV基于矩形分布(rectangular distribution), 故$\sigma^2_{EV} = (11^2 - 1)/12 = 10.00$.

```
# 工作时间等级数
> length(unique(bhr2000$HRS))
[1] 11
# 计算不同组GRP的工作时间HRS的一致性, 0.73>0.7表明一致性比较好
> RWG.HRS<-rwg(bhr2000$HRS,bhr2000$GRP,ranvar=10.00)
> mean(RWG.HRS[,2])
```

```
[1] 0.7353417
```

27.5 rwg.j()

函数rwg.j()与rwg()几乎一样, 但是估计多个item的一致性. 第一个参数为矩阵, 每列是一个item, 每行是一个观察(response), 默认使用5个等级. 下面看到2到12列总的一致性系数很高

```
> RWGJ.LEAD<-rwg.j(bhr2000[,2:12],bhr2000$GRP,ranvar=2)
> summary(RWGJ.LEAD)
    grpid       rwg.j          gsize
1      : 1   Min.   :0.7859   Min.   :  8.00
10     : 1   1st Qu.:0.8708   1st Qu.: 29.50
11     : 1   Median :0.8925   Median : 45.00
12     : 1   Mean   :0.8876   Mean   : 54.55
13     : 1   3rd Qu.:0.9088   3rd Qu.: 72.50
14     : 1   Max.   :0.9440   Max.   :188.00
(Other):93
```

27.6 rwg.j.lindell()

一般认为(rwg, rwg.j), 随着等级数(item)的增加, 偏差也会增大, 这是基于 Spearman-Brown reliability estimator. 但是t Lindell and colleagues等认为这个估计好像没有理论基础, 即没有理由认为随着等级数(item)的增加, 偏差会增大. 故Lindell and colleagues等发展了一个方法, 使用平均方差代替方差. 函数rwg.j.lindell()计算此定义的一致性. 可以看到结果(均值为0.43)明显低于rwg.j方法(均值0.89).

```
RWGJ.LEAD.LIN<-rwg.j.lindell(bhr2000[,2:12],
   bhr2000$GRP,ranvar=2)
```

```
> summary(RWGJ.LEAD.LIN)
    grpid     rwg.lindell        gsize
 1      : 1   Min.   :0.2502   Min.   :  8.00
 10     : 1   1st Qu.:0.3799   1st Qu.: 29.50
 11     : 1   Median :0.4300   Median : 45.00
 12     : 1   Mean   :0.4289   Mean   : 54.55
 13     : 1   3rd Qu.:0.4753   3rd Qu.: 72.50
 14     : 1   Max.   :0.6049   Max.   :188.00
 (Other):93
```

27.7 置信区间估计

基本的思想是基于一个已知的分布(一般是均匀分布), 随机抽样, 重复估计r_{wg},

Cohen et al., (2001)发现, 组的大小(group size)与item数量对r_{wg}影响很大,

2003, Dunlap and colleagues发现组大小和每个item的等级个数(response number)对r_{wg}影响很大. 在5个等级个数(例如, Strongly Disagree, Disagree, Neither, Agree, Strongly Agree),95%置信区间从3组的1.0变到150组的0.12.

rwg.sim()提供Dunlap and colleagues(2003)方法. 对于组数为10的item, 本身5个等级(response)的变量, 我们可以这样使用函数

```
# run for long time (>30s)
> RWG.OUT<-rwg.sim(gsize=10, nresp=5, nrep=10000)
> summary(RWG.OUT)
$rwg
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.0000  0.0000  0.1204  0.2000  0.8667

$gsize
[1] 10

$nresp
```

```
[1] 5

$nitems
[1] 1

$rwg.95
[1] 0.5277778
```

95%置信区间大小为0.53, 其它值可以得到稍微不同的结果.

还提供了一个泛型函数 quantile(agree.sim) 来计算其它置信水平下的置信区间大小.

```
> quantile(RWG.OUT,c(.90,.95,.99))
  quantile.values confint.estimate
1            0.90       0.4166667
2            0.95       0.5277778
3            0.99       0.6666667
```

函数rwg.j.sim()基于Cohen et al. (in press)的工作(扩展了 Dunlap et al., (2003)), 考察多个item和组数(group size), 等级数(response number).

一般模拟采样的次数大于10000.

Cohen et al., (2001)的工作表明相关的item与不相关的item的一致性估计结果差不多, 但是相关的情况更可靠, 推荐使用. 忽略参数 itemcors 表示假设item之间独立.

下面的例子是15个组, 7个item, 5个等级(response, $A = 5$). 95%置信区间上限为$0.54 < 0.7$, 那么0.7可能太严格了. 基于此, 我们可以调整0.55为两个一致性有区别的显著性阈值($p < 0.05$).

```
> RWG.J.OUT<-rwg.j.sim(gsize=15,nitems=7,nresp=5,nrep=1000)
> summary(RWG.J.OUT)
$rwg
```

```
   Min.  1st Qu.   Median    Mean  3rd Qu.    Max.
0.000000 0.000000 0.009447 0.155400 0.314100 0.753000

$gsize
[1] 15

$nresp
[1] 5

$nitems
[1] 1

$rwg.95
[1] 0.5425764
```

下面是一个实际的例子, 演示如何使用rwg.j.sim()来计算数据lq2002三个item的平均的一致性, 结果均值为0.58. 我们要考察0.58是否显著的一致. 模拟的阈值为$0.34 < 0.58$, 那么其item可以说是一致的($p = 0.05$)

```
> data(lq2002,package="multilevel")
> RWG.J<-rwg.j(lq2002[,c("TSIG01","TSIG02","TSIG03")],
+ lq2002$COMPID,ranvar=2)
> summary(RWG.J)
    grpid       rwg.j          gsize
 10     : 1   Min.   :0.0000   Min.   :10.00
 13     : 1   1st Qu.:0.5099   1st Qu.:18.00
 14     : 1   Median :0.6066   Median :30.00
 15     : 1   Mean   :0.5847   Mean   :41.67
 16     : 1   3rd Qu.:0.7091   3rd Qu.:68.00
 17     : 1   Max.   :0.8195   Max.   :99.00
 (Other):43

# 模拟, 阈值为0.34. 需要MASS包mvrnorm()函数产生多元正态分布随机数
> library(MASS)
>  RWG.J.OUT<-rwg.j.sim(gsize=42,nitems=3,nresp=5,
+   itemcors=cor(lq2002[,c("TSIG01","TSIG02","TSIG03")]),
+   nrep=1000)
```

```
> summary(RWG.J.OUT)
$rwg
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.00000 0.00000 0.00000 0.09055 0.17340 0.57660

$gsize
[1] 42

$nresp
[1] 5

$nitems
[1] 1

$rwg.95
[1] 0.3406643
```

27.8 平均偏差(AD)一致性估计

Burke, Finkelstein and Dusig (1999)建议使用平均偏差(Average Deviation, AD)指标度量祖辈一致性. Cohen et al., (in press)也称为Mean, Median Average Deviation(MAD). 每个组的AD为

$$AD = \sum |x_{ij} - X_j|/N$$

N为组内的样本数. 每个item的AD计算后, 取平均作为最后的AD值.

如果AD值小于$A/6$那么表明一致性好, 否则不好. (A为等级数, 例如A=5, Strongly Disagree, Disagree, Neither, Agree, Strongly Agree. $A/6 = 0.83$),

下面的AD值为$0.86 > 0.83$, 说明一致性不好.

```
data(bhr2000)
```

```
AD.VAL <- ad.m(bhr2000[, 2:12], bhr2000$GRP)
# 共99个
> AD.VAL
  grpid     AD.M gsize
1     1 0.8481366    59
2     2 0.8261279    45
3     3 0.8809829    83
4     4 0.8227542    26
5     5 0.8341355    82
>  mean(AD.VAL[,2:3])
     AD.M     gsize
 0.8690723 54.5454545
```

如果使用中位数Median, 结果会不同

```
>  AD.VAL <- ad.m(bhr2000[, 2:12], bhr2000$GRP,type="median")
>  mean(AD.VAL[,2:3])
     AD.M     gsize
 0.8297882 54.5454545
```

只计算一个item的一致性, 例如工时间, 里面有11个等级, 故阈值为$11/6 = 1.83$, 下面计算AD值为$1.25 < 1.83$, 说明一致性还可以.

```
>  AD.VAL.HRS <- ad.m(bhr2000$HRS, bhr2000$GRP)
>  mean(AD.VAL.HRS[,2:3])
    AD.M    gsize
 1.249275 54.545455
```

27.9 AD显著性检验

函数ad.m.sim()基于 Cohen et al. (in press) , Dunlap et al., (2003) 的工作. 原理也是基于均匀分布采样. 下面的例子表明, 如果有99%的把握说一致性是显著的, 那么AD值要小于1.108

```
library(MASS)
AD.SIM<-ad.m.sim(gsize=55,nresp=5,
   itemcors=cor(bhr2000[,2:12]),type="mean",nrep=1000)
> summary(AD.SIM)
$ad.m
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.088   1.180   1.207   1.207   1.234   1.322

$gsize
[1] 55

$nresp
[1] 5

$nitems
[1] 11

$ad.m.05
[1] 1.141397

$pract.sig
[1] 0.8333333

> quantile(AD.SIM,c(.10,.05,.01))
  quantile.values confint.estimate
1          0.10        1.156424
2          0.05        1.141397
3          0.01        1.108279
```

27.10 随机组采样方法

随机组采样方法(Random Group Resampling)函数为rgr.agree(), 类似rwg.j.sim(), 区别是rgr.agree使用实际的组数据, 而rwg.j.sim使用期望的均匀分布.下面例子结果得到随机采样的组内方差为3.32, 其标准差为0.79, 真实的组内的方差小于2.65, $z = -8.4 < -1.96 = z_{0.025}$, 显示真实的组内是一致的.

```
> RGR.HRS<-rgr.agree(bhr2000$HRS,bhr2000$GRP,1000)
> summary(RGR.HRS)
$`Summary Statistics for Random and Real Groups`
         随机采样方差                    真实组内方差
N.RanGrps Av.RanGrp.Var SD.Rangrp.Var Av.RealGrp.Var   Z-value
[1,]      990     3.322136     0.7927865      2.646583 -8.478535

$`Lower Confidence Intervals (one-tailed)`
   0.5%      1%     2.5%      5%     10%
1.314331 1.555556 1.935742 2.157274 2.405185

$`Upper Confidence Intervals (one-Tailed)`
    90%     95%    97.5%     99%    99.5%
4.278615 4.628046 4.978991 5.718599 6.353275
```

而且看到, 5%的随机的值小于2.16, 那么我们有95%的把握说组内的方差小于2.16.

27.11 组内相关系数(ICC)

函数ICC1和ICC2基于Bartko, (1976), James (1982), and Bliese (2000)的描述计算组内相关系数(intraclass correlation coefficient, ICC)(详细解释参考26.7.7)

其中ICC1等价于随机模型中个体水平方差被组内(个体内部)解释的程度. 结果为17%表明工作时间的方差有17%可以被组内(个体内部)解释. 即组内相关性(重复性)不好, 组内差异大.

ICC2组均值的可靠性(区分度,), 为0.92表明组之间的平均工作时间可以很好的区分. reliability of the group means.

```
> data(bhr2000)
> hrs.mod<-aov(HRS~as.factor(GRP),data=bhr2000)
> summary(hrs.mod)
                Df  Sum Sq Mean Sq F value    Pr(>F)
as.factor(GRP)   98  3371.4    34.4  12.498 < 2.2e-16 ***
```

```
Residuals     5301 14591.4    2.8
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
# 0.17表明组内差异大
> ICC1(hrs.mod)
[1] 0.1741008
# 0.92表明组间可以被工作时间平均值很好的区分
> ICC2(hrs.mod)
[1] 0.9199889



data(bhr2000)
graph.ran.mean(bhr2000$HRS, bhr2000$GRP, nreps=1000,
   limits=c(8,14),bootci=TRUE)
```

Part V

线性模型

Chapter 28

一般线性回归(Linear regression)

28.1 数据

下面是某药物浓度(x)与对应的吸光度(y), (实际上不是线性关系, 但是这里我们用做例子)

```
# 某药物浓度
x=c(15.625,  31.250,  62.500, 125.000, 250.000, 500.000, 1000.000)
# 对应的吸光度
y=c(0.103, 0.217, 0.364, 0.678, 0.968, 1.501, 1.927)
```

28.2 模型描述

$$y = b_0 + b_1 * x1$$

检验 b_0, b_1 是否等于 0.

28.3 平方和分解

28.3.1 总平方和=残差平方和+回归平方和

$$\sum_{i=1}^{n}(y_i-\bar{y})^2=\sum_{i=1}^{n}(\hat{y}_i-\bar{y})^2+\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$$

总平方和(TotalSS) = 残差平方和(ResSS)+回归平方和(RegSS), 即 TotalSS = ResSS + RegSS

实际计算 TotalSS, RegSS 与 ResSS

```
# 其中 res=lm(y~x)
TotalSS = sum( (y-mean(y))^2 ) # 总平方和
ResSS = sum( res$residuals^2 ) # 残差平方和
RegSS = TotalSS - ResSS # 回归平方和
```

```
> TotalSS
[1] 2.816866
> ResSS
[1] 0.2625031
> RegSS
[1] 2.554363
```

28.3.2 回归平均平方(RegMS)与残差平均平方(ResMS)及其自由度

回归平均平方(RegMS) 与 残差平均平方(ResMS)

RegMS = RegSS/模型中预测变量数(k), 在简单线性回归中, 由于 k=1, 所以 RegMS = RegSS.

ResMS = ResSS/(n-k-1), 在简单线性回归中, k=1. 我们称 n-k-1 为残差平方和的自由度, 记为 Res df. 残差平均平方(ResMS)有时在文献中也记为 $s^2_{y\cdot x}$

下面计算

```
RegMS=RegSS/1 # 回归平均平方, k=1, 结果为 2.554363
ResMS=ResSS/(7-1-1) # 残差平均平方, 结果为 0.05250063
```

28.4 拟合回归直线–最小二乘法

28.4.1 原始平方和与修正平方

定义: x 的原始平方和 (raw sum of squares for x) 为

$$\sum_{i=1}^{n} x_i^2$$

定义: x 的修正平方和 (corrected sum of squares for x) 常记为 L_{xx}, 定义如下

$$\sum_{i=1}^{n}(x_i - \bar{x})^2 = \sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2/n$$

它代表了 x 离均值的偏差平方和.

定义: 类似的, y 的原始平方和 (raw sum of squares for y) 为

$$\sum_{i=1}^{n} y_i^2$$

定义: y 的修正平方和 (corrected sum of squares for y) 记为 L_{yy}, 定义如下

$$\sum_{i=1}^{n}(y_i-\bar{y})^2=\sum_{i=1}^{n}y_i^2-(\sum_{i=1}^{n}y_i)^2/n$$

注意 L_{xx}, Lyy 分别是 x 的样本方差 (s_x^2) 及 y 的样本方差 (s_y^2) 的分子. 即

$$s_x^2=\sum_{i=1}^{n}(x_i-\bar{x})^2/(n-1), \qquad s_y^2=\sum_{i=1}^{n}(y_i-\bar{y})^2/(n-1)$$

定义: 叉积原始和 (raw sum of cross products) 定义为

$$\sum_{i=1}^{n}x_i^2y_i^2$$

定义: 叉积修正和为

$$\sum_{i=1}^{n}(x_i-\bar{x})^2(y_i-\bar{y})^2=\sum_{i=1}^{n}x_i^2y_i^2-(\sum_{i=1}^{n}x_i)(\sum_{i=1}^{n}y_i)/n$$

28.4.2 最小平方线

最小平方线 (least-square line) 或估计的回归线 (estimated regression line) 为 $y=a+bx$, 它寻找 a,b 使得下式最小

$$S=\sum_{i=1}^{n}d_i^2=\sum_{i=1}^{n}(y_i-a-bx_i)^2$$

这种估计回归直线参数的方法称为最小平方法.

可以证明对 β 的最好估计

$$b = L_{xy}/L_{xx}$$

因为 L_{xx} 总是正数, 因此 b 的符号总与 L_{xy} 相同.

容易得到截距 $a = \bar{y} - b\bar{x}$

给出 x 值后 y 的平均值的预测值就是 $\hat{y} = a + bx$, 它总是在直线上.

28.5 计算

28.5.1 回归函数lm()

直接的回归分析的结果比较少. 若需要进一步的结果, 通常需要对结果使用泛型函数, 例如 summary 等.

```
> res=lm(y~x)
> res

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)          x
   0.306489    0.001821

> names(res) # 可以使用的结果
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "xlevels"       "call"          "terms"         "model"
```

28.5.2 进一步分析的泛型函数

下面的显示了一些可以对分析结果对象做一些补充分析的泛型函数, 主要参数一般都是分析结果对象, 但是有些情况下,如泛型函数如predict 或 update 需要一些额外的参数.

```
add1  coef     effects kappa  predict residuals
alias deviance family  labels print   step
anova drop1    formula plot   proj    summary
```

下面是常用的几个介绍

- add1 连续测试所有可以加入模型的元素项
- drop1 连续测试所有可以从模型中移除的元素项
- step 通过AIC (调用add1 和drop1)选择一个模型
- anova 计算一个或多个模型的方差/残差分析表
- predict 通过拟合的模型计算一个新的数据集的预测值
- update 用新的数据或者公式拟合一个模型
- deviance 残差平方和
- logLik 对数似然值
- AIC 赤池信息量
- vcov 返回主要参数的协方差矩阵

28.5.3 summary()函数-对回归结果的统计与检验

summary(lm(x y)) 与 summary(lm(y x)) 的 R^2 是一样的.

summary 里有相关系数 R^2, summary(lm(y x))$r.squared. str(lm()) 里没有.

t 值表示系数是否显著不等于0的概率. 给出了总的F值. 若想知道每个系数的F值, 参考anova(res¡-lm(y x))

下面是summary结果及其含义. 各个值的计算见后面.

```
> summary(res)

Call:
lm(formula = y ~ x)

Residuals:
      1       2       3       4       5       6       7
-0.23193 -0.14638 -0.05627  0.14395  0.20638  0.28426 -0.20000

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.306488   0.113905   2.691 0.043260 *
x           0.001821   0.000261   6.975 0.000932 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2291 on 5 degrees of freedom # 残差的标准误
Multiple R-squared: 0.9068,     Adjusted R-squared: 0.8882 # 相关系数的平方
F-statistic: 48.65 on 1 and 5 DF,  p-value: 0.0009318 # F值, 整体回归的检验
```

28.5.4 使用anova检测系数显著性

anova 给出斜率是否显著不等于0的概率(F值及其p值). 结果与summary()函数的结果一样.

当 p 值很小时, 斜率就显著不为 0. 即 y 依赖于自变量x. 下面的结果告诉我们, y 依赖于 x, 冒的风险为 0.0009318.

```
> anova(res)
```

```
Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value    Pr(>F)
x          1 2.5544  2.5544  48.654 0.0009318 ***
Residuals  5 0.2625  0.0525
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

注意, 在多元回归中, 结果依赖于自变量的顺序. 顺序不同其p值是不同的. 有时候, 结果会相反.

28.5.5 回归系数的置信区间(CI)

help(lm) 并查询 interval. 函数为 confint()

```
> confint(res)
                2.5 %      97.5 %
(Intercept) 0.01368702 0.599289993
x           0.00114960 0.002491426
```

28.5.6 计算回归预测的y值及区间

```
# 参数 interval='confidence' 将得到3列值,后两列为上下区间
> predict(res,interval='confidence')
       fit       lwr      upr
1 0.3349340 0.04883083 0.6210372
2 0.3633795 0.08374200 0.6430171
3 0.4202706 0.15279957 0.6877416
4 0.5340526 0.28734357 0.7807617
5 0.7616168 0.53786603 0.9853675
6 1.2167450 0.95092253 1.4825675
7 2.1270015 1.59723497 2.6567680
```

28.6 检验

28.6.1 手工计算F值

与summary()函数的F值结果一样.

```
F=RegMS/ResMS # 结果 48.65395
```

与 summary 结果一样.

28.6.2 方差齐性的检验

两样本的方差齐性检验使用F检验. 多于两样本则使用bartlett.test. 2个非正态样本参考 ansari.test 或 mood.test, 它们是非参数检验. 多于2个非正态样本参考 fligner.test

28.6.3 回归系数的假设检验

我们要检验回归系数β

$$H_0 : \beta = 0$$
$$H_1 : \beta \neq 0$$

F检验和t检验都可以. F检验的自由度为 1 及 n-2. t检验等价于F检验, 而且还能提供斜率的区间估计, 故此方法广泛使用. F检验是在 H_0 成立下, $F = RegMS/ResMS \sim F$分布.

这里使用car包的linear.hypothesis函数检验回归假设. 可以检验任何指定系数, 或交互效应的系数的显著性, 可以指定使用卡方检验或F检验, 默认使用F检验.

(linear.hypothesis用法很多，也有很多等价形式，方便使用。详细见函数帮助)

```
> library(car)
> mod.davis <- lm(weight ~ repwt, data=Davis)
 # 检验截距=0,系数=1
> linear.hypothesis(mod.davis, diag(2), c(0,1))
Linear hypothesis test

Hypothesis:
(Intercept) = 0
repwt = 1

Model 1: weight ~ repwt
Model 2: restricted model

  Res.Df   RSS  Df Sum of Sq    F Pr(>F)
1    181 12828
2    183 13074  -2     -246 1.74   0.18

> mod.duncan.2 <- lm(prestige ~ type*(income + education), data=Duncan)
> coefs <- names(coef(mod.duncan.2))
  # 检验零假设: 除截距外, 其它都为0
> linear.hypothesis(mod.duncan.2, coefs[-1])

Linear hypothesis test

Hypothesis:
typeprof = 0
typewc = 0
income = 0
education = 0
typeprof:income = 0
typewc:income = 0
typeprof:education = 0
typewc:education = 0

Model 1: prestige ~ type * (income + education)
Model 2: restricted model

  Res.Df    RSS Df Sum of Sq     F    Pr(>F)
1     36   3351
2     44  43688 -8    -40337 54.174 < 2.2e-16 ***
---
```

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

最下面部分分别给出截距和repwt项的残差自由度, 残差平方和, 自由度, 平方和, F值, p值.

28.6.4 异残差检验(Breusch-Pagan test)–检验残差是否为常量

此检验常常称为Breusch-Pagan test. Cook and Weisberg (1983)也独立提出此检验.

```
> library(car)
> r=lm(interlocks~assets+sector+nation, data=Ornstein)
# 默认使用拟合值 ~ fitted.values检验
> ncv.test(r)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 46.98537    Df = 1     p = 7.151835e-12
# 可以指定残差方程, 此处为 ~ assets+sector+nation
> ncv.test(r,~ assets+sector+nation, data=Ornstein)
Non-constant Variance Score Test
Variance formula: ~ assets + sector + nation
Chisquare = 74.73535    Df = 13     p = 1.066320e-10
```

28.7 绘图

```
   # > a
#        brain   body   EQ root_to_tip_dN_dS    dN     dS
#  5pbht   1824 209530 4.14          0.44340 0.0020 0.0044
#  6lsht   2387 328000 4.01          0.69367 0.0031 0.0034
# 7twyht    940 261099 2.94          0.59890 0.0011 0.0019
#  9bj     2083 636000 2.24          0.83330 0.0036 0.0050
# 10baiji   510  82000 2.17          0.66220 0.0130 0.0158
```

```
# 11hj     1425 770500 1.39          0.51470 0.0154 0.0308
#  12zmxj   622 168500 1.63          0.62390 0.0271 0.0558
#  8jt      468 324000 3.71          0.85700 0.0061 0.0068

a=read.csv('KaKsAa.csv',h=T)
names=a[,1]
a=a[,2:7]
rownames(a)<-names
attach(a)
a1=a[order(log(dN)),]

res=lm(dN~EQ)
pre=predict(res)
res1=lm(dS~EQ)
pre1=predict(res1)
res2=lm(root_to_tip_dN_dS~EQ)
pre2=predict(res1)

names=rownames(a)

op<-par(mfrow=(c(2,2)))

plot(dN~EQ,xlim=c(1,5),ylim=c(0,0.06),pch=15,col="red")
abline(res,col="red",lwd="2")
text(EQ+0.2,dN,names,cex=0.7)

plot(dS~EQ,xlim=c(1,5),ylim=c(0,0.06),pch=17,col="blue")
abline(res1,col="blue",lwd="2")
text(EQ+0.2,dS,names,cex=0.7)

plot(root_to_tip_dN_dS~EQ,xlim=c(1,5),pch=16,col="blue")
abline(res2,col="blue",lwd="2")
text(EQ+0.2,root_to_tip_dN_dS,names,cex=0.7)

par(op)
```

28.8 TODO: 多元回归

假设检验为

$$H_0 : \beta_1 = \cdots = \beta_k = 0 \quad vs. \quad H_1 : 至少一个\beta_j \neq 0$$

使用F检验

具体步骤, 参考 [14] P. 448-449.

若拒绝零假设, 我们接下来应该检验

$$H_0 : \beta_l = 0另外所有\beta_j \neq 0 \quad vs. \quad H_1 : 所有\beta_j \neq 0$$

使用 t 检验或偏 F 检验, 这两个方法是完全等价的. 具体步骤, 参考 [14] P. 449-453.

与简单回归类似, 分析方法也相同. 使用公式

```
lm(y~x1+x2+...)
lm(y~.,␣data=mydata)␣#y␣对其它所有变量的回归
␣␣␣␣␣
```

对于自变量为离散或属性数据(性别,年龄等)的回归分析, 请参考 单因素协方差分析(ANCOVA)26.4

我们通常希望尽量减少自变量的个数, 于是一个直观的想法是做很多回归检验. 但是, 这会增加II型错误的概率. 最好使用 Tukey 等方法(anova?)

主成分分析及主成分回归是不错的选择.

Chapter 29

相关

有时候我们不一定对预测感兴趣, 而是对两个变量间的定量关系感兴趣. 如此我们引入相关系数, 描述两个变量之间的定量关系比回归系数更合适.

29.1 样本(Pearson)相关系数

L_{xx}, L_{yy}, L_{xy} 的定义参考 28.4.1

29.1.1 定义

样本 (Pearson) 相关系数定义为

$$r = L_{xy}/\sqrt{L_{xx}L_{yy}}$$

总在 (-1,+1) 之间, 其解释与总体相关系数一样.

$r > 0$: 正相关

$r = 0$: 不相关

$r < 0$: 负相关

29.1.2 与总体相关系数的关系

对 r 分子分母同时除以 $n-1$, 有

$$r=\frac{L_{xy}/(n-1)}{\sqrt{\frac{L_{xx}}{(n-1)}\frac{L_{yy}}{(n-1)}}}$$

记 $s_x^2=L_{xx}/(n-1), s_y^2=L_{yy}/(n-1), s_{xy}=L_{xy}/(n-1)$, 那么

$$r=\frac{s_{xy}}{s_x s_y}$$

此时可以看出, 样本相关系数与总体相关系数完全相似.

29.1.3 样本相关系数 r 与样本回归系数 b 的关系

注意到 (参考28.4.2)

$$b=L_{xy}/L_{xx},\qquad r=L_{xy}/\sqrt{L_{xx}L_{yy}}$$

有

$$b=r\frac{L_{yy}}{L_{xx}}=r\frac{s_y}{s_x}$$

回归系数 b 可以看做相关系数 r 的一个重新标度的变形, 尺度因子是 y 的标准差 (s_y)与 x 标准差 (s_x) 的比值.

b 的单位是 y/x 的单位.

总体相关系数实际上是中心化与标准化后的协方差

$$r=\frac{s_{xy}}{\sqrt{s_{xx}}\sqrt{s_{yy}}}$$

29.2 相关系数的统计推断

$R^2 = RegSS/TotalSS$ 与 summary 中 Multiple R-Squared 一致. 校正的(Adjusted R-squared) 值的计算方法未知???

```
> R2=RegSS/TotalSS
> R2
[1] 0.7044103
```

R 即为相关系数. Multiple R-Squared 是多重相关, 即y与所有预测变量的回归函数$b_1x_1 + b_2x_2 + \cdots$之间相关系数.

R^2为拟合精确性的指标的汇总(即使用x预测y的精确程度). 注意到 TotalSS = ResSS + RegSS, 得到 $ResSS = TotalSS(1 - R^2)$.

则R^2 可以看作y的方差被x的方差解释的比值. 换句话说, R^2 可以看作y的变异被x的变异解释的比值. 也就是x能够预测y的精确程度.

$R^2 = 1$时, 所有点落在回归线上. y的变异全部被x解释. x能够精确预测y.

$R^2 = 0$时, y的方差与x无关. x不能提供y的任何信息.

29.2.1 相关系数的单样本t检验

在数学上可以证明, 相关系数的单样本t检验等价于回归系数(斜率)的F检验和t检验, 即有相同的p-值.

检验假设

$$\rho = 0 \quad vs \quad \rho \neq 0$$

计算相关系数 r, 然后计算检验统计量

$$t = r\sqrt{n-2}/\sqrt{1-r^2}$$

零假设成立时, t服从n-2自由度的t分布. n为配对数.

下面是cor.test和手工计算的结果

```
> x=rnorm(10)
> y=rnorm(10)
> r<-cor(x,y)
[1] -0.2961736
> cor.test(x,y)

       Pearson's product-moment correlation

data:  x and y
t = -0.8771, df = 8, p-value = 0.406
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.7802920  0.4098881
sample estimates:
      cor
-0.2961736

> r*sqrt(8)/sqrt(1-r^2) # 手工计算检验统计量 t
[1] -0.8770552

> summary(lm(y~x)) # t值相同, p-值也相同

Call:
lm(formula = y ~ x)

Residuals:
   Min     1Q  Median     3Q    Max
-1.3977 -0.5536  0.1954  0.7472  0.9695

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.08585    0.32396   0.265    0.798
x           -0.25379    0.28937  -0.877    0.406

Residual standard error: 0.9316 on 8 degrees of freedom
Multiple R-squared: 0.08772,    Adjusted R-squared: -0.02632
```

```
F-statistic: 0.7692 on 1 and 8 DF,  p-value: 0.406
```

29.2.2 相关系数的Fisher变换(Z变换)

有时候, 我们需要检验相关系数是否与一个不为0的值相等. 即检验假设

$$\rho = \rho_0 \quad vs \quad \rho \neq \rho_0$$

如果使用t检验, 那么零假设成立时在非0的ρ下有一个倾斜的分布. 不容易用正态分布近似. Fisher考虑到这个问题, 提出了一个变换使我们可以用正态分布去检验.

相关系数的 Fisher 变换为

$$z = \frac{1}{2} ln(\frac{1+r}{1-r})$$

在零假设成立时近似于正态分布. 均值为

$$z_0 = \frac{1}{2} ln(\frac{1+\rho_0}{1-\rho_0})$$

方差为

$$s^2 = \frac{1}{n-3}$$

r的绝对值很小时, 与z接近, 但是r绝对值大时, z与r相差很大. 故需要z变换.

29.2.3 相关系数差异的单样本z检验

计算相关系数r和其z变换, 然后计算检验统计量

$$\lambda = (z - z_0\sqrt{n-3} \sim N(0,1)$$

例如检验x,y的相关系数是否=-0.5

```
> cor(x,y)
[1] -0.2961736
> r=cor(x,y)
> z=0.5*log((1+r)/(1-r)) # Fisher 变换
> z
[1] -0.30532
> T=(z-(-0.5))*sqrt(7) # 检验统计量
> T
[1] 0.5150748
> 2*(1-pnorm(abs(T))) # p-值接受零假设, 相关系数与-0.5没有显著差异
[1] 0.6065007
> T=(z-0.5)*sqrt(7) # 检验是否等于0.5
> T
[1] -2.130676
> 2*(1-pnorm(abs(T))) # 拒绝等于0.5的假设
[1] 0.03311580
```

29.2.4 相关系数的区间估计

(1) 计算相关系数 r

(2) 然后计算r的Fisher变换$z = \frac{1}{2}ln(\frac{1+r}{1-r})$. ρ的Fisher变换$z_0 = \frac{1}{2}ln(\frac{1+\rho_0}{1-\rho_0})$. 对于$z_0$的双侧$100\% * (1-\alpha)$置信区间为

$$z_1 = z - z_{1-\alpha/2}/\sqrt{n-3}$$

$$z_2 = z + z_{1-\alpha/2}/\sqrt{n-3}$$

(3)ρ的双侧$100\% * (1-\alpha)$置信区间为

$$\rho_1 = \frac{e^{2z_1}-1}{e^{2z_1}+1}$$

$$\rho_2 = \frac{e^{2z_2}-1}{e^{2z_2}+1}$$

下面是一个例子

```
> x=rnorm(100)
> y=rnorm(100)
> cor(x,y)
[1] -0.0320158
> cor.test(x,y)

       Pearson's product-moment correlation

data:  x and y
t = -0.3171, df = 98, p-value = 0.7518
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.2270064  0.1654427
sample estimates:
      cor
-0.0320158

> r=cor(x,y) # 相关系数
> z=0.5*log((1+r)/(1-r)) # r 的 Fisher 变换
> z
[1] -0.03202674
> z1=z-qnorm(0.975)/sqrt(97) # 上侧区间
> z1
[1] -0.2310309
> z2=z+qnorm(0.975)/sqrt(97) # 下侧区间
> z2
[1] 0.1669774
> rho1=(exp(2*z1)-1)/(exp(2*z1)+1) # 变换回r的区间
> rho1
[1] -0.2270064
> rho2=(exp(2*z2)-1)/(exp(2*z2)+1) # 变换回r的区间
> rho2
[1] 0.1654427
```

29.2.5 相关系数的功效及样本量估计

假设对指定的ρ_0检验

$$\rho = 0 \quad vs \quad \rho = \rho_0 > 0$$

单侧及显著性水平为α的检验, 在指定样本量n时

$$power = \Phi(z_0\sqrt{n-3} - z_{1-\alpha})$$

对于指定的$\rho = \rho_0$使用单侧及显著性水平为α, 功效为$1-\beta$的检验所需要的样本量为

$$n = [(z_{1-\alpha} + z_{1-\beta})^2/z_0^2] + 3$$

29.2.6 相关系数的两样本检验

有数据(x1,y1)相关系数为r1, (x2,y2)相关系数为r2. 两样本检验就是检验若 , r2 是否相等. 即

$$\rho_1 = \rho_2 \quad vs \quad \rho_1 \neq \rho_2$$

合理的做法是对相关系数做z变换得到z1, z2, 若变换后的差值$|z1 - z2|$大, 拒绝零假设. 检验统计量

$$T = \frac{z1 - z2}{\sqrt{\frac{1}{n_1-3} + \frac{1}{n_2-3}}} \sim N(0,1)$$

假设数据为正态数据.

29.3 偏相关

假设我们研究两个变量x与y的线性相关程度. 但是在控制其它协变量$z_1, \cdots, z_k$之后. 如下两个派生的变量之间的 Pearson 相

关称为x,y的偏相关(partial correlation). e_x=x在$z_1, \cdots, z_k$上的线性回归的残差. e_y=y在$z_1, \cdots, z_k$上的线性回归的残差.

下面是计算偏相关的例子.

```
library(corpcor)
x=rep(10,10)+rnorm(10)
y=x+rnorm(10)
z=y+rnorm(10)
m=matrix(c(x,y,z),nc=3,nr=10)

> cor(m) # 相关矩阵
          [,1]      [,2]      [,3]
[1,] 1.0000000 0.8241164 0.8025136
[2,] 0.8241164 1.0000000 0.9126755
[3,] 0.8025136 0.9126755 1.0000000
> cor2pcor(cov(m)) # 偏相关矩阵
          [,1]      [,2]      [,3]
[1,] 1.0000000 0.3759994 0.2175614
[2,] 0.3759994 1.0000000 0.7436428
[3,] 0.2175614 0.7436428 1.0000000
> cov2cor(cov(m)) # 也是相关矩阵
          [,1]      [,2]      [,3]
[1,] 1.0000000 0.8241164 0.8025136
[2,] 0.8241164 1.0000000 0.9126755
[3,] 0.8025136 0.9126755 1.0000000

# 手工计算偏相关系数
> e3=lm(x~z)$res # x在z上的线性回归残差
> e4=lm(y~z)$res # y在z上的线性回归残差
# x y的偏相关系数(控制z后) 与 cor2pcor(cov(m)) 结果一致
> cor(e3,e4)
[1] 0.3759994

# 下面看看相反的残差的相关
> e1=lm(z~x)$res #z在x上的线性回归残差
> e2=lm(z~y)$res #z在y上的线性回归残差
> cor(e1,e2)
[1] 0.5192303
```

29.4 多元相关

多元相关或复相关是 m 个自变量和因变量的总相关系数.

y与所有预测变量的回归函数$b_1x_1 + b_2x_2 + \cdots$之间相关系数称为y与$x_1, x_2, \cdots$的多重相关.

summary(lm(y x)) 结果中的 Multiple R-Squared 就是多元相关系数.

```
x = 1:10
y = x+10+rnorm(10)
z = x+y
l = lm(z~x+y) # 多元回归结果中 Multiple R-Squared 即多元相关
```

如果确定没有截距(intercept), 则可以加一个 -1 来表示, 即

```
l = lm(z~x+y-1)
```

29.5 其他相关

Pearson Spearman Kendall 等相关系数参考秩相关 48.5 和关联性(相依性)度量 47.5

Chapter 30

回归诊断

回归完成后应该立即做回归诊断, 看看回归效果如何. summary结果里包含了t检验.

下面是其它检验

回归诊断相关的函数

```
influence.measures rstandard rstudent dffits
cooks.distance    dfbeta   dfbetas  covratio
hatvalues         hat
```

30.1 图的威力

一些基本的图

```
x1=1:100
y=1:100+rnorm(100)*10
res=lm(y~x1) # 回归

# 残差图
```

```
plot(res$residual)
hist(res$residual) # 残差是否正态分布
boxplot(res$resdual)
plot(res$res ~ res$fitted.values) # 拟合值与残差作图

# 预测及上下界
pre=predict(res, interval='confidence') # 预测的直线
# 绘图
plot(y~x1) # 绘图在一起
lines(pre[,1]) # abline(res) 也可以
lines(pre[,2], col='red') # 上界
lines(pre[,3], col='red') # 下界



Anscombe <-data.frame(
  X=c(10.0, 8.0, 13.0, 9.0, 11.0, 14.0, 6.0, 4.0, 12.0, 7.0, 5.0),
  Y1=c(8.04,6.95, 7.58,8.81,8.33,9.96,7.24,4.26,10.84,4.82,5.68),
  Y2=c(9.14,8.14, 8.74,8.77,9.26,8.10,6.13,3.10, 9.13,7.26,4.74),
  Y3=c(7.46,6.77,12.74,7.11,7.81,8.84,6.08,5.39, 8.15,6.44,5.73),
  X4=c(rep(8,7), 19, rep(8,3)),
  Y4=c(6.58,5.76,7.71,8.84,8.47,7.04,5.25,12.50, 5.56,7.91,6.89)
)

 lm1=lm(Y1~X, data=Anscombe)
 lm2=lm(Y2~X, data=Anscombe)
 lm3=lm(Y3~X, data=Anscombe)
 lm4=lm(Y4~X4,data=Anscombe)

# 下面是系数部分的结果

# 拟合比较好
> summary(lm(Y1~X, data=Anscombe))

            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.0001     1.1247   2.667  0.02573 *
X             0.5001     0.1179   4.241  0.00217 **

# 实际上是曲线
> summary(lm(Y2~X, data=Anscombe))
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    3.001      1.125   2.667  0.02576 *
X              0.500      0.118   4.239  0.00218 **

# 有极端值的干扰
> summary(lm(Y3~X, data=Anscombe))

            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.0075     1.1244   2.675  0.02542 *
X             0.4994     0.1179   4.237  0.00218 **

# 分布很不均匀
> summary(lm(Y4~X4,data=Anscombe))

            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.0017     1.1239   2.671  0.02559 *
X4            0.4999     0.1178   4.243  0.00216 **

# 绘图查看
par(mfrow=(c(2,2)))
plot(Y1~X, data=Anscombe)
abline(lm(Y1~X, data=Anscombe))
plot(Y2~X, data=Anscombe)
abline(lm(Y2~X, data=Anscombe))
plot(Y3~X, data=Anscombe)
abline(lm(Y3~X, data=Anscombe))
plot(Y4~X4, data=Anscombe)
abline(lm(Y4~X4, data=Anscombe))
```

从图中看到, 只有第一组拟合是好的. 而单纯的回归结果都是显著的.

第二组可能是二次或更高次数的多项式.

```
> lm2.sol<-lm(Y2~X+I(X^2), data=Anscombe); summary(lm2.sol)

Call:
lm(formula = Y2 ~ X + I(X^2), data = Anscombe)
```

```
Residuals:
      Min        1Q    Median        3Q       Max
-0.0013287 -0.0011888 -0.0006294  0.0008741  0.0023776

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.9957343  0.0043299   -1385   <2e-16 ***
X            2.7808392  0.0010401    2674   <2e-16 ***
I(X^2)      -0.1267133  0.0000571   -2219   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘

Residual standard error: 0.001672 on 8 degrees of freedom
Multiple R-squared:    1,     Adjusted R-squared:    1
F-statistic: 7.378e+06 on 2 and 8 DF,  p-value: < 2.2e-16

> attach(Anscombe)
> plot(Y2~X)

# 下面绘制预测值曲线
> o <- order(X)
> Y2.pre <- predict(lm2.sol)
> Y2.pre.o<-Y2.pre[o]
> lines(X.o,Y2.pre.o,col="red")
```

第三组的异常值需要手工去除.

```
> i<-1:11; Y31<-Anscombe$Y3[i!=3]; X3<-Anscombe$X[i!=3]
>  lm3.sol<-lm(Y31~X3); summary(lm3.sol)

Call:
lm(formula = Y31 ~ X3)

Residuals:
      Min        1Q    Median        3Q       Max
-0.0060173 -0.0012121 -0.0010173 -0.0008225  0.0140693

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept) 4.0106277  0.0057115   702.2   <2e-16 ***
X3          0.3450433  0.0006262   551.0   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.006019 on 8 degrees of freedom
Multiple R-squared:     1,      Adjusted R-squared:     1
F-statistic: 3.036e+05 on 1 and 8 DF,  p-value: < 2.2e-16
```

最后数据没有足够的信息来判断. 它对单个的样本非常依赖, 这可能不是一个综合的分析.

30.2 残差及其检验

最小二乘法求回归模型时, 对残差的要求是独立等方差的.

30.2.1 简介 plot.lm()

用法为 which 为 1 是画普通残差与拟合值, 2 为正态 QQ 的残差图, 3 为标准化残差开方与拟合值的残差图, 4 为 Cook 统计量的残差图.

```
plot(x, which = c(1:3,5),
    caption = c("Residuals vs Fitted", "Normal Q-Q",
        "Scale-Location", "Cook's distance",
        "Residuals vs Leverage", "Cook's distance vs Leverage"),
    panel = if(add.smooth) panel.smooth else points,
    sub.caption = NULL, main = "",
    ask = prod(par("mfcol")) < length(which) && dev.interactive(),
    ...,
    id.n = 3, labels.id = names(residuals(x)), cex.id = 0.75,
    qqline = TRUE, cook.levels = c(0.5, 1.0),
    add.smooth = getOption("add.smooth"), label.pos = c(4,2),
    cex.caption = 1)
```

30.2.2 普通残差

设线性回归模型为

$$Y = Xb + \epsilon$$

Y 为 n 维向量, X 为 $n * (p+1)$ 阶设计矩阵, b 为 $p+1$ 向量. ϵ 为 n 维误差向量.

回归系数的估计为

$$\hat{b} = (X^T X)^{-1} X^T Y$$

拟合值

$$\hat{Y} = X\hat{b} = X(X^T X)^{-1} X^T Y = HY$$

其中

$$H = X(X^T X)^{-1} X^T$$

称 H 为帽子矩阵[1]

残差为

$$\hat{\epsilon} = Y - \hat{Y} = (I - H)Y$$

residuals() resid() 计算模型的残差. 然后我们可以对残差做正态检验

```
# 对第一组数据检验残差符和正态分布
> lm.1=lm(Y1~X, data=Anscombe)
> r1=resid(lm.1)
> shapiro.test(r1)

        Shapiro-Wilk normality test

data:  r1
W = 0.9421, p-value = 0.5456
```

[1]因为Y被H左乘后变为$\hat{Y}$, 由此得名

```
# 第三组数据的残差不符合正态分布
> lm.3=lm(Y3~X, data=Anscombe)
> r3=resid(lm.3)
> shapiro.test(r3)

        Shapiro-Wilk normality test

data:  r3
W = 0.7406, p-value = 0.00157

# 再次次绘图查看
> plot(Y3~X, data=Anscombe)

# 去掉极端值(第 3 个), 残差还是不能满足正态分布
>  i<-1:11; Y31<-Anscombe$Y3[i!=3]; X3<-Anscombe$X[i!=3]
> lm31<-lm(Y31~X3)
> r31<-resid(lm31)
> shapiro.test(r31)

        Shapiro-Wilk normality test

data:  r31
W = 0.7615, p-value = 0.004931

# 绘图查看, 发现第 9 个值异常
> plot(r31)

# 去掉第 9 个值, 残差符合正态分布
> i<-1:10; Y32<-Y31[i!=9]; X32<-X3[i!=9]
> lm32<-lm(Y32~X32)
> r32<-resid(lm32)
> shapiro.test(r32)

        Shapiro-Wilk normality test

data:  r32
W = 0.9839, p-value = 0.9813
```

30.2.3 标准化(内学生化)残差

由差向量 ϵ 的性质得到

$$E(\hat{\epsilon}) = 0,\ Var(\hat{\epsilon}) = \sigma^2(I - H)$$

故对每个 $\hat{\epsilon}$

$$\frac{\hat{\epsilon}_i}{\sigma\sqrt{1 - h_{ii}}} \sim N(0, 1)$$

其中 h_{ii} 是 H 对角线上的元素. 称

$$r_i = \frac{\hat{\epsilon}_i}{\hat{\sigma}\sqrt{1 - h_{ii}}}$$

为标准化残差(standardized residual), 也叫做内学生化残差(internally studentized residual). 因为 σ^2 的估计用到了包括第i个样本在内的全部数据. $r_i \sim N(0, 1)$

函数 rstandard() 计算标准化残差.

```
> rstandard(lm32)
       1        2        3        4        5        6        7
 0.0576117  0.2864334 -1.5984163  1.7677670 -0.4926925  0.5404789  0.8784586
       8        9
-0.1835970 -1.2598816
```

30.2.4 外学生化残差

记删除第i个样本数据后, 由余下的数据求得的回归系数为 $\hat{b}_{(i)}$, 则 σ^2 的估计为

$$\hat{\sigma}^2_{(i)} = \frac{1}{n - p - 2}\sum_{j \neq i}(Y_i - X_j\hat{b}_{(i)})^2$$

其中 X_j 为设计矩阵 X 的第j行. 称

$$\hat{\epsilon}_i(\hat{b}_{(i)}) = \frac{\hat{\epsilon}_i}{\hat{\sigma}_{(i)}\sqrt{1-h_{ii}}}$$

为学生化残差, 或称为外学生化残差(externally studentized residual).

函数 rstudent() 计算标准化残差.

```
> rstudent(lm32)
        1         2         3         4         5         6
 0.05335072  0.26675362 -1.85706229  2.19970673 -0.46426557  0.51116565
        7         8         9
 0.86220749 -0.17038855 -1.32647306
```

30.2.5 残差图

以残差 $\hat{\epsilon}_i$ 为纵坐标, 拟合值 $\hat{y}_i$ 或对应的数据观测序号i或观测时间为横坐标的散点图统称为残差图. 残差图是进行模型诊断的重要工具. (可以直接使用 plot.lm() 函数绘制)

下面我们绘制第一组回归(拟合比较好, 残差服从正态分布) lm1 的残差图和标准化残差图

```
> fit1=fitted(lm1) # predict(lm1) 也可以

# 残差图
> r1=resid(lm1);
> plot(r1~fit1)

# 标准化残差图
> rst1=rstandard(lm1)
> plot(rst1~fit1)

# 两个图画在一起
```

```
> par(mfrow=(c(1,2)))
> plot(r1~fit1)
> plot(rst1~fit1)
```

对于标准化残差, 应该有大约 95% 的样本落入$[-2,2]$之间, 则若拟合值 $\hat{Y}$ 为横坐标, 那么标准化残差大概落入 $[-2,2]$ 内, 且不呈现任何趋势. 否则回归模型可能有问题.

下面看第二组回归(曲线) lm2 . 可以看到, 曲线回归后残差图变好.

```
> rst2=rstandard(lm2)
> fit2=fitted(lm2)

# 曲线回归
> lm2curve=lm(Y2~X+I(X^2), data=Anscombe)

> rst2c=rstandard(lm2curve)
> fit2c=fitted(lm2curve)

# 绘制残差图
> par(mfrow=(c(1,2)))
> plot(rst2~fit2)
> plot(rst2c~fit2c)
```

30.2.6 残差的 Q-Q 图

可以使用 QQ 图检验残差的正态性.

设 $\hat{\epsilon}_{(i)}, i = 1, 2, \cdots, n$ 是残差的次序统计量, 而

$$q_{(i)} = \Phi^{-1}(\frac{i - 0.375}{n + 0.25})$$

为 $\hat{\epsilon}_{(i)}$ 的期望值. 其中 Φ^{-1} 为标准正态分布的反函数.

可以证明, 若 $\hat{\epsilon}_{(i)}$ 来自正态分布, 则 $(q_{(i)}, \hat{\epsilon}_{(i)})$ 应该在一条直线上. 若明显不在直线, 那么怀疑 $\hat{\epsilon}_{(i)}$ 是否为正态分布.

R 中直接使用 plot(lm, 2) 即可

30.3 影响分析

所谓影响分析就是探查对估计有异常大影响的数据. 例如第三组数据.

如果一个样本不遵守某个模型, 但是其余遵守, 称这个样本为强影响点(异常值点).

影响分析的重要功能就是区分这样的点.

30.3.1 帽子矩阵H的对角元素

从前面可以得到, $\hat{Y} = HY$, $\hat{Y}$ 是 Y 在 X 的列向量张成的子空间内的投影, 且有

$$\frac{\partial \hat{Y_i}}{\partial Y_i} = h_{ii}$$

h_{ii} 为 H 的对角元素.

故 h_{ii} 的大小可以表示第i个样本对 $\hat{Y_i}$ 的影响力. 考虑 $\hat{Y_i}$ 的方差

$$var(\hat{Y_i}) = h_{ii}\sigma^2$$

故 h_{ii} 也反映了回归值的波动情况.

由投影矩阵 H 的性质得到

$$0 <= h_{ii} <= 1$$
$$\sum H_{ii} = p + 1$$

所以 Hoaglin 和 Welsch(1978) 给出一致判断异常值的方法, 当

$$h_{i_0 i_0} >= \frac{2(p+1)}{n}$$

可以认为第 i_0 组样本影响较大, 结合其它准则, 可以考虑是否剔除.

由于 H 的对角元素是很重要的统计信息, 故 R 也有计算函数 hatvalues() 和 hat()

```
> hatvalues(lm1)
        1         2         3         4         5         6         7         8
0.1000000 0.1000000 0.2363636 0.0909091 0.1272727 0.3181818 0.1727273 0.3181818
        9        10        11
0.1727273 0.1272727 0.2363636
```

30.3.2 DFFITS 准则

Belsley, Kuh 和 Welsch(1980) 给出另外一致准则, 计算统计量

$$D_i(\sigma) = \sqrt{\frac{h_{ii}}{1-h_{ii}}} \frac{\epsilon_i}{\sigma\sqrt{1-h_{ii}}}$$

对第i个样本, 如果有

$$|D_i(\sigma)| > 2\sqrt{\frac{p+1}{n}}$$

则认为第i个样本影响比较大, 应引起注意.

R 中的函数为 dffits

```
> p=1
```

```
> n=nrow(Anscombe)

# 第三组的第三个样本可能异常
> d <- dffits(lm3)
> d> 2*sqrt((p+1)/n)
   1    2    3    4    5    6    7    8    9   10   11
FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
> which(d> 2*sqrt((p+1)/n))
3
3
```

30.3.3 Cook 统计量

Cook 在 1977 年提出了 Cook 统计量, 定义为

$$D_i = \frac{(\hat{b} - \hat{b}_{(i)})^T X^T X (\hat{b} - \hat{b}_{(i)})}{(p+1)\hat{\sigma}^2}$$

其中 $\hat{b}_{(i)}$ 为删除第i个样本数据后由余下的 $n-1$ 个样本数据求得回归系数. 经过推导, Cook 统计量可以改写为

$$D_i = \frac{1}{p+1}(\frac{h_{ii}}{1-h_{ii}})r_i^2$$

R 中 cooks.distance() 计算 Cook 统计量

```
> cooks.distance(lm3)
         1          2          3          4          5          6
0.0118305891 0.0021827101 1.3928277909 0.0055254398 0.0260716064 0.3006335925
         7          8          9         10         11
0.0004804045 0.0331943873 0.0596504117 0.0002176290 0.0067519721
```

直观上, Cook 统计量越大的点月可能是异常点, 但是判定异常值点的临界值的选择是很困难的, 应用中视具体情况而定.

30.3.4 COVARATIO 准则

利用全部样本回归系数的估计值的协方差矩阵为

$$var(\hat{b}) = \sigma^2(X^TX)^{-1}$$

去掉第i个样本点的回归系数的估计值的协方差矩阵为

$$var(\hat{b}_{(i)}) = \sigma^2_{(i)}(X^T_{(i)}X_{(i)})^{-1}$$

其中 $X_{(i)}$ 为 X 剔除第i行得到的矩阵.

考虑其协方差的比

$$covratio = \frac{det(\sigma^2_{(i)}(X^T_{(i)}X_{(i)})^{-1})}{det(\sigma^2(X^TX)^{-1})} = \frac{(\hat{\sigma}^2_{(i)})^{p+1}}{(\hat{\sigma}^2)^{p+1}}\frac{1}{1-h_{ii}}$$

如果一个样本点对应的 covratio 值离开 1 越远, 则认为哪个样本点的影响越大.

```
> covratio(lm3)
         1          2          3          4          5          6
1.340490e+00 1.393999e+00 7.360047e-10 1.358209e+00 1.337257e+00 1.362816e+00
         7          8          9         10         11
1.528312e+00 1.798031e+00 1.341787e+00 1.449234e+00 1.641337e+00

# 绘图查看, 第3个样本点接近 0.
> plot(covratio(lm3))
```

30.3.5 总结

influence.measures()可以作为诊断分析的概括. 返回每个样本的 dfbeta, dffit, cook,等统计量, 星号*是可能异常的点

```
> influence.measures(lm3)
Influence measures of
       lm(formula = Y3 ~ X, data = Anscombe) :

     dfb.1_    dfb.X   dffit    cov.r   cook.d   hat inf
1  -4.64e-03 -4.43e-02  -0.1468 1.34e+00 0.011831 0.100
2  -3.75e-02  1.88e-02  -0.0624 1.39e+00 0.002183 0.100
3  -1.83e+02  2.69e+02 342.7851 7.36e-10 1.392828 0.236   *
4  -3.31e-02 -2.11e-18  -0.0997 1.36e+00 0.005525 0.091
5   4.92e-02 -1.17e-01  -0.2197 1.34e+00 0.026072 0.127
6   4.90e-01 -6.67e-01  -0.7898 1.36e+00 0.300634 0.318
7   2.60e-02 -2.01e-02   0.0292 1.53e+00 0.000480 0.173
8   2.39e-01 -2.07e-01   0.2449 1.80e+00 0.033194 0.318   *
9   1.38e-01 -2.32e-01  -0.3365 1.34e+00 0.059650 0.173
10 -1.54e-02  1.05e-02  -0.0197 1.45e+00 0.000218 0.127
11  1.04e-01 -8.62e-02   0.1098 1.64e+00 0.006752 0.236
```

30.4 共线性,条件数,kappa()函数

当自变量彼此相关时, 某变量可能会因为其它变量的改变而改变其效应, 甚至改变符号. 自变量彼此相关称为共线性或多重共线性. 若出现共线性, 建议使用主成分回归.

30.4.1 什么是共线性

如果存在某些常数 c0,c1,c2 使得线性等式

$$c1X1 + c2X2 = c0$$

对数据成立, 则称两个变量 X1,X2 共线性.

精确共线性较少发生. 故若上式近似成立, 则两个变量 X1,X2 近似共线性.

常用单不完全合式的共线性度量为 X1,X2 相关系数的平方 r_{12}^2. 精确共线性对应 $r_{12}^2 = 1$, 非共线性对应 $r_{12}^2 = 0$

对 p 个自变量, 若有

$$c_1X_1 + c_2X_2 + \cdots + c_pX_p = c_0$$

近似成立, 称p个变量共线性.

30.4.2 共线性的发现

将 $X_1, X_2, \cdots, X_p$ 中心化和标准化得到 $X = x_{(1)}, x_{(2)}, \cdots, x_{(p)}$. 设 λ 为 X^TX (本质上就是$X_1, X_2, \cdots, X_p$ 的相关矩阵) 的一个特征值, φ 为对应的特征向量, 其长度为 1, 即 $\varphi^T\varphi = 1$. 若 $\lambda \approx 0$ 则

$$X^TX\varphi = \lambda\varphi \approx 0$$

用 φ^T 左乘上式, 得到

$$\varphi^TX^TX\varphi = \lambda\varphi^T\varphi = \lambda \approx 0$$

故有

$$X\varphi \approx 0$$

即

$$\varphi_1x_{(1)} + \varphi_2x_{(2)} + \cdots + \varphi_px_{(p)} \approx 0$$

表明向量 $X = x_{(1)}, x_{(2)}, \cdots, x_{(p)}$ 之间有近似线性关系. 那么 $X_1, X_2, \cdots, X_p$ 之间存在共线性. 其中

$$\varphi = (\varphi_1, \varphi_2, \cdots, \varphi_p)$$

度量共线性严重程度的一个重要指标为方阵 X^TX 的条件数

$$\kappa(X^TX) = \| X^TX \| * \| (X^TX)^{-1} \| = \frac{\lambda_{max}(X^TX)}{\lambda_{min}(X^TX)}$$

其中 $\lambda_{max}, \lambda_{min}$ 表示方阵 X^TX 的最大最小特征值.

直观上, 条件数 κ 刻画了方阵 X^TX 的特征值差异的大小. 经验上,

$$\begin{aligned} \kappa < 100, &\text{共线性程度很小} \\ 100 <= \kappa <= 1000, &\text{共线性程度中等} \\ \kappa > 1000, &\text{共线性程度严重} \end{aligned}$$

R 中函数 kappa() 计算矩阵的条件数. 下面的数据中自变量有 6 个, 每个有 12 个样本. 除第一个样本外, 其它 11 个满足

$$X_1 + X_2 + X_3 + X_4 = 10$$

```
d <- data.frame(
  Y=c(10.006, 9.737, 15.087, 8.422, 8.625, 16.289,
      5.958, 9.313, 12.960, 5.541, 8.756, 10.937),
  X1=rep(c(8, 0, 2, 0), c(3, 3, 3, 3)),
  X2=rep(c(1, 0, 7, 0), c(3, 3, 3, 3)),
  X3=rep(c(1, 9, 0), c(3, 3, 6)),
  X4=rep(c(1, 0, 1, 10), c(1, 2, 6, 3)),
  X5=c(0.541, 0.130, 2.116, -2.397, -0.046, 0.365,
      1.996, 0.228, 1.38, -0.798, 0.257, 0.440),
  X6=c(-0.099, 0.070, 0.115, 0.252, 0.017, 1.504,
      -0.865, -0.055, 0.502, -0.399, 0.101, 0.432)
)

# X^T X 本质上是原始矩阵的相关矩阵
> c <-cor(d[2:7])
> kappa(c,exact=T)
[1] 2195.908
```

计算特征值与特征向量

```
> eigen(c)
$values
[1] 2.428787365 1.546152096 0.922077664 0.793984690 0.307892134 0.001106051

$vectors
```

```
         [,1]       [,2]       [,3]       [,4]       [,5]        [,6]
[1,] -0.3907189  0.33968212  0.67980398 -0.07990398  0.25103700 -0.447679719
[2,] -0.4556030  0.05392140 -0.70012501 -0.05768633  0.34446553 -0.421140280
[3,]  0.4826405  0.45332584 -0.16077736 -0.19102517 -0.45363721 -0.541689124
[4,]  0.1876590 -0.73546592  0.13587323  0.27645223 -0.01520870 -0.573371872
[5,] -0.4977330  0.09713874 -0.03185053  0.56356440 -0.65128338 -0.006052127
[6,]  0.3519499  0.35476494 -0.04864335  0.74817535  0.43374630 -0.002166594

# 手工计算特征值
> e <- eigen(c)
> max(e$values)/min(e$values)
[1] 2195.908
```

最小的特征值$\lambda_{min} = 0.001106$, 对应的特征向量为

```
]
# e$vectors 的最后一列
> e$vectors[,which(e$values==min(e$values))]
[1] -0.447679719 -0.421140280 -0.541689124 -0.573371872 -0.006052127
[6] -0.002166594
```

则有

$$0.4476x_{(1)} + 0.4211x_{(2)} + 0.5417x_{(3)} + 0.5734x_{(4)} + 0.006052x_{(5)} + 0.002167x_{(6)} \approx 0$$

由于 $x_{(5)}, x_{(6)}$ 系数近似为 0, 故

$$0.4476x_{(1)} + 0.4211x_{(2)} + 0.5417x_{(3)} + 0.5734x_{(4)} \approx 0$$

所以存在

$$c_1X_1 + c_2X_2 + c_3X_3 + c_4X_4 \approx c_0$$

变量 X_1, X_2, X_3, X_4共线性.

另, kappa() 也可以计算线性模型的共线性, 但是计算的是 $X_1, X_2, X_3, \cdots, X_p, Y$构成的矩阵的条件数, 即

$$kappa(lm.model) = \kappa([X_1, X_2, X_3, \cdots, X_p, Y])$$

```
> kappa(lm3)
[1] 32.14227
```

Chapter 31

逐步回归

参考文献 [21] 下册 6.4 逐步回归.

31.1 是否拟合的足够好?

基本原理: 通过残差来看. 若模型恰当, 则$\hat{\sigma}$是σ的无偏估计. 若模型过于复杂, 即过拟合, 则$\hat{\sigma} < \sigma$. 若模型太简单, 则$\hat{\sigma} > \sigma$. 这时候, spline, lowess 等可以帮助你查看非线性关系

```
> lines(smooth.spline(x,y), col='red', lwd=2)
> lines(lowess(x,y), col='blue', lwd=2)
```

进一步的问题是什么时候才叫做合适? 这个问题很难回答, 也具有根本性. 拟合过火和不足, 都会导致预测能力下降. 有时候, 我们需要放弃线性和多项式模型, 转而寻找其它的方法(bayes, ann等)

31.1.1 σ^2已知

回忆

$$\frac{\hat{\sigma^2}}{\sigma^2} \sim \frac{\chi^2_{n-p}}{(n-p)}$$

那么若

$$\frac{(n-p)\hat{\sigma^2}}{\sigma^2} > \chi^2_{n-p,1-\alpha}$$

则拟合的不好. 这时我们需要一个更好的模型.

```
> summary(res)$sigma # 即 Residual standard error
[1] 246.0147
> 1-pchisq(summary(res)$sigma^2*44,44) # 此值过小说明拟合不
好
[1] 0
```

31.1.2 过拟合

模型过于复杂或样本过简单, 都可以导致过拟合.

```
> x <- seq(0,1,length=n)
> y <- 1-2*x+.3*rnorm(n)
> summary(lm(y~poly(x,10)))
错误在poly(x, 10) : 'degree'小于数据点的数目
> n
[1] 10
> summary(lm(y~poly(x,3)))

Call:
lm(formula = y ~ poly(x, 3))
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-0.52663 -0.05985  0.01600  0.10827  0.50769

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.1193     0.1036   1.151 0.293399
poly(x, 3)1  -2.2960     0.3277  -7.006 0.000422 ***
poly(x, 3)2  -0.1015     0.3277  -0.310 0.767343
poly(x, 3)3  -0.1751     0.3277  -0.534 0.612402
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3277 on 6 degrees of freedom
Multiple R-Squared: 0.8918,     Adjusted R-squared: 0.8377
F-statistic: 16.49 on 3 and 6 DF,  p-value: 0.002654
```

31.1.3 欠拟合

原因是模型太简单. 下面是一个欠拟合的例子.

```
> x <- runif(100, -1, 1)
> y <- 1-x+x^2+.3*rnorm(100)
> plot(y~x)
> abline(lm(y~x), col='red')
```

31.2 外推

我们经常想把数据外推(Extrapolation), 例如我们已知的数据范围是(0,1). 但是我们想知道数据在(0,10)的表现. 下面是几个常见的问题.

首先预测区间会线性变大. 下面是一个例子.数据范围是(-3,3), 我们看看(-20,20)是什么样子.

```
> n=20
> x <- rnorm(n)
> y <- 1 - 2*x - .1*x^2 + rnorm(n)
> plot(y~x, xlim=c(-20,20), ylim=c(-30,30)) # 绘出数据
> r <- lm(y~x)
> abline(r, col='red') # 绘出回归线
> xx <- seq(-20,20,length=100)
> p <- predict(r, data.frame(x=xx), interval='prediction') # 预测
值
> lines(xx,p[,2],col='blue') # 绘出上界
> lines(xx,p[,3],col='blue') # 绘出下界
```

有时候, 数据局部可能是接近线性的, 但是整体不是. 这个时候使用局部来预测整体就很危险. 接上个例子的数据

```
> yy <- 1 - 2*xx - .1*xx^2 + rnorm(n)
> points(yy~xx)
```

31.3 最优回归方程的选择

实际问题中影响因变量 y 的因素很多, 人们可以从中挑选若干建立回归方程. 若忽略了对 y 有显著影响的自变量, 那么误差就会很大. 若变量选择过多, 使用就不方便, 且当有的自变量对 y 的影响不大时, 可能因为自由度减小而对误差的估计变大, 从而影响预测精度.

那么如何选择自变量呢? 在不同的最优准则下可以有不同的选择. (即对 y 有显著影响的被选择, 影响不大的排除掉) 有很多方法可以获得最优回归方程, 例如, "一切子集回归法", "前进法", "后退法", "逐步回归法" 等. 其中 "逐步回归法" 由于计算机程序渐变, 使用也较为普遍.

31.4 逐步回归的计算

R 提供的 step(), add1(), drop1() 等函数可以实现. 其中 step() 以 AIC 信息统计量为准则, 通过下最小的 AIC 来达到删除或增加变量的目的. step() 的格式为

```
step(object, scope, scale = 0,
    direction = c("both", "backward", "forward"),
    trace = 1, keep = NULL, steps = 1000, k = 2, ...)
```

参数 object 主要为 lm 和 glm. (stepAIC in package 'MASS' 有关于 scope 用法的例子). 参数 direction 为 "both" 是 "一切子集回归法", " forward" 是 "前进法", "backward" 是 "后退法".

下面是一个例子. X1,X2,X3,X4 为水泥中的四种成分. Y 为凝固时释放的热量. 我们希望寻找其线性关系. 首先做线性回归.

```
cement<-data.frame(
  X1=c( 7,  1, 11, 11,  7, 11,  3,  1,  2, 21,  1, 11, 10),
  X2=c(26, 29, 56, 31, 52, 55, 71, 31, 54, 47, 40, 66, 68),
  X3=c( 6, 15,  8,  8,  6,  9, 17, 22, 18,  4, 23,  9,  8),
  X4=c(60, 52, 20, 47, 33, 22,  6, 44, 22, 26, 34, 12, 12),
  Y =c(78.5, 74.3, 104.3,  87.6,  95.9, 109.2, 102.7, 72.5,
      93.1,115.9,  83.8, 113.3, 109.4)
)

> lm.sol<-lm(Y ~ X1+X2+X3+X4, data=cement)
> summary(lm.sol)

Call:
lm(formula = Y ~ X1 + X2 + X3 + X4, data = cement)

Residuals:
   Min     1Q  Median     3Q    Max
-3.1750 -1.6709  0.2508  1.3783  3.9254

Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  62.4054    70.0710   0.891   0.3991
X1            1.5511     0.7448   2.083   0.0708 .
X2            0.5102     0.7238   0.705   0.5009
X3            0.1019     0.7547   0.135   0.8959
X4           -0.1441     0.7091  -0.203   0.8441
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.446 on 8 degrees of freedom
Multiple R-squared: 0.9824,     Adjusted R-squared: 0.9736
F-statistic: 111.5 on 4 and 8 DF,  p-value: 4.756e-07
```

看到系数都不显著, 效果不是很好. 下面使用 step() 来逐步回归.

```
> lm.step<-step(lm.sol)
Start:  AIC=26.94
Y ~ X1 + X2 + X3 + X4

       Df Sum of Sq    RSS    AIC
- X3    1     0.109 47.973 24.974
- X4    1     0.247 48.111 25.011
- X2    1     2.972 50.836 25.728
<none>              47.864 26.944
- X1    1    25.951 73.815 30.576

Step:  AIC=24.97
Y ~ X1 + X2 + X4

       Df Sum of Sq    RSS    AIC
<none>              47.97  24.97
- X4    1      9.93  57.90  25.42
- X2    1     26.79  74.76  28.74
- X1    1    820.91 868.88  60.63
```

start 步中, 全部变量回归时, AIC 为 26.94. 如果去掉 X3, AIC 变为 24.97, 去掉 X4, AIC 为 25.01. 去掉 X2 AIC 为 25.73, 去掉 X1

AIC 为 30.58. 故第一步完成后判断去掉 X3 AIC 最小, 故得到的模型为

```
# 第一步得到的模型. 去掉 X3
Y ~ X1 + X2 + X4
```

然后使用此模型进行下一轮计算. 在下一轮计算中, 看到无论去掉哪个变量 AIC 值都会增加, 故终止计算, 得到最优回归方程.

下面分析一下回归的显著性

```
> summary(lm.step)

Call:
lm(formula = Y ~ X1 + X2 + X4, data = cement)

Residuals:
   Min     1Q  Median     3Q    Max
-3.0919 -1.8016  0.2562  1.2818  3.8982

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  71.6483    14.1424   5.066 0.000675 ***
X1            1.4519     0.1170  12.410 5.78e-07 ***
X2            0.4161     0.1856   2.242 0.051687 .
X4           -0.2365     0.1733  -1.365 0.205395
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.309 on 9 degrees of freedom
Multiple R-squared: 0.9823,     Adjusted R-squared: 0.9764
F-statistic: 166.8 on 3 and 9 DF,  p-value: 3.323e-08
```

可以看到系数显著性水平有提高, 但是 X2, X4 变量的系数仍然不理想. 如何去做?

我们可以使用 drop1() 函数.(add1() 的用法见在线帮助)

```
> drop1(lm.step)
Single term deletions

Model:
Y ~ X1 + X2 + X4
      Df Sum of Sq    RSS    AIC
<none>              47.97  24.97
X1     1    820.91 868.88  60.63
X2     1     26.79  74.76  28.74
X4     1      9.93  57.90  25.42
```

可以看到, 去掉 X4 AIC 增加最小. 另外残差也是逐步回归的重要指标, 残差越小, 拟合就越好. 去掉 X4 也是残差增加最少的. 下面去掉 X4 做回归

```
> lm.opt<-lm(Y ~ X1+X2, data=cement); summary(lm.opt)

Call:
lm(formula = Y ~ X1 + X2, data = cement)

Residuals:
   Min     1Q Median     3Q    Max
-2.893 -1.574 -1.302  1.362  4.048

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 52.57735    2.28617   23.00 5.46e-10 ***
X1           1.46831    0.12130   12.11 2.69e-07 ***
X2           0.66225    0.04585   14.44 5.03e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.406 on 10 degrees of freedom
Multiple R-squared: 0.9787,     Adjusted R-squared: 0.9744
F-statistic: 229.5 on 2 and 10 DF,  p-value: 4.407e-09
```

看到系数都比较显著. 故最终的回归方程为

```
Y ~ X1 + X2
Y = 52.58 + 1.47*X1 + 0.66 * X2
```

31.5 更新拟合模型

参考[25] page 73, 11.5.

使用 update() 函数. 模型中"."表示旧的模型中对应部分. 下面是一个例子, 使用 x1,x2拟合y, 然后额外增加x3再进行拟合. 进一步, 对响应变量y的平方根变换后再拟合.

```
> x1=rnorm(100)
> x2=rnorm(100)
> x3=rnorm(100)
> y=rnorm(100)
> fm2 <- lm(y ~ x1 + x2)
> fm3 <- update(fm2, . ~ . + x3) # 增加x3再进行拟合
> smf3 <- update(fm3, sqrt(.) ~ .) # 对响应变量y的平方根变换后再拟合.
```

31.6 关于标准化回归系数

31.6.1 其它说法

有关标准化回归系数的说明(选自"百岛潮论坛"http://www.baidao.net/) 标准化回归系数（Beta值）在多元回归中被用来比较变量间的重要性。但是由于重要性这一词意义的含糊性，这一统计常被误用。

有时人们说重要性，是指同样的条件下，哪一个东西更有效。在提高教学质量上，是硬件条重要还是师资更重要？如果是师资更重要，那么同样的物力投在师资上就可以更快地提高教学质量。但是这里要比较的两者必须有同样的测量单位，如成本（元）。如果变量的单位不同，我们不能绝对地说那个变量更重要。不同单位的两个东西是不能绝对地比出高低轻重来。要想进行绝对地比较，就需要两个东西有着共同的测度单位，否则无法比较。　　而标准化回归系数说的重要性则与上面的意义不同，这是一种相对的重要性，与某一特定的情况下，自变量间的离散程度有关。比如说，虽然我们不能绝对地说出教育和年资在决定收入上那一个一定是重要的，但如果现在大家的教育程度比较相似，那么在收入的决定上，工作年数就是决定因素；反之，如果工作年数没有太大区别，那么教育就成为了重要原因。这里的重要性是相对的，是根据不同情况而改变的。再举一个通俗的例子，研究者研究的是遗传因素和后天因素对于人成长的影响。那么在一个社会境遇悬殊巨大的环境中，有人在贫民窟成长，有人在贵族学校上学，那么我们会发现人格的大部分差异会从后天环境因素得到解释，而遗传的作用就相对较小；相反，如果儿童都是在一个相差不大的环境中长大的，你会发现，遗传会解释大部分的人格差异。这种意义上的重要性，不仅与这一自变量的回归系数有关系，而且还与这个自变量的波动程度有关系：如果其波动程度较大，那么就会显得较为重要；否则，就显得不太重要。标准化回归系数正是测量这种重要性的。从标准化回归系数的公式中也可看出，Beta值是与自变量的标准差与成正比的，自变量波动程度的增加，会使它在这一具体情况下的重要性增加。

但是如果将两种重要性混同，就会得到误导性结论。如环境因素的Beta值比遗传因素的Beta值大，就认为在个体的人格发展上应更注意环境因素，而轻视遗传因素，在目前对于Beta值的错误观念非常流行，甚至是一些高手中。　　标准化回归系数的比较结果只是适用于某一特定环境的，而不是绝对正确的，它可能因时因地而变化。举例来说，从某一次数据中得出，在影响人格形成的因素中，环境因素的Beta值比遗传因素的Beta值大，这只能说明数据采集当时当地的情况，而不能加以任何不恰当的推论，不能绝对地不加任何限定地说，环境因素的影响就是比遗传因素大。事实上，如果未来环境因素的波动程度变小，很可能遗传因素就显得更为重要。数据的情况千差万别，变量的相对重要性也可能完全不同但都符合当时的实际情况。

另外一个例子: 回归系数包含了相对应自变量的测量单位的影响，比如：身高x1（cm）、体重x2(kg)对收入y(千元) 进行回归（仅用于说明问题，这种变量关系不合乎实际！1），得到方程y=0.2+0.5x1-0.9x2+e。通常的解释是：在控制体重x2的情况下，身高每增加1cm，导致收入增加0.5千元；同样地，在控制身高x1 的情况下，体重每增加1kg，导致收入减少0.9千元。显然两个变量对收入的贡献的大小是不可比的，因为身高和体重的单位不同。要想比较它们对收入的作用的相对大小，我们需要消除自变量各自单位(即方差大小) 的影响，标准化回归系数正是考虑了这一点才出现。

总之, 标准化回归系数是将自变量的方差标准化后对因变量的影响的大小. 也就是说假设它们的变异/波动程度相同, 那么各自变化1个单位给因变量带来的变化量, 但是这是不符合绝大部分的实际情况的, 实际上单位不同的变量其波动程度很少相同. 单位相同的变量的波动程度很多也不同.

31.6.2 个人认为

如果单位相同, 波动程度相同, 那么就没必要使用标准化了, 直接使用多重回归就可以了. 而标准化相关系数说明的就是这种情况. 所以, 标准化相关系数完全没有必要. 非标准化相关系数可以反映各自对因变量的贡献的大小, 标准化相关系数完全没有必要.

Chapter 32

多项式回归

32.1 模型函数

```
x = 1:10
y = b+b1*x+b2*x^2+...
l = lm(y~x+I(x^2)+I(x^3)+...)
(I函数可以让我们使用通常的方法来表达指数. 注意: 符
号 ^ 在模型中与在表达式中表示不同的意思)
y~poly(x,5) # 可以使用通用的表达方法来拟合,
```

```
>  summary(lm(y~poly(x,5)))

Call:
lm(formula = y ~ poly(x, 5))

Residuals:
      1       2       3       4       5       6       7       8
 0.08011 -0.30545  0.32234  0.06079 -0.12296 -0.29187  0.27484  0.14366
      9      10
-0.22817  0.06669

Coefficients:
          Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)  0.06675    0.10727   0.622  0.56748
poly(x, 5)1 -1.89826    0.33923  -5.596  0.00501 **
poly(x, 5)2 -0.35722    0.33923  -1.053  0.35173
poly(x, 5)3  0.30348    0.33923   0.895  0.42157
poly(x, 5)4 -0.32418    0.33923  -0.956  0.39337
poly(x, 5)5 -0.39118    0.33923  -1.153  0.31307
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

32.2 例子

```
> x = 1:10
> y=10+x+2*x^2+rnorm(10)
> lm(y~x+I(x^2))

Call:
lm(formula = y ~ x + I(x^2))

Coefficients:
(Intercept)            x       I(x^2)
    11.6736       0.4049       2.0484
```

如果确定没有截距(intercept), 则可以加一个 -1 来表示, 即

```
> lm(y~x+I(x^2)-1)

Call:
lm(formula = y ~ x + I(x^2) - 1)

Coefficients:
    x  I(x^2)
4.835   1.697
```

32.3 系数的置信区间(CI)

confint(l)

32.4 F-值, p-值

可以用 summary 查看

32.5 回归值

predict(res)

Chapter 33

广义线性(Generalized Linear)模型

《R导论》[25](page 73)中统计模型部分中有一个广义线性模型对广义线性模型有很好的描述，请参考之。

广义线性建模是线性建模的一种发展，它通过一种简洁而又直接的方式使得线性模型既适合非正态分布的响应值又可以进行线性变换。广义线性模型是基于下面一系列假设前提的：

- 有一个有意义的响应变量y和一系列刺激变量（stimulus variable）x1，x2，. . . 。这些刺激变量决定响应变量的最终分布。
- 刺激变量仅仅通过一个线性函数影响响应值y的分布。该线性函数称为线性预测器（linear predictor），常常写成

$$\eta = \beta_1 x_1 + \beta_2 x_2 + \Delta\Delta\Delta + \beta_p x_p$$

  因此x_i当且仅当$\beta_i = 0$时对y的分布没有影响。

- y分布的形式为

$$f_Y(y;\mu,\phi) = exp[\frac{A}{\phi}\{y\lambda(\mu)`\gamma(\lambda(\mu))\} + \tau(y,\phi)]$$

其中ϕ是尺度参数（scale parameter）(可能已知)，对所有观测恒定，A是一个先验的权重，假定知道但可能随观测不同有所不同，μ是y的均值。也就是说假定y的分布是由均值和一个可能的尺度参数决定的。

- 均值μ是线性预测器的平滑可逆函数（smooth invertible function）：

$$\mu = m(\eta), \eta = m^{`1}(\mu) = L(\mu)$$

其中的反函数(inverse function) L() 被称为关联函数（link function）。

这些假定比较宽松，足以包括统计实践中大多数有用的统计模型，同时也足够严谨，使得可以发展参数估计和统计推论(estimation and inference)中一致的方法（至少可以近似一致）。读者如果想了解这方面最新的进展，可以参考McCullagh, Nelder (1989) 或者Dobson (1990)。

33.1 概念

此部分来自 [21] 6.6 广义线性回归模型

广义线性模型对普通线性模型进行了两个方面的推广, 这些推广允许许多线性模型的方法能够应用于一般问题.

- 通过一个连接函数, 将响应变量的期望$E(y)$与线性自变量联系
- 对误差的分布给出一个误差函数

广义线性模型应有以下三个概念

1. 第i个响应变量的期望$E(y_i)$只能通过线性自变量$B^T x_i$依赖于x_i, $B = (p+1) * 1$的向量, 可能包含截距.
2. 连接函数, 说明线性自变量和$E(y_i)$的关系, 是线性模型的推广.

3. 误差函数, 说明广义线性模型最后一部分的随机成分

我们保留线性模型中样本相互独立的假设, 去掉可加和正态的假设. 可以从指数分布族中选择一个作为误差函数.

下面的表是常见的连接函数和误差函数

	连接函数	逆连接函数(回归模型)	典型误差函数
恒等(identity)	$x^T\beta = E(y)$	$E(y) = x^T\beta$	正态分布
对数	$x^T\beta = lnE(y)$	$E(y) = exp(x^T\beta$	Poisson 分布
Logit	$x^T\beta = LogitE(y)$	$E(y) = \frac{exp(x^T\beta)}{1+exp(x^T\beta)}$	二项分布
逆	$x^T\beta = \frac{1}{E(y)}$	$E(y) = \frac{1}{x^T\beta}$	Gamma 分布

对分布族提供的连接函数见下面一节.

33.2 族

R 提供了一系列广义线性建模工具，从类型上来说包括高斯(gaussian), 二项式(binomial), 泊松(poisson), 逆高斯(inverse gaussian) 和伽马(gamma) 模型的响应变量分布以及响应变量分布无须明确给定的拟似然（quasi-likelihood）模型。在后者，方差函数（variance function）可以由均值的函数指定，但在其它情况下，该函数可以由响应变量的分布得到。

每一种响应分布允许各种关联函数将均值和线性预测器关联起来。这些自动可用的关联函数如下表所示：

族名字	关联函数
binomial	logit, probit, log, cloglog
gaussian	identity, log, inverse
Gamma	identity, inverse, log
inverse.aussian	1/mu^2, identity, inverse, log
poisson	identity, log, sqrt
quasi	logit, probit, cloglog, identity, inverse, log, 1/mu^2, sqrt

这些用于模型构建过程中的响应分布，关联函数和各种其他必要的信息统称为广义线性模型的族（family）。

33.3 glm()函数

既然响应的分布仅仅通过单一的一个线性函数依赖于刺激变量，那么用于线性模型的机制同样可以用于指定一个广义模型的线性部分。但是族必须以一种不同的方式指定。

R用于广义线性回归的函数是glm()，它的使用形式为

```
> fitted .model <- glm(formula, family=family.generator , data=data.frame)
```

和lm()相比，唯一的一个新特性就是描述族的参数family.generator。它其实是一个函数的名字，这个函数将产生一个函数和表达式列表用于定义和控制模型的构建与估计过程。尽管这些内容开始看起来有点复杂，但它们非常容易使用。

这些名字是标准的。程序给定的族生成器可以参族部分表格中的"族名"。当选择一个关联函数时，该关联函数名和族名可以同时在括弧里面作为参数设定。在拟（quasi）家族里面，方差函数也是以这种方式设定。

一些例子可能会使这个过程更清楚。

33.4 gaussian族

gaussian族实际上就是普通线性模型. 命令

```
> fm <- glm(y ~ x1 + x2, family = gaussian, data = sales)
```

和下面的命令结果一致

```
> fm <- lm(y ~ x1+x2, data=sales)
```

但是效率上，前者差一点。注意，高斯族没有自动提供关联函数设定的选项，因此不允许设置参数。如一个问题需要用非标准关联函数的高斯族，那么只能采用我们后面讨论的拟族。

33.5 二项式族(logistic多元线性回归)

即广义线性回归中的二项式回归. 另外请参考流行病学部分的多重logistic回归69有详细的描述。

模型: 因变量为质反应时, 阳性的发生概率为 p, 不发生(阴性)的概率为 1-p. 阳性与阴性概率的比为 p/(1-p), 又称为优势比(odds ratio). 此比例的对数与影响阳性发生率的多个自变量呈线性关系. 可以表示为一个logistic多元线性回归方程.

$$ln(p/(1-p)) = a + b_1 * x_1 + b_2 * x_2 + \cdots + b_k * x_k$$

其中$ln(p/(1-p))$被称作logit变换, 即$logit(p) = ln(p/(1-p))$

则质反应为阳性的概率估计为

$$p = \frac{exp(a + b_1 * x_1 + b_2 * x_2 + \cdots + b_k * x_k)}{1 + exp(a + b_1 * x_1 + b_2 * x_2 + \cdots + b_k * x_k)}$$

阴性的概率估计为

$$1 - p = \frac{1}{exp(a + b_1 * x_1 + b_2 * x_2 + \cdots + b_k * x_k)}$$

回归方程左边的p代替以y, 阳性时取y=1, 阴性时取y=0, 则这个虚变量可以使ln(p/(1-p))与k个自变量的关系用最大似然方法估计回归系数$a, b_1, \cdots, b_k$.

参考 family predict.glm函数

glm中的 family 参数. family 是描述模型的 error 分布和连接函数的. family = binomial 为logistic回归(= gaussian 为一般线性模型)

33.5.1 例1

下面是一个连续变量对0,1应变量的例子.

```
 n <- 100
 x <- c(rnorm(n), 1+rnorm(n))
 y <- c(rep(0,n), rep(1,n)) # y 为0,1数据,代表阴性和阳性质反
应
 plot(y~x)
 abline(lm(y~x),col='red',lty=2) # 线性模型预测
 r <- glm(y~x, family=binomial) # 加入family表明是logistic回归
 xx <- seq(min(x), max(x), length=100)
 yy <- predict(r, data.frame(x=xx)) # 预测时还是线性, 与lm模
型一致
 lines(xx,yy, col='blue', lwd=5, lty=2)
 yy <- predict(r, data.frame(x=xx),type='response') # 加入type='response'预
测为logistic的
 lines(xx,yy, col='blue', lwd=5, lty=2)
> r

Call:  glm(formula = y ~ x, family = binomial)

Coefficients:
(Intercept)          x
   -0.4711      0.8418

Degrees of Freedom: 199 Total (i.e. Null);  198 Residual
Null Deviance:     277.3
Residual Deviance: 249  AIC: 253
```

```
> y.pre <-exp(r$coe[1]+r$coe[2]*xx)/(1+exp(r$coe[1]+r$coe[2]*xx)) # 按照公式计算预测值
> lines(y.pre~xx,col='red',lwd=3) # 绘出预测线, 与glm模型的预测完全一样
```

33.5.2 例2

考虑Silvey (1970) 提供的一个人造的小例子。参考文献 [25]

在Kalythos 的Aegean岛上，男性居民常常患有一种先天的眼科疾病，并且随着年龄的增长而变的愈明显。现在搜集了各种年龄段岛上男性居民的样本，同时记录了盲眼的数目。数据展示如下：

```
Age:         20 35 45 55 70
No.: tested: 50 50 50 50 50
No.: blind:  6  17 26 37 44
```

我们想知道的是这些数据是否吻合logistic 和probit 模型，并且分别估计各个模型的LD50，也就是一个男性居民盲眼的概率为50%时候的年龄。如果y 和n 分别是年龄为x时的盲眼数目和检测样本数目，两种模型的形式都为

$$y \sim B(n, F(\beta_0 + \beta_1 x))$$

其中在probit 模型中，$F(z) = \Phi(z)$ 是标准的正态分布函数，而在logit 模型(默认)中，$F(z) = e^z/(1 + e^z)$。这两种模型中，

$$LD50 = `\beta_0/\beta_1$$

，即分布函数的参数为0时所在的点。

第一步是把数据转换成数据框,

```
kalythos <- data.frame(x = c(20,35,45,55,70),
      y = c(6,17,26,37,44), n = rep(50,5))
```

在glm() 拟合二项式模型时，响应变量有三种可能性：

- 如果响应变量是向量，则假定操作二元（binary）数据，因此要求是0/1向量。
- 如果响应变量是双列矩阵，则假定第一列为试验成功的次数, 第二列为试验失败的次数。
- 如果响应变量是因子，则第一水平作为失败(0) 考虑而其他的作为‘成功’(1) 考虑。

这里，我们采用的是第二种惯例。我们在数据框中增加了一个矩阵：

```
> kalythos$Ymat <- cbind(kalythos$y, kalythos$n - kalythos$y)
> kalythos$Ymat
     [,1] [,2]
[1,]    6   44
[2,]   17   33
[3,]   26   24
[4,]   37   13
[5,]   44    6
```

为了拟合这些模型，我们采用

```
> fmp <- glm(Ymat ~ x, family = binomial(link=probit), data = kalythos)
> fml <- glm(Ymat ~ x, family = binomial, data = kalythos) # 默认为logit关联函数
```

既然logit的关联函数是默认的，因此我们可以在第二条命令中省略该参数。为了查看拟合结果，我们使用

```
> summary(fmp)

Call:
```

```
glm(formula = Ymat ~ x, family = binomial(link = probit), data = kalythos)

Deviance Residuals:
      1       2       3       4       5
-0.15582  0.02545 -0.08009  0.51246 -0.40097

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.102270   0.276287  -7.609 2.76e-14 ***
x            0.048147   0.005885   8.181 2.82e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 82.14455  on 4  degrees of freedom
Residual deviance:  0.45473  on 3  degrees of freedom
AIC: 24.270

Number of Fisher Scoring iterations: 4

>  summary(fml)

Call:
glm(formula = Ymat ~ x, family = binomial, data = kalythos)

Deviance Residuals:
     1       2       3       4       5
-0.1797  0.1157 -0.1182  0.3791 -0.3372

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.53778    0.50232  -7.043 1.88e-12 ***
x            0.08114    0.01082   7.498 6.47e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 82.14455  on 4  degrees of freedom
Residual deviance:  0.31707  on 3  degrees of freedom
```

```
AIC: 24.132

Number of Fisher Scoring iterations: 4
```

两种模型都拟合的很好。为了计算LD50，我们可以利用一个简单的函数：

```
>  ld50 <- function(b) -b[1]/b[2]
>  ldp <- ld50(coef(fmp)); ldl <- ld50(coef(fml)); c(ldp, ldl)
(Intercept) (Intercept)
  43.66335    43.60119
```

从这些数据中得到的年龄分别是43.663年和43.601年。

33.6 TODO: Poisson模型

Poisson族默认的关联函数是log。在实际操作中，这一族常常用于拟合计数资料的Poisson对数线性模型。这些计数资料的实际分布往往符合二项式分布。这是一个非常重要而又庞大的话题，我们不想在这里深入展开。它甚至是非-高斯广义模型内容的主要部分。(参考文献 [25])

有时候，实践中产生的Poisson数据在对数或者平方根转化后可当作正态数据处理。作为后者的另一种选择是，一个Poisson广义线性模型可以通过下面的方式拟合[1]：

```
> fmod <- glm(y ~ A + B + x, family = poisson(link=sqrt),
            data = worm.counts)
```

[1]找不到数据 worm.counts

33.7 TODO: 拟似然模型

对于所有的族，响应变量的方差依赖于均值并且拥有作为乘数（multiplier）的尺度参数。方差对均值的依赖方式是响应分布的一个特性；例如对于poisson分布$Var[y] = \mu$。

对于拟似然估计和推断，我们不是设定精确的响应分布而是设定关联函数和方差函数的形式，因为关联函数和方差函数都依赖于均值。既然拟似然估计和 gaussian 分布使用的技术非常相似，因此这一族顺带提供了一种用非标准关联函数或者方差函数拟合gaussian 模型的方法。(参考文献 [25])

例如，考虑非线性回归的拟合

$$y = \frac{\theta_1 z_1}{z_2 - \theta_2} + e$$

同样还可以写成

$$y = \frac{1}{\beta_1 x_1 + \beta_2 x_2} + e$$

其中$x_1 = z_2/z_1 . x_2 = -1/x_1 . \beta_1 = 1/\theta_1, \beta_2 = \theta_2/\theta_1$. 假如有适合的数据框，我们可以如下进行非线性拟合[2]

```
nlfit <- glm(y ~ x1 + x2 - 1,
             family = quasi(link=inverse, variance=constant),
             data = biochem)
```

如果需要的话，读者可以从其他手册或者帮助文档中得到更多的信息。

[2]找不到数据 biochem

33.8 其它资料找到的东东

33.8.1 数据

```
> m1<-rnorm(10) # 理论均值为0, 但是样本量小, 实际可能相差很大.
> a<-rnorm(200,m1,sd=0.3) # a的均值为m1的均值, 标准差为m1的标准差
> m2<-rnorm(10,1) # 理论均值为1, 但是样本量小, 实际可能相差很大.
> b<-rnorm(200,m2,sd=0.3) # a的均值为m2的均值, 标准差为m2的标准差
> x1<-c(a[2*1:100],b[2*1:100]) # x1为a,b的偶数位置的值
> x2<-c(a[2*1:100-1],b[2*1:100-1]) # x2为a,b的奇数位置的值
# y为被预测的值(分类), a偶数,奇数位置确定的分类是0, 用红色表示.
# b偶数,奇数位置确定的分类是1, 蓝色表示.
> y<-c(rep(0,100),rep(1,100))
> plot(x1,x2,col=c('red','blue')[1+y])
```

33.8.2 回归分析

使用线性模型对y分类的预测, 预测效果并不好. 增加交互效应后, 效果没有改善. 因为数据本身不是线性可分的.

```
> plot(x1,x2,col=c('red','blue')[1+y])
> r <- lm(y~x1+x2)
> abline(r)
> r2 <- lm(y~x1+x2+I(x1*x2)) # 增加交互效应
> abline(r2,col='red')
```

TODO: 有一个使用contour的例子, 但是没有看懂

33.8.3 Poisson回归

Poisson回归处理应变量为自然数的情况. 流行病学中, Poisson回归处理前瞻性研究(cohort study)数据, 分析感兴趣的时间-人的事件发生率. 常用于单位时间，单位面积，单位空间内某事件发生数(count)的影响因素分析。

Poisson回归的两个假设: 事件发生率是独立于时间的(与时间无关), 理论均值等于方差(这个是Poisson分布的数学性质).

Chapter 34

Generalized additive models

由 Trevor Hastie and Rob Tibshirani 1986 初始发展的模型, 是对广义线性模型的发展. 参考文献 Trevor Hastie and Robert Tibshirani Generalized additive models, Statistical Science, 1986, Vol. 1, No. 3, 297-318 **??**

http://en.wikipedia.org/wiki/Generalized_additive_model

Hastie, T. J. and Tibshirani, R. J. (1990).
Generalized Additive Models. Chapman & Hall/CRC. ISBN 9780412343902.

Wood, S. N. (2006).
Generalized Additive Models: An Introduction with R.
Chapman & Hall/CRC. ISBN 9781584884743.

线性模型

$$\eta = \sum \beta_j X_j$$

替换为

$$\eta = \sum s_j(X_j)$$

其中s_j为未知的平滑函数, 由散点图平滑器(scatterplot smoother)来估计. 使用迭代过程实现, 称为局部打分算法(local scoring algorithm).

这个技术可以用在任意的基于似然的回归模型中, 常规的广义线性模型包含了很多.

我们给出使用二值响应和生存模型的例子. 这两个对发现非线性相关有用处.

而且其自动实现的过程也对其他方法有优势. 理论上是最大化对数似然期望, 或等价的, 最小化到真实模型的Kullback-Leibler 距离.

下面的是私人数据的例子.

```
library(mgcv)
d=read.csv('Rpool.csv',sep=' ')
x=d[,1]
y=seq(56,16,len=59)
plot(y,x)
plot(y~x)
#x与y是反了~~~
#因此
t=y
y=x
x=t
plot(y~x)
视情况改变x轴
z=rev(y)
plot(z~x)
library(mgcv)
b=gam(z~s(x))
plot(b)
str(b)#b下的参数~~
b$fitted.values
plot(b$fitted.values~x,t='l')
```

Chapter 35

TODO: 岭回归(ridge regression)

在多元线性回归分析中，由于有很多变量，因此我们会选择对因变量显著性影响大的那些自变量。但是这样做之后，我们马上又遇到一个问题,例如，在某些情况下，当增加或剔出一个自变量时，回归系数变化很大，甚至改变符号。那么为什么会出现这种情况呢？很重要的原因是自变量之间存在高度的相关性，一个变量的剔除或增加严重影响了其他的自变量，进而影响了整个线性回归方程的系数。这种高度相关性被称为多重共线性，它的一个流行的解决方案是：岭回归。

多重共线性：

多重共线性就是说，自变量之间是线性相关的，即X非满秩。X非满秩，那么最小二乘估计中的X'X矩阵也不是满秩的，从而对系数估计的方差很大，进而导致估计的性质很不稳定。怎样发现多重共线性呢？一个很显然的方法就是在自变量之间作线性回归，如果发现对应于某个自变量的复相关系数很大，那么我们就有理由相信这个自变量可以由其他的变量线性表示。在做回归的时候，我们就可以剔除那些可以被其他变量线性表示的变量，但是理论上不能保证这样迭代后，共线性一定会降低到可接收的程度。许多统计学者在这方面做了大量的工作，下面介绍其中的一个方法：岭回归。

$$w = (X'X + \lambda I_n)^{-1} X'y$$

Chapter 36

非线性回归与非线性最小平方

主要内容来自 [42].

36.1 非线性回归

一般的回归为

$$y_i = x_i'\beta + \epsilon_i$$

x_i'为行向量, β为待估计参数, $\epsilon_i \sim N(0, \sigma^2)$.

非线性模型中

$$y_i = f(x_i', \beta) + \epsilon_i$$

f为非线性形式.

非线性回归的似然值为

$$L(\beta, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{n/2}} exp(-\frac{\sum_{i=1}^{n}[y_i - f(x_i', \beta)]^2}{2\sigma^2})$$

当残差平方和

$$S(\beta) = \sum_{i=1}^{n}[y_i - f(x_i', \beta)]^2$$

取最小时, 似然值取最大. 将$S(\beta)$对β取偏导数, 为0时则得到回归系数的估计.

由于方程非线性, 需要数值优化求解. 就像在线性模型中, 通常由残差平方和除以观察数与参数个数之差来估计误差的方差(线性模型中除以n).

协方差的估计略(见参考文献 [42] page 1-2).

36.2 logistic人口模型及使用nls()函数求解

此logistic不是彼logit

函数 nls(): 非线性模型参数的最小平方估计 (Determine the nonlinear (weighted) least-squares estimates of the parameters of a nonlinear model)

下面是人口增长的例子, 为 logistic 模型

$$y_i = \frac{b_1}{1 + e^{b_2 + b_3 x_i}} + \epsilon_i$$

x_i为时间, y_i为此时的人口. b_1为人口的最大容纳量, b_2为$x = 0$时的人口, b_3为人口增长速率.

下面是美国 1790年到 1990 年每隔 10 年的人口数据.

```
> library(car)
> data(US.pop)
> attach(US.pop) # 将 year population 两个变量纳入名字搜索空间.
> year
 [1] 1790 1800 1810 1820 1830 1840 1850 1860 1870 1880 1890 1900 1910 1920 1930
[16] 1940 1950 1960 1970 1980 1990
> population
 [1]   3.929   5.308   7.240   9.638  12.866  17.069  23.192  31.443  39.818
[10]  50.156  62.948  75.995  91.972 105.711 122.775 131.669 150.697 179.323
```

```
[19] 203.302 226.542 248.710

> plot(year, population)
```

看到1990年人口为250单位(million), 还没有看出到达上限, 我们不妨设$b_1 = 350$, 有

$$3.929 = \frac{350}{1 + e^{b_2 + b_3 0}}$$

可以解得

$$b_2 = log_e(350/3.929 - 1) = 4.5$$

然后将$x = 1(1800), b_2 = 4.5$带入, 可以得到$b_3 = -0.3$. 我们就获得了迭代的初始值. 下面就使用 nls 函数来拟合. ($trace = T$ 可以显示迭代的残差平方和(最前面的数字).)

```
> time <- 0:20
> pop.mod <- nls(population ~ beta1/(1 + exp(beta2 + beta3*time)),
+     start=list(beta1 = 350, beta2 = 4.5, beta3 = -0.3),
+     trace=T)
13007.48 :  350.0   4.5  -0.3
609.5727 :  351.8074862   3.8405002  -0.2270578
365.4396 :  383.7045367   3.9911148  -0.2276690
356.4056 :  389.1350260   3.9897242  -0.2265769
356.4001 :  389.1462874   3.9903758  -0.2266276
356.4001 :  389.1665272   3.9903412  -0.2266193
356.4001 :  389.1655106   3.9903457  -0.2266199

> pop.mod
Nonlinear regression model
  model:  population ~ beta1/(1 + exp(beta2 + beta3 * time))
   data:  parent.frame()
   beta1    beta2    beta3
389.1655   3.9903  -0.2266
 residual sum-of-squares: 356.4
```

```
Number of iterations to convergence: 6
Achieved convergence tolerance: 1.455e-06

> summary(pop.mod)

Formula: population ~ beta1/(1 + exp(beta2 + beta3 * time))

Parameters:
      Estimate Std. Error t value Pr(>|t|)
beta1 389.16551   30.81196   12.63 2.20e-10 ***
beta2   3.99035    0.07032   56.74  < 2e-16 ***
beta3  -0.22662    0.01086  -20.87 4.60e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.45 on 18 degrees of freedom

Number of iterations to convergence: 6
Achieved convergence tolerance: 1.455e-06
```

前面的模型的形式为了方便计算而写, 实际上的模型应该将指数部分变化一下形式, 变为

$$y_i = \frac{b_1}{1 + e^{-b_3(x_i - b_2/b_3}} + \epsilon_i$$

那么, $-b_3 = 0.22662$为实际的增长率. b_1, b_2 意义不变, 分别为人口的最大容纳量和和时间为 0 时的人口.

36.3 非线性最小二乘法和最大似然法模型

此节摘自《R导论》[25](page 78).

特定形式的非线性模型可以通过广义线性模型(glm())拟合。但是许多时候，我们必须把非线性拟合的问题作为一个非线性优化的问题解决。R的非线性优化

程序是optim(), nlm() 和nlminb()（自R2.2.0开始）。二者分别替换SPLUS 的ms()和nlminb()但功能更强。我们通过搜寻参数值使得缺乏度（lack-of- fit）指标最低，如nlm() 就是通过循环调试各种参数值得到最优值。和线性回归不同，程序不一定会收敛到一个稳定值。nlm()需要设定参数搜索的初始值，而参数估计是否收敛在很大程度上依赖于初始值设置的质量(可以用一些经验的方法判断初始的参数设定。)

36.3.1 nlm()函数的用法

```
nlm(f, p, ..., hessian = FALSE, typsize = rep(1, length(p)),
   fscale = 1, print.level = 0, ndigit = 12, gradtol = 1e-6,
   stepmax = max(1000 * sqrt(sum((p/typsize)^2)), 1000),
   steptol = 1e-6, iterlim = 100, check.analyticals = TRUE)
```

此函数使用 Newton 型算法求极小值, 返回极小值, 极小点的估计值, 极小点处的梯度, hessen 矩阵, 迭代次数等.

f: 求极小值的目标函数, 若其属性(attr)包含 'gradient' or both 'gradient' and 'hessian', 则在计算过程中会使用它们. 否则使用数值的方法来计算偏导数.

p: 参数初始值

hessian = True 会返回最小化时的 hessian 矩阵

36.3.2 最小二乘法

拟合非线性模型的一种办法就是使误差平方和（SSE）或残差平方和最小。如果观测到的误差极似正态分布，这种方法是非常有效的。

下面是例子来自Bates, Watts (1988)，51页。具体数据是：

```
x <- c(0.02, 0.02, 0.06, 0.06, 0.11, 0.11, 0.22, 0.22, 0.56,
```

```
   0.56, 1.10, 1.10)
y <- c(76, 47, 97, 107, 123, 139, 159, 152, 191, 201, 207, 200)
```

被拟合的模型是(实际上编写一个计算误差平方和的函数, nlm()对此函数进行最小化):

```
fn <- function(p) sum((y - (p[1] * x)/(p[2] + x))^2)
```

为了进行拟合，我们需要估计参数初始值。一种寻找合理初始值的办法把数据图形化，然后估计一些参数值，并且利用这些值初步添加模型曲线。

```
plot(x, y)
xfit <- seq(.02, 1.1, .05)
yfit <- 200 * xfit/(0.1 + xfit)
lines(spline(xfit, yfit))
```

当然，我们可以做的更好，但是初始值200 和0.1 应该足够了。现在做拟合：

```
> out <- nlm(fn, p = c(200, 0.1), hessian = TRUE)
> out
$minimum
[1] 1195.449

$estimate
[1] 212.68384222   0.06412146

$gradient
[1] -0.0001534973  0.0934205639

$hessian
          [,1]       [,2]
[1,]   11.94725   -7661.319
```

```
[2,] -7661.31875 8039421.153

$code
[1] 3

$iterations
[1] 26
```

拟合后，out$minimum 是误差的平方和（SSE），out$estimate 是参数的最小二乘估计值。为了得到参数估计过程中近似的标准误(SE)，我们可以计算：

```
> sqrt(diag(2*out$minimum/(length(y) - 2) * solve(out$hessian)))
[1] 7.173465192 0.008744815
```

上述命令中的2表示参数的个数。一个95% 的信度区间可以通过± 1.96 SE 计算得到。我们可以把这个最小二乘拟合曲线画在一个新的图上：

```
> plot(x, y)
> xfit <- seq(.02, 1.1, .05)
> yfit <- 212.68384222 * xfit/(0.06412146 + xfit)
> lines(spline(xfit, yfit))
```

标准包stats 提供了许多用最小二乘法拟合非线性模型的扩充工具。我们刚刚拟合过的模型是Michaelis-Menten 模型，因此可以利用下面的命令得到类似的结论。

```
> df <- data.frame(x=x, y=y)
> fit <- nls(y ~ SSmicmen(x, Vm, K), df)
> fit
Nonlinear regression model
  model: y ~ SSmicmen(x, Vm, K)
   data: df
```

```
      Vm        K
212.68371   0.06412
 residual sum-of-squares: 1195

Number of iterations to convergence: 0
Achieved convergence tolerance: 1.924e-06
> summary(fit)

Formula: y ~ SSmicmen(x, Vm, K)

Parameters:
   Estimate Std. Error t value Pr(>|t|)
Vm 2.127e+02  6.947e+00  30.615 3.24e-11 ***
K  6.412e-02  8.281e-03   7.743 1.57e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.93 on 10 degrees of freedom

Number of iterations to convergence: 0
Achieved convergence tolerance: 1.924e-06
```

36.3.3 最大似然法

最大似然法（Maximum likelihood）也是一种非线性拟合方法。它甚至可以用在误差非正态的数据中。这种方法估计的参数将会使得对数似然值最大或者负的对数似然值最小。下面的例子来自Dobson (1990), pp. : 108–111。这个例子对剂量－响应数据拟合logistic模型（当然也可以用glm() 拟合）。数据是：

```
x <- c(1.6907, 1.7242, 1.7552, 1.7842, 1.8113,
      1.8369, 1.8610, 1.8839)
y <- c( 6, 13, 18, 28, 52, 53, 61, 60)
n <- c(59, 60, 62, 56, 63, 59, 62, 60)
```

要使负对数似然值最小，则：

```
> fn <- function(p)
   sum( - (y*(p[1]+p[2]*x) - n*log(1+exp(p[1]+p[2]*x))
          + log(choose(n, y)) ))
```

我们选择一个适当的初始值，开始拟合：

```
> out <- nlm(fn, p = c(-50,20), hessian = TRUE)
> out
$minimum
[1] 18.71513

$estimate
[1] -60.71727  34.27021

$gradient
[1] 1.345785e-08 2.280689e-08

$hessian
         [,1]     [,2]
[1,]  58.48407 103.9787
[2,] 103.97873 184.9662

$code
[1] 1

$iterations
[1] 21
```

拟合后，out$minimum 就是负对数似然值，out$estimate 就是最大似然拟合的参数值。为了得到拟合过程近似的标准误，我们可以：

```
> sqrt(diag(solve(out$hessian)))
[1] 5.553083 3.122531
```

参数估计的95% 信度期间可由估计值 ± 1.96 SE 计算得到。

Chapter 37

偏最小二乘回归方法及其应用(理论)

(王惠文著, 国防工业出版社1999年版, http://www.pkumpa.cn/yb/sub/partial.html)

37.1 介绍

偏最小二乘回归≈多元线性回归分析+典型相关分析+主成分分析

与传统多元线性回归模型相比，偏最小二乘回归的特点是

1. 能够在自变量存在严重多重相关性的条件下进行回归建模
2. 允许在样本点个数少于变量个数的条件下进行回归建模
3. 偏最小二乘回归在最终模型中将包含原有的所有自变量
4. 偏最小二乘回归模型更易于辨识系统信息与噪声(甚至一些非随机性的噪声)
5. 在偏最小二乘回归模型中，每一个自变量的回归系数将更容易解释

在计算方差和协方差时，求和号前面的系数有两种取法：当样本点集合是随机抽取得到时，应该取$1/(n-1)$；如果不是随机抽取的，这个系数可取$1/n$。

37.2 多重相关性的诊断

37.2.1 经验式诊断方法

1. 在自变量的简单相关系数矩阵中，有某些自变量的相关系数值较大
2. 回归系数的代数符号与专业知识或一般经验相反；或者，它同该自变量与y的简单相关系数符号相反。
3. 对重要自变量的回归系数进行t检验，其结果不显著。特别典型的是，当F检验能在高精度下通过，测定系数R^2的值亦很大，但自变量的t检验却全都不显著，这时，多重相关性的可能性将很大。
4. 如果增加(或删除)一个变量，或者增加(或删除)一个观测值，回归系数的估计值发生了很大的变化。
5. 重要自变量的回归系数置信区间明显过大。
6. 在自变量中，某一个自变量是另一部分自变量的完全或近似完全的线性组合。
7. 对于一般的观测数据，如果样本点的个数过少，样本数据中的多重相关性是经常存在的。

但是，采用经验式方法诊断自变量系统中是否确实存在多重相关性，并不十分可靠，另一种较正规的方法是利用统计检验(回归分析)，检查每一个自变量相对其它自变量是否存在线性关系。

37.2.2 方差膨胀因子

最常用的多重相关性的正规诊断方法是使用方差膨胀因子。自变量x_j的方差膨胀因子记为$(VIF)_j$，它的计算方法为

$$(VIF)_j = (1 - R_j^2)^{-1}$$

式中，R_j^2是以x_j为因变量时对其它自变量回归的复测定系数。

所有x_j变量中最大的$(VIF)_j$通常被用来作为测量多重相关性的指标。一般认为，如果最大的$(VIF)_j$超过10，常常表示多重相关性将严重影响最小二乘的估计值。

$(VIF)_j$被称为方差膨胀因子的原因，是由于它还可以度量回归系数的估计方差与自变量线性无关时相比，增加了多少。

不妨假设$x_1, x_2, \ldots, x_p$均是标准化变量。采用最小二乘法得到回归系数向量B，它的精度是用它的方差来测量的。B的协方差矩阵为

$$Cov(B) = \sigma^2(X'X) - 1$$

式中，σ^2是误差项方差。所以，对于回归系数b_j，有

$$Var(b_j) = \sigma^2 c_{jj}$$

c_{jj}是$(X'X)^{-1}$矩阵中第j个对角元素。可以证明，

$$c_{jj} = (VIF)_j$$

37.3 岭回归分析

37.3.1 岭回归估计量

岭回归分析是一种修正的最小二乘估计法，当自变量系统

中存在多重相关性时，它可以提供一个比最小二乘法更为稳定的估计，并且回归系数的标准差也比最小二乘估计的要小。

根据高斯——马尔科夫定理，多重相关性并不影响最小二乘估计量的无偏性和最小方差性。但是，虽然最小二乘估计量在所有线性无偏估计量中是方差最小的，但是这个方差却不一定小。于是可以找一个有偏估计量，这个估计量虽然有微小的偏差，但它的精度却能够大大高于无偏的估计量。

在应用岭回归分析时，它的计算大多从标准化数据出发。对于标准化变量，最小二乘的正规方程为

$$r_{XX}b = r_yX$$

式中，r_{XX}是X的相关系数矩阵，r_yX是y与所有自变量的相关系数向量。

岭回归估计量是通过在正规方程中引入有偏常数$c(c>0)$而求得的。它的正规方程为

$$(r_{XX}+cI)b^R = r_yX$$

所以，在岭回归分析中，标准化回归系数为

$$b^R = (r_{XX}+cI)^{-1}r_yX$$

37.3.2 岭回归估计量的性质

(1) 岭回归系数是一般最小二乘准则下回归系数的线性组合，即

$$b^R = (I+cr_{XX}-1)^{-1}b$$

(2) 记β是总体参数的理论值。当$\beta \neq 0$时，可以证明一定存在一个正数c_0，使得当$0<c<c_0$时，一致地有

$$E||b^R-\beta||^2 <= E||b-\beta||2$$

(3) 岭回归估计量的绝对值常比普通最小二乘估计量的绝对值小，即

$$||b^R|| < ||b||$$

岭回归估计量的质量取决于偏倚系数c的选取。c的选取不宜过大，因为

$$E(b^R) = (I + cr_{XX}^{-1})^{-1} \quad E(b) = (I + cr_{XX}^{-1})^{-1}\beta$$

关于偏倚系数c的选取尚没有正规的决策准则，目前主要以岭迹和方差膨胀因子为依据。岭迹是指$p-1$个岭回归系数估计量对不同的c值所描绘的曲线(c值一般在$[0,1]$之间)。在通过检查岭迹和方差膨胀因子来选择c值时，其判断方法是选择一个尽可能小的c值，在这个较小的c值上，岭迹中的回归系数已变得比较稳定，并且方差膨胀因子也变得足够小。

从理论上，最佳的c值是存在的，它可以使估计量的偏差和方差的组合效应达到一个最佳水准。然而，困难却在于c的最优值对不同的应用而有所不同，对其选择还只能凭经验判断。

37.3.3 其他补救方法简介

最常见的一种思路是设法去掉不太重要的相关性变量。由于变量间多重相关性的形式十分复杂，而且还缺乏十分可靠的检验方法，删除部分多重相关变量的做法常导致增大模型的解释误差，将本应保留的系统信息舍弃，使得接受一个错误结论的可能和做出错误决策的风险都不断增长。另一方面，在一些经济模型中，从经济理论上要求一些重要的解释变量必须被包括在模型中，而这些变量又存在多重相关性。这时采用剔除部分相关变量的做法就不符合实际工作的要求。

另一种补救的办法是增加样本容量。然而，在实际工作中，由于时间、经费以及客观条件的限制，增大样本容量的方法常常是不可行的。

此外，还可以采用变量转换的方式，来削弱多重相关性的严重性。一阶差分回归模型有可能减少多重相关性的严重性。然而，一阶差分变换又带来了一些其它问题。差分后的误差项可能不满足总体模型中关于误差项不是序列相关的假定。事实上，在大部分情形下，在原来的误差项是不自相关的条件下，一阶差分所得到的误差项将会是序列相关的。而且，由于差分方法损失了一个观察值，这在小样本的情况下是极不可取的。另外，一阶差分方法在截面样本中是不宜利用的。

- 主成分分析主成分分析的计算结果必然受到重叠信息的影响。因此，当人为地采用一些无益的相关变量时，无论从方向上还是从数量上，都会扭曲客观结论。在主成分分析之前，对变量系统的确定必须是慎之又慎的。
- 特异点的发现 第i个样本点(样本量为n)对第h主成分的贡献率是

  $$CTR(i) = F_h^2(i)/(n\lambda_h)$$

  (若远超过1/n，为特异点)
- 典型相关分析

  从某种意义上说，多元回归分析、判别分析或对应分析等许多重要的数据分析方法，都可以归结为典型相关分析的一种特例，同时它还是偏最小二乘回归分析的理论基石。

  典型相关分析，是从变量组X中提取一个典型成分$F = Xa$，再从变量组Y中提取一个成分$G = Yb$，在提取过程中，要求F与G的相关程度达到最大。

  在典型相关分析中，采用下述原则寻优，即

  $$\begin{aligned} max < F, G > &= aX'Yba'X'Xa = 1 \\ b'Y'Yb &= 1 \end{aligned}$$

  其结果为，a是对应于矩阵$V_{11}^{-1}V_{12}V_{22}^{-1}V_{21}$最大特征值的特征向量，而b是对应于矩阵$V_{22}^{-1}V_{21}V_{11}^{-1}V_{12}$最大特征值的特征向量，这两个最大特征值相同。其中，

  $$V_{11} = X'X \text{fi} V_{12} = X'Y \text{fi} V_{22} = Y'Y$$

F与G之间存在着明显的换算关系。

有时只有一个典型成分还不够，还可以考虑第二个典型成分。

37.4 多因变量的偏最小二乘回归模型

37.4.1 工作目标

偏最小二乘回归分析的建模方法

设有q个因变量和p个自变量。为了研究因变量与自变量的统计关系，观测了n个样本点，由此构成了自变量与因变量的数据表X和Y。偏最小二乘回归分别在X与Y中提取出t和u，要求：

(1)t和u应尽可能大地携带它们各自数据表中的变异信息；

(2)t和u的相关程度能够达到最大。

在第一个成分被提取后，偏最小二乘回归分别实施X对t的回归以及Y对t的回归。如果回归方程已经达到满意的精度，则算法终止；否则，将利用X被t解释后的残余信息以及Y被t解释后的残余信息进行第二轮的成分提取。如此往复，直到能达到一个较满意的精度为止。若最终对X共提取了多个成分，偏最小二乘回归将通过施行y_k对X的这些成分的回归，然后再表达成y_k关于原自变量的回归方程。

37.4.2 计算方法-第一步

首先将数据做标准化处理。X经标准化处理后的数据矩阵记为$E_0 = (E - 01, \ldots, E_{0p})_{np}$，Y的相应矩阵记为$F0 = (F_{01}, \ldots, F_{0q})_{nq}$。

第一步记t_1是E_0的第一个成分，$t_1 = E_0 w_1$，w_1是E_0的第一个轴，它是一个单位向量，即$||w1|| = 1$。

记u_1是F_0的第一个成分，$u_1 = F_0 c_1$，c_1是F_0的第一个轴，并且$||c1|| = 1$。

于是，要求解下列优化问题，即

$$max_{(w_1,,c_1)} < E_0 w_1, F_0 c_1 >$$

使得

$$w_1' w_1 = 1$$
$$c_1' c_1 = 1$$

记$\theta_1 = w_1' E_0' F_0 c_1$，即正是优化问题的目标函数值。

采用拉格朗日算法，可得

$$E_0' F_0 F_0' E_0 w_1 = \theta_1^2 w_1$$
$$F_0' E_0 E_0' F_0 c_1 = \theta_1^2 c_1$$

所以，w_1是对应于$E_0' F_0 F_0' E_0$矩阵最大特征值的单位特征向量，而c_1是对应于$F_0' E_0 E_0' F_0$矩阵最大特征值θ_1^2的单位特征向量。

求得轴w_1和c_1后，即可得到成分

$$t_1 = E_0 w_1 u_1 = F_0 c_1$$

然后，分别求E_0和F_0对t_1的回归方程

$$E_0 = t_1 p_1' + E_1$$
$$F_0 = t_1 r_1' + F_1$$

式中，回归系数向量是

$$p_1 = E_0' t_1 / ||t_1||^2$$
$$r_1 = F_0' t_1 / ||t_1||^2$$

而E_1和F_1分别是两个方程的残差矩阵。

37.4.3 计算方法-第二步

第二步用残差矩阵E_1和F_1取代E_0和F_0，然后，求第二个轴w_2和c_2以及第二个成分$t_2$fiu_2，有

$$t_2 = E_1 w_2$$
$$u_2 = F_1 c_2 \theta_2 =< t_2, u_2 >= w_2' E_1' F_1 c_2$$

w_2是对应于$E_1' F_1 F_1' E_1$矩阵最大特征值的单位特征向量，而c_2是对应于$F_1' E_1 E_1' F_1$矩阵最大特征值θ_2^2的单位特征向量。计算回归系数

$$p_2 = E_1' t_2 / ||t_2||^2$$
$$r_2 = F_1' t_2 / ||t_2||^2$$

因此，有回归方程

$$E_1 = t_2 p_2' + E_2$$
$$F_1 = t_2 r_2' + F_2$$

如此计算下去，如果X的秩是A，则会有

$$E_0 = t_1 p_1' + \cdots + t_A p_A' F_0 = t_1 r_1' + \cdots + t_A r_A' + F_A$$

由于$t_1$fi...fit_A均可以表示成E_{01}fi...fiE_{0p}的线性组合，因此，上面的式子还可以还原成$y_k^* = F_{0k}$关于$x_j^* = E_{0j}$的回归方程形式，即

$$y_k^* = \alpha_{k1} x_1^* + \ldots + \alpha_{kp} x_p^* + F_{Ak} \text{fi} k = 1, 2, \ldots, q$$

F_{Ak}是残差矩阵F_A的第k列。

37.4.4 交叉有效性

如果多一个成分而少一个样本的预测误差平方和(所有因变量和预测样本相加)除以少一个成分的误差平方和(所有的因变量和样本相加)小于0.952，则多一个成分是值得的。

37.5 一种更简洁的计算方法

用下述原则提取自变量中的成分t_1，是与原则式(7-1)的结果完全等价的，即

$$max_{||w_i||=1}\sum_{k=1}^{q}Cov^2(F_{0k},E_0w_1)$$

(1) 求矩阵$E_0'F_0F_0'E_0$最大特征值所对应的单位特征向量w_1，求成分t_1，得

$$t_1=E_0w_1$$
$$E_1=E_0-t_1p_1'$$

式中，$p_1=E_0't_1/||t_1||^2$

(2) 求矩阵$E_1'F_0F_0'E_1$最大特征值所对应的单位特征向量w_2，求成分t_2，得

$$t_2=E_1w_2$$
$$E_2=E_1-t_2p_2'$$

式中，$p_2=E_1't_2/||t_2||^2$

(m) 至第m步，求成分$t_m=E_{m-1}w_m$，w_m是矩阵$E_{m-1}'F_0F_0'E_{m-1}$最大特征值所对应的单位特征向量.

如果根据交叉有效性，确定共抽取m个成分$t_1$fi...fit_m可以得到一个满意的观测模型，则求F_0在$t_1$fi...fit_m上的普通最小二乘回归方程为

$$F_0=t_1r_1'+\ldots+t_mr_m'+F_m$$

37.6 偏最小二乘回归的辅助分析技术

37.6.1 精度分析

定义自变量成分t_h的各种解释能力如下

(1)t_h对某自变量xj的解释能力

$$Rd(x_j;t_h)=r^2(x_j,t_h)$$

(2)t_h对X的解释能力

$$Rd(X;th)=[r2(x1,th)+\ldots+r2(xp,th)]/p$$

(3)$t_1,\cdots,t_m$对X的累计解释能力

$$Rd(X;t_1,\cdots,t_m)=Rd(X;t_1)+\ldots+Rd(X;t_m)$$

(4)$t_1,\cdots,t_m$对某自变量x_j的累计解释能力

$$Rd(x_j;t_1,\cdots,t_m)=Rd(x_j;t_1)+\ldots+Rd(x_j;t_m)$$

(5)t_h对某因变量y_k的解释能力

$$Rd(y_k;t_h)=r^2(y_k,t_h)$$

(6)t_h对Y的解释能力

$$Rd(Y;t_h)=[r^2(y_1,t_h)+\ldots+r^2(y_q,t_h)]/q$$

(7)$t_1,\cdots,t_m$对Y的累计解释能力

$$Rd(Y;t_1,\cdots,t_m)=Rd(Y;t_1)+\ldots+Rd(Y;t_m)$$

(8)$t_1,\cdots,t_m$对某因变量yk的累计解释能力

$$Rd(y_k;t_1,\cdots,t_m)=Rd(y_k;t_1)+\ldots+Rd(y_k;t_m)$$

37.6.2 自变量x_j在解释因变量集合Y的作用

x_j在解释Y时作用的重要性，可以用变量投影重要性指标VIP_j来测度

$$VIP_j^2=p[Rd(Y;t_1)w_{1j}^2+\ldots+Rd(Y;tm)w_{mj}^2]/[Rd(Y;t_1)+\ldots+Rd(Y;t_m)]$$

式中，w_{hj}是轴w_h的第j个分量。注意 $VIP_1^2+\ldots+VIP_p^2=p$

37.6.3 特异点的发现

定义第i个样本点对第h成分th的贡献率T_{hi}^2，用它来发现样本点集合中的特异点，即

$$T_{hi}^2 = t_{hi}^2/((n-1)s_h^2)$$

式中，s_h^2是成分t_h的方差。

由此，还可以测算样本点i对成分$t_1,\cdots,t_m$的累计贡献率

$$T_i^2 = T_{1i}^2 + \ldots + T_{mi}^2$$

当

$$T_i^2 >= m(n^2-1)F_{0.05}(m,n-m)/(n^2(n-m))$$

时，可以认为在95%的检验水平上，样本点i对成分$t_1,\cdots,t_m$的贡献过大。

37.7 单因变量的偏最小二乘回归模型

37.7.1 简化算法

第一步已知数据E_0fiF_0，由于$u_1 = F_0$，可得

$$\begin{aligned} w_1 &= E_0'F_0/||E_0'F_0|| \\ t_1 &= E_0w_1 \\ p_1 &= E_0't_1/||t_1||^2 \\ E_1 &= E_0 - t_1p_1' \end{aligned}$$

检验交叉有效性。若有效，继续计算；否则只提取一个成分t_1。

第h步$(h=2,\ldots,m)$, 已知数据E_{h-1}fiF_0, 有

$$\begin{aligned} w_h &= E'_{h-1}F_0/||E'_{h-1}F_0|| \\ t_h &= E_{h-1}w_h \\ p_h &= E'_{h-1}t_h/||t_h||^2 \\ E_h &= E_{h-1}-t_hp'_h \end{aligned}$$

检验交叉有效性。若有效, 继续计算$h+1$步; 否则停止求成分的计算。

这时, 得到m个成分$t_1$fi$\cdots$fit_m, 实施F_0在$t_1$fi$\cdots$fit_m上的回归, 得

$$F\text{0̸} = r_1t_1+\cdots+r_mt_m$$

由于$t_1$fi$\cdots$fit_m均是E_0的线性组合, 即

$$t_h = E_{h-1}w_h = E_0w_h*$$

所以F_0̂可写成E_0的线性组合形式, 即

$$F_0\hat{} = r_1E_0w_1*+\cdots+r_mE_0w_m* = E_0[r_1w_1*+\cdots+r_mw_m*]$$

最后, 也可以变换成y对$x_1$fi...fix_p的回归方程

$$\hat{y} = \alpha_0+\alpha_1x_1+\cdots+\alpha_px_p$$

Chapter 38

主成分分析(PCA)

参考 [21] 第九章, [13] 十四章第四节. [11] 第二章第四节主成分分析

主成分分析(principal component analysis, PCA)是 Pearson(1901)提出的. 后来被 Hotelling(1933) 发展.

PCA 是一种降维技术, 把多个变量化为少数几个主成分, 能够反映原始变量大部分信息, 通常表示为原变量的线性组合.

设 X 有 p 个变量, 为 $n * p$ 阶矩阵, 即 n 个样本的 p 维向量. 首先对 X 的 p 个变量寻找正规化线性组合, 使它的方差达到最大(谁的方差?), 这个新的变量称为第一主成分. 抽取第一主成分后, 第二主成分的抽取方法与第一主成分一样, 使抽取第一主成分后的留下的变量的剩余方差达到最大. 依次类推, 直到各主成分累积方差达到总方差的一定比例(一般为 80%)为止.

38.1 协方差矩阵求主成分

38.1.1 记号

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \dots \\ x_{n1} & x_{n2} & \vdots & x_{np} \end{bmatrix} = \begin{bmatrix} X_{(1)}^T \\ X_{(2)}^T \\ \vdots \\ X_{(k)}^T \\ \vdots \\ X_{(n)}^T \end{bmatrix} = [X_1, X_2, \cdots, X_i, \cdots, X_p]$$

其中 X_i 为第 i 列, $i = 1, 2, \cdots, p$

$$X_i = \begin{bmatrix} x_{1i} \\ x_{2i} \\ \vdots \\ x_{ni} \end{bmatrix}$$

其中 $X_{(k)}$ 为第 k 行(第 k 个样本), $k = 1, 2, \cdots, n$

$$X_{(k)} = \begin{bmatrix} x_{k1} \\ x_{k2} \\ \vdots \\ x_{kp} \end{bmatrix}$$

记 $\sum = Var(X)$ 为 X 的协方差矩阵. $\mu = E(X) = (\bar{X_1}, \cdots, \bar{X_p})$ 为 X 的均值向量.

一般, 对于协方差矩阵 $\sum$ 存在正交矩阵 Q, 将它化为对角矩阵, 即

$$Q^T \sum Q = \Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_p \end{bmatrix}$$

且 $\lambda_1 >= \lambda_2 >= \cdots >= \lambda_p$.

则$\lambda_1, \lambda_2, \cdots, \lambda_p$ 就是特征根, 矩阵 Q 的第 i 列就是对应特征根的特征向量.

为方便记 a_i 为 Q 的列向量

$$Q = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1p} \\ a_{21} & a_{22} & \dots & a_{2p} \\ \vdots & \vdots & \ddots & \dots \\ a_{p1} & a_{p2} & \vdots & a_{pp} \end{bmatrix} = [a_1, a_2, \cdots, a_i, \cdots, a_p]$$

38.1.2 求主成分

下面分解 X 的方差. 记

$$\begin{aligned} Z &= \begin{bmatrix} Z_{(1)}^T \\ Z_{(2)}^T \\ \vdots \\ Z_{(k)}^T \\ \vdots \\ Z_{(n)}^T \end{bmatrix} \\ &= XQ = \begin{bmatrix} X_{(1)}^T Q \\ X_{(2)}^T Q \\ \vdots \\ X_{(k)}^T Q \\ \vdots \\ X_{(n)}^T Q \end{bmatrix} \\ &= [X_1, X_2, \cdots, X_i, \cdots, X_p]Q = \\ &= X[a_1, a_2, \cdots, a_i, \cdots, a_p] = [Xa_1, Xa_2, \cdots, Xa_i, \cdots, Xa_p] \\ &= [Z_1, Z_2, \cdots, Z_i, \cdots, Z_p] \\ &= Z \end{aligned}$$

显然

$$\begin{aligned} Var(Z_i) &= Z_i^T Z_i = a_i^T X^T X a_i = a_i^T \sum a_i = \lambda_i, \quad i = 1, \cdots, p \\ Cov(Z_i, Z_j) &= a_i^T X^T X a_j = a_i^T \sum a_j = 0, \quad i, j = 1, \cdots, p \end{aligned}$$

则 Z_1 方差最大, Z_2 次之, $\cdots$.

其中 $Z_1, Z_2, \cdots, Z_i, \cdots, Z_p$ 分别称为 X 的第 1 主成分, 第 2 主成分, $\cdots$.

所有主成分方差的和为 $\lambda_1 + \lambda_2 + \cdots + \lambda_p$.

$$\begin{aligned} E(Z) &= E(XQ) = Q^T \mu \\ Var(Z) &= \Lambda \end{aligned}$$

称 $\lambda_i / \sum_{i=1}^{p} \lambda_i$ 为主成分 Z_i 的贡献率. 贡献率表示的是主成分解释原始变量 X 的能力, 主成分的贡献率越大, 解释原始变量的能力越强. 这样忽略贡献率小的主成分, 通常取前 m 个主成分(对主成分的累积贡献率 80%) 的主成分即可. 此时可以使用 $Z_1, Z_2, \cdots, Z_m$ 代替 $X_1, X_2, \cdots, X_i, \cdots, X_p$. 由于 $m < p$, 我们就达到了简化原始数据的目的. 累积贡献率是前 m 个主成分从原始变量提取了多少信息的度量.

38.1.3 原始变量与主成分的相关系数

由前面知($a_{(i)}$ 为 矩阵 Q 的第 i 行)

$$\begin{aligned} X &= ZQ^T \\ X_i = Za_{(i)} &= Z_1 a_{i1} + Z_2 a_{i1} + \cdots + Z_p a_{ip} \end{aligned}$$

对上式两边取方差为

$$\sigma_{ii} = \lambda_1 a_{i1}^2 + \cdots + \lambda_p a_{ip}^2$$

由于 $a_{i1}^2 + \cdots + a_{ip}^2 = 1$, 实际上 σ_{ii} 是 $\lambda_1, \cdots, \lambda_p$ 的加权平均.

故

$$\begin{aligned} Cov(X_i, Z_j) &= Cov(Z_j a_{ij}, Z_j) = a_{ij}\lambda_j, \quad i, j = 1, \cdots, p \\ \rho(X_i, Z_j) &= \frac{Cov(X_i, Z_j)}{\sqrt{X_i}\sqrt{Z_j}} = \frac{\sqrt{\lambda_j}}{\sqrt{\sigma_{ii}}} a_{ij}, i, j = 1, \cdots, p \end{aligned}$$

前面提到累积贡献率是前 m 个主成分 $Z_1, Z_2, \cdots, Z_m$ 从原始变量 $X_1, X_2, \cdots, X_i, \cdots, X_p$ 提取了多少信息的度量. 那么前 m 个主成分 $Z_1, Z_2, \cdots, Z_m$ 包含了 X_i 的多少信息呢? 这个是使用 X_i 与 $Z_1, Z_2, \cdots, Z_m$ 的复相关系数的平方来度量的, 称为前 m 个主成分 $Z_1, Z_2, \cdots, Z_m$ 对原始变量 X_i 的贡献率, 记为 $\rho^2_{i,1\cdots m}$

$$\begin{aligned} \rho^2_{i,1\cdots m} &= \sum_{j=1}^{m} \rho^2(X_i, Z_j) = \sum_{j=1}^{m} \lambda_j a^2_{ij}/\sigma_{ii} \\ &= \sum_{j=1}^{m} \lambda_j a^2_{ij}/(\lambda_1 a^2_{i1} + \cdots + \lambda_p a^2_{ip}) \\ \rho^2_{i,1\cdots p} &= 1 \end{aligned}$$

38.1.4 载荷(loading)

由于

$$Z_j = X a_j = X_1 a_{1j} + \cdots + X_p a_{pj}$$

称 a_{ij} 为第 j 主成分在第 i 个原始变量 X_i 上的载荷(loading), 它度量了 X_i 对 Z_j 的重要程度.

实际上, 在主成分分析中, 载荷就是正交矩阵 Q. 在因子分析中, 就是载荷因子矩阵.

38.2 相关矩阵求主成分

如果原始数据 X 各变量单位不同时, 应该将其标准化后求主成分, 此时协方差矩阵就变为相关矩阵(注: 中心化后协方差矩阵不变). 其它的推导方法内容等基本类似. 得到的主成分的性质更加简单.

设标准化后的 X 为 X^*, 则

$$X_i^* = \frac{X_i - \bar{X}_i}{\sqrt{\sigma_{ii}}}$$

其协方差矩阵, 也就是 X 的相关矩阵记为 R. R 的 p 个特征值记为

$$\lambda_1^* >= \lambda_2^* >= \cdots >= \lambda_p^*$$

相应的单位特征向量记为

$$a_1^*, \cdots, a_p^*$$

p 个主成分记为

$$\begin{aligned} Z^* &= [Z_1^*, \cdots, Z_p^*] \\ Z_i^* &= X_i^* a_i^* \\ Z^* &= X^* R \end{aligned}$$

Z^* 的性质如下

1. $E(Z^*) = 0, Var(Z^*) = \Lambda^*$. $\Lambda^* = diag(\lambda_1^*, \lambda_2^*, \cdots, \lambda_p^*)$
2. $\sum_{i=1}^{p} \lambda_i^* = p$
3. X_i^*, Z_j^* 的相关系数为

$$\rho(X_i^*, Z_j^*) = \sqrt{\lambda_j^*} a_{ij}^*, \quad i, j = 1, \cdots, p$$

4. 前 m 个主成分 $Z_1^*, Z_2^*, \cdots, Z_m^*$ 对 X_i^* 的贡献率为

$$\rho_{i,1\cdots m}^2 = \sum_{j=1}^{m} \rho^2(X_i^*, Z_j^*) = \sum_{j=1}^{m} \lambda_i^* a_{ij}^{*2}$$

5.

$$\rho_{i,1\cdots p}^2 = \sum_{j=1}^{p} \rho^2(X_i^*, Z_j^*) = \sum_{j=1}^{p} \lambda_i^* a_{ij}^{*2} = 1$$

38.3 主成分特征向量的具体问题的相关解释

详细的参考任何主成分分析的书, 有详细的解释. 此处简略一说.

例如特征矩阵如下(见例子)

```
Standard deviations: # 特征值
[1] 1.5748783 0.9948694 0.5971291 0.4164494

# 贡献率
Proportion of Variance 0.6200604 0.2474413 0.0891408 0.04335752

# 累积贡献率
Cumulative Proportion  0.62 0.868 0.9566 1.0000

Rotation: # 特征矩阵(载荷)
             PC1        PC2        PC3         PC4
Murder   -0.5358995  0.4181809 -0.3412327  0.64922780
Assault  -0.5831836  0.1879856 -0.2681484 -0.74340748
UrbanPop -0.2781909 -0.8728062 -0.3780158  0.13387773
Rape     -0.5434321 -0.1673186  0.8177779  0.08902432
```

每列绝对值大的几个代表的向量就是此主成分代表的含义. 第一列绝对值大的是 Murder, Assault. 那么 Murder, Assault 就是第一主成分, 这两个变量可以解释全部方差的 62%, 第二列绝对值大的是 UrbanPop, 这个可以解释全部方差的 24.7%, 这两个加起来可以解释全部方差的 86.8%. 最后一个主成分占方差较小, 就可以忽略了.

剩下的就是使用专业知识(或常识, 经验)解释这些东西了. Murder, Assault 是代表治安方面的问题, 而 UrbanPop 代表人口方面的问题.

38.4 例子

R 中的函数 princomp() 与 prcomp() 用法意义一样, 都是做主成分分析的. 其中一种用法为

```
princomp(x, cor = FALSE, scores = TRUE, covmat = NULL,
         subset = rep(TRUE, nrow(as.matrix(x))), ...)
```

$cor = TRUE$ 是使用相关矩阵求主成分, 否则使用协方差矩阵, 或自己指定. 协方差矩阵 covmat, 适合使用其它的相关系数或距离系数的情况.

```
prcomp(x, retx = TRUE, center = TRUE, scale. = FALSE,
       tol = NULL, ...)
```

$scale = TRUE$ 即使用相关矩阵求主成分, 否则使用协方差矩阵求主成分.

返回值为:

- sdev: 特征值的平方根.

- rotation: 每列为对应特征值的特征向量. princomp() 的返回值中为 loadings. 与 prcomp() 预测值(即计算出的主成分)稍微不同. 详细见例子.
- x: 如果"retx" 为 true, 为旋转数据(rotated data), 即中心化并归一化(如果scaled)乘以rotation矩阵). 使用 predict(prcomp(...)) 的结果就是返回这个矩阵x.
- center, scale: 中心化和归一化.

下面是几个相关的函数

- summary()
- predict(): 主成分向量. princomp() 与 prcomp() 预测值(即计算出的主成分向量)稍微不同. 详细见例子.
- loadings() 只用于 princomp()
- screeplot() 碎石图
- biplot() 主成分的散点图

下面是主成分的计算

```
# 数据
> X=USArrests
> X
             Murder Assault UrbanPop Rape
Alabama        13.2     236       58 21.2
Alaska         10.0     263       48 44.5
Arizona         8.1     294       80 31.0
...
West Virginia   5.7      81       39  9.3
Wisconsin       2.6      53       66 10.8
Wyoming         6.8     161       60 15.6

=======================
# 手工计算
> c=cor(X)
```

```
> c
           Murder   Assault   UrbanPop     Rape
Murder   1.00000000 0.8018733 0.06957262 0.5635788
Assault  0.80187331 1.0000000 0.25887170 0.6652412
UrbanPop 0.06957262 0.2588717 1.00000000 0.4113412
Rape     0.56357883 0.6652412 0.41134124 1.0000000
> eigen(c)
$values  # 特征值
[1] 2.4802416 0.9897652 0.3565632 0.1734301

$vectors # 特征向量(载荷矩阵)
         [,1]       [,2]       [,3]       [,4]
[1,] -0.5358995  0.4181809 -0.3412327  0.64922780
[2,] -0.5831836  0.1879856 -0.2681484 -0.74340748
[3,] -0.2781909 -0.8728062 -0.3780158  0.13387773
[4,] -0.5434321 -0.1673186  0.8177779  0.08902432

> e=eigen(c)
# 矩阵正交
> t(e$vectors)%*%e$vectors
             [,1]         [,2]         [,3]         [,4]
[1,]  1.000000e+00  1.010205e-16 9.343112e-17 -7.732394e-17
[2,]  9.722583e-17  1.000000e+00 2.534323e-17 -1.257742e-16
[3,]  9.343112e-17  5.149960e-19 1.000000e+00  8.527250e-17
[4,] -1.071056e-16 -1.257742e-16 8.952799e-17  1.000000e+00
# 产生 diag(2.4802416 0.9897652 0.3565632 0.1734301)
> t(e$vectors)%*%c%*%e$vectors
             [,1]         [,2]         [,3]         [,4]
[1,]  2.480242e+00  1.338502e-15 -4.781332e-17 -4.455597e-16
[2,]  1.373047e-15  9.897652e-01 -3.856236e-16 -2.785095e-16
[3,] -1.435484e-16 -3.969671e-16  3.565632e-01  2.092510e-16
[4,] -2.162255e-16 -3.227534e-16  1.710769e-16  1.734301e-01

# 计算标准化的主成分(与 prcomp() 函数的预测结果一样, 但是与 princomp() 稍微不同)
> scale( as.matrix(X))%*%e$vectors
                  [,1]       [,2]       [,3]        [,4]
Alabama      -0.97566045  1.12200121 -0.43980366  0.154696581
Alaska       -1.93053788  1.06242692  2.01950027 -0.434175454
Arizona      -1.74544285 -0.73845954  0.05423025 -0.826264240
Arkansas      0.13999894  1.10854226  0.11342217 -0.180973554
```

```
California     -2.49861285 -1.52742672  0.59254100 -0.338559240
...
Washington      0.21472339 -0.96037394  0.61859067 -0.218628161
West Virginia   2.08739306  1.41052627  0.10372163  0.130583080
Wisconsin       2.05881199 -0.60512507 -0.13746933  0.182253407
Wyoming         0.62310061  0.31778662 -0.23824049 -0.164976866

=======================
# prcomp() 的用法
> p=prcomp(USArrests, scale=T)
> p
Standard deviations: # 特征值
[1] 1.5748783 0.9948694 0.5971291 0.4164494

Rotation: # 特征向量矩阵
              PC1        PC2        PC3         PC4
Murder   -0.5358995  0.4181809 -0.3412327  0.64922780
Assault  -0.5831836  0.1879856 -0.2681484 -0.74340748
UrbanPop -0.2781909 -0.8728062 -0.3780158  0.13387773
Rape     -0.5434321 -0.1673186  0.8177779  0.08902432

# 第一行为特征值. 第二行为主分量百分比, 第三行为累加主分量百分比
> summary(p)
Importance of components:
                          PC1   PC2    PC3    PC4
Standard deviation     1.57 0.995 0.5971 0.4164
Proportion of Variance 0.62 0.247 0.0891 0.0434
Cumulative Proportion  0.62 0.868 0.9566 1.0000

# 计算主成分. 注意与手工计算一样. Q分析作图的数据. 把个体分类.
> predict(p) # 等于p$x, 见返回值的说明
                    PC1        PC2        PC3         PC4
Alabama       -0.97566045  1.12200121 -0.43980366  0.154696581
Alaska        -1.93053788  1.06242692  2.01950027 -0.434175454
Arizona       -1.74544285 -0.73845954  0.05423025 -0.826264240
Arkansas       0.13999894  1.10854226  0.11342217 -0.180973554
...
Washington     0.21472339 -0.96037394  0.61859067 -0.218628161
West Virginia  2.08739306  1.41052627  0.10372163  0.130583080
```

```
Wisconsin      2.05881199 -0.60512507 -0.13746933  0.182253407
Wyoming        0.62310061  0.31778662 -0.23824049 -0.164976866

# 绘图查看
> screeplot(p)
> biplot(p)

=======================
# princomp() 用法. 下面的相当于 prcomp(USArrests, scale=T)
> p1=princomp(USArrests, cor = TRUE)
> p1
Call:
princomp(x = USArrests, cor = TRUE)

Standard deviations:
   Comp.1    Comp.2    Comp.3    Comp.4
1.5748783 0.9948694 0.5971291 0.4164494

 4  variables and  50 observations.
> summary(p1)
Importance of components:
                       Comp.1    Comp.2    Comp.3     Comp.4
Standard deviation     1.5748783 0.9948694 0.5971291 0.41644938
Proportion of Variance 0.6200604 0.2474413 0.0891408 0.04335752
Cumulative Proportion  0.6200604 0.8675017 0.9566425 1.00000000

# 载荷矩阵.
> loadings(pr)

Loadings:
         Comp.1 Comp.2 Comp.3 Comp.4
Murder   -0.536  0.418 -0.341  0.649
Assault  -0.583  0.188 -0.268 -0.743
UrbanPop -0.278 -0.873 -0.378  0.134
Rape     -0.543 -0.167  0.818

               Comp.1 Comp.2 Comp.3 Comp.4
SS loadings     1.00   1.00   1.00   1.00
Proportion Var   0.25   0.25   0.25   0.25
Cumulative Var   0.25   0.50   0.75   1.00
```

```
# 预测. 注意与手工计算稍微不同
> predict(pr)
                Comp.1     Comp.2     Comp.3      Comp.4
Alabama      -0.98556588  1.13339238 -0.44426879  0.156267145
Alaska       -1.95013775  1.07321326  2.04000333 -0.438583440
Arizona      -1.76316354 -0.74595678  0.05478082 -0.834652924
Arkansas      0.14142029  1.11979678  0.11457369 -0.182810896
...
Washington    0.21690338 -0.97012418  0.62487094 -0.220847793
West Virginia 2.10858541  1.42484670  0.10477467  0.131908831
Wisconsin     2.07971417 -0.61126862 -0.13886500  0.184103743
Wyoming       0.62942666  0.32101297 -0.24065923 -0.166651801
```

38.5 主成分作图

主成分分析的数据, 每一行为一个分类单位(实体, 个体), 每一列为一个属性(性状, 指标).

对个体作图称为Q分析. 对属性作图称为R分析. 参考文献 [11] 第二章 第四节 主成分分析

38.5.1 R分析(属性作图)

计算得到的r矩阵是特征向量矩阵乘以特征值, 其的意义为: 为第i属性向量在第j个主成分向量上的投影, 即因子负载(factor loading). 例如: 第一个性状在第一个主成分上的负载为 -0.8439764, 在第二个主成分上的负载为 -0.9184432, 等等

```
> p=prcomp(USArrests, scale=T)
> r=t(p$rotation)*p$sdev; r # 注意列标题已经失去意义
      Murder    Assault   UrbanPop       Rape
PC1 -0.8439764 -0.9184432 -0.4381168 -0.85583939
PC2  0.4160354  0.1870211 -0.8683282 -0.16646019
PC3 -0.2037600 -0.1601192 -0.2257242  0.48831900
```

```
PC4  0.2703705 -0.3095916  0.0557533  0.03707412
> plot(r[1:2,]) # 绘制前2个性状的负载的图形. 看看哪些性状距离接近的就成为一类, 说明性状之间有关系.
```

38.5.2 Q分析(个体作图)

即主成分向量作图.

选择前2个主成分向量作图. 哪些个体距离接近的可以成为一类. 与聚类分析结果可以相互参照.

```
> p=prcomp(USArrests, scale=T)
> plot(p$x[,1:2]) # p$x 就是 loadings(p)
```

38.6 主成分回归

参考 [21] page 516.

当自变量出现多重共线性时, 经典回归方法做回归系数的最小二乘估计效果一般较差. 采用主成分回归能够克夫经典回归的不足.

下面是法国 1949 至 1959 共 11 年的经济分析数据. y 为进口总额. x1 为国内总产值, x2 为存储量, x3 为总消费量.(单位: 10 亿法郎)

```
x1=c(149.3, 161.2, 171.5, 175.5, 180.8, 190.7,
    202.1, 212.4, 226.1, 231.9, 239.0)
x2=c(4.2, 4.1, 3.1, 3.1, 1.1, 2.2, 2.1, 5.6, 5.0, 5.1, 0.7)
x3=c(108.1, 114.8, 123.2, 126.9, 132.1, 137.7,
    146.0, 154.1, 162.3, 164.3, 167.6)
y=c(15.9, 16.4, 19.0, 19.1, 18.8, 20.4, 22.7,
   26.5, 28.1, 27.6, 26.3)
```

38.6.1 线性回归

```
> r1 <- lm(y~x1+x2+x3)
> summary(r1)

Call:
lm(formula = y ~ x1 + x2 + x3)

Residuals:
    Min      1Q  Median      3Q     Max
-0.52367 -0.38953  0.05424  0.22644  0.78313

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -10.12799    1.21216  -8.355  6.9e-05 ***
x1           -0.05140    0.07028  -0.731 0.488344
x2            0.58695    0.09462   6.203 0.000444 ***
x3            0.28685    0.10221   2.807 0.026277 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4889 on 7 degrees of freedom
Multiple R-squared: 0.9919,     Adjusted R-squared: 0.9884
F-statistic: 285.6 on 3 and 7 DF,  p-value: 1.112e-07
```

回归方程为

$$y = -10.13 - 0.05 * x1 + 0.59 * x2 + 0.29 * x3$$

发现进口 y 与国内生产总值是负的关系, 这不太合理. 原因是三个变量存在共线性.

38.6.2 主成分分析

下面对三个变量使用主成分分析

```
> p<-princomp(~x1+x2+x3,cor=T)
```

```
> summary(p,loadings=TRUE)
Importance of components:
                       Comp.1    Comp.2      Comp.3
Standard deviation     1.413915 0.9990767 0.0518737839
Proportion of Variance 0.666385 0.3327181 0.0008969632
Cumulative Proportion  0.666385 0.9991030 1.0000000000

Loadings:
   Comp.1 Comp.2 Comp.3
x1  0.706        0.707
x2       -0.999
x3  0.707       -0.707
```

第一主成分是国内生产总值和总消费(x1, x3), 因此称第一主成分为产销因子. 第二主成分与存储(x2)相关, 称存储因子.

注意

$$\lambda_3^2 = 0.05^2 = 0.0025 \approx 0$$

故变量存在共线性.

38.6.3 主成分回归

取前 2 个主成分做回归

```
> pre<-predict(p)
> z1<-pre[,1]
> z2<-pre[,2]
> r2<-lm(y~z1+z2)
> summary(r2)

Call:
lm(formula = y ~ z1 + z2)

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-0.89838 -0.26050  0.08435  0.35677  0.66863

Coefficients:
          Estimate Std. Error t value Pr(>|t|)
(Intercept)  21.8909     0.1658 132.006 1.21e-14 ***
z1            2.9892     0.1173  25.486 6.02e-09 ***
z2           -0.8288     0.1660  -4.993  0.00106 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.55 on 8 degrees of freedom
Multiple R-squared: 0.9883,     Adjusted R-squared: 0.9853
F-statistic: 337.2 on 2 and 8 DF,  p-value: 1.888e-08
```

回归方程变为

$$y = 21.89 + 2.99 * z1 - 0.83 * z2$$

38.6.4 得到与原自变量的关系式

下面我们要得到 y 与 x1, x2, x3 的关系. 由于

$$z_i = Xa_i = a_{1i}X_1 + a_{2i}X_2 + a_{i3}X_3 = a_{1i}x1 + a_{2i}x2 + a_{i3}x3$$

将 z_1, z_2 带入回归方程既得 y 与 x1, x2, x3 的关系式.

Chapter 39

因子分析

数学比较复杂, 具体请参考 [21] 9.2 章因子分析. [13] 14.5 因子分析.

因子分析把数据看作公共因子, 特殊因子和误差构成. 主成分分析把方差划分为不同的正交成分, 因子分析则把方差划分为不同的起因因子. 其特征值计算是从相关矩阵出发, 且将主成分转换为因子, 并计算出因子得分. 目前在心理学, 生物学, 经济学中广泛使用.

39.1 数学模型

下面简单解释一下.

记号与主成分分析中的记号一致. 数学模型为

$$X = \mu + AF + e$$

即

$$\begin{aligned} X_1 - \mu_1 &= a_{11}f_1 + a_{12}f_2 + \cdots + a_{1m}f_m + e_1 \\ &\vdots \\ X_p - \mu_p &= a_{p1}f_1 + a_{p2}f_2 + \cdots + a_{pm}f_m + e_p \end{aligned}$$

其中

$X_{(n*p)}$ 为 p 维原始数据, $Var(X) = \sum = (\sigma_{ij})_{p*p}$

$\mu = \bar{X_1}, \cdots, \bar{X_p}$,

$A = (a_{ij})_{(p*m)}$ 为因子载荷矩阵,

$F = f_1, \cdots, f_m$ 为公共因子向量,

$e = e_1, \cdots, e_p$ 为特殊因子.

$m <= p$ 为公共因子数.

通常假设

$$E(F) = 0, Var(F) = I_m$$
$$E(e) = 0, Var(e) = D = diag(\sigma_1^2, \cdots, \sigma_p^2)$$
$$Cov(F, e) = 0$$

故公共因子 F 彼此不相关且具有单位矩阵.

特殊因子 e 也不相关且与 F 也不相关.

$\sum$ 可以分解为

$$\sum = AA^T + D$$

因子载荷矩阵 A 不是唯一的, 这样可以通过因子旋转使得新因子有更好的实际意义.

A 的统计意义如下

1.

$$Cov(X, F) = A$$
$$Cov(X_i, f_j) = a_{ij}$$

即 a_{ij} 是第 i 个变量与第 j 个公共因子的相关系数. 即度量 X_i 可以由 f_j 表示的强度.

2. 令 $h_i^2 = \sum_{j=1}^{m} a_{ij}^2$, 则 h_i^2 反映了公共因子对 X_i 的方差贡献, 称为 X_i 的共同度(communality)或共性方差(common variance). 而 $\sigma_i^2 = var(e_i)$ 为 X_i 的特殊方差, 是特殊因子 e_i 对 X_i 的贡献.

   当 X 标准化后, 此时

$$h_i^2 + \sigma_i^2 = 1, \quad i = 1, \cdots, p$$

39.2 例子

R 中函数 factanal() 执行因子分析. 用法

```
factanal(x, factors, data = NULL, covmat = NULL, n.obs = NA,
        subset, na.action, start = NULL,
        scores = c("none", "regression", "Bartlett"),
        rotation = "varimax", control = NULL, ...)
```

- x: 公式, 或数据
- factors: 因子个数
- covmat: 样本协方差矩阵或相关矩阵. 此时不需要 x
- scores: 因子得分方法. scores="regression" 表示用回归方法计算因子得分. scores="Bartlett" 表示用 Bartlett 方法计算因子得分.
- rotation: 表示旋转. 缺省为方差最大旋转

下面是 R 的例子

```
# 数据, 可以假设为某公司对18个新员工的6项个人能力打分
v1 <- c(1,1,1,1,1,1,1,1,1,1,3,3,3,3,3,4,5,6)
v2 <- c(1,2,1,1,1,1,2,1,2,1,3,4,3,3,3,4,6,5)
```

```
v3 <- c(3,3,3,3,3,1,1,1,1,1,1,1,1,1,1,5,4,6)
v4 <- c(3,3,4,3,3,1,1,2,1,1,1,1,2,1,1,5,6,4)
v5 <- c(1,1,1,1,1,3,3,3,3,3,1,1,1,1,1,6,4,5)
v6 <- c(1,1,1,2,1,3,3,3,4,3,1,1,1,2,1,6,5,4)
m1 <- cbind(v1,v2,v3,v4,v5,v6)

> cor(m1)
        v1       v2       v3       v4       v5       v6
v1 1.0000000 0.9393083 0.5128866 0.4320310 0.4664948 0.4086076
v2 0.9393083 1.0000000 0.4124441 0.4084281 0.4363925 0.4326113
v3 0.5128866 0.4124441 1.0000000 0.8770750 0.5128866 0.4320310
v4 0.4320310 0.4084281 0.8770750 1.0000000 0.4320310 0.4323259
v5 0.4664948 0.4363925 0.5128866 0.4320310 1.0000000 0.9473451
v6 0.4086076 0.4326113 0.4320310 0.4323259 0.9473451 1.0000000

# 默认不计算得分
> factanal(m1, factors=3) # varimax is the default
Call:
factanal(x = m1, factors = 3)

Uniquenesses:
   v1    v2    v3    v4    v5    v6
0.005 0.101 0.005 0.224 0.084 0.005

Loadings:
   Factor1 Factor2 Factor3
v1 0.944   0.182   0.267
v2 0.905   0.235   0.159
v3 0.236   0.210   0.946
v4 0.180   0.242   0.828
v5 0.242   0.881   0.286
v6 0.193   0.959   0.196

               Factor1 Factor2 Factor3
SS loadings      1.893   1.886   1.797
Proportion Var   0.316   0.314   0.300
Cumulative Var   0.316   0.630   0.929

The degrees of freedom for the model is 0 and the fit was 0.4755
```

结果中

- uniquenesses: 特殊方差. 即 $diag(cov(e))$
- loadings: 因子载荷矩阵, 即矩阵 A. 其中, Factor1 的 v1, v2 接近 1, Factor2 的 v5, v6 接近 1, Factor3 的 v3, v4 接近 1. 具体问题中可以根据经验总结其代表的实际意义.
- SS loadings: 公共因子 f_i 对变量 $X_1, \cdots, X_p$ 的总方差贡献.
- Proportion Var: 方差贡献率. 可以看到三个因子的贡献率差不多.
- Cumulative Var: 累积方差贡献率. 总的贡献率达到 0.929.

39.2.1 因子得分

得到公共因子 F 和因子载荷 A 后, 应该反过来考察每个样本的得分情况. 这样可以挑选某个因子得分较高或较低(或某几个因子得分都高/都低, 或指定哪个因子得分较高)的个体进一步研究

```
# 计算得分
> f<-factanal(m1, factors=3, scores="Bartlett")
> names(f)
 [1] "converged"    "loadings"     "uniquenesses" "correlation"  "criteria"
 [6] "factors"      "dof"          "method"       "scores"       "n.obs"
[11] "call"
> f$scores
       Factor1    Factor2    Factor3
 [1,] -0.9039949 -0.9308984  0.9475392
 [2,] -0.8685952 -0.9328721  0.9352330
 [3,] -0.9082818 -0.9320093  0.9616422
 [4,] -1.0021975 -0.2529689  0.8178552
 [5,] -0.9039949 -0.9308984  0.9475392
 [6,] -0.7452711  0.7273960 -0.7884733
 [7,] -0.7098714  0.7254223 -0.8007795
 [8,] -0.7495580  0.7262851 -0.7743704
 [9,] -0.8080740  1.4033517 -0.9304636
```

```
[10,] -0.7452711  0.7273960 -0.7884733
[11,]  0.9272282 -0.9307506 -0.8371538
[12,]  0.9626279 -0.9327243 -0.8494600
[13,]  0.9229413 -0.9318615 -0.8230509
[14,]  0.8290256 -0.2528211 -0.9668378
[15,]  0.9272282 -0.9307506 -0.8371538
[16,]  0.4224366  2.0453079  1.2864761
[17,]  1.4713902  1.2947716  0.5451562
[18,]  1.8822320  0.3086244  1.9547752

# 绘制前两个因子的散点图
> plot(f$scores[, 1:2], type="n")
> text(f$scores[,1], f$scores[,2])
```

39.2.2 与主成分分析对照

下面是主成分分析的结果, 做对照. 可以看到无明显主成分

```
# 主成分分析的结果
> prcomp(m1)
Standard deviations:
[1] 3.0368683 1.6313757 1.5818857 0.6344131 0.3190765 0.2649086

Rotation:
        PC1        PC2       PC3       PC4       PC5        PC6
v1 0.4168038 -0.52292304  0.2354298 -0.2686501  0.5157193 -0.39907358
v2 0.3885610 -0.50887673  0.2985906  0.3060519 -0.5061522  0.38865228
v3 0.4182779  0.01521834 -0.5555132 -0.5686880 -0.4308467 -0.08474731
v4 0.3943646  0.02184360 -0.5986150  0.5922259  0.3558110  0.09124977
v5 0.4254013  0.47017231  0.2923345 -0.2789775  0.3060409  0.58397162
v6 0.4047824  0.49580764  0.3209708  0.2866938 -0.2682391 -0.57719858
```

Chapter 40

典型相关分析

此部分主要参考 "统计建模与R软件" [21]

典型相关分析(canonical correlation analysis)是用于分析两组随机变量之间的相关程度的一种统计方法, 可以有效揭示两组随机变量之间的线性关系. 这个方法由 Hotelling (1935) 首先提出的.

如果需要寻找 X (p维) Y(q维) 的相关关系, 普通做法是列出$p*q$ 个相关系数, 然后进行分析. 缺点是不易把握.

典型相关分析原理是分别寻找 X Y 的线性组合

$$U_1 = Xa_1, V_1 = Yb_1$$

使其具有最大相关(注意并不唯一), 称 U_1, V_1 的相关系数为第一典型相关系数.其中 $a_1 = a_{11}, \cdots, a_{1p}$ $b_1 = b_{11}, \cdots, b_{1p}$.

然后如果存在 a_k, b_k 使得

1. $U_k = Xa_k, V_k = Yb_k$ 与前面的 $k-1$ 对典型变量都不相关
2. $Var(U_k) = 1, Var(V_k) = 1$
3. U_k, V_k 相关系数最大

称 U_k, V_k 为第 k 对典型变量, 称它们的相关系数为第 k 典型相关系数.

下面是 [21] 的例子.

X1: 体重. X2: 腰围. X3: 脉搏

Y1: 引体向上. Y2: 仰卧起坐. Y3: 跳跃次数.

```
test<-data.frame(
  X1=c(191, 193, 189, 211, 176, 169, 154, 193, 176, 156,
      189, 162, 182, 167, 154, 166, 247, 202, 157, 138),
  X2=c(36, 38, 35, 38, 31, 34, 34, 36, 37, 33,
      37, 35, 36, 34, 33, 33, 46, 37, 32, 33),
  X3=c(50, 58, 46, 56, 74, 50, 64, 46, 54, 54,
      52, 62, 56, 60, 56, 52, 50, 62, 52, 68),
  Y1=c( 5, 12, 13,  8, 15, 17, 14,  6,  4, 15,
       2, 12,  4,  6, 17, 13,  1, 12, 11,  2),
  Y2=c(162, 101, 155, 101, 200, 120, 215,  70,  60, 225,
      110, 105, 101, 125, 251, 210,  50, 210, 230, 110),
  Y3=c(60, 101, 58, 38, 40, 38, 105, 31, 25, 73,
      60, 37, 42, 40, 250, 115, 50, 120, 80, 43)
)
> test<-scale(test)
> ca<-cancor(test[,1:3],test[,4:6])
> ca
$cor
[1] 0.79560815 0.20055604 0.07257029

$xcoef
         [,1]       [,2]       [,3]
X1 -0.17788841 -0.43230348 -0.04381432
X2  0.36232695  0.27085764  0.11608883
X3 -0.01356309 -0.05301954  0.24106633

$ycoef
        [,1]       [,2]       [,3]
Y1 -0.0801801 -0.08615561 -0.29745900
Y2 -0.2418067  0.02833066  0.28373986
Y3  0.1643596  0.24367781 -0.09608099

$xcenter
         X1          X2          X3
```

```
 2.289835e-16  4.315992e-16 -1.778959e-16

$ycenter
         Y1          Y2          Y3
 1.471046e-16 -1.776357e-16  4.996004e-17
```

其中

- cor: 典型相关系数. 第1,2,3典型相关系数分别为: 0.79560815 0.20055604 0.07257029
- xcoef: 对应于X的系数.
- ycoef: 对应于Y的系数.
- xcenter: X的中心, 即均值. 因为已经标准化, 故为0
- ycenter: Y的中心, 即均值. 因为已经标准化, 故为0

计算典型变量下的得分

```
> U<-as.matrix(test[, 1:3])%*% ca$xcoef
> V<-as.matrix(test[, 4:6])%*% ca$ycoef
> cor(U[,1],V[,1])
[1] 0.7956082
> cor(U,V)
             [,1]         [,2]         [,3]
[1,]  7.956082e-01  3.069378e-17  1.386142e-16
[2,] -4.049495e-17  2.005560e-01 -4.029166e-17
[3,] -9.089002e-17 -3.131566e-17  7.257029e-02
> diag(cor(U,V))
[1] 0.79560815 0.20055604 0.07257029

# U1 V1 基本在一条直线上. 其它则分散
> plot(U[,1],V[,1])
> plot(U[,2],V[,2])
```

即

$$
\begin{aligned}
U_1 &= -0.178X_1 + 0.362X_2 - 0.136X_3 \\
V_1 &= -0.08Y_1 - 0.242Y_2 + 0.164Y_3 \\
\rho(U_1, V_1) &= 0.7956
\end{aligned}
$$

我们得到结论, 利用 U_1 可以预测 V_1, 即 X 与 Y 存在一定的线性关系.

40.1 TODO: 典型相关系数的检验

Chapter 41

CFA 分析(Configural 频率分析)

41.1 介绍

http://en.wikipedia.org/wiki/Configural_frequency_analysis

CFA 分析是一个探索性数据分析方法。configural频率分析的目标是发现比预期偶然发生的模式显著多（这种模式被称为type）或显著较少（这种模式被称为 Antitypes）的模式.

41.2 一个例子

假设一个数据集描述 n 个病人, 存在 m 个病症 $s_1, ..., s_m$.便于分析, 假设病症是有/无二态的.

数据集的每个纪录就是 m 个数据 $(x_1, ..., x_m)$, $x_i = 0 or 1$. 0 为无病症 i, 1 为有病症 i.

每个纪录叫做一个 configuration.

记 C 为所有可能的 configuration 个数, 也就是说, $C = (0, 1)^m$.

那么数据集可以描述为可能的 configuration 的频率的观测

$f(c)$.

CFA 基本的想法是估计每个 configuration 的频率, 假设病症是独立的.

记 $p_i(1)$ 为表现为病症 i 的, $p_i(0)$ 为不表现病症 i 的概率. 所有病症独立的情况下, 我们有 configuration $c = (c_1, ..., c_m)$ 的期望频率

$$E(c) = n \prod_{i=1}^{m} p_i(c_i)$$

那么 $f(c)$ 和 $E(c)$ 可以使用统计检验来比较. 一般使用卡方检验, 二项检验, 或超几何分布检验.

如果发现 $f(c)$ 和 $E(c)$ 有显著差异, 那么当 $f(c) > E(c)$ c 叫做一个 type. 反之叫做一个 antitype. 如果没有显著差异, 那么 c 就不是 type 或 antitype, 或此时叫做 not classified.

c 容易有 I 型错误, 所以通常使用 Bonferroni-adjustment 校正 α 水平.

非二态数据

病态可以有多于二种, 但是有限, 此时

$$C = s_1 x ... x s_m$$

机会模型(first-order CFA)

若病症不独立, 那么可以使用 chance model, 也称为 first-order CFA.

对应的, 所有 configurations 有相同概率的模型称为 zero-order CFA.

41.3 cfa包

41.3.1 bcfa–bootstrap-CFA

bcfa 速度很慢.

返回值

```
cnt.antitype    Number of antiypes
cnt.type       Number of types
pct.types      Number of types in percent
cnt.sig        Number of significant results
pct.cnt.sig    Number of significant results in percent
```

```
> library(cfa)
> example(bcfa)

bcfa> # library(cfa) if not yet loaded
bcfa> # Some random configurations:
bcfa> configs<-cbind(c("A","B")[rbinom(250,1,0.3)+1],c("C","D")[rbinom(250,1,0.1
bcfa+         c("E","F")[rbinom(250,1,0.3)+1],c("G","H")[rbinom(250,1,0.1)+1])

bcfa> counts<-trunc(runif(250)*10)

bcfa> bcfa(configs,counts,runs=25)
       cnt.antitype cnt.type pct.types cnt.sig pct.cnt.sig
B D E G         25       0         0       0           0
B C F H         25       0         0       0           0
B C F G          1      24        96       0           0
B C E H          7      18        72       0           0
B C E G          8      17        68       0           0
A D F G          0      25       100      10          40
A D E H          8      17        68       3          12
A D E G         25       0         0       0           0
A C F H         25       0         0       0           0
A C F G         21       4        16       0           0
A C E H          0      25       100       1           4
A C E G         16       9        36       0           0
```

```
> configs
      [,1] [,2] [,3] [,4]
  [1,] "A"  "C"  "F"  "G"
  [2,] "A"  "C"  "F"  "G"
  [3,] "A"  "D"  "F"  "G"
  [4,] "A"  "C"  "F"  "G"
  [5,] "B"  "C"  "E"  "G"
  [6,] "A"  "C"  "E"  "G"
  [7,] "A"  "C"  "E"  "G"
......
[247,] "A"  "D"  "E"  "H"
[248,] "A"  "C"  "E"  "H"
[249,] "B"  "C"  "E"  "G"
[250,] "A"  "C"  "E"  "G"
> counts
  [1] 1 6 8 6 8 6 0 0 8 6 1 3 6 0 2 4 1 0 8 3 1 0 7 4 8 3 2 5 4 2 6 1 9 0 8 8 3
 [38] 0 5 3 4 6 3 1 7 3 5 1 5 3 3 0 1 2 6 7 3 4 8 3 5 3 0 1 8 9 4 6 6 1 9 2 9 1
 [75] 7 8 1 4 0 5 9 3 6 9 2 6 3 1 4 8 4 8 0 3 8 2 0 6 5 7 4 1 9 4 7 0 3 3 4 1 2
[112] 8 4 4 7 6 1 8 0 2 0 9 0 7 7 0 1 4 9 3 1 3 3 6 7 7 5 0 4 7 7 3 9 7 9 0 5 9
[149] 1 0 0 8 1 2 0 7 8 1 8 3 1 7 7 5 5 1 1 5 6 9 0 8 4 4 0 8 0 5 3 4 1 6 7 5 4
[186] 9 7 1 6 2 4 7 4 3 9 6 3 6 7 6 8 2 8 4 8 1 9 6 8 4 9 6 9 3 8 0 1 1 8 0 4 6
[223] 1 0 8 7 1 3 9 1 2 4 2 9 5 7 7 2 7 0 0 6 2 9 5 5 6 1 2 2
```

41.3.2 cfa

```
> example(cfa)

cfa> # library(cfa) if not yet loaded
cfa> # Some random configurations:
cfa> configs<-cbind(c("A","B")[rbinom(250,1,0.3)+1],c("C","D")[rbinom(250,1,0.1)+
cfa+         c("E","F")[rbinom(250,1,0.3)+1],c("G","H")[rbinom(250,1,0.1)+1])

cfa> counts<-trunc(runif(250)*10)

cfa> cfa(configs,counts)

*** Analysis of configuration frequencies (CFA) ***
```

```
    label   n    expected          Q      chisq     p.chisq sig.chisq
1  B D F G  25   6.1167994 0.0166682678 58.29441874 2.253753e-14     TRUE
2  B D E G   3  15.1241595 0.0107878104  9.71923387 1.823495e-03     TRUE
3  B C E H   4  14.6878157 0.0095060925  7.77715395 5.291109e-03    FALSE
4  A D F H   3   0.9343164 0.0018150829  4.56702741 3.259307e-02    FALSE
5  A C E H  44  33.2474910 0.0097241552  3.47744884 6.221058e-02    FALSE
6  A D F G   8  13.8460502 0.0051957781  2.46830702 1.161630e-01    FALSE
7  A C F H  18  13.4465808 0.0040454936  1.54192552 2.143314e-01    FALSE
8  A C F G 182 199.2708576 0.0183785485  1.49686977 2.211536e-01    FALSE
9  B C F H   3   5.9403250 0.0025950310  1.45539358 2.276645e-01    FALSE
10 B C E G 225 217.6652681 0.0079609849  0.24716066 6.190815e-01    FALSE
11 A C E G 500 492.7093462 0.0112807663  0.10788030 7.425704e-01    FALSE
12 A D E G  35  34.2352034 0.0006922710  0.01708516 8.960046e-01    FALSE
13 B C F G  89  88.0323156 0.0009207556  0.01063715 9.178546e-01    FALSE
           z        p.z sig.z
1   7.45295161 4.563017e-14  TRUE
2  -3.26790590 9.994583e-01  TRUE
3  -2.93822533 9.983495e-01  TRUE
4   1.62044840 5.256799e-02 FALSE
5   1.80461106 3.556778e-02 FALSE
6  -1.71591764 9.569114e-01 FALSE
7   1.11197338 1.330748e-01 FALSE
8  -1.38594823 9.171186e-01 FALSE
9  -1.41523915 9.215008e-01 FALSE
10  0.51508637 3.032463e-01 FALSE
11  0.40612911 3.423239e-01 FALSE
12  0.04595184 4.816743e-01 FALSE
13  0.05189183 4.793074e-01 FALSE


Summary statistics:

Total Chi squared        =  91.18054
Total degrees of freedom  =  11
p                      =  0
Sum of counts            =  1139

Levels:

V1 V2 V3 V4
 2  2  2  2
```

41.3.3 其它cfa

fCFA Stepwise CFA approaches

hcfa Hierachical analysis of configuration frequencies

lcfa hypergeometrical, nonparametrical exact test of significance according to Lindner

mcfa Two or more-sample CFA

Chapter 42

关联分析(Correspondence Analysis)

http://cran.r-project.org/web/packages/anacor/ 的文档 Simple and Canonical Correspondence Analysis Using the R Package anacor.pdf 包含原理与用法.

参考 http://cran.r-project.org/web/views/Psychometrics.html Correspondence Analysis (CA) 部分

原理 http://doc.mbalib.com/view/ff7b6a847fdf996236394e642124d204.html

例子和原理 http://doc.mbalib.com/view/917b9b21e937d2f0beaef70025a1677b.html

42.1 原理

关连分析又叫做对应分析,相应分析.

根据数据类型, 分为定性(分类数据)和定量(连续数据)分析

根据变量多少, 定性分析分为简单和多重分析.

起源于20世纪20-30年代.

因子分析有R和Q分析.

R分析将变量转换为变量因子,对变量进行降维和消除相关性.

Q法纳西将样品转换为样品因子, 是对样品降维和消除相关性.

因子分析的局限性:

1. R和Q分析是分开的. 会损失很多有用信息, 而且不能揭示指标与样品的相关性.

2. 数据量大的时侯, Q分析计算量很大, 例如 100 个样品, 则需计算 100*100 矩阵的特征值和特征向量.

3. 变量进行标准化, 这样只按照列进行的标准化对样品和变量是非对称的, 使得分析R和Q的关系困难.

关连分析从R分析出发, 直接获得Q分析的结果, 克服了Q分析计算量大的问题, 而且可以对数据进行不同的归一化, 同时分析变量和样品.

关连分析实际上将变量和样品的交叉表变换为一个散点图, 得到变量,样品的位置关系.

在市场细分,产品定位,品牌形象,满意度研究中应用较多.

42.2 r 包

ade4 函数 cca: Canonical Correspondence Analysis

ca

anacor

cocorresp

cncaGUI: 基于 tk 的 GUI, 需要安装 unixodbc-dev 和 tk-dev. 简单的界面, 选择文件分析.

caGUI: 基于 ca, 界面好一点, 但是有 bug

其它很多包里面包含 ca 分析. 请搜索 r cran 网站.

42.3 anacor

42.3.1 例: 眼睛/头发颜色(jointplot-graphplot)

数据为眼睛颜色(行)和头发颜色(列). 共收集了 5387 人.

```
> library(anacor)
> data("tocher")
> tocher
      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

# 行列使用不同的 scaling,
> res <- anacor(tocher, scaling = c("standard", "centroid"))
> res

CA fit:
Sum of eigenvalues:  0.2293315

Total chi-square value: 1240.039

Chi-Square decomposition:
            Chisq Proportion Cumulative Proportion
Component 1 1073.331     0.866              0.866
Component 2  162.077     0.131              0.996
```

```
Component 3    4.630     0.004               1.000

# 椭圆表示 95% 置信区间
plot(res, plot.type = "jointplot", xlim = c(-2, 1.5),
   ylim = c(-2,1.5), asp = 1)

# 图论绘图. 线的粗细表示频率, 即连结强度.
plot(res, plot.type = "graphplot", xlim = c(-2, 1.5),
   ylim = c(-2,1.5), wlines = 5, asp = 1)
```

椭圆表示 95% 置信区间.

图论绘图. 线的粗细表示频率, 即连结强度. 行/列分组的距离可以解释: black/dark 头发与 fair/red 之间接近. blue/light 也接近.

可以使用下面命令, 清晰的看到结果的分组情况.

```
> d<-tocher
> rownames(d)<-c('a','b','c','d')
> d
  Fair Red Medium Dark Black
a  326  38    241  110     3
b  688 116    584  188     4
c  343  84    909  412    26
d   98  48    403  681    85
> res <- anacor(d, scaling = c("standard", "centroid"))
> plot(res, plot.type = "jointplot", xlim = c(-2, 1.5),
+    ylim = c(-2,1.5), asp = 1)
> plot(res, plot.type = "graphplot", xlim = c(-2, 1.5),
+    ylim = c(-2,1.5), wlines = 5, asp = 1)
>
```

对数据卡方检验

```
> chisq.test(tocher)

        Pearson's Chi-squared test
```

```
data:  tocher
X-squared = 1240.039, df = 12, p-value < 2.2e-16
```

第一主成分占 86%.

42.3.2 例: 2D-5D(benzplot)

```
> data(bitterling)
> bitterling
    jk tu  hb chs  ft  qu  le hdp  sk  sn chf ffl
jk  654  2 172  56  27  25   1   5   0  46  14  18
tu  101  3  62  27   5   1   1   1   0   8   5   9
hb  171  7 197 130   0  25   0   8  14  18  14  12
chs  60  5 152 135   0   8   0   7  16  15  12   4
ft   19  4   0   0 419  19   0   4   0  17   5  11
qu   36  9  18   5  12 789 119   6  26  70   1  14
le    4  1   0   0   0  57 167  10   0   8   0   0
hdp  22  8  40  37   5 245   7  12 287  53   8  13
sk    3  4   7  38   0 120   8   2  19  28   4   0
sn   42  4  17  16  20  70  11   9   9 225  12  12
chf  18  6  10  13   6   5   0  11   0  24  97   9
ffl  27  6   6   5  10  13   0   3   0  10   8  29

> res1 <- anacor(bitterling, ndim = 2, scaling = c("Benzecri",
+ "Benzecri"))
> res2 <- anacor(bitterling, ndim = 5, scaling = c("Benzecri",
+ "Benzecri"))
> res1

CA fit:
Sum of eigenvalues:  1.329079
Benzecri RMSE rows:  0.0008334884
Benzecri RMSE columns:  0.0008199739

Total chi-square value: 14589.07

Chi-Square decomposition:
```

```
             Chisq Proportion Cumulative Proportion
Component 1  4026.287     0.276              0.276
Component 2  3730.218     0.256              0.532
Component 3  1996.814     0.137              0.669
Component 4  1635.673     0.112              0.781
Component 5  1145.514     0.079              0.859
Component 6   904.313     0.062              0.921
Component 7   832.702     0.057              0.978
Component 8   284.566     0.020              0.998
Component 9    31.421     0.002              1.000
Component 10    1.357     0.000              1.000
Component 11    0.206     0.000              1.000

> res2

CA fit:
Sum of eigenvalues:  2.147791
Benzecri RMSE rows:  0.0002484621
Benzecri RMSE columns:  0.000225833

Total chi-square value: 14589.07

Chi-Square decomposition:
             Chisq Proportion Cumulative Proportion
Component 1  4026.287     0.276              0.276
Component 2  3730.218     0.256              0.532
Component 3  1996.814     0.137              0.669
Component 4  1635.673     0.112              0.781
Component 5  1145.514     0.079              0.859
Component 6   904.313     0.062              0.921
Component 7   832.702     0.057              0.978
Component 8   284.566     0.020              0.998
Component 9    31.421     0.002              1.000
Component 10    1.357     0.000              1.000
Component 11    0.206     0.000              1.000

> plot(res1, plot.type = "benzplot", main = "Benzecri Distances (2D)")
> plot(res2, plot.type = "benzplot", main = "Benzecri Distances (5D)")
```

5D 比较 2D 的提高: 理想是在对角线上. 这个绘图可以做为总

体的 goodness-of-fit plot 或解释单个的距离

42.3.3 例: glass(regplot)

glass 数据: 行表示father的职业, 列表示son的职业. 分类如下

'PROF' professional and high administrative

'EXEC' managerial and executive

'HSUP' higher supervisory

'LSUP' lower supervisory

'SKIL' skilled manual and routine nonmanual

'SEMI' semi-skilled manual

'UNSK' unskilled manual

```
> data(glass)
> glass
    PROF EXEC HSUP LSUP SKIL SEMI UNSK
PROF   50   19   26    8   18    6    2
EXEC   16   40   34   18   31    8    3
HSUP   12   35   65   66  123   23   21
LSUP   11   20   58  110  223   64   32
SKIL   14   36  114  185  714  258  189
SEMI    0    6   19   40  179  143   71
UNSK    0    3   14   32  141   91  106

> res <- anacor(glass)
> plot(res, plot.type = "regplot", xlab = "fathers occupation",
+ ylab = "sons occupation", asp = 1)
```

绘图: 左边是未 scaling 的结果, 右边是 scaling 的结果.

纵坐标是 sons 的数据, 横坐标是 fathers.

方格表示对应的频率.

红线表示 fathers 的职业的频率, 对应的son的期望职业也计算出来.

蓝色线表示根据son的职业计算的父亲的职业.

单调性是因为职业分类根据table来的, 变量相关比较高.$\chi^2 = 1361.742, df = 36, p < 0.000$

scaling 的结果:

方格间距不整齐: 因为 scaling 的原因.

简单 CA 的一个特点就是行列期望的总是线性回归. 这个现象又叫做 bilinearizability.

42.3.4 Canonical CA(orddiag-transplot)

数据包含三个疾病组: 精神分裂, 狂躁, 抑郁.

四个推测的症状: anxiety suspicion, schizophrenic type of thought disorders, and delusions of guilt. 每个是 0,1 表示有无.

四个症状组合为16个(2^4)不同的模式.

620 个病人, 疾病与症状模式表示为 table

```
> data(maxwell)
> maxwell
$table
  schizophrenic manic.depressive anxiety.disorder
1            38              69               6
2             4              36               0
3            29               0               0
4             9               0               0
5            22               8               1
```

```
6            5             9             0
7           35             0             0
8            8             2             0
9           14            80            92
10           3            45             3
11          11             1             0
12           2             2             0
13           9            10            14
14           6            16             1
15          19             0             0
16          10             1             0

$row.covariates
   anxiety suspicion thought.disorders delusions.of.guilt
1        0         0                 0                 0
2        0         0                 0                 1
3        0         0                 1                 0
4        0         0                 1                 1
5        0         1                 0                 0
6        0         1                 0                 1
7        0         1                 1                 0
8        0         1                 1                 1
9        1         0                 0                 0
10       1         0                 0                 1
11       1         0                 1                 0
12       1         0                 1                 1
13       1         1                 0                 0
14       1         1                 0                 1
15       1         1                 1                 0
16       1         1                 1                 1

>  res <- anacor(maxwell$table, row.covariates = maxwell$row.covariates,
+ scaling = c("Goodman", "Goodman"))

res$datname           res$col.scores        res$right.singvec     res$col.acov
res$tab               res$chisq.decomp      res$eigen.values      res$cancoef
res$ndim              res$chisq             res$scaling           res$sitescores
res$row.covariates    res$singular.values   res$bdmat             res$isetcor
res$col.covariates    res$se.singular.values res$rmse
res$row.scores        res$left.singvec      res$row.acov
```

```
> res

CA fit:
Sum of eigenvalues:  0.6553413

Total chi-square value: 406.312

Chi-Square decomposition:
            Chisq Proportion Cumulative Proportion
Component 1 302.568     0.650              0.650
Component 2 103.743     0.223              0.872

plot(res, plot.type = "orddiag", asp = 1)
plot(res, plot.type = "transplot", legpos = "topright")
```

使用了对称(Goodman scaled)二维 scaling. 解释达到 87.2%.

第一个图(orddiag): 表示精神病分化为不同的方向, 之间没有显著的关系. 我们可以看到疾病模式如何关连到疾病的.

transformation plot: 显示感兴趣的模式. 黑线: dimension 1, 表现出预测的周期性. y轴(scores), 对于 1-2, 3-4, 5-6 没有太大的变化. 注意到这些成对的数据对应 delusions of guilt, 成对之间的差异比较大, 对应 thought disorders: 1-2 为 0, 3-4 为 1, 5-6 为 0,... 所以 dimension 1 主要反映 thought disorders.

dimension 2 反映基于 delusions of guilt 的之间的差异, 而且有一点向下的趋势(anxiety)

Chapter 43

通径分析

http://wenku.baidu.com/view/f113c9fe910ef12d2af9e797.html

http://en.wikipedia.org/wiki/Path_analysis_(statistics)

http://cran.r-project.org/web/views/Psychometrics.html

http://cran.r-project.org/web/views/Multivariate.html

r 包: lavaan

http://users.ugent.be/~yrosseel/lavaan/lavaanIntroduction.pdf

43.1 介绍

多元回归, 因子分析, 通径分析都是 SEM 的特例. 推荐直接使用 SEM 分析代替通径分析.

在研究多个相关变量的线性关系时, 除了多元回归, 偏相关分析, 还可以采用通径分析.

最直观的是通径图.

例如, 三个变量, x1, x2, y 存在线性关系, x1,x2彼此相关. 回归

方程为

$$y = b_0 + b_1 x_1 + b_2 x_2 + e$$

下面是简单的通径图, 箭头表示存在因果关系. x1,x2之间有双向箭头

按照是否可以测量, 变量分为显变量(manifest variable),隐变量(latent variable,不可直接测量的变量).

外生变量: 按照因果关系, 把原因变量称为外生变量(exogenous variable,或源变量 source,上游变量), 独立变量(independent).

内生变量: 箭头指向的结果变量称为内生变量(endogenous variable, dependent, 下游变量)

描述变量因果关系强弱的指标是通径系数. 定义为: 结果变量在独立变量上的偏回归系数.

当数据标准化时, 通径系数就是标准化偏回归系数.

双箭头表示相关性.

43.2 简单回归系数的通径分析

- 计算一个变量对最终变量的各种影响
- 以不同通径传递的间接影响
- 在控制某些变量的条件下的总影响的分解

标准化数据, 可以求任何两个变量的相关系数.

若结构合适, 那么任何两个变量之间的相关系数, 就是两个点之间所有连结上的数值(相关系数和通径系数)的乘积.

43.3 递归模型

全部为单向, 没有反馈的模型称为递归模型.

假设条件

- 各变量之间关系为线性,可加的因果关系
- 每个内生变量的误差与前置变量不得相关,同时也不得与其它内生变量的误差相关
- 因果关系必须为单向, 不能包含直接或间接反馈
- 变量测量不存在误差

43.4 通径图模型的识别(确认)

43.4.1 完全性

- 所有外生变量如果有相关性, 应该用双箭头表示
- 内生变量之间不画相关性双箭头, 若残差有相关性, 需要双箭头画出,没有的认为是不相关的
- 内生变量与外生变量之间有显著意义的直接作用, 应该用箭头画出.

43.4.2 恰好通径图

通径图中独立未知的参数(包括隐变量的方差,残差的方差)的个数恰好与样本中能得到的方程组的个数相等.

43.4.3 识别不足通径图

通径图中独立未知的参数多于样本中能得到的方程组的个数. 参数的解有无限组, 故不确定.

43.4.4 过度识别通径图

通径图中独立未知的参数少于样本中能得到的方程组的个数. 可以在待沽参数上附加各种条件以满足统计学要求.

43.4.5 原则

尽可能用较少的参数拟合样本数据, 这样也容易寻找专业解释.

43.5 非递归模型

- 模型中任何两个变量存在双向因果关系, 即有直接反馈.
- 某个变量存在自身反馈
- 存在间接反馈
- 内生变量的误差与其它变量相关

Chapter 44

结构方程模型(SEM)

http://en.wikipedia.org/wiki/Structural_equation_model

课件

结构方程模型分析 http://wenku.baidu.com/view/bf3de402bed5b9f3f90f1c2c.html

结构方程模型amos http://wenku.baidu.com/view/cb693b5c312b3169a451a4c7.html

结构方程模型——Lisrel的的初级应用 http://wenku.baidu.com/view/fe181ebfc77da26925c5b0da.html

44.1 介绍

很多社会,心理学涉及的变量不能准确,直接的测量, 称为潜变量, 如工作自主权,工作满意度等. 只能用其它变量间接测量.

多元回归, 因子分析, 通径分析都是 SEM 的特例.

44.2 软件

有一些专门的软件实现 SEM.

LISREL（Jöreskog and Sörbom, 1989, 1996）、

EQS（Bentler，1985，1995）、

AMOS（Arbuckle，1997）、

MPLUS（Muthén and Muthén, 1998）、

CALIS（Hartmann，1992）、

RAMONA（Browne，Mels，and Cowan，1994）

spss 和 SAS 也有相关的模块.

R 的介绍与可用的包

http://cran.r-project.org/web/views/Psychometrics.html

http://cran.r-project.org/web/views/Multivariate.html

http://socserv.socsci.mcmaster.ca/jfox/Misc/sem/SEM-paper.pdf

44.3 结构方程模型的一些资料

参考《结构方程模型的一些资料》[转贴 2006-08-02 14:27:32] http://tieba.baidu.com/f?kz=606583980

下面是一些语句

SEM是一门基於统计分析技术的研究方法学（statistical methodology），用以处理复杂的多变量研究数据的探究与分析。一般而言，结构方程模式被归类於高等统计学，属於多变量统计（multivariate statistics）的一环[3]，但是由於结构方程

模式有效整合了统计学的两大主流技术「因素分析」与「路径分析」，在瑞士籍的统计学者Karl Jöreskog於1970年代提出相关的概念，并首先发展分析工具LISREL软体之后，有关结构方程模式的原理讨论与技术发展便蔚为风潮，普遍成为社会与行为科学研究者必备的专门知识之一。

以Jöreskog（1973）为例，他所提出的SEM原始构想中，最重要的概念由两个部分所组成，第一是测量模型（measurement model），反应了观察变项与潜在变项之间的关系，其构成的数学模型是验证性因素分析；第二是结构关系的假设考验，透过结构模型（structure model），使潜在变项之间的关系可以路径分析的概念来讨论。当观察变项没有测量误差时，也就是当潜在变项不存在时，SEM对於结构关系的假设考验就完全等同於经济计量学的联立方程模型分析。

尽管SEM的发展风起云涌，但是同时也遭遇到许多的困境与挑战。其中质疑音量最大者依然是南加大心理系教授Norman Cliff，他在1983即已经以相当严峻的口吻质疑SEM的不当使用。Cliff（1983）对於SEM研究者所企图追求的因果论证，提出了四个方法学的警告，第一，研究者所获得的数据无法替我们完全确认或否认一个模型的正确性，因为模型是人为的，而且可以以各种方法重新定义。第二，具有时间性的先后次序证据并不代表因果。第三，潜在变项的命名是一个主观的历程，而非客观的事实，潜在变项的估计存在著名义谬误（nominalistic fallacy）的陷阱。第四，事后的解释与调整具有诚信与可信度的问题，也就是驳斥部分研究者大量使用模型修饰程序来获得理想契合度的不当作法。

Cliff的批评可以说是一种警世之语，事实上，Cliff也相当看重SEM的发展，他在1983年的文章中最后提到：

「.....最后，我必须再次强调，像LISREL这类的分析工具的确提供了一个空前的、史无前例的机会，使我们能够把这类的研究好好的做好。」（p.125）

在Cliff的观点中，最佳的SEM使用典范，是在恒等性检验的应用上，而恒等性的研究多与跨样本或纵贯研究的资料分析有关，因此也与Collins的立场一致。

从技术的层面来看，SEM并非单指某一种特定的统计方法，而是一套用以分析共变结构的技术的整合。SEM有时

以共变结构分析（covariance structure analysis）、共变结构模型（covariance structure modeling）等不同的名词存在，有时则单指因素分析模式的分析，以验证性因素分析（CFA）来称呼之；有时，研究者虽然以SEM的分析软体来执行传统的路径分析，进行因果模型（causal modeling）的探究，但不使用SEM的名义，事实上这也是SEM的重要应用之一。

44.4 结构方程模型假设条件

来自 http://baike.baidu.com/view/1501442.htm

合理的样本量（James Stevens的Applied Multivariate Statistics for the Social Sciences一书中说平均一个自变量大约需要15个case；Bentler and Chou (1987)说平均一个估计参数需要5个case就差不多了，但前提是数据质量非常好；这两种说法基本上是等价的；而Loehlin (1992)在进行蒙特卡罗模拟之后发现对于包含2 4个因子的模型，至少需要100个case，当然200更好；小样本量容易导致模型计算时收敛的失败进而影响到参数估计；特别要注意的是当数据质量不好比如不服从正态分布或者受到污染时，更需要大的样本量）

连续的正态内生变量（注意一种表面不连续的特例：underlying continuous；对于内生变量的分布，理想情况是联合多元正态分布即JMVN）

模型识别（识别方程）（比较有多少可用的输入和有多少需估计的参数；模型不可识别会带来参数估计的失败)

完整的数据或者对不完整数据的适当处理（对于缺失值的处理，一般的统计软件给出的删除方式选项是pairwise和listwise，然而这又是一对普遍矛盾：pairwise式的删除虽然估计到尽量减少数据的损失，但会导致协方差阵或者相关系数阵的阶数n参差不齐从而为模型拟合带来巨大困难，甚至导致无法得出参数估计；listwise不会有pairwise的问题，因为凡是遇到case中有缺失值那么该case直接被全部删除，但是又带来了数据信息量利用不足的问题——全杀了吧，难免有冤枉的；不杀吧，又难免影响整体局势）

模型的说明和因果关系的理论基础（实际上就是假设检验的逻辑——你只能说你的模型不能拒绝，而不能下定论说你的模型可以被接受）

44.5 建模过程

44.5.1 模型建构

观测变量与潜变量的关系.

潜变量之间的关系(哪些因子相关或直接效应).

44.5.2 模型拟合

模型参数的估计.

44.5.3 模型评价

解是不是恰当.

参数与预计的模型的关系是否合理(与模型假设符合)

44.5.4 模型评价

检验不同类型的整体拟合指数是否达到要求.

1. χ^2/DF 2:1 到 3:1 之间是可以接受的.
2. $P < 0.1$
3. NFI(规范拟合指数), NNFI(不规范拟合指数), CFI(比较拟合指数), IFI(增量拟合指数), GFI(拟合优度指数), AGFI(调整

后拟合优度指数), RFI(相对拟合指数), RMR(均方根残差), RMSEA(近似均方根残差)

普遍认为, 大样本情况下, NFI, NNFI, CFI, IFI, GFI, AGFI, RFI > 0.9, $RMR < 0.035$, $RMSEA < 0.08$, 表示拟合程度很好.

44.5.5 模型修正

依据理论或有关假设, 提出几个合理的先验模型

检查潜在变量与指标的关系, 建立测量模型, 有时可能增删或重组题目

对每个模型, 检查标准误, t值,标准化残差,修正指数, 各种拟合指数,据此修改模型并重复这一步

最好用另外一个样本检验

44.6 sem 的例子

详细用法见 ?sem

44.6.1 pathDiagram

```
R.DHP <- readMoments(diag=FALSE, names=c('ROccAsp', 'REdAsp', 'FOccAsp',
         'FEdAsp', 'RParAsp', 'RIQ', 'RSES', 'FSES', 'FIQ', 'FParAsp'))
.6247
.3269  .3669
.4216  .3275  .6404
.2137  .2742  .1124  .0839
.4105  .4043  .2903  .2598  .1839
.3240  .4047  .3054  .2786  .0489  .2220
.2930  .2407  .4105  .3607  .0186  .1861  .2707
.2995  .2863  .5191  .5007  .0782  .3355  .2302  .2950
.0760  .0702  .2784  .1988  .1147  .1021  .0931 -.0438  .2087
```

```
model.dhp <- specifyModel()
RParAsp  -> RGenAsp, gam11,  NA
RIQ      -> RGenAsp, gam12,  NA
RSES     -> RGenAsp, gam13,  NA
FSES     -> RGenAsp, gam14,  NA
RSES     -> FGenAsp, gam23,  NA
FSES     -> FGenAsp, gam24,  NA
FIQ      -> FGenAsp, gam25,  NA
FParAsp  -> FGenAsp, gam26,  NA
FGenAsp  -> RGenAsp, beta12, NA
RGenAsp  -> FGenAsp, beta21, NA
RGenAsp  -> ROccAsp,  NA,      1
RGenAsp  -> REdAsp,  lam21,  NA
FGenAsp  -> FOccAsp,  NA,      1
FGenAsp  -> FEdAsp,  lam42,  NA
RGenAsp <-> RGenAsp, ps11,   NA
FGenAsp <-> FGenAsp, ps22,   NA
RGenAsp <-> FGenAsp, ps12,   NA
ROccAsp <-> ROccAsp, theta1, NA
REdAsp  <-> REdAsp,  theta2, NA
FOccAsp <-> FOccAsp, theta3, NA
FEdAsp  <-> FEdAsp,  theta4, NA

sem.dhp <- sem(model.dhp, R.DHP, 329,
fixed.x=c('RParAsp', 'RIQ', 'RSES', 'FSES', 'FIQ', 'FParAsp'))

# 添加 file='aa', 将绘图写入.pdf和.dot文件
pathDiagram(sem.dhp, file='aa',min.rank='RIQ, RSES, RParAsp, FParAsp, FSES, FIQ',
max.rank='ROccAsp, REdAsp, FEdAsp, FOccAsp')
```

Part VI

非参数统计

“非参数统计”参考文献除了[14]，主要框架及内容参考的是W.J.Conover 著, 崔恒建译《实用非参数统计(第三版)》。R部分主要参考了《simpleR》《Statistics with R》等。

Chapter 45

非参数统计的应用条件和基本概念

45.1 什么时候使用非参数方法

之前的数据被假设来自某个潜在的分布, 这个分布的一般形式是已知的, 只是参数的具体值未知. 估计和检验方法都是基于这个分布, 来得到具体值的点或区间等. 这种方法通常被称为参数统计方法.

如果分布的形状未知, 中心基线定理似乎又不太合适, 例如样本数太少, 这时就必须使用非参数统计方法(nonparametric statistical method). 该方法对分布形状很少有要求.

45.2 次序统计量

次序统计量(order statistic): 把观测值$x_1, x_2, \cdots, x_n$按从小到大排列, 取值为第k个值$x^{(k)}$的随机变量称为秩为k的次序统计量(order statistic of rank k). 秩为1的次序统计量总是取最小值.

45.3 无偏检验

无偏检验(unbiasd test)是零假设不成立时拒绝零假设的概率大于等于零假设成立时拒绝零假设的概率.

45.4 相对效率

相对效率(relative efficient): 两个检验用来检验相同的零假设和备择假设, 其对应的α, β相等, 那么两个检验的样本容量之比定义为相对效率.

45.5 渐近相对效率(A.R.E)

渐近相对效率(asymptotic relative efficient, A. R. E): 令n1,n2为相同显著性水平, 相同功效的两个检验T1,T2的样本容量. 若α, β固定, 当n1趋于无穷时, 极限n2/n1存在, 且与α, β独立, 那么, n2/n1的极限称为第一个检验对第二个检验的渐近相对效率.

因为功效依赖于太多的因素, 为了寻找具有最大功效的检验, 通常要找出具有最大渐近相对效率的检验. 故A. R. E 是很重要的. 通常两个检验的A. R. E计算比较困难, 其全面研究本身可以写一本书.

45.6 保守性

若真实的显著性水平比规定的低, 称为保守的.

45.7 结(tie)

如果秩次的差的绝对值相同, 称为结. 有结和无结的计算公式不同. why???

45.8 一致对与不一致对

在成对的匹配中, 结局相同的对称为一致对(concordant pair). 结局不同的称为不一致对(discordant pair). 此例中, 有510+90=600 个一致对. 有 5+16=21 个不一致对. 一致对不提供信息, 故分析时抛弃之. 我们集中研究一致对.

不一致对中, 使用A处理后有事件发生而B处理后未发生, 称为A型不一致对. 否则称为B型不一致对.

45.9 二项比例齐性检验与列联表的独立性检验的关系

二项比例齐性检验(test for homogeneity of binomial proportion)检验不同的组的潜在的成功的比例(二项分布的参数 p)是否相同, 或等于某个给定的值. 零假设为$H_0 : p_1 = p_2 = p$对$H_1 : p_1 \neq p_2$. 显著性检验基于两个比例的差值$p_1 - p_2$, 若与零差别显著则拒绝零假设, 否则接受零假设. 可以使用正态逼近法和列联表法.

实际上列联表法是从不同的角度考察问题, 但是与正态逼近法的检验是相同的. 列联表的另外一个用处是检验列联表中两个变量的独立性. 例如两次问卷同一批人的饮食习惯, 在同一个人上做的两次调查是否有某种关联性. 这种检验也称为两个特征的独立性检验(test of independence, 也称为一致性检验, test of concordance)或关联性检验(test of association). 齐性检验与独立性检验的方法是相同的.

我们可以不加区分的使用这些方法, 只是最后对结果的解释不同罢了.

Chapter 46

基于二项分布的检验

46.1 二项分布参数的假设检验

46.1.1 p值与区间

参考binom.test(). 其中第一个参数可以为1个2值向量, 分别为成功和失败的次数; 也可以为2个值, 分别为成功和总试验次数. 结果给出了p值和区间.

例如, 现在的前列腺手术约有一半有副作用(p=0.5). FDA(food and drug administration, 食品与药物管理局)研究了一项新手术, 19例中只有3例有这种不良反应. 那么是否能够说新方法可以有效减轻副作用? 零假设为p=0.5, 备择假设为p¡0.5. 这是一个左单边检验. p-值为0.002213. 所以我们拒绝零假设. 结论是新方法可以有效减轻副作用.

```
> binom.test(c(3,16),p=0.5,alternative="less")

        Exact binomial test

data:  c(3, 16)
number of successes = 3, number of trials = 19, p-value = 0.002213
alternative hypothesis: true probability of success is less than 0.5
95 percent confidence interval:
```

```
 0.0000000 0.3594256
sample estimates:
probability of success
           0.1578947
```

在简单孟德尔遗传中, 后代有1/4是矮的, 3/4是高的. 我们做了一个杂交试验验证. 得到243个矮的植株, 682个高的. 零假设为p=1/4, 备择假设为$p \neq 1/4$. p-值为0.3825, 不能拒绝零假设. 所以接受后代矮的概率为1/4.

```
# 另外一个用法也可以 binom.test(682, 682 + 243, p = 3/4)
> binom.test(c(682, 243), p = 3/4)

        Exact binomial test

data:  c(682, 243)
number of successes = 682, number of trials = 925, p-value = 0.3825
alternative hypothesis: true probability of success is not equal to 0.75
95 percent confidence interval:
 0.7076683 0.7654066
sample estimates:
probability of success
           0.7372973
```

某省随机选20个高中, 其中7个达到优秀. 那么该省所有高中符合优秀的比例p的95%置信区间是什么? 因为此处不要求估计二项比例p, 那么p可以任意选择. p的95%区间为$[0.1539092, 0.5921885]$

```
> binom.test(c(7,13),p=7/20,conf.level = 0.95)
> binom.test(c(7,13),conf.level = 0.95) # 二项比例p默认为0.5

        Exact binomial test

data:  c(7, 13)
number of successes = 7, number of trials = 20, p-value = 0.2632
```

```
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.1539092 0.5921885
sample estimates:
probability of success
                0.35
```

46.1.2 功效与样本量

参考 power.prop.test() 好像实用非参数统计(第三版) 3.3 节称作容忍限.

下面是样本量为50, 零假设p=0.5, 备择假设p=0.75, 置信水平=0.95, 双边检验的功效为 0.74

```
> power.prop.test(n = 50, p1 = .50, p2 = .75)

     Two-sample comparison of proportions power calculation

              n = 50
             p1 = 0.5
             p2 = 0.75
      sig.level = 0.05
          power = 0.7401659
    alternative = two.sided

 NOTE: n is number in *each* group
```

46.2 二项比例齐性检验: prop.test

二项比例齐性检验(test for homogeneity of binomial proportion)检验不同的组的潜在的成功的比例(二项分布的参数 p)是否相同, 或等于某个给定的值. 零假设为$H_0 : p_1 = p_2 = p$对$H_1 : p_1 \neq p_2$. 显

著性检验基于两个比例的差值$p_1 - p_2$, 若与零差别显著则拒绝零假设, 否则接受零假设. 可以使用正态逼近法和列联表法. 实际上列联表法是从不同的角度考察问题, 但是与正态逼近法的检验是相同的. 列联表的另外一个用处是检验列联表中两个变量的独立性. 例如两次问卷同一批人的饮食习惯, 在同一个人上做的两次调查是否有某种关联性. 这种检验也称为两个特征的独立性检验(test of independence, 也称为一致性检验, test of concordance)或关联性检验(test of association). 齐性检验与独立性检验的方法是相同的. 我们可以不加区分的使用这些方法, 只是最后对结果的解释不同罢了.

例如: x 为成功的次数, y 为试验的总次数, 检验其概率是否相等. p 值必须与 x y 的长度相等.

```
> x=1:5
> y=11:15
> prop.test(x,y)

        5-sample test for equality of proportions without continuity
        correction

data:  x out of y
X-squared = 2.6169, df = 4, p-value = 0.6238
alternative hypothesis: two.sided
sample estimates:
prop 1    prop 2    prop 3    prop 4    prop 5
0.0909091 0.1666667 0.2307692 0.2857143 0.3333333

Warning message:
In prop.test(x, y) : Chi-squared approximation may be incorrect

# 检验单个的二项比例时, 可以给出置信区间(CI)
> prop.test(1,5,p=0.6)

    1-sample proportions test with continuity correction

data:  1 out of 5, null probability 0.6
X-squared = 1.875, df = 1, p-value = 0.1709
alternative hypothesis: true p is not equal to 0.6
95 percent confidence interval:
```

```
0.01052995 0.70120895
sample estimates:
p
0.2

Warning message:
In prop.test(1, 5, p = 0.6) : Chi-squared approximation may be incorrect
```

46.3 二项比例中样本量及功效的估计

46.3.1 独立样本

二项比例在指定的假设$p = p_1$下, 功效的正态近似为

$$power = \Phi[\sqrt{(p_0q_0)/(p_1q_1)}(z_{\alpha/2} + |p_0 - p_1|\sqrt{n}/\sqrt{p_0q_0})]$$

样本量为

$$n = \frac{p_0q_0(z_{1-\alpha/2} + z_{1-\beta}\sqrt{(p_1q_1)/(p_0q_0)})^2}{(p_1 - p_0)^2}$$

例子为: 若一个地区的发病率为0.0015, 期望通过某种方法使发病率降低20%, 双侧检验水平为0.05, 功效为0.80, 则应该多大的样本才能发现差异?(生物统计学基础 10.5)

```
> power.prop.test( p1 = 0.0015, p2 = .0012,power=0.8)

     Two-sample comparison of proportions power calculation

              n = 235147.3
             p1 = 0.0015
             p2 = 0.0012
      sig.level = 0.05
          power = 0.8
    alternative = two.sided
```

```
NOTE: n is number in *each* group
```

46.3.2 配对样本

TODO:

46.4 分位数检验

分位数检验可以使用二项检验来做. 这种方法可以用于次序(顺序)数据.

例如[1], 某大学新生参加入学考试, 其中15名新生的分数如下: 189 233 195 160 212 176 231 185 199 213 202 193 174 166 248. 认为这15名新生是随机样本. 已知多年来的新生成绩的上四分位数(第75百分位数)为193. 那么某大学的新生与其它大学的比较的假设可以是: 这15个成绩来自一个上四分位数为193的总体. 即H_0: 上四分位数为193. H_1: 上四分位数不是193. 最后结果p-值为0.035, 拒绝零假设. 即上四分位数不是193, 而是高于193.(低于193分的若是8的p-值为0.1399675, 9的p-值为0.4570977, 即可以接受零假设)

```
> z
 [1] 189 233 195 160 212 176 231 185 199 213 202 193 174 166 248
> length(z[z<=193])
[1] 7
> ((binom.test(7,15,0.75))$p.value)*2
[1] 0.03459968
> pbinom(7,15,0.75)*2
[1] 0.03459968
```

[1]实用非参数统计(第三版) Page 99 例 3.2.1

46.5 符号检验(匹配数据)

符号检验实际上是二项检验的一个特例.

例子: 要检验两种防晒膏的效果. 随机涂敷于左右手臂, 阳光下一小时.假设我们只能判定手臂红色的程度

- A 防晒膏 > B 防晒膏, 记为+1.
- A 防晒膏 < B 防晒膏, 记为-1.
- 两者一样, 记为 0

45个人被测试, 22人A手臂较好, 18人B手臂较好. 5人两个手臂同样好.

首先去掉 0 值, 因为它对两种防晒膏的好坏不提供任何信息.

如果 +1 远多于 -1, 有理由相信, B 防晒膏的效果要好于 A. 若 -1 远多于 +1, 那么 A 的效果应该好于 B 的. 若 +1 和 -1 差不多, 那么两者效果可以认定没有显著差别.

实际上, 这是二项分布的一个特例. 此处假设

$$H_0 : p = 1/2 \quad vs. \quad H_1 : p \neq 1/2$$

此处 p 为 A 好于 B 的概率.

```
> binom.test(18,40)

        Exact binomial test

data:  18 and 40
number of successes = 18, number of trials = 40, p-value = 0.6358
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
```

```
 0.2925884 0.6150932
sample estimates:
probability of success
              0.45
```

我们自己可以编一个函数. 注意函数参数不需要 alternative = c("two.sided", "less", "greater") (alternative ¡- match.arg(alternative)). prob也一定是与0.5比较, 不需要其它值.

```
# 定义函数
sign.test <- function (x, mu=0) { # does not handle NA
  n <- length(x)
  y <- sum(x<mu) # should warn about ties!
  if(y>n/2) y=n-y
  p.value <- pbinom(y,n,.5)*2
}

# 产生数据
> x <- sample(c(-1,0,1), 100, replace=T, prob=c(.4,.2,.4))
> sum(x<0)
[1] 43
> sum(x>0)
[1] 35
> sum(x==0)
[1] 22

> sign.test(x)
[1] 0.1933479
```

46.6 Cox-Stuart趋势性检验

一列数如果后面数比前面数趋于变大(上升趋势)或变小(下降趋势), 则称为有趋势的. 这个检验将后面的数和前面的数组成对, 并在对上进行符号检验. 若有趋势, 则每一对的一个数比

另外一个有变大或变小的趋势. 如果没有趋势, 实际上代表独立同分布的随机变量.

数据组织如下. $X = x_1, \cdots, x_n$以某种顺序排列, 例如观察顺序. 把X从中间分开成为两个序列A与B. 若n为奇数则去掉中间的数. 将A,B按顺序一一对应. 如果$A_i > B_i$就用"+"代替, $A_i < B_i$就用"-"代替. 然后进行符号检验.

这个检验可以用来检验任何给定非随机模式. 我们假定

1. X 互相独立
2. X 至少是有序数据
3. X是同分布或有某种趋势

检验统计量 T="+" 的个数. 零分布为p=1/2.

下面是一个例子[2]. 记录了两年的小溪水流速度. 检验平均水流速度是否降低了. 结果p-值=0.3872. 接受零假设, 即水流速度没有降低.

月份	1	2	3	4	5	6	7	8	9	10	11	
第1年	14.60	12.20	104.00	220.00	110.00	86.00	92.80	74.40	75.40	51.70	29.30	1
第2年	14.20	10.50	123.00	190.00	138.00	98.10	88.10	80.00	75.60	48.80	27.10	1

```
> x=scan()
1: 14.6 14.2 12.2 10.5 104 123 220 190 110 138 86 98.1 92.8 88.1 74.4 80 75.4 75.6 5
25:
Read 24 items
> s=matrix(a,nr=2)
> s
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
[1,] 14.6 12.2  104  220  110 86.0 92.8 74.4 75.4  51.7  29.3  16.0
[2,] 14.2 10.5  123  190  138 98.1 88.1 80.0 75.6  48.8  27.1  15.7
> a=s[1,]
```

[2]实用非参数统计(第三版) Page 122 例 3.5.3

```
> b=s[2,]
> length(a[a<b])
[1] 5
> length(a[a>b])
[1] 7

> pbinom(5,12,p=0.5) # 直接计算p-值
[1] 0.387207
> binom.test(5,12,alt="less") # 使用二项检验

        Exact binomial test

data:  5 and 12
number of successes = 5, number of trials = 12, p-value = 0.3872
alternative hypothesis: true probability of success is less than 0.5
95 percent confidence interval:
 0.0000000 0.6847622
sample estimates:
probability of success
             0.4166667
```

Cox-Stuart趋势性检验作为一个简单方法, 还可以检验两个随机变量是否有相关性. 首先将其中一个变量排序(通常是结点较少的变量). 如果有相关性, 那么另一个变量将会呈现出趋势性. 趋势相同就是正相关, 否则就是负相关.

Cochran(1937)比较了一些病人对两种药的反应, 来说明反应是否有正相关. 零假设为没有正相关性. 备择假设为有正相关性. 结果p-value = 0.03125, 且5个对全部是小于. 故拒绝零假设.

病人	1	2	3	4	5	6	7	8	9	10
药物1	0.70	-1.60	-0.20	-1.20	-0.10	3.40	3.70	0.80	0.00	2.00
药物2	1.90	0.80	1.10	0.10	-0.10	4.40	5.50	1.60	4.60	3.40

```
> x=scan()
1: .7 1.9 -1.6 0.8 -.2 1.1 -1.2 .1 -.1 -.1 3.4 4.4 3.7 5.5 .8 1.6 0 4.6 2. 3.4
```

```
21:
Read 20 items
> m=matrix(x,nr=2)
> m
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]  0.7 -1.6 -0.2 -1.2 -0.1  3.4  3.7  0.8  0.0   2.0
[2,]  1.9  0.8  1.1  0.1 -0.1  4.4  5.5  1.6  4.6   3.4
> order(m[1,])
 [1]  2  4  3  5  9  1  8 10  6  7
> m[,order(m[1,])] # 按照第一行排序
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,] -1.6 -1.2 -0.2 -0.1  0.0  0.7  0.8  2.0  3.4   3.7
[2,]  0.8  0.1  1.1 -0.1  4.6  1.9  1.6  3.4  4.4   5.5
> y=m[,order(m[1,])][2,]
> y
 [1]  0.8  0.1  1.1 -0.1  4.6  1.9  1.6  3.4  4.4  5.5
> z=matrix(y,nc=2) # 第二行配对
> z
     [,1] [,2]
[1,]  0.8  1.9
[2,]  0.1  1.6
[3,]  1.1  3.4
[4,] -0.1  4.4
[5,]  4.6  5.5
> z1=z[,1]
> z2=z[,2]
> z1
[1]  0.8  0.1  1.1 -0.1  4.6
> z2
[1] 1.9 1.6 3.4 4.4 5.5
> length(z1[z1<z2])
[1] 5
> length(z1[z1>z2])
[1] 0

> binom.test(5,5,alt="gre") # 检验趋势性

        Exact binomial test

data:  5 and 5
number of successes = 5, number of trials = 5, p-value = 0.03125
```

```
alternative hypothesis: true probability of success is greater than 0.5
95 percent confidence interval:
 0.5492803 1.0000000
sample estimates:
probability of success
```

Chapter 47

列联表

47.1 2×2列联表

Cochran 建议, 格子期望数小于5的不超过总格子数的1/5, 且没有一个格子的期望数小于1, 才可以使用卡方检验.

一般来说, 除了 2×2 列联表以外不使用Yate连续性修正. 因为经验发现这个修正不能增加对卡方分布的近似性.

47.1.1 Yate修正卡方检验

正态近似法和列联表法(修正的和非修正的卡方检验)都要求正态近似二项分布是有效的. 当不满足时, 特别是小样本时, 请使用基于超几何分布的Fisher精确检验.

Pearson's Chi-squared test: 若x为matrix至少2行或列, 则被看作2维连续table. 否则x y必须长度相等. 边际值被计算. 执行 Pearson's Chi-squared test.

若 correct = TRUE(默认), 则执行 Yate 连续性修正. 否则不执行修正.

若 simulate.p.value = TRUE 则执行 Monte Carlo 模拟来计算 p

值. B 为模拟的次数.

下面是生物统计学基础乳腺癌与初娩年龄关系的一个例子. 初娩大于30岁老年患乳腺癌的为683, 未患的1498, 初娩小于30岁老年患乳腺癌的2537, 未患的8747. 卡方检验p值为很小(2.2e-16), 说明乳腺癌与初娩年龄关系很显著.

```
# 生物统计学基础 例 10.7 乳腺癌与初娩年龄的关系
> x <- matrix(c(683,1498,2537, 8747), nr = 2)
> x
     [,1] [,2]
[1,]  683 2537
[2,] 1498 8747

> prop = function(x) x/sum(x)
> apply(x,2,prop)
         [,1]      [,2]
[1,] 0.3131591 0.2248316
[2,] 0.6868409 0.7751684

> chisq.test(x)

        Pearson's Chi-squared test with Yates' continuity correction

data:  x
X-squared = 77.8851, df = 1, p-value < 2.2e-16

> chisq.test(x, simulate.p.value = TRUE, B = 10000)

        Pearson's Chi-squared test with simulated p-value (based on 10000
        replicates)

data:  x
X-squared = 78.3698, df = NA, p-value = 1e-04
```

下面是另外一个例子[1]. 从两辆货车上随机抽样来检查次品率是否一样. 第一辆次品有13件, 非次品73件. 第二辆次品17件,

[1]实用非参数统计(第三版). Page 130. 例 4.1.1

非次品57件. Yate修正卡方检验结果显示p-值=0.286, 未修正的显示p-值=0.204. 所以接受零假设, 即次品率无显著差异.

```
> x=matrix(c(13,17,73,57),nc=2)
> x
     [,1] [,2]
[1,]   13   73
[2,]   17   57
> chisq.test(x) # Yate修正卡方检验

        Pearson's Chi-squared test with Yates' continuity correction

data:  x
X-squared = 1.1372, df = 1, p-value = 0.2863

> chisq.test(x,corr=F) # 未经过Yate修正

        Pearson's Chi-squared test

data:  x
X-squared = 1.6116, df = 1, p-value = 0.2043
```

47.1.2 Fisher精确检验

当列联表的正态近似不满足时, 我们使用超几何分布的Fisher精确检验. 它特别适合于格子内期望数少(小样本)的情况, 即其中一个格子内的期望小于5.

假设行边际固定值为N_1, N_2, 列边际固定值为M_1, M_2. 下面考察4个边际全都固定的2×2表的个数. 我们重新安排行和列使总有$M_1 <= M_2, N_1 <= N_2$. 在边际全都固定时, 4个格子的观察数实际上只有1个可以固定, 例如(1,1)可以随机变动. 其它都可以由(1,1)及边际数给出. 记X为(1,1)格子内的数, 则X的概率分布为

$$P(X) = \frac{N_1!N_2!M_1!M_2!}{N!a!(N_1-a)!(M_1-a)!(M_2-N_1+a)!}, a = 0, 1, \cdots, min(N_1, M_1)$$

此处$N = N_1 + N_2 = M_1 + M_2$. 这样的概率分布为超几何分布. 其

期望为

$$E(X) = \frac{M_1 N_1}{N}$$

方差为

$$Var(X) = \frac{M_1 M_2 N_1 N_2}{N^2(N-1)}$$

fisher.test 检验2×2列联表的优势比是否为1. 详细参考流行病部分优势比和 Mantel-Haenszel检验. epicalc 包的 cc 函数可以精确计算优势比, 有时候与 fisher.test 结果不太一样.

下面是一个例子[2]. 考察饮食中高盐与低盐是否和心血管疾病有关. 收集了两组死亡的男性, 其中一组原因是心血管疾病, 35个中有5个是高盐的. 另外一组是其它疾病, 25人中有2人是高盐的. 结果无论是双侧还是单侧, 都不显著. 即饮食与死亡原因无显著关系.

```
> x=matrix(c(2,5,23,30),nc=2)
> x
     [,1] [,2]
[1,]    2   23
[2,]    5   30
> fisher.test(x) # 双侧检验

        Fisher's Exact Test for Count Data

data:  x
p-value = 0.6882
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.04625243 3.58478157
sample estimates:
odds ratio
  0.527113
```

[2]生物统计学基础. Page 358. 例 10.20

```
> fisher.test(x,alt="less") # 单侧检验

        Fisher's Exact Test for Count Data

data:  x
p-value = 0.3747
alternative hypothesis: true odds ratio is less than 1
95 percent confidence interval:
 0.000000 2.799135
sample estimates:
odds ratio
  0.527113
```

47.1.3 联合多个表: Mantel-Haenszel检验

有时候需要将多个2×2列联表合成一个做整体分析. 当一个整体试验包括几个在不同环境中操作的小试验时, 零假设下共同的概率随环境的不同而不同, 并且每个小试验都有自己的2×2列联表这时常需要这种处理. 因为每个列联表的环境不同, 它们不能合成单一的2×2列联表.

Mantel与Haenszel(1959)提出了一个合并多个2×2列联表的方法. 又称为 Cochran-Mantel-Haenszel 卡方检验. 假设表的数目$k >= 2$. 第i个表的形式为 每个列联表的假设条件与Fisher精确检验相

	列1	列2	
行1	x_i	$r_i - x_i$	r_i
行2	$c_i - x_i$	$N_i - r_i - c_i + x_i$	$N_i - r_i$
	c_i	$N_i - c_i$	N_i

同, 并且几个列联表是由独立的试验得到的.

零假设: 在第i个列联表中, 令p_{1i}是第一行第一列中的观测的概率, p_{2i}是第二行第二列相应的概率. 对于双边检验有

$$H_0 : p_{1i} = p_{2i}, i = 1, 2, \cdots, k$$

$H_1 : p_{1i} > p_{2i}$ $p_{1i} < p_{2i}$ 对某个i成立, 但不同时成立

对于左单边检验有

$$H_0 : p_{1i} >= p_{2i}, i = 1, 2, \cdots, k$$
$$H_1 : p_{1i} <= p_{2i} \text{对所有的i,且对某个i,} \quad p_{1i} < p_{2i}$$

对于右单边检验有

$$H_0 : p_{1i} <= p_{2i}, i = 1, 2, \cdots, k$$
$$H_1 : p_{1i} >= p_{2i} \text{对所有的i,且对某个i,} \quad p_{1i} > p_{2i}$$

若行列非随机, 检验统计量

$$T = \frac{\sum x_i - \sum \frac{r_i c_i}{N_i}}{\sqrt{\sum \frac{r_i c_i (N_i - r_i)(N_i - c_i)}{N_i^2 (N_i - 1)}}}$$

若零假设为真, T的分布近似标准正态分布, 并且可以通过连续修来提高精确性, 即对于左边的概率, 可以将T的分子加0.5, 对于右边的概率, 减去0.5. 这样得到的概率在多数情况下会更精确.

若行列总和是随机的, 那么用下面的统计量更准确

$$T = \frac{\sum x_i - \sum \frac{r_i c_i}{N_i}}{\sqrt{\sum \frac{r_i c_i (N_i - r_i)(N_i - c_i)}{N_i^3}}}$$

参考流行病部分 Mantel-Haenszel检验. epicalc 包的 mhor 函数也可以计算.

下面是R自带的一个例子. 比较立即注射和1.5小时后注射盘尼西林(Penicillin)的效果, 分为治愈(Cured)和死亡(Died). 盘尼西林的水平分为5个, "1/8", "1/4", "1/2", "1", "4". 双侧检验表明立即注射比1.5小时后注射的治愈率要高. 精确检验和单边检验的结果也相同.

```
> Rabbits <-
```

```
+     array(c(0, 0, 6, 5,
+             3, 0, 3, 6,
+             6, 2, 0, 4,
+             5, 6, 1, 0,
+             2, 5, 0, 0),
+           dim = c(2, 2, 5),
+           dimnames = list(
+              Delay = c("None", "1.5h"),
+              Response = c("Cured", "Died"),
+              Penicillin.Level = c("1/8", "1/4", "1/2", "1", "4")))
> Rabbits
, , Penicillin.Level = 1/8

     Response
Delay  Cured Died
  None    0   6
  1.5h    0   5

, , Penicillin.Level = 1/4

     Response
Delay  Cured Died
  None    3   3
  1.5h    0   6

, , Penicillin.Level = 1/2

     Response
Delay  Cured Died
  None    6   0
  1.5h    2   4

, , Penicillin.Level = 1

     Response
Delay  Cured Died
  None    5   1
  1.5h    6   0

, , Penicillin.Level = 4
```

```
     Response
Delay  Cured Died
  None     2    0
  1.5h     5    0

> mantelhaen.test(Rabbits) ## Classical Mantel-Haenszel test

       Mantel-Haenszel chi-squared test with continuity correction

data:  Rabbits
Mantel-Haenszel X-squared = 3.9286, df = 1, p-value = 0.04747
alternative hypothesis: true common odds ratio is not equal to 1
95 percent confidence interval:
  1.026713 47.725133
sample estimates:
common odds ratio
                7

> mantelhaen.test(Rabbits, exact = TRUE) # 精确法  p = 0.040
> mantelhaen.test(Rabbits, exact = TRUE, alt = "greater") #单  边
检验  p = 0.020
```

47.1.4 匹配数据二项比例检验–McNemar检验

如果数据不是独立的, 即可以形成匹配数据, 则Yate修正卡方检验是不合适的.

下面是《生物统计学基础》10.4 Page 360 中的一个例子. 按年龄(或其它条件)配对621对病人, 配对的1人随机指定使用A方法治疗, 另外一人使用B方法治疗. 其中A方法生存5年以上, B方法也生存5年以上的有510对 ; A方法生存5年以上, B方法生存少于5年的有5对 ; A方法生存少于5年, B方法生存5年以上的有16对 ; A方法生存少于5年, B方法也少于5年的有90对. 检验A, B两种方法的差异是否显著.

在成对的匹配中, 结局相同的对称为一致对(concordant pair). 结局不同的称为不一致对(discordant pair). 此例中, 有510+90=600 个一致对. 有 5+16=21 个不一致对. 一致对不提供

信息, 故分析时抛弃之. 我们集中研究一致对.

不一致对中, 使用A处理后有事件发生而B处理后未发生, 称为A型不一致对. 否则称为B型不一致对.

记p=A型不一致对的概率. 如果两个处理等效, 那么A型与B型不一致对的数目应该相等. 即p=1/2. 这时, 零假设为: p=1/2. 备择假设: $p \neq 1/2$.

此例的优势比估计请参考流行病部分的匹配数据优势比估计.

我们可以使用精确的二项比例检验, 也可以使用正态近似法. 两种方法都在 McNemar 检验中.

```
# 生物统计学基础 10.4 中的例子.
> Treat<-matrix(c(510,16,5,90),nr=2,
   dimnames=list("A result"=c("more 5 years","less 5 years"),
             "B result"=c("more 5 years","less 5 years")))
> Treat
           B result
A result      more 5 years less 5 years
  more 5 years         510          5
  less 5 years          16         90

> mcnemar.test(Treat)

       McNemar's Chi-squared test with continuity correction

data:  Treat
McNemar's chi-squared = 4.7619, df = 1, p-value = 0.02910
```

R中的一个例子

```
# R的例子
> Performance <-
+     matrix(c(794, 86, 150, 570),
+           nr = 2,
```

```
+           dimnames = list("1st Survey" = c("Approve", "Disapprove"),
+                           "2nd Survey" = c("Approve", "Disapprove")))
>     Performance
          2nd Survey
1st Survey   Approve Disapprove
  Approve        794        150
  Disapprove      86        570

> mcnemar.test(Performance)

      McNemar's Chi-squared test with continuity correction

data:  Performance
McNemar's chi-squared = 16.8178, df = 1, p-value = 4.115e-05
```

47.2 R×C列联表

47.2.1 概率差异(倾向性, 趋势性)的卡方检验

共有r个总体, 从每个总体抽取随机样本, 每个样本的每个观察都可以归入c个不同的类别. 数据排列为下面的形式 每个样

	类1	类2	...	类c	总和
总体1	O11	O12	...	O1c	n1
...	...	...	...	...	...
总体r	Or1	Or2	...	Orc	nr
总和	C1	C2	...	Cc	N

本都是随机样本, 不同样本输出结果是独立的, 每个观测只能归入其中一类.

检验统计量为

$$T = \sum_{i=1}^{r}\sum_{j=1}^{c}\frac{(O_{ij}-E_{ij})^2}{E_{ij}}, E_{ij} = \frac{n_i C_j}{N}$$

T的零分布为渐近自由度为$(r-1)(c-1)$的卡方分布. 其精确值很难计算, 所以几乎不用.

假设为

$$H_0 : p_{1j} = p_{2j} = \cdots = p_{rj} for all j$$
$$H_1 : 每列至少存在两个概率不相等$$

注意, 零假设只是说概率相等. 没有必要规定概率是多少. 而且检验结果并没有告诉我们关联性的性质, 即是否有倾向性.

Wilcoxon 秩和检验实际上是有倾向性卡方检验的一个特例.

Cochran(1952)发现, 如果存在$E_{ij} < 1$或超过20%的$E_{ij} < 5$, 那么近似可能很差. 但是根据很多其它学者未发表的研究表明, 这似乎太保守了. Conover(实用非参数统计的作者)认为, 即使一些$E_{ij} < 5$, 如果r与c不太小的话, 检验也是有效的.

下面是一个例子.[3]. 列代表初娩的年龄, 分别是小于20岁, 20-24岁, 25-29, 30-34, 大于35岁. 分为疾病和对照. p-值¡0.001, 说明初娩年龄与乳腺癌是有关系的.

```
> x=matrix(c(320,1422,1206,4432,1011,2893,463,1093,220,406),
   nr=2,dimnames=list(c("疾病","对照"),
   c("小于20岁", "20-24岁", "25-29", "30-34", "大于35岁")))
> x
     小于20岁 20-24岁 25-29 30-34 大于35岁
疾病      320    1206  1011   463      220
对照     1422    4432  2893  1093      406

> chisq.test(x)

        Pearson's Chi-squared test

data:  x
X-squared =
130.172, df = 4, p-value < 2.2e-16

# 按照生物统计学基础给出的算法编写的函数
```

[3]生物统计学基础 page 374. 例 10.33 初娩与乳腺癌关系的例子

```
chisq.tendency.test<-function(x)
{
   s=1:dim(x)[2]# 组的得分变量
   n_i=apply(x,2,sum) #
   n=sum(n_i) #
   x_all = sum(x[1,])
   p=x_all/n
   q=1-p
   A=sum(x[1,]*s)-x_all*sum(s*n_i)/sum(n)
   B=p*q*(sum(s^2*n_i)-sum(s*n_i)^2/sum(n))
   res.chisq=A^2/B
   res.p.value=1-pchisq(res.chisq,df=1)
   res=list(chisq=res.chisq,p.value=res.p.value,A=A)
   res
}
> chisq.tendency.test(x)
$chisq
[1] 128.8386

$p.value
[1] 0

$A
[1] 566.8084

# coin包里面的计算精确值的函数
> library(coin)
> chisq_test(as.table(x))

        Asymptotic Pearson's Chi-Squared Test

data:  Var2 by Var1 (A, B)
chi-squared = 130.172, df = 4, p-value < 2.2e-16
```

下面是另外一个例子[4]. 检验公立中学与私立中学的某次测验成绩是否一样. 结果p-值¡0.001, 因此我们说两种学校测验成绩不同.

[4]实用非参数统计(第三版). Page 144. 例 4.2.1

```
> x=matrix(c(6,30,14,32,17,17,9,3),nr=2,
   dimnames=list(c("私立","公立"),c("0-275","276-350","351-425","426-500")))
> x
    0-275 276-350 351-425 426-500
私立     6      14      17       9
公立    30      32      17       3
> chisq.test(x)

        Pearson's Chi-squared test

data:  x
X-squared = 17.2858, df = 3, p-value = 0.0006172

Warning message:
In chisq.test(x) : Chi-squared近似算法有可能不准
```

47.2.2 独立性卡方检验

此检验与概率差异的卡方检验计算方法是一样的, 只不过对数据的解释不同.

数据为: 已知容量为N的随机样本. 观察值根据两个准则划分为几类. 按照第一个准则, 每个观察值可以归入r类(行)中的一类. 按照第二个准则, 每个观察值可以归入c类(列)中的一个.

假设为: H_0: 对任意i,j, 事件"一个观测值在行i"与事件"同样的观测在列j"是独立的. 即P(行i,列j)=P(行i)-P(列j), 对所有i,j. H_1: P(行i,列j)≠P(行i)-P(列j), 对所有i,j.

下面是一个例子[5]. 学生根据被录取的院校和是否从州内和州外毕业两个标准来分类. 零假设是每个学生被录取的院系与是否在州内和州外读高中无关. 结果p-值较大, 接受零假设.

```
> x=matrix(c(16,14,14,6,13,10,13,8),nr=2,
   dimnames=list(c("州内","州外"),c("工程学院","艺术学院", "经济学院","其它")))
```

[5]实用非参数统计(第三版). Page 147. 例 4.2.2

```
> x
    工程学院 艺术学院 经济学院 其它
州内       16       14       13   13
州外       14        6       10    8
> chisq.test(x)

        Pearson's Chi-squared test

data:  x
X-squared = 1.5242, df = 3, p-value = 0.6767
```

47.2.3 固定边缘分布的卡方检验

数据纳入r×c列联表, 与前两个不同的是行列总和固定而非随机. 此处的假设检验可以取前两个之一.

固定边际总和的卡方检验也可以检验两个随机变量X和Y是否独立.

下面是一个例子[6]. X与Y的个数的观察值(落入坐标X,Y区域内的点的个数)如下, 构成二元随机变量(X,Y). 可以看到, p-值很小, 所以X与Y不独立.

```
> x=matrix(c(0,2,4,4,1,1,4,2,0,0,3,3),nr=3,dimnames=list("X"=c(),"Y"=c()))
> x
     Y
X     [,1] [,2] [,3] [,4]
  [1,]    0    4    4    0
  [2,]    2    1    2    3
  [3,]    4    1    0    3
> chisq.test(x)

        Pearson's Chi-squared test

data:  x
X-squared = 14, df = 6, p-value = 0.02964
```

[6]实用非参数统计(第三版). Page 150. 例 4.2.3

下面是一个具体的例子[7]. 一位心理学家要求被测人学习25个单词. 给被测人25张蓝色卡片, 其中名词, 动词, 形容词, 副词, 介词各5个. 白色卡片是另外25个词, 词性及个数与蓝色一样. 允许被测人5分钟配对卡片, 5分钟学习卡片. 然后给被测人读蓝色卡片的单词, 被测人尽量提供与所读单词相关的白色卡片上的词. 心理学家关心配对的结构是否显示有某种次序, 例如与词性相关. 零假设为: 没有按照词性配对. 备择假设: 倾向于将蓝色卡片的一种词性与白色的卡片的一种词性配对(不一定相同). 结果显示p-值很小, 拒绝零假设.

```
> a=scan()
1: 0 4 0 0 1 3 1  0 0 1 0 0 0 5 0 0 0 5 0 0 2 0 0 0 3
26:
Read 25 items
> b=matrix(a,nr=5,dimnames=list(c("名词", "动词", "形容词", "副
词", "介词"),
   c("名词", "动词", "形容词", "副词", " 介词")))
> b
      名词 动词 形容词 副词 介词
名词     0    3      0    0    2
动词     4    1      0    0    0
形容词   0    0      0    5    0
副词     0    0      5    0    0
介词     1    1      0    0    3
> chisq.test(b)

        Pearson's Chi-squared test

data:  b
X-squared = 66, df = 16, p-value = 4.953e-08

Warning message:
In chisq.test(b) : Chi-squared近似算法有可能不准
```

[7]实用非参数统计(第三版). Page 151. 例 4.2.4

47.3 三向及多向列联表

以上的列联表分为行列两个方向, 也可以称为双向列联表(two-way contingency table). 若观测按照三个或以上准则分类, 那么数据可以使用三向(或多向列联表). 我们将检验统计量变换为

$$T = \sum_{ij} \frac{[O_{ij} - N\frac{R_i}{N}\frac{C_j}{N}]^2}{N\frac{R_i}{N}\frac{C_j}{N}}$$

T具有$(r-1)(c-1)$的自由度. 在三向列联表中, 有r行, c列, t块. 记块总和为B_k

$$R_i = \sum_{jk} O_{ijk}$$

$$C_j = \sum_{ik} O_{ijk}$$

$$B_k = \sum_{ij} O_{ijk}$$

期望的估计为

$$E_{ijk} = N\frac{R_i C_j B_k}{N * N * N}$$

检验统计量为

$$T = \sum_{ijk} \frac{(O_{ijk} - E_{ijk})^2}{E_{ijk}}$$

T为自由度为$rct - r - c - t + 2$的卡方分布.

对数线性模型可以成功的用来分析多向列联表.

47.4 中位数(分位数)检验

不同总体是否有相同的中位数? 中位数检验实际上是固定行列总和的卡方检验的具体应用. 因为非常重要, 所以单独讨论.

从c个总体中抽取容量为n_i的随机样本. 首先确定联合总体的中位数, 称为总中位数. 然后每个随机样本计算超过总中位数的个数为O_1i, 小于等于总中位数的个数为O_2i, 将频数排列成2×c列联表中.

零假设为: c个总体有相同的中位数. 备择假设为: 至少两个总体中位数不同.

若零假设被拒绝, 则可以对2×2列联表重复检验, 以发现哪两个总体的中位数不同. 但是要小心显著性水平α上升(下降??).

下面是一个例子[8]. 使用4种不同的方法培植玉米. 在分隔成若干块的地上随机使用1种方法. 计算每亩的产量. 为确定产量是否由种植方法不同引起, 我们采用中位数检验. 零假设为: 所有方法有相同的亩产中位数. 备择假设: 至少两种方法亩产中位数不同. 结果p-值很小, 拒绝零假设.

```
> x1=c(83,91,94,89,89,96,91,92,90)
> x2=c(91,90,81,83,84,83,88,91,89,84)
> x3=c(101,100,91,93,96,95,94)
> x4=c(78,82,81,77,79,81,80,81)
> my_aaa<-function(x,med){
+ a=length(x[x<=med])
+ b=length(x[x>med])
+ res=c(a,b)
+ res}
> y=matrix(c(my_aaa(x1,89),my_aaa(x2,89),my_aaa(x3,89),my_aaa(x4,89)),
   nr=2,dimnames=list(c("<=89",">89"),c("x1","x2","x3","x4")))
> y
    x1 x2 x3 x4
<=89  3  7  0  8
```

[8]实用非参数统计(第三版). Page 158. 例 4.3.1

```
>89   6  3  7  0
> chisq.test(y)

        Pearson's Chi-squared test

data:  y
X-squared = 17.5431, df = 3, p-value = 0.0005464
```

如果检验的是上分位数或下分位数, 那么把列联表的计数更换为相应的分位数即可.

47.5 关联性(相依性)度量

相依性度量很大程度上取决于个人的决定. 一般会依据传统的习惯, 而不是统计学的考虑.

47.5.1 Cramer关联系数

由Cramer(1946)提出, 使用T除以可能达到的最大值(极端不平衡列联表中达到最大, 为N(min(r,c)-1)). 计算公式为

$$R = \sqrt{\frac{T}{N(min(r,c)-1)}}$$

其中T为卡方检验统计量, N为观测总数. 若强行列相关, 则R接近1.

下面是计算公立,私立学校考试成绩的例子.

```
coef.cramer<-function(x){
   r=sqrt(chisq.test(x)$statistic/(sum(x)*(min(dim(x))-1)))
   names(r)<-"Cramer Coefficient"
```

```
   r
}
> x=matrix(c(6,30,14,32,17,17, 9,3),nr=2)
# 行为公立, 私立, 列为90-100,80-90,70-80,60-70分数段的学生数
> x
     [,1] [,2] [,3] [,4]
[1,]    6   14   17    9
[2,]   30   32   17    3
> coef.cramer(x)
Cramer Coefficient
       0.3674853
# x10倍后检验统计量增加10倍, 但是cramer系数不变
> chisq.test(x*10)

       Pearson's Chi-squared test

data:  x * 10
X-squared = 172.8581, df = 3, p-value < 2.2e-16

> coef.cramer(x*10)
Cramer Coefficient
       0.3674853
```

47.5.2 Pearson关联系数

均方关联系数 Pearson's coefficient of mean square contingency, 由Yule和Kendall(1950)给出. 文献[14]也称为列联系数(contingency coefficient). 定义为

$$R=\sqrt{\frac{T}{N+T}}$$

记q=min(r,c), 因为T的最大值为N(q-1), 故R的最大值为

$$R_{max}=\sqrt{\frac{N(q-1)}{N+N(q-1)}}=\sqrt{\frac{q-1}{q}}<1.0$$

```
coef.pearson<-function(x){
   s<-chisq.test(x)$statistic
   r<-sqrt(s/(sum(x)+s))
   names(r)<-"contingency coefficient"
   r}
> coef.pearson(x)
contingency coefficient
           0.3449319
> coef.pearson(x*10)
contingency coefficient
           0.3449319
```

47.5.3 Pearson均方关联系数

此系数也具有Pearson关联系数的特点, 被Yule和Kendall(1950)称为mean square contingency coefficient. 定义为

$$R = T/N$$

我们有$0 <= R <= q - 1$

```
> r=chisq.test(x)$statistic/sum(x)
> names(r)<-"mean square contingency coefficient"
> r
mean square contingency coefficient
                     0.1350454
```

47.5.4 TschuProw系数

定义

$$R = \sqrt{\frac{T}{N\sqrt{(r-1)(c-1)}}}$$

47.5.5 正关联和负关联

2×2列联表有时候区分正关联和负关联是有意义的. 例如根据父亲和母亲的头发颜色将40个孩子分类[19](Page 167 例 4.4.5)[9].

```
> x<-matrix(c(28,5,0,7),nr=2,
   dimnames=list("母亲"=c("黑色","金色"),"父亲"=c("黑色","金色")))
> x
     父亲
母亲 黑色 金色
  黑色   28    0
  金色    5    7
```

Phi系数(phi coefficient)就是这样的系数, 定义为

$$R = \frac{ad - bc}{\sqrt{r_1 r_2 c_1 c_2}}$$

其中$r_1 r_2 c_1 c_2$为行列的和. abcd分别为四个格子的观测数. 当$ad - bc > 0$为正关联, $ad - bc < 0$为负关联. 下面计算头发颜色例子的Phi系数.

```
> r<-(x[1,1]*x[2,2]-x[1,2]*x[2,1])/sqrt(prod(colSums(x),rowSums(x))) # prod 为乘法函数
> r
[1] 0.7035265
```

其它2×2列联表关联性系数还有Yule和Kendall(1950)提出的

$$R = \frac{ad - bc}{ad + bc}$$

[9]作者注: 此例似乎应该再根据孩子的头发颜色分为几个2×2列联表

Ives和Gibbons(1967)提出的

$$R = \frac{(a+d)-(b+c)}{a+b+c+d}$$

可定义的关联性度量方法有很多, 选择什么取决于个人喜好.

47.5.6 kappa统计量-重复性度量

在两个类型变量彼此不做预测时, 这个指标很有用处([14], page 386). 它可以表示两个类型数据关联性大小. 特别在可靠性研究(reliability study)中, 人们希望定量表示出对相同变量做多次测量时, 它的重复性有多大.

假设有n个受试者, 都接受了两次关于同一问题的调查, 则kappa统计量常用于测度两次调查的可重复性. 公式为

$$k = \frac{p_o - p_e}{1 - p_e}$$

其中p_o为两次调查中一致性概率. $p_e = \sum_{i=1}^{c} a_i b_i$为零假设下(两次调查彼此独立, 即无重复性)两次调查的期望一致的概率, 此处$a_i b_i$为c×c列联表中两个调查第i个类型的边际概率.

零假设: 两次调查彼此独立, 即无重复性. 备择假设: 两次调查有一定的重复性.

Landis 及 Koch (1977) 提出下面的参考标准

`k>0.75` 表示极好的重复性
`0.4<=k<=0.75` 好的重复性
`0<=k<0.4` 边界(勉强够格)的重复性

Fleiss 提供了kappa统计量的进一步信息, 包括多于两次调查时如何判断重复性.

kappa 值也常常用做相同变量重复估计之间是否有重复性的一种测度.

如果我们对两个变态变量上反应的一致性有兴趣, 而其中一个变量的反应可以作为金标准, 则灵敏度及特异度是比 kappa 统计量更好的指标.

下面是函数及一个例子. 数据 x 为第一次调查及第二次调查牛肉消费的结果, 分为每周消费1次以下和多于1次. 最后看看两次调查的重复性如何. 结果p-值很小, 拒绝零假设, 两次调查有重复性, 重复性大小为 0.378.

```
kappa.test <- function(x)
{
   N=sum(x)
   Po=sum(diag(x)/N) # 观察到的一致数
   mr=apply(x,1,sum) # 行边际
   mr=mr/N
   mc=apply(x,2,sum) # 列边际
   mc=mc/N
   Pe=sum(mc * mr) # 期望一致数
   k=(Po-Pe)/(1-Pe) # kappa统计量
   se_k = sqrt((Pe+Pe^2-sum(mr*mc*(mr+mc)))/(N*(1-Pe)^2)) # kappa统
计量的标准误
   z=k/se_k # 检验统计量
   p.value=1-pnorm(z) # p 值
   res=list(kappa=k,se_k=se_k,p.value=p.value,z=z)
}
```

```
> x=matrix(c(136,69,92,240),nr=2, dimnames=list("1st survey"=c("<= 1 time/week","
> x
             2st survey
1st survey      <= 1 time/week > 1 time/week
  <= 1 time/week          136           92
  > 1 time/week            69          240

> k=kappa.test(x)
> k
$kappa
[1] 0.3781906
```

```
$se_k
[1] 0.04298259

$p.value
[1] 0

$z
[1] 8.798692

> library(epiR)
> epi.kappa(a=136,c=69,b=92,d=240)
$kappa
       est    lower    upper
1 0.3781906 0.2978196 0.4585616

$mcnemar
  test.statistic df    p.value
1      3.285714  1 0.06988521
```

47.5.7 相关性的检验

如何使用相关系数R作为检验统计量检验

H_0: 不存在相关(正相关或负相关), H_1: 存在相关(正相关或负相关)

我们可以看到当卡方检验统计量T大时, R也比较大. 我们可以使用R表示T, 然后用T作为检验统计量. 当T显著时, R也显著. 正负可以看$ad - bc$的值.

47.6 卡方拟合优度检验

拟合优度检验: 若x是matrix只有一行或一列, 或x是vector, 且y没

有给出, 那么会执行"goodness-of-fit test"(拟合优度检验), x被认为是一维列联表. 检验x概率是否与p相等, 若p未给出, 则检验是否概率都一样, 即均匀分布.

Cochran(1952)建议观测期望E_i不小于1, 且不超过20%的不小于5. 最近的研究表明这个限制可以放宽. [19](page 173, 其它修正方法)

(二项比例关联性检验(Pearson's Chi-squared test): 若x为matrix至少2行或列, 则被看作2维连续table. 否则x y必须长度相等. 边际值被计算. 执行 Pearson's Chi-squared test. 此检验为二项比例关联性检验)

下面是一个正态拟合的例子

```
> n=1000
> x=rnorm(n)
> b=seq(-2,2,0.2)

# 计算正态分布的理论概率
> p=c(pnorm(b[1]),diff(pnorm(b)),1-pnorm(b[length(b)]))
> p
 [1] 0.02275013 0.01318019 0.01886897 0.02595737 0.03431301 0.04358558
 [7] 0.05320014 0.06239772 0.07032514 0.07616203 0.07925971 0.07925971
[13] 0.07616203 0.07032514 0.06239772 0.05320014 0.04358558 0.03431301
[19] 0.02595737 0.01886897 0.01318019 0.02275013
> sum(p)
[1] 1

# 实际频数
> bre=c(-1000,b,1000)
> h=hist(x, breaks=bre)
> h$counts # 实际频数
 [1] 18  8 26 23 31 48 53 71 72 84 79 62 61 73 64 49 50 37 24 19 21 27
> sum(h$counts)
[1] 1000

# 卡方检验. 默认执行连续性修正. p值>0.05则两个频率差异不显著
> chisq.test(p*1000,h$counts)
```

```
        Pearson's Chi-squared test

data:  p * 1000 and h$counts
X-squared = 440, df = 420, p-value = 0.2412

Warning message:
Chi-squared近似算法有可能不准 in: chisq.test(p * 1000, h$counts)
```

47.7 相关观测的Cochran检验

普通的处理是所有样本分为c组, 每组使用一个处理方法, 得到一个2×c列联表. 但是, 为了提高功效, 我们有时候需要对每个样本都用c种方法独立的处理. 在这里我们使用r为样本个数或区组数(区别于通常的n). 我们得到了一个r×c列联表, 其中每个观测值为0或1. 行总和为R_i, 列总和为C_i.

假设样本是随机的(即随机选取的). 处理的结果可以按照某种方式分为两种, 记为0和1.

检验统计量为

$$T = c(c-1)\frac{\sum_{j=1}^{c}(C_j - \frac{N}{c})^2}{\sum_{i=1}^{r} R_i(c-R_i)}$$

下式计算更适合

$$T = \frac{c(c-1)\sum_{j=1}^{c} C_j^2 - (c-1)N^2}{cN - \sum_{i=1}^{r} R_i^2}$$

T的精确分布难以求得, 大样本(r比较大)逼近后近似自由度c-1的卡方分布. 零假设为: 所有处理效果相同. 备择假设为: 处理之间效果有差异. 记$p_j = P$为列j中出现1的概率, 则零假设可以描述为: 每个处理中有$p_1 = p_2 = \cdots = p_c$. 备择假设为: 某两个处理i,j有$p_i \neq p_j$.

若拒绝了零假设, 可以使用McNemar对c个处理进行两两比较.

若仅考虑c=2(两种处理), 那么Cochran检验与McNemar检验是一样的.

下面是一个例子([19], page 181, 例 4.6.1). 3个篮球爱好者对12场比赛进行预测. 比赛是从所有比赛中随机选取的. 预测准确记为1, 否则记为0. 零假设为: 3个爱好者的预测是等有效的. 备择假设为: 其中至少2个爱好者的预测不是等有效的. 数据及结果见下面. p-值很小, 拒绝零假设.

```
cochran.test<-function(x){
   c=dim(x)[2]
   C=colSums(x)
   R=rowSums(x)
   N=sum(x)
   T=c*(c-1)*sum((C-N/c)^2)/sum(R*(c-R))
   p=1-pchisq(T,df=c-1)
   res<-list(statistic=T,p.value=p, df=c-1)
   res
}
> x<-matrix(rbinom(36,size=1,p=0.7),nc=3)
> colSums(x)
[1]  3 10  7
> x
      [,1] [,2] [,3]
 [1,]    0    1    0
 [2,]    1    1    1
 [3,]    0    1    0
 [4,]    1    1    1
 [5,]    1    1    1
 [6,]    0    1    1
 [7,]    0    1    1
 [8,]    0    1    0
 [9,]    0    1    1
[10,]    0    0    0
[11,]    0    0    1
[12,]    0    1    0
> cochran.test(x)
$statistic
[1] 9.25
```

```
$p.value
[1] 0.009803655

$df
[1] 2
```

47.8 其它分析方法

47.8.1 似然比统计量

$$T = \frac{\sum(O_i - E)^2}{E}$$

这个统计量是Pearson(1900,1922)引入的, 称为Pearson卡方统计量. 以上使用的分析方法都是这种. 下面是一种不同的方法, 称为似然比检验法, 统计量为

$$T = 2\sum O_i ln(\frac{O_i}{E_i})$$

它来自于统计学里的似然比理论, 与Pearson统计量服从同样自由度的卡方分布, 属于Wilks(1935,1938),收到广泛运用. 但是它的一个弊端是, 如果$N/rc < 5$, 卡方分布的效果就不好. Agresti(1990)说明如果$N/rc < 1$, Pearson方法的近似也不好.

47.8.2 对数线性模型

这种方法可以很好的分析三维以上的列联表. 也可以在线性对数模型中使用 Pearson 统计量或似然比统计量, 不同的是估计E的方法是迭代法.

在双向列联表中, 零假设可以描述为: 对所有ij, $p_{ij} = p_i * p_j$. 两边取对数, 零假设变为: 对所有ij, $logp_{ij} = logp_i + logp_j$. 对零假

设的检验变为检验格子概率的对数是否是边际概率对数的线性函数.

Chapter 48

秩检验

48.1 Wilcoxon符号-秩检验(匹配数据)

Wilcoxon符号-秩检验类似于配对的t检验. 是基于二项分布的检验. 在这里, 我们不关心具体打分的大小, 但是打分的秩次(相对大小)是有意义的. ([14] Page323 例 9.12. 计算方法及公式见参考文献)

函数 wilcox.test(), 当参数只有x 或 x,y都给出且paired =TRUE(匹配样本), 为符号秩检验. 比较x,y的均值. 零假设为 x 或 x-y的分布关于 mu 对称. 默认小于50样本量, 或指定exact=T, 计算精确p值, 否则使用正态近似计算p值. (注: 当xy都给出且paired = FALSE(独立样本) 为秩和检验. 比较x,y的均值. 零假设为xy之差为mu. 见下面秩和检验)

下面是一个例子. 若防晒膏问题中晒红程度打分左手臂打分为x, 右手臂打分为y 则这种情况下适合使用Wilcoxon符号-秩检验. 检验的是变量是否关于 mu 对称. 即为中位数的检验. 设差值d=x-y.

```
# 打分的差值
> d=c(-8,rep(-7,3),-6,-6,-5,-5,-4,rep(-3,5),rep(-2,4),
   rep(-1,4),rep(3,2),rep(2,6),rep(1,10))
> d
```

```
 [1] -8 -7 -7 -7 -6 -6 -5 -5 -4 -3 -3 -3 -3 -3 -2 -2 -2 -2 -1 -1 -1 -1  3  3  2
[26]  2  2  2  2  2  1  1  1  1  1  1  1  1  1  1  1

# 一个参数时候是符号秩(signed rank)检验.
# 两个参数, 且 paired=TRUE 的时候也是符号秩(signed rank)检
验.
# 否则是秩和检验
# 默认检测 mu=0, 或 delta=0 (差值=0)
> wilcox.test(d)

        Wilcoxon signed rank test with continuity correction

data:  d
V = 248, p-value = 0.02869
alternative hypothesis: true location is not equal to 0

Warning message:
In wilcox.test.default(d) : 无法精 算带连结的p值

# 中位数不变, 结果也不变
> wilcox.test(d*10)

        Wilcoxon signed rank test with continuity correction

data:  d * 10
V = 248, p-value = 0.02869
alternative hypothesis: true location is not equal to 0

Warning message:
In wilcox.test.default(d * 10) : 无法精 算带连结的p值

# 注意, mu值对结果有显著的影响
> wilcox.test(d+10)

        Wilcoxon signed rank test with continuity correction

data:  d + 10
V = 820, p-value = 3.397e-08
alternative hypothesis: true location is not equal to 0

Warning message:
```

In wilcox.test.default(d + 10) : 无法精 算带连结的p值

48.2 Mann-Whitney U检验(非匹配数据, 即Wilcoxon秩和检验)和Hodges-Lehmann估计

Wilcoxon秩和检验在某些文献上也叫做 Mann-Whitney U检验. 此处介绍的是类似于两个独立样本t检验的非参数检验. 前面提出的Wilcoxon符号-秩检验类似于配对的t检验.

Mann-Whitney U检验是建立在匹配观察值(x_i, y_i)上, 比如$x_i < y_i$的数目上. 另外, 如果有$x_i = y_i$, 则对(x_i, y_i)的检验统计量加上0.5.

Mann-Whitney U检验与 Wilcoxon 秩和检验完全等价. 其计算出的p值相同. ([14], Page 328. [19] Page 195. 参考文献有详细描述和其它方法的比较).

函数 wilcox.test 执行单样本(y=NULL)和两样本(x,y)符号秩检验与秩和检验. 后者即Mann-Whitney U test 或 Wilcoxon test.

当只有x 或 x, y都给出且paired =TRUE(匹配样本), 为符号秩检验. 比较x,y的均值. 零假设为 x 或 x, y对的分布关于 mu 对称.

当xy都给出且paired = FALSE(独立样本) 为秩和检验. 比较x,y的均值. 零假设为xy之差为mu.

默认小于50样本量, 或指定exact=T, 计算精确p值, 否则使用正态近似计算p值.

Mann-Whitney 检验类似于两独立样本的t检验. (大数据量时可以使用t.test. kruskal.test 用于两或多样本的检验).

参数 conf.int=TRUE: Hodges-Lehmann 估计位移的置信区间: Mann-Whitney检验不能检验出差值(位移)的置信区间, 若想知道区间, 一个方法是使用Mann-Whitney检验多次变换不同的

差值检验. 然而, Hodges-Lehmann 估计可以实现这个功能. 幸好wilcox.test有这个功能. 只要使参数conf.int=TRUE 即可. 当可以计算精确p-值时, 使用Bauer (1972)的算法, 和 Hodges-Lehmann estimator. 若无精确p-值, 则使用正态逼近.

wilcox.exact in 'exactRankTests' 可以在有结的情况下计算精确的p值. coin 包里的函数 wilcox_test, 可以计算精确的p值.

假设数据来自两个随机样本, $x_1, x_2, \cdots, x_n$来自样本1, $y_1, y_2, \cdots, y_m$来自样本2. 给这N=m+n个观测从小到大排序并赋秩$(1, 2, \cdots, N)$. 若样本的分布一样, 那么其秩和相等. 检验就基于此原理.

令$F(x)$为X的分布函数, $G(x)$为Y的分布函数, 零假设为: $F(x) = G(x)$. 备择假设为: $F(x) \neq G(x)$. 实际上备择假设可以转换为$P(x > y) \neq P(x < y)$, 这样有利于计算.

此函数会自动把数值转换为秩次(符号-秩检验的时候并不是这样), 故不需要自己计算秩次, 只需提供原始数据(及分组信息, 见例子).

下面是一个例子[1]. 选择12名三年级的同学, 其中4名上过幼儿园的考试成绩排序分别为2,5,6,9. 要检验的零假设是: 三年级学生的学习表现不取决于是否上过幼儿园. 备择假设: 学习表现与是否上过幼儿园不独立.

模型假设12个孩子是三年级学生的一组随机样本, 并根据学习成绩从好到差排序标记. "不独立"是指上过幼儿园的整体表现比没有上过的好, 或不好. 那么假设可以重新描述为, 零假设: 上过幼儿园的4个孩子的秩是秩1-12的一个随机样本. 备择假设: 上过幼儿园的4个孩子的秩整体比12个孩子中随机抽取4个孩子的秩要大或小.

我们选择检验统计量T是上过幼儿园的秩的和. 若T很大或很小, 则拒绝零假设. 故该检验是双边的. 每一个可能的结果是从1-12中抽取4个数, 对应上过幼儿园的4个孩子的秩. 样本空间是$\binom{12}{4} = 495$. 下面是我们的计算. 最后结果是没有显著差异.

```
# x可以是任何的12个不同的排序过的数字. 例
```

[1]实用非参数统计(第三版). Page 71. 例 2.3.2

```
如 x=3*(1:12), sort(rnorm(12)) 均可
> x=1:12
> y=rep(0,12)
> y[c(2,5,6,9)]=1 # y是分组
> y
 [1] 0 1 0 0 1 1 0 0 1 0 0 0
> wilcox.test(x~factor(y))

        Wilcoxon rank sum test

data:  x by factor(y)
W = 20, p-value = 0.5697
alternative hypothesis: true location shift is not equal to 0

# 也可以按照下面的方法组织数据
> x=1:12
> y=c(2,5,6,9)
> x=x[-y]
> x
[1]  1  3  4  7  8 10 11 12
> group=c(rep(0,8),rep(1,4))
> wilcox.test(c(x,y)~group)

        Wilcoxon rank sum test

data:  c(x, y) by group
W = 20, p-value = 0.5697
alternative hypothesis: true location shift is not equal to 0

# 手工计算
> choose(12,4) # 495种组合
[1] 495
> c=combn(12,4) # 从1-12选择4个数字的所有组合
> a=colSums(c) # 所有组合的和(秩和)
> length(a[a<=22]) # 所有和小于22的组合个数. 22=sum(2,5,6,9)
[1] 141
> 2*length(a[a<=22])/495 # 所有和小于22的双侧频率
[1] 0.569697

# 考试名次为1,2,3,4时的检验
> y1=rep(0,12)
```

```
> y1[1:4]=c(1,1,1,1)
> wilcox.test(x~factor(y1),conf.inf=T)

        Wilcoxon rank sum test

data:  x by factor(y1)
W = 32, p-value = 0.00404
alternative hypothesis: true location shift is not equal to 0

# 手工计算
> 1/495*2
[1] 0.004040404

# 计算差值(位移)的置信区间
> wilcox.test(x~factor(y),conf.int=T)

        Wilcoxon rank sum test

data:  x by factor(y)
W = 20, p-value = 0.5697
alternative hypothesis: true location shift is not equal to 0
95 percent confidence interval:
 -4  6
sample estimates:
difference in location
                     2
```

另外一个例子 ([14], 生物统计学基础. Page 328. 例 9.15). 我们要比较10-19岁不同遗传形式(RP)的视敏度. 设25人显性病, 30人有伴性病. 这些人好的眼睛的最好修正视敏度见下表.

视敏度	显性	伴性
20-20	5	1
20-25	9	5
20-30	6	4
20-40	3	4
20-50	2	8
20-60	0	5
20-70	0	2
20-80	0	1

```
> x=c(rep(2,5),rep(2.5,9),rep(3,6),rep(4,3),rep(5,2))
> x
 [1] 2.0 2.0 2.0 2.0 2.0 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 3.0 3.0 3.0 3.0 3.0
[20] 3.0 4.0 4.0 4.0 5.0 5.0
> y=c(rep(2,1),rep(2.5,5),rep(3,4),rep(4,4),rep(5,8),rep(6,5),rep(7,2),1)
> y
 [1] 2.0 2.5 2.5 2.5 2.5 2.5 3.0 3.0 3.0 3.0 4.0 4.0 4.0 4.0 5.0 5.0 5.0 5.0 5.0
[20] 5.0 5.0 5.0 6.0 6.0 6.0 6.0 6.0 7.0 7.0 1.0
> g=c(rep(0,25),rep(1,30))

# 实际上, 只要x与y的相对位置不变, 最后结果就不变.
# 例如 wilcox.test(c(x*10,y*10)~g) 结果是一样的.
> wilcox.test(c(x,y)~g)

        Wilcoxon rank sum test with continuity correction

data:  c(x, y) by g
W = 179, p-value = 0.0007813
alternative hypothesis: true location shift is not equal to 0

Warning message:
In wilcox.test.default(x = c(2, 2, 2, 2, 2, 2.5, 2.5, 2.5, 2.5,  :
  无法精 算带连结的p值

# 使 用 coin 的 wilcox_test 函 数 计 算 精 确p值. minitab的 结 果
是0.0002
# 注 意: corr = T 和 F 结果是一样的.
> library(coin)
> wilcox_test(c(x,y)~factor(g))
```

```
        Asymptotic Wilcoxon Mann-Whitney Rank Sum Test

data:  z by factor(g) (0, 1)
Z = -3.7975, p-value = 0.0001461
alternative hypothesis: true mu is not equal to 0
```

48.3 多组数据秩检验–Kurskal-Wallis 检验

Kurskal-Wallis 检验是Wilcoxon方法在多于两个样本的时候的推广

普通参数方法称为"单因素方差分析", 或有时候称为单因素F检验. 违反正态假设可能对F有一些影响, 但是某些非正态分布的数据(例如有极值)F检验的功效会比Kurskal-Wallis 检验小很多. 相对于F检验, Kurskal-Wallis 检验的A.R.E.从来不会小于0.864, 若是正态分布, A.R.E.=0.955. 均匀分布=1.0, 双指数分布=1.5.

Kurskal-Wallis 检验是多个独立样本的检验([19] Page 207). 对于有很多结的情况, 应当毫不犹豫的使用 Kurskal-Wallis 检验. 事实上 Kurskal-Wallis 检验是用于列联表的一个非常好的检验. 对差异很敏感. Kurskal-Wallis 检验统计量合理的运用了卡方逼近.

对于两样本的情况, Kurskal-Wallis 检验和 Mann-Whitney检验是等价的.

下面是一个例子([19] Page 209). 检验4个品种的玉米的产量是否不同. p-值很小, 说明不同. 然后可以使用两两比较(Mann-Whitney检验或Kurskal-Wallis 检验都可以)来检验哪两个品种不同.

```
> x1=scan()
1: 83 91 94 89 89 96 91 92 90
10:
Read 9 items
> x2=scan()
1: 91 90 81 83 84 83 88 91 89 84
```

```
11:
Read 10 items
> x3=scan()
1: 101 100 91 93 96 95 94
8:
Read 7 items
> x4=scan()
1: 78 82 81 77 79 81 80 81
9:
Read 8 items
> x=c(x1,x2,x3,x4)
> x
 [1]  83  91  94  89  89  96  91  92  90  91  90  81  83  84  83  88  91  89  84
[20] 101 100  91  93  96  95  94  78  82  81  77  79  81  80  81
> g=c(rep(1,9),rep(2,10),rep(3,7),rep(4,8))
> kruskal.test(x,g)

        Kruskal-Wallis rank sum test

data:  x and g
Kruskal-Wallis chi-squared = 25.6288, df = 3, p-value = 1.141e-05
```

48.4 方差齐性检验

两样本的方差齐性检验使用F检验. 多于两样本则使用bartlett.test. 2个非正态样本参考ansari.test或mood.test, 它们是非参数检验. 多于2个非正态样本参考fligner.test

TODO: 算法参考[19] Page 217. 方差检验的方法很多. Conover, Johnson, Johnson (1981)对56个方差检验方法作了全面比较.

48.5 秩相关度量

Kruskal(1958)的一篇综述讨论了很多相关度量. 若x,y独立, 则一些相关度量有分布函数, 且不依赖于(x,y)的二维分布函

数([19] Page 227).

函数cor.test可以使用三种方法,只要指定参数method = c("pearson", "kendall", "spearman")其中的一种即可.它会自动将数据转换为秩并自动对结校正.

48.5.1 Pearson关联系数

最常用的就是Pearson乘积矩关联系数([19] Page 226).

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{[\sum(x_i - \bar{x})^2(y_i - \bar{y})^2]^{1/2}}$$

48.5.2 Spearmanρ

Spearman(1904)给出了相关性度量.计算方法如下([19] Page 227):设数据为二维随机变量$(x_1, y_1), \cdots, (x_n, y_n)$.分别对x,y排序,分别取得它们的秩为$R(x_i), R(y_i)$.即若$x_i$为最小的x值, $R(x_i) = 1$, x_i为次小的x值, $R(x_i) = 2$.有结时,赋予没有结时本应秩的平均值.Spearman相关系数为:

$$\rho = \frac{\sum_{i=1}^n R(x_i)R(y_i) - n(\frac{n+1}{2})^2}{(\sum_{i=1}^n R(x_i)^2 - n(\frac{n+1}{2})^2)^{1/2}(\sum_{i=1}^n R(y_i)^2 - n(\frac{n+1}{2})^2)^{1/2}}$$

实际上是基于秩与平均秩的简单Pearson乘积矩关联系数.若数据用秩代替,则

$$\bar{R(x)} = \frac{n+1}{2}$$

对y也是一样的.

48.5.3 Kendall τ

Kendall([19] Page 230)(1938)提出的.设数据为二维随机变量$(x_1, y_1), \cdots, (x_n, y_n)$.若一个观测的两个元素比另外一个观

测的两个元素都大, 或都小, 称为是协调的(concordant), 例如(1.3,2.2)和(1.6,2.7)是协调的. 若一个观测的两个元素比另外一个观测的两个元素大小相反, 称为不协调的(discordant), 例如(1.3,2.2)和(1.6,1.1). 记N_c为协调的观测对数, N_d为不协调的观测对数. 由于n个观测可能有$\binom{n}{2} = n(n-1)/2$种不同方式的配对, N_c, N_d与带结的对数之和将等于$\binom{n}{2} = n(n-1)/2$. Kendall提出没有结的相关性度量为

$$\tau = \frac{N_c - N_d}{n(n-1)/2}$$

若所有对是协调的, $\tau = 1$, 若所有对是不协调的, $\tau = -1$. 如果有结, 修正为

$$\tau = \frac{N_c - N_d}{N_c + N_d}$$

48.5.4 Daniels趋势性检验

Daniels(1950)提出用Spearman ρ 作为趋势性检验([19] Page 234).

48.5.5 Jonckheere-Terpstra 检验

Spearman ρ 或 Kendall τ可以用于几个独立样本的零假设: 所有样本来自同一分布, 即

$$H_0 : F_1(x) = \cdots = F_k(x)$$

备择假设: 分布是在有序的某个方向上

$$H_1 : F_1(x) >= F_2(x) >= \cdots >= F_k(x)$$

至少有一个不等式成立.

注意: 此数据集与 Kruskal-Wallis 检验相同. 但是 Kruskal-Wallis 检验对任何差异敏感. 而 Spearman ρ 或 Kendall τ仅对H_1中的特殊有序敏感.

48.5.6 TODO: Kendall偏相关系数

48.5.7 几个例子

直接使用 cor(), 默认方法为 pearson 方法, 适用于连续数据.

下面是一个强相关的例子

```
> n <- 100
> x <- runif(n)
> b <- rep(NA,n)
> b[1] <- 0
> for (i in 2:n) {
+   b[i] <- b[i-1] + .1*rnorm(1)
+ }
> y <- 1-2*x+b[1:n]
> plot(x,y) # 绘图查看
> cor(x,y)
[1] -0.8217834
> cor.test(x,y) # p 值很小, 说明相关系数显著不等于 0

        Pearson's product-moment correlation

data:  x and y
t = -14.2774, df = 98, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.8766919 -0.7457370
sample estimates:
      cor
-0.8217834
```

spearman相关: 如果配对的数据不是连续的或不满足正态分布, 则可以视数据为秩次值. 由

$$S_x^2 = S_y^2 = n(n+1)/12$$

$$S_{xy} = n(n+1)/12 - 6\sum d^2/12(n-1)$$

其中n为观测样本量. 可以得到

$$r_s = 1 - \frac{6\sum d^2}{n(n^2-1)}$$

对r的显著性检验, 当$n > 10$时, 可以应用t检验. 若p值很小, 则r值显著不为0, 即x, y显著相关. 其中

```
t = r \sqrt{n-2} / \sqrt{1-r^2}  (df = n-2)
```

下面是一个例子. 两个医师对10张片子做评价, 打分结果(x,y)为病情的轻重. 判断两个医师的评价是否一致.

```
> x=c(3,1,5,6,2,4,8,7,9,10)
> y=c(3,2,6,10,1,5,9,7,4,8)
> cor(x,y)
[1] 0.6969697
> cor(x,y,method="spearman")
[1] 0.6969697
```

下面是自己编写的函数, 与cor结果一样.

```
> d=y-x
> d
 [1]  0  1  1  4 -1  1  1  0 -5 -2
> sum(d^2)
[1] 50
> r=1-6*sum(d^2)/(10*(10^2-1))
> r
[1] 0.6969697
> t=r*sqrt(8)/sqrt(1-r^2)
> t
[1] 2.749026
> pt(t,df=8)
```

```
[1] 0.9874517
> 1-pt(t,df=8) # p 值, 说明 r值显著不等于0. 即x y评价显著相关
[1] 0.01254834
```

48.6 多个相关样本

Milton Friedman 检验是符号检验的推广([19] Page 268). Quade 检验是符号秩检验的推广. Friedman检验使用更加广泛, 使用假定更少, 但是只有3个处理时, 功效不如符号秩检验, 4,5个处理时, 功效与Quade检验相当, 6个以上时, 功效比较大.

48.6.1 Friedman 检验

试验通常为随机化的完全区组设计. 对应的参数方法叫做双因素方差分析. 秩方法依赖于每组观测的秩, 其方法的发明者是一个著名的经济学家: Milton Friedman.

函数 friedman.test 按照统计量T_1计算([19] Page 270). T_1逼近自由度为k-1的卡方分布. 修正统计量与T_1的关系为

$$T_2 = \frac{(b-1)T_1}{b(k-1)-T_1} \sim F[k-1,(b-1)(k-1)]$$

其中b为区组数, k为处理数. 最近的研究表明, T_2有更好的逼近分布.

TODO: 若拒绝零假设, 多重比较见[19] Page 270.

下面是一个例子. 随机的12名业主在自己的院子的等面积的土地上分别种植4种不同的草, 一段时间后按喜好程度排名, 最喜欢的为4, 最不喜欢的为1. 最后想看看是否哪种草更加受欢迎. 我们最后给出修正的统计量. 两个统计量的p-值差不多.

```
x<-matrix(c(4,3,2,1,
```

```
   4,2,3,1,
   3,1.5,1.5,4,
   3,1,2,4,
   4,2,1,3,
   2,2,2,4,
   1,3,2,4,
   2,4,1,3,
   3.5,1,2,3.5,
   4,1,3,2,
   4,2,3,1,
   3.5,1,2,3.5),
   nc=4,byrow=T,dimnames=list(1:12,c("a","b","c","d")))
> x
    a   b   c   d
1  4.0 3.0 2.0 1.0
2  4.0 2.0 3.0 1.0
3  3.0 1.5 1.5 4.0
4  3.0 1.0 2.0 4.0
5  4.0 2.0 1.0 3.0
6  2.0 2.0 2.0 4.0
7  1.0 3.0 2.0 4.0
8  2.0 4.0 1.0 3.0
9  3.5 1.0 2.0 3.5
10 4.0 1.0 3.0 2.0
11 4.0 2.0 3.0 1.0
12 3.5 1.0 2.0 3.5

> friedman.test(x)

       Friedman rank sum test

data:  x
Friedman chi-squared = 8.0973, df = 3, p-value = 0.04404

# 修正的统计量
T2<-function(T1,b,k){
   T2<- (b-1)*T1/(b*(k-1)-T1)
   names(T2)<-"Correct Friedman F"
   p<-1-pf(T2,b-1,(b-1)*(k-1))
   names(p)<-"p value"
   res<-list(statistic=T2,p.value=p)
```

```
   res}

> T2(friedman.test(x)$statistic,dim(x)[1],dim(x)[2])
$statistic
Correct Friedman F
        3.192198

$p.value
   p value
0.004782398
```

48.6.2 Quade检验

Quade检验([19] Page 272).建立在每一区组原始观测值极差的基础上.

下面是一个例子. 5种品牌的洗衣粉在7个商店排开, 一周后, 计算销售数量, 看看是否品牌之间的销售有差异.

```
x<-matrix(c(5,4,7,10,12,
   1,3,1,0,2,
   16,12,22,22,35,
   5,4,3,5,4,
   10,9,7,13,10,
   19,18,28,37,58,
   10,7,6,8,7),
   nc=5,byrow=T,
   dimnames=list(1:7,c("A","B","C","D","E")))
> x
   A  B  C  D  E
1  5  4  7 10 12
2  1  3  1  0  2
3 16 12 22 22 35
4  5  4  3  5  4
5 10  9  7 13 10
6 19 18 28 37 58
7 10  7  6  8  7
```

```
> quade.test(x)
        Quade test

data:  x
Quade F = 3.8293, num df = 4, denom df = 24, p-value = 0.01519
```

TODO: 若拒绝零假设, 多重比较见参考文献 [19] Page 273.

48.6.3 Friedman检验与Kendall系数及Spearman系数的关系

TODO: 参考文献 [19] Page 279.

48.6.4 交互作用

对于交互作用, 没有什么好的非参数方法, 参考文献 [19] Page 281.

48.7 平衡的不完全区组设计

完全区组设计中, 每个区组应用所有的处理. 当区组大小有限, 处理又比较多的时候, 很难做到. 每个区组就使用所有处理中的一部分, 叫做不完全区组. 平衡指满足下面条件的设计:(1) 每个区组有k个试验单元, (2) 每个处理出现在r个组中, (3) 每个处理出现的次数相同.

参数方法处理不完全区组设计基本是基于正态假设的. Durbin(1951)提出了一个秩检验可以检验平衡的不完全区组设计的零假设: 不同的处理之间没有显著差异. 若处理数和每个区组的单元数一样, Durbin检验可以转化为 Friedman 检验.

对于不完全区组设计的分析可以首先在每个区组将数据转化为秩, 然后应用软件中相应的程序, 例如SAS中的用于秩的不

完全区组设计程序, 或广义线性模型.

下面是 Durbin 检验的算法([19] Page 284). 我们记

- t=处理数
- k=每个区组的单元数 k¡t
- b=区组数
- r=每个处理出现的次数
- λ=同时出现第i处理和第j处理的区组数, 这里要求对每一对处理的λ相同
- x_{ij}表示区组i处理j的结果
- $R(x_{ij})$为每个区组赋秩
- $R_j = \sum_{i=1}^{b} R(x_{ij})$为第j个处理下的r个观测值的秩和. 若某些观测的秩相等, 推荐使用平均秩.

检验统计量为

$$T_1 = \frac{12(t-1)}{rt(k-1)(k+1)} \sum_{j=1}^{t} (R_j - \frac{r(k+1)}{2})^2$$

如果存在结, 则使用平均秩的方法. 记A为秩与平均秩的平方和

$$A = \sum_{i=1}^{b} \sum_{j=1}^{t} [R(x_{ij})]^2$$

同时计算校正因子

$$C = \frac{bk(k+1)^2}{4}$$

调整后的统计量为

$$T_1 = \frac{(t-1)\sum_{j=1}^{t}(R_j - \frac{r(k+1)}{2})^2}{A-C} = \frac{(t-1)[\sum_{j=1}^{t} R_j^2 - rC]}{A-C}$$

另外一个等价的方法是在秩与平均秩上使用通常的方差分析方法, 它仅是T_1的一个函数. 但是近年来的研究表明它更精确一些, 因此人们更愿意使用

$$T_2 = \frac{T_1/(t-1)}{(b(k-1)-T_1)/(bk-b-t+1)}$$

零分布: T_1逼近服从自由度为t-1的卡方分布. T_2逼近自由度为 (t-1,bk-b-t+1)的F分布.

零假设为: 每个区组中, 所有的赋秩都是等可能的, 即处理效应相同. 备择假设: 至少一个处理的效应表现的比某个其它处理不同.

多重比较: 若拒绝零假设, 则使用下面方法进行多重比较. 下式成立则认为处理i,j不同.

$$|R_i - R_j| > t_{1-\alpha/2}[\frac{(A-C)2r}{bk-b-t+1}(1-\frac{T-1}{b(k-1)}]^{1/2}$$

其中t分布的自由度为bk-b-t+1

下面是一个例子([19] Page 286). 7种冰激凌, 请7个人品尝打分. 每个人品尝3种. 并用1,2,3打分(赋秩)表明喜欢程度. 使用 Youden 方阵编排.

```
x<-matrix(c(2,3,0,1,0,0,0,
   0,3,1,0,2,0,0,
   0,0,2,1,0,3,0,
   0,0,0,1,2,0,3,
   3,0,0,0,1,2,0,
   0,3,0,0,0,1,2,
   3,0,1,0,0,0,2),
   nr=7,byrow=T, dimnames=list("人"=LETTERS[1:7],"冰 激 凌"=letters[1:7]))
> x
  冰 激 凌
人 a b c d e f g
  A 2 3 0 1 0 0 0
```

```
  B 0 3 1 0 2 0 0
  C 0 0 2 1 0 3 0
  D 0 0 0 1 2 0 3
  E 3 0 0 0 1 2 0
  F 0 3 0 0 0 1 2
  G 3 0 1 0 0 0 2
# 将0转换为缺失数据
> y[y==0]<-NA
> y
   冰激凌
人  a  b  c  d  e  f  g
  A  2  3 NA  1 NA NA NA
  B NA  3  1 NA  2 NA NA
  C NA NA  2  1 NA  3 NA
  D NA NA NA  1  2 NA  3
  E  3 NA NA NA  1  2 NA
  F NA  3 NA NA NA  1  2
  G  3 NA  1 NA NA NA  2
```

其中t=7,k=3,b=7,r=3,$\lambda = 1$.

```
durbin.test<-function(x){
   Rj=colSums(x,na.rm=T) # 列秩和
   d=dim(x) # 维数
   t=d[2] # 冰激凌种类
   b=d[1] # 区组数
   r=length(x[,1][!is.na(x[1,])]) # 每个处理被处理的次
数, 冰激凌被品尝的次数
   k=length(x[1,][!is.na(x[1,])]) # 区组的单元数
   T1=12*(t-1)*sum((Rj-r*(k+1)/2)^2)/(r*t*(k-1)*(k+1))
   T2=(T1/(t-1))/((b*(k-1)-T1)/(b*k-b-t+1))
   res=list(T1=T1,T2=T2)
   res
}

> durbin.test(y)
$T1
[1] 12
```

```
$T2
[1] 8
```

48.8 A.R.E. 不低于1的检验

本节描述的方法的渐近相对效率 A.R.E. 几乎总是大于1. 参数检验如果合适, 则 A.R.E. =1. 否则, A.R.E. 几乎总是大于1. 但是注意, A.R.E. 只是衡量检验的一个方法. 由于很难考虑所有的情况, 故通常使用 A.R.E. ([19] Page 290)

48.8.1 几个独立样本的 van der Waerden (正态得分)检验

van der Waerden (1952,1953)建议了一个简单的方法([19] Page 291). 即在计算中不是使用秩来替换数据, 而是用另外一些数据替换原始数据的秩, 例如近似正态分布的数据.

假设数据为k个随机样本组成. 每一个可能有不同的样本容量. 记第i个样本为 $x_{i1}, \cdots, x_{in_i}. i = 1, \cdots, k$. N 表示样本的总数. 给N个样本排序, 从1到N赋秩. 存在结时使用平均秩. 记 x_{ij} 的秩为 $R(x_{ij})$. 变换 $R(x_{ij})$ 为正态得分, 即标准正态分布的R/(N+1)分位数, 记作 $A_{ij} = z_{R(x_{ij})/(N+1)}$. k个样本每个的平均正态得分为

$$\bar{A}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} A_{ij}$$

方差为

$$S^2 = \frac{1}{N-1} \sum A_{ij}^2$$

检验统计量为

$$T_1 = \frac{1}{S^2}\sum_{i=1}^{k} n_i(\bar{A}_i)^2$$

零分布: 在分析了A的所有置换后, 可以得到T_1的精确分布, 但是很困难. 故经常使用自由度为k-1的卡方分布近似. 近似通常很好.

零假设为: 所有k个总体分布函数相同. 备择假设为: 至少一个总体比另外一个分布产生较大的观测值.

多重比较: 若拒绝了零假设, 那么, 如果下式成立, 则总体i,j不同.

$$|\bar{A}_i - \bar{A}_j| > t_{1-\alpha/2}(S^2\frac{N-1-T_1}{N-k})^{1/2}(\frac{1}{n_i}+\frac{1}{n_j})^{1/2}$$

其中t分布的自由度为N-k

下面是那个玉米的例子([19] Page 209, 293). 我们给出了正态得分. 剩下的计算统计量的任务交给读者完成吧.

```
> x1=scan()
1: 83 91 94 89 89 96 91 92 90
10:
Read 9 items
> x2=scan()
1: 91 90 81 83 84 83 88 91 89 84
11:
Read 10 items
> x3=scan()
1: 101 100 91 93 96 95 94
8:
Read 7 items
> x4=scan()
1: 78 82 81 77 79 81 80 81
9:
```

```
Read 8 items
> x=c(x1,x2,x3,x4)
> x
 [1]  83  91  94  89  89  96  91  92  90  91  90  81  83  84  83  88  91  89  84
[20] 101 100  91  93  96  95  94  78  82  81  77  79  81  80  81

# 分组
> g=c(rep(1,9),rep(2,10),rep(3,7),rep(4,8))
> summary(factor(g))
 1  2  3  4
 9 10  7  8

> rank(x) # 秩次
 [1] 11.0 23.0 28.5 17.0 17.0 31.5 23.0 26.0 19.5 23.0 19.5  6.5 11.0 13.5 11.0
[16] 15.0 23.0 17.0 13.5 34.0 33.0 23.0 27.0 31.5 30.0 28.5  2.0  9.0  6.5  1.0
[31]  3.0  6.5  4.0  6.5
> qnorm(rank(x)/(length(x)+1)) # 正态得分
 [1] -0.48373855  0.40467790  0.89380063 -0.03581663 -0.03581663  1.28155157
 [7]  0.40467790  0.65217899  0.14372923  0.40467790  0.14372923 -0.89380063
[13] -0.48373855 -0.29050677 -0.48373855 -0.18001237  0.40467790 -0.03581663
[19] -0.29050677  1.90221650  1.57921952  0.40467790  0.74355976  1.28155157
[25]  1.06757052  0.89380063 -1.57921952 -0.65217899 -0.89380063 -1.90221650
[31] -1.36762792 -0.89380063 -1.20404696 -0.89380063
```

48.8.2 等方差检验的正态得分法

Klotz(1962)介绍了应用正态得分进行两个样本的等方差检验的方法 ([19] Page 294). 计算检验统计量

$$T_3 = \frac{\sum_{i=1}^{n} A_i^2 - \frac{n}{N}\sum_{i=1}^{N} A_i^2}{(\frac{nm}{N(N-1)}[\sum_{i=1}^{N} A_i^4 - \frac{1}{N}(\sum_{i=1}^{N} A_i^2)^2])^{1/2}}$$

A_i表示正态得分, 样本容量为n,m. N=n+m. 若两个样本均值不同, 则应该先减去均值再赋秩计算正态得分. 还可以编程检验 Klotz 检验的精确p-值.

48.8.3 正态得分用于回归

将正态得分赋秩后, 按照回归算法计算.([19] Page 296)

48.8.4 正态得分与相关系数

换算正态得分后, 按照算法计算 Pearson 相关系数,([19] Page 296) 即为正态得分相关系数(而不是秩相关系数, 如 spearman相关系数)

48.8.5 随机正态离差

使用伪随机正态分布的数按照大小相应的替换秩. 这种方法感觉象蒙特卡洛方法. 每次, 每个人的替换都不同. 在现实的分析中很少使用, 但是它的 A.R.E. 与正态得分相同, 精确分布与参数检验相同, 所以人们从理论的角度很感兴趣.([19] Page 296)

48.8.6 寻找精确分布的方法

我们使用下面的例子介绍([19] Page 298). 例如 Mann-Whitney 检验中, 两个独立样本 $x_1, \cdots, x_n, y_1, \cdots, y_m$. 赋给$x_i$的秩是1到n+m中等可能取到的任何一个. 对于其它的值$x_1, \cdots, x_n, y_1, \cdots, y_m$也可以类推. 给x赋n个秩, 有$\binom{m+n}{n}$种可能, 每一种都是等可能的. 出现概率为$1/\binom{m+n}{n}$. 所以, 使用计数的方法可以得到统计量的零分布.

48.9 Fisher 随机化方法

用数据本身作为得分,是 Fisher(1935) 引出的, 结果检验就是传统的随机化方法([19] Page 300). 好像需要通过计数的方法来自己推导一下零假设的分布.

48.9.1 两个独立样本

两个独立样本 $x_1,\cdots,x_n,y_1,\cdots,y_m$. 统计量是

$$T_1=\sum x_i$$

假设是

$$H_0: E(x)=E(y)$$
$$H_1: E(x)\neq E(y)$$

对于双边检验, 我们将x,y视为一个含有m+n个数据的数组. 取出n个样本, 则方法有$\binom{m+n}{n}$种. 为了找到p分位数w_p, 考虑第$\binom{m+n}{n}(p)$个最小的次序和, 即T_1, 其中最大的T_1就是w_p. 通过计算从m+n个数中选择n个数使和小于或等于(或大于等于, 若T_1在右边)T_1的方式的个数, 再除以$\binom{m+n}{n}$就是p-值. 双边检验乘以2.

下面是一个例子([19] Page 302). 假设随机样本为x=(0,1,1,0,-2) y=(6,7,7,4,-3,9,14). 考察两个期望是否一样. 从12个数里面选择5个的方式为$\binom{12}{5}=792$. 显著性水平设为0.05, 双边. 那么(792)(0.025)=19.8 即寻找最小的20组统计量. x的和为0, 寻找小于等于0的和的组合数.

```
> a=c(0,1,1,0,-2,6,7,7,4,-3,9,14)
> k=combn(a,5) # 所有组合
> dim(k) # 组合数
[1]   5 792
> k[,1]
[1]  0  1  1  0 -2
> c=colSums(k) # 观测值的和
> length(c[c<=0]) # 0为x的和. 小于等于x和的组合个数有11个.
[1] 11
> 2*11/length(c) # 双边概率
[1] 0.02777778
> sort(c)[1:20] # 前20个最小的和
 [1] -4 -4 -3 -3 -1 -1  0  0  0  0  0  1  1  2  2  2  2  2  2  2
```

48.9.2 配对的随机化检验

([19] Page 303). 与符号检验的原理一样. 还可以参考二项分布中符号检验的例子.

Chapter 49

检验数据是否来自指定分布–Kolmogorov-Smirnov型统计量

原理是考察样本数据的经验分布函数与假设的分布函数之间之间垂直距离最大的差值作为统计量. 其统计量的分布比较复杂. (参考文献 [19] Page 317. 由于统计量的分布此参考文献中也没有列出详细的统计量分布的推导过程, 但是给出了统计量的计算方法)

49.1 检验数据是否来自某个分布–Kolmogorov-Smirnov Test

注意: 有另外一个函数专门检验正态分布– Shapiro–Wilk test, 但是样本量必须在3-5000之间

若y类型为numeric, 检验为xy是否来自同一个连续分布.

若y为字符串, 且表示一个分布, 则零假设为x来自y定义的分布. 后面是y分布的参数

```
> x <- rnorm(50)
> y <- runif(30)
# x y 是否来自同一个分布?
> ks.test(x, y)

        Two-sample Kolmogorov-Smirnov test

data:  x and y
D = 0.48, p-value = 0.0002033
alternative hypothesis: two.sided

# x 是否来自 a shifted gamma 分布 with shape 3 and rate 2?
> ks.test(x+2, "pgamma", 3, 2) # two-sided, exact

        One-sample Kolmogorov-Smirnov test

data:  x + 2
D = 0.317, p-value = 5.742e-05
alternative hypothesis: two.sided

# x 是否来自正态分布
> x<-rnorm(100)
> ks.test(x, "pnorm", 0, 1)

        One-sample Kolmogorov-Smirnov test

data:  x
D = 0.0634, p-value = 0.816
alternative hypothesis: two.sided
```

49.2 正态性检验: Shapiro–Wilk test

检验样本是否来自正态分布. 样本量必须在3-5000之间

```
> shapiro.test(rnorm(100, mean = 5, sd = 3))

        Shapiro-Wilk normality test
```

```
data:  rnorm(100, mean = 5, sd = 3)
W = 0.9821, p-value = 0.1930

> shapiro.test(rnorm(10000, mean = 5, sd = 3))
错 误 在shapiro.test(rnorm(10000, mean = 5, sd = 3)) :
       样 本 大 小 必 需 在3和5000之 间
```

Chapter 50

TODO:非参数回归

Chapter 51

其它非参数检验

51.1 其它非参数检验

```
?kruskal.test  Kruskal-Wallis rank sum test.
?ansari.test
?mood.test
?fligner.test
library(help=ctest)
help.search('test')
```

Chapter 52

Randomization test of independence(permutation test)

http://en.wikipedia.org/wiki/Randomization_test
http://udel.edu/~mcdonald/statrandind.html

也称为 permutation test

52.1 使用条件

两个正态分布的变量, 考察其独立性. 我们经常称为 R*C 列联表.

检验独立性的时候, 当样本量小的时候(某个格子观测数少), 此方法比 chi-squared test or G-test 要精确.

但是与 Fisher 精确性检验相当. 因为有时候没有适合的Fisher精确性检验的程序, 可以使用此方法.

52.2 零假设

零假设: 两个变量互相独立.

52.3 原理

在零假设下, 对列联表产生与原数据相同边际的随机数. 对于每次产生的随机 R*C 列联表, 计算 Pearson's chi-square 统计量. 大于或等于原 chi-square 值的比例就是 p-value.

例如下面是 Custer and Galli (2002) 跟踪两种鸟 great blue herons (Ardea herodias) and great egrets (Casmerodius albus), 统计它们降落的地点的类型

```
␣␣␣␣␣␣␣␣␣␣␣␣␣Heron␣␣␣Egret
Vegetation␣␣␣␣␣15␣␣␣␣␣␣␣8
Shoreline␣␣␣␣␣␣20␣␣␣␣␣␣␣5
Water␣␣␣␣␣␣␣␣␣14␣␣␣␣␣␣␣7
Structures␣␣␣␣␣6␣␣␣␣␣␣␣1


>␣m<-t(matrix␣(c(x,y),nc=2))
>␣m
␣␣␣␣␣[,1]␣[,2]␣[,3]␣[,4]
[1,]␣␣␣15␣␣␣20␣␣␣14␣␣␣␣6
[2,]␣␣␣␣8␣␣␣␣5␣␣␣␣7␣␣␣␣1
>␣chisq.test␣(m)␣#␣pearson's␣chi-square␣test

␣␣␣␣␣␣␣␣Pearson's␣Chi-squared␣test

data:␣␣m
X-squared␣=␣2.2812,␣df␣=␣3,␣p-value␣=␣0.5161

警告信息:
In␣chisq.test(m)␣:␣Chi-squared近似算法有可能不准

>␣chisq.test␣(m,sim=T,B=100000)␣#␣MC␣模拟
```

␣␣␣␣␣␣␣␣Pearson's␣Chi-squared␣test␣with␣simulated␣p-value␣(based␣on␣1e+05
␣␣␣␣␣␣␣␣replicates)

data:␣␣m
X-squared␣=␣2.2812,␣df␣=␣NA,␣p-value␣=␣0.538

>␣fisher.test␣(m)␣#␣fisher␣精 确 检 验

␣␣␣␣␣␣␣␣Fisher's␣Exact␣Test␣for␣Count␣Data

data:␣␣m
p-value␣=␣0.5491
alternative␣hypothesis:␣two.sided

Chapter 53

G-test for goodness-of-fit

参考

http://en.wikipedia.org/wiki/G-test
http://udel.edu/~mcdonald/statgtestgof.html

53.1 前言

G-test 也称为 likelihood-ratio or maximum likelihood statistical significance tests .

G-test for goodness-of-fit, 也称为 likelihood ratio test for goodness-of-fit, 是可以代替卡方拟合优度检验(chi-square test of goodness-of-fit)的方法.

以前建议使用卡方检验的地方现在越来越多的建议使用这个 G-test.

一般的卡方拟合优度检验是基于列连表的, 实际上它是对数似然比(log-likelihood ratio)的近似, 而后者是 G-test 的基础.

卡方拟合优度检验是 Karl Pearson 发展的方法, 因为那时候计算对数似然值还比较困难. 但是现在由于计算机的快速发

展已经不是问题了. 尤其是 1994 年 Sokal and Rohlf 的统计书出版后越来越流行.

53.2 使用条件

正态分布, 两个或多个变量. 检验观测值与理论频率是否相等(例如 1:1 或 1:2:1).

如果期望数很小, 可能 G-test 不能给出精确的答案. 此时应该使用 exact test 或 randomization test. 以前的建议是如果观测值小于 5 应该使用精确的方法. 那是在手工计算阶乘等比较麻烦的时候的标准. 现在, 建议样本量在 1000 一下就使用精确的方法.

在样本量为 50-1000的时候, 使用何种方法的结果差别并不明显, 所以可以使用卡方检验或 G-test.

53.3 零假设

零假设: 每个类别的观测数目符合理论的比例.

备择假设: 每个类别的观测数目不符合理论的比例.

53.4 检验统计量

卡方检验的统计量(Pearson's chi-squared test statistic)

$$\chi^2 = \sum_{ij} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{53.1}$$

O_{ij}是第i行j列的观测数, E_{ij}是理论数.

G-test 的检验统计量为

$$G = 2\sum_{ij} O_{ij} \cdot \ln(O_{ij}/E_{ij}) \tag{53.2}$$

53.5 分布与使用

G 的分布近似服从同样自由度的卡方分布.

对于样本量不是太小的情况, G-test 和卡方检验的结果相同.

但是, 当有某个 $|O_i - E_i| > E_i$ 时, G-test 对卡方分布的近似要比 Pearson 卡方检验的近似要好. 所以在这种情况下应该总是使用 G-test.

53.6 Chi-square vs. G-test

因为 G-test 是增量的, 所以可以用于精细的分析中, 例如 repeated G-tests of goodness-of-fit.

G-test 是似然比检验的一种, 很多时候没有对应的卡方检验. 但是卡方检验更为人所熟悉, 在结果差不多的时候, 应该使用读者熟悉的检验方法.

53.7 R 程序

下面链接有一个R程序实现 G-test

`http://www.psych.ualberta.ca/~phurd/cruft/`

53.8 Replicated G-tests of goodness-of-fit

有 4 个零假设.

零假设1: 每个单独的类别(实验)观测数与理论数相等. 这个和 G-test 一样. 检测每个独立的实验.

零假设2: 综合所有的类别(实验)与理论值相等.

零假设3: 所有的类别(实验)的比例相等.

零假设4: 综合的数据比例与理论值(期望)一致.

首先, 对每个单独的类别(实验)做 G-test, 其结果为 "individual G-values". 同时记录每个类别的自由度.

然后, 把单独的 G 值相加得到 "total G-value", 把自由度也相加, 得到总的自由度. 例如你的总 G=12.33, df=6, 使用

```
> 1-pchisq (13.43,6) # p-value
[1] 0.03669360
```

如果 p-value 很小, 则可以拒绝零假设: 综合数据的比例与理论相同.

然后, 把每个类别的观测值相加. 例如将所有实验的红色花朵个数相加, 粉红色个数相加, 白色个数相加. 做一个 G-test. 得到 "pooled G-value." 自由度为 "pooled degrees of freedom" (总个数减 1)

最后, total G-value 减去 pooled G-value, 同样 total degree 减去 pooled degree, 得到 "heterogeneity G-value" and "heterogeneity degrees of freedom." 使用卡方分布得到 p-value

如果 heterogeneity G-value 不显著, 那么可以接受零假设3: 各类别比例相同. 可以将数据混合到一起, 使用 pooled G-value 来检验零假设: 数据与理论比例相等.

如果 heterogeneity G-value 显著, 那么不可以混合数据, pooled

G-value 不可用. 这种情况下, 必须研究各类别的数据, 使用更加复杂的方法. (see Sokal and Rohlf 1995, pp. 722-724).

如果 heterogeneity G-value 和 pooled G-value 都不显著, 但是 total G-value 显著. 拒绝所有数据与理论一致, 但是不清楚是数据的异质性, 还是方差导致的.

```
Cross␣␣␣Red␣Pink␣White␣G-value␣d.f.␣P-value
A␣␣␣␣␣␣␣28␣␣␣56␣␣␣27␣␣␣␣␣␣␣0.03␣␣␣␣␣2␣␣␣␣0.986
B␣␣␣␣␣␣␣29␣␣␣56␣␣␣15␣␣␣␣␣␣␣5.98␣␣␣␣␣2␣␣␣␣0.050
C␣␣␣␣␣␣␣23␣␣␣53␣␣␣17␣␣␣␣␣␣␣2.73␣␣␣␣␣2␣␣␣␣0.256
D␣␣␣␣␣␣␣30␣␣␣60␣␣␣12␣␣␣␣␣␣␣11.16␣␣␣␣2␣␣␣␣0.004
E␣␣␣␣␣␣␣29␣␣␣49␣␣␣37␣␣␣␣␣␣␣3.49␣␣␣␣␣2␣␣␣␣0.174
F␣␣␣␣␣␣␣27␣␣␣46␣␣␣19␣␣␣␣␣␣␣1.40␣␣␣␣␣2␣␣␣␣0.497
G␣␣␣␣␣␣␣32␣␣␣52␣␣␣33␣␣␣␣␣␣␣1.46␣␣␣␣␣2␣␣␣␣0.481
H␣␣␣␣␣␣␣32␣␣␣58␣␣␣16␣␣␣␣␣␣␣6.38␣␣␣␣␣2␣␣␣␣0.041
pooled␣␣␣230␣␣430␣␣176

␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣total␣G␣␣␣␣32.63␣␣␣␣16␣␣␣0.008

␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣pooled␣G␣␣␣␣7.89␣␣␣␣␣2␣␣␣␣0.019
␣␣␣␣␣␣␣heterogeneity␣G␣␣␣␣24.74␣␣␣␣14␣␣␣0.037
```

Part VII

试验设计与分析

Chapter 54

参考文献和包介绍

54.1 主要参考文献

茆诗松, 周纪芗, 陈颖. 《试验设计》 中国统计出版社. 2004. [16]

Vikneswaran *An R companion to Experimental Design.* (ebook) [32]

54.2 R软件包

AlgDesign:

crossdes (依赖于 AlgDesign):

54.3 函数介绍

54.3.1 all.combin()

返回所有的可能的处理组合, 下面例子是3个阶段, 每个阶

段4个处理方法的所有不重复的处理的组合. 总的试验次数为$4*3*2=24$, 即4个不同数字的3排列数.

```
> all.combin(4,3)
     [,1] [,2] [,3]
 [1,]    1    2    3
 [2,]    1    2    4
 [3,]    1    3    2
 [4,]    1    3    4
 [5,]    1    4    2
 [6,]    1    4    3
 [7,]    2    1    3
 [8,]    2    1    4
 [9,]    2    3    1
[10,]    2    3    4
[11,]    2    4    1
[12,]    2    4    3
[13,]    3    1    2
[14,]    3    1    4
[15,]    3    2    1
[16,]    3    2    4
[17,]    3    4    1
[18,]    3    4    2
[19,]    4    1    2
[20,]    4    1    3
[21,]    4    2    1
[22,]    4    2    3
[23,]    4    3    1
[24,]    4    3    2
```

54.4 注意factor的使用

因子或区组要使用factor显示指明是因子(或在数据输入的时候就使用factor, 或在公式中使用factor), 否则自由度及方差分析结果不正确.

Chapter 55

单因子试验设计与分析

例子: 考察不同地方的绿茶的叶酸(folacin)的含量是否不同. [16] Page 8

绿茶是一个因子, 用A表示. 产地是水平, 如今选择了4个产地, 分别为 A1,A2,A3,A4, 就是因子A的4个水平.

为了测定误差需要重复. 重复相等的设计称为平衡设计, 重复数不相等的称为不平衡设计.

现在选择不平衡设计, 4个产地分别需要采7,5,6,6个样品. 每采一个样品就是一次试验. 下面把试验编号

Table 55.1: 24个样品(试验)的编号

A1	A2	A3	A4
1	8	13	19
2	9	14	20
3	10	15	21
4	11	16	22
5	12	17	23
6		18	24
7			

假设一天按照顺序完成24个试验. 考虑到, 人的注意力逐渐

减弱, 光照, 茶叶的状态等也在不断改变. 假设A4的叶酸含量低, 究竟是产地的问题还是光照的问题就不好说了.

为了避免这种现象, 一般采用随机化. 即按照随机的顺序来做24个试验. 最后的结果就可以排除其它干扰因素, 比较有说服力了.

这样安排的单因子试验称为不平衡(或平衡)的完全随机试验.

数据分析方法就是单因素方差分析(ANOVA).

Chapter 56

区组设计: 完全区组设计

区组中因子的水平被称为处理.

56.1 随机化完全(不完全)区组设计

如果每个区组包含所有的处理, 那么称为随机化完全区组设计

如果每个区组小于所有的处理个数, 即容纳不下所有的处理, 那么称为随机化不完全区组设计

下面是一个例子. ([16]Page 70-71, 例 3.1.1). 设有4种杀虫剂, 记为 A1,A2,A3,A4. 考察其杀虫效果. 这里杀虫剂是因子, 4种杀虫剂就是4个水平(处理). 选择了20亩地, 每块1亩, 如何安排试验?

方法1: 随机化设计. 随机选择5块, 试验杀虫剂A1, 在余下的随机选择5块试验A2, 依此类推. 数据分析试验单因子方差分析.

假设20亩地的某个特征(高低, 距离河远近, 肥力等)的差别会影响杀虫剂的效果, 此时就要区分此特征.

方法2: 随机化区组设计. 例如, 见示意图, 20亩地依据河远近划分为5个区组,这样每个区组的差异就尽可能的小, 每个区

组4块, 在区组内实施随机化.

```
===============
+++++++++++++++ 这是一条河
===============
A1 A3 A4 A2 区组1 (区组按照距离河的远近划分, 区组内4个水平随机分配)
A2 A4 A1 A3 区组2
A4 A1 A2 A3 区组3
A2 A3 A4 A1 区组4
A3 A2 A1 A4 区组5
```

这样, 每个区组内个各个处理出现1次, 次序随机.

56.2 统计分析(固定效应)

56.2.1 例子数据准备

例子来自 [16]Page 70-71, 例 3.1.3.

检验4种化学试剂对某新型抗侵蚀布料的作用. 选择了5匹布, 考虑到布匹的差异, 每匹布为一个区组. 以抗拉强度作为指标(都-70后, 不影响方差分析的结果). 数据见下面

```
intensity=c(3,-1,3,1,-3,3,-2,4,2,-1,5,2,4,3,-2,5,2,7,5,2) # 抗拉强度
deal=gl(4,5) # 4种试剂(处理)
block=gl(5,1,20) # 5匹布(区组)
data=data.frame(intensity,deal,block);
```

```
> data
   intensity deal block
1          3    1     1
2         -1    1     2
3          3    1     3
```

```
4          1    1     4
5         -3    1     5
6          3    2     1
7         -2    2     2
8          4    2     3
9          2    2     4
10        -1    2     5
11         5    3     1
12         2    3     2
13         4    3     3
14         3    3     4
15        -2    3     5
16         5    4     1
17         2    4     2
18         7    4     3
19         5    4     4
20         2    4     5
```

56.2.2 数据的一般表示

Table 56.1: 随机化完全区组设计的数据的一般表示

	区组1	区组2	$\cdots$	区组b	和	平均
处理1	y_{11}	y_{12}	$\cdots$	y_{1b}	T_1	$\bar{T}_1$
处理2	y_{21}	y_{22}	$\cdots$	y_{2b}	T_2	$\bar{T}_2$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\bar{T}_1$
处理v	y_{v1}	y_{v1}	$\cdots$	y_{vb}	T_v	$\bar{T}_v$
和	B_1	B_2	$\cdots$	B_b		$T=\sum_{i=1}^{v}\sum_{j=1}^{b} y_{ij}$
平均	$\bar{B}_1$	$\bar{B}_2$	$\cdots$	$\bar{B}_b$		$\bar{y}=T/vb$

56.2.3 统计模型(固定效应)

随机化完全区组设计的统计模型(固定效应)为

$$y_{ij}=\mu+a_i+b_j+\xi_{ij},\quad i=1,\cdots,v\quad j=1,\cdots,b$$

其中

- y_{ij}为第i个处理在第j区组内的试验结果
- μ为总均值,待估计
- a_i第i个处理的效应,待估计,且满足$a_1 + \cdots + a_v = 0$
- b_j第j个区组的效应,待估计,且满足$b_1 + \cdots + b_b = 0$
- ξ_{ij}为相互独立的试验误差, 待估计, 且服从$N(0, \sigma^2)$

56.2.4 处理和区组的均值和效应的估计

此小节两个函数的使用参考文献 An R companion to Experimental Design [32], Page 43

replications() 函数可以查看各因素的水平数. 由于在非平衡时返回list()对象, 所以可以快速判断一个设计是否是平衡的.

```
> replications(intensity~deal+block,data)
 deal block
    5     4
> !is.list(replications(intensity~deal+block,data)) # 是平衡的设计
[1] TRUE
```

```
> model.tables(aov(intensity~deal+block,data),"effects")  # 效应
Tables of effects

 deal
deal  # 模型系数a_i
  1   2   3   4
-1.5 -0.9  0.3  2.1

 block  # 模型系数b_i
block
   1    2    3    4    5
 1.90 -1.85  2.40  0.65 -3.10
> model.tables(aov(intensity~deal+block,data),"means")
```

```
Tables of means
Grand mean

2.1    # 总均值

 deal
deal    # 各处理均值
  1   2   3   4
0.6 1.2 2.4 4.2

 block
block    # 各区组均值
   1    2    3    4    5
 4.00  0.25  4.50  2.75 -1.00
```

56.2.5 方差分析的假设

我们关心的是v个处理是否彼此相等, 即假设检验为

```
H_0: a1=a2=...=0
H_1: a_i不全为0
```

56.2.6 方差分析表和检验统计量公式

Table 56.2: 随机化完全区组设计的方差分析表

来源	平方和	自由度	均方和	F比
处理	$S_A = \frac{1}{b}\sum_{i=1}^{v} T_i^2 - \frac{T^2}{vb}$	$f_A = v-1$	$MS_A = S_A/f_a$	$F = \frac{MS_A}{MS_e}$
区组	$S_B = \frac{1}{v}\sum_{j=1}^{b} B_j^2 - \frac{T^2}{vb}$	$f_B = b-1$	$MS_B = S_B/f_b$	$-$
误差	$S_e = S_T - S_A - S_B$	$f_e = (v-1)(b-1)$	$MS_e = S_e/f_e$	
总和	$T = \sum_{i=1}^{v}\sum_{j=1}^{b} y_{ij} - \frac{T^2}{vb}$	$f_T = vb-1$		

- 处理效应皆为0时, $S_A/\sigma^2 \sim \chi^2(v-1)$

- 区组效应皆为0时, $S_B/\sigma^2 \sim \chi^2(b-1)$
- $S_e/\sigma^2 \sim \chi^2(b-1)(v-1)$

并且它们相互独立.

因此检验处理效应皆为0的假设的检验统计量为

$$F = \frac{MS_A}{MS_e} \sim F(f_A, f_e)$$

56.2.7 结果与解释

我们将处理(deal)作为主要因子, 区组作为误差项, 分析结果为

```
> summary(aov(intensity~deal+Error(block),data=data))

Error: block
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals  4 91.300  22.825

Error: Within
          Df Sum Sq Mean Sq F value    Pr(>F)
deal       3 37.800  12.600  14.131 0.0003045 ***
Residuals 12 10.700   0.892
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

由于我们的兴趣只在不同处理(4种试剂)之间的效果是否显著不同, 为了排除布匹带来的影响, 我们划分了区组. 这时候, 就像清除垃圾一样, 把区组(不同布匹)的平方和分解出来, 简单的忽略掉就可以了.

处理对应的p值为$0.0003045 < 0.05$, 所以4种试剂对新型布料的抗拉强度影响差异显著. 因此应该改进布料设计.

此试验的误差估计为 $\sigma^2 = 0.892, \sigma = 0.94$

另外, 在平衡的完全区组设计中, 区组作为协方差和把block看作误差项(Error)是一样的(见注意的问题一节).

```
# 下面的结果和把block看作误差项(Error)是一样的. 但是后面的BIB等设计就不同了.
# 故建议使用 Error 项
# summary(aov(lm(intensity~deal+block,data=data))) 也可以
> anova(lm(intensity~deal+block,data=data))
          Df Sum Sq Mean Sq F value    Pr(>F)
deal       3 37.800  12.600  14.131 0.0003045 ***
block      4 91.300  22.825  25.598  8.49e-06 ***
Residuals  12 10.700   0.892
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

其中

```
Df       自由度
Sum Sq   平方和
Mean Sq  均方和 = 平方和/自由度
F value  F值=对应均方和/残差均方和
           例如 deal: F=14.131=12.600/0.892
                block: F=25.598=22.825/0.892
Pr(>F)   p值, <0.05 在此因子上不同水平之间有显著差异.
```

56.2.8 其它: 效应(系数)的估计

利用最小二乘法很容易获得各效应(系数)的估计

- $\hat{\mu} = \bar{y}$

- $\hat{a}_i = \bar{T}_i - \bar{y}$
- $\hat{b}_j = \bar{B}_j - \bar{y}$

拟合值与残差为

- $\hat{y}_{ij} = \hat{\mu} + \hat{a}_i + \hat{b}_j = \bar{T}_i + \bar{B}_j - \bar{y}$
- $e_{ij} = y_{ij} - \hat{y}_{ij} = y_{ij} - \bar{T}_i - \bar{B}_j + \bar{y}$

```
> lm(intensity~deal+block,data=data) # 求得各系数

Call:
lm(formula = intensity ~ deal + block, data = data)

Coefficients:
(Intercept)       deal2       deal3       deal4      block2      block3
      2.50        0.60        1.80        3.60       -3.75        0.50
     block4      block5
     -1.25       -5.00
> residuals(r.lm) # 残差, 使用 sum(residuals(r.lm)) 获得残差
和
          1           2           3           4           5
 5.000000e-01  2.500000e-01 -2.039181e-16 -2.500000e-01 -5.000000e-01
          6           7           8           9          10
-1.000000e-01 -1.350000e+00  4.000000e-01  1.500000e-01  9.000000e-01
         11          12          13          14          15
 7.000000e-01  1.450000e+00 -8.000000e-01 -5.000000e-02 -1.300000e+00
         16          17          18          19          20
-1.100000e+00 -3.500000e-01  4.000000e-01  1.500000e-01  9.000000e-01
> predict(r.lm) # 拟合值(预测值)
   1    2    3    4    5    6    7    8    9   10   11   12   13
2.50 -1.25 3.00 1.25 -2.50 3.10 -0.65 3.60 1.85 -1.90 4.30 0.55 4.80
  14   15   16   17   18   19   20
3.05 -0.70 6.10 2.35 6.60 4.85 1.10
```

56.3 多重比较

若处理效应是固定的, 并且方差分析认为处理效应之间有显著差异, 则不论区组是否固定, 都应该对处理效应做多重比较.

下面使用TukeyHSD方法多重比较. 结果显示, 除了 1-2, 2-3组之间外, 其它所有处理之间都差异显著($p > 0.05$.忽略区组效应).

```
> r=aov(lm(intensity~deal+block,data=data))
> TukeyHSD(r)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = lm(intensity ~ deal + block, data = data))

$deal
    diff        lwr     upr    p adj
2-1  0.6 -1.17307455 2.373075 0.7497637
3-1  1.8  0.02692545 3.573075 0.0462510
4-1  3.6  1.82692545 5.373075 0.0003006
3-2  1.2 -0.57307455 2.973075 0.2378784
4-2  3.0  1.22692545 4.773075 0.0014627
4-3  1.8  0.02692545 3.573075 0.0462510

$block
    diff       lwr       upr    p adj
2-1 -3.75 -5.8782725 -1.6217275 0.0008738
3-1  0.50 -1.6282725  2.6282725 0.9403445
4-1 -1.25 -3.3782725  0.8782725 0.3808236
5-1 -5.00 -7.1282725 -2.8717275 0.0000592
3-2  4.25  2.1217275  6.3782725 0.0002827
4-2  2.50  0.3717275  4.6282725 0.0192616
5-2 -1.25 -3.3782725  0.8782725 0.3808236
4-3 -1.75 -3.8782725  0.3782725 0.1278790
5-3 -5.50 -7.6282725 -3.3717275 0.0000227
5-4 -3.75 -5.8782725 -1.6217275 0.0008738
```

56.4 注意的问题

56.4.1 区组的必要性

假设不区分区组, 使用单因子方差分析, 结果就不显著($F = 1.97, p = 0.16$). 这样的结果就是错误的结果. 因为区组的方差混杂在处理中了.

```
> summary(aov(lm(intensity~deal,data=data)))
          Df  Sum Sq Mean Sq F value Pr(>F)
deal       3  37.800  12.600  1.9765 0.1582
Residuals  16 102.000   6.375
```

56.4.2 是否把区组看作另外一个因子(区组作为协方差)

1. 忽略区组(作为误差项)

前面提到, 如果只对处理感兴趣, 对区组不感兴趣, 把区组作为误差项(Error) 简单忽略掉Error就可以了.

若区组平方和相对误差平方和(残差平方和)较小, 那么以后的试验区组就不是必需的了,否则区组就很必要. 象前面的例子, 区组p值很小, 说明区组是很必要的.

2. 区组作为协方差项

假设我们还对不同区组感兴趣, 那么就可以把区组看作协方差, 使用协方差分析.

注意: 在不完全的平衡/非平衡设计中作为误差项和协方差分析的结果是不同的.

作为协方差分析的区组的F值, 存在争议

1) Anderson 和 MeLean (1974) 指出, 由于区组之间不是随机化

的, 所以区组的F值不是一个有意义的检验.

2) Box, Hunter 和 Hunter (1978) 指出: 虽然区组之间不是随机化的, 所以区组的F值不是一个有意义的检验, 不再显示出合理性, 但是如果诸误差是来自正态分布的样本, 则F值还是可以用来比较区组之间的显著性的.

对区组B的效应也可以有如下假设

H_0: b1=...=b_b=0
H_1: 至少1个b_j不为0

检验统计量为

$$F = \frac{MS_B}{MS_e}$$

3. 区组作为另外一个因子.

如果还要考察处理与区组的交互作用, 就按照双因子试验设计, 这时候需要对处理和区组的每一组合进行重复才能完成试验的分析, 否则残差全部被交互作用吃掉, 变为0, 无法得到F值.

双因子方差分析的结果为

```
> summary(aov(intensity~deal*block,data))
            Df Sum Sq Mean Sq
deal         3 37.800  12.600
block        4 91.300  22.825
deal:block  12 10.700   0.892

# 与上面一样
> summary(aov(lm(intensity~deal*block,data=data)))
            Df Sum Sq Mean Sq
deal         3 37.800  12.600
block        4 91.300  22.825
```

```
deal:block  12 10.700   0.892

# 与上面一样
> anova(lm(intensity~deal*block,data=data))
Analysis of Variance Table

Response: intensity
          Df Sum Sq Mean Sq F value Pr(>F)
deal       3 37.800  12.600
block      4 91.300  22.825
deal:block 12 10.700   0.892
Residuals   0  0.000
```

实践中, 建议, 首先不把区组作为因子; 其次需要考察区组效应的时候, 可以作为参考.

56.4.3 附: 协方差的假设条件

如果在Y和X（X是协变量）之间存在合理的关系，为了提高精确度和调整协变量中的处理平均值方差，我们就可以考虑使用协方差分析。利用协方差分析就可以确定X与Y之间关系的紧密程度。应该坚持协方差的有关假设条件，它们是：A）协变量是固定的，测量中不存在误差而且独立于处理；B）在排除区组和处理差异后，Y对X的回归是线性的并与处理和区组无关；C）误差是独立正态分布，具有零均值和普通方差。(参考第64章 统计咨询工作者被经常问及的三十个问题及解答)

56.5 随机效应

前面模型中的讨论都假定处理效应和区组效应是固定的. 在实际中这些效应可能是随机的. 有三种情况

1. 仅仅处理效应是随机的

2. 仅仅区组效应是随机的

3. 处理效应和区组效应都是随机的

下面以第二种情况为例. 另外两种情况可类似.

56.5.1 区组效应随机的统计模型

随机化完全区组设计的统计模型为

$$y_{ij} = \mu + a_i + b_j + \xi_{ij}, \quad i = 1, \cdots, v \quad j = 1, \cdots, b$$

与固定效应的区别在于

- b_j第j个区组的效应, 来自正态分布$N(0, \sigma_b^2)$[1],其中σ_b^2是区组效应的方差分量
- ξ_{ij}为相互独立的试验误差, 来自$N(0, \sigma^2)$. 且b_j, ξ_{ij}相互独立

56.5.2 建立假设

对于固定效应(处理效应)的假设为

```
H_0: a1=a2=...=0
H_1: a_i不全为0
```

对于随机效应(区组效应)假设为

$$H_{01} : \sigma_b^2 = 0$$
$$H_{11} : \sigma_b^2 > 0$$

[1]此处原文为$N(0, \sigma^2)$, 应为笔误

56.5.3 检验统计量

首先指出, 总平方和的分解是代数恒等式, 与效应是否随机无关. 方差分析表与固定效应完全一样.

在前面随机效应的假设下, 可以证明数学期望

$$\begin{aligned} E(S_A) &= (v-1)\sigma^2 + b\sum_{i=1}^{v} a_i^2 \\ E(S_B) &= (b-1)\sigma^2 + v(b-1)\sigma_b^2 \\ E(S_e) &= (v-1)(b-1)\sigma^2 \end{aligned}$$

使用下面统计量检验第一对假设

$$F = \frac{MS_A}{MS_e} \sim F(f_A, f_e)$$

使用下面统计量检验第二对假设

$$F = \frac{MS_B}{MS_e} \sim F(f_B, f_e)$$

56.5.4 方差分量的估计

由上面的数学期望容易得到

$$\begin{aligned} \hat{\sigma^2}_b &= \frac{MS_B - MS_e}{v} \\ \hat{\sigma^2} &= MS_e \end{aligned}$$

再次把前面布匹与化学试剂的例子方差分析结果列在下面.

```
> anova(lm(intensity~deal+block,data=data))
          Df Sum Sq Mean Sq F value    Pr(>F)
deal       3 37.800  12.600  14.131 0.0003045 ***
block      4 91.300  22.825  25.598  8.49e-06 ***
Residuals  12 10.700   0.892
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

此处5布匹是随机抽取的, 区组效应就是随机的, 且

$$b_j \sim N(0, \sigma_b^2), \quad j = 1, \cdots, b$$

计算估计区组效应的方差分量为

$$\sigma_b^2 = \frac{MS_B - MS_e}{v} = \frac{22.83 - 0.89}{4} = 5.485$$

```
> res=anova(lm(intensity~deal+block,data=data))
# 区组效应的方差分量为
> (res$Mean[2]-res$Mean[3])/4
[1] 5.483333
```

与误差0.892比较大的多, 故布匹之间的一致性是较差的. 从blok的F值, p值也能够看出来, p值很小, 说明其方差比误差很大.

56.6 模型的适合性

适合性包括正态性和方差齐性两个问题.

在缺少重复的时候, 对误差的方差齐性还缺少检验方法. 我们只能从数据产生过程中推断方差齐性. 例如, 数据是在相同或类似的环境下产生的, 可以认为方差是齐的.

关于正态性的诊断可以借助残差分析进行. 参考回归诊断 30

```
> a=resid(r) # 残差
> b=fitted(r) # 拟合值
> plot(a~b) # 残差图
> a1=rstandard(r)
> plot(a1~b) # 标准化残差图
```

56.7 另外一个例子

56.7.1 数据

8窝老鼠, 3种营养液, 看看体重情况. 这里窝是区组, 营养液为处理.

```
> xx # 体重
 [1] 50.1 58.2 64.5 47.8 48.5 62.4 53.1 53.8 58.6 63.5 64.2 72.5 71.2 68.4 79.3
[16] 41.4 45.7 38.4 61.9 53.0 51.2 42.2 39.8 46.2

> W=data.frame(block=gl(8,3),food=gl(3,1,24),w=xx)
> W
   block food    w
1      1    1 50.1
2      1    2 58.2
3      1    3 64.5
4      2    1 47.8
5      2    2 48.5
6      2    3 62.4
7      3    1 53.1
8      3    2 53.8
9      3    3 58.6
10     4    1 63.5
11     4    2 64.2
12     4    3 72.5
13     5    1 71.2
14     5    2 68.4
15     5    3 79.3
16     6    1 41.4
```

```
17     6    2 45.7
18     6    3 38.4
19     7    1 61.9
20     7    2 53.0
21     7    3 51.2
22     8    1 42.2
23     8    2 39.8
24     8    3 46.2
```

56.7.2 方差分析

此处使用一般的协方差分析或作为误差项处理结果一样. [2]

```
# 区组作为误差
> summary(aov(w~food+Error(block),data=W))

Error: block
          Df  Sum Sq Mean Sq F value Pr(>F)
Residuals  7 2376.38  339.48

Error: Within
          Df Sum Sq Mean Sq F value  Pr(>F)
food       2 144.92   72.46  2.9788 0.08358 .
Residuals 14 340.54   24.32
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

# 区组作为协方差
> summary(aov(w~food+block,data=W))
            Df  Sum Sq Mean Sq F value    Pr(>F)
food         2  144.92   72.46  2.9788   0.08358 .
block        7 2376.38  339.48 13.9564 2.461e-05 ***
Residuals   14  340.54   24.32
```

[2]或许是因为数据是平衡的, 且是完全的

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

56.7.3 多组比较

由于 TukeyHSD 不能接受带有 Error 项的 aov 结果, 故使用下面的形式. 结果是一样的.

```
# 两两比较的结果()
> TukeyHSD(aov(w~food,W))
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = w ~ food, data = W)

$food
     diff       lwr     upr    p adj
2-1 0.0500 -14.284993 14.38499 0.9999574
3-1 5.2375  -9.097493 19.57249 0.6334067
3-2 5.1875  -9.147493 19.52249 0.6388008

# 两两比较, 增加block, food的结果与前面是一样的
> TukeyHSD(aov(w~food+block,W))
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = w ~ food + block, data = W)

$food
     diff      lwr      upr    p adj
2-1 0.0500 -6.404193  6.504193 0.9997734
3-1 5.2375 -1.216693 11.691693 0.1207439
3-2 5.1875 -1.266693 11.641693 0.1249915

$block
        diff        lwr        upr    p adj
```

```
2-1  -4.700000 -18.9097676   9.5097676 0.9288512
3-1  -2.433333 -16.6431009  11.7764343 0.9982127
......
```

Chapter 57

区组设计: BIB设计(平衡不完全区组设计)

57.1 BIB设计(平衡不完全区组设计)

平衡不完全区组设计(balanced incomplete block design, BIB设计)

57.1.1 例子

假设某因子有4个处理(水平), 又有4个区组. 假设每个区组可以放4个处理, 共16次试验. 这是完全区组设计.

Table 57.1: 完全区组设计

	区组1	区组2	区组3	区组4
处理1	y_{11}	y_{12}	y_{13}	y_{14}
处理2	y_{21}	y_{22}	y_{23}	y_{24}
处理3	y_{31}	y_{32}	y_{33}	y_{34}
处理4	y_{41}	y_{42}	y_{43}	y_{44}

假设想省略4次试验, 做12次, 就是不完全区组设计. 下面是两个方案. 第二个是平衡的不完全区组(BIB)设计, 比较好.

Table 57.2: 不完全区组设计(不平衡, 不好)

	区组1	区组2	区组3	区组4
处理1	y_{11}	y_{12}	y_{13}	y_{14}
处理2	y_{21}	y_{22}		y_{24}
处理3	y_{31}	y_{32}	y_{33}	y_{34}
处理4		y_{42}		

Table 57.3: 不完全区组设计(平衡, 好)–BIB

	区组1	区组2	区组3	区组4
处理1	y_{11}	y_{12}		y_{14}
处理2		y_{22}	y_{23}	y_{24}
处理3	y_{31}	y_{32}	y_{33}	
处理4	y_{41}		y_{43}	y_{44}

57.1.2 BIB符合的三个条件

BIB设计符合3个条件. 设区组b个, 处理v个

1. 每个区组处理数相同. 记为k个.
2. 每个处理都在r个不同的区组出现, r称为处理重复数.
3. 任何一对处理在λ个不同区组相遇, λ称为相遇数.

这样, v个处理可以得到公平的比较, 因为每个处理重复数相同, 任何一个处理与其它处理可以在相同条件下比较的次数相同.

57.1.3 BIB的五个参数与三个必要条件

下面是BIB设计的五个参数, 称为BIB设计参数

- v: 处理数
- k: 每个区组的处理数
- r: 每个处理的重复数(每个处理在r个区组出现)
- b: 区组数
- λ: 相遇数, 任何一对处理在λ个不同区组相遇

例子中(表57.1.1) 的五个参数为

$$v = 4 \quad k = 3 \quad r = 3 \quad b = 4 \quad \lambda = 2$$

但是任意给定五个参数, 对应的BIB设计未必存在. 例如$v = 4, b = 3$就找不到BIB设计.

下面是BIB设计存在的三个必要条件

- $vr = bk$
- $r(k-1) = \lambda(v-1)$
- $b >= v, \quad r >= k$

证明: 略(见[16] Page 88)

57.1.4 使用R进行BIB设计

我们使用 crossdes 包的函数 find.BIB() 进行BIB设计. 此函数调用的是 AlgDesign 包的 optBlock() 函数.

用法

```
find.BIB(trt, b, k, iter = 30)
```

- trt: 处理数

- b: 区组数
- k: 每个区组包含的处理数
- iter: 调用 optBlock 函数的次数
- 结果: 每一行为一个区组,

注意需要首先使用 set.seed() 函数产生随机数种子. 结果在 .Random.seed 内.

产生后需要使用 isGYD() 函数检验是否为BIB设计.

下面是总处理数为4, 区组为4, 每个区组处理数为3 的BIB设计

```
> library(crossdes)
> set.seed(runif(1)) # 输入 .Random.seed 查看随机数种子结果
> d=find.BIB(trt=4,k=3,b=4); d # 产生BIB设计
     [,1] [,2] [,3]
[1,]    1    3    4
[2,]    1    2    4
[3,]    2    3    4
[4,]    1    2    3
> isGYD(d)  # 检验是否平衡(Yes)

[1] The design is a balanced incomplete block design w.r.t. rows.

> d=find.BIB(trt=4,k=3,b=6) # 区组为6个就不能得到BIB设计
> isGYD(d)

[1] The design is neither balanced w.r.t. rows nor w.r.t. columns.

> d=find.BIB(trt=4,k=3,b=8) # 区组为8个可以是BIB设计
> isGYD(d)

[1] The design is a balanced incomplete block design w.r.t. rows.

-------------------------------------
# 下面看看 isGYD 结果包含的内容
```

```
> a=isGYD(d)

[1] The design is a balanced incomplete block design w.r.t. rows.

> a
[[1]]
[1]  TRUE  TRUE  TRUE  TRUE FALSE FALSE

[[2]]
[1]  TRUE FALSE FALSE FALSE FALSE FALSE

$`Number of occurrences of treatments in d`
1 2 3 4
6 6 6 6

$`Row incidence matrix of d`
  1 2 3 4 5 6 7 8
1 0 1 1 1 0 1 1 1
2 1 0 1 1 1 0 1 1
3 1 1 1 0 1 1 0 1
4 1 1 0 1 1 1 1 0

$`Column incidence matrix of d`
  1 2 3
1 6 0 0
2 2 4 0
3 0 4 2
4 0 0 6

$`Concurrence w.r.t. rows`
  1 2 3 4
1 6 4 4 4
2 4 6 4 4
3 4 4 6 4
4 4 4 4 6

$`Concurrence w.r.t. columns`
   1  2  3  4
1 36 12  0  0
2 12 20 16  0
3  0 16 20 12
```

```
4  0  0 12 36
```

57.2 统计模型及分析

57.2.1 例子描述与BIB参数

要比较5种工艺(处理)对产品质量的影响. 有10个车间参与试验, 但是每个车间只能承担3种工艺试验. 每个车间的条件都被认可, 但是之间有差异, 故把10个车间看作10个区组. 每个区组内3个处理.

此处设计参数为

$$v = 5 \quad k = 3 \quad r = 6 \quad b = 10 \quad \lambda = 3$$

57.2.2 产生BIB设计

使用R产生BIB设计. 从结果看出, 这组参数可以产生BIB设计.

```
> d=find.BIB(trt=5,k=3,b=10); d
     [,1] [,2] [,3]
 [1,]    1    3    5
 [2,]    1    2    4
 [3,]    2    3    4
 [4,]    1    2    5
 [5,]    3    4    5
 [6,]    2    3    5
 [7,]    1    4    5
 [8,]    1    3    4
 [9,]    1    2    3
[10,]    2    4    5
> isGYD(d)

[1] The design is a balanced incomplete block design w.r.t. rows.
```

57.2.3 试验结果数据

下面进行试验, 结果如下(数据见 [16]Page 94). 为了和参考文献一致, 这里使用参考文献的BIB设计, 没有使用上面的BIB设计的结果.

```
# 试验数据
trt1=c(54.2,28.1,58.7,50.9,60.2,44.3)
trt2=c(51.0,36.6,42.6,25.7,21.3,30.9)
trt3=c(33.1,18.1,39.1,34.5,16.0,39.4)
trt4=c(46.6,45.4,31.6,30.6,46.6,55.3)
trt5=c(38.6,33.2,14.0,13.1,35.6,44.6)

# 处理和区组数据
y<-c(trt1,trt2,trt3,trt4,trt5)
tr<-gl(5,6)
b<-c(1:6,1,2,3,7,8,9,1,4,5,7,8,10,2,4,6,7,9,10,3,5,6,8,9,10)
data1<-data.frame(y=y,treat=factor(tr),workshop=factor(b))

> data1
     y treat workshop
1  54.2     1        1
2  28.1     1        2
3  58.7     1        3
4  50.9     1        4
5  60.2     1        5
6  44.3     1        6
7  51.0     2        1
8  36.6     2        2
9  42.6     2        3
10 25.7     2        7
11 21.3     2        8
12 30.9     2        9
13 33.1     3        1
14 18.1     3        4
15 39.1     3        5
16 34.5     3        7
17 16.0     3        8
18 39.4     3       10
19 46.6     4        2
```

```
20 45.4    4       4
21 31.6    4       6
22 30.6    4       7
23 46.6    4       9
24 55.3    4      10
25 38.6    5       3
26 33.2    5       5
27 14.0    5       6
28 13.1    5       8
29 35.6    5       9
30 44.6    5      10
```

57.2.4 统计模型

此统计模型与完全区组设计是一样的, 除了处理数不同. 为了方便参考列在下面

$$y_{ij} = \mu + a_i + b_j + \xi_{ij}, \quad i = 1, \cdots, v \quad j = 1, \cdots, b$$

其中

- y_{ij}为第i个处理在第j区组内的试验结果
- μ为总均值,待估计
- a_i第i个处理的效应,待估计,且满足$a_1 + \cdots + a_v = 0$
- b_j第j个区组的效应,待估计,且满足$b_1 + \cdots + b_b = 0$
- ξ_{ij}为相互独立的试验误差, 待估计, 且服从$N(0, \sigma^2)$

57.2.5 处理和区组的均值和效应的估计

此小节两个函数的使用参考文献 An R companion to Experimental Design [32], Page 43

replications() 函数可以查看各因素的水平数. 由于在非平衡时返回list()对象, 所以可以快速判断一个设计是否是平衡的.

```
> replications(y~treat+workshop,data=data1) # 处理数
  treat workshop
      6        3
> !is.list(replications(y~treat+workshop,data=data1))
[1] TRUE
```

各处理的均值和区组的均值. 此处的函数model.tables(r2,"means") 仅仅计算分组数据的均值, 不进行修正. 但是也有一定的参考意义.[1]

```
> r2=aov(y~treat+workshop,data=data1) # 后面再分析此结果
> model.tables(r2,"means")
Tables of means
Grand mean

37.33     # 截距, 总平均值

 treat
treat  # 各处理的估计量(均值)
   1    2    3    4    5
49.40 34.68 30.03 42.68 29.85

 workshop
workshop   # 各区组的估计量(均值)
   1    2    3    4    5    6    7    8    9   10
45.39 32.17 45.99 34.76 45.07 26.65 31.80 22.61 39.29 49.57
```

看到1,4比较高(好), 5比较低(差). 具体差异是否显著要看方差分析的结果.

处理效应和区组效应的估计量(均值)为

```
> model.tables(r2,"effects")  #
```

[1]此处各处理均值和区组均值与参考文献《试验设计》[16]的结果稍微有差异. 参考文献的第一个处理效应的估计为10.86, 其它也略有不同. 参考文献是算法对均值做了修正. 详细见参考文献[16]Page 91-95, 3.2.2节的算法描述

```
Tables of effects

 treat
treat    # 处理效应的估计量(模型中的a_i)
    1     2     3     4     5
12.070 -2.647 -7.297  5.353 -7.480

 workshop
workshop    # 区组效应的估计量(模型中的b_i)
     1      2      3      4      5      6      7      8      9     10
 8.061 -5.156  8.656 -2.572  7.739 -10.678 -5.533 -14.722  1.961 12.244
```

处理效应的极差为 $12.070-(-7.480)=19.550$, 区组效应的极差为 $12.244-(-14.722)=26.966$. 区组效应的极差大于处理效应的极差, 这是区组效应不能忽略的信号.

57.2.6 方差分析

我们关心的仍然是v个处理是否彼此相等, 即假设检验为

```
H_0: a1=a2=...=0
H_1: a_i不全为0
```

下面是方差分析的结果. 把区组(workshop)作为误差项. [2]

```
# 正确用法
> summary(aov(y~treat+Error(workshop),data=data1))

Error: workshop
          Df  Sum Sq Mean Sq F value Pr(>F)
treat      4  445.03  111.26  0.2763 0.8817
Residuals  5 2013.47  402.69
```

[2]如果区组作为协方差, 那么结果会不一样. 详细见参考文献[16]Page 91-95, 3.2.2节的算法描述

```
Error: Within
          Df  Sum Sq Mean Sq F value   Pr(>F)
treat      4 1575.33  393.83  7.0081 0.001852 **
Residuals 16  899.15   56.20
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

# 错误!!!!
> summary(aov(y~treat+workshop,data=data1))
           Df  Sum Sq Mean Sq F value   Pr(>F)
treat       4 1743.24  435.81  7.7550 0.001128 **
workshop    9 2290.59  254.51  4.5289 0.004265 **
Residuals  16  899.15   56.20
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

# 错误!!!!
# anova 得到同样的结果
> anova(lm(y~treat+workshop,data=data1))
Analysis of Variance Table

Response: y
          Df  Sum Sq Mean Sq F value   Pr(>F)
treat      4 1743.24  435.81  7.7550 0.001128 **
workshop   9 2290.59  254.51  4.5289 0.004265 **
Residuals 16  899.15   56.20
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

57.2.7 多重比较

TukeyHSD 不能接受带有 Error 项的 aov 结果, 故使用下面的形式. 结果是一样的.

结果发现, 1-4之间差异不显著, 1和其它差异显著. 选择工艺应该在1,4挑选.

若低者为好, 那么5-3之间差异不显著, 应该推广5,3工艺.

```
# 下面两个形式结果是一样的
> TukeyHSD(aov(y~treat,data=data1))
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = y ~ treat, data = data1)

$treat
         diff       lwr       upr     p adj
2-1 -14.7166667 -33.869466  4.4361324 0.1924719
3-1 -19.3666667 -38.519466 -0.2138676 0.0465595
4-1  -6.7166667 -25.869466 12.4361324 0.8391485
5-1 -19.5500000 -38.702799 -0.3972009 0.0437827
3-2  -4.6500000 -23.802799 14.5027991 0.9515505
4-2   8.0000000 -11.152799 27.1527991 0.7362579
5-2  -4.8333333 -23.986132 14.3194657 0.9446042
4-3  12.6500000  -6.502799 31.8027991 0.3235218
5-3  -0.1833333 -19.336132 18.9694657 0.9999998
5-4 -12.8333333 -31.986132  6.3194657 0.3100057

# 结果一样, 忽略block
> TukeyHSD(aov(y~treat+workshop,data=data1))
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = y ~ treat + workshop, data = data1)

$treat
         diff        lwr       upr     p adj
2-1 -14.7166667 -27.9765084 -1.4568249 0.0260316
3-1 -19.3666667 -32.6265084 -6.1068249 0.0030359
4-1  -6.7166667 -19.9765084  6.5431751 0.5462231
5-1 -19.5500000 -32.8098417 -6.2901583 0.0027892
3-2  -4.6500000 -17.9098417  8.6098417 0.8167745
4-2   8.0000000  -5.2598417 21.2598417 0.3819344
5-2  -4.8333333 -18.0931751  8.4265084 0.7955741
4-3  12.6500000  -0.6098417 25.9098417 0.0652490
5-3  -0.1833333 -13.4431751 13.0765084 0.9999992
```

```
5-4 -12.8333333 -26.0931751  0.4265084 0.0602630

$workshop
          diff        lwr       upr    p adj
2-1  -13.2166667 -35.5054086  9.0720753 0.5166657
3-1    0.5944444 -21.6942975 22.8831864 1.0000000
4-1  -10.6333333 -32.9220753 11.6554086 0.7613632
......
```

所有10个区组两两之间的比较, 但是这里我们只关心处理之间的.

Chapter 58

区组设计: 链式区组设计

设因子A有v个水平(处理), 但是v比较大, 例如几十个水平需要比较, 此时使用链式区组设计只需要比v略多一些的观察就可以组成链式区组设计. 试验费用较大和试验误差较小的时候可以采用链式区组设计.

前提条件: 在使用链式区组设计之前, 必须确认处理需要间的重要差异明显大于试验误差.

58.1 构造链式区组设计

链式区组设计的构造较简便. 下面是一个例子[16] 3.3.1 Page 102

对同一炉铁水铸成的42根棒要用光谱法测定镍含量. 一个光谱底版只能同时测量18根. 不同的光谱底版对测量结果有影响. 如何安排试验获得镍含量呢?

可以将每个底版看作一个区组, 42根棒子至少需要3个区组, 共可以测量$18 * 3 = 54$根棒, 比需要的多12根. 多于部分可以用于重复. 即42根棒中有12根重复测量一次. 这样42根棒按照随机方法分为两组, 一组30根, 一组12根.

12根的组每根观察2次. 如此共24次观察. 如何将24次观察分到3个区组是链式设计的关键.

把12根棒记为$1, 2, \cdots, 12$, 分为3组

$$A_1 = 1, 2, 3, 4 \quad A_2 = 5, 6, 7, 8 \quad A_3 = 9, 10, 11, 12$$

把这3组依次放入三个区组.

若把重复观察的12根棒记为$1', 2', \cdots, 12'$, 也分为3组. 为了能够形成链状, 把$A_1' = 1', 2', 3', 4'$放第三区组, $A_2' = 5', 6', 7', 8'$放第一区组, $A_3' = 9', 10', 11', 12'$放第二区组.

余下的30根随机的尽量平均的放入3个区组.

Table 58.1: 3区组42处理链式区组设计,$n = 54$

1	2	3
1	5	9
2	6	10
3	7	11
4	8	12
5’	9’	1’
6’	10’	2’
7’	11’	3’
8’	12’	4’
13	23	33
14	24	34
15	25	35
16	26	36
17	27	37
18	28	38
19	29	39
20	30	40
21	31	41
22	32	42

58.2 数据和分析

58.2.1 数据

```
# y为测量结果, 已经减去一个常数
y=c(8,7,14,9,13,15,12,9,11,5,17,14,12,13,14,12,8,21,
   4,3,10,6,5,7,2,6,10,9,6,7,6,4,7,7,9,10,
   -1,0,-3,-8,1,5,2,0,5,-1,-3,-6,2,-2,-2,0,1,2)
x=c(1:8,13:22,5:12,23:32,9:12,1:4,33:42) # 编号
spec=data.frame(y,id=factor(x),block=gl(3,18))
```

```
> spec
   y id block
1   8  1     1
2   7  2     1
3  14  3     1
4   9  4     1
5  13  5     1
6  15  6     1
7  12  7     1
8   9  8     1
9  11 13     1
10  5 14     1
11 17 15     1
12 14 16     1
13 12 17     1
14 13 18     1
15 14 19     1
16 12 20     1
17  8 21     1
18 21 22     1
19  4  5     2
20  3  6     2
21 10  7     2
22  6  8     2
23  5  9     2
24  7 10     2
25  2 11     2
26  6 12     2
```

```
27 10 23     2
28  9 24     2
29  6 25     2
30  7 26     2
31  6 27     2
32  4 28     2
33  7 29     2
34  7 30     2
35  9 31     2
36 10 32     2
37 -1  9     3
38  0 10     3
39 -3 11     3
40 -8 12     3
41  1  1     3
42  5  2     3
43  2  3     3
44  0  4     3
45  5 33     3
46 -1 34     3
47 -3 35     3
48 -6 36     3
49  2 37     3
50 -2 38     3
51 -2 39     3
52  0 40     3
53  1 41     3
54  2 42     3
```

58.2.2 方差分析

$F = 0.867, p = 0.65$, 结果不显著.

```
> summary(aov(y~id+Error(block),spec))

Error: block
  Df  Sum Sq Mean Sq
```

```
id  2 1377.33  688.67

Error: Within
          Df Sum Sq Mean Sq F value Pr(>F)
id        41 422.00   10.29  0.8674 0.6509
Residuals 10 118.67   11.87

# 错误!!!! block作为协方差, 错误. 可能因为是不平衡引起的.
> summary(aov(y~id+block,spec))
            Df  Sum Sq Mean Sq F value   Pr(>F)
id          41 1507.00   36.76  3.0974 0.029685 *
block        2  292.33  146.17 12.3174 0.002006 **
Residuals   10  118.67   11.87
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

58.2.3 TODO: 处理效应和区组效应的估计

略. R未找到产生的函数. 参考文献试验设计 [16] Page 105-110, 有算法描述.

Chapter 59

正交设计: 正交设计

59.1 多因子试验

59.1.1 多因子试验的复杂性

实际问题中, 影响试验的因子往往很多, 这涉及多因子试验的问题. 最大的问题是试验次数太多. 例如: 10个因子, 每个因子仅仅取2个水平, 就有$2^10 = 1024$个不同水平组合, 每个组合就是一个试验条件, 每个条件重复3次就是$2^10 * 3 = 3072$次试验. 这往往是不能忍受的.

传统采用单因子轮换的方法, 即逐个改变因子水平, 其它因子水平固定, 挑选最好的结果. 这样就化为多个单因子试验. 但是每个单因子试验挑选的最好水平组合起来不一定是全局最好的.

59.1.2 常用的多因子试验设计方法

常用的方法有

- 正交设计

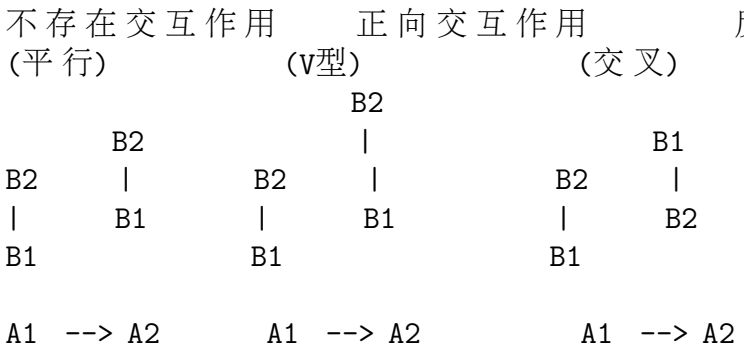
- 参数设计
- 回归设计
- 均匀设计
- 混料设计

59.1.3 交互作用

定义: 一个因子的水平的好坏(程度)受另一因子水平制约的情况, 称为因子A与B的交互作用. 记为AB或AB

- 不存在交互作用: A水平变化时, B的各水平之间的差异程度不变(绝对值可能变化)
- 正向交互作用: A水平均值提高, B的各水平之间的差异变大或变小
- 反向交互作用: A水平1时, $B1 > B2$, 但是A水平2时, $B1 < B2$, 即B的差异相反

图示三种交互作用



交互作用随因子数增加而增加, 例如4因子A,B,C,D之间的交互作用有

- 二级交互作用6个: AB,AC,AD,BC,BD,CD
- 三级交互作用4个: ABC,ABD,ACD,BCD
- 四级交互作用1个: ABCD

共11个, 比因子数目还多.

实际经验表明, 多数交互作用不存在或很小, 以致可以忽略不计. 实际中主要考虑部分二级交互作用. 具体哪些还需要依赖专业知识.

59.2 正交表

59.2.1 正交表的符号表示

正交表使用符号

$$L_n(q^p)$$

表示. 其中

- L: 正交表的代号
- n: 表的行数, 即不同条件的试验个数
- q: 因子的水平数
- p: 表的列数, 即因子数

59.2.2 正交表的正交性

正交表的正交性指

- 每列重复数字的重复次数相同
- 任意两列的同行数字看作一个数对, 那么一切可能数对重复次数相同

59.2.3 正交表的分类

按照水平数分类

- 二水平正交表: $L_4(2^3), L_8(2^7), L_16(2^15), L_12(2^11)$
- 三水平正交表: $L_9(3^4), L_27(3^13), L_18(3^7)$
- 四水平正交表: $L_16(4^15)$
- 五水平正交表: $L_25(5^6)$
- 混合水平正交表: $L_18(2*3^7)$

按照行列,水平等关系分类

- 完全正交表:

$$n = q^k, \quad k = 2, 3, \cdots \qquad p = \frac{n-1}{q-1}$$

  可以考察因子的交互作用, 每个正交表都附一个交互作用列表. 其中$L_4(2^3), L_9(3^4), L_16(4^15)L_25(5^6)$中任意两列的交互作用是其它各列, 故不再给出交互作用表.

- 不完全正交表, 上面关系至少1个不成立.

59.3 无交互作用的正交设计

整体设计: 正交表安排试验是事先一起设计好的. 而不是等一个试验结束再决定下一个试验的水平组合. 这样的设计称为整体设计. 例子就是整体设计.

部分实施: 例子中3因子3水平, 不同的组合有27个, 但是只做9个组合, 是一个部分实施的设计方案. 由于仅仅做了1/3, 也称为1/3实施.

某化工厂希望提高产品转化率. 安排试验时一般考虑下面步骤.

1. 明确试验目的: 本试验目的提高转化率

2. 明确试验指标: 判断水平组合的好坏. 直接使用转化率. 越大越好, 是望大指标

3. 确定因子与水平: 本试验经分析确定影响转化率的可能因子有3个

   A:反应时间　　B:反应温度　　C:加碱量

   经过专业人员分析, 根据可能的范围, 采用下面的水平

Table 59.1: 因子水平表

水平	1	2	3
A:温度	80.00	85.00	90.00
B:时间	90.00	120.00	150.00
C:加碱量(%)	5.00	6.00	7.00

4. 选择合适的正交表, 进行表头设计. 表头不同, 选择的9个试验也不同但是效果是相同的.

   (a) 根据考察的水平选择具有该水平的正交表, 再根据因子的个数选择具体的表. 本试验3水平, 3因子, 故选择$L_9(3^4)$是合适的.

   (b) 表头设计: 把因子放到表的列上, 称为表头设计. 不考虑交互作用的时候, 可以任意把因子放到任意列. 一个因子一列.

Table 59.2: 表头设计

表头设计	A	B	C	
$L_9(3^4)$ 列号	1	2	3	4

5. 列出试验计划: 将正交表列中的数字换为因子的相应水平即可. 不放因子的列不考虑.

6. 进行试验和记录试验结果: 为避免某些系统误差, 9次试验的次序要随机. 而且试验中其它条件要控制的一致. 避免操作人员, 仪器的差异引起系统误差. 不可避免的时候可以增加一个区组因子, 例如试验由3个人做, 可以把"人"看作一个因子, 放在正交表的空白列. 该列的1,2,3对应3个人.

   一个水平组合下也可以分析, 但是可能的情况下, 同一水平组合若干重复, 这样可以观察试验的稳定性, 还可以对误差进行估计.

   试验要专业人员做. 结果的记录也要可靠.

   下面是试验结果.

Table 59.3: 转化率试验结果

	A	B	C	(无用, 或人员列区组)	试验结果(y)
1	1	1	1	1	31
2	1	2	2	2	54
3	1	3	3	3	38
4	2	1	2	3	53
5	2	2	3	1	49
6	2	3	1	2	42
7	3	1	3	2	57
8	3	2	1	3	62
9	3	3	2	1	64

59.4 数据直观分析

59.4.1 试验结果

数据如下

```
# 9次试验水平组合及其结果
tr=data.frame(matrix(c(1,1,1,2,2,2,3,3,3,
   1,2,3,1,2,3,1,2,3,
   1,2,3,2,3,1,3,1,2,
```

```
   1,2,3,3,1,2,2,3,1,
   31,54,38,53,49,42,57,62,64),
   nc=5,dimnames=list(c(),c("A","B","C","","y"))))
> tr # v4没有用到，如果需要,可以作为区组
  A B C V4  y
1 1 1 1  1 31
2 1 2 2  2 54
3 1 3 3  3 38
4 2 1 2  3 53
5 2 2 3  1 49
6 2 3 1  2 42
7 3 1 3  2 57
8 3 2 1  3 62
9 3 3 2  1 64
```

59.4.2 直接观察

试验结果最大的为最好. 看到第9号试验结果为64, 最大, 故可以认为对应的水平组合$A_3B_3C_2$最好.

```
> which(tr$y==max(tr$y)) # 试验结果最大的试验号
[1] 9
```

但是在全部27个试验中是否最好? 这需要利用正交表的综合可比性来分析.

59.4.3 综合可比性

首先看第一列.

第一列1,2,3表示因子A的3个水平. 1对应的三个试验都采用A的一水平, 但是B的三个水平各自参加了一次试验, C的三个水平也各自参加了一次试验. 2和3对应的3个试验也一样. 和与均值为

```
> attach(tr)
> T=aggregate(y,by=list(A),FUN="sum"); T #A因子3个水平的求和
  Group.1   x
1       1 123 # A水平1对应的试验的和
2       2 144 # A水平2对应的试验的和
3       3 183 # A水平3对应的试验的和
> M=aggregate(y,by=list(A),FUN="mean");M # 均值
  Group.1  x
1       1 41 # A水平1对应的试验的均值
2       2 48 # A水平2对应的试验的均值
3       3 61 # A水平3对应的试验的均值
```

这样, T之间的差异(也就是M的差异)只反应了因子A的3个水平之间的差异. 所以可以通过这三个值的大小比较因子A的三个水平的好坏.

可知因子A的三水平最好, 其指标最大.

这种方法称为综合比较, 是由正交表的正交性决定的.

其它两列也类似. 结果列在下面

```
Ta=aggregate(y,by=list(A),FUN="sum")#A因子3个水平的求和
Tb=aggregate(y,by=list(B),FUN="sum")#B因子3个水平的求和
Tc=aggregate(y,by=list(C),FUN="sum")#...
Td=aggregate(y,by=list(V4),FUN="sum")

Ma=aggregate(y,by=list(A),FUN="mean")
Mb=aggregate(y,by=list(B),FUN="mean")
Mc=aggregate(y,by=list(C),FUN="mean")
Md=aggregate(y,by=list(V4),FUN="mean")

> tmp1=matrix(c(Ta[,2],Tb[,2],Tc[,2],Td[,2]),nc=4)
> tmp1
     [,1] [,2] [,3] [,4]
[1,]  123  141  135  144
[2,]  144  165  171  153
[3,]  183  144  144  153
> tmp2=matrix(c(Ma[,2],Mb[,2],Mc[,2],Md[,2]),nc=4)
```

```
> tmp2
     [,1] [,2] [,3] [,4]
[1,]   41   47   45   48
[2,]   48   55   57   51
[3,]   61   48   48   51
```

将统计数据放入表中, 如下

Table 59.4: 转化率试验结果(直观统计表)

	A	B	C	(无用, 或人员列区组)	试验结果(y)
1	1	1	1	1	31
2	1	2	2	2	54
3	1	3	3	3	38
4	2	1	2	3	53
5	2	2	3	1	49
6	2	3	1	2	42
7	3	1	3	2	57
8	3	2	1	3	62
9	3	3	2	1	64
T1(水平1的和)	123	141	135	144	
T2(水平2的和)	144	165	171	153	
T3(水平3的和)	183	144	144	153	
M1(水平1的均值)	41	47	45	48	
M2(水平2的均值)	48	55	57	51	
M3(水平3的均值)	61	48	48	51	

59.4.4 水平均值图

将每个因子的不同水平均值绘图,

```
par(mfrow=(c(1,3)))
plot(Ma[,2]~Ma[,1],t='o',xlab="levels",ylab="mean",main="A")
plot(Mb[,2]~Mb[,1],t='o',xlab="levels",ylab="mean",main="B")
```

```
plot(Mc[,2]~Mc[,1],t='o',xlab="levels",ylab="mean",main="B")
```

可以看到, 每个因子最好的水平分别为A_3, B_2, C_2.

也可以看到每个因子水平之间的最大差异.

59.4.5 极差

利用极差分析个因子对指标的影响程度. 极差大的因子对指标的影响就比较大.

```
> max(Ma[,2])-min(Ma[,2]) # A因子3个水平的极差, R_A
[1] 20
> max(Mb[,2])-min(Mb[,2]) # B因子3个水平的极差,R_B
[1] 8
> max(Mc[,2])-min(Mc[,2]) # C因子3个水平的极差, R_C
[1] 12

> max(Md[,2])-min(Md[,2]) # 空白列极差, 表明误差. 最小
[1] 3
```

看到因子A对指标影响最大, B最小, C其次. 通常记为$R_A > R_C > R_B$.

空白列极差, 表明误差较小.

59.4.6 总结

利用直观分析可以获得下面的结论

1. 获得最佳水平组合. 最好的组合是A_3, B_2, C_2, 与9个试验的最好水平组合A_3, B_3, C_2不同. 因为直观分析是从27个可能的水平组合中比较出来的. 至于B_2, B_3对指标影响多大, 还需要进一步分析, 或试验验证.

2. 区分因子的主次. 本例中, 因子A是主要因子, C次之, B再次之, 空白列极差最小. 表明试验误差比较小.

59.5 数据的方差分析

59.5.1 统计模型

若想使用方差分析, 需要对试验结果做出某些假设

1. 假定同一水平组合下试验结果的全体构成一个总体, 服从正态分布, 本例中共有9个总体
2. 各正态总体的方差是相同的, 均为σ^2
3. 各正态均值与水平组合有关, 即

$$\mu_{ijk} = \mu + a_i + b_j + c_k$$

a_i, b_j, c_k分别为因子A第i水平的主效应, 等等. 满足

$$a_1 + a_2 + a_3 = 0$$
$$b_1 + b_2 + b_3 = 0$$
$$c_1 + c_2 + c_3 = 0$$

称为效应可加模型.

4. 不同水平组合下的试验是相互独立的.

模型可以表示为

$$\mu_{ijk} = \mu + a_i + b_j + c_k + \xi_{ijk}, \qquad \xi \sim N(0, \sigma^2)$$

59.5.2 假设检验

在上述指定模型下, 检验假设

$$H_{A0} : a_1 = a_2 = a_3 = 0 \qquad H_{A1} : \text{至少一个} a_i \neq 0$$
$$H_{B0} : b_1 + b_2 + b_3 = 0 \qquad H_{B1} : \text{至少一个} b_i \neq 0$$
$$H_{C0} : c_1 + c_2 + c_3 = 0 \qquad H_{C1} : \text{至少一个} c_i \neq 0$$

59.5.3 方差分析和结论

方差分析公式: 略

使用如下R模型进行方差分析, 两个都可以.(注意factor的使用. 不加入factor结果是错误的)

```
> anova(lm(y~factor(A)+factor(B)+factor(C),data=tr))
Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value  Pr(>F)
factor(A)  2    618     309 34.3333 0.02830 *
factor(B)  2    114      57  6.3333 0.13636
factor(C)  2    234     117 13.0000 0.07143 .
Residuals  2     18       9
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

# the same
> summary(aov(y~factor(A)+factor(B)+factor(C),data=tr))
            Df Sum Sq Mean Sq F value  Pr(>F)
factor(A)    2    618     309 34.3333 0.02830 *
factor(B)    2    114      57  6.3333 0.13636
factor(C)    2    234     117 13.0000 0.07143 .
Residuals    2     18       9
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

可以看到, 因子A,C都对指标影响显著, B的影响是不显著的.

对于显著的因子要选择最好的水平, 因为其变化常常引起指标的显著不同. 不显著的指标可以任意选择, 实际中可以根据需要选择最节约的, 成本最低的, 操作最方便的等等.

本例中A,C要选择最好的, 最后, 选择的最佳组合水平为A_3BC_2. B为了节约时间可以选择B_1. 最后确定的组合即为$A_3B_1C_2$. (直观分析中结果最好的是$A_3B_2C_2$, 也是从27个组合中选择的)

从下面的两两比较也可以看出来水平之间的差异显著性($p < 0.05$). 不过比较麻烦

```
> TukeyHSD(aov(y~factor(A)+factor(B)+factor(C),data=tr))
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = y ~ factor(A) + factor(B) + factor(C), data = tr)

$`factor(A)`
   diff      lwr     upr    p adj
2-1    7 -7.429339 21.42934 0.1826267
3-1   20  5.570661 34.42934 0.0266692
3-2   13 -1.429339 27.42934 0.0608887

$`factor(B)`
   diff       lwr      upr    p adj
2-1    8  -6.429339 22.429339 0.1461079
3-1    1 -13.429339 15.429339 0.9158942
3-2   -7 -21.429339  7.429339 0.1826267

$`factor(C)`
   diff       lwr      upr    p adj
2-1   12  -2.429339 26.429339 0.0707078
3-1    3 -11.429339 17.429339 0.5481840
3-2   -9 -23.429339  5.429339 0.1191149
```

59.5.4 最佳水平组合均值的点估计

由前面的模型, 利用最小二乘法, 可以得到一般均值和每个因子每个水平的效应的估计

$$\hat{\mu} = \bar{y}$$

因子A第i水平的主效应a_i的估计为其均值减去整体均值(Ma[,2]为R的按照水平统计的均值结果, 见直观分析部分)

$$\hat{a}_i = Ma[,2][i] - \bar{y}$$

其它类似. 它们都是相应参数的无偏估计.

在本例中, 最佳组合水平为A_3BC_2, 故计算

$$\begin{aligned}\hat{\mu} &= \bar{y} = 50\\ \hat{a}_3 &= 61 - 50 = 11\\ \hat{c}_2 &= 57 - 50 = 7\end{aligned}$$

从而A_3C_2水平组合下指标的无偏估计为

$$\hat{\mu}_{3,.,2} = \hat{\mu} + \hat{a}_3 + \hat{c}_2 = 50 + 11 + 7 = 68$$

使用R计算如下

```
> mean(tr$y) # 总的均值
[1] 50

# 效应的计算
> model.tables(aov(y~factor(A)+factor(B)+factor(C),data=tr))
Tables of effects

 factor(A)
factor(A) # 因子A的3个水平的主效应
 1  2  3
-9 -2 11

 factor(B)
factor(B)  # 因子B的3个水平的主效应
 1  2  3
-3  5 -2

 factor(C)
factor(C)  # 因子C的3个水平的主效应
 1  2  3
-5  7 -2
```

59.5.5 最佳水平组合均值的区间估计

需要估计最佳组合A_3C_2的方差或标准差.

根据假设, $\hat{\mu}_{3,.,2}$是$y_1, \cdots, y_9$的线性组合, 故也是正态分布. 查看试验设计表, 得到

$$\begin{aligned}\hat{\mu}_{3,.,2} &= \hat{\mu} + \hat{a}_3 + \hat{c}_2 \\ &= \frac{1}{9}\sum_{i=1}^{9} y_i + \frac{1}{3}(y_7 + y_8 + y_9) + \frac{1}{3}(y_2 + y_4 + y_9)\end{aligned}$$

由于假设y的方差一致, 为σ^2(未知, 可以由残差估计), 那么根据变量和的方差的线性组合的公式

$$\begin{aligned}D(\hat{\mu}_{3,.,2}) &= [(\frac{1}{9})^2 * 9 + (\frac{1}{3})^2 * 3 + (\frac{1}{3})^2 * 3] * \sigma^2 \\ &= \frac{5}{9}\sigma^2\end{aligned}$$

这样得到最佳组合均值的方差与系统方差σ^2的关系.

下面估计σ^2. 可以使用残差来估计. 但是可以将不显著的因子的方差并入残差, 以提高误差的估计, 同时自由度也并入残差自由度. 这样根据方差分析结果

```
          Df Sum Sq Mean Sq F value  Pr(>F)
factor(A)   2    618     309 34.3333 0.02830 *
factor(B)   2    114      57  6.3333 0.13636
factor(C)   2    234     117 13.0000 0.07143 .
Residuals   2     18       9
```

得到

$$\begin{aligned}S'_e &= S_e + S_B = 18 + 114 = 132 \\ f'_e &= f_e + f_B = 2 + 2 = 4 \\ \hat{\sigma}^2 &= MS'_e = S'_e/f'_e = 132/4\end{aligned}$$

从而$\hat{\mu}_{3,.,2}$的置信区间为

$$\hat{\mu}_{3,.,2} \pm t_{1-\alpha/2,f'_e} * \sqrt{D(\hat{\mu}_{3,.,2})}$$

即

$$68 \pm t_{0.975,4} * \sqrt{\frac{5}{9} * 132/4} = 68 \pm 11.9 = (56.1, 79.9)$$

59.5.6 验证试验

实际中最佳水平组合不一定出现, 所以通常要进行验证性试验. 例如$A_3B_1C_2$,不在9次试验中, 是否真的符合需要? 例如对$A_3B_1C_2$做了3次重复, 得到结果: 62,68,71, 均衡为67, 看来不错.

59.6 贡献率分析

当指标不服从正态分布, 进行方差分析的依据就不足. 此时可以比较各因子的"贡献率"来衡量因子的作用大小.

由于因子的方差中除了因子效应外, 还包含了误差, 那么去除因子方差中的误差后称为因子的纯平方和, 即因子A的纯平方和为

$$S'_A = S_A - f_A \cdot MS_e$$

称因子A的纯平方和与总平方和的比为因子的贡献率ρ_A

$$\rho_A = \frac{S'_A}{S_T} = \frac{618 - 2 * 9}{984} = 60.97\%$$

类似可计算纯误差平方和为

$$S_e + f_A MS_e + f_B MS_e + f_C MS_C = f_T MS_e$$

具体计算如下

```
# 再次写出方差分析表
> s=summary(aov(y~factor(A)+factor(B)+factor(C),data=tr))
```

```
> s=s[[1]];s
          Df Sum Sq Mean Sq F value  Pr(>F)
factor(A)   2    618     309 34.3333 0.02830 *
factor(B)   2    114      57  6.3333 0.13636
factor(C)   2    234     117 13.0000 0.07143 .
Residuals   2     18       9
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> sp=(s[,2]-s[,1]*9) # 计算因子平方和
> sp
[1] 6.0000e+02 9.6000e+01 2.1600e+02 3.1974e-14
> sp[4]=sum(s[,2])-sum(sp) # 设置误差平方和
> sp
[1] 600  96 216  72
> cbind(s[,1:2],sp/sum(sp)) # 最后一列即为贡献率
          Df Sum Sq sp/sum(sp)
factor(A)   2    618   0.609756
factor(B)   2    114   0.097561
factor(C)   2    234   0.219512
Residuals   2     18   0.073171
```

从贡献率可知, 因子A最重要, 其水平变化引起的总数据波动占60.9%, 其次为C, 最后为B. 而且可以认为B不重要, 因为与误差差不多.

Chapter 60

正交设计: 有交互作用的正交设计

60.1 表头设计

多因子试验中, 有时候两个因子正交还存在交互作用. 与无交互作用的区别主要在于表头设计.

为提高某产品的收率, 需要试验.

60.1.1 确定试验因子和水平

1. 明确试验目的: 提高产品收率

2. 明确指标: 收率. 越高越好.

3. 确定因子和水平, 并确定可能存在的交互作用.

经过分析, 影响产品收率的因子有4个, 温度,时间, 两种原料配比, 真空度. 根据经验温度与时间的交互作用对除了有较大影响.

Table 60.1: 4个因子的水平

因子	水平1	水平2
A:温度	60	70
B:时间	2.5	3.5
C:原料配比	1.1:1	1.2:1
D:真空度	50	60

60.1.2 自由度的确定

与正交表有关的自由度

- 表的自由度: 试验次数减1, 即$f_{table} = n - 1$, n为表的列数
- 列的自由度: 水平数减1, 即$f_{col} = q - 1$, q为列的水平数

例如, 正交表$L_8(2^7)$中, 表的自由度为$8 - 1 = 7$, 任意列的自由度为$2 - 1 = 1$

因子与交互作用的自由度

- 因子的自由度: 水平数减1
- 交互作用的自由度: 对应两个因子自由度乘积

例如, 二水平因子A与B的交互作用自由度为$f_{A*B} = f_A * f_B = 1 * 1 = 1$. 三水平因子A与B的交互作用自由度为$f_{A*B} = f_A * f_B = 2 * 2 = 4$.

60.1.3 表的选择

1. 因子的自由度应该等于所在列的自由度

2. 交互作用的自由度应该等于所在列的自由度, 或其之和

3. 所有因子与交互作用自由度的和不能超过正交表的自由度

本例中, 考察的水平都是2, 可以用2水平正交表. 由于有4个因子和1个交互作用, 自由度之和为5, 应该$5 <= n-1$, 即$n >= 6$, 故选择$L_8(2^7)$. 一个因子占一列, 交互作用也占一列.

60.1.4 表头设计

表头设计的时候需要使用正交表的交互作用表.(请参考一般试验设计教科书的附录正交表)

应该先把存在交互作用的因子放到表头上, 可以放在任意两列, 例如放在1,2列. 查表得其交互作用列为第三列, 在第三列上标记$A*B$, 再将其它因子放在其它空白列上, 例如C,D放在4,7列. 表头设计如下

Table 60.2: 4个因子的水平

表头设计	A	B	A*B	C			D
列号	1	2	3	4	5	6	7

60.1.5 列出试验计划

只要将正交表中的因子1,2改为因子的实际水平即可.

按照随机化次序进行试验, 并记录试验结果.

60.1.6 试验结果

```
t1=gl(2,4)
t2=factor(c(1,1,2,2,1,1,2,2))
t3=factor(c(1,1,2,2,2,2,1,1))
t4=factor(c(1,2,1,2,1,2,1,2))
t5=factor(c(1,2,1,2,2,1,2,1))
t6=factor(c(1,2,2,1,1,2,2,1))
t7=factor(c(1,2,2,1,2,1,1,2))
y=c(86,95,91,94,91,96,83,88)
```

```
out=data.frame(A=t1,B=t2,AB=t3,C=t4,t5,t6,D=t7,y=y)
```

```
> out
  A B AB C t5 t6 D  y
1 1 1  1 1  1  1 1 86
2 1 1  1 2  2  2 2 95
3 1 2  2 1  1  2 2 91
4 1 2  2 2  2  1 1 94
5 2 1  2 1  2  1 2 91
6 2 1  2 2  1  2 1 96
7 2 2  1 1  2  2 1 83
8 2 2  1 2  1  1 2 88
```

60.2 方差分析

60.2.1 统计模型

$$y_{ijkl} = \mu + a_i + b_j + c_k + d_l + (ab)_{ij} + \xi_{ijkl}$$

其中各因子的主效应满足

$$a_1 + a_2 = 0 \quad b_1 + b_2 = 0 \quad c_1 + c_2 = 0 \quad d_1 + d_2 = 0$$

交互作用的效应满足

$$\sum_{j=1}^{2}(ab)_{ij} = 0 \quad i = 1, 2 \qquad \sum_{i=1}^{2}(ab)_{ij} = 0 \quad j = 1, 2$$

各ξ独立同分布且

$$\xi \sim N(0, \sigma^2)$$

60.2.2 平方和分解

略. 参考文献[16] 4.3.2 节 Page 142-143,

60.2.3 方差分析结果

交互作用的分析可以由正交表的交互作用列确定其方差, 也可以由R自行判断其交互作用. 使用R自行判断时, 例如, 判断因子A与B的交互作用, 公式为$A*B$. 详细见R的统计模型描述25.

```
y~A+B+A*B
```

```
# 所有列的方差
> summary(aov(y~.,data=out))
            Df Sum Sq Mean Sq
A            1    8.0     8.0
B            1   18.0    18.0
AB           1   50.0    50.0
C            1   60.5    60.5
t5           1    0.5     0.5
t6           1    4.5     4.5
D            1    4.5     4.5

# 将t5,t6方差作为误差, 获得F值的估计。显式指明交互作用列
> summary(aov(y~A+B+AB+C+D,data=out))
            Df Sum Sq Mean Sq F value  Pr(>F)
A            1    8.0     8.0     3.2 0.21554
B            1   18.0    18.0     7.2 0.11535
AB           1   50.0    50.0    20.0 0.04654 *
C            1   60.5    60.5    24.2 0.03893 *
D            1    4.5     4.5     1.8 0.31175
Residuals    2    5.0     2.5
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

# R自行判断交互作用的方差
```

```
> summary(aov(y~A+B+A*B+C+D,data=out))
          Df Sum Sq Mean Sq F value  Pr(>F)
A          1    8.0     8.0     3.2 0.21554
B          1   18.0    18.0     7.2 0.11535
C          1   60.5    60.5    24.2 0.03893 *
D          1    4.5     4.5     1.8 0.31175
A:B        1   50.0    50.0    20.0 0.04654 *
Residuals  2    5.0     2.5
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

从方差分析表中得到, 因子C与交互作用AB对指标有显著影响.

60.2.4 最佳水平组合的选择

1. 对于显著的因子, 通过比较两个水平下的数据的均值或数据的和可以得到, 因子C取2水平比较好

```
# 看到因子C的水平2均值高, 故选择因子C的2水平
> aggregate(out$y,by=list(out$C),FUN="mean")
  Group.1    x
1       1 87.75
2       2 93.25
```

2. 对于显著的交互作用, 先计算两个因子水平的所有组合下数据的均值, 再通过比较得到哪种水平组合比较好.

```
> aggregate(out$y,by=list(out$A,out$B),FUN="mean")
  Group.1 Group.2    x
1       1       1 90.5
2       2       1 93.5
3       1       2 92.5
4       2       2 85.5
```

看到A的2水平和B的1水平组合的均值最高. 故选择A_2B_1搭配.

在交互作用显著时, 不论因子是否显著, 都只要从诸A与B的搭配中选择最好的组合即可.

3. D不显著, 可以任意选择, 或根据实际需要. 真空度60可以节省时间.

60.3 指标均值的估计

60.3.1 点估计

总均值$\hat{\mu}$为90.5

```
> mean(out$y)
[1] 90.5
```

现在要估计最佳水平组合$A_2B_1C_2$的均值. 由于A与B不显著, 则可以认为$a_i = 0, b_j = 0$.

$$\mu_{212} = \mu + c_2 + (ab)_{21} = Mean_{c2} + Mean_{(ab)_{21}} - \mu$$

其中c_2为93.25

```
# 计算c_2
> aggregate(out$y,by=list(out$C),FUN="mean")
  Group.1    x
1       1 87.75
2       2 93.25
```

$(ab)_{21}$为93.5

```
> aggregate(out$y,by=list(out$A,out$B),FUN="mean")
  Group.1 Group.2    x
1       1       1 90.5
2       2       1 93.5
3       1       2 92.5
4       2       2 85.5
```

最佳水平组合$A_2B_1C_2$的均值结果为[1]

$$\mu_{212} = 93.5 + 93.25 - 90.5 = 96.25$$

60.3.2 TODO:区间估计

略. 可以参考前面一章的区间估计.

60.4 避免混杂-表头设计的一个原则

若一列出现2个因子, 或2个交互作用, 或一个因子与一个交互作用. 称为混杂.

当混杂的列显著时, 难以区别哪个因子(或交互作用)是显著的. 所以在表头设计的时候尽量避免混杂. 这是一个重要原则.

现在正交表的时候必须满足: 所有因子与交互作用的自由度之和<= $n-1$. n是正交表的行数. 不过在存在交互作用的场合, 这一条件满足的时候不一定可以用来安排试验. 所以这是一个必要条件.

60.4.1 两个例子

例1 A,B,C,D为2水平因子, 且要考察交互作用$A*B, A*C$, 请设计表头.

自由度为$df = f_A + f_B + f_C + f_D + f_{AB} + f_{AC} = 6$根据条件选择正交表$L_8(2^7)$, 表头如下

表头设计	A	B	A*B	C	A*C	D	
列号	1	2	3	4	5	6	7

[1]参考文献[16]Page 145 结果有误, 其结果为96.75

例1 A,B,C,D为2水平因子, 且要考察交互作用$A*B, C*D$, 请设计表头.

正交表$L_8(2^7)$无法安排. 因为无论4个因子放在哪一列, 两个交互作用或一个因子与一个交互作用总会共用一列. 譬如

表头设计	A	B	A*B,C*D	C			D
列号	1	2	3	4	5	6	7

60.4.2 正交表的交互作用

正交表的列是分组的. 对等水平的完全正交表$L_n(q^k)$来讲, 如果$n = q^k$, 那么全部列被分为k组. 各组的列数分别为$q^0, q^1, q^2, \cdots, q^{k-1}$. 譬如$L_8(2^7)$列被分为3组

- 第一组: 第1列
- 第二组: 2,3列
- 第三组: 4,5,6,7列

正交表的有交互作用的两列如果不在同一组, 那么其交互作用列必在组别高的组中. 如果在同一组, 交互作用必在低组别的组中. 譬如

- 因子A与B分别在1,2列时, 第1列在第一组, 第2列在第二组, 那么交互作用在第二组, 查表为第3列.
- 因子A与B分别在1,4列时, 第1列在第一组, 第4列在第三组, 那么交互作用在第三组, 查表为第5列.
- 因子A与B分别在2,3列时, 都在第二组, 那么交互作用在第一组, 查表为第1列.
- 因子A与B分别在4,7列时, 都在第三组, 那么交互作用在第一组或第二组, 查表为第3列, 在第二组.

表头设计	A	B	A*B	C			D				C*D			
列号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

出现混杂的时候, 只要选择较大的正交表就可以避免了. 譬如选择$L_{16}(2^15)$.

在表上有多个空白列, 为避免可能存在的交互作用, 可以首先将因子放在各组的第一列(也称为基本列, 第1,2,4,8列)

60.4.3 列排满的处理方法

当考察的因子与交互作用自由度之和$=n-1$, 表的各列排满了, 称为饱和设计. 此时处理方法有

- 重复试验后进行方差分析
- 改用较大的正交表, 补充做一些试验
- 将平方和较小的列看作误差列
- 作为饱和设计进行分析

60.5 有重复试验情况下的数据分析

参考文献 [16] 4.4

60.5.1 因子水平与表头设计

重复指在同一水平组合下进行若干次试验. 这种情况下, 试验设计并没有变化, 但数据分析有一些变化. 下面通过一个例子来说明.

例 4.4.1 某工厂为提高零件研磨工艺进行工艺参数的选优. 考察孔的锥度值. 越小越好.

Table 60.3: 因子水平表

因子	水平1	水平2
A:研孔设备	通用夹具	专用夹具
B:生铁研圈材质	特殊铸铁	一般铸铁
C:留研量(mm)	0.01	0.015

用正交表$L_8(2^7)$, 表头设计如下

Table 60.4: 4个因子的水平

表头设计	A	B		C			D
列号	1	2	3	4	5	6	7

60.5.2 试验结果

每一水平下加工了4个零件. 测量其锥度. 数据如下

```
# 试验结果
y<-c(1.5,1.7,1.3,1.5,
   1,1.2,1,1,
   2.5,2.2,3.2,2.0,
   2.5,2.5,1.5,2.8,
   1.5,1.8,1.7,1.5,
   1.0,2.5,1.3,1.5,
   1.8,1.5,1.8,2.2,
   1.9,2.6,2.3,2.0)

r=gl(8,1,32)# 重复

# 正交表排列
t1=gl(2,4)
t2=factor(c(1,1,2,2,1,1,2,2))
t3=factor(c(1,1,2,2,2,2,1,1))
t4=factor(c(1,2,1,2,1,2,1,2))
t5=factor(c(1,2,1,2,2,1,2,1))
```

```
t6=factor(c(1,2,2,1,1,2,2,1))
t7=factor(c(1,2,2,1,2,1,1,2))

tmp=data.frame(A=t1,B=t2,t3=t3,C=t4,t5,t6,t7=t7)

ff<-rbind(tmp,tmp,tmp,tmp) # 重复4次
out<-data.frame(ff,rep=r,y=c(t(matrix(y,nr=4))))

> tmp # 正交表
  A B t3 C t5 t6 D
1 1 1  1 1  1  1 1
2 1 1  1 2  2  2 2
3 1 2  2 1  1  2 2
4 1 2  2 2  2  1 1
5 2 1  2 1  2  1 2
6 2 1  2 2  1  2 1
7 2 2  1 1  2  2 1
8 2 2  1 2  1  1 2

> out # 试验结果
   A B t3 C t5 t6 D rep   y
1  1 1  1 1  1  1 1   1 1.5
2  1 1  1 2  2  2 2   2 1.0
3  1 2  2 1  1  2 2   3 2.5
4  1 2  2 2  2  1 1   4 2.5
5  2 1  2 1  2  1 2   5 1.5
6  2 1  2 2  1  2 1   6 1.0
7  2 2  1 1  2  2 1   7 1.8
8  2 2  1 2  1  1 2   8 1.9
9  1 1  1 1  1  1 1   1 1.7
10 1 1  1 2  2  2 2   2 1.2
11 1 2  2 1  1  2 2   3 2.2
12 1 2  2 2  2  1 1   4 2.5
13 2 1  2 1  2  1 2   5 1.8
14 2 1  2 2  1  2 1   6 2.5
15 2 2  1 1  2  2 1   7 1.5
16 2 2  1 2  1  1 2   8 2.6
17 1 1  1 1  1  1 1   1 1.3
18 1 1  1 2  2  2 2   2 1.0
19 1 2  2 1  1  2 2   3 3.2
20 1 2  2 2  2  1 1   4 1.5
```

```
21 2 1  2 1  2  1 2  5 1.7
22 2 1  2 2  1  2 1  6 1.3
23 2 2  1 1  2  2 1  7 1.8
24 2 2  1 2  1  1 2  8 2.3
25 1 1  1 1  1  1 1  1 1.5
26 1 1  1 2  2  2 2  2 1.0
27 1 2  2 1  1  2 2  3 2.0
28 1 2  2 2  2  1 1  4 2.8
29 2 1  2 1  2  1 2  5 1.5
30 2 1  2 2  1  2 1  6 1.5
31 2 2  1 1  2  2 1  7 2.2
32 2 2  1 2  1  1 2  8 2.0
```

60.5.3 方差分析

```
> summary(aov(y~.,data=out))
          Df Sum Sq Mean Sq F value   Pr(>F)
A          1 0.0078  0.0078  0.0495  0.82581
B          1 4.7278  4.7278 29.9584 1.26e-05 ***
t3         1 1.0153  1.0153  6.4337  0.01812 *
C          1 0.0378  0.0378  0.2396  0.62894
t5         1 0.4278  0.4278  2.7109  0.11270
t6         1 0.2628  0.2628  1.6653  0.20918
t7         1 0.0078  0.0078  0.0495  0.82581
Residuals  24 3.7875  0.1578
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

可以看到, 因子B和AB的交互作用(t3)是显著的. 合并其它不显著的项的平方和以提高误差估计的精度,

```
> summary(aov(y~B+t3,data=out))
          Df Sum Sq Mean Sq F value    Pr(>F)
B          1 4.7278  4.7278 30.2559 6.318e-06 ***
t3         1 1.0153  1.0153  6.4976   0.01635 *
Residuals  29 4.5316  0.1563
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

误差平方和由3.7875 变成 4.5316, 自由度也增加到29. 这样提高了F检验的精度.

60.5.4 最佳水平组合的选择

因子B和AB的交互作用(t3)是显著的. 只要从A和B的组合中选择锥度最小的就可以.

```
> aa=aggregate(out$y,by=list(out$A,out$B),FUN="mean")
> names(aa)<-c("A","B","mean")
> aa
  A B   mean
1 1 1 1.2750
2 2 1 1.6000
3 1 2 2.4000
4 2 2 2.0125
```

综合上面结果, 组合A_1B_1为最佳水平.

Chapter 61

正交设计: 水平数不等情况下的试验设计

61.1 混合水平正交表

参考文献附录中给出了若干混合水平正交表.

$L_16(2^15)$可以改造称为其它混合水平正交表, 例如$L_16(4*2^12), L_16(4^2*2^9)$等等.

改造方法: 略. 并列法.

61.2 直接选用混合水平正交表

例 4.5.1 某种化油器的设计中, 希望寻找一种结构, 在不同天气条件下具有较小的比油耗.

61.2.1 因子水平和表头设计

其中一个是2水平, 4个是3水平. 选择混合正交表$L_18(2*3^7)$, 表

Table 61.1: 因子水平表

因子	水平1	水平2	水平3
A:大喉管直径	32	34	36
B:中喉管直径	22	21	20
C:环行小喉管直径	10	9	8
D:空气量孔直径	1.2	1.0	0.8
E:天气	高气压	低气压	

头设计如下

表头设计	E		A	B	C	D		
列号	1	2	3	4	5	6	7	8

61.2.2 TODO: 数据分析

类似前面的数据分析. 略.

61.3 TODO: 拟水平法

当使用q水平正交表安排试验时, 如果存在水平数小于q的因子时可以采用拟水平法. 常用的是三水平正交表中安排少量二水平因子.

由于三水平列中有3个数字, 因子A只有2个水平. 对A可以虚拟一个水平, 即第三个水平为原来的2个水平的任意一个. 从而二水平因子的两个水平参与的试验次数不等, 现在的试验缺乏正交性.

分析: 略

61.4 TODO: 组合法

如果在一个试验中采用q水平正交表安排试验, 考察的因子有水平数小于q的两个因子, 且这两个因子无交互作用, 它们的自由度之和有恰好是$q-1$, 那么可以采用组合法.

略.

61.5 赋闲列法

赋闲列法是在一张q水平正交表上安排若干水平数不等的因子的一一种方法. 这里仅介绍正交表中安排若干2水平与3水平因子的方法

由于3水平因子自由度为2, 故在2水平正交表中应占2列. 可以任取2列安排3水平因子A. 但是这2列的交互作用列不能安排新因子或交互作用, 也不能用来计算误差.

表头设计的时候通常取第一列为赋闲列, 再取2列, 只要它们的交互作用列是第一列即可.

如果有几个3水平因子, 为了节约试验次数, 可将赋闲列都置于一列, 譬如, 2个3水平因子, 可取2,3列安排一个, 4,5列再安排一个, 其交互作用列都是第一列.

表头设计	赋闲	A	B	C		
列号	1	2,3	4	5	6	7

分析:略

Chapter 62

TODO

裂区法

饱和设计与超饱和设计

参数设计：稳健设计，灵敏度设计

回归设计

其他试验试验设计方法介绍：均匀设计，混料设计，全因子试验数据分析

Chapter 63

附: 正交表程序说明

作者: laomai

发布时间: 2007/12/09

http://blog.csdn.net/laomai

第一章 正交表的概念

§1-1 引子

在科研和生产实践中，人们往往要做许多次实验来进行某项研究。实验条件一般包括很多因素，当因素的值不同时，实验的结果也不一样。如果想把每个因素的每个值都要实验一遍，总实验数就等于各因素的值的个数的乘积，而这个数往往很大，超过了可接受的成本。

例如，假设某个实验由A,B,C,D四个因素，每个因素都有10个不同的取值，那么如果想把每个因素都考虑到，我们需要做 10*10*10*10=10000次实验。

为了减少实验数目，我们必须选出那些最有代表性的例子。于是，就要用到了正交表法(Orthogonal Array Testing Strategy)。

正交表是一种筛选实验用例的方法。在介绍其具体内容

前，我们先引入几个基本概念

⑴因素个数Factors，以后在本文中用F代替，因素就对应着正交表中的一列.

⑵水平数Levels,以后简写为L。他的含义就是每个因素可取值的个数，注意这里我们不关心每个具体的值是多少，关心的是其个数。

变量的具体取值我们称做水平值，在与水平数不发生混淆的情况下，简称水平，用变量名+编号表示。比如，一个因素A可能有三个水平，则可记为A1,A2,A3.

⑶强度Strength，以后简写为S:强度是构造正交表的一个最重要的指标，具体含义我们会在后面详细解释，这里只简单的说一下,正交表的核心性质就是后S个因素的每个水平值要相互碰一次且只碰一次。

⑷次数（Runs）：最后生成的正交表的记录行数，一行记录也就是一次实验。

⑸正交表的符号表示:先以字母L打头，下标r表示记录数，

括号中为具有相同水平数的因子数的项的连乘记。

Lr(Levelsfactors×Levelsfactors×...×Levelsfactors)

给几个具体例子，

⑴设有3个因素A,B,C,每个因素的水平数均为3时，生成的正交表为L27(33)(取强度等于3的情况)，记录数27=3*3*3

⑵设有5因素,每个因素的水平数分别2,2,2,3,3时，不同强度S的生成正交表分别为

s=2时，结果为L9(23×32)，记录数为最后两个变量的水平数乘积3*3=9

s=3时，结果为L18(23×32)，记录数为最后三个变量的水平数乘积2*3*3=18

§1-2 手工构造正交表

看一个具体例子：设有4个变量A,B,C,D，前三个变量的水平数为3,最后一个变量的水平为4，那么，根据不同的强度，可得到不同的正交表。

强度s=2时，首先得到基本正交表L12(33×4)

记录号 A B C D

1：1 1 1 1

2：1 2 2 2

3：1 3 3 3

4：2 1 2 3

5：2 2 1 4

6：3 1 3 2

7：3 3 2 1

8：0 2 3 1

9：0 3 1 2

10：0 0 1 3

11：0 0 2 4

12：0 0 3 4

可以看出C的每个水平值与D的每个水平值各碰一次且仅碰一次。而A、B的每个水平出现的次数也很均匀。并且任何在相同位置的两列组成的有序数对没有重复值。

为了保持取值的均匀性，用因素的水平值循环填充为0的项，得到最终的正交表为

A B C D

1 : 1 1 1 1

2 : 1 2 2 2

3 : 1 3 3 3

4 : 2 1 2 3

5 : 2 2 1 4

6 : 3 1 3 2

7 : 3 3 2 1

8 : 1 2 3 1

9 : 2 3 1 2

10 : 3 1 1 3

11 : 1 2 2 4

12 : 2 3 3 4

红色数字就是我们用水平值循环填充后的结果。

类似的我们可以得到强度为3的正交表L36(33×4),

记录数为4*3*3=36.

A B C D

1 : 1 1 1 1

2 : 1 1 2 2

3 : 1 1 3 3

4 : 1 2 1 2

5 : 1 2 2 1

6 : 1 2 3 4

7 : 1 3 1 3

8 : 1 3 2 4

9 : 1 3 3 1

10 : 2 1 1 2

11 : 2 1 2 1

12 : 2 1 3 4

13 : 2 2 1 1

14 : 2 2 2 2

15 : 2 2 3 3

16 : 2 3 1 4

17 : 2 3 2 3

18 : 2 3 3 2

19 : 3 1 1 3

20 : 3 1 2 4

21 : 3 1 3 1

22 : 3 2 1 4

23 : 3 2 2 3

24 : 3 2 3 2

25 : 3 3 1 1

26：3 3 2 2

27：3 3 3 3

28：0 1 1 4

29：0 1 2 3

30：0 1 3 2

31：0 2 1 3

32：0 2 2 4

33：0 2 3 1

34：0 3 1 2

35：0 3 2 1

36：0 3 3 4

读者可自行填充其中的0项，得到最后的结果

1-3 正交表的基本数学性质

设正交表的强度为S, 则正交表有以下数学性质

1、正交性：正交性有两个含义

⑴在最后S列中，每列因素的一个水平值与其他列中的每个水平值相碰一次且只碰一次。换句话说，最后S列构成的子表是满的。因此，正交表的记录个数=最后S列的水平数的乘积。

⑵在相同位置的任意S列，其构成的S维有序数对的集合中没有重复元素。特别的，当因素的水平数都相等时，对每个由S 列构成的集合N, 这个集合N将S维空间(既该S列的笛卡儿乘积)中的每个点都遍历一次且只遍历一次，形象的说，就是"既不重又不漏",。而因素不等的情况时，则最后S 列的集合必然是满的。也就是⑴中所说的情况。

由此看出强度的作用，强度就象一个筛子，筛选出解空间中符合S维正交性原则的所有记录，当S=1时，只遍历最后一个变量的所有取值。当S=因素个数时，得到的便是整个解空间。

2、均匀性：每个因素的水平值在表中的出现的次数是均匀的，对最后的S列中的每一列（他们必然是满的），每个水平值出现的次数相等。

Chapter 64

附: 统计咨询工作者被经常问及的三十个问题及解答

来自 http://cclab.caas.ac.cn/jrepository/articletext.jsp?id=417

64.1 试验设计

1) 在田间试验中应该使用多少次重复？ 它主要依赖于因子的数量，但又是可以从在特定研究领域中以往的试验结果和经验进行推算的。所选择的α水平（在处理间差异并不真实存在的情况下，差异发生的可能性）和要求的精度都是十分重要的。如果条件不允许设置足够重复的话，就要考虑放弃试验的可能性。在许多情况下，为了获得满意的结果需设置较多重复时，我们可以设置三次或四次重复。

2) 田间试验的小区应是多大？ 主要根据试验的目的和其它影响因素的数量，必须依章、务实地考虑问题。另外也应知道，对于土地有限的地区来说，较小的小区就意味着有更多的重复。通过设置更多的重复可以弥补小区面积不足的问题。另外，也可以用其它的方法来提高试验的

精确性。

3) 田间试验中的小区和区组的形状应是什么样？

应尽量设置正方形区组和长方形小区，目标是相对的。这些规则允许出现例外。

4) 某人要进行因子试验，在什么条件下他可以使用因子的随机完全区组设计？在什么时候他可以用裂区设计？

析因设置处理的随机完全区组设计更好些，除非存在机械性限制，即一个因素要求的小区面积不同于另一个因素。裂区设计常常对全部小区因素的检验缺乏能力。

5) 什么时候可以使用不完全区组设计？ 在由于处理数很大而使区组过大的试验中和想在大区组内对误差更好地控制时，就可以用不完全区组设计。另外，当自然地段小于区组的一个完全重复时，也可以使用这种设计。

6) 在什么条件下设计试验中可以使用条块？
当大型的设备或其它操作要求有较大的小区，条块会使机械操作更方便。分条应在每个区组内独立完成不要跨越区组。

7) 对于一个给定的试验环境，有多少种设计方法可以选择？ 一般来说，由于很多限制因素的存在，对于一种环境通常只有一种或两种合理的设计。通常这是排除那些不合理设计的过程。

8) 试验设计与处理设计有什么不同？ 试验设计是指处理对小区的随机化过程。处理设计是指处理的结构（即它们之间的关系如何）。

9) 在农业试验中，如何减少变异的影响？
A）选择均质的试验单元；B）改进试验技术；C）区组设置；D）设置更多的重复；E）避免过失误差（试验管理的）；F）计算协变量，然后用在协方差的分析中。

64.2 分析

10) 方差分析在什么时候需要数据转换？ 如果方差分析的假设条件不能满足，就应对数据进行转换。否则就不用转换。注意，有可能会校正过度或对假设条件产生新的破坏。

11) 平均值可以转换吗？方差需要转换吗？
在某些转换中可以这样做，但转换后的平均数经常不用来做比较。（可以用转换后的标准误差来对转换后的平均数进行比较）。不可以。必须保证方差的转换尺度。

12) 农业试验中的显著性检验或置信限应该使用什么样概率水平？ 常用0.01和0.05两种。0.05对于农业试验来说似乎更好些，因为在我们采用更有潜力的新方法时不应过分保守。报告实际的概率水平（例如 0.035）比只基于给定概率水平下的显著性检验结果来回答是和否要好得多，这可以让读者自己决定在一种特定情形下什么样的概率水平是可以接受的。

13) 在分析数据中处理缺值的最好方法是什么？
A）完全最小二乘法，其处理均方是无偏的。B）利用协方差估计缺值。C）利用Yates 近似法（1933）估计缺值，然后把它插入数据中并加以分析。然后从误差自由度中减去缺区估计的个数。这种方法的缺点在于处理平方和偏差略有提高。

14) 如果某人做了一系列普通设计的试验，用什么方法综合试验的结果？ 首先进行单个分析，然后进行方差的综合分析，要保证在综合之前误差方差是非异质的。如果处理结构允许，可以试着在综合分析时对处理平方和分组，然后再用这些分组与位点（或年份）交互，得到交互作用的分组。需要模型做一些假设，以确定适当的检测模式。初步显著性检验有利于确定为了获得一个试验误差的稳定估计值，合并是否是一种合理的选择。

15) 用随机区组设计来设计试验，分析时却如同是用完全随机设计的，这种情况合理吗？

不合理。应该使用针对于实际应用的试验设计的分析方法

来分析数据。

16) 在方差分析中，合并不同来源的变异以获得误差估计值，这合理吗？ 合并均质性的方差似乎是合理的。

17) 人们怎么知道是否一套数据需要转换？如果需要，用哪种方法？ 在转换前后可以测定残差，检查变异系数，对不同部分数据方差的差异进行检验。对二项式数据的反正弦转换有理论方差，因此，在转换时可以把方差分析中的误差与理论方差进行比较。在转换前后我们也可以利用Tukey非加性测验（1949）来检查加和性假设条件。

18) 什么是合理的变异系数（CV）？ 这依赖于试验的类型。10%对许多试验来说是比较合适的变异系数（CV）。虫害和病害的研究中经常有较高的CV（20% 25%）。在发展中国家，如果土壤肥力和土壤管理试验的变异系数能保持在15%以下，就被认为是可以接受的。

19) 如果外部因素对试验处理的影响存在差异，如何进行分析？ 如果在Y和X（X是协变量）之间存在合理的关系，为了提高精确度和调整协变量中的处理平均值方差，我们就可以考虑使用协方差分析。利用协方差分析就可以确定X与Y之间关系的紧密程度。应该坚持协方差的有关假设条件，它们是：A）协变量是固定的，测量中不存在误差而且独立于处理；B）在排除区组和处理差异后，Y对X的回归是线性的并与处理和区组无关；C）误差是独立正态分布，具有零均值和普通方差。

20) 使用多重比较如LSD、Duncan's新复极差检验等，会有什么问题？ 这些方法经常被滥用和误用。如果能确定处理结构，最好作一系列比较。如果不存在这样的处理结构而且必须要使用多重比较方法的话，一般来说，应该有一种多重比较方法最适合该种情况。例如，就可以用Dunnett's 法（1955）来比较对照和各处理，比如在系统法温室试验中把最优值的平均值与其它处理的相比较（如最优值对opt-P处理等）。没有"最好的办法"。但对于肥料试验，用Duncan新复极差测验来检验养分水平的增量却不合适。

21) 为什么在环境群体是非均质的时候，在一年进行的单个试验不足以得出概括性结果？ 因为它们是

近似值又是暂时性意义的，而且在以后所做的相似试验可能会由于环境的影响而得到相反的趋势。我们通常可以通过增加重复数或提高试验的精确度来改变试验的说服力。

22) 什么是空间分析？它是如何进入田间试验数据分析的？ 最近邻分析法是一个例子。我们可以用邻区的残差作为协方差分析中的协变量。利用这项技术通常可以显著提高精确性。来自空间分析的结果看上去很有希望。但在一般应用之前需要进行更多的检验。

23) 生物学显著性与统计学显著性的差异是什么？
因为试验的误差往往很大，具有生物学差异显著性在统计学上却不一定就显著。另一方面，在统计学上有显著性差异，而在生物学上却未必有什么意义。我们可以靠增加重复数或改进试验技术来改变产生统计学显著差异的概率，但却不可能改变生物学显著性。

24) 单尾检验与双尾检验的使用条件有何差异？
除非对一个方向上的差异感兴趣或有期望，否则，就应使用双尾检验。这种期望应基于一定的理论或过去对现象的经验。适当的时候也可以用单尾检验，因为它们更有力。

25) 我们为什么不为每个处理计算误差而为所有处理计算合并误差估计量？ 这不是一个好想法。因为这些估计值都是基于很小的自由度，因而都十分不稳定。如果方差是均质的话，最好是利用整个试验的数据计算出一个单独的误差方差。对于上面的规则来说存在许多例外。例如，方差随试验中某一因素（如时间）水平的变化而变化，为了研究误差模式而计算该因素每一水平的标准误就会很有帮助。

64.3 取样

26) 在样本调查时，什么因素决定了样本规模？
所需样本的大小依赖于因素的多少，如取样单元中的内在变异、估计平均值的精度和所使用误差率等。在许多时候

资金问题是决定样本的大小的上限。

27) 如果在一个小区内进行取样，取多少样本才能代表整个小区？　　在每个小区中的采样数量决定于：可以处理多少植物材料、小区内植物变异的水平、是否要求对这种变异进行估计、涉及的成本等。Gomez和Gomez（1984）在他们的一本书中给出了一个公式，用来估计主要基于方差考虑的每个小区的植株数量。

28) 对于特定的作物，什么是最好的小区取样过程？
　随机开始（即第一棵植株的选择是随机的），在行上采取系统样本（如每10株采一株）。

29) 为什么我们严格要求科学取样？　　科学的样本选择可以确保能利用独立的、正态的推断过程而不会怀疑它们的可用性。

30) 在确定样本大小时应该考虑取样成本吗？
成本问题很重要。在确定样本大小的时候应该认真考虑这个问题。一般来说，确定取样过程的每一个步骤的成本是可能的，虽然这些估计值并非完全准确，但这样估计的样本数会比凭空猜想的要精确得多。

参考文献

1. Dunnett, C.W. 1955. A multiple comparisons procedure for comparing several treatments with a control. J.Amer. Statist. Ass. 50: 1096-1121.

2. Gomez, K.A. and A.A. Gomez. 1984. Statistical procedures for agricultural research, 2nd Ed. Wiley, New York.

3. Tukey, J.W. 1949. One degree of freedom for non-additivity. Biometrics 5: 232-242.

4. Yates, F. 1933. The analysis of replicated experiments when the field results are incomplete. Empire J. Exper. Agr. 1: 129-142.

Part VIII

流行病学

“流行病学”参考文献除了[14] 第13,14章的内容. 《Analysis of Epidemiological Data using R and Epicalc》也是主要的一个。

另外一个流行病学的包是 epiR, 也非常的好.

Chapter 65

基本概念

65.1 前瞻性研究

前瞻性研究 (perspective study): 在开始的时间点上没有疾病的一群人, 经过一段时间后, 其中某些人发生了疾病. 发生疾病的人可能与开始时受某个变量(一般称暴露变量)的影响有关. 前瞻性研究的总体常称为队列(cohort), 因此又称为队列研究(cohort study).

65.2 回顾性研究

回顾性研究(retrospective study): 在这个研究中, 共有两组人群: (1)一组在研究中有病(病例) (2)另一组在研究中没有疾病(对照). 研究者要寻找在过去某一时间段内两组人的某种卫生习惯是否有差异. 这种研究也常常称为病例-对照研究(case-control study).

65.3 现状研究

现状研究(cross-sectional study): 在某个时间点上, 询问研究总

体的所有成员, 请他们回答现在的疾病状况及他们现在或过去的暴露状况. 有时候也称为患病率研究(prevalence study). 因为它可以在某时刻即时的比较暴露与未暴露个体之间的患病率. 前瞻性研究中感兴趣的是发病率而不是患病率.

65.4 危险率差与比(RR)

令p_1, p_2分别为暴露受试者和非暴露受试者中有病的概率.

危险率差(risk difference)定义为: $p_1 - p_2$. 也称为 attributable risk(AR)

危险比(或相对危险度 risk ratio)(relative risk)定义为 p_1/p_2.

65.5 优势及优势比(OR)

危险率比 RR 受分母的影响太大. 为避免这一限制, 使用另外一个比例的测度称为优势比(odds ratio, OR).

优势: 如果一个事件成功发生的概率为p, 则有利于成功发生的优势为$p/(1-p)$.

优势比: 记p_1, p_2为两组中成功发生的概率. 则优势比定义为

$$OR = \frac{p_1/q_1}{p_2/q_2} = \frac{p_1 q_2}{p_2 q_1}$$

65.6 优效性研究与等效性研究

建立在空白对照上的的无效假设为: 两个处理有相同的效应; 对两个处理的效应彼此不同. 临床研究常常是这种形式, 称为优效性研究(superiority study).

有些人认为建立在空白对照上的优效性研究常常是为了考察一种处理的效率. 另外一些人认为如果标准方法已经被证明有效, 则对病人不给治疗的做法是不道德的. 例如对精神分裂症病人使用空白对照去估计某种新方法的疗效).

近年来提出了一种新的研究设计形式, 主要目标是研究两种方法是否等效而不是一种优于另一种. 这种研究称为等效研究(equivalence study).

65.7 筛选检验的一般性概念

设通过筛选检验的结果来假设检验为

$$H_0: \text{此人未患病} \quad vs. \quad H_1: \text{此人患病}$$

筛选检验的结果为两种: 阴性(-), 阳性(+)

65.7.1 预测值阳性/阴性

预测值阳性(predictive value positive. PV^+)是指一个人在该试验中呈阳性条件下患病的概率. 即

$$PV^+ = P(\text{疾病}|\text{检验}^+)$$

实际上是检验的功效, 即此检验判断此人患病而此人确实患病的概率.

预测值阴性(predictive value negative. PV^-)是指一个人在该试验中呈阴性条件下未患病的概率. 即

$$PV^- = P(\text{无疾病}|\text{检验}^-)$$

即此检验判断此人患病而此人确实患病的概率.

例1(预测值阳性/阴性): 设10000名乳房X射线照片检查后为阴性的妇女, 2年内发现有乳腺癌的是20例. 则预测值阴性

$$\begin{aligned} PV^- &= P(\text{无疾病}|\text{检验}^-) \\ &= 1 - P(\text{有疾病}|\text{检验}^-) \\ &= 1 - 0.0002 = 0.9998 \end{aligned}$$

10名X射线检验为阳性的妇女在此2年内发现乳腺癌1例, 则则预测值阳性

$$PV^+ = P(\text{疾病}|\text{检验}^+) = 1/10 = 0.1$$

就是说, 如果X射线阴性, 则几乎可以肯定2年内此妇女不会患乳腺癌($PV^- \approx 1$). 如果X射线阳性, 则2年内此妇女患乳腺癌的概率有10%($PV^+ = 0.1$).

某检验这两个值较高说明此检验有较高的价值. 实际上我们总是寻找$PV^+ = 1, PV^- = 1$的检验, 即只要检验阳性, 就可以判断患病, 只要检验阴性, 就可以判断不患病, 我们就可以准确的对每个病人做出判断.

65.7.2 灵敏度/特异度

一个症状(或一组症状, 或筛选检验)的灵敏度(sensitivity)是疾病发生后出现症状的概率.

一个症状(或一组症状, 或筛选检验)的特异度(specificity)是疾病不发生时不出现症状的概率.

下面表参考 http://www.poems.msu.edu/EBM/Diagnosis/SensSpec.htm

␣␣␣␣␣␣␣␣实际患病␣␣␣␣实际无病
检测阳性␣␣␣␣a␣␣␣␣␣␣␣␣␣␣b
检测阴性␣␣␣␣c␣␣␣␣␣␣␣␣␣␣d

sensitivity(灵敏度)␣=␣a␣/␣(a+c)
specificity(特异度)␣=␣d␣/␣(b+d)
␣␣␣␣␣␣␣␣␣␣␣␣

I型错误（假阴性）：发生疾病，但无症状（因此判断其没有疾病）

II型错误（假阳性）：疾病不发生，但有症状（因此判断其有疾病）

所以

灵敏度␣==␣1␣-␣I型错误␣␣==␣1-\alpha
特异度␣==␣1␣-␣II型错误␣==␣1-\beta␣␣==␣power
␣␣␣␣␣␣␣␣␣␣␣␣␣

例2(灵敏度和特异度): 假设肺癌中90%抽烟, 没有肺癌的30%抽烟. 此处疾病为肺癌, 症状为抽烟. 灵敏度为肺癌中抽烟的概率为0.9, 特异度为没有肺癌的不抽烟的概率0.7.

65.7.3 症状有效

预测疾病中某个症状是有效的, 指该症状的两个指标(灵敏度,特异度)都是高的.

65.7.4 假阴性/假阳性

某试验结果是阴性但实际上是阳性(即实际上此人患病)称为假阴性(false negative),

某试验结果是阳性但实际上是阴性(即实际上此人未患病)称为假阳性(false positive),

65.7.5 Bayes法则的应用

我们已经知道某疾病症状的灵敏度和特异度, 还知道此疾病的先验概率(此疾病总的发病率), 那么我们可以由Bayes法则求出某人出现此症状(试验阳性)时的患病概率(预测值阳性/阴性). (Bayes法则的描述见附录111.3)

假设自动血压计把有高血压的84%诊断为高血压, 正常的23%诊断为高血压, 已知成年人中20%为高血压. 那么此血压计的预测值阳性与预测值阴性是多少?

记A=症状, B=疾病. 那么

- 预测值阳性为$PV^+ = P(B|A)$

- 预测值阴性为$PV^- = P(\bar{B}|\bar{A})$
- 灵敏度为$P(A|B) = 0.84$
- 特异度为$P(\bar{A}|\bar{B} = 1 - 0.23 = 0.77$
- 疾病的先验概率$P(B) = 0.2$

那么由贝叶斯法则得

$$PV^+ = P(B|A) = 0.84 * 0.2/(0.84 * 0.2 + 0.23 * 0.8) = 0.48$$
$$PV^- = P(\bar{B}|\bar{A}) = 0.77 * 0.8/(0.77 * 0.8 + 0.16 * 0.2) = 0.95$$

从结果可以看到, 此血压计对于阴性结果的人有很高的预测能力(95%的把握保证此人无高血压), 但是对于阳性的人预测能力不足(48%的把握保证此人有高血压)

65.8 ROC曲线

ROCR包可以计算假阳性, 假阴性, ROC曲线, chi方, 优势比等.

BioConductor项目也有一个ROC包

65.8.1 定义

在某些情况下, 试验结果可能有多个等级而不是简单的阳性/阴性. 另外一些情况下, 试验结果可能是连续的. 在这种情况下, 判断阳性/阴性的切断点(cut-off-point)常常是任意的.

例如: 可能有神经系统疾病的109名受试者(是否有病早已知道)接受某放射学家的CT成像技术检验. 结果用等级(rating)表示. 如果把所有CT结果当做阳性(出现症状), 那么其灵敏度为 判断为不正常的人数/实际患病人数=51/51=1, 特异度为 判断正常的人数/实际正常人数=0/51=0.

如果把肯定正常当做阴性, 其它结果当做阳性(出现症状)结果, 则灵敏度为 出现症状的人数/实际患病人数=48/51=0.94, 特异度为 未出现症状的人数/实际正常人数=33/58=0.56.

Table 65.1: CT成像等级结果

CT结果	实际正常	实际不正常
肯定正常	33	3
可能正常	6	2
有问题	6	2
可能不正常	11	11
肯定不正常	2	33
总数	58	51

依次类推, 我们可以构造一个ROC(receiver operating characteristic)曲线, x轴为1-特异度, y轴为灵敏度作图, 不同的点对应不同的切断点识别阳性.

曲线下的面积是这个诊断方法的精度的合理指标. 实际上通过观察可以知道, 无论是灵敏度增大还是特异度增大, 曲线向上凸起的程度都增大. 而灵敏度和特异度都大表明检验方法比较好.

65.8.2 从数据直接计算

函数roc.from.table计算ROC并绘制曲线. 此例子曲线下面积为0.89, 意味着放射学家能够按照CT等级的相对顺序把一个正常人从不正常人中识别出的概率为89%[1].

```
library(epicalc)
t=cbind(c(33,6,6,11,2),c(3,2,2,11,33))
> roc.from.table(t)
$auc 曲线下面积
[1] 0.893171

$original.table
   实际正常    实际患病
 Non-diseased Diseased
         33        3
          6        2
```

[1]此说法可能证据不足–孙尚拱

```
          6       2
         11      11
          2      33

$diagnostic.table
      1-特异度  灵敏度
  1-Specificity Sensitivity
     1.00000000   1.0000000
>    0.43103448   0.9411765
>    0.32758621   0.9019608
>    0.22413793   0.8627451
>    0.03448276   0.6470588
>    0.00000000   0.0000000
# 其它的用法
> roc.from.table(t, title=TRUE, auc.coords=c(.4,.1), cex=1.2)
```

65.8.3 logistic回归的ROC曲线

使用函数lroc(), 参考多重logistic回归部分69.9

65.9 生存分析一般概念

65.9.1 (累加)发病率

在类型数据的统计分析中, 很多时候"人"是分析的单位. 前瞻性研究中, 在基线时间把个体分为暴露非暴露组, 比较两组一段时间内的发病的比例, 我们把这些比例称为发病率(incidence rate), 更确切的名称应该称为累加发病率. 累加发病率是一种比例, 以人为分析单位, 值在0,1之间. 计算中隐含所有人被跟踪相同时间. 但是常常不能满足.

65.9.2 发病密度

发病密度(incidence density, ID)定义为该组群中发病的人数除以研究过程中累加的人-时间(年)总数. 分母是人-年数, 值可以是$0, \infty$.

有时候发病密度用更常用的术语发病率(incidence rate)λ表示, 以区别于时间t内的累加发病率CI(t).

下面是一个例子([14] Page 648). 研究口服避孕药(OC)与乳腺癌的关系. 由护士研究课题所收集. 她们在1976年没有乳腺癌, 但OC的使用情况不同. 每两年调查一次, 最后累加使用或不使用OC的时间, 如何判断这些数据在乳腺癌发病率上的差异?

使用OC的情况	病例数	人-年数
现在使用者	9	2935
从不使用者	239	135130

65.9.3 累加发病率与发病密度的关系

为简单起见, 一段时间t内发病密度是不变的, 则由微积分可以证明

$$CI(t) = 1 - e^{-\lambda t}$$

其中, CI(t): 累加发病率. λ:发病密度.

如果累加发病率< 0.1, 则近似有$e^{-\lambda t} = 1 - \lambda t$, 则

$$CI(t) = 1 - (1 - \lambda t) = \lambda t$$

例如, 40-44岁绝经前妇女每100000人-年有200人患乳腺癌, 则40岁没有乳腺癌的妇女今后5年的累加发病率是多少? 此处$\lambda = 200/10^5, t = 5$

```
> lambda=200/10^5; lambda # 发病密度
[1] 0.002
```

```
> t=5
> CI5=1-exp(-lambda*t); CI5 # 累加发病率
[1] 0.009950166
> CI=lambda*t; CI # 近似的累加发病率
[1] 0.01
```

65.9.4 率比(RR)

类似于危险率的比(risk ratio, RR), 那里的单位是人, 我们也可以使用于人-时间数据两个发病率的比较. 记λ_1, λ_2分别是暴露和非暴露组的发病率, 称λ_1/λ_2为率比(rate ratio)

65.10 交叉设计

65.10.1 交叉设计(cross over design)

交叉设计(cross over design)是随机临床试验的一种形式. 每个受试者被随机指定为组1或组2, 在第一期处理内组1接受药物A, 组2接受药物B. 第二期相反处理, 即组1接受药物B, 组2接受药物A. 两期之间常常有一段洗脱时间, 以消除药物的残留效应.

若可以控制药物的残留效应, 那么交叉设计是值得考虑的, 例如象高血压这样的研究. 否则需要使用平行组设计. 大多数临床三期试验是长期的研究, 因此使用平行组设计.

65.10.2 洗脱期

洗脱期(washout period)是安排在两个药物处理期之间以消除药物的残留效应的一段时间.

65.10.3 残留效应(剩余效应)

药物的残留效应(carry over effect)指第一个处理期内的一个或多个药物会在第二期内有剩余的生物学效应.

65.11 常用的回归分析

流行病学中常用的回归为线性回归(固定模型和随机模型及其混合模型), logistic回归和 Poisson 回归. 应变量为二分变量的使用logistic回归, 应变量为单位时间(面积)的计数(自然数)的使用Poisson 回归. 详细见 "回归与方差分析".

Chapter 66

包与函数介绍

66.1 epicalc包

cs: 前瞻性和现状研究. 计算危险率, 危险率差, 危险率比.

cc: 计算回顾性研究的优势比和95%置信区间. 结果基于精确方法.

ci: 可以计算binomial(二项比例), poisson(累加发病率), numeric(均值) 的估计与置信区间.

mhor: 计算回顾研究(case-control study)分层数据的Mantel-Haenszel 优势比. 基于精确方法. 有时候与 mantelhaen.test 结果不太一样. matchTab: 匹配数据的优势比估计.

66.2 rateratio.test包

rateratio.test: 计算率比

66.3 epiR包

此包的函数很强.

epi.2by2: 计算2×2列联表计数数据的各种值。四种方法, method= cohort.count(前瞻性研究数据), cohort.time(人-时间数据), case.control(对照-病例研究数据), cross.sectional(现状研究数据) 分析四种数据. 根据数据类型不同结果不同. 若是多层数据, 返回每一层与联合的的OR, RR等对应结果, 和 Mantel-Haenszel 联合结果, 每一层和联合的卡方值(齐性检验).

epi.kappa: 计算 kappa 统计量, 与 mcnemar 检验(p-值)

epi.dsl: 随机效应的 DerSimonian-Laird meta-analysis

epi.mh: 固定效应的 Mantel-Haenszel meta-analysis

66.4 rmeta

meta.DSL: 随机效应的 DerSimonian-Laird meta-analysis

meta.MH(rmeta): 固定效应的 Mantel-Haenszel meta-analysis

66.5 stats包

prop.test: 二项比例检验

prop.trend.test: 二项比例趋势性检验

fisher.test: 精确二项比例检验(独立性检验), 与 cc 结果一样, 也是精确方法. 但是没有给出卡方值.

Chapter 67

类型(属性)数据的效应测度

我们要在暴露与非暴露的受试者之间比较疾病发生的频率. 这在前瞻性研究中比较的是发病率, 而在现状研究中比较的是患病率.

67.1 危险率差的估计

令p_1, p_2分别为暴露受试者和非暴露受试者中有病的概率.

危险率差(risk difference)定义为: $p_1 - p_2$.

危险比(或相对危险度 risk ratio)(relative risk)定义为 p_1/p_2.

设 $\hat{p_1}, \hat{p_2}$ 分别为暴露和未暴露受试者样本中有病的比例. 样本量分别为 n_1, n_2. 则 $p_1 - p_2$ 的无偏点估计为 $\hat{p_1} - \hat{p_2}$. 假设这个二项分布中正态分布的假定成立, 则可以使用正态分布理论近似求出置信区间的估计. 我们已知

$$\hat{p_1} \sim N(p_1, p_1 q_1/n_1)$$
$$\hat{p_2} \sim N(p_2, p_2 q_2/n_2)$$

因为是两个独立样本, 我们有

$$\hat{p_1} - \hat{p_2} \sim N(p_1 - p_2, \frac{p_1 q_1}{n_1} + \frac{p_2 q_2}{n_2}$$

正态分布假设成立时, 使用 $\hat{p_1}, \hat{p_2}$ 代替 p_1, p_2, 我们可以导出危险率差的点及区间估计. 无偏点估计为 $\hat{p_1} - \hat{p_2}$. 区间估计为

$$\hat{p_1} - \hat{p_2} - [1/(2n_1) + 1/(2n_2)] \pm z_{1-\alpha/2}\sqrt{\hat{p_1}\hat{q_1}/n_1 + \hat{p_2}\hat{q_2}/n_2}, \quad if \quad \hat{p_1} >= \hat{p_2}$$

$$\hat{p_1} - \hat{p_2} + [1/(2n_1) + 1/(2n_2)] \pm z_{1-\alpha/2}\sqrt{\hat{p_1}\hat{q_1}/n_1 + \hat{p_2}\hat{q_2}/n_2}, \quad if \quad \hat{p_1} <= \hat{p_2}$$

置信区间适用条件为 $n_1\hat{p_1}\hat{q_1} >= 5, n_2\hat{p_2}\hat{q_2} >= 5$.

(详细的推导见[14] Page 554-555)

下面是一个例子([14] Page 345, 表 10.2 Page 555, 解答). 研究口服避孕药(OC)对心脏病的影响. 5000名在开始服用OC的妇女, 3年后发展有心肌梗塞(MI)的有13人, 10000名未服用OC的妇女3年内有7例MI. 请估计服用及未服用OC的MI发病率的点估计及区间估计.

```
> x=matrix(c(9993,7,4983,13),nc=2,dimnames=list(c("No-MI","MI"),
   c("No-Expose","Expose")))
> x
     No-Expose Expose
No-MI     9993   4983
MI           7     13

> library(epicalc)
# 注: cs, cc等函数只有控制台输出, 无返回值
> cs(outcome=NULL, exposure=NULL, cctable=x,decimal=6)

            Exposure
Outcome       Non-exposed Exposed Total
  Non-diseased 9993        4987    14980
  Diseased     7           13      20
  Total        10000       5000    15000

              Rne         Re      Rt
  Risk         7e-04       0.0026  0.001333

                                      Estimate Lower95ci Upper95ci
 Risk difference (attributable risk)     0.0019   0.000661  0.003139
 Risk ratio                           3.714286 1.467171  9.403078
 Attr. frac. exp. -- (Re-Rne)/Re         0.730769
```

```
 Attr. frac. pop. -- (Rt-Rne)/Rt*100 %  47.5

# 手工计算, 结果与cs不一样. cs算法未知
> n1=5000
> n2=10000
> p1=13/n1
> q1=1-p1
> p2=7/10000
> q2=1-p2
> p1-p2
[1] 0.0019
> p1-p2-(1/(2*n1)+1/(2*n2))-qnorm(0.975)*sqrt(p1*q1/n1+p2*q2/n2)
[1] 0.0002463116
> p1-p2-(1/(2*n1)+1/(2*n2))+qnorm(0.975)*sqrt(p1*q1/n1+p2*q2/n2)
[1] 0.003253688
```

Rne, Re, Rt 分别为: 未暴露患病率, 暴露患病率, 总患病率. Risk difference 为危险率差及上下置信区间. Risk ratio 为危险率比.

67.2 危险率比(RR)的估计

危险率比(risk ratio, RR=p_1/p_2). 其对数$ln(RR)$的样本分布比RR本身更接正态分布. 详细算法参考 [14] Page 556. 参考文献的置信区间的结果为(1.5, 9.3), 与cs的结果稍稍不同. 用法见上面.

67.3 优势比(OR)的估计

危险率比 RR 受分母的影响太大. 为避免这一限制, 使用另外一个比例的测度称为优势比(odds ratio, OR).

优势: 如果一个事件成功发生的概率为p, 则有利于成功发生的优势为$p/(1-p)$.

优势比: 记p_1, p_2为两组中成功发生的概率. 则优势比定义为

$$OR = \frac{p_1/q_1}{p_2/q2} = \frac{p_1 q_2}{p_2 q_1}$$

对优势比点估计及区间估计的具体算法参考 [14] Page 562. epicalc 包中的 cc 函数执行优势比的估计. 其方法应该为 woolf 方法. (下面的例子来自 [14] Page 560 例 13.10. 数据表为 Page 344 表 10.1)对初娩年龄与子宫癌发病的统计. 优势比及其置信区间见下面. 结果与文献结果一致.

```
> x=matrix(c(8747,2537,1498,683),nc=2,
   dimnames=list(c("No-Cancer","Cancer"),c("<=29",">=30")))
> x
          <=29 >=30
No-Cancer 8747 1498
Cancer    2537  683

> fisher.test(x)

       Fisher's Exact Test for Count Data

data:  x
p-value < 2.2e-16
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 1.419073 1.740189
sample estimates:
odds ratio
  1.571925

> library(epicalc)
> cc(cctable=x,decimal=5)

            Exposure
Outcome       Non-exposed Exposed Total
  Non-diseased       8747    1498 10245
  Diseased           2537     683  3220
  Total             11284    2181 13465
```

```
OR =  1.57192
95% CI = 1.41907 1.74019
Chi-squared = 78.36984 ,  1 d.f. , P value = 0
Fisher's exact test (2-sided) P value = 0
```

67.4 优势比与危险率的比较

当样本中病例(暴露中病例+非暴露病例)比例f_1与非病例(暴露中非病例+非暴露非病例)比例f_2不同时, $\hat{RR}$不是RR的无偏估计, 而$\hat{OR}$是OR的无偏估计. 在病例-对照研究中, f_1, f_2几乎总是不同的, f_1 几乎总是要大于f_2. (证明见 [14] Page 559)

67.5 混杂与分层

混杂变量(confounding variable): 是一个与疾病和暴露变量都有关的变量.

分层(stratification): 在疾病暴露关系分析中, 把数据按照一个或多个潜在的混杂变量的水平分成若干组, 这称为分层. 这些小组称为"层".

正混杂(positive confounder): 如果该变量与疾病和暴露两者关系都是正向的, 或都是负向的. 这样的混杂称为正混杂.

负混杂(negative confounder): 如果该变量与疾病呈正关系而和暴露是负向的关系, 或相反. 这样的混杂称为负混杂.

例如: 我们考察酗酒与肺癌的关系, 可以得到一个2×2列联表. 在混杂变量中, 抽烟是其中一个. 按照抽烟与否, 可以把此列联表分为两个2×2列联表, 其中一个是抽烟的酗酒与肺癌的关系. 另外一个是不抽烟的酗酒与肺癌的关系. 抽烟的列联表中OR=1.0, 非抽烟的列联表中OR=1.0. 故控制抽烟后酗酒与肺癌无关系. 抽烟与肺癌和酗酒都是正向的关系, 是正混杂.

67.6 分层的类型数据统计推断方法-Mantel-Haenszel检验

67.6.1 Mantel-Haenszel检验及优势比估计

Mantel-Haenszel检验也是基于超几何分布, 与 Fisher 检验原理一样. 详细算法请参考 [14] Page 570.

目的是判断二态疾病和二态暴露变量在控制一个或多个混杂变量后的关联性.

哪一行或列被安排在第一是任意的. 即这个检验统计量及判断的显著性不受行列顺序的影响.

零假设: 疾病与暴露之间无联系.

67.6.2 公共优势比与效应修正

一般检验优势比是否齐性是重要的. 如果每一层的关联程度相同, 则可以给出公共优势比. 否则公共优势比没有意义, 应该给出各层单独的优势比.

假设考察疾病变量D和暴露变量E的关联性, 但是有混杂变量C. 于是我们按变量C把总体分成g层且计算每层的优势比. 若各层的真实的优势比不同, 我们认为在E与C之间存在交互作用(interaction)或效应修正(effect modification), 变量C称为效应修正因子(effect modifier). 即若C是效应修正因子, 则C的不同水平会有不同的疾病与暴露的关系.

67.6.3 例子

下面是一个例子 ([14] Page 569). 研究目的是看被动吸烟(passiveSmoke)对癌症(ill)危险率的影响. 此处被动吸烟是暴露变量, 其配偶每天至少1支且吸烟6个月以上. 混杂变量就是被动

吸烟者本人是否吸烟(smoke=yes, no). 因为本人是否吸烟与配偶是否吸烟和癌症都有关系的变量.

各层优势比齐性检验: 首先看 Homogeneity test, 卡方自由度为1, 值 = 3.254582, p-值=0.0712. 接受各层优势比是齐性的, 没有显著不同. 若不同, 则看 Var3 A Var3 B, 分别给出各层的优势比.

关联性检验: M-H Chi2(1)这一行. p-值=0.0001461 很小(也就是卡方值大. M-H Chi2(1)=14.42230, 自由度为1.), 说明控制本人是否吸烟后被动吸烟与癌症还是有高度显著的正相关联系. 公共优势比为 1.63.

```
> x=array(c(120,80,111,155,161,130,117,124),
   dim=c(2,2,2),
   dimnames=list(c("ill","control"),
      "passiveSmoke"=c("yes","no"),
      c("smoke=yes","smoke=no")))
> x
, ,  = smoke=yes

        passiveSmoke
         yes  no
  ill     120 111
  control  80 155

, ,  = smoke=no

        passiveSmoke
         yes  no
  ill     161 117
  control 130 124

> mantelhaen.test(x) # 使用连续修正

      Mantel-Haenszel chi-squared test with continuity correction

data:  x
Mantel-Haenszel X-squared = 13.9423, df = 1, p-value = 0.0001885
alternative hypothesis: true common odds ratio is not equal to 1
95 percent confidence interval:
```

```
 1.263955 2.090024
sample estimates:
common odds ratio
        1.625329

> mantelhaen.test(x,exact=T) # 精确计算

       Exact conditional test of independence in 2 x 2 x k tables

data:  x
S = 281, p-value = 0.0001665
alternative hypothesis: true common odds ratio is not equal to 1
95 percent confidence interval:
 1.254833 2.109589
sample estimates:
common odds ratio
        1.626181

> library(epicalc)
> mhor(mhtable=x,decimal=6)

Stratified analysis by  Var3
             OR lower lim. upper lim.  P value
Var3 A      2.09     1.418      3.10 0.000120
Var3 B      1.31     0.918      1.88 0.138248
M-H combined 1.63     1.264      2.09 0.000146

M-H Chi2(1) = 14.42230 , P value = 0.0001461
Homogeneity test, chi-squared 1 d.f. = 3.254582 , P value = 0.0712241
```

67.7 匹配研究中优势比的估计

匹配数据的McNemar检验与分层数据的Mantel-Haenszel检验密切相关. 匹配是分层的一种特例. 每个配对对应样本量为2的一个层. 可以证明, McNemar检验是层中样本量为2的Mantel-Haenszel检验的一个特例.

在成对的匹配中, 结局相同的对称为一致对(concordant pair). 结局不同的称为不一致对(discordant pair). 不一致对中, 使用A处理后有事件发生而B处理后未发生, 称为A型不一致对. 否则称为B型不一致对.

匹配数据 Mantel-Haenszel 检验中, 疾病与暴露关系的优势比为

$$OR = n_A/n_B$$

n_A为A型不一致对数. n_B为B型不一致对数.

匹配研究中$ln(OR)$近似服从正态分布, 方差为

$$Var[ln(OR)] = 1/(npq)$$

n为不一致对总数. p为A型不一致对的比例. q=1-p.

双侧 $100\% * (1-\alpha)$ 置信区间为 (e^{c_1}, e^{c_2}), 其中

$$c_1 = ln(OR) - z_{1-\alpha}\sqrt{\frac{1}{npq}}$$

$$c_1 = ln(OR) + z_{1-\alpha}\sqrt{\frac{1}{npq}}$$

$$n = n_A + n_B$$

$$p = n_A/n$$

下面是《生物统计学基础》10.4 Page 360 中的一个例子. 按年龄(或其它条件)配对621对病人, 配对的1人随机指定使用A方法治疗, 另外一人使用B方法治疗. 其中A方法生存5年以上, B方法也生存5年以上的有510对 ; A方法生存5年以上, B方法生存少于5年的有5对 ; A方法生存少于5年, B方法生存5年以上的有16对 ; A方法生存少于5年, B方法也少于5年的有90对. 检验A, B两种方法的差异是否显著.

此例中, 有 510+90=600 个一致对. 有 5+16=21 个不一致对. 一致对不提供信息, 故分析时抛弃之. 我们集中研究一致对.

由于 matchTab 函数只是针对原始数据, 没有针对列联表的方法. 当我们有列联表而没有原始数据的时候, 需要把数据仔

细的再还原一下. 设A方法为case, B方法为control. 生存大于5年为暴露, 小于5年为非暴露.

注意变量0,1的取值相反, 结果会不同. 区别在于一个是另一个的倒数.

结果与 [14] Page 577 一致.

```
> x
           B result
A result      more 5 years less 5 years
  more 5 years          90          16
  less 5 years           5         510

# 还原数据并使用 matchTab 函数
> a5=rep(1,106) # A 方法存活大于5年(暴露)
> a4=rep(0,515) # A 方法存活小于5年(非暴露)
> b5=a5
> b4=a4
> b5[91:106]=0 # A>5年的B有16个小于5年
> b4[1:5]=1 # A<5年的B有5个大于5年
> a=c(a5,a4) # A方法(case)的所有结果
> b=c(b5,b4) # B方法(control)的所有结果
> table(a,b)
   b
a    0   1
  0 510   5
  1  16  90

> caseControl=c(rep(0,621),rep(1,621))
> expose=c(a,b)
> pair=c(1:621,1:621) # 配对
> matchTab(caseControl,expose,pair)

Exposure status: expose = 1

Total number of match sets in the tabulation = 621

Number of controls = 1
                No. of controls exposed
```

```
No. of cases exposed   0  1
                0 510 16
                1   5 90

Odds ratio by Mantel-Haenszel method = 0.312

Odds ratio by maximum likelihood estimate (MLE) method = 0.313
 95%CI= 0.114 , 0.853

# 如果expose变量相反, 则结果会不同, 请注意
> ex2=-expose+1 # 0, 1 相反
> matchTab(caseControl,ex2,pair)

Exposure status: ex2 = 1

Total number of match sets in the tabulation = 621

Number of controls = 1
                 No. of controls exposed
No. of cases exposed  0   1
                0 90   5
                1 16 510

Odds ratio by Mantel-Haenszel method = 3.2

Odds ratio by maximum likelihood estimate (MLE) method = 3.2
 95%CI= 1.172 , 8.735

# 手工计算
> i=x[1,2]
> j=x[2,1]
> n=i+j
> p=i/n
> q=1-p
> or=min(i,j)/max(i,j) # odds ratio
> or
[1] 0.3125
> c1=exp(log(or)-qnorm(0.975)*sqrt(1/(n*p*q))) # CI 1
> c1
[1] 0.1144825
> c2=exp(log(or)+qnorm(0.975)*sqrt(1/(n*p*q))) # CI 2
```

```
> c2
[1] 0.8530236

> or1=max(i,j)/min(i,j)
> or1
[1] 3.2
> c1=exp(log(or1)-qnorm(0.975)*sqrt(1/(n*p*q)))
> c1
[1] 1.172301
> c2=exp(log(or1)+qnorm(0.975)*sqrt(1/(n*p*q)))
> c2
[1] 8.734961
```

67.8 存在混杂的趋势性检验

如果有一个二态疾病变量(D), 一个二态暴露变量E, 及一个类型混杂变量C. 则在控制C后, 用Mantel-Haenszel检验去判断D与E的关联性. 若没有混杂, 使用二项比例的两样本检验, 或2×2 列联表法, 如果存在混杂, 使用Mantel-Haenszel检验

如果E是类型变量但多于2个水平, 例如2×k 列联表, 如果没有混杂, 使用趋势性卡方检验. 如果存在混杂, 使用Mantel-Extension 检验.

算法参考的是 [14] Page 578.

假设我们有s层. 每层二态疾病变量和k个有序类型的暴露变量的关系形成 2×k 列联表. 对于第j个类型有分数(打分)x_j, 如下表所示

疾病+	n_{i1}	n_{i2}	...	n_{ik}	n_i
疾病-	m_{i1}	m_{i2}	...	m_{ik}	m_i
	t_{i1}	t_{i2}	...	t_{ik}	N_i
分数(打分)	x_1	x_2	...	x_k	

要检验假设: $H_0 : \beta = 0 \quad H_1 : \beta \neq 0$. 此处$p_{ij}$ =第i层 第j暴露水平上个体中有病的比例=$\alpha_i + \beta x_j$.

计算检验统计量

$$X^2_{TR} = (|O - E| - 0.5)^2/V \sim \chi^2_1(H_0)$$

其中

$$O = \sum_{i=1}^{s} O_i = \sum_{i=1}^{s}\sum_{j=1}^{k} n_{ij}x_j$$

$$E = \sum_{i=1}^{s} E_i = \sum_{i=1}^{s}[(\sum_{j=1}^{k} t_{ij}x_j)\frac{n_i}{N_i}]$$

$$V = \sum_{i=1}^{s} V_i = \sum_{i=1}^{s} \frac{n_i m_i(N_i s_{2i} - s^2_{1i})}{N_i^2(N_i - 1)}$$

$$s_{1i} = \sum_{j=1}^{k} t_{ij}x_j, i = 1, 2, \cdots, s$$

$$s_{2i} = \sum_{j=1}^{k} t_{ij}x_j^2, i = 1, 2, \cdots, s$$

使用条件为 $V >= 5$.

若$X^2_{TR} > \chi^2_{1-\alpha}$我们拒绝$H_0$, 否则接受.

下面是一个例子([14] Page 578). 研究打鼾(ill)与年龄的关系, 混杂变量是性别. R的习惯将有病放在下面第二行, 暴露也放在右边第二列.

```
> x=array(c(603,196,486,223,232,103,348,188,383,313,206,232),
   dim=c(2,3,2),
   dimnames=list("ill"=c("no","yes"),
                 "age"=c("30-39","40-49","50-60"),
                 "sex"=c("M","F")))
> x
```

```
, , sex = M

    age
ill   30-39 40-49 50-60
  no    603   486   232
  yes   196   223   103

, , sex = F

    age
ill   30-39 40-49 50-60
  no    348   383   206
  yes   188   313   232

# R 中没有找到计算 Mantel-Extension 的函数
Mantel.Extension.test<-function(x){
   d=dim(x)
   s=d[3] # s层
   b=d[1] # 疾病二态, b=2
   k=d[2] # 暴露的k个水平
   score=1:k # 打分
   O=sum(x[2,,1:s]*score) # R的习惯将有病放在第二行, 暴露
也放在第二列.
   Ni=array(0,s) # 第s层总和
   ni=array(0,s) # 第s层第二行边际和
   mi=array(0,s) # 第s层第一行边际和
   for(i in 1:s){
      Ni[i]=sum(x[,,i])
      ni[i]=sum(x[2,,i])
      mi[i]=sum(x[1,,i])
}

   s1=colSums(colSums(x[,,1:s])*score)
   s2=colSums(colSums(x[,,1:s])*score^2)
   E=sum(s1*ni/Ni)
   V=sum(ni*mi*(Ni*s2-s1^2)/(Ni^2*(Ni-1)))
   X=(abs(O-E)-0.5)^2/V
   p=1-pchisq(q=X,df=1)
   cat("chi square: ",X," df=",1," p value=",p,"\n")
   res=list(statistics=X,df=1,p.value=p)
}
```

```
> r=Mantel.Extension.test(x)
chi square:  35.05958   df= 1   p value= 3.197706e-09
> r
$statistics
[1] 35.05958

$df
[1] 1

$p.value
[1] 3.197706e-09
```

Chapter 68

样本量及功效的估计

本节主要参考 [14] Page 580 13.6 和 [34] chapter 24 Sample size calculation.

68.1 计算样本量的函数

epicalc 包有4个计算样本量的函数.

第一个计算现状调查(prevalence survey, 流行度调查)的样本量.

第二个计算两个比例比较(comparison of two proportions)的样本量, 可以是 case-control study, cross-sectional study, cohort study or randomised controlled trial 之一.

第三个计算两个均值比较(comparison of two means)的样本量.

第四个是批质量检验抽样(lot quality assurance sampling.)样本量.

68.2 现场调查(Field survey)

现场调查(Field survey)的目的主要是获得某些人群的某种比例, 例如蠕虫病的发病率, 医疗服务的覆盖率等. 样本量依赖于估计的流行度(prevalence), 即比例, 和可接受的错误水平.

许多情况下采用整群抽样 (cluster sampling), 主要是为了减少采样时间和出行费用. 例如, 一个随机采样需要调查96个村子的96个人. 我们可以把村子减少到例如30个, 通过增加样本量来补偿这种整群抽样带来的影响. 实际上, 整群抽样减少了独立性, 也叫做设计效应(design effect).

函数 n.for.survey 用于计算现场调查的样本量. 首先看看参数

```
> args(n.for.survey)
function (p, delta = "auto", popsize = NULL, deff = 1, alpha = 0.05)
```

p: 估计的发病比例, 0,1 之间.

delta: p与置信区间的差. 例如估计p=0.3, 而最大的比例可能是0.5, 则 $delta = 0.5 - 0.3 = 0.2$. 函数中delta的值根据p值变化. $If 0.3 <= p <= 0.7, delta = 0.1. If 0.1 <= p < .3, or 0.7 < p <= 0.9, then delta = .05. Finally, if p < 0.1, then delta = p/2. If 0.9 < p, then delta = (1-p)/2$. delta 应该设的比较小, 以保证精确性, 但样本量会比较大. 一般从 0.1 每增加 0.1, 样本量会减少一半.

deff: 设计效应(design effect). 默认的是随机抽样deff为1. 对于群(cluster)大小很大, 群内的相似度很高, 那么deff就会很大. 样本量也会升高. 一般deff增加一倍, 样本量增加一倍.

alpha: I 型错误概率.

popsize: 所有总体的总的数量大小. 当比较大时样本量的变化就不大了.

对于整群抽样, 例如要到30个村子抽样, 样本量计算出来是210, 那么每个村子抽样7个就可以了.

```
> n.for.survey( p = .8, delta = .1, popsize = 500000, deff =2)

Sample size for survey.
Assumptions:
  Proportion      = 0.8
  Confidence limit = 95 %
  Delta          = 0.1 from the estimate.
  Population size  = 5e+05
  Design effect    = 2

  Sample size     = 123
# 改变 popsize
> n.for.survey( p = .8, delta = .1, popsize = 500, deff =2)
......
  Sample size     = 109

> n.for.survey( p = .8, delta = .1, popsize = 5000, deff =2)
......
  Sample size     = 121

> n.for.survey( p = .8, delta = .1, popsize = 50000, deff =2)
......
  Sample size     = 123

# 改变deff
> n.for.survey( p = .8, delta = .1, popsize = 50000, deff =4)
......
  Sample size     = 246

> n.for.survey( p = .8, delta = .1, popsize = 50000, deff =8)
......
  Sample size     = 491

# 改变delta
> n.for.survey( p = .8, delta = .2, popsize = 50000, deff =8)
......
  Sample size     = 123

> n.for.survey( p = .8, delta = .3, popsize = 50000, deff =8)
......
  Sample size     = 55
```

```
> n.for.survey( p = .8, delta = .4, popsize = 50000, deff =8)
......
  Sample size     = 31

# 默认popsize为极大值
> n.for.survey( p = .8, delta = .4, deff =8)
......
  Sample size     = 31
```

68.3 两个比例的比较

先看参数

```
> args(n.for.2p)
function (p1, p2, alpha = 0.05, power = 0.8, ratio = 1)
```

在回顾性研究(case-Control study)中, p1为case(diseased group, 患病人群)中暴露于危险因子(药物, 辐射等等)的比例, p2为control(non-diseased group, 对照人群, 非患病人群)中暴露于危险因子的比例.

在前瞻性研究(cohort study)中, p1为暴露人群的发病率, p2为非暴露人群的发病率.

在随机对照试验(randomised controlled trial)中, p1为给予新的治疗方法后有效或治愈的比例, p2为旧的治疗方法有效或治愈的比例.

alpha 为 I 型错误.

power 为功效, 即零假设不正确时拒绝零假设的概率.

ratio 比例p2所在样本量(control)与p1所在样本量(case)的比. 最有效的比例是 1:1.

例如, 在疾病人群(case)中的危险率为0.5, 而对照(control)中的危险率为0.2, 那么能够检验出疾病-对照差别的最小的样本量计算为

```
> n.for.2p(p1=0.5, p2=0.2)

Estimation of sample size for testing Ho: p1==p2
Assumptions:

    alpha = 0.05
    power = 0.8
      p1 = 0.5
      p2 = 0.2
    n2/n1 = 1

Estimated required sample size:

       n1 = 45
       n2 = 45
  n1 + n2 = 90
```

若疾病比较罕见, 例如一年才10个病例, 研究者想早点结束研究, 那么可以提高case:control的比例到, 例如1:4. 那么样本量为

```
> n.for.2p(p1=0.5, p2=0.2, ratio=4)

Estimation of sample size for testing Ho: p1==p2
Assumptions:

    alpha = 0.05
    power = 0.8
      p1 = 0.5
      p2 = 0.2
    n2/n1 = 4

Estimated required sample size:

       n1 = 27
```

```
      n2 = 108
  n1 + n2 = 135
```

比例再提高可能就不合适了, 例如1:9, n1下降了4个, 而n2几乎增加一倍.

```
> n.for.2p(p1=0.5, p2=0.2, ratio=9)
......
      n1 = 23
      n2 = 207
  n1 + n2 = 230
```

power增加样本量也增加

```
> n.for.2p(p1=0.5, p2=0.2, power=0.9)
......
      n1 = 58
      n2 = 58
  n1 + n2 = 116
```

68.4 病例-对照研究中p1,p2与优势比的关系

优势比的定义为

$$OR = \frac{p_1/q_1}{p_2/q2} = \frac{p_1 q_2}{p_2 q_1}$$

当$p1 = 0.5, p2 = 0.2$时, 优势比为

```
> .5/(1-.5)/(.2/(1-.2))
[1] 4
```

有时候知道p2和优势比, 需要求得p1, 那么根据公式计算就可以

```
> p2=0.2
> or=4 # 优势比
> odds2=p2/(1-p2) # p2的优势
> odds2
[1] 0.25
> odds1=or*odds2 # p1的优势
> odds1
[1] 1
> p1=odds1/(1+odds1) # p1
> p1
[1] 0.5
```

优势比减少, 则样本量会增加. 因为区别减少了, p1与p2的比例更接近1了. 若优势比为1, 样本量趋于无穷.

```
> p2=0.2
> or=2
> odds2=p2/(1-p2)
> odds1=or*odds2
> p1=odds1/(1+odds1)
> p1
[1] 0.3333333
> n.for.2p(p1,p2)
......
       n1 = 187
       n2 = 187
   n1 + n2 = 374
> n.for.2p(p1,p1)
......
Estimated required sample size:

       n1 = NaN
       n2 = NaN
   n1 + n2 = NaN
> n.for.2p(p2,p2)
......
```

```
    n1 = NaN
    n2 = NaN
n1 + n2 = NaN
```

68.5 前瞻性研究和随机对照试验中的样本量估计

使用方法与病例-对照研究一样.

68.6 现状研究中的样本量估计

现状研究(cross-sectional survey)有两个目的: (1) 发现发病率 (2) 检验暴露与结果的关系[1]. 前者是一个描述性的估计, 后者是一个假设检验. 这两者的计算方法是不同的.

对于现状研究中的假设检验, 应该使用 n.for.2p . p1为暴露组的阳性(positive outcomes)率, p2为非暴露组的阳性率. 换句话说, ratio 必须是暴露与非暴露组的比例.

例如, 在一个调查中, 暴露组的发病率(prevalence)可能估计为0.2, 非暴露的患病率可能是0.05. 由于暴露组的发病率(prevalence)可能估计为0.2, ratio(n2/n1)应该为 0.8/0.2=4.[2]

```
> n.for.2p(p1=0.2, p2=0.05, ratio=4)

Estimation of sample size for testing Ho: p1==p2
Assumptions:

    alpha = 0.05
    power = 0.8
```

[1]原文是: to test the association between the exposure and the outcome.

[2]原文描述可能有误, 上一段说 ratio 必须是暴露与非暴露组的比例. 而这里使用的是暴露组的非发病率除以发病率. 我认为上一段描述是正确的. ratio=0.2/0.05=4

```
      p1 = 0.2
      p2 = 0.05
   n2/n1 = 4

Estimated required sample size:

      n1 = 48
      n2 = 192
  n1 + n2 = 240
```

暴露组样本量为48, 非暴露组192.

我们还应该使用其它目的的检验来验证一下, 例如现场调查的方法, 估计的暴露组发病率为0.2.

```
> n.for.survey(p=0.2) # delta = 0.05

Sample size for survey.
Assumptions:
  Proportion      = 0.2
  Confidence limit = 95 %
  Delta          = 0.05 from the estimate.

  Sample size     = 246

> n.for.survey(p=0.2,delta=0.1)

Sample size for survey.
Assumptions:
  Proportion      = 0.2
  Confidence limit = 95 %
  Delta          = 0.1 from the estimate.

  Sample size     = 61
```

68.7 比较两个均值的样本量估计

在流行病学中, 比较两个均值不如比较比例的情况多. 因为治疗的决定和结果常常是二态的. 但是有一些结果是连续变量, 例如智商, 痛苦分数, 生活质量等.

两个均值常常有两个标准差, 因此参数也多一点

```
> args(n.for.2means)
function (mu1, mu2, sd1, sd2, ratio = 1, alpha = 0.05, power = 0.8)
```

对均值和标准差估计后就可以计算大概的样本量了

```
>  n.for.2means(mu1=0.8, mu2=0.6, sd1=0.2, sd2=0.25)

Estimation of sample size for testing Ho: mu1==mu2
Assumptions:

    alpha = 0.05
    power = 0.8
    n2/n1 = 1
      mu1 = 0.8
      mu2 = 0.6
      sd1 = 0.2
      sd2 = 0.25

Estimated required sample size:

       n1 = 21
       n2 = 21
  n1 + n2 = 42
```

68.8 批质量检验的样本量估计

批质量检验抽样(lot quality assurance sampling, LQAS)最初应用于工业领域. 目的是在一批产品中抽样检验, 如果合格率大于某个值(不合格率小于某个值), 这批产品就可以投放市场或交付使用, 否则这批产品被拒绝.

与其它抽样方法的区别是, LQAS不估计精确的次品率. 只是检验不合格率是否被超过. 这样所需的样本量就小于需要估计整体精确次品率(或总体发病率)的样本量. 这样在检验的费用很高时是一个替代的方法.

卫生系统采用LQAS主要是应用于监视问题的比例. 例如, anti-TB 药物的质量监控中, 成分化验和溶液检验非常昂贵. 于是使用LQAS来计算能够保证质量检验合格的最小的样本量.

假设最高可接受的次品率是1%, 如果研究表明比例小于等于此比例, 那么这批药物就被接受. 否则整批药物被拒绝. 而这批药物真实的不合格率并不重要. 如果样本量太小, 例如只有20, 那么即使所有样本合格, 也不能保证1%的不合格率, 而样本量太大, 例如1000, 那么就要浪费1000个药物.

```
> args(n.for.lqas)
function (p0, q = 0, N = 10000, alpha = 0.05, exact = FALSE)
> n.for.lqas(p=0.01)

    Lot quality assurance sampling

                         Method = Normal approximation
                  Population size = 10000
  Maximum defective sample accepted = 0
    Probability of defect accepted = 0.01
                          Alpha = 0.05
             Sample size required = 262
> n.for.lqas(p=0.01,N=1000)

    Lot quality assurance sampling

                         Method = Normal approximation
```

```
                Population size = 1000
 Maximum defective sample accepted = 0
   Probability of defect accepted = 0.01
                         Alpha = 0.05
           Sample size required = 212
```

总体为10000, 样本量为262, 那么检验后我们剩下10000-262=9738件药物可以使用, 而又保证了1%以下的次品率. 总体1000时, 样本量下降到212, 检验后还剩1000-212=788可用.

68.9 两个比例比较的功效

```
> table1 <- matrix(c(35,70,20,30),nr=2)
> table1
     [,1] [,2]
[1,]   35   20
[2,]   70   30
> library(epicalc)
> cc(cctable=table1)

            Exposure
Outcome       Non-exposed Exposed Total
  Non-diseased          35      20    55
  Diseased              70      30   100
  Total                105      50   155

OR =  0.75
95% CI = 0.35 1.61
Chi-squared = 0.66 ,  1 d.f. , P value = 0.417
Fisher's exact test (2-sided) P value = 0.474
```

优势比为0.75, 但是有比较宽的置信区间. 我们想知道当真正的优势比为0.5时其功效为多少?

```
> odds.a=20/30
```

```
> odds.treat=0.5*odds.a
> p.a=20/(20+30)
> p.treat=odds.treat/(1+odds.treat)
> power.for.2p(p1=p.treat,p2=p.a,n1=105,n2=50)

    alpha = 0.05
      p1 = 0.25
      p2 = 0.4
      n1 = 105
      n2 = 50


     power = 0.4082
```

此样本量有40%的概率发现真实的差异是0.5. 故此研究是不太可信的.

68.10 两个均值比较的功效

注意有图绘出.

```
> args(power.for.2means)
function (mu1, mu2, n1, n2, sd1, sd2, alpha = 0.05)
>  power.for.2means(mu1=95, mu2=100, sd1=11.7, sd2=10.1, n1=100, n2=100)

    alpha = 0.05
     mu1 = 95
     mu2 = 100
      n1 = 100
      n2 = 100
     sd1 = 11.7
     sd2 = 10.1

     power = 0.8988
```

68.11 分层类型数据样本量及功效的估计

TODO: 参考[14] 13.6 Page 580

Chapter 69

多重logistic回归

多重logistic回归实际上属于广义线性模型中的二项式回归. 参考广义线性回归部分, 对此有一些描述33.5

Mantel-Haenszel 检验和 Mantel-Extension 检验是对单个类型协变量C控制后, 检验二态疾病D和一个类型暴露变量E的关联性. 但是, 当下面条件之一成立

- E 是连续变量
- C 是连续变量
- 有多个混杂变量, 每个可能是类型或连续的

我们很难或不可能使用前面的方法去控制混杂.

多重logistic回归技术即可以处理前面的Mantel-Haenszel 检验和 Mantel-Extension 检验的情况, 也可以处理这里提出的三种情况.

多重logistic回归类似于多重线性回归, 但结果变量(或应变量)是二态而不是正态分布的.

69.1 一般模型

考虑一个模型

$$p = a + b_1x_1 + b_2x_2 + \cdots + b_kx_k$$

这里p=疾病发生的概率. 方程的右边可能会出现小于0或大于1的情况, 这是不应该的. 我们使用p的logit变换(即 logistic 变换)作为应变量

$$logit(p) = ln[p/(1-p)]$$

logit(p)可以取任何值. 函数编制很容易

```
logit<-function(p){
   res=log(p/(1-p))
   res
}
> logit(0.1)
[1] -2.197225
> logit(0.95)
[1] 2.944439
```

若把logit(p)作为独立变量$x_1, \cdots, x_k$的函数, 解得p则可得到下面的多重logistic回归模型

$$p = \frac{e^{a+b_1x_1+b_2x_2+\cdots+b_kx_k}}{1+e^{a+b_1x_1+b_2x_2+\cdots+b_kx_k}}$$

下面是一个虚构的例子, 两个物种$species = 0, 1$, 两种处理方法$run = 0, 1$, 一周后看看是否发病ill.

```
> x=0:1
> species=sample(x,200,replace=TRUE)
> n<-100
> run<-sample(x,200,replace=TRUE)
> ill<-c(rep(0,n), rep(1,n))
```

```
# logistic回归
> r<-glm(ill~species+run, family=binomial)
> r

Call:  glm(formula = ill ~ species + run, family = binomial)

Coefficients:
(Intercept)     species         run
    0.1107      0.1513     -0.3736

Degrees of Freedom: 199 Total (i.e. Null);  197 Residual
Null Deviance:     277.3
Residual Deviance: 275.4       AIC: 281.4

# summary 结果更丰富
> summary(r)

Call:
glm(formula = ill ~ species + run, family = binomial)

Deviance Residuals:
      Min         1Q     Median         3Q        Max
-1.2904870  -1.1304017   0.0001820   1.1307753   1.2908787

Coefficients:
          Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.1107     0.2553   0.434    0.665
species       0.1513     0.2876   0.526    0.599
run          -0.3736     0.2855  -1.308    0.191

(Dispersion parameter for binomial family taken to be 1)

   Null deviance: 277.26  on 199  degrees of freedom
Residual deviance: 275.36  on 197  degrees of freedom
AIC: 281.36

Number of Fisher Scoring iterations: 3
```

69.2 回归参数的解释

回归系数Coefficients类似于多重线性回归的偏回归系数. 假设两个个体A和B, 除了第j个暴露变量外其它都相同. 此处A为暴露($x_j = 1$)B为非暴露($x_j = 0$). 个体A,B成功概率的logit变换为$logit(p_A), logit(p_B)$, 则

$$logit(p_A) = a + b_1x_1 + \cdots + b_j(1) + \cdots + b_kx_k$$
$$logit(p_B) = a + b_1x_1 + \cdots + b_j(0) + \cdots + b_kx_k$$

两式相减有

$$logit(p_A) - logit(p_B) = b_j$$

由定义知

$$logit(p_A) = ln[p_A/(1 - p_B)]$$
$$logit(p_B) = ln[p_B/(1 - p_B)]$$

带入得

$$ln[p_A/(1 - p_B)] - ln[p_B/(1 - p_B)] = b_j$$

即

$$ln[\frac{p_A/(1 - p_B)}{p_B/(1 - p_B)}] = b_j$$

取反对数既得

$$\frac{p_A/(1 - p_B)}{p_B/(1 - p_B)} = e^{b_j}$$

由优势比的定义我们可以重写为

$$\frac{Odd_A}{Odd_B} = b_j$$

我们知道Odd_A/Odd_B就是第j个暴露变量与疾病的优势比. 而这两个个体其它变量都是相同的. 即此优势比是调整模型中其它危险因素后得到的.

我们总结如下.

69.2.1 二态独立变量在多重logistic回归模型中优势比的估计

对于多重logistic回归模型, 假设一个二态变量1表示有暴露, 0表示无暴露. 这个暴露变量对于应变量的优势比(OR)被估计为

$$\hat{OR} = e^{b_j}$$

这个优势比是调整模型中其它变量后的结果. 它的双侧置信区间为

$$[e^{b_j - z_{1-\alpha/2} se(b_j)}, e^{b_j + z_{1-\alpha/2} se(b_j)}]$$

例子中控制了物种 species 后两种处理方法run患病的优势比及区间为

```
# 处理方法的系数
> b2=r$coeff[3]
> b2
      run
-0.3735543

# R函数计算系数置信区间, 使用的是t分布近似
> confint(r)
Waiting for profiling to be done...
              2.5 %    97.5 %
(Intercept) -0.3899164 0.6146411
species     -0.4119937 0.7173819
run         -0.9367140 0.1843467

# 下面是手工计算

# 获取se
> summary(r)$coeff
            Estimate Std. Error    z value  Pr(>|z|)
(Intercept)  0.1107039  0.2552502  0.4337073 0.6645010
species      0.1512521  0.2875845  0.5259395 0.5989302
run         -0.3735543  0.2855024 -1.3084104 0.1907341
```

```
> summary(r)$coeff[3,2]
[1] 0.2855024
> se=summary(r)$coeff[3,2]
> se
[1] 0.2855024

# 系数置信区间

# 正态分布近似
> b2-qnorm(0.975)*se
      run
-0.9331286
> b2+qnorm(0.975)*se
     run
0.1860201

# t分布近似
> b2-qt(0.975,199)*se
[1] -0.9365526
> b2+qt(0.975,199)*se
[1] 0.1894440

# 优势比
> OR=exp(b2) # 优势比
> OR
     run
0.6882836

# 优势比的置信区间. Rosner 给出的是使用正态分布的近似
> exp(b2-qnorm(0.975)*se)
     run
0.3933212
> exp(b2+qnorm(0.975)*se)
    run
1.204446
```

69.2.2 logistic回归分析和列联表分析的关系

设我们有一个二态疾病变量D和一个二态暴露变量E, 数据是由前瞻性研究, 回顾性研究或现状研究的任何一种产生, 列联表如下我们可以用下面两个等价的方法任何一种估

	E(+)	E(-)
D(+)	a	b
D(-)	c	d

计D与E之间的优势比

- 直接从列联表中求出优势比=ad/bc
- 我们建立一个logistic回归模型

$$ln[p/(1-p)] = \alpha + \beta E$$

p=在暴露变量E下有病D的概率. 此处产生的优势比为e^{β}

对于前瞻性研究或现状研究, 我们可以用下面两个等价的方法任何一种估计个体在暴露下的疾病概率(p_E)及未暴露下的疾病概率($p_{\bar{E}}$

- 从列联表中有

$$p_E = a/(a+c), p_{\bar{E}} = b/(b+d)$$

- 由logistic回归模型

$$p_E = e^{\alpha+\beta}/(1+e^{\alpha+\beta}), p_{\bar{E}} = e^{\alpha}/(1+e^{\alpha})$$

p=在暴露变量E下有病D的概率. 此处产生的优势比为e^{b}

对于回顾性研究(病例-对照研究)我们不可能估计疾病发生的概率, 除非病例及对照样本数在总体中的比例已知, 这几乎不可能.

下面是一个例子. 数据来自[14] 例 10.7 数据为表 10.1 Page 344. 关于乳腺癌与初次生育年龄的关系. 数据见下. 列联表分析的Fisher检验给出优势比为 1.57. 使用logistic回归(需要由列联表重构原始数据)的系数为 0.4523, 优势比为$e^{0.4523} = 1.57$, 与列联表法估计相同.

```
> x <- matrix(c(683,1498,2537, 8747), nr = 2,
       dimnames=list(c("D+","D-"),c(">=30","<=29")))
> x
   >=30 <=29
D+  683 2537
D- 1498 8747
> fisher.test(x)

        Fisher's Exact Test for Count Data

data:  x
p-value < 2.2e-16
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 1.419073 1.740189
sample estimates:
odds ratio
  1.571925

# 重构原始数据
> d=c(rep(1,683+2537),rep(0,1498+8747))
> e=c(rep(1,683),rep(0,2537),rep(1,1498),rep(0,8747))
> table(d,e)
   e
d     0    1
  0 8747 1498
  1 2537  683

# logistic 回归 e~d
> glm(e~d,fam=binomial)

Call:  glm(formula = e ~ d, family = binomial)

Coefficients:
```

```
(Intercept)          d
   -1.7646      0.4523

Degrees of Freedom: 13464 Total (i.e. Null);  13463 Residual
Null Deviance:     11930
Residual Deviance: 11850       AIC: 11860

# logistic 回归 d~e
> glm(d~e,fam=binomial)

Call:  glm(formula = d ~ e, family = binomial)

Coefficients:
(Intercept)          e
   -1.2377      0.4523

Degrees of Freedom: 13464 Total (i.e. Null);  13463 Residual
Null Deviance:     14810
Residual Deviance: 14740       AIC: 14740

# 计算优势比
> exp(0.4523)
[1] 1.571923
```

69.3 协方差,标准差,t值,置信区间等

变量多于1个且不独立时, 有对称的协方差矩阵, 可以是尺度的或非尺度的('scaled' or 'unscaled'). 尺度因子实际上是glm的dispersion.

接乳腺癌与初次生育年龄的关系的例子

```
> r=glm(d~e,fam=binomial)
> names(summary(r))
 [1] "call"          "terms"         "family"        "deviance"
 [5] "aic"           "contrasts"     "df.residual"   "null.deviance"
 [9] "df.null"       "iter"          "deviance.resid" "coefficients"
```

```
[13] "aliased"      "dispersion"    "df"           "cov.unscaled"
[17] "cov.scaled"

# 协方差矩阵
> summary(r)$cov.scaled
            (Intercept)           d
(Intercept)  0.0007818816 -0.0007818816
d           -0.0007818816  0.0026401751

> summary(r)$cov.unscaled
            (Intercept)           d
(Intercept)  0.0007818816 -0.0007818816
d           -0.0007818816  0.0026401751

# 直接计算协方差矩阵
> vcov(r)
            (Intercept)           d
(Intercept)  0.0007818816 -0.0007818816
d           -0.0007818816  0.0026401751

# 尺度因子
> summary(r)$dispersion
[1] 1
```

d的标准差为

```
# 直接计算
> vcov(r)[2,2]^.5->se2
> se2
[1] 0.05138263
> vcov(r)[2,2]^.5
[1] 0.05138263

# summary里的标准差
> summary(r)$coefficients
            Estimate Std. Error    z value    Pr(>|z|)
(Intercept) -1.7645799 0.02796215 -63.106027 0.000000e+00
d            0.4523372 0.05138263   8.803309 1.328402e-18
```

t-值为系数除以标准差(z-值)

```
> t<-summary(r)$coefficients[2,1]/summary(r)$cov.scaled[2,2]^.5
> t
[1] 8.803309

# 方差开平方, 与coefficients里的标准差一致
> summary(r)$cov.scaled[2,2]^.5
[1] 0.05138263
```

p-值

```
> pt(q=t, df=13465, lower.tail=FALSE) * 2
[1] 1.488717e-18
```

置信区间

```
> b=summary(r)$coefficients[2,1]
> confint(r)
Waiting for profiling to be done...
               2.5 %    97.5 %
(Intercept) -1.8197435 -1.7101259
d           0.3512376  0.5526807
> b1=b+qt(c(0.025, 0.975), 13)*se2
> b1
[1] 0.3413318 0.5633426
> b1=b+qnorm(c(0.025, 0.975))*se2
> b1
[1] 0.3516291 0.5530453
```

69.4 logistic.display函数

详见[34] 15章

```
> library(epicalc)
> logistic.display(r)

Logistic regression predicting e

         OR(95%CI)        P(Wald's test) P(LR-test)
d: 1 vs 0  1.57 (1.42,1.74)  < 0.001       < 0.001

Log-likelihood = -5926.7668
No. of observations = 13465
AIC value = 11857.5336
```

69.5 连续独立变量在多重logistic回归模型中优势比的估计

假设有一个连续变量(x_j), 两个个体在x_j上取值分别为$x_j+\delta, x_j$. 则第一个个体相对于第二个个体的优势比估计为

$$OR = e^{\beta_j+\delta}$$

它的双侧置信区间为

$$[e^{[b_j-z_{1-\alpha/2}se(b_j)]\delta}, e^{[b_j+z_{1-\alpha/2}se(b_j)]\delta}]$$

δ常常取一个自己确定的有意义的值.

例子参考 [14] Page 589, 例13.34.

69.6 假设检验

假设检验

$$H_0: \ b_j = 0, \ all \ other \ b_l \neq 0$$
$$H_1: \ b_j \neq 0$$

计算检验统计量, 其中se可以使用summary函数得到. 零假设下有

$$z = b_j/se(b_j) \sim N(0,1)$$

双侧检验, 若$z < z_{\alpha/2}$或$z > z_{1-\alpha/2}$, 拒绝零假设. 否则接受.

最后, 仅在x_j与应变量没有显著关系时, OR的置信区间包含1.

乳腺癌与初次生育年龄的关系的例子中, summary函数可以得到z值与p-值.

```
> summary(glm(e~d,fam=binomial))

Call:
glm(formula = e ~ d, family = binomial)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-0.6905  -0.5623  -0.5623  -0.5623   1.9609

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.76458    0.02796 -63.106   <2e-16 ***
d            0.45234    0.05138   8.803   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

   Null deviance: 11928  on 13464  degrees of freedom
Residual deviance: 11854  on 13463  degrees of freedom
AIC: 11858

Number of Fisher Scoring iterations: 4

# 手工计算z值
> z=0.45234/0.05138
> z
[1] 8.803815
```

```
> (1-pnorm(z))*2 # p-值
[1] 0
```

69.7 多重logistic回归中的预测

我们可以使用多重logistic回归模型去预测有协变量$x_1, \cdots, x_k$的个体的患病概率. 若回归参数已知, 则

$$p = \frac{e^L}{1+e^L}$$

$$L = a + b_1 x_1 + b_2 x_2 + \cdots + b_k x_k$$

L有时称为线性预测量. 线性预测量的置信区间

$$(L_1, L_2) = L \pm z_{1-\alpha/2} se(L)$$

变换回概率尺度得到

$$p_1 = \frac{e^{L_1}}{1+e^{L_1}}, p_2 = \frac{e^{L_2}}{1+e^{L_2}}$$

se(L)的计算比较复杂, 需要矩阵知识, 但是可以由电脑计算出.

实际上R有一个针对glm的函数 predict 可以计算预测值.

69.8 logistic模型回归拟合优良性的估计

TODO: 参考[14] Page 597 13.6.7节

我们可以用预测概率去定义残差及判断logistic回归模型拟合的优良性.

logistic回归中的残差: 若我们的数据以非群组的形式出现, 即每个个体都有一组协变量, 我们可以定义第i个个体的

Pearson 残差

$$r_i = \frac{y_i - \hat{p}_i}{se(\hat{p}_i)}$$

此处 $y_i = 1$若第i个体是成功, 否则$y_i = 0$.

$$\hat{p}_i = \frac{e^{L_i}}{1 + e^{L_i}}$$

$$L_i = \hat{a} + \hat{b_1}x_1 + \cdots + \hat{b_k}x_k = \text{第i个体的预测值}$$

$$se(\hat{p}_i) = \sqrt{\hat{p}_i(1 - \hat{p}_i)}$$

若我们的数据是群组的形式, 即一些个体有相同的协变量形成一个组, 则第i组的 Pearson 残差为

$$r_i = \frac{y_i - \hat{p}_i}{se(\hat{p}_i)}$$

此处

$$y_i = \text{第i组成功的比例}$$

$$\hat{p}_i = \frac{e^{L_i}}{1 + e^{L_i}}(\text{与非群组相同})$$

$$L_i = \hat{a} + \hat{b_1}x_1 + \cdots + \hat{b_k}x_k = \text{第i个体的预测值}$$

$$se(\hat{p}_i) = \sqrt{\hat{p}_i(1 - \hat{p}_i)/n_i}$$

$$n_i = \text{第i组中个体数目}$$

这里的 pearson 残差类似于线性回归中的 Studentized 残差, 残差是各不相同的. 这里的标准误基于二项分布. 在分组数据中, 若$\hat{p}_i$接近0或1, 或n_i增大时, 标准误下降.

pearson 残差比较大时我们可能需要修正模型. 还可以使用 pearson 残差识别异常值. 但是个体残差的使用要比线性回归模型有更多的限制, 特别在非群组的时候.

可以使用其它方法判断拟合优良性. 例如, 可以考察每个值如何影响回归系数. 假设第j个回归系数为b_j, 删除第i个观察值

后回归系数为b_j^i, 那么第i观察值对b_j的影响的测度可以用下式衡量

$$\delta b_j^i = \frac{b_j - b_j^i}{se(b_j)}$$

pearson 残差需要手工计算.

我们可以使用step, add等逐步回归函数来检验. 下面是一个step的例子

```
> x1=rbinom(100,size=1,prob=0.5)
> x2=rbinom(100,size=1,prob=0.5)
> x3=rbinom(100,size=1,prob=0.5)
> y=rbinom(100,size=1,prob=0.5)
> r=glm(y~x1+x2+x3,family=binomial)
> s=step(r)
Start:  AIC=146.35
y ~ x1 + x2 + x3

      Df Deviance    AIC
- x3    1   138.40 144.40
- x2    1   138.42 144.42
- x1    1   138.54 144.54
<none>     138.35 146.35

Step:  AIC=144.4
y ~ x1 + x2

      Df Deviance    AIC
- x2    1   138.46 142.46
- x1    1   138.59 142.59
<none>     138.40 144.40

Step:  AIC=142.46
y ~ x1

      Df Deviance    AIC
```

```
- x1    1   138.63 140.63
<none>      138.46 142.46

Step:  AIC=140.63
y ~ 1

> summary(s)

Call:
glm(formula = y ~ 1, family = binomial)

Deviance Residuals:
   Min     1Q  Median     3Q    Max
-1.177  -1.177   0.000   1.177   1.177

Coefficients:
            Estimate Std. Error   z value Pr(>|z|)
(Intercept) -9.498e-18  2.000e-01 -4.75e-17       1

(Dispersion parameter for binomial family taken to be 1)

   Null deviance: 138.63  on 99  degrees of freedom
Residual deviance: 138.63  on 99  degrees of freedom
AIC: 140.63

Number of Fisher Scoring iterations: 2

> anova(s)
Analysis of Deviance Table

Model: binomial, link: logit

Response: y

Terms added sequentially (first to last)


    Df Deviance Resid. Df Resid. Dev
NULL                 99     138.63
```

69.9 logistic回归的ROC曲线

定义参考概念部分ROC65.8

函数lroc计算及绘制logistic回归的ROC曲线. 下面是帮助的例子.

```
library(epicalc)
> model1 <- glm(case ~ induced + spontaneous, data=infert, family=binomial)
> logistic.display(model1)

Logistic regression predicting case

                     crude OR(95%CI)  adj. OR(95%CI)    P(Wald's test)
induced (cont. var.)    1.05 (0.74,1.5)  1.52 (1.02,2.27)  0.042

spontaneous (cont. var.) 2.9 (1.97,4.26)  3.31 (2.19,5.01)  < 0.001

                     P(LR-test)
induced (cont. var.)    0.042

spontaneous (cont. var.) < 0.001

Log-likelihood = -139.806
No. of observations = 248
AIC value = 285.612

>     # Having two spontaneous abortions is quite close to being infertile!
>     # This is actually not a causal relationship
>
> lroc(model1, title=TRUE, auc.coords=c(.5,.1))
$model.description
[1] "logit (case ~ induced + spontaneous)"

$auc
[1] 0.7285506

$predicted.table
 predicted.prob Non-diseased Diseased
```

```
        0.1534          60       7
        0.2158          33      12
        0.2949          20       9
        0.3750          25      22
        0.4768          11       5
        0.5806           4       4
        0.6651          11      18
        0.7511           1       6

$diagnostic.table
        1-Specificity Sensitivity
          1.000000000  1.00000000
> 0.1534   0.636363636  0.91566265
> 0.2158   0.436363636  0.77108434
> 0.2949   0.315151515  0.66265060
> 0.3750   0.163636364  0.39759036
> 0.4768   0.096969697  0.33734940
> 0.5806   0.072727273  0.28915663
> 0.6651   0.006060606  0.07228916
> 0.7511   0.000000000  0.00000000
```

Chapter 70

meta再分析

本节主要参考

《A Handbook of Statistical Analyses Using R》 Brian S. Everitt and Torsten Hothorn. CHAPTER 12, Meta-Analysis: Nicotine Gum and Smoking Cessation and the Efficacy of BCG Vaccine in the Treatment of Tuberculosis.

《生物统计学基础》13.8 再分析。

70.1 软件包

rmeta: Functions for simple fixed and random effects meta-analysis for two-sample comparisons and cumulative meta-analyses. Draws standard summary plots, funnel plots, and computes summaries and tests for association and heterogeneity

meta: Fixed and random effects meta-analysis. Functions for tests of bias, forest and funnel plot.

metafor: The metafor package consists of a collection of functions for conducting meta-analyses in R. The package includes functions to calculate various effect size or outcome measures, fit fixed-, random-, and mixed-effects

models to such data, carry out moderator and meta-regression analyses, and create various types of meta-analytical plots. For meta-analyses of 2x2 table data, the Mantel-Haenszel and Peto's method are also implemented. http://www.metafor-project.org/. 文 献 Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. Journal of Statistical Software, 36(3), 1-48. 介 绍 了 其 他 meta 分 析 的 软 件. http://www.metafor-project.org/article_list.txt 列 出 了 使 用 metafor 的 文 献. 可 以 参 考.

MAMA: Implementation of methods for microarray meta-analysis. (MAMA: a 9 in 1 R package for Meta-Analysis of MicroArray, 47page Vignettes)

http://www.lyonsmorris.com/ma1/index.cfm 是 在 线 meta 分 析 的 工 具.

Mix 是 针 对 excel 的meta分 析 工 具.

RevMan 是 java 写 的 meta 分 析 的 工 具.

Meta-Stat is a DOS-based computer program that automates the many complex tasks that are required to perform a meta-analysis.

Comprehensive Meta Analysis: 针 对 windows 平 台, 商 业 软 件.

70.2 概 念

前 面 的 分 析, 我 们 都 是 对 某 个 研 究 中 的 数 据 做 分 析. 但 是 在 某 种 研 究 中, 我 们 希 望 把 不 同 组 群 的 研 究 分 析 结 果 综 合 成 一 个 结 果. 某 些 研 究 中, 不 同 的 研 究 结 果 似 乎 矛 盾, 另 外 一 些 研 究 中, 它 们 之 间 似 乎 没 有 什 么 显 著 差 异.

现 在 的 问 题 是, 用 什 么 方 法 把 这 些 研 究 联 合 起 来 以 便 减 少 抽 样 误 差 并 增 加 研 究 的 功 效? 如 何 解 决 不 同 研 究 中 的 不 相 容 性? 完 成 这 样 研 究 的 技 术 称 为 再 分 析(meta analysis).

70.3 DerSimonian-Laird 方 法(随 机 效 应 模 型)

假 设 有k个 研 究, 每 个 的 目 标 都 是 估 计 优 势 比exp(u)–在 每 个 处 理 组 中 的 疾 病 优 势 相 对 于 对 照 组 中 的 疾 病 优 势.

(1) 把k个 研 究 联 合, 平 均 对 数 优 势 比$u = ln(OR)$的 最 好 估 计 为

$$\hat{u} = \sum_{i=1}^{k} w_i' y_i / \sum_{i=1}^{k} w_i'$$

此 处

$$\begin{aligned} y_i &= \text{第i个 研 究 中 的 对 数 优 势 比} \\ w_i' &= (s_i^2 + \delta^2)^{-1} \\ 1/w_i &= s_i^2 = 1/a_i + 1/b_i + 1/c_i + 1/d_i = \text{第i个 研 究} \end{aligned}$$

a_i, b_i, c_i, d_i是 第i个 研 究 中2*2列 联 表 的 计 数

$$\delta^2 = max\{0, [Q_w - (k-1)]/[\sum_{i=1}^{k} w_i - (\sum_{i=1}^{k} w_i^2 / \sum_{i=1}^{k}$$

$$Q_w = \sum_{i=1}^{k} w_i (y_i - \bar{y_w})^2 = \sum_{i=1}^{k} w_i y_i^2 - (\sum_{i=1}^{k} w_i y_i)^2 / \sum_{i=1}^{k}$$

$$\bar{y_w} = \sum_{i=1}^{k} w_i y_i / \sum_{i=1}^{k} w_i$$

对 应 的 优 势 比 的 点 估 计$= exp(\hat{u})$

(2) $\hat{u}$的 标 准 误 为

$$se(\hat{u}) = (1/\sum_{i=1}^{k} w_i')^{1/2}$$

(3) u的100%$(1-\alpha)$置 信 区 间 为

$$\hat{u} \pm z_{1-\alpha/2} se(\hat{u}) = (u_1, u_2)$$

OR 的100%$(1-\alpha)$置信区间为$(exp(u_1), exp(u_2))$

(4) 检验假设 $H_0 : u = 0 \quad vs \quad H_1 : u \neq 0$即$H_0 : OR = 1 \quad vs \quad H_1 : OR \neq 1$ 检验统计量为

$$z = \hat{u}/se(\hat{u}) \sim N(0,1)$$

双侧p-值为为$2[1-\Phi(|z|)]$.

meta.DSL 用于随机效应和异质性的 Woolf's test. 先看看参数

```
> library(rmeta)
> args(meta.DSL)
function (ntrt, nctrl, ptrt, pctrl, conf.level = 0.95, names = NULL,
    data = NULL, subset = NULL, na.action = na.fail, statistic = "OR")
```

ntrt: 暴露组(treated/exposed group)的个体数目.

nctrl: 对照组的个体数目.

ptrt: 暴露组的成功/发病个体数目(Number of events in treated/exposed group)

pctrl: 对照组的成功/发病个体数目(Number of events in control group)

statistic: OR是优势比, RR为相对危险率

下面是一个例子, 取自[29]

```
>  data("smoking", package = "HSAUR")
# 每一行为一个单独的研究.
# qt,tt分别为暴露组的发病个体数目和个体总数. qc, tc分别为对照组的发病数目和个体总数.
> smoking
            qt  tt  qc  tc
Blondal89    37  92  24  90
Campbell91   21 107  21 105
Fagerstrom82 30  50  23  50
```

```
Fee82        23 180  15 172
Garcia89     21  68   5  38
Garvey00     75 405  17 203
Gross95      37 131   6  46
Hall85       18  41  10  36
Hall87       30  71  14  68
Hall96       24  98  28 103
Hjalmarson84  31 106  16 100
Huber88      31  54  11  60
Jarvis82     22  58   9  58
Jensen91     90 211  28  82
Killen84     16  44   6  20
Killen90    129 600 112 617
Malcolm80     6  73   3 121
McGovern92   51 146  40 127
Nakamura90   13  30   5  30
Niaura94      5  84   4  89
Pirie92      75 206  50 211
Puska79      29 116  21 113
Schneider85   9  30   6  30
Tonnesen88   23  60  12  53
Villa99      11  21  10  26
Zelman92     23  58  18  58

>  smokingDSL <- meta.DSL(smoking[["tt"]], smoking[["tc"]],
+                   smoking[["qt"]], smoking[["qc"]],
+                   names = rownames(smoking))
> smokingDSL
Random effects ( DerSimonian-Laird ) meta-analysis
Call: meta.DSL(ntrt = smoking[["tt"]], nctrl = smoking[["tc"]], ptrt = smoking[["q
   pctrl = smoking[["qc"]], names = rownames(smoking))
Summary OR= 1.75   95% CI ( 1.48, 2.07 )
Estimated random effects variance: 0.05

# summary函数除了上面的结果, 还有详细的每个的OR的估计.
> summary(smokingDSL)
Random effects ( DerSimonian-Laird ) meta-analysis
Call: meta.DSL(ntrt = smoking[["tt"]], nctrl = smoking[["tc"]], ptrt = smoking[["q
   pctrl = smoking[["qc"]], names = rownames(smoking))
------------------------------------
            OR (lower  95% upper)
```

```
Blondal89    1.85    0.99      3.46
Campbell91   0.98    0.50      1.92
Fagerstrom82 1.76    0.80      3.89
Fee82        1.53    0.77      3.05
Garcia89     2.95    1.01      8.62
Garvey00     2.49    1.43      4.34
Gross95      2.62    1.03      6.71
Hall85       2.03    0.78      5.29
Hall87       2.82    1.33      5.99
Hall96       0.87    0.46      1.64
Hjalmarson84 2.17    1.10      4.28
Huber88      6.00    2.57     14.01
Jarvis82     3.33    1.37      8.08
Jensen91     1.43    0.84      2.44
Killen84     1.33    0.43      4.15
Killen90     1.23    0.93      1.64
Malcolm80    3.52    0.85     14.54
McGovern92   1.17    0.70      1.94
Nakamura90   3.82    1.15     12.71
Niaura94     1.34    0.35      5.19
Pirie92      1.84    1.20      2.82
Puska79      1.46    0.78      2.75
Schneider85  1.71    0.52      5.62
Tonnesen88   2.12    0.93      4.86
Villa99      1.76    0.55      5.64
Zelman92     1.46    0.68      3.14
-----------------------------------
SummaryOR= 1.75  95% CI ( 1.48,2.07 )
Test for heterogeneity: X^2( 25 ) = 34.87 ( p-value 0.0905 )
Estimated random effects variance: 0.05
```

70.4 Mantel-Haenszel 方法(固定效应模型)

固定效应模型使用 Mantel-Haenszel 方法, 在个体数目比较少(小于5)时比较精确. 其它的方法有 Peto's method, 计算上简单, 是 Mantel-Haenszel 方法的近似(rmeta没有提供).

固定效应模型函数 meta.MH 的参数与随机效应模型的 meta.DSL 的参数一样.

```
> args(meta.MH)
function (ntrt, nctrl, ptrt, pctrl, conf.level = 0.95, names = NULL,
   data = NULL, subset = NULL, na.action = na.fail, statistic = "OR")
```

下面与随机效应模型是同一个例子.

```
# 固定效应模型
>  smokingOR <- meta.MH(smoking[["tt"]], smoking[["tc"]],
+                    smoking[["qt"]], smoking[["qc"]],
+                    names = rownames(smoking))
> summary(smokingOR)
Fixed effects ( Mantel-Haenszel ) meta-analysis
Call: meta.MH(ntrt = smoking[["tt"]], nctrl = smoking[["tc"]], ptrt = smoking[["qt
   pctrl = smoking[["qc"]], names = rownames(smoking))
-------------------------------------
             OR (lower  95% upper)
Blondal89    1.85    0.99      3.46
Campbell91   0.98    0.50      1.92
Fagerstrom82 1.76    0.80      3.89
Fee82        1.53    0.77      3.05
Garcia89     2.95    1.01      8.62
Garvey00     2.49    1.43      4.34
Gross95      2.62    1.03      6.71
Hall85       2.03    0.78      5.29
Hall87       2.82    1.33      5.99
Hall96       0.87    0.46      1.64
Hjalmarson84 2.17    1.10      4.28
Huber88      6.00    2.57     14.01
Jarvis82     3.33    1.37      8.08
Jensen91     1.43    0.84      2.44
Killen84     1.33    0.43      4.15
Killen90     1.23    0.93      1.64
Malcolm80    3.52    0.85     14.54
McGovern92   1.17    0.70      1.94
Nakamura90   3.82    1.15     12.71
```

```
Niaura94     1.34    0.35      5.19
Pirie92      1.84    1.20      2.82
Puska79      1.46    0.78      2.75
Schneider85  1.71    0.52      5.62
Tonnesen88   2.12    0.93      4.86
Villa99      1.76    0.55      5.64
Zelman92     1.46    0.68      3.14
-------------------------------------
Mantel-Haenszel OR =1.67 95% CI ( 1.47,1.9 )
Test for heterogeneity: X^2( 25 ) = 34.9 ( p-value 0.09 )
```

70.5 优势比的齐性检验

检验假设对数优势比$u = ln(OR)$

$H_0 : u_1 = \cdots = u_k \quad vs \quad H_1 :$至少两个对数优势比不同

使用下面的检验统计量

$$Q_w = \sum_{i=1}^{k} w_i (y_i - \bar{y_w})^2 \sim \chi^2_{k-1}$$

固定效应模型中不同研究之间的方差在近似研究权重时被忽略了, 而仅考察内部方差. 故方程中仅用w_i代替w_i'. 有一个争论是如果优势比有实质性的差异, 则应该研究差异的来源且不应该报告联合的优势比.

一般讲, 固定模型比随机模型会有更小的置信区间, 更易得出显著性的结论. 但是固定模型和随机模型会有不同的权, 故两个模型可以有不同的优势比结果. 更详细的讨论参考 [14] Page 607.

一般, 使用下面的规则决定用什么模型. 若异质性检验的p-值

- >= 0.5 使用固定效应模型.

- $0.05 <= p < 0.5$使用随机效应模型.
- < 0.05 不要报告合并的优势比, 寻找异质性的来源.

70.6 解释

随机效应的结果为:

```
SummaryOR= 1.75  95% CI ( 1.48,2.07 )
Test for heterogeneity: X^2( 25 ) = 34.87 ( p-value 0.0905 )
Estimated random effects variance: 0.05
```

固定效应的结果为:

```
Mantel-Haenszel OR =1.67 95% CI ( 1.47,1.9 )
Test for heterogeneity: X^2( 25 ) = 34.9 ( p-value 0.09 )
```

我们看到随机效应模型比固定效应模型的CI要宽泛. 异质性检验(Test for heterogeneity)的p-值为0.09, 所有我们最后决定使用随机效应模型.

70.7 绘图

可以对所有OR及置信区间绘图.

```
> plot(smokingOR)
```

Chapter 71

等效性研究(equivalence study)

参考 [14] 13.9 等效性研究. Page 608.

近年来提出了一种新的研究设计形式, 主要目标是研究两种方法是否等效而不是一种优于另一种. 这种研究称为等效研究(equivalence study). 具体参考定义部分.

71.1 统计推断

等效性研究实际上是考察危险率差的单侧检验, 即两个二项比例之差的较低的单侧检验.

设p_1, p_2分别是标准方法和试验方法中的生存率. 我们将寻找一个$p_1 - p_2$的较低的单侧$100\%(1-\alpha)$置信区间. 单侧置信区间为

$$p_1 - p_2 < \hat{p_1} - \hat{p_2} + z_{1-\alpha}\sqrt{\hat{p_1}\hat{q_1}/n_1 + \hat{p_2}\hat{q_2}/n_2}$$

若右边不超过事先指定的差值δ, 称这两个处理是等效的.

p-值也是可以计算出来的, 只要将$p_1 - p_2$标准化后(近似服从标准正态分布)计算超过此值的概率.

71.2 样本量的估计

如果试验组样本量(n_2)是标准组(n_1)的k倍(k是事先指定的), 我们有

$$n_1 = \frac{(\hat{p_1}\hat{q_1} + \hat{p_2}\hat{q_2}/k)(z_{1-\alpha} + z_{1-\beta})^2}{[\delta - (p_1 - p_2)]^2}$$

$$n_2 = kn_1$$

无效假设下, 可以认为$\hat{p_1} = \hat{p_2} = p$, 带入上式既得

$$n_1 = \frac{(pq)(1 + 1/k)(z_{1-\alpha} + z_{1-\beta})^2}{\delta^2}$$

$$n_2 = kn_1$$

TODO: 编写函数, 例子

Chapter 72

交叉设计

交叉设计(cross over design)类似于 Wilcoxon 符号-秩检验. 但是考虑到了前后匹配的效应.

72.1 综合的处理效应的估计

记 x_ijk=交叉设计中病人在第k周期, 第i组 第j个病人的得分值, $k = 1, 2; i = 1, 2; j = 1, \cdots, n_i$.

(1) 计算处理有效性的总估计

$$\bar{d} = (\bar{d_1} + \bar{d_2})/2$$

$$\bar{d_1} = \sum_{j=1}^{n_1} d_{1j}/n_1$$

$$\bar{d_2} = \sum_{j=1}^{n_2} d_{2j}/n_2$$

$$d_{1j} = x_{1j1} - x_{1j2}$$
$$d_{2j} = x_{2j2} - x_{2j1}$$

(2) $\bar{d}$的标准误估计为

$$se = \sqrt{\frac{s^2_{d,pooled}}{4}(1/n_1 + 1/n_2)} = \frac{s_{d,pooled}}{2}\sqrt{(1/n_1 + 1/n_2)}$$

$$s^2_{d,pooled} = \frac{(n_1 - 1)s^2_{d_1} + (n_2 - 1)s^2_{d_2}}{n - 1 + n_2 - 2}$$

$$s^2_{d_1} = \sum_{j=1}^{n_1}(d_{1j} - \bar{d_1})/(n_1 - 1)$$

$$s^2_{d_2} = \sum_{j=1}^{n_2}(d_{2j} - \bar{d_2})/(n_2 - 1)$$

(3) 记Δ=真实的平均处理有效性. 检验假设$H_0 : \Delta = 0 \quad vs \quad H_1 : \Delta \neq 0$. 检验统计量

$$t = \frac{\bar{d}}{se} \sim t_{n_1+n_2-2}$$

(4) 判断

$$|t| > t_{n_1+n_2-2,1-\alpha/2}$$

拒绝零假设, 否则接受.

置信区间为

$$\bar{d} \pm t_{n_1+n_2-2,1-\alpha/2}se$$

下面是一个虚拟的例子. 分组为1,2组. 药物为A,B. 第一组先用A, 后用B. 第二组先用B, 后用A. 打分为疼痛减轻的程度. 0为疼痛无减轻, 6为疼痛完全消失. p-值比较大, 说明差异不显著.

```
# 两种药物比较疼痛减轻程度, d1,d2是两组疼痛减轻打分差值.
```

```
>  x_1A=round(runif(10,0,6)) # 第一组用A疼痛减轻的程度
>  x_1B=round(runif(10,0,6)) # 第一组用B疼痛减轻的程度
>  x_2A=round(runif(10,0,6)) # 第二组用A疼痛减轻的程度
>  x_2B=round(runif(10,0,6)) # 第二组用B疼痛减轻的程度
>  d1=x_1A-x_1B
>  d2=x_2B-x_2A
> d1
 [1] -3 -3 -1 -1  3  2 -3 -1  0 -5
> d2
 [1]  1  3  2 -3  0  1  2 -5  0  2
>  d=(mean(d1)+mean(d2))/2
>  d
[1] -0.45
>  se=(9*var(d1)+9*var(d2))/18
>  se
[1] 6.094444
>  t=d/se
>  t
[1] -0.07383774
>  qt(0.025,df=18)
[1] -2.100922
>  p=pt(t,df=18)*2
> p
[1] 0.9419539
```

72.2 剩余效应的估计

如果有剩余效应, 先给药后给安慰剂的组的平均效应大于先给安慰剂后给药的组. 定义

$\bar{x_{ij}} = (x_{ij1} + x_{ij2})/2 =$ 两个处理合并后第j受试者在第i组的平均得分

$$\bar{x}_i = \sum_{j=1}^{n_i} \bar{x_{ij}}/n_i == \text{两个处理合并后第i组的平均得分}$$

我 们 假 定$\bar{x_{ij}} \sim N(u_i, \sigma^2), i = 1, 2, j = 1, \cdots, n_i$, 检 验 假 设

$$H_0 : u_1 = u_2 \quad vs \quad H_1 : u_1 \neq u_2$$

计 算 检 验 统 计 量

$$t = \frac{\bar{x_1} - \bar{x_2}}{\sqrt{s^2(1/n_1 + 1/n_2)}}$$

$$s^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}$$

$$s_i^2 = \sum_{j=1}^{n_i} (x_{ij} - \bar{x_i})^2/(n_i - 1), i = 1, 2$$

如 果

$$|t| > t_{n_1+n_2-2,1-\alpha/2}$$

拒 绝 零 假 设, 否 则 接 受.

TODO: 例 子

72.3 样 本 量 的 估 计

没 有 剩 余 效 应 时, 每 组 近 似 样 本 量 为

$$n = \frac{\sigma_d^2(z_{1-\alpha/2} + z_{1-\beta})^2}{2\Delta^2}$$

σ_d^2为 获 益 得 分 的 方 差, Δ为 两 个 处 理 内 在 的 获 益 的 差, 即$\bar{d}$.

TODO: 例 子

Chapter 73

聚集性的二态数据

请参考 [14] Page 618. 13.11 聚集性的二态数据.

二项比例的两样本检验是最常用的统计方法, 它要求样本中的观察值是统计独立的. 下面的一个例子却不能认为是独立的. Rowe 等报告了一个经典的临床试验, 使用 3% 的阿糖苷(vidarbine)对比安慰剂处理多发性嘴唇疱疹. 在有效的药物期内, 对31个病人的53个损伤性特征使用阿糖苷, 对39个病人的69个损伤性特征使用安慰剂. 都治疗7天, 我们要比较损伤性的比例是否相同. 1个病人身上的多个特征是有关联的.

我们把这种数据称为聚集性数据, 也称为相关性二态数据(correlated binary data).

这时, 随机化单元可以不同于分析使用的单元, 例如, 临床中的随机化是人为单位的, 但是分析单元是以疱疹或牙齿等特征. 例如, 5个学校随机化的取做有效的食物干预组(目的是减少脂肪摄入量), 另外5个学校取做对照. 假设计算结果是1年后干预组脂肪摄入量比对照少30%, 此结果当然是学校的学生计算出的. 同一学校的学生可能有类似的饮食. 那么学生的反应应该是相关性的二态数据.

可以把聚集性二态数据用于基于 Mantel-Haenszel 检验的控制混杂变量上, 也可以扩展到连续变量并做回归分析. 回归分析中, 同一个单元内的观察值之间的相关性收到重视, 这种回归有时称为相关反应模型, 也称为谱系模型, 混合效应模型或

多水平模型.

73.0.1 聚集性数据二项比例的两样本检验

假设我们有两个受试者组, 样本量分别为n_1, n_2, m_{ij}是第i组 第j个个体的提供的观察数, 其中成功了a_{ij}个. 要检验$H_0 : p_1 = p_2 \quad vs \quad H_1 : p_1 \neq p_2$. 计算检验统计量

$$z = [|p_1 - p_2| - (\frac{c_1}{2M_1} + \frac{c_2}{2M_2})]/\sqrt{pq(c_1/M_1 + c_2/M_2)}$$

其 中

$$p_{ij} = a_{ij}/m_{ij}$$

$$p_i = \sum_{j=1}^{n_i} a_{ij} / \sum_{j=1}^{n_i} m_{ij} = \sum_{j=1}^{n_i} m_{ij}p_{ij} / \sum_{j=1}^{n_i} m_{ij} = \text{i组 总 成 功 比 例}$$

$$M_i = \sum_{j=1}^{n_i} m_{ij}$$

$$p = \sum_{i=1}^{2}\sum_{j=1}^{n_i} a_{ij} / \sum_{i=1}^{2}\sum_{j=1}^{n_i} m_{ij} = \sum_{i=1}^{2} M_i p_i / \sum_{i=1}^{2} M_i$$

$$q = 1 - p$$

$$c_i = \sum_{j=1}^{n_i} m_{ij}c_{ij}/M_i = \text{第i组 聚 集 性 修 正 因 子}$$

$$c_{ij} = 1 + (m_{ij} - 1)\rho$$

$$\rho = (MSB - MSW)/[MSB + (m_A - 1)MSW] = \text{类 内 的 相 关 系 数}$$

$$MSB = \sum_{i=1}^{2}\sum_{j=1}^{n_i} m_{ij}(p_{ij} - p_i)^2/(N-2) = \text{个 体 之 间 均 方 误 差}$$

$$MSW = \sum_{i=1}^{2}\sum_{j=1}^{n_i} a_{ij}(1 - p_{ij})/(M - N) = \text{内 部 均 方 误 差}$$

$$m_A = [M - \sum_{i=1}^{2}(\sum_{j=1}^{n_i} m_{ij}^2/M_i)]/(N-2)$$

$$N = n_1 + n_2$$

$$M = \sum_{i=1}^{2} M_i$$

聚 集 性 修 正 因 子 有 时 候 也 称 为 设 计 效 应(design effect).

显 著 性 检 验: 若$|z| > z_{1-\alpha/2}$拒 绝 零 假 设, 否 则 接 受.

对$p_1 - p_2$的近似100%$(1-\alpha)$置信区间为

$$if\ p_1 > p_2, p_1 - p_2 - [c_1/(2M_1) + c_2/(2M_2)] \pm z_{1-\alpha/2}\sqrt{p_1q_1c_1/M_1 + p_2q_2c_2/M_2)}$$

$$if\ p_1 <= p_2, p_1 - p_2 + [c_1/(2M_1) + c_2/(2M_2)] \pm z_{1-\alpha/2}\sqrt{p_1q_1c_1/M_1 + p_2q_2c_2/M_2)}$$

适用条件为$M_1pq >= 5, M_2pq >= 5$.

下面是[14] Page 621, 例 13.52. expose为每个病人暴露牙龈的的数目, damage为龋齿损伤的数目. 11个男性的27个暴露牙龈6个损伤(22.6%), 29个妇女的99个暴露牙龈6个损伤(6.1%). 判断男性和女性的牙面是否有相同的龋齿发病率.

聚集性二项比例的检验p-值=0.186, 差异不显著[1]. 而卡方检验, 精确Fisher检验, 正态方法得到的结果都是p-值=0.03, 差异显著

```
# cat 是为了调试.
aggregation.test<-function(expose,damage,group,alpha=0.05){
   group=factor(group)
   l=levels(group)
   nl=nlevels(group)
   m1j=expose[group==l[1]]
   m2j=expose[group==l[2]]
   a1j=damage[group==l[1]]
   a2j=damage[group==l[2]]
   M1=sum(m1j)
   M2=sum(m2j)
   p1j=a1j/m1j
   p2j=a2j/m2j
   p1=sum(a1j)/M1
   p2=sum(a2j)/M2
   p=sum(damage)/sum(expose)

#cat(p1,p2,p)

   M1=sum(m1j)
```

[1]文献的结果与本结果有差异, 调试发现MSW结果不同, 可能原始数据输入有误

```
    M2=sum(m2j)
    M=M1+M2
    N=length(expose)

#cat("==",M1,M2,M,N,"==\n")

    MSB=(sum(m1j*(p1j-p1)^2)+sum(m2j*(p2j-p2)^2))/(N-2)

    MSW=sum(damage*(1-damage/expose)) / (M-N)
# the same
#MSW=(sum(a1j*(1-p1j))+sum(a2j*(1-p2j))) / (M-N)
    mA=(M-(sum(m1j^2)/M1+sum(m2j^2)/M2))/(N-2)
    rho=(MSB-MSW)/(MSB+(mA-1)*MSW)

#cat("==",MSB,MSW,mA,rho,"==\n")

    C1j=1+(m1j-1)*rho
    C2j=1+(m2j-1)*rho
    C1=sum(m1j*C1j)/M1
    C2=sum(m2j*C2j)/M2
    se=sqrt(p*(1-p)*(C1/M1+C2/M2))
    tmp1=C1/(2*M1)+C2/(2*M2)
    z=(abs(p1-p2)-tmp1)/se

#cat("==",C1,C2,M1,M2,se,tmp1,"==\n")

    z_=qnorm(1-alpha/2)
    se1=sqrt(p1*(1-p1)*C1/M1+p2*(1-p2)*C2/M2)
    CI1=0.0
    CI2=0.0
    delta=p1-p2
    if(delta>0){
       CI1=delta-tmp1-z_*se1
       CI2=delta-tmp1+z_*se1
}
    if(delta<=0){
       CI1=delta+tmp1-z_*se1
       CI2=delta+tmp1+z_*se1
}
    if(M1*p*(1-p)<5 || M2*p*(1-p)<5){
       cat("\ndamage may not enough\n")
```

```
}
   res=list(delta=delta, z=z,p.value=(1-pnorm(abs(z)))*2,conf.int.delta=c(CI1,CI
   res
}

> expose=c(4,1,2,2,4,3,3,3,1,2,2,
   2,6,8,5,4,4,2,4,4,4,6,2,4,4,2,3,2,2,4,2,2,2,2,4,4,4,3,2,2)
> damage=c(0,1,2,0,2,0,0,0,0,1,0,
   1,1,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0)
> sex=c(rep("M",11),rep("F",29))

# 聚集性二项比例的检验
> aggregation.test(expose,damage,sex)
damage may not enough
$delta
[1] -0.1616162

$z
[1] 1.322749

$p.value
[1] 0.1859188

$conf.int.delta
[1] -0.3387652  0.1056469

$alpha
[1] 0.05

# 卡方检验
> chisq.test(x)

       Pearson's Chi-squared test with Yates' continuity correction

data:  x
X-squared = 4.6918, df = 1, p-value = 0.03031

Warning message:
In chisq.test(x) : Chi-squared近似算法有可能不准

# 二项比例齐性检验(与卡方检验一样)
```

```
> prop.test(x)

        2-sample test for equality of proportions with continuity correction

data:  x
X-squared = 4.6918, df = 1, p-value = 0.03031
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.65355134  0.02197239
sample estimates:
   prop 1    prop 2
0.1842105 0.5000000

Warning message:
In prop.test(x) : Chi-squared近似算法有可能不准

# 参考文献的结果. 正态分布计算出的p-值(即prop.test)
> 2*(1-pnorm(2.166))
[1] 0.03031119

# 精确Fisher检验
> fisher.test(x)

        Fisher's Exact Test for Count Data

data:  x
p-value = 0.02077
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.05506787 0.94962880
sample estimates:
odds ratio
 0.2293778
```

73.0.2 样本量及功效估计

假设我们要检验$H_0 : p_1 = p_2 \quad vs \quad H_1 : p_1 \neq p_2$. 如果用双侧检验, 显著性水平为$\alpha$, 功效为$1-\beta$, 则合适的样本量(指每组观察总

数)为

$$M_s = M[1 + (\bar{m} - 1)\rho] = \text{每组观察总数}$$

此处

$$M = (z_{1-\alpha/2}\sqrt{2\bar{p}\bar{q}} + z_{1-\beta}\sqrt{p_1q_1 + p_2q_2})^2/(p_1 - p_2)^2$$
$$\bar{p} = (p_1 + p_2)/2$$
$$\bar{q} = 1 - \bar{p}$$

每组个体数为

$$n = Ms/\bar{m}$$

这里$\bar{m}$为每个受试者平均观察数. ρ为类内相关系数.

如果每组观察总数固定, 对特定备择假设的功效为

$$power = \Phi(z_{1-\beta})$$

此处

$$z_{1-\beta} = \frac{\sqrt{M_s/[1 + (\bar{m} - 1)\rho]}|p_1 - p_2| - z_{1-\alpha/2}\sqrt{2\bar{p}\bar{q}}}{\sqrt{p_1q_1 + p_2q_2}}$$

TODO: 例子

Chapter 74

TODO:测量误差方法

请参考 [14] Page 627. 13.12 测量误差方法, 讲误差对结果的影响.

Chapter 75

人-时间数据及生存分析

75.1 单样本发病率数据的统计推断

75.1.1 大样本方法

假设随访研究过程的t人-年中有a个事件, 且ID=未知的发病密度(率). 检验 $H_0 : ID = ID_0 \quad vs \quad H_1 : ID \neq ID_0$. 计算检验统计量

$$X2 = \frac{(a - u_0)^2}{u_0} \sim \chi_1^2$$

$$u_0 = t * ID_0$$

75.1.2 精确方法

如果事件a太少, 应该使用建立在Poisson分布基础上的精确检验方法. 发病密度(率)即Poisson分布的参数λ此处为$u_0 = t * (ID)$.

注意H_0下, 事件数a服从Poisson分布, 且有参数 $u_0 = t * ID_0$, 则

精确p-值为

$$p = min(2 * \sum_{k=0}^{a} \frac{e^{-u_0} u_0^k}{k!}, 1), if \quad a < u_0$$

$$p = min[2 * (1 - \sum_{k=0}^{a-1} \frac{e^{-u_0} u_0^k}{k!}), 1], if \quad a >= u_0$$

例子([14] Page 650-651 例 14.4 14.6). 1990-1994年建立了一套记录系统, 对还没有乳腺癌但是怀疑有遗传性乳腺癌的妇女作了标记. 500名60-64岁的妇女被识别并随访至2000年末. 整个随访长度为4000人-年. 此期间28例乳腺癌发生. 已知全国60-64岁乳腺癌平均发病率为$400/(10^5)$人-年. (1) 判断这些人乳腺癌发病率与全国是否有差异?

此处$a = 28, u_0 = 4000 * (400/10^5) = 16$, 则检验统计量为

```
> a=28
> u0=4000*(400/10^5)
> X2=(a-u0)^2/u0; X2
[1] 9
> p=1-pchisq(9,df=1); p
[1] 0.002699796
```

(2) 假设500名有遗传学标记的妇女中125人有乳腺癌家族史. 此125人的1000人年共发生8例乳腺癌. 判断这个人群的乳腺癌发病率是否与全国水平一样?

此处$a = 8, u_0 = 1000 * (400/10^5) = 4$, 使用精确方法. p-值为

```
> p=2*(1-ppois(7,4)); p
[1] 0.1022672
```

故没有显著差异. 要想检出差异, 必须加大样本量.

75.1.3 发病率的置信区间

Poisson 分布下, 我们有$\hat{u} = a, var(\hat{u}) = a$. 在t个人年中正态分布近似得发病密度ID的点估计为$\hat{ID} = a/t$, u的双侧估计为$a \pm z_{1-\alpha/2}\sqrt{a} = (c1, c2)$, 若$a < 10$, 使用精确的置信区间. ID的双侧置信区间为$(c1/t, c2/t)$.

ci函数(epicalc包)可以计算 binomial(二项比例), poisson(累加发病率), numeric(均值) 的估计与置信区间(confidence interval).

例如, 上面例子中500人的发病率(发病密度(率)即 Poisson 分布的参数λ此处为$u_0 = t * (ID)$).的点估计为$ID = 28/4000 = 0.007 = 700/10^5$人年. 置信区间为$(28 \pm 1.96\sqrt{28})$. 精确置信区间可以使用ci函数.

```
> c1=28-pnorm(0.975)*sqrt(28); c1
[1] 23.58043
> c2=28+pnorm(0.975)*sqrt(28); c2
[1] 32.41957
> ID1=c1/4000; ID1
[1] 0.005895108
> ID2=c2/4000; ID2
[1] 0.008104892

> ci.poisson(28,4000, alpha=.05) # 500人中的发病率估计
 events person.time incidence         se exact.lower95ci exact.upper95ci
     28       4000     0.007 0.001322876       0.004648       0.010122
#
> ci.poisson(4,1000, alpha=.05)
 events person.time incidence    se exact.lower95ci exact.upper95ci
      4       1000     0.004 0.002       0.001088       0.010244

# ID 的置信区间
> ID.conf.int=ci.poisson(4,1000, alpha=.05)[5:6]*1000; ID.conf.int
 exact.lower95ci exact.upper95ci
         1.088         10.244
```

75.2 两样本发病率数据的统计推断

暴露组	事件数	人-时间数
1	a1	t1
2	a2	t2
总数	a1+a2	t1+t2

我们要比较ID1=组1的真实发病密度(组1单位人-时间的事件发生数) 与ID2=组2的真实发病密度(组2单位人-时间的事件发生数) 是否一样.

零假设下, 两个组可以合并. 一个事件属于组1的个数被看作二项随机变量. 参数$n = a1 + a2, p_0 = t1/(t1 + t2)$. 零假设可以描述为$H_0 : p = p_0(ID1 = ID2)$. 近似正态分布的平均数为$n * p_0 = (a1 + a2)t1/(t1 + t2) = E$, 方差为$np_0q_0 = (a1 + a2)t1t2/(t1 + t2)^2 = V$. 正态分布近似检验统计量为

$$z = \frac{a1 - E - 0.5}{\sqrt{V}}, if \quad a1 > E$$

$$z = \frac{a1 - E + 0.5}{\sqrt{V}}, if \quad a1 <= E$$

$z \sim N(0, 1)$

如果事件数比较小(5), 那么我们使用精确二项分布. p-值为

$$p = 2\sum_{k=0}^{a1} \binom{a1 + a2}{k} p_0^k q_0^{a1+a2-k}, if \quad a1 < (a1 + a2)p_0$$

$$p = 2\sum_{k=a1}^{a1+a2} \binom{a1 + a2}{k} p_0^k q_0^{a1+a2-k}, if \quad a1 >= (a1 + a2)p_0$$

30-34岁妇女乳腺癌与OC使用的关系. 判断使用者和不使用者的发病率的差异显著性. a1=3, a2=9, t1=8250, t2=17430. 计算$V = 2.62 < 5$, 使用精确方法. $n = 3 + 9 = 12, p = 8250/25680 = 0.321, a1 = 3 < 12 * 0.321 = 3.9$, p-值为

使用OC的情况	病例数	人-年数
现在使用者	3	8250
从不使用者	9	17430

```
> 2*pbinom(3,12,prob=8250/25680)
[1] 0.8564199
```

75.3 率比

类似于危险率的比(risk ratio, RR), 那里的单位是人, 我们也可以使用于人-时间数据两个发病率的比较. 记λ_1, λ_2分别是暴露和非暴露组的发病率, 称λ_1/λ_2为率比(rate ratio). 属于 Poisson 分布. 精确的置信区间值来自 binom.test(此处的推导)

率比的点估计为

$$RR = (a1/t1)/(a2/t2)$$

ln(RR)近似于正态分布, 则

$$Var(ln(RR)) = 1/a1 + 1/a2$$

ln(RR)的置信区间为

$$(d_1, d_2) = ln(RR) \pm z_{1-\alpha/2}\sqrt{1/a1 + 1/a2}$$

(d_1, d_2)取反对数既得RR的置信区间.

ln(RR)的精确分布为二项分布.(推导略, 见[50] 公式 1)

对于下面的数据估计率比的点估计和区间估计.

使用OC的情况	病例数	人-年数
现在使用者	9	2935
从不使用者	239	135130

```
> library(rateratio.test)
> t1=2935
> t2=135130
> a1=9
> a2=239
# 精确的置信区间值来自 binom.test
> rateratio.test(c(a1, a2), c(t1, t2))

       Exact Rate Ratio Test, assuming Poisson counts

data:  c(a1, a2) with time of c(t1, t2), null rate ratio 1
p-value = 0.1702
alternative hypothesis: true rate ratio is not equal to 1
95 percent confidence interval:
 0.7831867 3.3470290
sample estimates:
 Rate Ratio     Rate 1     Rate 2
1.733757208 0.003066440 0.001768667

# 与二项分布比例的比较
> binom.test(a1,a1+a2, p = t1/(t1 + t2))

       Exact binomial test

data:  a1 and a1 + a2
number of successes = 9, number of trials = 248, p-value = 0.1160
alternative hypothesis: true probability of success is not equal to 0.02125810
95 percent confidence interval:
 0.01672615 0.06777020
sample estimates:
probability of success
          0.03629032
# 从二项分布计算置信区间
> b.ci=binom.test(a1,a1+a2, p = t1/(t1 + t2))$conf.int
> lambda.ci=t2 * b.ci/(t1 * (1 - b.ci))
> lambda.ci
[1] 0.7831867 3.3470290


>  fisher.test(matrix(c(a1, a2, t1-a1, t2-a2), 2, 2))
```

```
        Fisher's Exact Test for Count Data

data:  matrix(c(a1, a2, t1 - a1, t2 - a2), 2, 2)
p-value = 0.1158
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.7834183 3.3558917
sample estimates:
odds ratio
  1.736001
```

75.4 人-时间数据的功效及样本量估计

实际上是单样本二项检验的特例.

二项比例在指定的假设$p = p_1$下, 双侧检验的功效的正态近似为

$$power = \Phi[\sqrt{(p_0q_0)/(p_1q_1)}(z_{\alpha/2} + |p_0 - p_1|\sqrt{m}/\sqrt{p_0q_0})]$$
$$p_0 = t1/(t1 + t2)$$
$$p_1 = t1RR/(t1RR + t2)$$
$$m = m1 + m2 = \text{两组联合后事件的期望数}$$
$$m1 = n1[1 - exp(-ID_1t1^*)]$$
$$m2 = n2[1 - exp(-ID_2t2^*)]$$
$$n1, n2 = \text{组1,组2的个体数}$$
$$t1, t2 = \text{组1,组2的人-年数}$$
$$t1^*, t2^* = \text{组1,组2的个体平均人-年数}$$
$$ID_1, ID_2 = H_1\text{成立时组1,组2的发病密度}$$

对应于样本量, 两组联合后事件的期望数

$$m = \frac{p_0 q_0 (z_{1-\alpha/2} + z_{1-\beta}\sqrt{(p_1 q_1)})^2}{(p_1 - p_0)^2}$$

$$p_0 = t1/(t1 + t2)$$
$$p_1 = t1RR/(t1RR + t2)$$
$$t1, t2 = \text{组1,组2的人-年数}$$
$$ID_1, ID_2 = H_1\text{成立时组1,组2的发病密度}$$

对应于上述m, 每组个体数分别为

$$n1 = \frac{m}{(k+1) - exp(-ID_1 t1^*) - kexp(-ID_2 t2^*)}$$

$$n2 = kn1$$

假定10000名绝经后妇女, 没有癌症. 5000人随机指定接受雌激素补充疗法(ERT), 另外5000人指定安慰剂. 每个人平均随访5年, 对照组中期望发病率$300/10^5$, 假定ERT可以增加乳腺癌25%的发病率. 求此研究的功效.

下面是按照公式的解

```
power.persontime<-function(p0,p1,m,alpha=0.05){
   power=pnorm( sqrt(p0*(1-p0)/(p1*(1-p1))) * ( qnorm(alpha/2)+abs(p0-p1) * sqrt(
   power
}

> n1=n2=5000
> t_1=t_2=5 # t_1,t_2 为平均人年数
> ID2=300/10^5 # 发病密度/发病率
> ID1=1.25*ID2
> RR=ID1/ID2 # 率比
> m1=n1*(1-exp(-ID1*5)); m1
[1] 92.87656
> m2=n2*(1-exp(-ID2*5)); m2
[1] 74.4403
```

```
> m=m1+m2; m # 总的事件数
[1] 167.3169
> t1=t2=5000*5 # 人年数
> p0=t1/(t1+t2)
> p1=t1*RR/(t1*RR+t2)
> power.persontime(p0,p1,m)
[1] 0.2994232
```

需要多少样本量才会到80%的功效?

```
# t_1,t_2 为平均人年数
n.persontime<-function(p0,p1,ID1,ID2,t_1,t_2,alpha=0.05,power=0.8,k=1){
   m=( sqrt(p0*(1-p0))*qnorm(1-alpha/2) + sqrt(p1*(1-p1))*qnorm(power) )^2 /(p0-p
   n1=m/((k+1)-exp(-ID1*t_1)-k*exp(-ID2*t_2))
   n2=k*n1
   res=list(n1=n1,n2=n2)
   res
}

> n.persontime(p0,p1,ID1,ID2,t_1,t_2,alpha=0.05,power=0.8,k=1)
$n1
[1] 18928.05

$n2
[1] 18928.05
```

75.5 分层的人-时间数据的统计推断

一个研究是绝经后期妇女使用绝经后期激素是否引起心血管疾病和癌症的发生? 从1976年到1986年采用邮寄问卷在每2年随访得到下面的数据. 1976年有23607个绝经后妇女没有癌症, 其它妇女在随访期间都变成绝经后期. 随访在下列条件之一结束: 乳腺癌, 死亡, 到达随访最后. 乳腺癌及绝经后激素的使用与年龄有关, 因此控制年龄很重要.

年龄	(现在使用激素)病例数	人-年	(从不使用激素)病例数	人-年
39-44	12.00	10199.00	5.00	4722.00
45-49	22.00	14044.00	26.00	20812.00
50-54	51.00	24948.00	129.00	71746.00
55-59	72.00	21576.00	159.00	73413.00
60-64	23.00	4876.00	35.00	15773.00

我们可以象对累加发病率数据或计数数据(poisson分布)使用Mantel-Haenszel检验一样, 分析此处数据.

假设疾病与暴露的率比为RR(rate ratio), 我们假定所有层中的$RR = p_{1i}/p_{2i}$是相同的. 要检验假设 $H_0 : RR = 1 \quad vs \quad H_1 : RR = 1$

另外参考cox回归分析. 多个混杂变量时, 此方法也是合适的, 但是比较麻烦, 可以使用 Poisson 回归代替.

```
> a=c(12,22,51,72,23)
> b=c(10199,14044,24948,21576,4876)
> c=c(5,26,129,159,35)
> d=c(4722,20812,71746,73413,15773)
> epi.2by2(a, b, c, d, method = "cohort.time", conf.level = 0.95,verbose=T)

# incidence rate ratio
$IRR
      est      se    lower    upper
1 1.111168 1.702828 0.3914651 3.154033
2 1.253927 1.336004 0.7107125 2.212334
3 1.136953 1.179874 0.8221422 1.572309
4 1.540769 1.152634 1.1663375 2.035404
5 2.125741 1.307897 1.2561175 3.597415

# 直接计算RR
$IRR.crude
      est      se    lower    upper
1 1.253430 1.095866 1.047556 1.499764

# Mantel-Haenszel adjusted RR
$IRR.summary
      est      se lower upper
```

```
1 1.396736 47354.32    0   Inf

# 危险率差 Risk difference (attributable risk)
$AR
          est          se        lower       upper
1 0.0001177126 0.0005827568 -0.0010244697 0.0012598948
2 0.0003172260 0.0004142095 -0.0004946098 0.0011290618
3 0.0002462424 0.0003271105 -0.0003948824 0.0008873672
4 0.0011712122 0.0004291462  0.0003301011 0.0020123233
5 0.0024979993 0.0010526489  0.0004348454 0.0045611533

$AR.crude
          est          se       lower       upper
1 0.0004811295 0.0002040578 8.118347e-05 0.0008810755

$AR.summary
         est se    lower    upper
1 0.000536203  0 -0.792985 0.7940575

# population attributable risk
$PAR
          est          se        lower       upper
1 8.046047e-05 0.0005482703 -0.0009941295 0.0011550505
2 1.278151e-04 0.0003154916 -0.0004905372 0.0007461673
3 6.353295e-05 0.0002105057 -0.0003490507 0.0004761166
4 2.660316e-04 0.0002347413 -0.0001940529 0.0007261161
5 5.898709e-04 0.0005260331 -0.0004411350 0.0016208768

# population attributable fraction
$PAF
        est      lower    upper
1 0.07062062 -1.09729861 0.4820793
2 0.09281504 -0.18655943 0.2511615
3 0.03412920 -0.06129480 0.1031313
4 0.10939426  0.04445151 0.1585549
5 0.21000422  0.08085536 0.2863193

# 暴露与非暴露比例的差异
$chisq
  test.statistic df    p.value
1     0.0392107  1 0.843031819
```

```
2     0.6118946  1 0.434075353
3     0.6017654  1 0.437905228
4     9.3771646  1 0.002197051
5     8.2403647  1 0.004096890

# 暴露与非暴露比例的联合差异
$chisq.summary
  test.statistic df   p.value
1      6.100691  1 0.01351290
```

下面是使用前瞻性方法的结果

```
> r=epi.2by2(a, b, c, d, method = "cohort.count", conf.level = 0.95,verbose=T)
> r
# risk ratio
$RR
      est      se    lower    upper
1 1.111037 1.702333 0.3916422 3.151865
2 1.253530 1.335729 0.7107738 2.210742
3 1.136673 1.179682 0.8222029 1.571420
4 1.538970 1.152392 1.1654570 2.032189
5 2.120456 1.307245 1.2542193 3.584966

$RR.crude
      est      se   lower    upper
1 1.252829 1.077345 1.082623 1.449793

$RR.summary
      est        se   lower    upper
1 1.396736 0.09258788 1.16494 1.674655

# ODDS ratio
$OR
      est      se    lower    upper
1 1.111168 1.703324 0.3912419 3.155832
2 1.253927 1.336278 0.7104259 2.213226
3 1.136953 1.180067 0.8218792 1.572812
4 1.540769 1.152877 1.1658554 2.036245
```

```
5 2.125741 1.308551 1.2548879 3.600940

$OR.crude
      est      se    lower    upper
1 1.253430 1.095977 1.047348 1.500063

$OR.summary
      est       se    lower    upper
1 1.397811 0.1361457 1.070437 1.825306

$AR
         est          se        lower       upper
1 0.0001174499 0.0005817975 -0.0010228523 0.0012577521
2 0.0003163347 0.0004133069 -0.0004937319 0.0011264013
3 0.0002452990 0.0003261382 -0.0003939201 0.0008845181
4 0.0011647941 0.0004271271  0.0003276404 0.0020019477
5 0.0024807669 0.0010457419  0.0004311505 0.0045303832

$AR.crude
         est          se      lower       upper
1 0.0004790778 0.0002033675 8.04849e-05 0.0008776707

$AR.summary
         est          se       lower      upper
1 0.0007177012 0.0003394705 5.235121e-05 0.001383051

$AF
       est     lower    upper
1 0.09994006 -1.5533511 0.6827276
2 0.20225288 -0.4069174 0.5476632
3 0.12023980 -0.2162448 0.3636330
4 0.35021476  0.1419675 0.5079197
5 0.52840334  0.2026913 0.7210573

$PAR
         est          se        lower       upper
1 8.028389e-05 0.0005473838 -0.0009925687 0.0011531365
2 1.274800e-04 0.0003148774 -0.0004896684 0.0007446284
3 6.330109e-05 0.0002099306 -0.0003481553 0.0004747575
4 2.648127e-04 0.0002339375 -0.0001936964 0.0007233217
5 5.869163e-04 0.0005240597 -0.0004402218 0.0016140545
```

```
$PAF
       est      lower    upper
1 0.07054593 -1.09648311 0.4819253
2 0.09269924 -0.18650383 0.2510123
3 0.03406794 -0.06126936 0.1030294
4 0.10915785  0.04424960 0.1583126
5 0.20953926  0.08037758 0.2859365

$chisq
  test.statistic df    p.value
1     0.0392107  1 0.843031819
2     0.6118946  1 0.434075353
3     0.6017654  1 0.437905228
4     9.3771646  1 0.002197051
5     8.2403647  1 0.004096890

$chisq.summary
  test.statistic df   p.value
1      6.100691  1 0.01351290

# RR的齐性检验
$RR.homog
  test.statistic df   p.value
1      4.774888  4 0.3111848

# OR的齐性检验
$OR.homog
  test.statistic df   p.value
1      4.002160  4 0.4057135
```

75.6 分层的人-时间数据的功效及样本量

TODO: [14] Page 671, 14.6

75.7 发病率数据中趋势性的检验

TODO: [14] Page 676, 14.7

Chapter 76

生存分析

前面的发病率比较中, 一个假设是发病率不随时间变化. 但是在许多情况下这个假设是不能保证的. 这样就产生了生存分析.

R-cran 网站的介绍生存分析的页面很好 http://cran.r-project.org/web/views/Survival.html

(参考 prodlim , survival 包)

76.1 概念

76.1.1 危险率(hazard rate)

可以随时间变化的发病率称为危险率(hazard rate)

76.1.2 死亡危险率(mortality risk)

生物统计中危险率函数常被看作是死亡危险率的(mortality risk)指标

76.1.3 生存概率(survival probability)

不发生疾病的概率通常称为生存概率(survival probability)

76.1.4 生存函数(survival function)

将生存概率记为时间的函数, 即对每个$t >= 0$的点, 可以存活到时间t以上的概率称为生存函数(survival function).

76.1.5 危险函数(hazard function)

h(t)是单位时间内时刻t上一个事件瞬时发生的概率, 即一个到t时刻存活的个体(即还没有发生事件)在t时的瞬时发病率. 特例

$$h(t) = \lim_{\Delta t \to 0}[\frac{S(t) - S(t + \Delta t)}{\Delta t}]/S(t)$$

例如, 0岁(出生时)的100000名男性, 80908名活到60岁, 79539名活到61岁, 则60岁的危险率近似为

$$h(60) = \frac{80908 - 79539}{80908} = \frac{1369}{80908} = 0.017$$

即60岁存活的男性, 下一年有1.7%的人会死亡.

76.1.6 失访或截尾观察(censored observation)

随访周期内未达到疾病终点的病人称为失访或截尾观察(censored observation). 一个病人在随访到时刻t时失访, 称为t时失访. 即缺失状态. problem 函数可以计算失访.

76.2 时间序列的 Kaplan-Meier 估计

Kaplan-Meier 估计又叫做乘积限(product limit)估计

Kaplan与Meier(1985)提供了一种从缺失(loss)数据中获取信息的方法, 即在缺失前死亡(death)还没有发生.([19] Page 62) 一个事实是: 若死亡发生在时刻x后, 那么很显然也发生在x前的任意时刻之后. 由条件概率的定义, 对于$x_0 < x_1$, 我们有

$$P(X > x_1) = P(X > x_1, X > x_0) = P(X > x_1 | X > x_0) P(X > x_0)$$

假设第一年初有100个研究对象, 年底剩下30个存活. 我们用下式估计$P(1)$

$$P(1) = P(X > 1) = 30/100 = 0.3$$

这里X表示研究对象个体的寿命.

第二年初又有另外1000个个体参加试验. 第二年底, 1000个中有250个存活, 而最初的100个中存活的只有10个了. 我们可以使用最初的100个个体来估计$P(2)$

$$P(2) = P(X > 2) = 10/100 = 0.1$$

但是我们可以用第二年新参加的个体信息来更新估计$P(1)$. 因为到第二年底参加了1年的个体共有1100个, 其中共有$250 + 30 = 280$个存活, 改进后的$P(1)$的估计为

$$P(1) = P(X > 1) = 280/1100 = 0.255$$

由条件概率, 我们使用改进后的$P(1)$来改进$P(2)$

$$P(2) = P(X > 2) = P(X > 2 | X > 1) P(X > 1)$$

不幸的是, 我们无法改进$P(X > 2|X > 1)$, 因为第三年的试验还没有做, 我们不知道在接下来的1年1000个个体有多少存活.故我们使用下面的估计量(它仅用到了已知信息. 即第一年底有30个存活, 第二年底有10个存活)

$$P(X > 2|X > 1) = 10/30$$

那么$P(2)$的改进为

$$P(2) = P(X > 2|X > 1) P(X > 1) = \frac{10}{30} \frac{280}{1100} = 0.085$$

Kaplan与Meier推广了上面的方法. 设$u_1 < u_2 < ... < u_k$表示k个个体的寿命(从开始到死亡, 或缺失的持续时间). 令

$$p_i = P(X > u_i | X > u_{i-1}$$

用下式估计

$$p_i = \frac{\text{到时刻}u_i\text{存活的个体数}}{\text{到时刻}u_{i-1}\text{仍然观测到的存活的个体数}}$$

在时刻u_i缺失的个体, 可以认为在时刻u_i以后仍然存活. 第一次死亡或缺失的计算中, P_1的分母是个体的总数.

$P(x)$的Kaplan-Meier估计为

$$P(x) = \begin{cases} 1 & if x < u_1 \\ \prod_{u_i <= x} p_i & if x >= u_i \end{cases}$$

有时候需要求出删失数据的方差.

下面是另外一个例子. 要测试10个汽车风扇皮带的质量. 我们记录每个皮带所能承受的里程数. 测试结束后, 5个带都断了, 寿命分别为77,47,81,56,80(千英里). 另外5个没有断, 分别是62,60,43,71,37. 那么生存函数Kaplan-Meier估计如下

	u	r	p_i	$P(u_i)$
1	37.00	loss	10/10	1.00
2	43.00	loss	9/9	1.00
3	47.00	death	7/8	0.88
4	56.00	death	6/7	0.75
5	60.00	loss	6/6	0.75
6	62.00	loss	5/5	0.75
7	71.00	loss	4/4	0.75
8	77.00	death	2/3	0.50
9	80.00	death	1/2	0.25
10	81.00	death	0/1	0.00

survfit函数用法: 使用 Kaplan-Meier 方法计算校验数据(censored data)的生存曲线, 或使用 Fleming-Harrington 方法计算Cox比例风险模型的预测生存函数. status:

下面是survival包的结果, 与手工计算一致

```
> f= survfit(Surv(time, status) )
> summary(f)
Call: survfit(formula = Surv(time, 1 - status))

 time n.risk n.event survival std.err lower 95% CI upper 95% CI
   47      8       1    0.875   0.117       0.673           1
   56      7       1    0.750   0.153       0.503           1
   77      3       1    0.500   0.228       0.204           1
   80      2       1    0.250   0.210       0.048           1
   81      1       1    0.000      NA          NA          NA
> plot(f) # 同时绘制上界和下界
```

下面是R prodlim 包的结果, 和手工计算一致.

```
> time=c(37,43,47,56,60,62,71,77,80,81)
# 缺失的状态设为0, 死亡的设为1
> status=c(0,0,1,1,0,0,0,1,1,1)
> fit=prodlim(Hist(time,status)~1)
> summary(fit) # surv 为概率, 后面依次为surv的标准误, 下界, 上界
   n.risk n.lost n.event  surv   se.surv     lower    upper
37     10      1       0 1.000 0.0000000 1.00000000 1.0000000
43      9      1       0 1.000 0.0000000 1.00000000 1.0000000
47      8      0       1 0.875 0.1169268 0.64582770 1.0000000
56      7      0       1 0.750 0.1530931 0.44994302 1.0000000
60      6      1       0 0.750 0.1530931 0.44994302 1.0000000
62      5      1       0 0.750 0.1530931 0.44994302 1.0000000
71      4      1       0 0.750 0.1530931 0.44994302 1.0000000
77      3      0       1 0.500 0.2282177 0.05270146 0.9472985
80      2      0       1 0.250 0.2104064 0.00000000 0.6623889
81      1      0       1 0.000       NaN       NaN      NaN

> plot(fit) # 绘出生存函数的图像
> fit

Call: prodlim(formula = Hist(time, status) ~ 1)
```

```
Kaplan-Meier estimator for the event time survival function

No covariates

RightCensored response of a survival model

No.Observations: 10

Pattern:
              Freq
 event         5
 right.censored 5
```

下面是另外一个例子

```
# 原始数据为
      MR组       对照组
时间
0hr 活：6；  死：0 活：12；  死：0
3hr 活：6；  死：0 活：11；  死：1
4hr 活：6；  死：0 活：9：  死：2
6hr 活：6；  死：0 活：7；  死：2
24hr 活：6；  死：0 活：7；  死：0
48hr 活：6；  死：0 活：5；  死：2
72hr 活：6；  死：0 活：5；  死：0
150hr 活：6；  死：0 活：5；  死：0

# 构造生存表，0表示到达此时失访(还存活)
> data
time status x
150 0 MR
150 0 MR
150 0 MR
150 0 MR
150 0 MR
150 0 MR
3 1 CTL
4 1 CTL
```

```
4 1 CTL
6 1 CTL
6 1 CTL
48 1 CTL
48 1 CTL
150 0 CTL
150 0 CTL
150 0 CTL
150 0 CTL
150 0 CTL

# survival包
> f= survfit(Surv(time, status)~x,data=data )
> summary(f)
Call: survfit(formula = Surv(time, status) ~ x, data = data)

                x=CTL
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    3     12       1    0.917  0.0798        0.773        1.000
    4     11       2    0.750  0.1250        0.541        1.000
    6      9       2    0.583  0.1423        0.362        0.941
   48      7       2    0.417  0.1423        0.213        0.814

# 使用prodlim包
> fit=prodlim(Hist(time,status)~x,data=data)
> summary(fit)
x=CTL :
    n.risk n.lost n.event      surv   se.surv    lower    upper
3       12      0       1 0.9166667 0.0797856 0.7602898 1.0000000
4       11      0       2 0.7500000 0.1250000 0.5050045 0.9949955
6        9      0       2 0.5833333 0.1423188 0.3043937 0.8622730
48       7      0       2 0.4166667 0.1423188 0.1377270 0.6956063
150      5      5       0 0.4166667 0.1423188 0.1377270 0.6956063

x=MR  :
    n.risk n.lost n.event surv se.surv lower upper
3        6      0       0    1       0     0     1
4        6      0       0    1       0     0     1
6        6      0       0    1       0     0     1
48       6      0       0    1       0     0     1
150      6      6       0    1       0     1     1
```

```
> plot(fit)
```

76.3 对数秩(log rank)检验

累加发病率如果随时间不同, 则前面介绍的两个发病率之间的比较的方法不是很有效. 我们将使用对数秩检验方法(此方法与"对数"完全没有关系)检验两个生存曲线的发病率是否相同. [1]

下面表的行为年龄, 列为戒烟天数的人数, 例如大于40岁的戒烟天数小于90天的人数为92, 即92人在小于90天内又开始吸烟.

年龄/戒烟天数	<= 90	91-180	181-270	271-364	365
> 40	92	4	4	1	19
<= 40	88	7	3	2	14

我们把时间分段, 数据归为4个列联表, 然后使用Mantel-Haenszel检验, 就得到对数秩检验. 如果检验统计量chisq比较小, 则接受零假设, 此处 X-squared = 0.2932, df = 1, p-value = 0.5882, 即两个年龄的恢复吸烟的发病率上没有显著不同. [2]

[1]关于logrank名称的解释(下面资料来自网络): SAS的"LOG窗口"的中文意思是"对数窗",因为生存分析的Log rank在网络上就被译为"对数秩"。不信? 在Google里用"对数秩"检索, 至少可见四五个页面都是"对数秩 (log rank)", 其中也有很出名的院校的统计教学计划, 还有教材、辅导、教学大纲, 更有著名杂志和期刊。学学生存分析的log rank检验, 就知道log rank检验和"对数"毫无关系, log rank检验的LOG是SAS"LOG窗口"LOG, 非"对数"LOG。如果Log rank译为"对数秩", SAS的"LOG窗口"当然就是"对数窗"了。log还有登录, 日志的意思.

最近一本翻译的美国生物统计教材, 也把Log rank译为"对数秩", 正式出版物, 或许也不算错。

log rank可以翻译成"时序秩", 更切合生存分析的用途, 也比较合本意!

[2]TODO: 此处结果与survdiff及surv_test函数的结果不同, 差距较大, 不知为何? 可能是我的数据重构有问题

```
> x=array(c(92,88,28,26,4,7,24,19,4,3,20,16,1,2,19,14),
       dim=c(2,2,4),
       dimnames=list(c(">40","<=40"),
          c("恢复抽烟","继续戒烟"),
          c("0-90天","91-180天","181-270天","271-365天")))
> x
, , 0-90天

    恢复抽烟 继续戒烟
>40       92       28
<=40      88       26

, , 91-180天

    恢复抽烟 继续戒烟
>40        4       24
<=40       7       19

, , 181-270天

    恢复抽烟 继续戒烟
>40        4       20
<=40       3       16

, , 271-365天

    恢复抽烟 继续戒烟
>40        1       19
<=40       2       14

# 对数秩检验
> mantelhaen.test(x)

       Mantel-Haenszel chi-squared test with continuity correction

data:  x
Mantel-Haenszel X-squared = 0.2932, df = 1, p-value = 0.5882
alternative hypothesis: true common odds ratio is not equal to 1
95 percent confidence interval:
 0.5036551 1.3988182
sample estimates:
```

```
common odds ratio
        0.839358
```

下面给出另外几个例子(取自[28]). 注意数据需要是data.frame, time 为存活时间, event=TRUE 为死亡, =FALSE 为中途缺失, group 为分组.

```
> data("glioma", package = "coin")
> g3 <- subset(glioma, histology == "Grade3")
> g3
   no. age    sex histology   group event time
1    1  41 Female    Grade3    RIT  TRUE   53
2    2  45 Female    Grade3    RIT FALSE   28
3    3  48   Male    Grade3    RIT FALSE   69
4    4  54   Male    Grade3    RIT FALSE   58
5    5  40 Female    Grade3    RIT FALSE   54
6    6  31   Male    Grade3    RIT  TRUE   25
7    7  53   Male    Grade3    RIT FALSE   51
8    8  49   Male    Grade3    RIT FALSE   61
9    9  36   Male    Grade3    RIT FALSE   57
10  10  52   Male    Grade3    RIT FALSE   57
11  11  57   Male    Grade3    RIT FALSE   50
20   1  27   Male    Grade3 Control  TRUE   34
21   2  32   Male    Grade3 Control  TRUE   32
22   3  53 Female    Grade3 Control  TRUE    9
23   4  46   Male    Grade3 Control  TRUE   19
24   5  33 Female    Grade3 Control FALSE   50
25   6  19 Female    Grade3 Control FALSE   48
> survdiff(Surv(time, event) ~ group, data = g3)
Call:
survdiff(formula = Surv(time, event) ~ group, data = g3)

               N Observed Expected (O-E)^2/E (O-E)^2/V
group=Control  6        4     1.49     4.23     6.06
group=RIT     11        2     4.51     1.40     6.06

 Chisq= 6.1  on 1 degrees of freedom, p= 0.0138
>  library("coin")
> surv_test(Surv(time, event) ~ group, data = g3,distribution = "exact")
```

```
        Exact Logrank Test

data:  Surv(time, event) by group (Control, RIT)
Z = 2.1711, p-value = 0.02877
alternative hypothesis: two.sided
```

一些计算对数秩相关的函数: survival: survdiff(surv.marr sex)

survival: summary(clogit(case alcohol + strata(matset)))

coin: surv_test(Surv(time, event) stadium, data = ocarcinoma)

Hmisc: cpower(2, 1000, .2, 25, accrual=2, tmin=1, noncomp.c=10, noncomp.i=17.5)

76.4 Cox比例风险回归模型

当有多个危险因素, 又有多层时, 方便的方法是对生存数据使用回归模型, 常用的是Cox比例风险回归模型(Cox proportional hazards regression model).

76.4.1 模型及检验

在模型中, 危险率可以表示为

$$h(t) = h_0(t)exp(b_1x_1 + \cdots + b_kx_k)$$

此处$x_1, \cdots, x_k$是一组独立变量, $h_0(t)$是基准状态下t时刻的基准危险率, 它代表所有变量全部取0时的危险率. 假设$H_0 : b_i = 0 \quad vs \quad H_1 : b_i \neq 0$. 对此的检验方法为:

(1) 计算检验统计量$z = \hat{b_i}/se(\hat{b_i}) \sim N(0, 1)$

(2) 判断显著性, $|z| > z_{1-\alpha/2}$, 拒绝零假设, 否则接受

我们把方程变形, 可以写作

$$ln[\frac{h(t)}{h_0(t)}] = b_1x_1 + \cdots + b_kx_k$$

我们可以按照多重logistic回归模型的方式去解系数b_i, 特别在x为二态独立变量时.

76.4.2 对二态独立变量危险比的估计

设有一个二态危险因子x_i, 当危险存在时$x_i = 1$, 不存在时$x_i = 0$, 量$exp(b_i)$代表了如下两个个体的危险率之比: 在其它协变量全部相同的情况下, 一个个体有x_i出现($x_i = 1$)而另外一个没有($x_i = 0$), 这个危险率之比可以称为相对危险率, 可以看作其它协变量全部相同时, 在t时刻有危险因子($x_i = 1$)相对于没有危险因子($x_i = 0$)的个体在单位时间内发生事件的相对危险率. b_i的双侧$100\%(1-\alpha)$CI为(e^{c1}, e^{c2})

$$c1 = \hat{b_i} - z_{1-\alpha/2}se(\hat{b_i})$$

$$c2 = \hat{b_i} + z_{1-\alpha/2}se(\hat{b_i})$$

76.4.3 对连续独立变量危险比的估计

设有一个连续的危险因子x_i, 两个个体在其它协变量全部相同, 仅在第i个独立变量(危险因子)x_i上相差Δ, 则量$exp(b_i\Delta)$两个个体的危险率比. 可以看作其它协变量全部相同时, 在t时刻一个危险因子取$x_i + \Delta$另外一个危险因子取x_i在单位时间内发生事件的瞬时相对危险率. $b_i\Delta$的双侧$100\%(1-\alpha)$CI为(e^{c1}, e^{c2})

$$c1 = \Delta[\hat{b_i} - z_{1-\alpha/2}se(\hat{b_i})]$$

$$c2 = \Delta[\hat{b_i} + z_{1-\alpha/2}se(\hat{b_i})]$$

它可以看作是多重 logistic 回归的拓广: 即事件发生与时间有关, 而不是简单考察事件是否发生.

由于没有数据, 我使用[28] 的例子

```
> library(ipred)
> data(GBSG2)
> GBSG2
   horTh age menostat tsize tgrade pnodes progrec estrec time cens
1     no  70     Post    21     II      3      48     66 1814    1
2    yes  56     Post    12     II      7      61     77 2018    1
3    yes  58     Post    35     II      9      52    271  712    1
4    yes  59     Post    17     II      4      60     29 1807    1
5     no  73     Post    35     II      1      26     65  772    1
6     no  32      Pre    57    III     24       0     13  448    1
......
683   yes  53     Post    25    III     17       0      0  186    0
684    no  51      Pre    25    III      5      43      0  769    1
685    no  52     Post    23     II      3      15     34  727    1
686    no  55     Post    23     II      9     116     15 1701    1

# 对去除了 time, cens 的所有其它变量进行回归
# exp(coef) 即其它协变量全部相同时, 在t时刻的瞬时相对危险比
> coxph(Surv(time, cens) ~ ., data = GBSG2)
Call:
coxph(formula = Surv(time, cens) ~ ., data = GBSG2)


                coef exp(coef) se(coef)     z      p
horThyes    -0.346278     0.707 0.129075 -2.683 7.3e-03
age         -0.009459     0.991 0.009301 -1.017 3.1e-01
menostatPost  0.258445     1.295 0.183476  1.409 1.6e-01
tsize        0.007796     1.008 0.003939  1.979 4.8e-02
tgrade.L     0.551299     1.736 0.189844  2.904 3.7e-03
tgrade.Q    -0.201091     0.818 0.121965 -1.649 9.9e-02
pnodes       0.048789     1.050 0.007447  6.551 5.7e-11
progrec     -0.002217     0.998 0.000574 -3.866 1.1e-04
estrec       0.000197     1.000 0.000450  0.438 6.6e-01

Likelihood ratio test=105  on 9 df, p=0  n= 686
```

76.4.4 功效及样本量估计

TODO: [14] Page 699, 14.12

Part IX

时间序列与信号处理

主要参考 [46] chapter 15, Time series

Chapter 77

时间序列相关的概念

77.1 Hermitian 矩阵与函数

77.1.1 Hermitian 矩阵

若矩阵的值符合 $a_{ij} = \bar{a_{ji}}$, 此矩阵为 Hermitian 矩阵, 即矩阵本身与其共轭转置一样.

对于实矩阵, 实际上就是实对称矩阵. 例如

$$\begin{bmatrix} 3 & 2+i \\ 2-i & 1 \end{bmatrix}$$

求矩阵 A 的 Hermitian 矩阵

```
Conj(t(A))
```

77.1.2 Hermitian 函数

Hermitian 函数是复函数, 如果复共轭等于原始值的相反数. 实部为偶函数, 虚部为奇函数.

$$f(-x) = f(\bar{x})$$

两个参数的也可以.

$$f(-x1, -x2) = f(x\bar{1}, x2)$$

77.2 自相关(Auto-correlation, ACF)

参考 http://en.wikipedia.org/wiki/Autocorrelation

时间序列数据是不独立的. 我们首先可以看看它的自相关函数: AutoCorrelation Function (ACF). 严格来讲, 自相关分为样本自相关和理论自相关, 分别来自样本数据和理论模型. 延迟 k 的自相关是观测 n 与观测 n-k 之间的相关. 可以假设自相关只和 k 有关, 和 n 无关.

77.2.1 定义

$$R(t,s) = \frac{E[(X_t - \mu_t)(X_s - \mu_s)]}{\sigma_t \sigma_s}$$

若定义 $\tau = t - s$, 则写作熟悉的方式

$$R(\tau) = \frac{E[(X_t - \mu)(X_{t+\tau} - \mu)]}{\sigma^2}$$

这实际上是偶函数, 写作

$$R(\tau) = R(-\tau)$$

对于离散序列 $X_1, \cdots, X_n$, 自回归为

$$R(k) = \frac{1}{(n-k)\sigma^2} \sum_{t=1}^{n-k} [X_t - \mu][X_{t+k} - \mu]$$

若 μ, σ 已知, 此为无偏估计. 但若使用样本均值和方差代替是有偏估计.

77.2.2 例子

acf() 函数参数 lag.max 默认 $10 * log10(N/m)$. N 为观测个数, m 为序列个数, 此处为 1. [1]

```
# my.acf() 函数计算自回归
my.acf <- function (
    x,
    lag.max = ceiling(5*log(length(x)))
){
  m <- matrix(
    c( NA,
       rep( c(rep(NA, lag.max-1), x),
            lag.max ),
       rep(NA,, lag.max-1)
     ),
    byrow=T,
    nr=lag.max)
  x0 <- m[1,]
  apply(m,1,cor, x0, use="complete")
}

# 计算自回归
> x=1:10
> my.acf(x,lag.max=3)
[1] 1 1 1

# 函数的矩阵 m 是这样
```

[1] 貌似[46] 15.1.4 的 my.acf() 函数不正确, 见计算

```
> lag.max=3
> m <- matrix(
+   c( NA,
+     rep( c(rep(NA, lag.max-1), x),
+         lag.max ),
+     rep(NA,, lag.max-1)
+    ),
+   byrow=T,
+   nr=lag.max)
> m
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]   NA   NA   NA    1    2    3    4    5    6    7
[2,]   NA   NA    1    2    3    4    5    6    7    8
[3,]   NA    1    2    3    4    5    6    7    8    9

# my.acf 实际使用的函数
# lag = 1
> cor(m[1,],m[2,],use="complete.obs")
[1] 1
# lag = 2
> cor(m[1,],m[3,],use="complete.obs")
[1] 1

=======================
# 按照公式手工计算, 与 R 函数一致
> u=mean(x)
> v=var(x)
> v
[1] 9.166667
# lag = 1,2 3
> sum((x[1:9]-u)*(x[2:10]-u))/(9*v)
[1] 0.7
> sum((x[1:8]-u)*(x[3:10]-u))/(9*v)
[1] 0.4121212
> sum((x[1:7]-u)*(x[4:10]-u))/(9*v)
[1] 0.1484848

=======================
# R 函数计算
> a=acf(x,lag.max=3);a
```

```
Autocorrelations of series ‘x’, by lag

   0    1    2    3
1.000 0.700 0.412 0.148

# 真实的例子. lag.max=19
x <- LakeHuron
acf(x, main="ACF of a time series (Lake Huron)")
```

77.3 互相关(Cross-correlation, CCF)

参考 http://en.wikipedia.org/wiki/Cross-correlation

77.3.1 定义

连续函数的互相关为[2]

$$(f \star g)(t) = \int_{-\infty}^{\infty} f^*(\tau) g(t+\tau) d\tau$$

其中 f^* 为复共轭

类似, 离散互相关为

$$(f \star g)[n] = \int_{m=-\infty}^{\infty} f^*[m] g[n+m]$$

自相关是序列对自身的互相关.

标准化的互相关为

$$\frac{1}{(n-1)\sigma_f \sigma_g} \sum (f - \bar{f})(g - \bar{g})$$

[2]我们使用符号 $\star$ 表示互相关. $*$ 表示卷积

实际上是序列 f,g 标准化后的内积除以其 L^2 范数.

77.3.2 性质

- 互相关与卷积的关系

$$(f \star g)(t) = f^*(-t) * g(t)$$

- 若 f,g 都是 Hermitian 的, 那么互相关等于卷积:

$$f \star g = f * g$$

-

$$(f \star g) \star (f \star g) = (f \star f) \star (g \star g)$$

- 与卷积一样,

$$F(f \star g) = F(f)^* \cdot F(g)$$

  其中 F 为傅立叶变换.

- f,h 卷积与 g 的互相关等于 h 与 f,g 互相关的卷积

$$(f * h) \star g = h * (f \star g)$$

77.3.3 例子

```
  x<-1:10
  y=c(3,4,5,1,2,3,6,7,8,9)
> y1=scale(y)
> c=ccf(x,y,plot=F);c

Autocorrelations of series 'X', by lag

   -6    -5    -4    -3    -2    -1     0     1     2     3     4
-0.378 -0.117  0.181  0.503  0.523  0.609  0.745  0.427  0.126 -0.144 -0.367
    5     6
-0.380 -0.291
```

```
# =================
#内 积(点 积)
> x1[,]%*%y1[,]
        [,1]
[1,] 6.709354
# =================
# lag=0
> x1[,]%*%y1[,]/9
         [,1]
[1,] 0.7454838
# =================
# lag=1
> y1[1:9]%*%x1[2:10]/9
         [,1]
[1,] 0.4265824
# lag=2
> y1[1:8]%*%x1[3:10]/9
         [,1]
[1,] 0.1256278
# lag=6
> y1[1:4]%*%x1[7:10]/9
          [,1]
[1,] -0.2912909
# =================
# lag=-1
> x1[1:9]%*%y1[2:10]/9
         [,1]
[1,] 0.6088118
# lag=-2
> x1[1:8]%*%y1[3:10]/9
         [,1]
[1,] 0.5232192
# lag=-6
> x1[1:4]%*%y1[7:10]/9
         [,1]
[1,] -0.378264
```

77.4 偏自相关(Partial Autocorrelation, PACF)

参考 http://www.qualityamerica.com/knowledgecente/knowctrPartial_Autocorrelation_Func

Partial Autocorrelation Function(PACF): The Partial Autocorrelation at the given lag. The PACF will vary between -1 and +1, with values near 1 indicating stronger correlation. The PACF removes the effect of shorter lag autocorrelation from the correlation estimate at longer lags. This estimate is only valid to one decimal place.

$$\Phi_{m,m} = \frac{r_m - \sum_{j=1}^{m-1} \Phi_{m-1,j} r_{m-1}}{1 - \sum_{j=1}^{m-1} \Phi_{m-1,j} r_j}$$

其中 r_m 是自相关函数.

pacf() 计算偏自相关.

77.5 卷积(Convolution)

参考 http://en.wikipedia.org/wiki/Convolution

[3]

77.5.1 定义

$$\begin{aligned}(f * g)(t) &= \int_{-\infty}^{\infty} f(\tau) g(t-\tau) d\tau \\ &= \int_{-\infty}^{\infty} f(t-\tau) g(\tau) d\tau\end{aligned}$$

[3]我们使用符号 $\star$ 表示互相关. $*$ 表示卷积

更一般的, 若 f,g 为空间 R^d 的复函数, 其卷积为

$$\begin{aligned}(f*g)(x) &= \int_{R^d} f(y)g(x-y)dy \\ &= \int_{R^d} f(x-y)g(y)dy\end{aligned}$$

循环卷积: 若 g_T 为周期函数, f 不是. 则

$$(f*g_T)(t) = \int_{t_0}^{t_0+T}[\sum_{k=-\infty}^{\infty} f(\tau+kT)]g_T(t-\tau)d\tau$$

离散卷积:

$$\begin{aligned}(f*g)[n] &= \sum_{m=-\infty}^{\infty} f[m]g[n-m] \\ &= \sum_{m=-\infty}^{\infty} f[n-m]g[m]\end{aligned}$$

循环离散卷积: 若 g_N 为周期函数, f 不是. 则

$$(f*g_N)[n] = \sum_{m=0}^{N-1}(\sum_{k=-\infty}^{\infty} f[m+kN])g_N[n-m]$$

当 f,g 都在 $[0, N-1]$ 有定义, 则循环离散卷积变为

$$\begin{aligned}(f * g_N)[n] &= \sum_{m=0}^{N-1} f[m] g_N[n-m] \\ &= \sum_{m=0}^{n} f[m] g[n-m] + \sum_{m=n+1}^{N-1} f[m] g[N+n-m] \\ &= \sum_{m=0}^{N-1} f[m] g[(n-m)_{mod N}] = (f *_N g)[n]\end{aligned}$$

其中 $(f *_N g)[n]$ 表示对整数 N 卷积.

快速计算: 根据卷积定理, 利用快速傅立叶变换(fft)计算卷积.

77.5.2 性质(不全)

- 可交换(Commutativity)
$$f * g = g * f$$
- 结合(Associativity)
$$f * (g * h) = (f * g) * h$$
- 分配(Distributivity)
$$f * (g + h) = (f * g) + (f * h)$$
- 系数
$$a(f * g) = (af) * g = f * (ag)$$
- δ 为冲击函数
$$f * \delta = f$$
- 卷积定理(F为傅立叶变换)
$$F(f * g) = k \cdot F(f) F(g)$$
- 与反函数卷积(记$f^{(-1)}$为f的反函数)
$$f^{(-1)} * f = \delta$$

77.5.3 例子

R 函数 convolve() 使用 fft 计算卷积. 类型为非循环($type = 'open'$), 循环($type =$ "$circular$"). 默认为循环卷积.

非循环时, 设

```
'r <- convolve(x,y, type = "open")'
'n <- length(x)'
'm <- length(y)'
```

那么

```
r[k] = sum(i; x[k-m+i] * y[i])
k = 1,..., n+m-1
```

对所有能够成立的 i (即不超出index范围). 里面有一些重复计算的步骤, 如果可以充分利用, 我们可以设计一个精巧的算法(例如象 fft 的蝴蝶算法?)

```
> x=1:10
> y=11:15
> convolve(x,y,t='o')
 [1]  15  44  86 140 205 270 335 400 465 530 430 326 219 110
```

```
# 手工计算
# r1=15
r1=x[1-5+5]*y[5]

# r2=44
r2=x[2-5+4]*y[4]+
   x[2-5+5]*y[5]

# r3=86
```

```
r3=x[3-5+3]*y[3]+
   x[3-5+4]*y[4]+
   x[3-5+5]*y[5]

......

# 总结算式
my_open_conv <-function(x,y,k){
   n<-length(x)
   m<-length(y)
   i=1:m
   a=k-m+i
   a=a[a>0 & a<=n] # 保证下标不越界
   sum(x[a]*y[i[k-m+i>0][1:length(a)]])
}
> c=c()
> for (i in 1:14) c=append(c,my_open_conv(x,y,i));c
 [1]  15  44  86 140 205 270 335 400 465 530 430 326 219 110
```

如果是循环卷积, 那么需要 x,y 的长度一样. 上面的算法还是有效的,

```
r[k] = sum(i; x[k-m+i] * y[i])
k = 1,..., n

> x=1:5
> y=2:6
> convolve(x,y)
[1] 70 60 55 55 60

my_circular_conv <-function(x,y,k){
   n<-length(x)
   m<-length(y)
   if (n != m) stop('length x,y must be same')
   i=1:m
# 求模. 根据公式应该是 a=(k-m+i)%%5, 但是 R 的结果为下面才对
   a=(k-m+i-1)%%n
   a[a==0] = a[a==0]+n
```

```
   sum(x[a]*y)
}
# 确实是循环卷积
> c=c()
> for (i in 1:20) c=append(c,my_circular_conv(x,y,i));c
 [1] 70 60 55 55 60 70 60 55 55 60 70 60 55 55 60 70 60 55 55 60
```

77.6 白噪声(white noise)及其检验

残差的随机性检验在建立模型时非常重要.

正态分布的随机数就是白噪声.

```
rnorm(n)
```

如何判断一个序列是白噪声? 下面是几种方法.

77.6.1 ACF系数

看看 ACF, 如果自相关系数迅速衰减, 就可能是白噪声

```
> z <- rnorm(200)
> op <- par(mfrow=c(2,1), mar=c(5,4,2,2)+.1)
> plot(ts(z))
> acf(z, main = "")
> par(op)
```

77.6.2 Box–Pierce(Ljung–Box) test

此检验考察自相关系数的和服从卡方分布. Ljung–Box 检验对于小样本给出更好的卡方近似. 也叫做 portmanteau test.

零假设为: 给定序列是时间独立的.

```
> x=seq(0,10,by=0.1)
> y=cos(2*pi*x)+0.2*sin(x)+2*cos(x-1)
> plot(ts(y))
# y 不是时间独立的序列, p值很小, 拒绝零假设
> Box.test(y)

        Box-Pierce test

data:  y
X-squared = 91.9547, df = 1, p-value < 2.2e-16

> Box.test(y,type="Ljung-Box")

        Box-Ljung test

data:  y
X-squared = 94.7134, df = 1, p-value < 2.2e-16

# p值较大, 接受零假设, 此序列为时间独立的
> Box.test(rnorm(100))

        Box-Pierce test

data:  rnorm(100)
X-squared = 1.7053, df = 1, p-value = 0.1916

> Box.test(rnorm(100),type="Ljung-Box")

        Box-Ljung test

data:  rnorm(100)
X-squared = 0.4211, df = 1, p-value = 0.5164
```

77.6.3 其它检验

其它还有 McLeod-Li, Turning-point, difference-sign, rank 检验等.

还可以使用 Durbin–Watson 检验.

```
> library(car)
> ?durbin.watson
> durbin.watson(y)
[1] 0.07259672
```

77.6.4 游程检验(runs.test)

零假设为: 游程是随机的. 备择假设: 游程是增加的(或减少的). 检验基于游程的频率.

```
> library(tseries)
> ?runs.test
> x <- factor(sign(rnorm(100)))  # randomness
# p值较大, 是随机的游程
>     runs.test(x)

       Runs Test

data:  x
Standard Normal = 0.6416, p-value = 0.5212
alternative hypothesis: two.sided

# p值很小, 不是随机游程.
>  x <- factor(rep(c(-1,1),50))  # over-mixing
>     runs.test(x)

       Runs Test

data:  x
Standard Normal = 9.8499, p-value < 2.2e-16
```

```
alternative hypothesis: two.sided
```

77.6.5 tsdiag()

用于绘制标准化残差, 自相关的残差, portmanteau test(Box–Pierce(Ljung–Box) test) 的p值. 输入是 arima() 函数拟合的结果(拟合 ARIMA 模型).

```
data(co2)
r <- arima(
  co2,
  order = c(0, 1, 1),
  seasonal = list(order = c(0, 1, 1), period = 12)
)
tsdiag(r)
```

```
> r

Call:
arima(x = co2, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period = 12))

Coefficients:
          ma1     sma1
      -0.3501  -0.8506
s.e.   0.0496   0.0257

sigma^2 estimated as 0.0826:  log likelihood = -86.08,  aic = 178.16
```

Chapter 78

线性模型

78.1 时间序列分析的主要问题

统计的时候我们喜欢独立数据, 而时间序列包含非独立数据. 时间序列分析的目的是提取结构并将序列转换为独立的数据(经常叫做 “innovations”), 通常是提供一个模型(model/recipe)来构建接近原始时间序列, 即去除噪声的部分.

我们可以从另一个方面来看: 当研究统计现象时, 一般它有不同的实现(realizations). 对于时间序列, 只有一个实现. 于是, 我们把研究一个时间点的不同实现改变为研究不同时间点的同一个实现. 对于不同的统计现象, 这两种观点可以一样, 也可能不一样.

78.2 介绍

基本的包就是 tseries, sound, signal

另外 seewave 包是时间, 波形数据分析和可视化的包. 里面有函数计算熵. 2D, 3D谱图等很多函数.

dse: Dynamic Systems Estimation. 可以分析多元时间序列数据.

有一个 dse-guide.pdf. 这个是下载地址为 http://www.bank-banque-canada.ca/pgilbert/dse-guide.pdf.

参考: CRAN Task View: Time Series Analysis 有很多的介绍.

78.3 arima.sim()函数-模拟产生各种时间序列

78.3.1 ts()的用法

ts() 用于产生时间序列. 用法如下.(例子来自在线帮助)

```
> ts(data=1:10, frequency = 4, start = c(1959, 2))
     Qtr1 Qtr2 Qtr3 Qtr4
1959        1    2    3
1960    4    5    6    7
1961    8    9   10
> ts(1:10, frequency = 4, start = c(1959, 2),end=c(1970,3))
     Qtr1 Qtr2 Qtr3 Qtr4
1959        1    2    3
1960    4    5    6    7
1961    8    9   10    1
1962    2    3    4    5
1963    6    7    8    9
1964   10    1    2    3
1965    4    5    6    7
1966    8    9   10    1
1967    2    3    4    5
1968    6    7    8    9
1969   10    1    2    3
1970    4    5    6

# frequency >=5 就需要使用 print() 显示其格式
> ts(1:10, frequency = 4, start = c(1959, 2))
     Qtr1 Qtr2 Qtr3 Qtr4
1959        1    2    3
```

```
1960   4   5   6   7
1961   8   9  10
> ts(1:10, frequency = 5, start = c(12, 2))
Time Series:
Start = c(12, 2)
End = c(14, 1)
Frequency = 5
 [1]  1  2  3  4  5  6  7  8  9 10
> ts(1:10, frequency = 7, start = c(12, 2))
Time Series:
Start = c(12, 2)
End = c(13, 4)
Frequency = 7
 [1]  1  2  3  4  5  6  7  8  9 10

# 打印时间序列
> print( ts(1:10, frequency = 7, start = c(12, 2)), calendar = TRUE)
   p1 p2 p3 p4 p5 p6 p7
12    1  2  3  4  5  6
13  7  8  9 10

# 绘图
> gnp <- ts(cumsum(1 + round(rnorm(100), 2)),
            start = c(1954, 7), frequency = 12)
> plot(gnp)
```

78.3.2 产生时间序列

模拟: 时间序列模拟的目的之一是发现序列的结构, 即序列怎么样构成的, 即发现一个算法来近似生成已知序列.

下面是几个模拟的时间序列.

```
n <- 100
k <- 5
N <- k*n
x <- (1:N)/n # x 为时间, 1-5 500个分隔
```

```
# 高斯噪声(白噪声)
y1<-rnorm(N)
plot(ts(y1))

# 累积噪声(随机漫步)
 y2<-cumsum(y1)
 plot(y2) # 普通绘图为散点图
 plot(ts(y2))

# 累积噪声(随机漫步)+高斯噪声
 y3<-cumsum(y1)+rnorm(N)
 plot(ts(y3))

# 累积累积噪声(累积随机漫步)
y4 <- cumsum(cumsum(y1))

# 乱七八糟的累积
y5 <- cumsum(cumsum(y1)+rnorm(N))+rnorm(N)

# 趋势+漫步+噪声
y6 <- 1 - x + cumsum(y1) + .2 * rnorm(N)

# x 的二次函数构成趋势, 然后+漫步+噪声
y7 <- 1 - x - .2*x^2 + cumsum(y1) +
     .2 * rnorm(N)

# 季节趋势+噪声
z <- .3 + .5*cos(2*pi*x) - 1.2*sin(2*pi*x) +
     .6*cos(2*2*pi*x) + .2*sin(2*2*pi*x) +
     -.5*cos(3*2*pi*x) + .8*sin(3*2*pi*x)
y8 <- z + .2 * rnorm(N)
y9 <- z+  cumsum(rnorm(N)) + .2*rnorm(N)

# 画图
op <- par(mfrow = c(3,3))
plot(ts(y1))
plot(ts(y2))
plot(ts(y3))
plot(ts(y4))
plot(ts(y5))
```

```
plot(ts(y6))
plot(ts(y7))
plot(ts(y8))
lines(z,type='l',lty=3,lwd=3,col='red')
plot(ts(y9))
par(op)
```

78.3.3 arima.sim()函数产生AR,MA或ARMA过程

我们可以使用 arima.sim() 函数产生一个模拟的 AR, MA 或 ARMA 过程. 下面是 R 的例子.

```
# 产生ARMA 过程, 指定MA的方差为 0.1796
arima.sim(n = 63, list(ar = c(0.8897, -0.4858), ma = c(-0.2279, 0.2488)),
           sd = sqrt(0.1796))
# mildly long-tailed. 可以指定随机数产生函数. 默认为正态
分布 rnorm
    arima.sim(n = 63, list(ar=c(0.8897, -0.4858), ma=c(-0.2279, 0.2488)),
            rand.gen = function(n, ...) sqrt(0.1796) * rt(n, df = 5))

# 产生 ARIMA 序列. 其 d=1. 即1 阶差分是平稳的
    ts.sim <- arima.sim(list(order = c(1,1,0), ar = 0.7), n = 200)
    ts.plot(ts.sim)
```

78.4 经典模型

一般, 我们想找到(分解)一个时间序列的3个部分: 整体趋势, 周期部分, 噪声.

下面看看时间序列的3个典型部分

```
# 自己构建有3个部分的数据
> x=seq(0,10,by=0.1)
```

```
> y=x+sin(2*pi*x)+3*cos(2*pi*x)+rnorm(length(x))
> plot(ts(y))

# 或 R 的数据
> data(co2)
> plot(co2)
```

此时使用各种回归都不好使.

78.4.1 一般回归

```
data(co2)
plot(co2)
x <- as.vector(time(co2))
y <- as.vector(co2)

# 对时间序列做线性回归并作预测曲线, 可以看到其拟合不太好
r <- lm( y ~ poly(x,1) + cos(2*pi*x) + sin(2*pi*x) )
plot(y~x, type='l', xlab="time", ylab="co2")
lines(predict(r)~x, lty=3, col='red', lwd=3)
# 拟合的残差图可能更加明显
plot( y-predict(r),
     main = "The residuals are not random yet",
     xlab = "Time",
     ylab = "Residuals" )

# 多项式拟合
r1 <- lm( y ~ poly(x,2) + cos(2*pi*x) + sin(2*pi*x) )
plot(y~x, type='l', xlab="time", ylab="co2")
lines(predict(r1)~x, lty=3, col='red', lwd=3)
#残差
plot( y-predict(r1),
     main = "Better residuals -- but still not random",
     xlab = "Time",
     ylab = "Residuals" )
```

```
# 进一步增加高频成分
r2 <- lm( y ~ poly(x,2) + cos(2*pi*x) + sin(2*pi*x)
                  + cos(4*pi*x) + sin(4*pi*x) )
plot(y~x, type='l', xlab="time", ylab="co2")
lines(predict(r2)~x, lty=3, col='red', lwd=3)
# 残差
plot( y-predict(r2),
     main = "Are those residuals any better?",
     xlab = "Time",
     ylab = "Residuals" )

# 对刚才的两个拟合的残差做自相关, 可以看到衰减比较慢, 其残差非独立
op <- par(mfrow=c(2,1))
acf(y - predict(r1))
acf(y - predict(r2))
par(op)
```

78.4.2 fft()寻找趋势

寻找趋势就是滤波除去高频成分.

```
> x<-co2
> plot(x) # 周期曲线
> a=fft(x)
> a[20:(l-19)]<-0 # 去除高频成分
> y<-fft(a,inv=T) # 傅立叶逆变换
> plot(Re(y)) # 几乎成直线

# 去除越来越多的高频成分
n <- 1000
x <- cumsum(rnorm(n))+rnorm(n)
plot(x, type='l', ylab="",
    main="FFT: Removing more and more high frequencies")
for (i in 1:10) {
  y <- fft(x)
```

```
  y[(1+i):(length(y)-i)] <- 0
  y <- Re(fft(y, inverse=T)/length(y))
  lines(y, col=rainbow(10)[i])
}
```

78.5 分解时间序列

78.5.1 decompose()

函数 decompose() 用法为:

```
decompose(x, type = c("additive", "multiplicative"), filter = NULL)
```

设 T 为趋势(trend), S 为周期(seasonal), e 为噪声.

- type = "additive": 使用模型

$$Y[t] = T[t] + S[t] + e[t]$$

- type = "multiplicative": 使用模型

$$Y[t] = T[t] * S[t] + e[t]$$

- filter: 滤波系数

```
# decompose 的用法
r <- decompose(co2)
plot(r) # 绘制原始数据, trend, seasonal, 噪声 4个图
```

78.5.2 stl()

函数 stl() 是更加复杂的分解函数. 使用 Loess 方法(但不是 stats 包的 loess() 函数)分解周期性时间序列. 用法见帮助

下面是例子

```
# stl 的用法
s <- stl(co2, s.window="periodic")
r <- stl(co2, s.window="periodic")$time.series

> names(s)
[1] "time.series" "weights"    "call"       "win"        "deg"
[6] "jump"       "inner"      "outer"

# r 包括周期,整体趋势和噪声三部分
> r
          seasonal    trend    remainder
Jan 1959 -0.06100103 315.1954  0.2856440966
Feb 1959  0.59463870 315.3023  0.4130545587
Mar 1959  1.32899651 315.4093 -0.2382530567
Apr 1959  2.46904706 315.5147 -0.4237112836
May 1959  2.95704630 315.6201 -0.4471182024

op <- par(mfrow=c(4,1), mar=c(3,4,0,1), oma=c(0,0,2,0))
plot(co2)
lines(r[,2], col='blue') # 趋势线
lines(r[,2]+r[,1], col='red') # 趋势+周期
plot(r[,1],t='l',col='blue') # 周期部分
plot(r[,3]) # 噪声
acf(r[,3], main="residuals") # 噪声自相关
par(op)
mtext("STL(co2)", line=3, font=2, cex=1.2)

# 实际上 stl 有 plot
plot(s) # 绘制 原始数据, trend, seasonal, 噪声 4个图
```

78.5.3 HoltWinters 分解

相关函数有 predict.HoltWinters, plot.HoltWinters. 用法见帮助.

```
> (m <- HoltWinters(co2))
Holt-Winters exponential smoothing with trend and additive seasonal component.

Call:
 HoltWinters(x = co2)

Smoothing parameters:
 alpha:  0.4907075
 beta :  0.01197529
 gamma:  0.4536582

Coefficients:
          [,1]
a   364.6866567
b     0.1268701
s1    0.2812220
s2    1.0173743
s3    1.6642371
s4    2.9411121
s5    3.3487805
s6    2.5064789
s7    0.9613233
s8   -1.3122489
s9   -3.3464772
s10  -3.1988220
s11  -1.8558114
s12  -0.5254438
> plot(m)
> lines(co2,col='red')
```

下面是 [46] 15.2.14 的例子

```
data(LakeHuron)
x <- LakeHuron
```

```
before <- window(x, end=1935) # 1935年之前的数据
after <- window(x, start=1935) # 1935之后的数据
# 优化的初始值
a <- .2
b <- 0
g <- 0
model <- HoltWinters(
  before,
  alpha=a, beta=b, gamma=g)
# 对1935年后的37年预测
forecast <- predict(
  model,
  n.ahead=37,
  prediction.interval=T)

# 绘图.
plot(model, predicted.values=forecast,
    main="Holt-Winters filtering: constant model")
lines(after)
```

78.6 MA(Moving Average models)-滑 动 平均模型

参考 [46] 15.3.3

filter() 函数使用 fft 计算卷积.

```
filter(x, filter, method = c("convolution", "recursive"),
      sides = 2, circular = FALSE, init)
```

参数

- filter: 滤波系数向量, 顺序是与时间序列逆的

- method: 如果是 convolution, 使用滑动平均(默认值). 公式为[1]

  ```
  'y[i] = f[1]*x[i+o] + ... + f[p]*x[i+o-(p-1)]'
  ```

  如果是 recursive, 使用自回归. 公式为

  ```
  'y[i] = x[i] + f[1]*y[i-1] + ... + f[p]*y[i-p]'
  ```

- side: 只对 convolution 有效. =1, filter系数只对过去的值有效. =2, filter系数在延迟 0 上对称, 此时系数的个数需为奇数. 如果是偶数, 那么前面会多一个.

比较: 自回归相当于加入一个非截断窗, 可以有效的消除滑动平均的截断效应. 即其窗口为直接截断. 如果有异常值, 滑动平均往往有大的变化.

78.6.1 产生滑动平均序列

下面通过白噪声构造 MA 序列. 注意滤波系数的顺序是x的逆顺序, 自相关函数在阶数多的时候衰减慢.

```
# 简单的例子, 纯手工计算
> x=1:10
> y=filter(x, filter=c(0.5,0.3));y
 [1] 1.3 2.1 2.9 3.7 4.5 5.3 6.1 6.9 7.7  NA

# 依次取 1:k, 2:k+1 ... m:k+m-1
# ma[m]=x[m:(k+m-1)]*rev(filter)
1.3=2*0.5+1*0.3
2.1=3*0.5+2*0.3


# 白噪声
n <- 200
```

[1]详细参考帮助

```
x <- rnorm(n)

# 一阶滑动平均
y <- ( x[2:n] + x[2:n-1] ) / 2 # filter=c(1/2,1/2)

op <- par(mfrow=c(3,1), mar=c(2,4,2,2)+.1)
plot(ts(x), xlab="", ylab="white noise")
plot(ts(y), xlab="", ylab="MA(1)")
acf(y, main="") # 查看其自相关系数
par(op)

# 三阶滑动平均, 滤波系数 filter=c(1/4,1/4,1/4,1/4)
y <- ( x[1:(n-3)] + x[2:(n-2)] + x[3:(n-1)] + x[4:n] )/4

op <- par(mfrow=c(3,1), mar=c(2,4,2,2)+.1)
plot(ts(x), xlab="", ylab="white noise")
plot(ts(y), xlab="", ylab="MA(3)")
acf(y, main="")
par(op)

# 二阶滑动平均, 滤波系数 filter=c(3,-2,1)
y <- 3*x[3:n] - 2 * x[2:(n-1)] + x[1:(n-2)]

op <- par(mfrow=c(3,1), mar=c(2,4,2,2)+.1)
plot(ts(x), xlab="", ylab="white noise")
plot(ts(y), xlab="", ylab="Momentum(2)")
acf(y, main="")
par(op)

# 使用R函数 filter 计算滑动平均
y <- filter(x, c(3,-2,1))

op <- par(mfrow=c(3,1), mar=c(2,4,2,2)+.1)
plot(ts(x), xlab="", ylab="White noise")
plot(ts(y), xlab="", ylab="Momentum(2)")
acf(y, na.action=na.pass, main="")
par(op)
```

78.6.2 使用滑动平均查看序列的趋势

MA 相当于低通滤波器. 我们可以使用 fft 去掉高频成分来达到同样的效果.

下面是滑动平均的 [46] 的很好的一个例子

```
# 查看前向平均与后向平均的不同
x <- co2
plot(window(x, 1990, max(time(x))), ylab="co2")
k <- 12
lines( filter(x, rep(1/k,k)),
      col='red', lwd=3)
lines( filter(x, rep(1/k,k), sides=1),
      col='blue', lwd=3)
legend(par('usr')[1], par('usr')[4], xjust=0,
      c('smoother', 'filter'),
      lwd=3, lty=1,
      col=c('red','blue'))
```

78.7 AR(Auto-Regressive models)自回归模型

参考 [46] 15.3.4

1927, 英国统计学家 G. U. Yule 提出 AR 模型. 不久后, 英国数学家, 天文学家 G. T. Walker 爵士提出 MA 模型.

78.7.1 AR(1)

一阶 AR 模型为

$$X(n+1) = aX(n) + noise$$

当系数 $a=1$, 实际上是随机漫步. 随机漫步的自相关衰减很慢. 下面是一个 AR(1) 的例子, 因为可以从 $y[n-1]$ 预测到 $y[n]$

```
# 简单的例子, 纯手工计算
> x=1:10; x
 [1]  1  2  3  4  5  6  7  8  9 10
> y=filter(x,c(1,2),method='r');y
 [1]    1    3    8   18   39   81  166  336  677 1359

> f=c(1,2)
> y1=x[1]; y1
[1] 1
> y2=x[2]+f[1]*y1; y2
[1] 3
> y3=x[3]+f[1]*y2+f[2]*y1; y3
[1] 8
> y4=x[4]+f[1]*y3+f[2]*y2; y4
[1] 18

# 随机漫步就是 AR(1), a=1
n <- 200
x<-rnorm(n)
y <- rep(0,n)

# 按照定义随机漫步
y[1]=x[1]
for (i in 2:n) {
  y[i] <- x[i]+y[i-1]
}

# 实际上可以使用 cumsum() 函数直接得到
# y1==y
y1 <- cumsum(x)
op <- par(mfrow=c(3,1), mar=c(2,4,2,2)+.1)
plot(ts(x), xlab="", ylab="")
plot(ts(y1), xlab="", ylab="AR(1)")
acf(y, main="") # 随机漫步的自相关衰减很慢
par(op)

# 可以使用 filter() 函数, 参数 method='recursive'
```

```
# y2==y1==y
y2<-filter(x,filter=1,method='r')
```

78.7.2 AR(p)

p=3

```
n <- 200
x<-rnorm(n)
f=c(.3,-.7,.5)
y <- rep(0,n)
y[1:3]=x[1:3]
for (i in 4:n) {
  y[i] <- f[1]*y[i-1] +f[2]*y[i-2] + f[3]*y[i-3] + x[i]
}
op <- par(mfrow=c(3,1), mar=c(2,4,2,2)+.1)
plot(ts(y), xlab="", ylab="AR(3)")
acf(y, main="", xlab="")
pacf(y, main="", xlab="")
par(op)

# 我们可以使用 arima.sim() 函数产生一个模拟的 AR(p) 过程
n <- 200
x <- arima.sim(list(ar=c(.3,-.7,.5)), n)

op <- par(mfrow=c(3,1), mar=c(2,4,2,2)+.1)
plot(ts(x), xlab="", ylab="AR(3)")
acf(x, xlab="", main="")
pacf(x, xlab="", main="")
par(op)
```

78.8 平稳性与各态遍历性

78.8.1 平稳性

弱平稳: 如果时间序列 $X(t)$ 不依赖于时间 t, 并且$X(t), X(s)$的协方差只依赖于$abs(t-s)$, 说这个时间序列是弱平稳的.

平稳: 若$X(t)$同分布, 且$X(t), X(s)$的联合分布对给定$abs(t-s)$也是同分布, 此时间序列叫做平稳. 意味着弱平稳的序列其二阶平稳

例如, 如果时间序列有趋势, 即 $E(X(t))$ 不是常数, 则此时间序列不平稳.

```
n <- 200
x <- seq(0,2,length=n)
trend <- ts(sin(x))
plot(trend,
     ylim=c(-.5,1.5),
    lty=2, lwd=3, col='red',
    ylab='')

# r 是平稳的
r <- arima.sim(
  list(ar = c(0.5,-.3), ma = c(.7,.1)),
  n,
  sd=.1
)

# trend+r 是不平稳的
lines(trend+r)
```

随机漫步也不是平稳的, 期望保持 0, 但是其方差增大.

```
n <- 200
k <- 10
```

```
x <- 1:n
# 产生10个随机漫步序列
r <- matrix(nr=n,nc=k)
for (i in 1:k) {
  r[,i] <- cumsum(rnorm(n))
}
matplot(x, r,
        type = 'l',
        lty = 1,
        col = par('fg'),
        main = "A random walk is not stationnary")
abline(h=0,lty=3)
```

78.8.2 各态遍历(Ergodicity)

给定随机过程 $X(n)$, 如何估计 $X(1)$? 两种方法:

- 将此过程实现 k 次, 计算每次的 $X(1)$ 的平均
- 实现一次, 使用 $mean(X(1), X(2), \cdots$

我们需要的是第一个方法. 但是如果此时间序列是各态遍历(Ergodicity)的, 第二种也可以(好像还有其它条件)[2]

78.8.3 TODO: AR的平稳性

对于 AR(1)

$$Y(t+1) = a * Y(t) + e(t)$$

若 $abs(a) < 1$, 则可以对于 AR(1)

$$Y(t+1) - a * Y(t) = e(t)$$

[2]请参考随机过程教科书

或一般写作

$$\phi(B)Y = e$$

$\phi(B)$叫做一个算子, 而

$$\phi(u) = 1 - a_1 u - a_2 u^2 - \cdots - a_p u^p$$

当 ϕ 的所有根的模大于 1, 此时间序列是平稳的.

78.8.4 TODO: MA与可逆性(invertibility)

$$\begin{aligned} Y &= \psi(B)e \\ where \quad \psi(u) &= 1 + b_1 u + b_2 u^2 + \cdots + bqu^q \end{aligned}$$

依然要求 ψ 的所有根的模大于 1, 此时间序列是可逆的(invertible). 若无此假设, 其自相关函数不能唯一确定其平均的系数. 例如一个 MA(1) 过程

$$Y(t+1) = Z(t+1) + a * Z(t)$$

可以使用 $1/a$ 代替 a, 而其自相关系数不变.

下面是一个例子

```
> x=rnorm(100)
> f1=filter(x,f=2)
> f2=filter(x,f=0.5)
# a1==a2
> a1=acf(f1,plot=F); a1

Autocorrelations of series ‘f1’, by lag

    0     1     2     3     4     5     6     7     8     9    10
1.000 -0.080  0.024  0.035  0.049 -0.008 -0.087  0.211 -0.085  0.050 -0.045
   11    12    13    14    15    16    17    18    19    20
-0.085  0.023  0.009 -0.012 -0.008 -0.021 -0.091  0.037  0.085 -0.006
> a2=acf(f2,plot=F); a2
```

```
Autocorrelations of series 'f2', by lag

    0     1     2     3     4     5     6     7     8     9    10
1.000 -0.080  0.024  0.035  0.049 -0.008 -0.087  0.211 -0.085  0.050 -0.045
   11    12    13    14    15    16    17    18    19    20
-0.085  0.023  0.009 -0.012 -0.008 -0.021 -0.091  0.037  0.085 -0.006
```

下面是 [46] 的例子, 供参考

```
n <- 200
ma <- 2
mai <- 1/ma
op <- par(mfrow=c(4,1), mar=c(2,4,1,2)+.1)
# 系数为 2
x <- arima.sim(list(ma=ma),n)
plot(x, xlab="", ylab="")
acf(x, xlab="", main="")
lines(0:n,
     ARMAacf(ma=ma, lag.max=n),
     lty=2, lwd=3, col='red')
# 系数为1/2
x <- arima.sim(list(ma=mai),n)
plot(x, xlab="", ylab="")
acf(x, main="", xlab="")
lines(0:n,
     ARMAacf(ma=mai, lag.max=n),
     lty=2, lwd=3, col='red')
par(op)
```

78.9 ARMA

自回归-滑动平均(Auto-Regression–Moving Average, ARMA)模型, 具有AR阶数p和MA阶数q的ARMA过程常记作ARMA(p,q).

ARMA(p,q)可以用线性差分方程进行描述

$X[t] = a[1]X[t-1] + ... + a[p]X[t-p] + e[t] + b[1]e[t-1] + ... + b[q]e[t-q]$

其中 $a[1]X[t-1] + ... + a[p]X[t-p]$为自回归, $e[t] + b[1]e[t-1] + ... + b[q]e[t-q]$是滑动平均.

显然, ARMA模型描述的是一个平稳(时不变)的线性系统.

78.10 差分-得到平稳过程

使用 ARMA 模型拟合一个时间序列, 此时间序列必须平稳.

要想得到平稳序列(简言之, 去除趋势), 可以尝试对其差分.

通常不平稳可以通过图看出, 也可以查看其自相关函数(ACF). 若平稳, ACF通常很快衰减到零(一般指数衰减)

```
data(BJsales)
op <- par(mfrow=c(3,1), mar=c(2,4,3,2)+.1)
plot(BJsales, xlab="",
   main="The trend disappears if we differentiate")

acf(BJsales, xlab="", main="")
# 差分的acf
acf(diff(BJsales), xlab="", main="",
   ylab="ACF(diff(BJsales)")
par(op)
```

78.11 ARIMA过程

参考[46] 15.3.15 和 http://wiki.mbalib.com/wiki/ARIMA模型

78.11.1 起源

1970 年, 美国统计学家 G.E.P. Box 和英国统计学家 G.M. Jenkins 出版了《Time Series Analysis Forecasting and Control》一书. 系统阐述了 ARIMA 模型. 为纪念他们的贡献, 常常把 ARIMA 模型称为 Box-Jenkins 模型.

自回归移动平均模型(Autoregressive Integrated Moving Average Model,简记ARIMA)

78.11.2 什么是ARIMA模型

ARIMA模型全称为自回归移动平均模型(Autoregressive Integrated Moving Average Model,简记ARIMA), 是由博克思(Box)和詹金斯(Jenkins)于70年代初提出的一著名时间序列预测方法, 所以又称为box-jenkins模型、博克思-詹金斯法。其中ARIMA(p, d, q)称为差分自回归移动平均模型, AR是自回归, p为自回归项; MA为移动平均, q为移动平均项数, d为时间序列成为平稳时所做的差分次数。

78.11.3 ARIMA模型的基本思想

ARIMA模型的基本思想是: 将预测对象随时间推移而形成的数据序列视为一个随机序列, 用一定的数学模型来近似描述这个序列。这个模型一旦被识别后就可以从时间序列的过去值及现在值来预测未来值。现代统计方法、计量经济模型在某种程度上已经能够帮助企业对未来进行预测。

一般写作

$$\phi(B)(1-B)^d X(t) = \theta(B)e(t)$$

ARIMA 是不平稳的过程. ARIMA 就是 ARMA(p,q) 过程的积分, 设积分次数为 d, 则 ARIMA 记为 ARIMA(p,d,q). 即其 d 次差分是平稳的. 例如对于 ARMA(0,0) 的 1 阶 ARIMA 过程, 即随机漫步, 其方差是随时间 t 增大的. 这是对其差分的主要原因.

可以使用差分直到其 ACF 迅速衰减来推断 ARIMA 的阶数 d. 你可能想对一个数据连续差分, 但是如果其 ACF 迅速衰减, 就应该停止. 过多差分是不好的.

下面几个例子, 是否参数设置合适才符合判断标准?

```
n <- 200

# MA 序列
m <- arima.sim(list(ma=c(0.5,0.5)),n )
op <- par(mfrow=c(3,1), mar=c(2,4,3,2)+.1)
plot(m)
acf(m)
pacf(m)
par(op)

# AR 序列
a <- arima.sim(list(ar=0.7),n )
op <- par(mfrow=c(3,1), mar=c(2,4,3,2)+.1)
plot(a)
acf(a)
pacf(a)
par(op)

# ARMA 序列
arma<-arima.sim(list(ar=0.7),n )
```

78.11.4 一些例子与arima()拟合

我们先看一些一般的例子. 最后使用 arima() 函数拟合.

例如可以看到对数据 co2 差分直到 4 次, 其 ACF 迅速衰减. 但是, 其周期为 12, 我们延迟12差分 1, 2 次后其 ACF 分别直线衰减, 指数衰减

```
# co2 差分 4 次
op <- par(mfrow=c(5,2), mar=c(2,4,3,2)+.1)
```

```
plot(co2, xlab="",
   main="The trend disappears if we differentiate")
acf(co2, xlab="", main="")

plot(diff(co2), xlab="",
   main="The trend diff 1")
acf(diff(co2), xlab="", main="",
   ylab="ACF(diff=1)")

plot(diff(co2,diff=2), xlab="",
   main="The trend diff 2")
acf(diff(co2,diff=2), xlab="", main="",
   ylab="ACF(diff=2)")

plot(diff(co2,diff=3), xlab="",
   main="The trend diff 3")
acf(diff(co2,diff=3), xlab="", main="",
   ylab="ACF(diff=3)")

# ACF 迅速衰减
plot(diff(co2,diff=4), xlab="",
   main="The trend diff 4")
acf(diff(co2,diff=4), xlab="", main="",
   ylab="ACF(diff=4)")
par(op)

=================
# 延迟12差分 1, 2 次后其 ACF 分别直线衰减, 指数衰减
op <- par(mfrow=c(3,2), mar=c(2,4,3,2)+.1)
plot(co2, xlab="",
   main="The trend disappears if we differentiate")
acf(co2, xlab="", main="")

# 延迟 12 差分后 ACF 直线衰减
plot(diff(co2,lag=12), xlab="",
   main="The trend diff 1")
acf(diff(co2,lag=12), xlab="", main="",
   ylab="ACF(diff=1)")

# 延迟 12 差分后 ACF 指数衰减
plot(diff(co2,diff=2,lag=12), xlab="",
```

```
   main="The trend diff 2")
acf(diff(co2,diff=2,lag=12), xlab="", main="",
   ylab="ACF(diff=2)")
par(op)
```

对于数据 sunspot.month 其1 阶差分就基本平稳了

```
op <- par(mfrow=c(4,1), mar=c(2,4,3,2)+.1)
plot(sunspot.month, xlab="", ylab="sunspot")
acf(sunspot.month, xlab="", main="")
plot(diff(sunspot.month),
    xlab="", ylab="diff(sunspot)")
acf(diff(sunspot.month), xlab="", main="")
par(op)
```

数据 JohnsonJohnson 其1 阶差分就基本平稳了

```
data(JohnsonJohnson)
x <- log(JohnsonJohnson)
op <- par(mfrow=c(4,1), mar=c(2,4,3,2)+.1)
plot(x, xlab="", ylab="JJ")
acf(x, main="")
plot(diff(x), ylab="diff(JJ)")
acf(diff(x), main="")
par(op)
```

下面看到, 去除了一个数据的趋势后, 其 1 阶差分的 ACF 指数衰减, 更高次数差分的 ACF 衰减更厉害

```
data(austres)
x <- lm(austres ~ time(austres))$res
op <- par(mfrow=c(6,1), mar=c(2,4,0,2)+.1)
plot(x)
acf(x)
```

```
plot(diff(x))
acf(diff(x))
plot(diff(x, difference=2))
acf(diff(x, difference=2))
par(op)
```

模拟一个 2 阶的 ARIMA 过程, 看到其 2 阶差分是指数衰减的.

```
n <- 200
x <- arima.sim(
  list(
    order=c(2,2,2),
    ar=c(.5,-.8),
    ma=c(.9,.6)
    ),
  n
)
op <- par(mfrow=c(3,1), mar=c(2,4,4,2)+.1)
acf(x, main="You will have to differentiate twice")
acf(diff(x), main="First derivative")
acf(diff(x, differences=2), main="Second derivative")
par(op)

# 使用 arima 函数估计其参数
> arima(x,c(2,2,2))

Call:
arima(x = x, order = c(2, 2, 2))

Coefficients:
        ar1     ar2    ma1    ma2
     0.5390  -0.8366  0.7967  0.5815
s.e.  0.0425   0.0398  0.0616  0.0622

sigma^2 estimated as 1.013:  log likelihood = -287.32,  aic = 584.64
```

78.12 如何选择模型: Box-Jenkins 方法

参考 http://en.wikipedia.org/wiki/Box-Jenkins

由统计学家 George Box 和 Gwilym Jenkins 命名. 目的是从过去的时间序列的拟合来预测未来的走势.

78.12.1 模型的步骤

1. 模型识别与选择: 确定序列平稳, 识别周期性(如果必要, 对周期差分), 绘图查看自相关(ACF)与偏自相关(PACF)来决定模型中的 MA 和 AR 成分.
2. 使用例如最大似然法或非线性最小方差法来估计模型参数. [3]
3. 检验模型: 残差应该互相独立, 残差的均值与方差应该平稳(使用 Ljung-Box test(函数 Box.test()), 或绘残差的 ACF 及 PACF 图), 若不符合要求, 回到第一步.

78.12.2 检验平稳性

第一步是识别平稳性. 可以绘出时间序列的图(run sequence plot, 也叫 run chart). 也可以由 ACF 图来查看, 如果衰减很慢, 则不平稳.

78.12.3 检验周期性

可以使用 ACF plot, a seasonal subseries plot(例如先绘制所有第一个月的数据, 然后是第二个月的数据, 然后...), or a spectral plot(谱分析).

下面是 seasonal subseries plot 的例子

[3] Brockwell and Davis, (1987,2002) for the mathematical details.[51]

```
fit <- stl(log(co2), s.window = 20, t.window = 20)
plot(fit)
op <- par(mfrow = c(2,2))
monthplot(co2, ylab = "data", cex.axis = 0.8)
monthplot(fit, choice = "seasonal", cex.axis = 0.8)
monthplot(fit, choice = "trend", cex.axis = 0.8)
monthplot(fit, choice = "remainder", type = "h", cex.axis = 0.8)
par(op)
```

78.12.4 差分得到平稳序列

Box and Jenkins 建议使用差分得到平稳序列.

但是曲线拟合, 然后数据减去拟合值也可以得到平稳序列.

78.12.5 周期差分

识别的目的是检验周期性, 若存在, 识别其MA和AR的阶数(order). 对很多时间序列来说, 周期是知道的, 并且单一的周期足够了. 例如对于月份数据, 周期往往是 AR(12) 或 MA(12). 拟合的时候一般不去除周期, 而使用 ARIMA 来代表它. 但是对其按周期差分可能对拟合会有帮助.

78.12.6 确定参数 p 和 q

一旦平稳性和周期性确定后, 下一步就是确定 AR 的参数 p 和 MA 的参数 q.

基本的方法是绘图 ACF 和 PACF.

78.12.7 AR参数p

对于 AR(1) 过程, 其 ACF 指数衰减.

但是高阶的 AR 过程其 ACF 出现指数衰减和正弦成分的混合. 需要联合使用 ACF 与 PACF. AR(p) 的 PACF 在 $p+1$ 处衰减为 0. 故我们检查 PACF 在何处基本为 0(不明显异于 0). 一般绘出 95% 的置信区间(一般的软件都会给出, 若没有, 则大概是 $\pm 2/\sqrt{N}$, N 为时间序列样本量).

78.12.8 MA参数q

MA(q) 的 ACF 在 $q+1$ 处及其之后衰减为 0. 一般也是绘出 95% 的置信区间(一般的软件都会给出, 若没有, 则大概是 $\pm 2/\sqrt{N}$, N 为时间序列样本量).

PACF 一般对 MA 没有什么帮助.

78.12.9 总结

下面的表是如何使用ACF来选择模型

ACF shape	Indicated Model
指数衰减到 0	AR模型, 使用 PACF 确定其阶数 p
正负交替衰减到 0	AR模型, 使用 PACF 确定其阶数 p
一个或多个尖突起, 其它为 0	MA模型, 阶数由衰减到 0 的点确定
几个延迟后衰减	ARMA 混合模型
全部为 0	数据是随机的
一定区间内值很大	包含周期的 AR 模型
不衰减到 0	序列不平稳

78.12.10 混合模型难以识别

实际中, ACF 和 PACF 多有随机, 使得模型识别困难. 而混合模型识别尤其困难.

虽然经验是有帮助的, 但是使用这些方法发现一个好的拟合要多多试验. 近年来发展了基于信息的 FPE (Final prediction error) and AIC (Akaike Information Criterion) 判别方法, 便于自动选择模型. [4]

78.12.11 Box-Jenkins model diagnostics

Box-Jenkins model 的诊断类似于非线性最小方差拟合的诊断. 即残差应该为白噪声(绘图查看, 或 Box-Ljung statistic).

78.12.12 TODO:例子

78.13 异方差的情况

参考 [24] 5.4

使用 ARIMA 拟合非平稳($d \neq 0$)时间序列有一个重要的假定: 残差为零均值白噪声. 即

- $E(e) = 0$
- 残差为随机序列, 即 $Cov(e_i, e_j) = 0$
- 方差齐性

均值为 0 很容易满足, 直接中心化即可. 但是残差齐性如果不满足, 则需对时间序列进行变换, 如果我们知道方差与均值(时间)之间的函数关系的话.

[4] 进一步请参考 Brockwell and Davis (1987, 2002).[51]

方差齐性变换见 "数据变换–稳定方差的变换" 部分.

此方法为异方差时间序列的拟合提供了精巧的方法, 但是, 实际中往往不知道异方差的函数形式. 通常只是通过残差图凭经验得到残差方差的函数. 一般的金融序列标准差与均值具有正相关关系, 故异方差函数通常假定为

$$h(\mu_t) = \mu_t^2$$

此种变换被普遍采用. 但是大量实践证明这个假设太简化了. Engle 1982 年提出了条件异方差模型.

78.14 ARCH(条件异方差模型)与GARCH等

参考 [24] 5.6

78.14.1 起源

Engle 1982 年提出了ARCH(条件异方差)模型. [5]

78.14.2 ARCH

条件异方差模型(Autoregressive conditional heteroskedasticity, ARCH)全称为自回归条件异方差模型. 其结构为

$$\begin{aligned} x_t &= f(t, x_{t-1}, x_{t-2}, \cdots) + \epsilon_t \\ \epsilon t_t &= \sqrt{h_t} e_t \\ h_t &= \omega + \sum_{j=1}^{q} \lambda_j \epsilon_{t-j}^2 \end{aligned}$$

[5] 1987 年, 计量经济学家 C. Granger 提出了协整(co-integration)理论, 多变量时间序列建模过程中'变量平稳'不再是必须条件, 只要求它们的某种线性组合平稳. Granger 和 Engle 一起获得 2003 年诺贝尔经济学奖

其中 $f(t, x_{t-1}, x_{t-2}$ 为回归函数. λ_j 为系数

原理如下

- 假设数据有异方差性

$$Var(\epsilon_t) = h_t$$

- 在正态分布假定下有

$$\epsilon_t / \sqrt{h_t} \sim N(0, 1)$$

- 异方差等价于残差平方的均值

$$E(\epsilon_t^2) = h_t$$

- 使用残差平方序列的自相关函数(ACF)可以考察异方差的自相关性
- 考察结果无外乎下面两个
  - 自相关系数恒为零. 表示不能有历史数据预测未来的方差.
  - 某个自相关系数不为零, 说明异方差存在自相关性, 我们可以由历史数据预测未来的方差.

实质是使用误差平方序列的 q 阶移动平均MA(q)拟合当前方差. 由于MA(q)具有 q 阶截尾, 故 ARCH 实际上只适合具有短期异方差自相关过程的数据.

78.14.3 GARCH

GARCH 为广义条件异方差模型(Generalized autoregressive conditional heteroskedasticity, GARCH)

有些长期自相关的异方差会产生高的MA阶数, 并影响精度.

为修正此问题, Bollerslov (1985) 提出了 GARCH 模型. 其结构为

$$\begin{aligned} x_t &= f(t, x_{t-1}, x_{t-2}, \cdots) + \epsilon_t \\ \epsilon t_t &= \sqrt{h_t} e_t \\ h_t &= \omega + \sum_{i=1}^{p} \eta_i h_{t-i} \epsilon_{t-j}^2 + \sum_{j=1}^{q} \lambda_j \epsilon_{t-j}^2 \end{aligned}$$

其中 $f(t, x_{t-1}, x_{t-2}$ 为回归函数. η_i, λ_j 为系数.

实际上在 ARCH 的基础上增加了考虑异方差函数的 p 阶自相关性. 那么, ARCH(q) 就是 $p = 0$ 的 GARCH(p,q).

GARCH 要求

- 参数非负, 即 $\omega > 0, \eta_i >= 0, \lambda_j >= 0$
- 参数有界, 即 $\sum_{i=1}^{p} \eta_i + + \sum_{j=1}^{q} \lambda_j < 1$

78.14.4 TODO: 其它变体

EGARCH(指数GARCH), IGARCH(方差无穷GARCH), GARCH-M(依均值GARCH), AR-GARCH, NGARCH(非线性)

78.14.5 例子

下面是 garch() 函数在线例子

```
library(tseries)
n <- 1100
a <- c(0.1, 0.5, 0.2)  # ARCH(2) coefficients
e <- rnorm(n)
x <- double(n)
x[1:2] <- rnorm(2, sd = sqrt(a[1]/(1.0-a[2]-a[3])))
```

```
# 产生 ARCH(2)过程
for(i in 3:n)  # Generate ARCH(2) process
{
x[i] <- e[i]*sqrt(a[1]+a[2]*x[i-1]^2+a[3]*x[i-2]^2)
}
x <- ts(x[101:1100])
# 拟合ARCH(2)
x.arch <- garch(x, order = c(0,2))  # Fit ARCH(2)
# 诊断检验, 检验残差是否随机. 使用方法: Jarque Bera Test, Box-Ljung test
summary(x.arch)                    # Diagnostic tests
plot(x.arch)

data(EuStockMarkets)
dax <- diff(log(EuStockMarkets))[,"DAX"]
dax.garch <- garch(dax)  # Fit a GARCH(1,1) to DAX returns
summary(dax.garch)      # ARCH effects are filtered. However,
plot(dax.garch)         # conditional normality seems to be violated
```

$$\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \cdots + \alpha_q \epsilon_{t-q}^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2$$

78.15 co-integration(协整)

包 urca 执行单位根检验和co-integration分析

78.15.1 起源

1987 年, 计量经济学家 C. Granger 与 Engle 提出了协整(co-integration)理论, 多变量时间序列建模过程中'变量平稳'不再是必须条件, 只要求它们的某种线性组合平稳. [6]

[6]Engle 1982 年提出了ARCH(条件异方差)模型. Granger 和 Engle 一起获得 2003 年诺贝尔经济学奖

78.15.2 概念

有些序列的自身虽然非平稳, 但是某些序列之间具有紧密长期的均衡关系.

例如: 农村家庭人均纯收入对数序列 (lnx_i) 和人均生活消费 (lny_i), 自身都是非平稳的, 但是它们之间具有非常稳定的线性相关关系. 构造回归模型

$$y_i = \beta_0 + \sum_{i=1}^{k} \beta_i x_{it} + \epsilon_t$$

假定回归残差 ϵ_t 平稳, 我们称 x_i, y_i 之间具有协整关系.

意味着, 我们建模不需要所有序列平稳, 只需要有协整关系即可. 这极大拓宽了动态建模的范围.

78.15.3 Phillips-Ouliaris test

Phillips-Ouliaris test 检验多元时间序列是否协整.

若 x 为多元时间序列. Phillips-Ouliaris test 的零假设是: x 非协整.

下面是 R 的例子

```
> library(tseries)
> # no cointegration (非协整)
> x <- ts(diffinv(matrix(rnorm(2000),1000,2)))
> po.test(x)

        Phillips-Ouliaris Cointegration Test

data:  x
Phillips-Ouliaris demeaned = -8.7542, Truncation lag parameter = 10,
p-value = 0.15
```

```
Warning message:
In po.test(x) : p-value greater than printed p-value

# 协整
> x <- diffinv(rnorm(1000))
> y <- 2.0-3.0*x+rnorm(x,sd=5)
> z <- ts(cbind(x,y))  # cointegrated
> po.test(z)

       Phillips-Ouliaris Cointegration Test

data:  z
Phillips-Ouliaris demeaned = -1170.862, Truncation lag parameter = 10,
p-value = 0.01

Warning message:
In po.test(z) : p-value smaller than printed p-value
```

Chapter 79

VAR模型(少例子)

来自 http://en.wikipedia.org/wiki/Vector_autoregression

http://en.wikipedia.org/wiki/General_matrix_notation_of_a_VAR(p)

部分翻译, 部分未翻译. 讲解很好.

VAR(Vector autoregression)模型(向量自回归模型)是一个经济模型, 用来发掘多元时间序列的变化和相互依赖关系. 是AR模型的推广.

79.1 简化模型的定义

其数学描述如下

79.1.1 Var(p)

$$y_t = c + A_1 y_{t-1} + A_2 y_{t-2} + \cdots + A_p y_{t-p} + e_t$$

79.1.2 大矩阵形式

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \\ \vdots \\ y_{k,t} \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{bmatrix} + \begin{bmatrix} a_{1,1}^1 & a_{1,2}^1 & \cdots & a_{1,k}^1 \\ a_{2,1}^1 & a_{2,2}^1 & \cdots & a_{2,k}^1 \\ \vdots & \vdots & \ddots & \vdots \\ a_{k,1}^1 & a_{k,2}^1 & \cdots & a_{k,k}^1 \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \\ \vdots \\ y_{k,t-1} \end{bmatrix} + \cdots + \begin{bmatrix} a_{1,1}^p & a_{1,2}^p & \cdots & a_{1,k}^p \\ a_{2,1}^p & a_{2,2}^p & \cdots & a_{2,k}^p \\ \vdots & \vdots & \ddots & \vdots \\ a_{k,1}^p & a_{k,2}^p & \cdots & a_{k,k}^p \end{bmatrix} \begin{bmatrix} y_{1,t-p} \\ y_{2,t-p} \\ \vdots \\ y_{k,t-p} \end{bmatrix} + \begin{bmatrix} e_{1,t} \\ e_{2,t} \\ \vdots \\ e_{k,t} \end{bmatrix}$$

79.1.3 方程式形式

Rewriting the y variables one to one gives:

$$\begin{aligned} y_{1,t} &= c_1 + a_{1,1}^1 y_{1,t-1} + a_{1,2}^1 y_{2,t-1} + \cdots + a_{1,k}^1 y_{k,t-1} + \cdots + a_{1,1}^p y_{1,t-p} + a_{1,2}^p y_{2,t-p} + \cdots + a_{1,k}^p y_{k,t-p} \\ y_{2,t} &= c_2 + a_{2,1}^1 y_{1,t-1} + a_{2,2}^1 y_{2,t-1} + \cdots + a_{2,k}^1 y_{k,t-1} + \cdots + a_{2,1}^p y_{1,t-p} + a_{2,2}^p y_{2,t-p} + \cdots + a_{2,k}^p y_{k,t-p} \\ y_{k,t} &= c_k + a_{k,1}^1 y_{1,t-1} + a_{k,2}^1 y_{2,t-1} + \cdots + a_{k,k}^1 y_{k,t-1} + \cdots + a_{k,1}^p y_{1,t-p} + a_{k,2}^p y_{2,t-p} + \cdots + a_{k,k}^p y_{k,t-p} \end{aligned}$$

79.1.4 浓缩矩阵

$$Y = BZ + U$$

其中

$$Y = \begin{bmatrix} y_p & y_{p+1} & \cdots & y_T \end{bmatrix} = \begin{bmatrix} y_{1,p} & y_{1,p+1} & \cdots & y_{1,T} \\ y_{2,p} & y_{2,p+1} & \cdots & y_{2,T} \\ \vdots & \vdots & \vdots & \vdots \\ y_{k,p} & y_{k,p+1} & \cdots & y_{k,T} \end{bmatrix}$$

$$B = \begin{bmatrix} c & A_1 & A_2 & \cdots & A_p \end{bmatrix} = \begin{bmatrix} c_1 & a_{1,1}^1 & a_{1,2}^1 & \cdots & a_{1,k}^1 & \cdots & a_{1,1}^p & a_{1,2}^p & \cdots & a_{1,k}^p \\ c_2 & a_{2,1}^1 & a_{2,2}^1 & \cdots & a_{2,k}^1 & \cdots & a_{2,1}^p & a_{2,2}^p & \cdots & a_{2,k}^p \\ \vdots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\ c_k & a_{k,1}^1 & a_{k,2}^1 & \cdots & a_{k,k}^1 & \cdots & a_{k,1}^p & a_{k,2}^p & \cdots & a_{k,k}^p \end{bmatrix}$$

$$Z = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ y_{p-1} & y_p & \cdots & y_{T-1} \\ y_{p-2} & y_{p-1} & \cdots & y_{T-2} \\ \vdots & \vdots & \ddots & \vdots \\ y_0 & y_1 & \cdots & y_{T-p} \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ y_{1,p-1} & y_{1,p} & \cdots & y_{1,T-1} \\ y_{2,p-1} & y_{2,p} & \cdots & y_{2,T-1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{k,p-1} & y_{k,p} & \cdots & y_{k,T-1} \\ y_{1,p-2} & y_{1,p-1} & \cdots & y_{1,T-2} \\ y_{2,p-2} & y_{2,p-1} & \cdots & y_{2,T-2} \\ \vdots & \vdots & \ddots & \vdots \\ y_{k,p-2} & y_{k,p-1} & \cdots & y_{k,T-2} \\ \vdots & \vdots & \ddots & \vdots \\ y_{1,0} & y_{1,1} & \cdots & y_{1,T-p} \\ y_{2,0} & y_{2,1} & \cdots & y_{2,T-p} \\ \vdots & \vdots & \ddots & \vdots \\ y_{k,0} & y_{k,1} & \cdots & y_{k,T-p} \end{bmatrix}$$

and

$$U = \begin{bmatrix} e_p & e_{p+1} & \cdots & e_T \end{bmatrix} = \begin{bmatrix} e_{1,p} & e_{1,p+1} & \cdots & e_{1,T} \\ e_{2,p} & e_{2,p+1} & \cdots & e_{2,T} \\ \vdots & \vdots & \ddots & \vdots \\ e_{k,p} & e_{k,p+1} & \cdots & e_{k,T} \end{bmatrix}.$$

下面就可以解系数矩阵B了(例如,使用一般最小方差(ordinary least squares)估计$Y \approx BZ$)

79.1.5 解释

VAR模型描述k维数据从时间$t = 1, ..., T$的变化, 将之看作它自己过去的一个线性函数. 时间t的变量y_t是一个$k * 1$的向量, 例如, 第i个变量是GDP, 那么$y_{i,t}$是时间t的GDP.

其中e_t满足

1. $\mathrm{E}(e_t) = 0$ 误差均值为零
2. $\mathrm{E}(e_t e_t') = \Omega$ 误差协方差矩阵是正则的
3. $\mathrm{E}(e_t e_{t-k}') = 0$ 对于$k \neq 0$, 误差互协方差为零

79.1.6 Order of integration of the variables

Note that all the variables used have to be of the same order of integration. We have so the following cases:

- All the variables are I(0) (stationary): one is in the standard case, ie. a VAR in level
- All the variables are I(d) (non-stationary) with d¿1:
  - The variables are cointegrated: the error correction term has to be included in the VAR. The model becomes a Vector error correction model (VECM) which can be seen as a restricted VAR.
  - The variables are not cointegrated: the variables have first to be differenced d times and one has a VAR in difference.

79.1.7 简单例子

二维向量的VAR(1)可以写作

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \end{bmatrix} + \begin{bmatrix} e_{1,t} \\ e_{2,t} \end{bmatrix}$$

或等价的

$$\begin{aligned} y_{1,t} &= c_1 + A_{1,1}y_{1,t-1} + A_{1,2}y_{2,t-1} + e_{1,t} \\ y_{2,t} &= c_2 + A_{2,1}y_{1,t-1} + A_{2,2}y_{2,t-1} + e_{2,t} \end{aligned}$$

注意到, 每个向量有一个方程式, 每个向量当前的状态不不仅依赖于自己的过去状态, 还依赖于其它序列的过去状态

79.1.8 将VAR(p)写作VAR(1)

通过变换系数, 我们总可以把延迟为p的形式写作延迟为1的形式. 例如VAR(2)模型

$$yt = c + A1yt`1 + A2yt`2 + et$$

可以写作VAR(1)模型

$$\begin{bmatrix} y_t \\ y_{t-1} \end{bmatrix} = \begin{bmatrix} c \\ 0 \end{bmatrix} + \begin{bmatrix} A_1 & A_2 \\ I & 0 \end{bmatrix} \begin{bmatrix} y_{t-1} \\ y_{t-2} \end{bmatrix} + \begin{bmatrix} e_t \\ 0 \end{bmatrix}$$

其中I为单位矩阵.

VAR(1)形式分析起来更加方便, 写起来更简便.

79.2 Structural vs. reduced form

79.2.1 Structural VAR

p延迟的structural VAR为

$$B_0 y_t = c_0 + B_1 y_{t-1} + B_2 y_{t-2} + \cdots + B_p y_{t-p} + \epsilon_t$$

其中c_0为$k1$常数向量, B_i是kk矩阵(for every $i = 0, ..., p$), $_t$为$k1$误差向量. B_0矩阵的主对角成分都为1.

误差项$_t$(structural shocks)满足定义中条件(1) - (3), 即误差项$_t$(structural shocks)不相关($E(_t') = = 0$.

例如二维的VAR(1)为

$$\begin{bmatrix} 1 & B_{0;1,2} \\ B_{0;2,1} & 1 \end{bmatrix} \begin{bmatrix} y_{1,t} \\ y_{2,t} \end{bmatrix} = \begin{bmatrix} c_{0;1} \\ c_{0;2} \end{bmatrix} + \begin{bmatrix} B_{1;1,1} & B_{1;1,2} \\ B_{1;2,1} & B_{1;2,2} \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \end{bmatrix} + \begin{bmatrix} \epsilon_{1,t} \\ \epsilon_{2,t} \end{bmatrix}$$

其中

$$\Sigma = \mathrm{E}(\epsilon_t \epsilon_t') = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}$$

显式写出第一个方程式, 并带入$y_{2,t}$到右边得到

$$y_{1,t} = c_{0;1} - B_{0;1,2} y_{2,t} + B_{1;1,1} y_{1,t-1} + B_{1;1,2} y_{2,t-2} + \epsilon_{1,t}$$

注意到如果$B_{0;1,2}$不等于零, $y_{2,t}$可以影响同时的$_{1,t}$. 这同B_0为单位矩阵的情况不同, 其$y_{2,t}$可以直接影响$y_{1,t+1}$从而影响将来的值, 但不是$y_{1,t}$.

因为普通最小方差估计(ordinary least squares estimation)确定structural VAR参数的问题, 即产生无解估计(yield inconsistent parameter estimates), 我们可以将其表示为简化方式(reduced form).

下面未翻译

From an economic point of view it is considered that, if the joint dynamics of a se

1. Error terms are not correlated. The structural, economic shocks which dri

2. Variables can have a contemporaneous impact on other variables. This is a

79.2.2 Reduced VAR

左乘B_0的逆

$$y_t = B_0^{-1} c_0 + B_0^{-1} B_1 y_{t-1} + B_0^{-1} B_2 y_{t-2} + \cdots + B_0^{-1} B_p y_{t-p} + B_0^{-1} \epsilon_t$$

并表示为

$$B_0^{-1}c_0 = c, \quad B_0^{-1}B_i = A_i \text{ for } i = 1, \ldots, p \text{ and } B_0^{-1}\epsilon_t = e_t$$

可以得到p阶简化的VAR模型

$$y_t = c + A_1 y_{t-1} + A_2 y_{t-2} + \cdots + A_p y_{t-p} + e_t$$

下面未翻译

```
Note that in the reduced form all right hand side variables are predetermined at ti

However, the error terms in the reduced VAR are composites of the structural shocks
```

$$\Omega = \mathrm{E}(e_t e_t') = \mathrm{E}(B_0^{-1}\epsilon_t \epsilon_t' (B_0^{-1})') = B_0^{-1}\Sigma(B_0^{-1})'$$

79.3 估计

79.3.1 估计回归系数

由精简形式

$$Y = BZ + U$$

得到B的Multivariate Least Square (MLS):

$$\hat{B} = YZ'(ZZ')^{-1}$$

还可以写作

$$\mathrm{Vec}(\hat{B}) = ((ZZ')^{-1}Z \otimes I_k)\ \mathrm{Vec}(Y)$$

下面未翻译

Where \otimes denotes the Kronecker product and Vec the vectorization of the matrix

This estimator is consistent and asymptotically efficient. It is furthemore equal t

* As the explanatory variables are the same in each equation, the Multivariate M

79.3.2 误差协方差矩阵的估计

As in the standard case, the MLE estimator of the covariance matrix differs from the OLS estimator.

MLE estimator: $\hat{\Sigma} = \frac{1}{T}\sum_{t=1}^{T} \hat{\epsilon}_t \hat{\epsilon}_t'$

OLS estimator: $\hat{\Sigma} = \frac{1}{T-kp-1}\sum_{t=1}^{T} \hat{\epsilon}_t \hat{\epsilon}_t'$ for a model with a constant, k variables and p lags

In a matrix notation, this gives:

$$\hat{\Sigma} = \frac{1}{T-kp-1}(Y - \hat{B}Z)(Y - \hat{B}Z)'$$

79.3.3 参数协方差矩阵的估计

$$\widehat{\mathrm{Cov}}(\mathrm{Vec}(\hat{B})) = (ZZ')^{-1} \otimes \hat{\Sigma}$$

79.4 参考文献

略

79.5 相关函数

ar() 可以实现平稳序列的VAR分析

dse包, dlm包, mAr包等都有相关函数.

Chapter 80

卡尔曼滤波(理论,少例子)

参考 http://zh.wikipedia.org/wiki/卡尔曼滤波

参考"卡尔曼滤波器最好的入门教程" http://bbs.powershock.cn/thread-45-1-1.html

参考 http://www.cs.unc.edu/ welch/kalman/

Andrew D. Straw 非常好的一个介绍,并一个python的例子 http://www.cs.unc.edu/ welch/kalman/kalmanIntro.html

姚旭晨翻译的Andrew D. Straw 的介绍并改编的matlab例子,非常好 http://yaoxuchen.googlepages.com/kalman

为方便查看,下面内容全文来自 http://zh.wikipedia.org/wiki/卡尔曼滤波

80.1 介绍

卡尔曼滤波是一种高效率的递归滤波器(自回归滤波器),它能够从一系列的不完全及包含噪声的测量(英文:measurement)中，估计动态系统的状态。

80.2 应用实例

卡尔曼滤波的一个典型实例是从一组有限的，包含噪声的，对物体位置的观察序列(可能有偏差)预测出物体的位置的坐标及速度。在很多工程应用(如雷达、计算机视觉)中都可以找到它的身影。同时，卡尔曼滤波也是控制理论以及控制系统工程中的一个重要课题。

例如,对于雷达来说，人们感兴趣的是其能够跟踪目标。但目标的位置、速度、加速度的测量值往往在任何时候都有噪声。卡尔曼滤波利用目标的动态信息，设法去掉噪声的影响，得到一个关于目标位置的好的估计。这个估计可以是对当前目标位置的估计(滤波)，也可以是对于将来位置的估计(预测)，也可以是对过去位置的估计(插值或平滑)。

80.3 命名

这种滤波方法以它的发明者鲁道夫.E.卡尔曼(Rudolph E. Kalman)命名，但是根据文献可知实际上Peter Swerling在更早之前就提出了一种类似的算法。

斯坦利.施密特(Stanley Schmidt)首次实现了卡尔曼滤波器。卡尔曼在NASA埃姆斯研究中心访问时，发现他的方法对于解决阿波罗计划的轨道预测很有用，后来阿波罗飞船的导航电脑便使用了这种滤波器。

关于这种滤波器的论文由Swerling (1958)、Kalman (1960)与Kalman and Bucy (1961)发表。

目前,卡尔曼滤波已经有很多不同的实现.卡尔曼最初提出的形式现在一般称为简单卡尔曼滤波器。除此以外，还有施密特扩展滤波器、信息滤波器以及很多Bierman, Thornton 开发的平方根滤波器的变种。也许最常见的卡尔曼滤波器是锁相环，它在收音机、计算机和几乎任何视频或通讯设备中广泛存在。

以下的讨论需要线性代数以及概率论的一般知识。

80.4 基本动态系统模型

卡尔曼滤波建立在线性代数和隐马尔可夫模型(hidden Markov model)上。其基本动态系统可以用一个马尔可夫链表示，该马尔可夫链建立在一个被高斯噪声(即正态分布的噪声)干扰的线性算子上的。系统的状态可以用一个元素为实数的向量表示。随着离散时间的每一个增加,这个线性算子就会作用在当前状态上，产生一个新的状态,并也会带入一些噪声,同时系统的一些已知的控制器的控制信息也会被加入。同时，另一个受噪声干扰的线性算子产生出这些隐含状态的可见输出。

为了从一系列有噪声的观察数据中用卡尔曼滤波器估计出被观察过程的内部状态，我们必须把这个过程在卡尔曼滤波的框架下建立模型。也就是说对于每一步k，定义矩阵F_k, H_k, Q_k, R_k，有时也需要定义B_k，如下图(略, 卡尔曼滤波器的模型。圆圈代表向量，方块代表矩阵，星号代表高斯噪声，其协方差矩阵在右下方标出。)

卡尔曼滤波模型假设k时刻的真实状态是从(k 1)时刻的状态演化而来，符合下式

$$\mathbf{x}_k = \mathbf{F}_k\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{u}_k + \mathbf{w}_k$$

其中

- F_k 是作用在$x_{k`1}$上的状态变换模型(/矩阵/矢量)。
- B_k 是作用在控制器向量u_k上的输入－控制模型。
- w_k 是过程噪声，并假定其符合均值为零，协方差矩阵为Q_k的多元正态分布。

$$\mathbf{w}_k \sim N(0, \mathbf{Q}_k)$$

时刻k，对真实状态 x_k的一个测量z_k满足下式:

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k$$

其中H_k是观测模型,它把真实状态空间映射成观测空间，v_k 是观测噪声，其均值为零，协方差为R_k,且服从正态分布。

$$\mathbf{v}_k \sim N(0, \mathbf{R}_k)$$

初始状态以及每一时刻的噪声$x_0, w_1, ..., w_k, v_1...v_k$ 都为认为是互相独立的.

实际上，很多真实世界的动态系统都并不确切的符合这个模型；但是由于卡尔曼滤波器被设计在有噪声的情况下工作,一个近似的符合已经可以使这个滤波器非常有用了。更多其它更复杂的卡尔曼滤波器的变种，在下边讨论中有描述。

80.5 卡尔曼滤波器

卡尔曼滤波是一种递归的估计，即只要获知上一时刻状态的估计值以及当前状态的观测值就可以计算出当前状态的估计值，因此不需要记录观测或者估计的历史信息。卡尔曼滤波器与大多数滤波器不同之处，在于它是一种纯粹的时域滤波器，它不需要像低通滤波器等频域滤波器那样，需要在频域设计再转换到时域实现。

卡尔曼滤波器的状态由以下两个变量表示：

- $\hat{\mathbf{x}}_{k|k}$，在时刻k 的状态的估计；
- $\mathbf{P}_{k|k}$，误差相关矩阵，度量估计值的精确程度。

卡尔曼滤波器的操作包括两个阶段：预测与更新。在预测阶段，滤波器使用上一状态的估计，做出对当前状态的估计。在更新阶段，滤波器利用对当前状态的观测值优化在预测阶段获得的预测值，以获得一个更精确的新估计值。

80.5.1 预测

- $\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k\mathbf{u}_k$ (预测状态)
- $\mathbf{P}_{k|k-1} = \mathbf{F}_k\mathbf{P}_{k-1|k-1}\mathbf{F}_k^T + \mathbf{Q}_k$ (预测估计协方差)

80.5.2 更新

- $\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}$ (测量余量, measurement residual)
- $\mathbf{S}_k = \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k$ (测量余量协方差)
- $\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{S}_k^{-1}$ (卡尔曼增益)
- $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\tilde{\mathbf{y}}_k$ (更新的状态估计)
- $\mathbf{P}_{k|k} = (I - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}$ (更新的协方差估计)

使用上述公式计算$\mathbf{P}_{k|k}$仅在最优卡尔曼增益的时候有效。使用其他增益的话，公式要复杂一些，请参见推导。

80.5.3 不变量(Invariant)

如果模型准确，而且$\hat{\mathbf{x}}_{0|0}$与$\mathbf{P}_{0|0}$的值准确的反映了最初状态的分布，那么以下不变量就保持不变：所有估计的误差均值为零

- $\mathrm{E}[\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}] = \mathrm{E}[\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}] = 0$
- $\mathrm{E}[\tilde{\mathbf{y}}_k] = 0$

且协方差矩阵准确的反映了估计的协方差:

- $\mathbf{P}_{k|k} = \mathrm{cov}(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})$
- $\mathbf{P}_{k|k-1} = \mathrm{cov}(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})$
- $\mathbf{S}_k = \mathrm{cov}(\tilde{\mathbf{y}}_k)$

请注意，其中$\mathrm{E}[\mathbf{a}]$表示a的期望值, $\mathrm{cov}(\mathbf{a}) = \mathrm{E}[\mathbf{a}\mathbf{a}^T]$。

80.6 实例

考虑在无摩擦的、无限长的直轨道上的一辆车。该车最初停在位置0处,但时不时受到随机的冲击。我们每隔Δt秒即测量车的位置，但是这个测量是非精确的；我们想建立一个关于其位置以及速度的模型。我们来看如何推导出这个模型以及如何从这个模型得到卡尔曼滤波器。

因为车上无动力,所以我们可以忽略掉B_k和u_k。由于F、H、R和Q是常数，所以时间下标可以去掉。

车的位置以及速度(或者更加一般的，一个粒子的运动状态)可以被线性状态空间描述如下:

$$\mathbf{x}_k = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$$

其中$\dot{x}$是速度，也就是位置对于时间的导数。我们假设在(k`1)时刻与k时刻之间，车受到a_k的加速度,其符合均值为0，标准差为σ_a的正态分布。根据牛顿运动定律，我们可以推出

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{G}a_k$$

其中

$$\mathbf{F} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$

且

$$\mathbf{G} = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix}$$

我们可以发现((因为 σ_a 是一个标量)

$$\mathbf{Q} = \operatorname{cov}(\mathbf{G}a) = \mathrm{E}[(\mathbf{G}a)(\mathbf{G}a)^T] = \mathbf{G}\mathrm{E}[a^2]\mathbf{G}^T = \mathbf{G}[\sigma_a^2]\mathbf{G}^T = \sigma_a^2\mathbf{G}\mathbf{G}^T$$

在每一时刻,我们对其位置进行测量,测量受到噪声干扰.我们假设噪声服从正态分布，均值为0，标准差为σ_z。

$$\mathbf{z}_k = \mathbf{H}\ \mathbf{x}_k + \mathbf{v}_k$$

其中

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

且

$$\mathbf{R} = \mathrm{E}[\mathbf{v}_k \mathbf{v}_k^T] = \begin{bmatrix} \sigma_z^2 \end{bmatrix}$$

如果我们知道足够精确的车最初的位置，那么我们可以初始化

$$\hat{\mathbf{x}}_{0|0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

并且,我们告诉滤波器我们知道确切的初始位置,我们给出一个协方差矩阵：

$$\mathbf{P}_{0|0} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

如果我们不确切的知道最初的位置与速度，那么协方差矩阵可以初始化为一个对角线元素是B的矩阵，B取一个合适的比较大的数。

$$\mathbf{P}_{0|0} = \begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix}$$

此时，与使用模型中已有信息相比，滤波器更倾向于使用初次测量值的信息。

80.7 推导

80.7.1 推导后验协方差矩阵

按照上边的定义，我们从误差协方差Pk—k开始推导如下：

$$\mathbf{P}_{k|k} = \mathrm{cov}(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})$$

代入$\hat{\mathbf{x}}_{k|k}$

$$\mathbf{P}_{k|k} = \mathrm{cov}(\mathbf{x}_k - (\hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\tilde{\mathbf{y}}_k))$$

再代入 $\tilde{\mathbf{y}}_k$

$$\mathbf{P}_{k|k} = \mathrm{cov}(\mathbf{x}_k - (\hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1})))$$

与 $\mathbf{z}_k$

$$\mathbf{P}_{k|k} = \mathrm{cov}(\mathbf{x}_k - (\hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1})))$$

整理误差向量，得

$$\mathbf{P}_{k|k} = \mathrm{cov}((I - \mathbf{K}_k\mathbf{H}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}) - \mathbf{K}_k\mathbf{v}_k)$$

因为测量误差vk 与其他项是非相关的，因此有

$$\mathbf{P}_{k|k} = \mathrm{cov}((I - \mathbf{K}_k\mathbf{H}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})) + \mathrm{cov}(\mathbf{K}_k\mathbf{v}_k)$$

利用协方差矩阵的性质，此式可以写作

$$\mathbf{P}_{k|k} = (I - \mathbf{K}_k\mathbf{H}_k)\mathrm{cov}(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(I - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathrm{cov}(\mathbf{v}_k)\mathbf{K}_k^T$$

使用不变量$P_{k|k-1}$以及R_k的定义这一项可以写作：$\mathbf{P}_{k|k} = (I - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}(I - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T$ 这一公式对于任何卡尔曼增益Kk都成立。如果Kk是最优卡尔曼增益，则可以进一步简化，请见下文。

80.7.2 最优卡尔曼增益的推导

卡尔曼滤波器是一个最小均方误差估计器，后验状态误差估计(英文:a posteriori state estimate)是

$$\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}$$

我们最小化这个矢量幅度平方的期望值，$\mathrm{E}[|\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}|^2]$，这等同于最小化后验估计协方差矩阵 $P_{k|k}$的迹(trace)。将上面方程中的项展开、抵消，得到：

$$\begin{aligned}\mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{K}_k^T + \mathbf{K}_k(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k)\mathbf{K}_k^T \\ &= \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{K}_k^T + \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T\end{aligned}$$

当矩阵导数是 0 的时候得到$P_{k|k}$的迹(trace)的最小值：

$$\frac{d\,\mathrm{tr}(\mathbf{P}_{k|k})}{d\,\mathbf{K}_k} = -2(\mathbf{H}_k\mathbf{P}_{k|k-1})^T + 2\mathbf{K}_k\mathbf{S}_k = 0$$

此处须用到一个常用的式子, 如下：

$$\frac{d\,\mathrm{tr}(\mathbf{BAC})}{d\,\mathbf{A}} = B^T C^T$$

从这个方程解出卡尔曼增益K_k:

$$\begin{aligned}\mathbf{K}_k\mathbf{S}_k &= (\mathbf{H}_k\mathbf{P}_{k|k-1})^T = \mathbf{P}_{k|k-1}\mathbf{H}_k^T \\ \mathbf{K}_k &= \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{S}_k^{-1}\end{aligned}$$

这个增益称为最优卡尔曼增益，在使用时得到最小均方误差。

80.7.3 后验误差协方差公式的化简

在卡尔曼增益等于上面导出的最优值时，计算后验协方差的公式可以进行简化。在卡尔曼增益公式两侧的右边都乘以 $S_kK_k^T$ 得到

$$\mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T = \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{K}_k^T$$

根据上面后验误差协方差展开公式，

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{K}_k^T + \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T$$

最后两项可以抵消，得到

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1} = (I - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}.$$

这个公式的计算比较简单，所以实际中总是使用这个公式，但是需注意这公式仅在使用最优卡尔曼增益的时候它才成立。如果算术精度总是很低而导致数值稳定性出现问题，或者特意使用非最优卡尔曼增益，那么就不能使用这个简化；必须使用上面导出的后验误差协方差公式。

80.8 与递归Bayesian估计之间的关系

假设真正的状态是无法观察的马尔可夫过程，测量结果是从隐性马尔可夫模型观察到的状态。

Image:HMMKalmanFilterDerivation.png

根据马尔可夫假设，真正的状态仅受最近一个状态影响而与其它以前状态无关。

$$p(\mathbf{x}_k|\mathbf{x}_0, \ldots, \mathbf{x}_{k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$$

与此类似，在时刻 k 测量只与当前状态有关而与其它状态无关。

$$p(\mathbf{z}_k|\mathbf{x}_0, \ldots, \mathbf{x}_k) = p(\mathbf{z}_k|\mathbf{x}_k)$$

根据这些假设，隐性马尔可夫模型所有状态的概率分布可以简化为：

$$p(\mathbf{x}_0, \ldots, \mathbf{x}_k, \mathbf{z}_1, \ldots, \mathbf{z}_k) = p(\mathbf{x}_0)\prod_{i=1}^{k} p(\mathbf{z}_i|\mathbf{x}_i)p(\mathbf{x}_i|\mathbf{x}_{i-1})$$

然而，当卡尔曼滤波器用来估计状态x的时候，我们感兴趣的机率分布，是基于目前为止所有个测量值来得到的当前状态之机率分布

$$p(\mathbf{x}_k|\mathbf{Z}_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})\, d\mathbf{x}_{k-1}$$

80.9 信息滤波器

80.9.1 非线性滤波器

基本卡尔曼滤波器(The basic Kalman filter)是限制在线性的假设之下。然而，大部份非平凡的(non-trial)的系统都是非线性系统。其中的“非线性性质”(non-linearity)可能是伴随存在过程模型(process model)中或观测模型(observation model)中，或者两者兼有之。

80.9.2 扩展卡尔曼滤波器

在扩展卡尔曼滤波器(Extended Kalman Filter，简称EKF)中状态转换和观测模型不需要是状态的线性函数，可替换为(可微的)函数。

$$\begin{aligned} \mathbf{x}_k &= f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k) \\ \mathbf{z}_k &= h(\mathbf{x}_k, \mathbf{v}_k) \end{aligned}$$

函数f可以用来从过去的估计值中计算预测的状态，相似的，函数h可以用来以预测的状态计算预测的测量值。然而f和h不能直接的应用在协方差中，取而代之的是计算偏导矩阵(Jacobian)。

在每一步中使用当前的估计状态计算Jacobian矩阵，这几个矩阵可以用在卡尔曼滤波器的方程中。这个过程，实质上将非线性的函数在当前估计值处线性化了。

这样一来，卡尔曼滤波器的等式为：

预测

$$\begin{aligned}\hat{\mathbf{x}}_{k|k-1} &= f(\mathbf{x}_{k-1}, \mathbf{u}_k, 0) \\ \mathbf{P}_{k|k-1} &= \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k\end{aligned}$$

使用Jacobians矩阵更新模型

$$\begin{aligned}\mathbf{F}_k &= \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k} \\ \mathbf{H}_k &= \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k|k-1}}\end{aligned}$$

更新

$$\begin{aligned}\tilde{\mathbf{y}}_k &= \mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1}, 0) \\ \mathbf{S}_k &= \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \\ \mathbf{K}_k &= \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k \\ \mathbf{P}_{k|k} &= (I - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}\end{aligned}$$

预测

如同扩展卡尔曼滤波器(EKF)一样, UKF的预测过程可以独立于UKF的更新过程之外，与一个线性的(或者确实是扩展卡尔曼滤波器的)更新过程合并来使用；或者，UKF的预测过程与更新过程在上述中地位互换亦可。

80.10 应用

- 自动驾驶仪
- 动态定位系统
- 经济学, 特别是宏观经济学，时间序列模型, 以及计量经济学

- 惯性引导系统
- 雷达跟踪器
- 卫星导航系统

80.11 参见

快速卡尔曼滤波

比较: 维纳滤波及 the multimodal Particle filter estimator.

80.12 例子

80.12.1 Andrew D. Straw的例子

最初来自 Andrew D. Straw http://www.scipy.org/Cookbook/KalmanFiltering

姚旭晨改编为matlab http://yaoxuchen.googlepages.com/kalman

me 改编为 R

```
# Kalman filter example demo in Matlab

# This M code is modified from Andrew D. Straw's Python
# implementation of Kalman filter algorithm.
# The original code is here:
# http://www.scipy.org/Cookbook/KalmanFiltering
# Below is the Python version's comments:

# Kalman filter example demo in Python

# A Python implementation of the example given in pages 11-15 of "An
# Introduction to the Kalman Filter" by Greg Welch and Gary Bishop,
# University of North Carolina at Chapel Hill, Department of Computer
```

```
# Science, TR 95-041,
# http://www.cs.unc.edu/~welch/kalman/kalmanIntro.html

# by Andrew D. Straw

# matlab by Xuchen Yao
# R by me

# intial parameters
n_iter = 50;
sz = c(n_iter, 1); # size of array
x = -0.37727; # truth value (typo in example at top of p. 13 calls this z)
z = x + sqrt(0.01)*rnorm(n_iter); # observations (normal about x, sigma=0.1)

Q = 1e-5; # process variance

# allocate space for arrays
xhat=rep(0,n_iter);      # a posteri estimate of x
P=rep(0,n_iter);         # a posteri error estimate
xhatminus=rep(0,n_iter); # a priori estimate of x
Pminus=rep(0,n_iter);    # a priori error estimate
K=rep(0,n_iter);         # gain or blending factor

R = 0.01; # estimate of measurement variance, change to see effect

# intial guesses
xhat[1] = 0.0;
P[1] = 1.0;

for (k in 2:n_iter){
# time update(predict)
   xhatminus[k] = xhat[k-1];
   Pminus[k] = P[k-1]+Q;

# measurement update
   K[k] = Pminus[k]/( Pminus[k]+R );
   xhat[k] = xhatminus[k]+K[k]*(z[k]-xhatminus[k]);
   P[k] = (1-K[k])*Pminus[k];
}
# plot predicted value
plot(z, xlab='Iteration',ylab='Voltage')
```

```
lines(xhat,col='red')
lines(x*rep(1,50))

# plot error
valid_iter = 2:n_iter
plot(Pminus[valid_iter]~valid_iter,t='l');
```

80.12.2 kfilter()函数

它属于sspir包

Chapter 81

谱分析

81.1 推荐

《小波与傅里叶分析基础》[23], 入门极佳, 工科足够.

signal 包: 是一个类似Matlab/Octave信号处理命令的工具. 包含滤波, 采样, 差值, 可视化等命令. 命令全, 比较方便. (R自带的命令不太全)

81.2 介绍

谱分析是根据时间序列的频域性质对其统计推断的方法.

一些具有周期的序列其周期如果是复合的, 则很难通过图来看出. 这时候需要使用频域的方法.

81.3 傅立叶变换(FFT)

```
> x=c(15 , -2, 12, 20, -5 , 0 , -8 , -4 , -8, -22)
> ft=fft(x); ft
```

```
 [1]  -2.00000+ 0.00000i   2.39261-55.36555i -12.61397-13.81682i
 [4]  28.10739+ 3.98825i  51.11397-13.79658i  14.00000- 0.00000i
 [7]  51.11397+13.79658i  28.10739- 3.98825i -12.61397+13.81682i
[10]   2.39261+55.36555i
> Mod(ft) # 相当于 abs(ft)
 [1]  2.00000 55.41722 18.70873 28.38893 52.94321 14.00000 52.94321 28.38893
 [9] 18.70873 55.41722
> abs(ft)
 [1]  2.00000 55.41722 18.70873 28.38893 52.94321 14.00000 52.94321 28.38893
 [9] 18.70873 55.41722

> ft=fft(x); ft
 [1]  -2.00000+ 0.00000i   2.39261-55.36555i -12.61397-13.81682i
 [4]  28.10739+ 3.98825i  51.11397-13.79658i  14.00000- 0.00000i
 [7]  51.11397+13.79658i  28.10739- 3.98825i -12.61397+13.81682i
[10]   2.39261+55.36555i

x<-seq(0,1,by=0.001)

# y 在 100, 200, 300 处有峰值
y <- sin(200*pi*x) +3*sin(400*pi*x)+6*sin(600*pi*x)
op <- par(mfrow=c(3,1))
plot(Mod(fft(y)),t='l') # 模
plot(Re(fft(y)),t='l') # 实部
plot(Im(fft(y)),t='l') # 虚部
par(op)
```

81.4 窗函数

在谱分析的时候为了减小截断边界时产生的吉布斯(Gibbs)效应, 往往需要加窗([23] 第一章). fir1() fir2() spectgram() 函数使用 window 参数加窗. 下面是 signal 包提供的窗函数.

```
bartlett(n)
blackman(n)
boxcar(n)
flattopwin(n, sym = c('symmetric', 'periodic'))
```

```
gausswin(n, w = 2.5)
hamming(n)
hanning(n)
triang(n)

# 查看各种窗函数的形状
n = 51
op = par(mfrow=c(3,3))
plot(bartlett(n), type = "l", ylim = c(0,1))
plot(blackman(n), type = "l", ylim = c(0,1))
plot(boxcar(n), type = "l", ylim = c(0,1))
plot(flattopwin(n), type = "l", ylim = c(0,1))
plot(gausswin(n, 5), type = "l", ylim = c(0,1))
plot(hanning(n), type = "l", ylim = c(0,1))
plot(hamming(n), type = "l", ylim = c(0,1))
plot(triang(n), type = "l", ylim = c(0,1))
par(op)

# kaiser 窗
plot(kaiser(101, 2), type = "l", ylim = c(0,1))
lines(kaiser(101, 10), col = "blue")
lines(kaiser(101, 50), col = "green")

# Dolph-Chebyshev window coefficients
plot(chebwin(50, 100))
```

81.5 Periodogram(周期图)

81.5.1 简介

周期图也叫做样本谱(sample spectrum), 实际上就是离散傅立叶变换.

功率谱估计可以分为经典谱估计方法与现代谱估计方法。

经典谱估计中最简单的就是周期图法，又分为直接法与间

接法。都可以编程实现，很简单。

- 直接法先取N点数据的傅里叶变换(即频谱)，然后取频谱与其共轭的乘积，就得到功率谱的估计；
- 间接法先计算N点样本数据的自相关函数，然后取自相关函数的傅里叶变换，即得到功率谱的估计.

但是周期图法估计出的功率谱不够精细，分辨率比较低。因此需要对周期图法进行修正，可以将信号序列x(n)分为n个不相重叠的小段，分别用周期图法进行谱估计，然后将这n段数据估计的结果的平均值作为整段数据功率谱估计的结果。还可以将信号序列x(n)重叠分段，分别计算功率谱，再计算平均值作为整段数据的功率谱估计。这2种称为分段平均周期图法，一般后者比前者效果好。加窗平均周期图法是对分段平均周期图法的改进，即在数据分段后，对每段数据加一个非矩形窗进行预处理，然后在按分段平均周期图法估计功率谱。相对于分段平均周期图法，加窗平均周期图法可以减小频率泄漏，增加频峰的宽度。welch法就是利用改进的平均周期图法估计估计随机信号的功率谱，它采用信号分段重叠，加窗，FFT等技术来计算功率谱。与周期图法比较，welch法可以改善估计谱曲线的光滑性，大大提高谱估计的分辨率。

现代谱估计主要针对经典谱估计分辨率低和方差性不好提出的，可以极大的提高估计的分辨率和平滑性。可以分为参数模型谱估计和非参数模型谱估计。参数模型谱估计有AR模型，MA模型，ARMA模型等；非参数模型谱估计有最小方差法和MUSIC法等。由于涉及的问题太多，这里不再详述，可以参考有关资料。[1]

81.5.2 例子

```
> x=c(15 , -2, 12, 20, -5 , 0 , -8 , -4 , -8, -22)
> ft=fft(x)
```

[1]来自网络资料

```
# 直接法. 与共轭的乘积
> a=ft*Conj(ft); a # 虚部全部为零. 相当于 abs(ft)^2
 [1]    4.0000+0i 3071.0684+0i  350.0167+0i  805.9316+0i 2802.9833+0i
 [6]  196.0000+0i 2802.9833+0i  805.9316+0i  350.0167+0i 3071.0684+0i

> Re(a/10)
 [1]   0.40000 307.10684  35.00167  80.59316 280.29833  19.60000 280.29833
 [8]  80.59316  35.00167 307.10684

# 间接法. 自相关的傅立叶变换
> b=fft(acf(x,plot=F)$acf); b
, , 1

                  [,1]
 [1,] 0.5000000+0.0000000i
 [2,] 1.5771143-1.1254158i
 [3,] 0.6227612-0.0241158i
 [4,] 0.7826640-0.3125117i
 [5,] 1.4830890-0.2939962i
 [6,] 0.5687430+0.0000000i
 [7,] 1.4830890+0.2939962i
 [8,] 0.7826640+0.3125117i
 [9,] 0.6227612+0.0241158i
[10,] 1.5771143+1.1254158i

> abs(b)
, , 1

         [,1]
 [1,] 0.5000000
 [2,] 1.9374856
 [3,] 0.6232279
 [4,] 0.8427494
 [5,] 1.5119480
 [6,] 0.5687430
 [7,] 1.5119480
 [8,] 0.8427494
 [9,] 0.6232279
[10,] 1.9374856

> cor(a,b)
```

```
[1] 0.9833278

# 图形基本一样
op=par(mfrow=c(2,1))
plot(a,t='l')
plot(b,t='l')
par(op)
```

R 函数 spectrum() 使用两种方法(当参数 method="pgram" 为 spec.pgram(), method="ar" 为 spec.ar())计算周期图. 一种是 pgram 法, 另外一个是 ar 法(AR模型平滑后的周期图). spec.pgram() 用法如下

```
    spec.pgram(x, spans = NULL, kernel, taper = 0.1,
               pad = 0, fast = TRUE, demean = FALSE, detrend = TRUE,
               plot = TRUE, na.action = na.fail, ...)
```

```
# 手工计算的周期图
f=fft(co2)
p=Re(f*Conj(f))
p=p/length(p) # 此处好像是定义中有的

op<-par(mfrow=c(4,1))
# 如下参数即为无任何处理的周期图. (绘图过程中有处理, 故图看起来有点不同)
spec.pgram(co2,kernel=NULL,taper=0,fast=F,demean=F,det=F)
plot(p[2:(length(p)/2+1)],t='l')
plot(log10(p[2:468/2+1]),t='l') # 取对数后单位变为分贝
plot(log10(p[2:468/2+1]),t='l')
par<-op

# 查看数据
> x=c(15 , -2,  12,  20,  -5 ,  0 , -8 , -4 , -8, -22)
> spectrum(x,taper=0,fast=F,demean=F,det=F,plot=F)$spec[1:10]
 [1] 307.10684  35.00167  80.59316 280.29833  19.60000       NA       NA
 [8]       NA       NA       NA
> Re(fft(x)*Conj(fft(x)))/length(x)
```

```
 [1]   0.40000 307.10684  35.00167  80.59316 280.29833  19.60000 280.29833
 [8]  80.59316  35.00167 307.10684
```

过度平滑不好

```
f=fft(co2)
p=Re(f*Conj(f))
p=p/length(p) # 此处好像是定义中有的

f=filter(co2,f=c(.5,.5))
f=na.exclude(f)
f=fft(f)
p1=Re(f*Conj(f))/length(f)

f=filter(co2,f=rep(1/3,3))
f=na.exclude(f)
f=fft(f)
p2=Re(f*Conj(f))/length(f)

f=filter(co2,f=rep(1/4,4))
f=na.exclude(f)
f=fft(f)
p3=Re(f*Conj(f))/length(f)

op<-par(mfrow=c(4,1))
plot(log10(p),t='l')
plot(log10(p1),t='l')
plot(log10(p2),t='l')
plot(log10(p3),t='l')
par<-op
```

Chapter 82

sound

82.1 载入声音文件并查看信息

假设有一个声音文件名为 "frog.wav"

```
> library(sound)
> x <- loadSample("frog.wav")
> typeof(x)
[1] "list"
# .wav 对象属于 Sample 类
> class(x)
[1] "Sample"
> names(x)
[1] "sound" "rate"  "bits"

# 查看信息
> print(x)
type     : mono
rate     : 22050 samples / second
quality  : 16 bits / sample
length   : 73611 samples
R memory  : 294444 bytes
HD memory : 147266 bytes
duration  : 3.338 seconds
```

```
# 获取声音数据
> s=sound(x) # 等价于 x$sound
> dim(x$sound)
[1]     1 73611
# 时间长度
> duration(x)
[1] 3.338367
# 采样位数
> bits(x)
[1] 16
# 采样率
> rate(x)
[1] 22050
```

82.2 声谱,播放,频率图

```
# 绘出声谱
> plot(x)
# 播放声音
> play(x,command='mplayer')

# 绘制fft图
n <- length(x$sound)
n <- round(n/3)
y <- x$sound[ n:(n+2000) ]
n <- length(y)
op <- par(mfrow=c(3,1), mar=c(2,4,2,2)+.1)
plot(y, type='l')
plot(Mod(fft(y)[1: ceiling((length(y)+1)/2) ]), type='l')
```

82.3 修改声音

```
d=matrix(0,nc=500,nr=2)
q=rnorm(500)
d[1,]=q
d[2,]=q
y=loadSample('a1.wav')
sound(y)<-d
saveSample(y,'a6.wav')
```

82.4 产生调频信号

函数用法

```
chirp( t, f0 = 0, t1 = 1, f1 = 100,
      form = c("linear", "quadratic", "logarithmic"),
      phase = 0
    )
```

- t: 时间向量. (array of times at which to evaluate the chirp signal)
- f0: t=0 的频率 (frequency at time t=0.)
- t1: 时间, 单位秒. (time, s.)
- f1: t=t1 的频率 (frequency at time t=t1.)
- form: 调频(频率变化)的形状. (shape of frequency sweep, one of "linear", "quadratic", or "logarithmic".) 定义为
  - 'linear' is: $f(t) = (f1 - f0) * (t/t1) + f0$
  - 'quadratic' is:$f(t) = (f1 - f0) * t/t1^2 + f0$
  - 'logarithmic' is:$f(t) = (f1 - f0)^{t/t1} + f0$
- phase: t=0 的相位. (phase shift at t=0.)

下面是在线例子

```
ch = chirp(seq(0, .6, len=5000))
plot(ch, type = "l")

# Shows a quadratic chirp of 400 Hz at t=0 and 100 Hz at t=10
# Time goes from -2 to 15 seconds.
# 时 间 为 -2, 15 s, t=0 频 率 为 400, t=10 为 100, 变 化 形 状 是 quadratic
specgram(chirp(seq(-2, 15, by=.001), 400, 10, 100, "quadratic"))

# Shows a logarithmic chirp of 200 Hz at t=0 and 500 Hz at t=2
# Time goes from 0 to 5 seconds at 8000 Hz.
specgram(chirp(seq(0, 5, by=1/8000), 200, 2, 500, "logarithmic"))
```

82.5 语图

spectgram() 函数绘制黑白图(灰度图). image() 绘制彩图(library(graphics))

```
library(signal)

# 下 面 使 用 signal 包 内 的 函 数 spectgram() 来 绘 制 语 图(spectrogram)
wav <- loadSample("frog.wav") # library(sound)
Fs = wav$rate
step = trunc(5*Fs/1000);    # one spectral slice every 5 ms
window = trunc(40*Fs/1000); # 40 ms data window
fftn = 2^nextpow2(window); # next highest power of 2
spg = specgram(wav$sound, fftn, Fs, window, window-step)
S = abs(spg$S[2:(fftn*4000/Fs),]) # magnitude in range 0<f<=4000 Hz.
S = S/max(S)        # normalize magnitude so that max is 0 dB.
S[S < 10^(-40/10)] = 10^(-40/10)   # clip below -40 dB.
S[S > 10^(-3/10)] = 10^(-3/10)    # clip above -3 dB.
image(t(20*log10(S)), axes = FALSE) #, col = gray(0:255 / 255))
```

Chapter 83

小波

83.1 推荐

《小波与傅里叶分析基础》[23], 入门极佳, 工科足够.

R 的包(下面的介绍来自 CRAN Task View: Time Series Analysis):

- wavelets: 包含计算小波滤波, 小波变换, 多尺度分析的内容.
- wmtsa: 基于 Percival and Walden (2000) 的时间序列分析的小波方法
- waveslim: time series (1D), image (2D) and array (3D) analysis. 实现了众多方法(包括 wmtsa).[1]
- brainwaver: 依赖于 waveslim[2]

[1]原包的介绍如下: Basic wavelet routines for time series (1D), image (2D) and array (3D) analysis. The code provided here is based on wavelet methodology developed in Percival and Walden (2000); Gencay, Selcuk and Whitcher (2001); the dual-tree complex wavelet transform (CWT) from Kingsbury (1999, 2001) as implemented by Selesnick; and Hilbert wavelet pairs (Selesnick 2001, 2002). All figures in chapters 4-7 of GSW (2001) are reproducible using this package and R code available at the book website(s) below.

[2]原包的介绍如下: This package computes the correlation matrix for

- wavethresh: 1d, 2d 小波分析[3]
- rwt: 依赖于 matlab

这里使用 waveslim 包

83.2 介绍

小波的出现尽管可以追溯到几十年前, 但是只是在最近的二十多年才成为信号分析流行的工具. 一定程度上, 这应当归功于 Ingrid Daubechies 女士[4]在构造紧支撑正交小波方面的杰出工作. 大部分的小波文章和参考资料需要复杂的数学背景(研究生程度的实分析课程). 傅里叶变换的一个缺点是, 它的构造块是无始无终的周期性正弦和余弦波, 适合压缩(滤除, 分析)那些具有近似周期性的波动信号. 而对于有显著局部特征的信号就无能为力了.

而小波不同于正弦波和余弦波, 它仅仅在有限的一段非零. 小波可以平移和伸缩, 然后将给定的信号展开成小波的伸缩和平移之和, 然后把欲舍弃的系数进行适当处理或直接丢弃. 这就是小波变换.

小波有很多种. 它们(包括傅里叶变换的正弦波和余弦波)应该具有一些基本性质, 其中一个就是正交性, 包括平移和伸缩后. 而正弦波和余弦波具备这样的性质, 导致了求解傅里叶系数的简单公式和高效算法(FFT). ([23] 前言部分)

---

each scale of a wavelet decomposition, namely the one performed by the R package waveslim (Whitcher, 2000). An hypothesis test is applied to each entry of one matrix in order to construct an adjacency matrix of a graph. The graph obtained is finally analysed using the small-world theory (Watts and Strogatz, 1998) and using the computation of efficiency (Latora, 2001), tested using simulated attacks. The brainwaver project is complementary to the camba project for brain-data preprocessing. A collection of scripts (with a makefile) is avalaible to download along with the brainwaver package, see information on the webpage mentioned below.

[3]原包的介绍如下: Software to perform 1-d and 2-d wavelet statistics and transforms

[4]Ingrid Daubechies 女士现在为普林斯顿大学数学系教授

83.3 小波的类型

参考 http://en.wikipedia.org/wiki/Wavelet

83.3.1 Discrete wavelets

- Beylkin (18)
- BNC wavelets
- Coiflet (6, 12, 18, 24, 30)
- Cohen-Daubechies-Feauveau wavelet (Sometimes referred to as CDF N/P or Daubechies biorthogonal wavelets)
- Daubechies wavelet (2, 4, 6, 8, 10, 12, 14, 16, 18, 20)
- Binomial-QMF
- Haar wavelet
- Mathieu wavelet
- Legendre wavelet
- Villasenor wavelet
- Symlet

83.3.2 Continuous wavelets

Real valued

- Beta wavelet
- Hermitian wavelet
- Hermitian hat wavelet
- Mexican hat wavelet

- Shannon wavelet

Complex valued

- Complex mexican hat wavelet
- Morlet wavelet
- Shannon wavelet
- Modified Morlet wavelet

83.3.3 TOBEDEL: wt.filter()支持的小波

wavelets 包返回的值是 S4 对象. wt.filter() 函数产生各种小波.

```
d    Daubechies 2,4,6,8,10,12,14,16,18,20.

la    Least Asymetric 8,10,12,14,16,18,20.

bl    Best Localized 14,18,20.

c    Coiflet 6,12,18,24,30.
```

83.3.4 wave.filter()函数支持的小波

这里我们使用 waveslim 包. waveslim 的文档并没有给出可用的下面小波的全称. 最常用的是 Daubechies wavelet

```
haar
bl14 # Best Localized 小波(or Beylkin 小波??)
bl20
bs3.1
```

```
d16
d4 # Daubechies wavelet
d6
d8
fk14
fk22
fk4
fk6
fk8
la16 # Least Asymetric 小波
la20
la8
mb16
mb24
mb4
mb8
w4
```

83.4 例子

waveslim包dwt() 函数的用法

```
dwt(x, wf="la8", n.levels=4, boundary="periodic")
    dwt.nondyadic(x)

# 返回值
     d?: Wavelet coefficient vectors. 小波系数

     s?: Scaling coefficient vector. 尺度系数

 wavelet: Name of the wavelet filter used.

boundary: How the boundaries were handled.
```

构造一个零之前为高频, 零之后为低频的信号([46] 15.6.1)

```
N <- 1024
k <- 6
x <- ( (1:N) - N/2 ) * 2 * pi * k / N
y <- ifelse( x>0, sin(x), sin(3*x) )
plot(y~x, type='l')

z<-y+rnorm(N)/10

library(waveslim)
# 图的上面是低频尺度系数, 下面四个是高频小波系数. 频率从高到低
d<-dwt(z)
op<-par(mfrow=c(5,1))
plot(d$s4,t='l',ylab='s4')
plot(d$d1,t='l',ylab='d1')
plot(d$d2,t='l',ylab='d2')
plot(d$d3,t='l',ylab='d3')
plot(d$d4,t='l',ylab='d4')
par(op)

# 过滤掉高频成分并重构信号
d$d1<-rep(0,length(d$d1))
d$d2<-rep(0,length(d$d2))
d$d3<-rep(0,length(d$d3))
id<-idwt(d)
#
op<-par(mfrow=c(2,1))
plot(z~x,t='l')
plot(id~x,t='l')
par(op)

# 过滤掉低频成分并重构信号
d$s4<-rep(0,length(d$s4))
id<-idwt(d)

op<-par(mfrow=c(2,1))
plot(z~x,t='l')
plot(id~x,t='l')
par(op)
```

Part X

数据挖掘—机器学习

Chapter 84

R包介绍与参考文献

84.1 参考文献

(新西兰) Ian H. Witten, Eibe Frank 著. 董琳 邱泉 于晓峰 吴韶群 孙立骏 译 数据挖掘–实用机器学习技术*(*第二版*)*, 机械工业出版社. 2006.2. [18] 此文献是软件 weka 的开发者写的. 后半部分介绍 weka 的使用. weka 有一个很好的图形界面. 使用 java 实现. RWeka 是 R 对此软件的接口.

参考网页 http://cran.r-project.org/web/views/MachineLearning.html

不完全. 其它还有 Optimization using Genetic Algorithms Package, Association Rules 等.

84.2 机器学习包

84.2.1 Support Vector Machines and Kernel Methods

svmpath: 计算2类别svm分类器的全部正则化路径(regularization path)

kernlab: 弹性的核方法框架, 分类, 新颖性检测等. 包括 SVM, RVM, 谱方法, 核PCA, QP算子等其它的学习算法.

e1071: svm() 函数实现了 libsvm 接口. tune() 函数可以调节参数.

klaR: SVMlight 方法接口(只针对 one-against-all classification)

rdetools: 相对维过程, 并且提供模型选择和预测的功能.

84.2.2 Bayesian Methods

BayesTree: Bayesian 辅助回归树(Bayesian Additive Regression Trees, BART), 最后的模型定义为很多弱的学习器的求和.(并非整合方法, not unlike ensemble methods)

tgp: Bayesian 非平稳, 半参数非线性回归. 和基于高斯过程的 Bayesian CART(分类和回归树)树线性方法(treed liner model)

BPHO: Bayesian logistic 回归模型, 考虑高阶相互作用.

predbayescor: Bayesian 的朴素贝叶斯模型, 针对二分分类器, 同时做偏置修正.

84.2.3 Recursive Partitioning

rpart, tree: 树模型回归和分类. 实现了分类与回归树 CART book 的方法. (Classification and Regression Trees, CART)

RWeka: weka 软件的函数界面. 包括 J4.8-variant of C4.5 and M5.

party: 两个递归分类器算法, 对变量选择无偏, 基于统计的停止规则. 主要函数为 ctree(), 条件推断树, 内含树结构的回归模型, 定义良好的条件推断过程理论. 非参数回归可以应用于各种类型的回归问题, 包括名词性数据, 有序数据, 数值数据, 相应是多元数据, 任意的协方差也可以. cforest() 提供一个 Breiman's 随机森林. mob()函数实现一个基于参数模型(线性

模型, 广义线性模型, 生存回归等)的递归分类算法, 为一个模型树. 对分隔的选择执行参数不稳定检验(employing parameter instability tests for split selection). 还有扩展的函数用于树和分类结果的可视化.

84.2.4 randomForest

randomForest: 随机森林分类器.

varSelRF: 从随机森林中选择变量. 主要用于高维数据(微阵列数据, 其它基因组和蛋白质组数据). . 基于后向剔除方法(适合非冗余的小的变量集合)和 importance spectrum 方法(重要谱方法, 类似scree plots, 适合选择大的可能高度相关的变量). 可以使用 rpvm 代替 Rmpi, 但是只对 Rmpi 测试过.

ipred: 回归, 分类, 生存分析

party: 实现了一个随机森林, 基于规则的分类树. cforest() 提供一个 Breiman's 随机森林.

randomSurvivalForest: 针对检查过的数据(代表性好) censored data

84.2.5 Elements of Statistical Learning

ElemStatLearn: 数据, 函数, 例子, 《The Elements of Statistical Learning, Data Mining, Inference, and Prediction》 by Trevor Hastie, Robert Tibshirani and Jerome Friedman

Chapter 85

概念

参考文献 [18] 第2章

85.1 四种完全不同的学习方式

数据挖掘领域存在四种完全不同的学习方式.

85.1.1 分类学习(classification learning)

用一个已经分类的样本集来表示学习方案, 并希望从此样本集中学习对未来样本进行分类的方法.

有时候称为有指导(supervised)的学习. 即每个训练样本都有一个明确的结论, 称为样本的类.

85.1.2 关联学习(association learning)

寻找任何特性之间的关联, 不仅仅是为了预测某个特定的类值.

85.1.3 聚类(clustering)

寻找能够组合在一起的样本, 并依次分组.

85.1.4 数值预测(numeric prediction)

预测出的不是离散类而是一个数值量.

85.2 样本

机器学习方案的输入是一个实例集, 这些实例由机器学习方案进行分类, 关联, 聚类. 它们被称为样本, 更专业的术语称为实例.

85.3 闭合世界假定

闭合世界假定(closed world assumption): 只明确指出肯定实例, 并假设剩余的都是否定的实例, 这种做法称为闭合世界假定.

例如, 在一个大的家庭关系中, 有6对有姐妹关系, 其中大部分成员对并不存在姐妹关系.

85.4 递归技术

很多问题的数据是无限的. 例如, 虽然人类的家族树或许有限, 但是很大. 在可能的实例数量无限的情况下, 计算机科学家通常使用递归的方法进行处理. 例如下面的关系

```
if A is parent of B
   then A is an ancestor of B
```

```
if A is parent of B
   and B is an ancestor of C
   then A is an ancestor of C
```

85.5 属性

每个固定的单一的实例是由一组固定的和预先定义的特征或属性值作为输入提供给机器学习的.

如果不同的实例有不同的属性会怎么样? 例如, 假设实例是交通工具, 车轮的数量可以是许多车辆的特征, 但是不能用于船只; 桅杆的根数是船只的特征, 但是不适用于车辆. 一种标准的处理方法是把每个可能的值作为一个属性, 并使用一个"无关值"的标记值促对于一个特定的实例哪个属性不适用.

许多数据挖掘系统只采用两种度量标准: 名词性值和有序值.

85.5.1 数值性值

85.5.2 名词性值(nominal)

有时候称为范畴的, 可枚举的或离散的属性. 它的一个特例是二分值, 有无, true false, yes no 等.

85.5.3 有序值

85.5.4 区间值

85.5.5 比率值

85.6 VC维理论

参考维基 http://en.wikipedia.org/wiki/VC_dimension

统计学习理论中, 有时候一些计算理论中, VC dimension (for Vapnik-Chervonenkis dimension) 是对统计分类算法的容量的度量. 定义为算法可以打散(shatter)的最大点数的势(cardinality). 它是 Vapnik-Chervonenkis theory [1]的核心概念, 最初由 Vladimir Vapnik and Alexey Chervonenkis 提出.

不太正式的讲, 分类模型的容量与它的复杂度有关. 例如, 考虑高维多项式的阈值: 若计算结果大于0, 此点分类为"+", 否则分类"-". 高维的多项式可以摆动的很厉害的, 故可以对很多的训练点拟合的很好. 但是可以想像它对其它的点分类错误很多. 这样的多项式容量很高. 一个简单的可能对训练样本分类不是很好, 因为它的容量很小.

85.6.1 Shattering(打散)

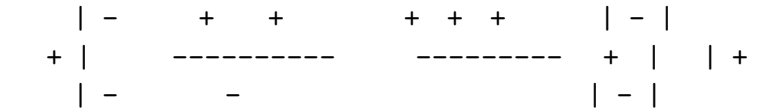
一个分类模型f, 参数向量为θ, 待分类的点$x = x_1, \cdots, x_n$, 如果存在θ使得f对x的分类没有错误, 那么我们说f可以打散(shatter)这些点.

f的VC维h'是样本点x的可以被f打散的最大的势h. 例如, 考虑一个感知机的模型f为直线, 则此直线可以分类正负点. 存在3个点的集合可以被f打散(实际上, 任何非共线性的3点集合都可以被打散). 但是, 任何4点的集合都不能被打散. 故此分类器的VC维是3. 重要的是记住可以随意改变点的位置, 但是不能改

[1]见维基 http://en.wikipedia.org/wiki/Vapnik-Chervonenkis_theory

变它的标签(正/负). 注意到, 3个点只有$2^3 = 8$种标签排列(如果标签可以变换)

下面是直线将3个点分类的示意图, 最后一个是4个点的.



85.6.2 用途

VC维可以预测分类器在测试集上的误差上界. 独立同分布(i.i.d.)测试数据的分类模型的误差为

$$Training\ error + \sqrt{\frac{h(log(2N/h)+1) - log(\eta/4)}{N}}$$

概率为$1-\eta$, h为分类器f的VC维, N为测试样本的个数(容量), 理论上, 此公式在$h < N$时有效. 类似的一个复杂的上界可以由Rademacher complexity给出, 但是有时候Rademacher complexity可以提供更加深入的计算, 例如有核的统计方法.

85.6.3 vc维理论的其它资料

所谓VC维是对函数类的一种度量，可以简单的理解为问题的复杂程度，VC维越高，一个问题就越复杂。正是因为SVM关注的是VC维，后面我们可以看到，SVM解决问题的时候，和样本的维数是无关的（甚至样本是上万维的都可以，这使得SVM很适合用来解决文本分类的问题，当然，有这样的能力也因为引入了核函数）。

VC维被认为是数学和计算机科学中非常重要的定量化概念，它可用来刻画分类系统的性能. 模式识别中VC维的直观定义是:对一个指示函数集，如果存在h个样本能够被函数集中的函数按所有可能的2h种形式分开，则称函数集能够把h个

样本打散，函数集的VC维就是它能打散的最大样本数目h，若对任意数目的样本都有函数能将它们打散.则函数集的VC维是无穷大。有界实函数的VC维可以通过用一定的阈值将它转化成指示函数来定义。VC维反映了函数集的学习能力，VC维越大则学习机器越复杂，所以VC维又是学习机器复杂程度的一种衡量。换一个角度来理解，如果用函数类f(z,a)代表一个学习机，a 确定后就确定了一个判别函数了EF，而VC维为该学习机能学习的可以由其分类函数正确给出的所有可能二值标识的最大训练样本数。

Chapter 86

算法: 基本方法

参考文献 [18] 第4章

最具有指导意义的一句话: 简单的方法通常能够很好的工作. (Ockham's Razor, 奥卡姆剃刀原理)实际分析中, 建议采用简单优先的原则.

复杂的方法很多时候可能完全丢失其它不同结构的规律性, 结果得到结构复杂的, 难以理解的一种分类结构, 而不是简单的, 优美的, 能够立刻被理解的另一种结构形式.

86.1 1规则(1-rule)

86.1.1 介绍

1规则简称1R. 产生一层决策树, 只在某个特定的属性上测试. 1R是一个简单, 廉价的方法, 但是查出能够得到非常好的规则来描述数据结构. 其结果经常能达到高的令人吃惊的正确率. 也许, 真实世界中的数据集往往结构相当基本. 仅仅用一个属性就可以准确判断出一个实例的类别.

在任何分析中应该首先采用最简单的方法.

86.1.2 残缺值

1R处理残缺值和数值属性的方法既简单又高效. 对于残缺值, 把残缺值看作另一个属性值. 例如, 天气有 sunny, overcast, rainy, 那么残缺值为第四个值.

86.1.3 数值属性

可以采用一个简单的离散方法把数值属性转换为名词属性. 首先按照数值排序, 例如

```
> x=sample(64:85,14)
> x # 温度
 [1] 66 77 68 65 71 76 70 81 78 85 67 64 72 82
> y=sample(c("y","n"),14,rep=T)
> y # 天气
 [1] "y" "n" "n" "n" "y" "n" "y" "n" "n" "y" "y" "n" "n" "y"

# 按照数值属性x排序
> o=order(x)
> y[o]
 [1] "n" "n" "y" "y" "n" "y" "y" "n" "n" "n" "n" "n" "y" "y"
> x[o]
 [1] 64 65 66 67 68 70 71 72 76 77 78 81 82 85
```

离散通过在这个序列上设置断点来达到分隔. 一个可行的方法是在类值发生变化的地方放置断点.

```
> y[o]
 [1] "n" "n"| "y" "y"| "n" |"y" "y"| "n" "n" "n" "n" "n" |"y" "y"
> x[o]
 [1] 64 65| 66 67 |68| 70 71 |72 76 77 78 81 |82 85
```

断点设置在中间位置, 即 $65.5, 67.5, 68.5, 71.5, 81.5$

离散存在一个严重的问题, 有可能形成大量的类别范畴. 1R算法倾向于选择能够分裂为很多范畴的属性. 一个极端的例子是, 某个实例拥有一个不同的值. 它产生的误差是0, 但是在训练集以外的数据上不会产生正确的预测. 这被称为过度拟合(overfitting)

对于1R算法, 当一个属性存在大量的可能值时 ,过度拟合很可能要发生. 所以离散的时候, 需要采用一个规则, 例如设置最小的样本数量为3

86.2 统计建模–贝叶斯方法

86.2.1 朴素贝叶斯方法(Naive Bayes)

通过4个条件预测某人是否出去玩. 收集到14个样本. 我们想根据这些数据预测某种天气情况下此人是否出去玩.

```
outlook    temperature    humidity    windy      play
-----------------------------------------------------------
sunny,     85,            85,         FALSE,     no
sunny,     80,            90,         TRUE,      no
overcast,  83,            86,         FALSE,     yes
rainy,     70,            96,         FALSE,     yes
rainy,     68,            80,         FALSE,     yes
rainy,     65,            70,         TRUE,      no
overcast,  64,            65,         TRUE,      yes
sunny,     72,            95,         FALSE,     no
sunny,     69,            70,         FALSE,     yes
rainy,     75,            80,         FALSE,     yes
sunny,     75,            70,         TRUE,      yes
overcast,  72,            90,         TRUE,      yes
overcast,  81,            75,         FALSE,     yes
rainy,     71,            91,         TRUE,      no
```

为方便将数据离散化, 总结为概率如下

E	阴	晴		温	度		湿	度		刮	风	玩	玩
H	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	2	3	hot	2	2	high	3	4	False	6	2	9	5
overcast	4	0	mild	4	2	normal	6	1	True	3	3		
rainy	3	2	cool	3	1								
Sum	9	5		9	5		9	5		9	5	14	14

新的一天的情况如下, 判断某人是否出去玩.

```
条件            值
---------------------------
outlook:       sunny
temperature:    cool
humidity:      high
windy:         true
```

根据新的一天的情况(E)计算其玩(H=yes/no)的概率. 如果$P(yes|E) > P(no|E)$, 我们判断某人会出去玩. 否则判断某人不会出去玩.

贝叶斯法则指出,

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$

其中$P(E|H)$为关于E的似然概率(或条件概率), $P(E)$为先验概率, $P(H|E)$为后验概率.

下面我们计算似然概率. 假设条件的重要程度是一样的, 彼此独立的, 那么在新的一天的条件下

```
yes的似然=2/9*3/9*3/9*3/9*9/14=0.0053
```

其中, 前4项是4个条件下玩的概率, 最后是玩的先验概率$(9/(9+5) = 9/14)$. 同样的,

no的似然=3/5*1/5*4/5*3/5*5/14=0.0206

对于这个新的一天, 玩是no的概率为yes的概率的4倍. 通过正常化(归一化)将它们的概率和为1

yes的概率=0.0053/(0.0053+0.0206)=20.5%
no的概率=0.0206/(0.0053+0.0206)=79.5%

对应贝叶斯法则, 新的一天的条件E下完出现yes的概率为

$$P(yes|E) = \frac{2/9 * 3/9 * 3/9 * 3/9 * 9/14}{P(E)}$$

分母将在正常化(归一化)过程中消失, 因此不需要实际计算出来.

这种方法称为朴素贝叶斯, 因为它基于贝叶斯规则并朴素的假设各条件之间独立. 只有条件独立的时候相乘才是有效的–概率的乘法法则. 尽管独立性的假设与现实不太一致, 但是在实际数据集上测试的时候它工作的很好. 特别是和属性选择器一起去除掉一些冗余属性使得属性之间的独立性提高之后.

86.2.2 概率为0的问题–拉普拉斯估计器

如果某个属性并不包含所有的分类值, 即在某个类别上的出现的次数为0, 朴素贝叶斯方法会出错. 假设, 天气中sunny对应的玩的次数为0, 导致其概率$P(outlook = sunny|yes) = 0$. 其它概率与这个概率相乘, 总会得到0. 所以概率0此时掌握了否决权. 然而可以对频率做一些小的调整, 就可以很容易的弥补这个缺陷. 我们可以在每个分子上加1, 并在分母上加3进行补偿, 这就保证概率为0时会得到一个很小的但是不是0的概率. 在每个结果上加1的方法是一个标准的技术, 称为拉普拉斯估计器(laplace estimator).

尽管拉普拉斯估计器可以很好的工作, 但是没有特别的理由一定要加1. 取而代之, 可以使用一个很小的量μ,

$$\frac{2+\mu/3}{9+\mu}, \quad \frac{4+\mu/3}{9+\mu}, \quad \frac{3+\mu/3}{9+\mu}$$

当$\mu = 3$, 就是拉普拉斯估计器.

86.2.3 关于先验概率

对于拉普拉斯估计器的改进, 进一步, 可以使用不同的μ来代表先验概率的重要性. 这里使用另外一个量p_i来代替, 变为

$$\frac{2+\mu p_1/3}{9+\mu}, \quad \frac{4+\mu p_2/3}{9+\mu}, \quad \frac{3+\mu p_3/3}{9+\mu}$$

其中$p_1 + p_2 + p_3 = 1$

现在它是一个完整的贝叶斯公式. 先验概率已经分配到每一个属性的值上. 但是缺点是很多时候并不清除先验概率权重p的分布形式. 在实际中, 只要训练实例数量合适, 使用不同的先验概率几乎没有差别.

人们通常使用拉普拉斯估计器, 将计数的结果初始化为1(统一加1).

86.2.4 残缺值

计算的时候只要忽略残缺值即可. 即在计数的时候, 如果一个属性值残缺, 就不包含在频率的计算内. 所以概率的计算取决于真正出现属性值的实例的个数, 而不是实例的总数.

似然值会增大, 但是归一化后这个问题变不复存在.

86.2.5 数值属性

处理数值属性的时候, 通常假设为正态分布. 下面是天气情况的数值统计结果

通过4个条件预测某人是否出去玩. 收集到14个样本. 我们想根据这些数据预测某种天气情况下此人是否出去玩.

```
outlook    temperature    humidity    windy      play
-------------------------------------------------------
sunny,     85,            85,         FALSE,     no
sunny,     80,            90,         TRUE,      no
overcast,  83,            86,         FALSE,     yes
rainy,     70,            96,         FALSE,     yes
rainy,     68,            80,         FALSE,     yes
rainy,     65,            70,         TRUE,      no
overcast,  64,            65,         TRUE,      yes
sunny,     72,            95,         FALSE,     no
sunny,     69,            70,         FALSE,     yes
rainy,     75,            80,         FALSE,     yes
sunny,     75,            70,         TRUE,      yes
overcast,  72,            90,         TRUE,      yes
overcast,  81,            75,         FALSE,     yes
rainy,     71,            91,         TRUE,      no
```

其中温度和湿度为数值属性, 统计结果列在下面

```
温度={83,70,68,64,69,75,75,72,81}     玩=yes
   均值: 73    标准差: 6.2
温度={85,80,65,72,71}                 玩=no
   均值: 74.6  标准差: 7.9

湿度={86,96,...,90,75}                玩=yes
   均值: 79.1  标准差: 10.2
湿度={85,90,70,95,91}                 玩=no
   均值: 86.2  标准差: 9.7
```

将$\mu = 73, \sigma = 6.2$代入正态分布公式, 当观测到玩=yes的时候, 温度为66的概率为

$$P(temperature = 66|yes) = dnorm(66, mean = 73, sd = 6.2) = 0.03401871$$

新的一天的情况如下, 判断某人是否出去玩.

```
条件            值
----------------------------
outlook:        sunny
temperature:     66
humidity:       90
windy:          true
```

使用上面的概率得到[1]

```
yes的似然=2/9*0.0340*0.0221*3/9*9/14=0.000036
 no的似然=3/5*0.0279*0.0381*3/5*5/14=0.000108
```

归一化后产生的概率为

```
yes的概率=0.000036/(0.000036+0.000108)=25.0%
 no的概率=0.000108/(0.000036+0.000108)=75.0%
```

与前面离散化的结果非常接近.

实际使用的时候, 常常取似然的对数, 以避免过小的值使计算机下溢.

[1]参考文献[18]Page 63, no的似然第二项为0.0221, 有笔误, 应该为0.0279, 最后的结果未修改, 便于参照

86.2.6 用于文档分类的贝叶斯模型–多项朴素贝叶斯

机器学习的一个重要领域是文档分类. 一个文档就是一个实例. 实例的类就是文档的主题类别. 例如, 文档是新闻, 它的类可能是国内新闻, 海外新闻, 财经新闻, 体育新闻等. 文档的特性是由出现在文档内的单词描述的. 一种机器学习方法是使用布尔值表示每个单词的出现与否. 文档分类方面朴素贝叶斯方法是深受欢迎的技术, 因为它的处理速度快而且正确率高.

然而朴素贝叶斯方法忽略了单词出现的次数. 而这些信息对于文档分类有潜在的重要价值. 取而代之, 一个文档可以看作一袋单词, 袋内的单词可以重复出现, 即不考虑单词的顺序, 但是考虑单词出现的次数. 采用一个修改过的朴素贝叶斯便可以利用单词的频率. 这个修改过的朴素贝叶斯有时候称为多项朴素贝叶斯.

假设$n_1, \cdots, n_k$是单词i在文档中出现的次数. $P_1, \cdots, P_k$是在所有H类文档中得到的单词的概率, 从训练文档中统计每个单词出现的频率估计得到的. 假设概率与单词的上下文以及单词在文档中的位置无关. 这些假设产生了一个文档概率的多项式分布(multinormial distribution). 在此分布下, 对于给定的类别H, 文档E(已经转换为一袋单词, 不考虑次序, 即位置无关)的概率(即贝叶斯公式中的条件概率)为

$$P(E|H) \approx N! * \prod_{i=1}^{k} \frac{P_i^{n_i}}{n_i!}$$

$N!$为文档中单词的数量. 使用阶乘是因为单词出现的次序并不重要. 实际上应该还有一项, 类别H产生与E长度相等的文档的概率,(这是为什么使用≈而不是=), 但是通常假设这个概率相等, 因此忽略这一项.

假设文档单词表只有两个单词: 黄(yellow,y)和蓝(blue,b). 某个文档类别H(或许可以称为黄绿文档)存在

$$P(y|H) = 75\% \qquad P(b|H) = 25\%$$

如果E是新文档(b,y,b), 长度3个单词. H下它出现的概率为

$$p(E|H) = P((b, y, b)|H) = 3! * 0.25^2/2! * 0.75^1/1! = 9/64 = 0.14$$

假如另一个类别H',我们称为蓝绿文档, 拥有的概率

$$P(y|H) = 10\% \qquad P(b|H) = 90\%$$

此模型产生新文档E的概率为

$$p(E|H') = P((b, y, b)|H') = 3! * 0.9^2/2! * 0.1^1/1! = 9/64 = 0.24$$

如果类别只有这两个H和H', 是否意味着新文档E是蓝绿文档呢? 不一定. 这里我们还没有考虑它们的先验概率. 例如, 如果已知蓝绿文档出现的概率是黄绿文档出现概率的1/2, 即$P(H) = 2/3, P(H') = 1/3$, 那么其贝叶斯估计的后验概率为

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)} = 0.14 * 2/3 = 0.093$$

$$P(H'|E) = \frac{P(E|H')P(H')}{P(E)} = 0.24 * 1/3 = 0.08$$

由于产生长度相等的文档的概率相等, $P(E)$可以忽略. (即使长度不等, 前面假设其出现的概率也相等)

考虑了先验概率后, 结果倾向于新文档E是黄绿文档.

前面概率公式中的阶乘并不需要真正计算, 因为对每一个类别来讲都是一样的, 会在归一化过程中消失.

概率相乘产生很小的数的问题也可以通过取对数解决.

86.2.7 讨论

在许多问题上, 朴素贝叶斯方法可以与复杂的方法媲美, 甚至更加出色. 所以, 始终从简单的方法开始吧.

但是, 朴素贝叶斯方法在很多数据集上表现比较差. 原因很容易发现. 因为朴素贝叶斯认为属性之间是独立的. 所以一些

冗余数据会破坏机器学习的过程. 例如, 在天气的例子中再增加一个新的属性, 这个新属性的值与温度的值一样, 那么温度属性的影响力会增加, 温度的所有概率会平方. 进一步, 如果在数据集中加入10个一样的属性, 那么最终的决策将会依据温度做出.

属性之间的依赖性会降低朴素贝叶斯方法的预测能力. 但是这种情况可以通过仔细挑选属性子集的方法来避免.

对于数值属性, 正态分布是另一个问题. 许多属性并不服从正态分布. 如果我们知道一个数值属性服从的分布, 就可以使用这个分布来计算概率. 如果怀疑正态分布, 但又不知道是什么分布, 可以使用"核密度估计"(kernel density estimation)过程.

另一种可行的方法是把数值属性离散化.

86.3 TODO: 贝叶斯网络

86.3.1 分类的概率估计

朴素贝叶斯和logistic回归模型都产生概率估计来代替类别预测. 对于每个类别, 都是估计属于这个类的概率. 大部分其它分类器都可以强制产生这类信息. 例如, 通过计算叶节点每个类别的相对频率, 便可以从决策树中获得概率. 同样, 检验某条规则覆盖的实例数目, 也可以从决策列中获得概率.

概率估计常常比仅仅预测更加有用. 它们可以对所做的预测进行排名, 使期望成本达到最小.

把分类当做概率估计来完成还存在很大的争议. 所估计的是类别值的条件概率分布, 条件是其它属性值, 分类模型是用一种简洁易懂的形式来表达这个条件分布的. 由此看来, 朴素贝叶斯, logistic回归模型, 决策树等等只是用不同的方法表达这个条件分布. 当然, 它们的表达能力有所差别. 朴素贝叶斯和logistic回归模型只能表达简单的分布. 而决策树可以近似表达任意分布. 但决策树也有缺陷: 将训练集分隔为越来越小的数据集, 必然造成概率估计可靠性的下降. 并且还存在重复子

树的问题. 规则学习似乎能克服这些缺点, 但是其启发式方法尚缺乏理论依据.

有一种基于统计理论的方法, 采用图解的方式简洁易懂的表达概率分布的方法, 这个结构称为贝叶斯网络(Bayesian networks). 它是属于更广泛的概率图的一种. 隐马尔可夫模型(HMM)可以表示为无向的概率图.

在贝叶斯网络中, 每个节点代表一个属性, 节点间用有向线连接, 但是不能形成环, 称为有向无环图(directed acyclic graph)

86.3.2 TODO: 贝叶斯网络的一个简单例子

假设所有属性是名词性质的, 没有残缺值. 下面是天气数据的贝叶斯网络的简单实例.

图略, 参考文献[18] 6.7 page 181.

另一个贝叶斯网络图 参考文献[18] 6.7 page 183.

86.4 分治法: 创建决策树

86.4.1 使用信息增益选择属性

创建决策树的问题可以用递归的形式表示. 首先, 选择一个属性放置在根节点上, 每个可能的属性值产生一个分支. 然后在每个分支上各自选择另外的属性(属性可以重复选择), 递归重复这个过程, 仅使用到达这个分支的实例. 如果一个节点上所有实例类别相同, 停止分支.

唯一的问题是, 选择哪个属性进行分裂. 我们希望分支递归过程尽早停止. 如果能够测量节点的纯度, 就可以选择最纯分支的属性进行分裂.

我们使用的纯度的度量称为信息量. 例如, 天气例子中, 选择阴晴(outlook)作为根节点, 分支情况为

```
        阴晴(outlook)
-----------|------------
|        |          |
(sunny)  (overcast)  (rainy)
yes        yes        yes
yes        yes        yes
no         yes        yes
no         yes        no
no                    no
```

阴晴的3个属性值分裂为3个节点, yes 和 no 类的实例数量分别是$[2,3],[4,0],[3,2]$. 因此3个节点的信息量(此处使用单位bit, 以2为底数)分别是

```
> library(entropy)
> entropy(c(2,3),unit="log2") # sunny 节点的信息量, info([2,3])
[1] 0.9709506
> entropy(c(4,0),unit="log2") # overcast 节点的信息量
[1] 0
> entropy(c(3,2),unit="log2") # rainy 节点的信息量
[1] 0.9709506

# 实际上从原数据计算应该这样. sunny. 其它类似
> entropy(table(c('y','y','y','n','n')),unit="log2")
[1] 0.9709506
```

计算它们的平均信息量, 考虑每个分支的实例的数量, 采用加权方法:

```
info(sunny,overcast,rainy)
   =(5/14)*0.971+(4/14)*0+(5/14)*0.971
   =0.693
```

这个(加权)平均值代表了期望的信息量, 即使用指定属性对一个新实例的类别进行说明所必须的信息量.

处于根节点的样本为9个yes和5个no, 对应的信息量为

```
info(root)=info([9,5])=0.940
```

因此在阴晴属性上建立分支获得的信息增益(information gain)为

```
gain(outlook)=info([9,5])-info([2,3])-info([4,0])-info([3,2])
          =0.940-0.693=0.247 bit
```

随后的方法就很清楚了. 为每个属性计算信息增益, 选择获得最多信息量的属性进行分支.

86.4.2 改进

使用信息增益会出现问题. 假设一个属性拥有的可能值很多, 那么信息增益就会很大. 一个极端的例子是, 每一个属性对于一个实例, 譬如把数据集的行编号也作为一个属性, 那么这个属性的信息量为

```
inf([0,1])+...+info([0,1])=0
```

其信息增益就是根节点的信息量, 例如$info([9,5]) = 0.940$, 那么毫无疑问此节点会被选择为分支属性. 但是这样的分支对预测实例类别没有任何帮助, 也没有描述任何有关决策的结构. 而这两点正是机器学习的目标.

为了弥补这一缺陷, 一个称为增益率(gain ratio)的度量被广泛使用. 增益率考虑分支后产生的子节点的数量和规模, 不再考虑子节点内部的类别信息. 增益率为信息增益/此属性的信息值.

这样, 上面的极端情况的信息量为

```
info([1,1,...,1])=-1/14*log(1/14)=3.807
```

越高度分支的属性信息量越大. 此处增益率为0.940/3.807 = 0.247.

我们选择增益率高的属性作为分支属性, 可以防止无用属性在增益方面的优势.

不幸的是, 有时候增益率会修正过度, 造成选择某个属性仅仅因为此属性的内在信息量很小, 增益率就变得很大. 一个标准的方法是选择能够得到最大增益率的属性, 而且信息增益至少至少为所有信息增益的平均值.

86.4.3 讨论

前面描述的信息增益的方法, 从本质上看与称为ID3的方法一致. 使用增益率是多年来改进ID3 的方法之一. Quinlan 认为这是一个广泛适用于不同情况的稳定方案. 但是它牺牲了一些信息增益标准部分的清晰优雅的理论动机.

在称为C4.5的决策树系统中, 积累了一系列改进的ID3方法, 包括处理数值属性, 残缺值, 干扰呼数据及由树产生规则的方法.

86.5 覆盖算法: 建立规则

决策树是自上而下的. 如果有必要, 可以将之转换为一个分类规则集, 尽管转换过程并不简单.

另一个方法是依次取出一个类, 寻找一个方法/规则能够覆盖所有属于这个类的实例, 同时剔除不属于这个类的实例. 这种方法称为覆盖(covering)方法. 在某个阶段都要产生一个能够覆盖部分实例的规则. 所以这个方法决定了它将产生一个规则集, 而不是一个决策树.

规则和树在表达的清晰程度上是有差异的. 规则可以是对称的, 树则必须选择一个属性进行分裂, 这会导致树比规则集大很多. 另外一个不同是, 决策树将考虑所有类别的情况, 试图使分裂纯度最大化, 而规则一次只考虑一个类, 不考虑其它类发生的情况.

86.5.1 一个简单的覆盖算法

我们将要讨论的覆盖算法要选择一个属性-值配对(attribute-value pair)使得期望类别概率最大化.

下面使用隐形眼镜的例子. 共24个实例. 最后要预测的建议为3类隐形眼镜: none, soft, hard

	年龄	视力诊断	散光	泪流量	推荐镜片
	age	spectacle-prescrip	astigmatism	tear-prod-rate	contact-lenses
1	young	myope	no	reduced	none
2	young	myope	no	normal	soft
3	young	myope	yes	reduced	none
4	young	myope	yes	normal	hard
5	young	hypermetrope	no	reduced	none
6	young	hypermetrope	no	normal	soft
7	young	hypermetrope	yes	reduced	none
8	young	hypermetrope	yes	normal	hard
9	pre-presbyopic	myope	no	reduced	none
10	pre-presbyopic	myope	no	normal	soft
11	pre-presbyopic	myope	yes	reduced	none
12	pre-presbyopic	myope	yes	normal	hard
13	pre-presbyopic	hypermetrope	no	reduced	none
14	pre-presbyopic	hypermetrope	no	normal	soft
15	pre-presbyopic	hypermetrope	yes	reduced	none
16	pre-presbyopic	hypermetrope	yes	normal	none
17	presbyopic	myope	no	reduced	none
18	presbyopic	myope	no	normal	none
19	presbyopic	myope	yes	reduced	none
20	presbyopic	myope	yes	normal	hard

```
21    presbyopic      hypermetrope        no       reduced          none
22    presbyopic      hypermetrope        no        normal          soft
23    presbyopic      hypermetrope       yes       reduced          none
24    presbyopic      hypermetrope       yes        normal          none
```

我们依次生成3个规则, 分别覆盖3个类别(none, soft, hard). 首先对 hard 类寻找规则.

```
IF ? THEN recmendation = hard
```

对于未知条件 ? 存在 9 个选择

```
age = young                        2/8
age = pre-predbayescor              1/8
age = predbayescor                 1/8
spectacle-prescrip = myope          3/12
...
astigmatism = yes                  4/12
...
tear-production-rate = normal       4/12
```

右边的数值表示由此规则选择的实例集中, 正确的比例. 例如 $age = young$ 有8个实例, 其中2个建议hard镜片.

其中比例最大的为4/12, 但是有两个规则, 我们任意选择一个, 建立规则

```
IF astigmatism = yes THEN recmendation = hard
```

这个规则并不十分正确, 因此进一步修正

```
IF astigmatism = yes AND ? THEN recmendation = hard
```

未知条件有7个(去除astigmatism的两个). 有

```
age = young                        2/4
age = pre-predbayescor               1/4
age = predbayescor                  1/4
spectacle-prescrip = myope           3/6
spectacle-prescrip = hypermetrope    1/6
tear-production-rate = reduced       0/6
tear-production-rate = normal        4/6
```

最后一个胜出, 比例为4/6. 相应的规则为

```
IF astigmatism = yes
   AND tear-production-rate = normal
      THEN recmendation = hard
```

下一个可能的条件为

```
age = young                        2/2
age = pre-predbayescor               1/2
age = predbayescor                  1/2
spectacle-prescrip = myope           3/3
spectacle-prescrip = hypermetrope    1/3
```

需要在条件1(2/2)和条件4(3/3)中选择一个. 在同等条件下, 我们总是选择覆盖量大的. 因此选择第4个. 得到规则

```
IF astigmatism = yes
   AND tear-production-rate = normal
   AND spectacle-prescrip = hypermetrope
      THEN recmendation = hard
```

我们看一下规则覆盖的实例

```
            age spectacle-prescrip astigmatism tear-prod-rate contact-lenses
            ---------------------------------------------------------------
8         young      hypermetrope        yes        normal         hard
16 pre-presbyopic     hypermetrope        yes        normal         none
24    presbyopic      hypermetrope        yes        normal         none
```

但是所有的hard实例有4个.

因此, 去除这3个实例, 在剩下的实例集中重新开始寻找规则

```
IF ? THEN recmendation = hard
```

按照同样的程序, 最终发现$age = young$是第一个条件的最佳规则.

```
   age spectacle-prescrip astigmatism tear-prod-rate contact-lenses
   ---------------------------------------------------------------
1 young           myope         no       reduced         none
2 young           myope         no        normal         soft
3 young           myope        yes       reduced         none
4 young           myope        yes        normal         hard
5 young      hypermetrope        no        reduced         none
6 young      hypermetrope        no         normal         soft
7 young      hypermetrope       yes        reduced         none
8 young      hypermetrope       yes         normal         hard****
```

其中第8个在前面的规则已经被去除了, 所以$age = young$覆盖7个实例. 然后是$astigmatism = yes$, 然后是$tear - production - rate = normal$. 最终的规则为

```
IF age=young
   AND astigmatism = yes
   AND tear-production-rate = normal
      THEN recmendation = hard
```

这个规则实际上覆盖了数据集中的3个实例, 其中2个已经被第一个覆盖. 但是没有关系, 这两个规则给出的建议是一样的.

通过这两个规则, 所有的hard类都已经覆盖了. 下一步是使用同样的步骤生成soft的规则, 接着是none的规则. 记得在每个类别开始的时候初始化数据集到完整的状态.

以上的描述是PRISM法来创建规则. 只是建立正确或完美的规则. 它不断向每个规则增加条件, 直到正确率为100%. 诸如此类的算法可以描述为分治算法/割治算法(separate-and-conquer): 由于其建立规则后剔除所覆盖的实例, 所以算法的效率很高.

不同的类产生的规则没有顺序关系.

当不同的规则冲突时, 一个策略是在模糊实例上强制执行一个决策, 从所预测的类别中选择训练实例数量最多的类别, 或如果没有预测出的类别, 选择在总体上拥有最多实例的类别.

86.6 挖掘关联规则

关联规则(association rules)与分类规则近似, 可以采用同样的方法, 对每个可能出现在规则右边的表达式, 执行一个分治算法得到规则的归纳. 问题是任何属性的任何值都能够出现在右边的表达式, 而且一个单独的关联规则经常能够预测出不止一个属性的值. 必须对右边的每一种可能的属性组合, 用每种可能的属性值的组合, 执行一次规则的归纳过程. 这样会产生数量庞大的关联规则, 因此必须根据它们的覆盖量和正确率进行修剪, 但是这种方法常常不可行.

覆盖量常常称为支持(support), 正确率(accuracy), 置信度(confidence)

86.6.1 项集

项集就是属性取不同值的组合. 分为单项集, 二项集, 三项集, 等等. 例如, 天气数据中,

单项集 二项集 三项集 四项集
outlook=sunny outlook=sunny outlook=sunny outlook=sunny
temperature=hot temperature=hot temperature=hot
humidity=high humidity=high
play=no
outlook=overcast 所有2个属性值的组合 所有3个属性值的组合 所有4个属性值的组合

没有理由建立相同属性, 不同属性值的项集, 例如二项集: outlook=sunny, outlook=overcast. 这在实际中不可能出现, 因为它们之间是互斥的.

假如需要寻找最小覆盖量为2的项集, 那么覆盖量小于2的(0,1)的项集就要去除.

86.6.2 关联规则

一旦建立了符合最小覆盖量的项集, 就要把项集转换为拥有最小正确率的规则或规则集.

有一些项集会产生多个规则. 其它一些可能不产生任何规则.

例如, 一个覆盖量为4的三项集:

```
humidity=normal, windy=false, play=yes
```

产生7个潜在的规则

```
IF humidity=normal AND windy=false THEN play=yes       4/4
IF humidity=normal AND play=yes THEN windy=false       4/6
IF windy=false AND play=yes THEN humidity=normal       4/6

IF humidity=normal THEN windy=false AND play=yes       4/7
...
...

IF - THEN humidity=normal AND windy=false AND play=yes  4/12
```

86.6.3 有效的建立规则

详细考虑一个产生最小覆盖量和正确率的算法: 首先产生达到指定最小覆盖量的项集, 然后从每一个项集中找出能够达到指定最小正确率的规则.

TODO: 详细的过程

86.6.4 讨论

关联规则的算法复杂度很高, 通常还要在大的数据集中操作, 需要多次从磁盘读取数据. 因此效率很重要.

实际上, 建立关联规则计算量取决于指定的最小覆盖量, 正确率影响比较小.

86.7 线性模型

决策树或规则模型可以扩展到数值属性, 直接使用或离散化. 然而, 有一些方法可以自然的运用到数值属性.

86.7.1 数值预测: 线性回归

当结论或类是数值, 所有的属性也都是数值, 线性回归是一种自然考虑的方法.

线性回归方法: 略.

86.7.2 线性分类: logistic回归

线性回归可以方便的用于含数值属性的分类问题. 实际上, 任何回归技术都可以用来分类, 无论线性还是非线性. 技巧是对每个类执行一个回归, 属于该类的实例输出为1, 不属于该类的输出为0. 结果每个类得到该类的一个线性表达式. 对于给定的未知的实例, 计算每个线性表达式的结果, 选择其中最大的. 这种方法有时候称为多反馈线性回归(multiresponse linear regression).

但是这种方法存在两个缺陷: 1. 结果产生的不是概率, 因为结果可能落到0,1之外; 2. 最小平方回归假设误差是独立的, 且是标准差相等的正态分布. 结果为0,1显然违背此假设.

logistic回归不存在这种缺陷. logistic 回归将目标(函数对0,1的逼近值)转换, 并建立一个线性模型.

首先假设2类情况. 原式目标变量

$$p = Pr(1|a_1, \cdots, a_k)$$

这个无法用线性函数正确近似的变量转换为

$$\log \frac{p}{1-p}$$

结果的范围从0,1变到负无穷和正无穷. 这个变换称为对数变换.

转换后的变量由一个线性函数表示

$$\log \frac{p}{1-p} = \sum w_i a_i$$

解得

$$p = \frac{1}{1 + exp(\sum w_i a_i)}$$

略.

推广到多类情况, 一个可能的方法如前面提到的多反馈线性回归(multiresponse linear regression), 为每个类别形成一个logistic回归. 不幸的是, 得到的概率和不为1. 有必要将个体的logistic回归模型结合起来, 这将产生一个联合优化问题. 已经有一些方案处理这个问题.

86.7.3 成对分类

一个概念更简单, 并且非常通用的解决多类问题的方法是成对分类(pairwise classification). 为每一对类别建立一个分类器, 仅仅使用属于这两个类别的实例. 预测的时候看哪个类别能够得到最多的投票.

从分类误差来讲, 这种方法通常能产生正确的结论. 它也能够通过运用一个称为成对合并(pairwise coupling) 的方法产生概率估计. 这种方法可以校正从不同分类器产生的各自的概率估计.

如果有k个类, 成对分类共建立$k(k-1)/2$个分类器. 尽管看起来这个计算强度是不必要的, 但实际上, 如果类分布均衡, 那么成对分类法的处理速度至少和其它处理多类别问题的方法一样快. 原因是成对分类只考虑属于两个类别的实例. 这种方法的计算量与类别数量呈线性关系. 例如k个类, 每个类别对的实例数目为$2n/k$, 所有计算量为$k(k-1)/2 * 2n/k = (k-1)n$. 如果学习算法需要花费更多的时间, 例如与n^2正比, 那么成对分类的优势就更加明显.

线性明显和logistic模型处理的都是线性可分的实例, 其边界是在实例空间上的线性平面. 多反馈线性回归(multiresponse linear regression)也存在这个问题.

86.7.4 使用感知器的线性分类

logistic回归试图通过将数据集的概率最大化产生正确的概率估计, 进而产生正确分类. 但是, 如果模型的唯一目的是预测类的标签, 不需要进行概率估计. 另一个方法就是学习一个分类超平面, 能够将不同的类分开.

如果使用一个超平面可以将数据分开, 那么称数据是线性可分的. 一个非常简单的方法是称为感知器的方法. 感知器是神经网络的祖先, 实际上是一个没有隐层的神经网络.

感知器构造与学习: 略

86.7.5 使用winnow的线性分类

对于二值属性的数据集, 有一种方法称为winnow算法, 与感知器方法很类似, 二者的不同在于更新权值的策略不同. winnow算法是当出现错误的时候才更新权值向量. 它是错误驱动的(mistake driven)

略.

86.8 基于实例的学习

基于实例的学习中, 样本被完整的保存. 使用指定的距离函数计算未知实例与样本数据集中的实例的距离, 最靠近未知实例的样本实例所在的类就是未知实例所在的类. 剩下的唯一问题就是确定距离函数. 它并不十分困难.

距离函数: 略.

86.8.1 有效寻找最近邻-kD树与kD球树

由于每次预测都要计算与所有样本的距离, 因此速度比较慢.

一个有效的方法是使用树来表示样本数据集, 这样每次预测可以从线性时间缩短到对数时间. 其中一种合适的树是kD树.

kD树是一个二叉树, 用一个超平面将样本实例空间分隔, 然后递归的进行分裂. 所有的分隔平面与坐标轴平行或垂直. 称为kD树是因为所有的点是k维的, k是属性的个数.

如果数据平衡, kD树就基本是平衡的, 每个区域是一个接近正方形. 如果不均衡, 或连续在一个属性上面分隔, 那么很可能出现长条区域, 寻找其邻近区域的计算量就会变得很大.

新的实例假如训练集后需要更新kD树. 判断哪个叶子节点包含此新实例, 如果叶子的超矩形为空, 就把新实例放入. 否则废革超矩形. 分隔在最长的边上进行, 以保持正方形. 这种简单的方法并不能保证一系列点加入后树依然平衡, 也不能保证分隔后有良好的超矩形. 有时候重新构树是好办法, 例如, 在树的深度到达最佳深度的2倍的时候.

kD树的正方形并不是完美的形状. 一种替代方案是使用超球体, 称为kD球树(ball tree). 当然, 相邻的球体可能重叠, 这并不是一个问题, 因为kD树的最邻近算法不需要不重叠.

kD球树查询与更新: 略

86.8.2 讨论

最近邻法起源于几十年前. 50年代就分析了k最近邻方法. 当k和n达到无穷, 则误差概率达到最小.

数据库容易有干扰样本, 一个解决方法是使用k个最近邻, 然后投票得出测试实例的类别. 另一个方法是预处理, 仔细挑选样本.

如果实例数目很少, k个近邻的方法是危险的.

kD树是90年代开始使用, 尽管本身发展的很早. 当属性数量(维数)增加时, kD树效率就很低, 一般上限为10. 球树是最近发展的方法, 是更通用的结构的一个实例. 有时候称为测量树. 一些高级算法产生的球树可以处理上千维的空间.

另一个方法是压缩训练集到多个区域, 取代存储所有的样本实例. 一个简单的技术是, 记录在训练样本上的每个属性的值和每个类别的值的区域. 对于未知的实例, 找到样本实例各个属性值所在的区域, 选择与未知实例属性值区域正确对应数量达到最多的那个类作为这个实例的类别.

另一个更精细的技术是对每个属性建立多个区间, 统计样本数据集每个属性值区间的每个类的频率. 数值属性可以被离散化为区间. 用投票的方法对测试实例进行分类. 称为投票特征区间(voting feature interval).

这些是近似方法, 但是很快, 可以对大的数据集进行初步的分析.

86.9 聚类

不是预测, 而是将实例分成自然的组时, 就需要聚类即使.

86.9.1 基于距离的迭代聚类

86.9.2 快速距离计算

使用kD树或球树.

86.9.3 如何选择类别数目k?

一种方法是使用不同的k值, 选择其中最好的. 这需要用到机

器学习的评估方法.

Chapter 87

TODO: 可信度: 评估机器学习结果

参考文献 [18] 第5章

87.1 交叉验证

87.2 预测概率

87.2.1 二次损失函数

87.2.2 信息损失函数

87.3 计算成本

87.4 评估数值预测

Chapter 88

TODO: 转换: 处理输入和输出

参考文献 [18] 第7章

88.1 属性的选择

独立于方案的选择

搜索属性空间

特定方案的选择

88.2 离散数值属性

无指导的离散

基于熵的离散

其它离散方法

基于熵和基于误差的离散

离散属性转换为数值属性

88.3 一些有用的转换

主分量分析

随机投影

从文本到属性向量

时间序列

88.4 自动数据清理

改进决策树

稳健回归

侦察异情

88.5 组合多种模型

88.6 使用没有类标签的数据

Chapter 89

树模型

名词: 分类与回归树(Classification and Regression Trees, CART)

89.1 决策树

89.1.1 数值属性

对于数值属性, 我们将之限定为二元分裂. 考虑天气的例子, 温度属性为

```
yes  no  yes  yes  yes  no  no/yes  yes/yes  no  yes  yes  no
64   65  68   69   70   71  72      75       80  81   83   85
```

重复的数值叠在一起. 共有11个间断点, 如果不允许将属于一个类的实例分开, 就有8个. 每个的信息增量可以使用普通的方法计算. 例如使用$temperature = 71.5$测试, 划分的2类$< 71.5, > 71.5$分别产生yes和no的数目为$[4, 2], [5, 3]$, 信息值为

$$info([4, 2], [5, 3]) = 6/14 * info([4, 2]) + 8/14 * info([5, 3]) = 0.939bit$$

使用71和72的平均值作为分隔是一般的做法. 更精细的方法可能产生更好的效果, 但是由于复杂而使用很少.

一旦第一次排序完成, 后续的子节点就不需要再次排了.

对于名词属性, 可以对每个属性值创建分支, 但是, 数值属性被限定为二元分裂. 那么名词属性在根到叶子的路径中只能测试1次, 而数值属性可以被测试好多次. 对于单个数值属性的测试不是集中在一起的, 造成树的凌乱, 难以理解. 这是与名词属性的重大不同.

一个更简单但是功效欠佳的方法是离散化.

89.1.2 残缺值

一个方案是把残缺值当做另外一个属性值.

另一个是记录叶节点的实例数量, 如果一个实例的值残缺, 就分配到最多实例的叶节点上的相应的类别

更成熟的方法是将实例分裂成几部分, 分别分配到下面的分支, 并由此向下, 直到叶节点. 分裂过程采用0,1之间的权值完成. 一个分支的权值与其实例数量正比. 所有权值和为1.

另外一个问题是分裂属性选择后, 如何分裂训练集?

89.1.3 修剪

不加修剪的树往往很复杂且拟合过度, 预测性能不如简单的树.

先建立完整的树, 然后修剪, 称为后修剪(post-pruning). 对应的是前修剪(pre-pruning). 前修剪需要决定什么时候停止树的生长. 比后修剪计算量小, 但是当两个属性结合后提供的信息强时, 前修剪就不能发现了.

后修剪过程中考虑两个完全不同的操作: 子树置换和子树提升. 每个节点都要决定是替换, 提升, 还是保持不变.

子树置换是一个主要的修剪方法. 想法是选择一些子树, 用

一个节点代替.

提升是否必要不是很清楚. 但是在C4.5算法中使用. 它是使用一个节点的某个子树替换此节点. 提升的计算比较耗时, 实际中, 被限制在最为普及的分支.

89.1.4 估计误差率

修剪操作的决定是由误差率来决定的. 这通常使用交叉验证技术得到. 实际上, 在训练集上计算误差率没有意义, 因为不会得到任何的修剪支持.

具体算法: 略

89.1.5 从决策树到规则

通过每个节点建立一个规则, 联合所有节点的条件, 就直接得到一组规则集. 但是常常不必要的复杂.

还是使用误差率对规则进行修剪. 对于一个具体的规则, 考虑去除每个条件后的误差率, 并与原理的误差率比较. 如果误差率比较好, 则接受这个去除. 然后继续直到所有规则都修剪. 最后还要检查是否有重复的规则. 这是一个贪心算法, 很多时候可以产生满意的规则集.

另外的改进是考虑所有条件的子集, 但是计算复杂性太高.

另一种方法是使用模拟退伙或遗传算法来选择.

最后可以直接形成规则而不是先构树. 这种直接的方法可以较快.

89.2 数值预测

用于数值预测的树和普通的决策树一样. 但是在叶节点存储的是实例的平均值, 这种树称为回归树. 或者, 在叶节点存储一个能够达到此叶节点的实例的一个回归模型. 这种树称为模型树

回归树是模型树的特殊情况.

回归树和模型树先使用决策树归纳算法建立一个初始的树. 在决策树中, 属性分裂是根据信息增益最大决定的. 在数值预测中, 应该使节点内部类脂变化最小. 一旦建立基本树, 就要考虑修剪, 象决策树一样.

回归树和模型树归纳法的区别在于, 模型树的每个节点用一个回归平面代替一个常量值. 参与准确定义回归的属性, 正是那些参与决定子树修剪的属性, 即当前节点的下层节点(的属性).

89.2.1 平滑

节点根据属性的值决定分裂. 叶节点含有基于部分属性值的线性模型, 使得测试数据得到一个原始的预测值.

修剪过的树两个邻近的叶节点的线性模型之间不可避免的会冲突. 使用平滑处理可以大大提高预测的正确性. 当一个测试实例到达叶节点并根据模型得到一个预测值后, 就沿着树过滤返回根节点. 每个节点将其值与该节点提供的预测值结合进行平滑处理. 一个适当的平滑为

$$p' = \frac{np + kq}{n + k}$$

p'为要向上传递的值, p为下层传递来的值, q为此节点预测值. n是下层节点的实例数量, k是平滑系数(常量).

在树构建完成后, 把内部节点的模型组合到叶节点与平滑的效果一样, 这样只需要使用叶节点的模型. 但是叶节点模型

变得很大, 难以理解.

89.2.2 误差

可以把数据的标准差看作节点的误差, 衡量每个属性分裂后的误差减少值, 达到最大的作为分裂属性.

当误差占原始标准差很小的部分, 或叶节点的实例数量很少(例如, 小于4个), 终止分裂.

试验证明对这些阈值的选择并不敏感.

89.2.3 修剪树

每个节点都有一个模型.

修剪树也是根据误差来的. 对于某个节点修剪时, 只有这个节点下层的子树测试属性才用于回归及误差计算, 因为其它因素已经在产生此节点的过程中考虑进入了.

89.2.4 名词属性

在构建模型树前, 所有的名词属性都转换为二进制变量, 并被当做数值一样对待. 例如, 如果某个名词属性有3个不同的值, 那么可以将这个属性分为3个属性, 每个使用0,1表示

```
属性={A,B,C}
A: 1,0,0
B: 0,1,0
C: 0,0,1
```

可以证明, 在某个节点对于含k个值的名词属性, 最好的分裂点是按照每个属性值的平均类值大小排序所得到的$k-1$个位

置中的一个. 可以在建树前进行一次排列, 之后就避免每个节点排序的操作了.

89.2.5 残缺值

一个有趣的方法称为代理分裂(surrogate splitting). 它要寻找一个与含有残缺值相关程度最高的属性替代原来的属性. 但是这种技术复杂度很高.

一个比较简单的方法是使用类值(回归值)作为代理属性. 我们相信它是最有可能与当前属性相关的属性.

89.2.6 TODO: 从模型树到规则

89.2.7 局部加权线性回归

局部加权回归是在预测的时候产生一个局部模型. 计算待预测的实例与训练实例之间的距离, 距离近的训练实例权值高, 远的权值低. 也就是说, 为某个具体的测试实例特制一个线性模型.

距离可以使用欧氏距离, 权值为其倒数. 或欧氏距离与高斯核函数的组合. 主要的是衡量距离函数的平滑参数的选择. 距离要和此参数的倒数相乘. 如果太小, 只有非常靠近测试实例的训练实例才会得到显著的权值. 如果这个值大, 那么远的训练实例也有显著影响. 一种方法是设为离开第k个最近训练实例的距离. 从而随着训练集的增加, 距离越来越近. 而且容易受到噪声的干扰. 通常合适的参数通过交叉验证得到.

一个主要的优点是适合递增学习(在线学习): 所有训练在预测时完成.新的实例随时可以加入训练集. 但是获得对一个实例的预测会很慢.

实践证明局部加权的朴素贝叶斯法工作非常出色. 比朴素贝叶斯法和k最邻近法都好. 与增强的朴素贝叶斯法比较, 也能获得较好的结果.

局部加权学习从根本上来说是一种能够让简单模型更灵活的方法. 如果基本的学习算法已经足够好, 那么没有理由使用局部加权学习.

尽管如此, 局部加权学习可以改进其它简单模型, 例如, 线性支持向量机和logistic回归模型.

89.2.8 讨论

对于数值预测, 神经网络的方法更加常用, 虽然其理解起来比较困难, 相同的数据的训练可能导致内部结构差异很大的网络.

模型树提供了可复制的, 至少较易于理解的一种方法.

89.3 R包-party

参考文献: Torsten Hothorn, Kurt Hornik, Achim Zeileis *party: A Laboratory for Recursive Partytioning* 描述了算法与应用.

有漂亮的图.

89.3.1 回归分类树-ctree()

ctree() 用于条件推断树, 估计回归关系, 使用二分递归分裂, 基于条件推断框架. 算法为

1. 测试全局零假设, 即任意变量与相应变量(y值)之间无相关性. 如果不能拒绝, 停止算法. 否则, 选择相关性最强的输入变量(基于零假设的p-值)
2. 对选择的输入变量执行二分分裂
3. 递归前两个步骤

计 算 Multiplicity-adjusted Monte-Carlo p-values. 使 用 min-p 方 法. 对 于 数 据 的 每 个 随 机 排 列 的 单 个p-值 基 于 极 限 分 布(卡 方 或 正 态 分 布)计 算. 这 意 味 着 可 以 使 用 二 次 检 验 统 计 量(quadratic test statistic)当 因 子 are in play.

返 回 一 个 'BinaryTree-class' 类

其 参 数 由 另 外 一 个 函 数 控 制.详 细 见 帮 助

```
ctree_control(teststat = c("quad", "max"),
          testtype = c("Bonferroni", "MonteCarlo",
                    "Univariate", "Teststatistic"),
          mincriterion = 0.95, minsplit = 20, minbucket = 7,
          stump = FALSE, nresample = 9999, maxsurrogate = 0,
          mtry = 0, savesplitstats = TRUE, maxdepth = 0)
```

可 以 使 用 'predict' or 'treeresponse' 来 对 新 数 据 预 测.

```
# 下 面 是 分 类 问 题 的 一 个 例 子
ls <- data.frame(y = gl(3, 50, labels = c("A", "B", "C")),
   x1 = rnorm(150) + rep(c(1, 0, 0), c(50, 50, 50)),
   x2 = runif(150))

> ls
   y          x1          x2
1   A  2.418155814 0.6050644317
2   A  1.066076358 0.1948558574
3   A  1.281547459 0.6496647848
4   A  0.122317705 0.8234470612
5   A  0.792376555 0.2358141476
.....
51  B  0.270898357 0.5995101435
52  B -0.597492960 0.7725180071
53  B  0.644545301 0.3579502141
54  B -1.127929612 0.4000479376
55  B -0.434122667 0.1661824398
...
101 C  0.164744319 0.4000487912
```

```
102 C -1.873386524 0.1101657436
103 C  0.443162450 0.9538319607
104 C  0.414716965 0.2973919096
...

> library("party")
> ct=ctree(y ~ x1 + x2, data = ls); ct # 执行一个分类训练

        Conditional inference tree with 3 terminal nodes

Response:  y
Inputs:  x1, x2
Number of observations:  150

1) x1 <= 1.537034; criterion = 1, statistic = 46.816
  2) x1 <= -0.3102009; criterion = 0.996, statistic = 12.664
    3)*  weights = 35
  2) x1 > -0.3102009
    4)*  weights = 89
1) x1 > 1.537034
  5)*  weights = 26

> plot(ct) # 绘制分类树, 分支带分裂规则

# 对角线是分类准确的, 非对角线是错误的. 上下各是假阳
性, 假阴性, right???
> table(predict(ct),ls$y)

    A  B  C
  A 23  0  3
  B  1 24 10
  C 26 26 37

> predict(ct) # 在训练集上的表现.
  [1] A C C C C A C A A C C C A A A C C C A A A A C A C C A C C C C C C A C A C A
 [38] C A A C A C C A C A B A A C B C B B C B B C B C C B B B C C C B B B C C C
 [75] B C B C C C B C B B C B C C C C B B C B B B C C B C C B C C C B C C C C C
[112] C B C C A B C C C B C C B B C C C C C C A C C C C A C C C C B C C C C C B
[149] B C
Levels: A B C
```

```
# 对新数据的预测
> predict(ct,ls[2:3][1:10,])
 [1] A C C C C A C A A C
Levels: A B C

# 新数据的相应.
> treeresponse(ct, newdata = ls[1:10,])
[[1]]
[1] 0.8846154 0.0000000 0.1153846

[[2]]
[1] 0.2921348 0.2921348 0.4157303

[[3]]
[1] 0.2921348 0.2921348 0.4157303

[[4]]
[1] 0.2921348 0.2921348 0.4157303

[[5]]
[1] 0.2921348 0.2921348 0.4157303

[[6]]
[1] 0.8846154 0.0000000 0.1153846

[[7]]
[1] 0.2921348 0.2921348 0.4157303

[[8]]
[1] 0.8846154 0.0000000 0.1153846

[[9]]
[1] 0.8846154 0.0000000 0.1153846

[[10]]
[1] 0.2921348 0.2921348 0.4157303


# 对于回归问题, 其预测输出为. 详细见在线帮助
airq <- subset(airquality, !is.na(Ozone))
airct <- ctree(Ozone ~ ., data = airq,
```

```
          controls = ctree_control(maxsurrogate = 3))
airct
plot(airct)
mean((airq$Ozone - predict(airct))^2)

> predict(airct,airq[1:10,])
       Ozone
 [1,] 18.47917
 [2,] 18.47917
 [3,] 18.47917
 [4,] 18.47917
 [5,] 18.47917
 [6,] 18.47917
 [7,] 18.47917
 [8,] 18.47917
 [9,] 55.60000
[10,] 18.47917
```

89.3.2 模型树-mob()

参考文献 Achim Zeileis, Torsten Hothorn, Kurt Hornik *party with the mob: Model-based Recursive Partitioning in R* 有介绍算法和用法.

mob() 函数为基于参数模型(线性模型, 广义线性模型, 生存回归等)的递归分类算法, 产生一个模型树.

使用如下算法

1. 拟合一个模型(默认为广义线性模型 glm)
2. 使用用于分裂的属性评估模型参数的稳定性. 如果有全局不稳定的变量/属性, 选择最小p值的变量z分裂, 否则终止.
3. 搜索局部最优分裂, 基于最小化技术, 例如方差或对数似然
4. 在子节点重新拟合模型, 从第2步开始.

用法很直观, 参考帮助

```
example(mob)
```

89.4 R包-rpart

用法很直观.

```
rpart(formula, data, weights, subset, na.action = na.rpart, method,
     model = FALSE, x = FALSE, y = TRUE, parms, control, cost, ...)
```

参数 method 为 ”’anova”’, ”’poisson”’, ”’class”’ or ”’exp”’ 之一. 若缺失, 函数会自行判断是回归还是分类. 最好指定它.

- 如果y是 survival 对象, method=”exp”
- 如果y有2列, method=”poisson”
- 如果y为因子(factor), method=”class”
- 其它, method=”anova”

另外, ’method’ 也可以是一个函数列表, 叫做 ’init’, ’split’ and ’eval’. 例子在 ’tests/usersplits.R’.

model: 逻辑变量, 结果中是否保存模型框架.

parms: 分裂函数的参数选项.

- Anova splitting 没有参数.
- Poisson splitting 有一个单参数, 为速率先验分布的变量的系数(the coefficient of variation of the prior distribution on the rates.) 默认为1.

- Exponential splitting 与 Poisson splitting 参数一样.
- classification splitting, 此参数可以是: 先验概率向量(大于0且和为1), loss matrix(对角元素为0, 非对角大于0, 默认1), splitting index(可以是gini或information, 默认gini). 默认的先验概率为训练集的实例个数.

分裂参数的设定使用 rpart.control

```
rpart.control(minsplit=20, minbucket=round(minsplit/3), cp=0.01,
            maxcompete=4, maxsurrogate=5, usesurrogate=2, xval=10,
            surrogatestyle=0, maxdepth=30, ...)
```

- minsplit: 节点内最小的实例数
- minbucket: 叶节点最小实例数
- maxsurrogate: 代理变量的数目
- ...

例子参考帮助

```
example(rpart)
```

89.5 随机森林

89.5.1 资料1

参考网页: http://en.wikipedia.org/wiki/Random_forest

89.5.1.1 学习算法

1. 令实例总数量为N, 属性(变量)数目为M
2. 在一个树的节点, 使用m个属性决定类别. $m << M$
3. 对此树选择训练集N次, 使用重复采样, 例如 bootstrap 采样. 使用剩余的实例估计误差.
4. 对树的每个节点, 随机选择m个属性, 并基于此m个属性进行决策. 在训练集里基于m个属性计算最好的分叉.
5. 每个树充分生长, 不剪枝. 就像构建一个普通的分类树一样.

89.5.1.2 优点

- 对于很多数据集, 可以产生高度精确的分类结果
- 可以处理很大的属性个数(维数很高)
- 它可以估计属性的重要性
- 在构树的过程中产生误差的内部无偏估计(internal unbiased estimate of the generalization error)
- 包括一个好方法估计缺失数据, 当很大比例的数据缺失, 仍然保持高的正确率
- 提供一个经验方法估计属性的相互作用
- 数据集类别不平衡的情况下它可以平衡误差(It can balance error in class population unbalanced data sets)
- 它可以计算实例之间的相关性, 对于聚类, 异常检验(新颖性检验)和可视化数据(by scaling)比较有用
- 使用上述特点, 可以用于无标签数据, 进而执行一个无监督的聚类, 新颖性检测和数据可视化
- 学习是很快的

89.5.1.3 缺点

- 对于某些数据集可能过拟合. 对于一些有噪声的数据集更是如此
- 不能处理大量的不相干特征和整合熵减小的决策树(Random Forest does not handle large numbers of irrelevant features as well as ensembles of entropy-reducing decision trees)
- 它更容易选择一个随机的决策边界, 而不是熵-减小的决策边界. 这导致大的整合更加随意. 最初看来这可能象一个优点, 但是会导致其计算从训练的时候向预测的时候偏移, 对于很多应用来说是一个缺点.(It is more efficient to select a random decision boundary than an entropy-reducing decision boundary, thus making larger ensembles more feasible. Although this may seem to be an advantage at first, it has the effect of shifting the computation from training time to evaluation time, which is actually a disadvantage for most applications.)

89.5.2 TODO: 资料2

参考网页: http://www.stat.berkeley.edu/ breiman/RandomForests/cc_home.htm

特点多了一条: 它并不高于目前的分类器算法的正确率.(与文献1冲突)

早期的文献表明, 随机森林的误差由下面两个事情决定

- 任意两个树的相关性. 增加相关性增加误差
- 每个树的强度. 一个低错误率的树称为强分类器(strong classifier) 增加强度, 增加错误率.

减小m同时减小相关性和强度. m大概有一个最优的范围, 但是这个范围一般很大. 使用 oob(out of bag) error rate 在范围内的m可以很快发现. 这是随机森林唯一可调节的敏感参数.

速度测试: 50,000实例, 100个属性的数据集, 产生100个树, 在800Mhz 机器上用了 11min.

如果需要估计实例的相关性, 内存的需要是实例数乘以树的数目.

89.5.2.1 TODO: 误差估计

out-of-bag error 估计

89.5.2.2 TODO: 属性的重要性评估

对每个树, 计算正确分类的个数. 对于属性m, 随机排列oob实例的值, 将这些实例放入树里. 减去正确分类的投票数. 这些数值的平均值(打分)就是属性m重要性的原始值.

如果打分是树之间独立的, 那么标准差就可以计算. 经过验证, 它们的相关性是很低的. 即标准差是可靠的. 然后根据标准差计算其打分的z值, 并假设z值是正态分布的.

如果属性的个数很多, 可以对所有属性运行一次, 然后只使用最重要的属性运行.

89.5.2.3 TODO: Gini 重要性

每次根据m个属性分裂一个节点, 其2个分支的 gini 不纯性标准小于父节点. 略.....

89.5.2.4 TODO: 相互作用

89.5.2.5 TODO: 实例的相似性(proximities)

89.5.2.6 TODO: 缩放

89.5.2.7 TODO: 原型

89.5.2.8 TODO: 缺失数据的替换

89.5.2.9 TODO: 缺失标签的实例

89.5.2.10 TODO: 异常值检测

89.5.2.11 TODO: 无监督学习

89.5.2.12 TODO: 平衡预测误差

有时候预测误差在类别之间很不平衡.

89.5.2.13 TODO: 新颖性检测

使用异常值检测可以执行新颖性检测.

Chapter 90

判别分析(Discriminant Analysis)

参考生物数学[11] 2.3 (生物统计数学模型判别分析数学模型) 和 [21] 8.1 前者的假设与推导过程基于 Fisher 判别, 后者的三者都介绍了, 且自己编写若干函数.

90.1 判别分析与主成分分析的关系

主成分分析(PCA)方法对于代表数据样本非常有效, 但是却不是分类的有效方法. 例如, 区分大写字母 "O" 与 "Q", PCA 可以发现两个字母的相似之处, 却很可能把区分字母的"尾巴"特征抛弃掉了. 也就是说, PCA 寻找有效表示数据的主轴方向, 判别分析 (Discriminant Analysis) 是寻找的是用来有效分类的方向.

90.2 基于 Mahalanobis 距离的数学模型

设两个总体 X_1, X_2 的均值向量分别为 μ_1, μ_2, 协方差矩阵分别为 $\sum_1, \sum_2$, 今有一样本 x, 判断其来自哪个总体.

需要计算 x 与两个总体的 Mahalanobis 距离(的平方)然后比较. x 来自 X_1 若

$$d^2(x, X_1) <= d^2(x, X_2)$$

x 来自 X_2 若

$$d^2(x, X_1) > d^2(x, X_2)$$

即空间划分两个集合(判别准则)

$$R_1 = \{x | d^2(x, X_1) <= d^2(x, X_2)\}$$
$$R_2 = \{x | d^2(x, X_1) >= d^2(x, X_2)\}$$

90.2.1 协方差矩阵相同

当 $\mu_1 \neq \mu_2$, $\sum_1 = \sum_2 = \sum$, 考虑

$$\begin{aligned}
d^2(x, X_2) - d^2(x, X_1) &= (x-\mu_2)^T \sum^{-1}(x-\mu_2) - (x-\mu_1)^T \sum^{-1}(x-\mu_1) \\
&= (x^T \sum^{-1} x - 2x^T \sum^{-1} \mu_2 + \mu_2^T \sum^{-1} \mu_2) - (x^T \sum^{-1} x - 2x^T \sum^{-1} \mu_1 + \mu_1^T \sum^{-1} \\
&= 2x^T \sum^{-1}(\mu_1 - \mu_2) + (\mu_1 + \mu_2)^T \sum^{-1}(\mu_2 - \mu_1) \\
&= 2(x - \frac{\mu_1 + \mu_2}{2})^T \sum^{-1}(\mu_1 - \mu_2) \\
&\equiv 2(x - \bar{\mu})^T \sum^{-1}(\mu_1 - \mu_2) \\
&\equiv 2w(x)
\end{aligned}$$

其中

$$\bar{\mu} = \frac{\mu_1 + \mu_2}{2}$$

称 $w(x)$ 为两个总体距离的判别函数.

$$w(x) = (x - \bar{\mu})^T \sum^{-1}(\mu_1 - \mu_2)$$

判别准则变为

$$R_1 = \{x|w(x) >= 0\}$$
$$R_2 = \{x|w(x) < 0\}$$

实际上总体的均值与协方差矩阵是未知的, 需要用样本的均值与协方差矩阵来代替.

设 $x_1^{(1)}, \cdots, x_{n1}^{(1)}$ 来自总体 X_1 的 $n1$ 个样本, $x_1^{(2)}, \cdots, x_{n2}^{(2)}$ 来自总体 X_2 的 $n2$ 个样本. 则样本的均值为

$$\begin{aligned}
\hat{\mu_i} &= \frac{1}{n_i}\sum_{j=1}^{n_i} x_j^{(i)}, i = 1, 2 \\
\hat{\sum} &= \frac{1}{n_1 + n_2 - 2}\sum_{i=1}^{2}\sum_{j=1}^{n_i}(x_j^{(i)} - \hat{\mu_i})(x_j^{(i)} - \hat{\mu_i})^T \\
&= \frac{1}{n_1 + n_2 - 2}(S_1 + S_2) \\
\hat{\mu} &= \frac{\mu_1 + \mu_2}{2}
\end{aligned}$$

判别函数变为

$$\hat{w}(x) = (x - \hat{\mu})^T \hat{\sum}^{-1} (\hat{\mu_1} - \hat{\mu_2})$$

判别准则变为

$$R_1 = \{x|\hat{w}(x) >= 0\}$$
$$R_2 = \{x|\hat{w}(x) < 0\}$$

90.2.2 协方差矩阵不同

判别准则不变, 与协方差矩阵相同时相同.

判别函数为

$$w(x) = (x-\mu_2)^T \sum_2^{-1}(x-\mu_2) - (x-\mu_1)^T \sum_1^{-1}(x-\mu_1)$$

使用样本代替后判别函数变为

$$\hat{w}(x) = (x-\hat{\mu_2})^T \hat{\sum}_2^{-1}(x-\hat{\mu_2}) - (x-\hat{\mu_1})^T \hat{\sum}_1^{-1}(x-\hat{\mu_1})$$

其中

$$\begin{aligned} \hat{\sum}_i &= \frac{1}{n_i-1}\sum_{j=1}^{n_i}(x_j^{(i)}-\hat{\mu_i})(x_j^{(i)}-\hat{\mu_i})^T \\ &= \frac{1}{n_i-1}S_i, \quad i=1,2 \end{aligned}$$

90.3 Bayes 判别

Bayes 判别是假定分析之前对研究对象已经有一定的认识, 这种认识数学上描述为先验概率. 取得样本后, 再使用样本修正先验概率分布, 就得到后验概率分布. 然后使用后验概率分布进行各种统计推断.

90.3.1 先验概率与损失函数

考虑两个总体的情况. 设 X_1, X_2 分别具有概率密度函数 $f_1(x), f_2(x)$. 其中 x 是 p 维向量. 记 Ω 为样本空间(即 x 所有可能的观察值的全体). R_1 为根据某种规则判为 X_1 的 x 的全体(这些 x 不一定都来自 X_1), R_2 为根据某种规则判为 X_2 的 x 的全体(这些 x 也不一定都来自 X_2), 而 $R_1 + R_2 = \Omega$.

某些样本来自 X_1 但是被误判为 X_2 的概率为

$$P(2|1) = P\{x \in R_2|X_1\} = \int_{R_2} f_1(x)dx$$

来自 X_2 但是被误判为 X_1 的概率为

$$P(1|2) = P\{x \in R_1|X_2\} = \int_{R_1} f_2(x)dx$$

类似, 来自 X_1 被判为 X_1 的概率为

$$P(1|1) = P\{x \in R_1|X_1\} = \int_{R_1} f_1(x)dx$$

来自 X_2 被判为 X_2 的概率为

$$P(2|2) = P\{x \in R_2|X_2\} = \int_{R_2} f_2(x)dx$$

设 p_1, p_2 为总体的先验概率, 且 $p_1 + p_2 = 1$, 于是

$$\begin{aligned} P(\text{正确的判为}X_1) &= P(\text{来自}X_1, \text{被判为}X_1) \\ &= P(x \in R_1|X_1) * P(X_1) \\ &= P(1|1) * p_1 \\ P(\text{误判为}X_1) &= P(\text{来自}X_2, \text{被判为}X_1) \\ &= P(x \in R_1|X_2) * P(X_2) \\ &= P(1|2) * p_2 \end{aligned}$$

类似的

$$\begin{aligned} P(\text{正确的判为}X_2) &= P(2|2) * p_2 \\ P(\text{误判为}X_2) &= P(2|1) * p_1 \end{aligned}$$

设 $L(1|2)$ 表示来自 X_2 被误判为 X_1 引起的损失, $L(2|1)$ 表示来自 X_1 被误判为 X_2 引起的损失, 并规定 $L(1|1) = L(2|2) = 0$.

将误判概率与误判损失结合, 定义平均误判损失(expected cost of misclassification, ECM) 为

$$ECM(R_1, R_2) = L(2|1)P(2|1)p_1 + L(1|2)P(1|2)p_2$$

合理的选择是使 ECM 达到极小.

90.3.2 两个总体的 Bayes 判别

可以证明, 极小化平均误判损失函数的划分为

$$R_1 = \{x|\frac{f_1(x)}{f_2(x)} >= \frac{L(1|2)}{L(2|1)}\frac{p_2}{p_1}\}$$

$$R_2 = \{x|\frac{f_1(x)}{f_2(x)} < \frac{L(1|2)}{L(2|1)}\frac{p_2}{p_1}\}$$

因此可以将此式作为 Bayes 判别的准则. 我们只需要计算

- 样本点 x 的概率密度函数比 $f_1(x)/f_2(x)$
- 损失比 $L(1|2)/L(2|1)$
- 先验概率比 p_2/p_1

首先考虑总体协方差矩阵相同的情况, 此时 X_i 的密度为

$$f_i(x) = (2\pi)^{-\pi/2}|\sum|^{-1/2}exp\{-\frac{1}{2}(x-\mu_i)^T\sum^{-1}(x-\mu_i)\}, \quad i = 1, 2$$

因此, R_1, R_2 的划分区域等价于

$$R_1 = \{x|W(x) >= \beta\}$$
$$R_2 = \{x|W(x) < \beta\}$$

其中

$$\begin{aligned} W(x) &= \frac{1}{2}(x-\mu_2)^T\sum^{-1}(x-\mu_2) - \frac{1}{2}(x-\mu_1)^T\sum^{-1}(x-\mu_1) \\ &= [x - \frac{1}{2}(\mu_1+\mu_2)]^T\sum^{-1}(\mu_1-\mu_2) \\ \beta &= ln\frac{L(1|2)p_2}{L(2|1)p_1} \end{aligned}$$

不难看出, 对于正态分布总体的 Bayes 判别, 其判别规则可以看成 Mahalanobis 距离判别的推广, 当

$$p_1 = p_2, \quad L(1|2) = L(2|1), \quad \beta = 0$$

就是 Mahalanobis 距离判别.

考虑协方差矩阵不同的情况,

$$R_1 = \{x|W(x) >= \beta\}$$
$$R_2 = \{x|W(x) < \beta\}$$

其中

$$\begin{aligned} W(x) &= \frac{1}{2}(x-\mu_2)^T \sum_2^{-1}(x-\mu_2) - \frac{1}{2}(x-\mu_1)^T \sum_1^{-1}(x-\mu_1) \\ \beta &= ln\frac{L(1|2)p_2}{L(2|1)p_1} + \frac{1}{2}ln(\frac{|\sum_1|}{|\sum_2|}) \end{aligned}$$

编写 R 程序: 略

90.3.3 多分类问题的 Bayes 判别

从上面的计算过程可以看到, Bayes 判别本质上就是找到一种判别准则, 使得平均误判损失达到最小, 即相应的概率达到最大.

假设样本有 k 类, 分别为 $X_1, \cdots, X_k$, 相应的先验概率为 $p_1, \cdots, p_k$, 假设所有错判的损失是相同的, 因此相应的判别准则为

$$R_i = \{x|p_i f_i(x) = max_{1<=j<=k} p_j f_j(x)\}, \quad i = 1, \cdots, k$$

当 k 类总体的协方差矩阵相同, 即

$$\sum_1 = \cdots = \sum_k = \sum$$

此时概率密度函数为

$$f_j(x) = (2\pi)^{-\pi/2}|\sum|^{-1/2}exp\{-\frac{1}{2}(x-\mu_j)^T\sum^{-1}(x-\mu_j)\}, \quad j = 1, \cdots, k$$

则计算函数

$$d_j(x) = \frac{1}{2}(x-\mu_j)^T\sum^{-1}(x-\mu_j) - lnp_j$$

计算中, 协方差矩阵使用其估计值代替.

当协方差矩阵不同时, 此时概率密度函数为

$$f_j(x) = (2\pi)^{-\pi/2}|\sum_j|^{-1/2}exp\{-\frac{1}{2}(x-\mu_j)^T\sum_j^{-1}(x-\mu_j)\}, \quad j = 1, \cdots, k$$

则计算函数

$$d_j(x) = \frac{1}{2}(x-\mu_j)^T\sum_j^{-1}(x-\mu_j) - lnp_j - \frac{1}{2}ln(|\sum_j|)$$

计算中, 协方差矩阵也分别使用其估计值代替.

判别准则等价为

$$R_i = \{x|d_i(x) = min_{1<=j<=k}d_j(x)\}, \quad i = 1, \cdots, k$$

编写 R 程序: 略

90.4 Fisher 判别

具体算法: 参考 [11] [21]

90.4.1 问题描述

假设有两个类群已知其分类为1和2, 属于类群1的标记为+1, 属于类群2的标记为−1, 即训练集合$S = ((x_1, y_1), \cdots, (x_l, y_l))$, 其中$x_i$来自$R^n$维空间, 是个n维向量. 对应标签$y_i$来自一维空间, 取值为+1, −1. 目标是寻找比较简单的原则(函数), 将这两个类别分开, 使得误差(错误分类的个数)尽可能的小.

数据如下,

考虑如何确定分类的判别函数. 确定判别函数的方法很多. 有数值的方法和非数值的方法. 非数值方法例如, knn聚类, 系统聚类, 神经网络, 决策树等等. 下面我们考虑数值的方法.

数值方法中现在流行的有支持向量机(SVM)方法, 原理是考虑对分类起决定作用的实际上取决于类别边际的点, 而类别内部的点实际上对分类不起作用. 这样连接类别边际的点, 可以得到一个凸的多面体(二维的就是多边形). 如果类别之间的多边形不相交的话, 就可以把类别分开. 而非线性可分的多个类别实际上可以使用某种非线性映射, 总可以使其称为线性可分的, 然后再使用SVM方法来分类即可. 这个非线性映射函数的寻找现在还只是凭借经验, 没有什么理论可循. 映射之后的判别函数实际上可以使用内积的形式来计算, 这样实际上不需要计算映射, 只要计算内积就可以, 这种方法叫做核方法(kernel method). 在映射函数计算比较复杂的情况下是很有用的, 而且映射到无穷维都有可能了. 因为无穷维的内积可能不是无穷的. 这个问题后面讨论.

介绍Fisher判别的思路前先复习下点与超平面的距离

90.4.2 点与超平面的距离

回忆几何学中的定义, 有

若在平面坐标几何上的直线定义为 $ax + by + c = 0$, 点的座标为 $(x0, y0)$, 则它们之间的距离为:

$$d = \left| \frac{ax_0 + by_0 + c}{\sqrt{a^2 + b^2}} \right|$$

若点坐标为$(x0, y0, z0)$,平面为$Ax + By + Cz + D = 0$,则点到平面的距离为:

$$d = \left| \frac{Ax_0 + By_0 + Cz_0 + D}{\sqrt{A^2 + B^2 + C^2}} \right|$$

推广到超平面, 若R^n空间中点坐标为$x = (x_1, \cdots, x_n) \in R^n$, R^n空间中的超平面可以使用一个系数向量w和平移(偏置)b表示为$< w, x > +b$, 设$f(x) =< w, x > +b$, $f(x) = 0$就是R^n空间中的一个超平面. 角度旋转由w控制, 平移由b控制. 某点与此超平面的距离为

$$d = \left| \frac{f(x)}{||w||} f(x) \right| = \left| \frac{w_1 x_1 + \cdots + w_n x_n + b}{\sqrt{w_1^2 + \cdots + w_n^2}} \right|$$

90.4.3 数据描述

下面列出数据符号表示. 假设n维数据(即n个属性, 指标等等), 类别1的样本数为p, 类别2的样本数为q, 类别1的数据表示如下

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \cdots & & & \\ x_{p1} & x_{p2} & \cdots & x_{pn} \end{bmatrix}$$

类别2的数据为

$$X' = \begin{bmatrix} x'_{11} & x'_{12} & \cdots & x'_{1n} \\ x'_{21} & x'_{22} & \cdots & x'_{2n} \\ \cdots & & & \\ x'_{q1} & x'_{q2} & \cdots & x'_{qn} \end{bmatrix}$$

$x_i = (x_{i1}, x_{i2}, \cdots, x_{in})$, $i = 1, \cdots, p$为类别1的第i个样本. $x'_i = (x'_{i1}, x'_{i2}, \cdots, x'_{in})$, $i = 1, \cdots, q$为类别2的第i个样本.

$\bar{x} = (\bar{x}_{\cdot 1}, \bar{x}_{\cdot 2}, \cdots, \bar{x}_{\cdot n})$为 类 别1的 质 心. $\bar{x'} = (\bar{x'}_{\cdot 1}, \bar{x'}_{\cdot 2}, \cdots, \bar{x'}_{\cdot n})$为 类 别2的 质 心, 即 每 列 的 平 均 值. 其 中

$$\bar{x}_{\cdot k} = \frac{1}{p}\sum_{i=1}^{p} x_{ik}$$

为 第k列 的 平 均 值, $k = 1, \cdots, n$.

Fisher判 别 是 要 寻 找 一 条 直 线, 平 面 或 超 平 面, 将 类 别 为1的 点 尽 可 能 的 分 在 一 侧, 类 别 为2的 在 另 一 侧. 此 判 别 直 线, 平 面 或 超 平 面 统 一 称 为 判 别 曲 面. 记 此 判 别 曲 面 为

$$f(x) =< w, x > +b = 0$$

其 中$w = w_1, w_2, \cdots, w_n$, 和b是 需 要 确 定 的.

类 别1的 每 个 样 本 点x_i到 此 判 别 曲 面 的 距 离 为

$$d_i =< w, x_i > +b/||w||$$

因 为 是 线 性 函 数, 其 平 均 距 离 就 是 质 心 到 判 别 曲 面 的 距 离, 为

$$d =< w, \bar{x} > +b/||w||$$

同 样, 类 别2的 每 个 样 本 点x'_i和 质 心 到 此 判 别 曲 面 的 距 离 分 别 为

$$d'_i =< w, x'_i > +b/||w||$$
$$d' =< w, \bar{x'} > +b/||w||$$

90.4.4 Fisher判 别 分 类 的 思 路

类 别 差 异 的 度 量

观 察 到d, d'符 号 相 反. $|d| + |d'| = |d - d'|$. 记 两 个 类 别 之 间 的 距 离 度 量 为

$$A = (d - d')^2$$

若只考虑此距离, 当取距离质心相等的点, 作其连线的垂直线(曲面)为判别曲面, 或进一步, 考虑两个类别点的数量不同, 取距离样本数多的类别远, 样本数少的质心近的点, 作其连线的垂直线(曲面)为判别曲面, 实际上是根据样本数的多少取加权距离. 但是还要考虑类别内离散程度的度量.

类别内离散程度的度量

一般考虑离散程度的度量为点到质心的距离平方和的平均, 即类内方差. 但如此考虑的话, 就与判别曲面没有关系了, 因为只要样本点确定了, 其方差就确定了. Fisher考虑的是每个样本点到判别曲面的距离减去质心到判别曲面的距离的方差, 作为类别内部离散程度的度量, 即

$$B = \sum_{i=1}^{p}(d - d_i)^2 + \sum_{i=1}^{q}(d' - d_i')^2$$

判别准则

Fisher判别考虑使A尽可能的大, B尽可能的小. 联合这两个要求, 构造函数

$$I = \frac{A}{B}$$

使得I尽可能的大.

观察到A和B中参数b已经被减去, 实际上I只是w的函数. 下面是要求I的极大值. 但是直接求$\frac{\partial I}{\partial w_k} = 0$比较困难. 考虑到对数函数是严格单调的, 对I进行对数变换, 有

$$\ln I = \ln A - \ln B$$

求I的极值与求$\ln I$的极值是一致的. 分别对上式作$w_k, k = 1, \cdots, n$的偏导, 得到n个方程式

$$\frac{1}{I}\frac{\partial I}{\partial w_k} = \frac{1}{A}\frac{\partial A}{\partial w_k} - \frac{1}{B}\frac{\partial B}{\partial w_k} = 0$$

即

$$\frac{1}{I}\frac{\partial A}{\partial w_k} = \frac{\partial B}{\partial w_k}$$

其 中

$$A = (d - d')^2 =< w, (\bar{x} - \bar{x'}) >^2 = [\sum_{k=1}^{n} w_k(\bar{x}_{\cdot k} - \bar{x'}_{\cdot k})]^2$$

$$\begin{aligned} B &= \sum_{i=1}^{p}(d - d_i)^2 + \sum_{i=1}^{q}(d' - d'_i)^2 \\ &= \sum_{i=1}^{p} < w, (\bar{x} - x_i) >^2 + \sum_{i=1}^{q} < w, (\bar{x'} - x'_i) >^2 \\ &= \sum_{i=1}^{p}[\sum_{k=1}^{n} w_i(\bar{x}_{\cdot k} - x_{ik})]^2 + \sum_{i=1}^{q}[\sum_{k=1}^{n} w_i(\bar{x'}_{\cdot k} - x'_{ik})]^2 \end{aligned}$$

偏 导 方 程 式 对w_k求 导, 左 边 为

$$\frac{1}{I}\frac{\partial A}{\partial w_k} = \frac{2}{I}(\bar{x}_{\cdot k} - \bar{x'}_{\cdot k}) \cdot [\sum_{k=1}^{n} w_k(\bar{x}_{\cdot k} - \bar{x'}_{\cdot k})] = \frac{2}{I}(\bar{x}_{\cdot k} - \bar{x'}_{\cdot k}) \cdot (d - d')$$

右 边 为

$$\begin{aligned} B &= 2\sum_{i=1}^{p} < w, (\bar{x} - x_i) > \cdot(\bar{x}_{\cdot k}) + 2\sum_{i=1}^{q} < w, (\bar{x'} - x'_i) > \cdot(\bar{x'}_{\cdot k}) \\ &= 2 < w, [\sum_{i=1}^{p}(\bar{x} - x_i) \cdot (\bar{x}_{\cdot k}) + \sum_{i=1}^{q}(\bar{x'} - x'_i) \cdot (\bar{x'}_{\cdot k})] > \\ &= 2 < w, S_k > \end{aligned}$$

依次对$k=1,\cdots,n$求导, 得到n个方程式

$$< w, S_1 >= \frac{1}{I}(\bar{x}_{\cdot 1} - \bar{x'}_{\cdot 1}) \cdot (d - d')$$
$$\cdots$$
$$< w, S_n >= \frac{1}{I}(\bar{x}_{\cdot n} - \bar{x'}_{\cdot n}) \cdot (d - d')$$

注意到右边因子$\frac{1}{I}(d-d')$将其除到左边, 相当于w每个值除以因子, 其解差一个常数因子. 记因子$\frac{1}{I}(d-d') = \alpha, w' = w/\alpha$, 方程组变为

$$< w', S_1 >= (\bar{x}_{\cdot 1} - \bar{x'}_{\cdot 1})$$
$$\cdots$$
$$< w', S_n >= (\bar{x}_{\cdot n} - \bar{x'}_{\cdot n})$$

据此解得w'. 下面我们会看到因子α不需要求出.

判别函数

过类别1的质心的曲面设为

$$f(x) =< w, \bar{x} > +b1 =< w'\alpha, \bar{x} > +b1 = 0 \Longrightarrow b1 = - < w'\alpha, \bar{x} >$$

同理类别2的质心的曲面设为

$$f(x) =< w, \bar{x'} > +b2 =< w'\alpha, \bar{x'} > +b2 = 0 \Longrightarrow b2 = - < w'\alpha, \bar{x'} >$$

考虑判别曲面的偏置b, 如果两个类别的样本数差别比较大, 那么, 判别曲面应该靠近样本数小的一边(具体可以考虑样本服从正态分布), 偏置可以取为b1,b2的加权平均

$$b = \frac{pb1 + qb2}{p+q} = \frac{\alpha(-p < w', \bar{x} > -q - < w', \bar{x'} >)}{p+q} = \alpha b'$$

设有新的点x_{new}, 实际上判别函数是

$$f(x_{new}) = sgn(< w'\alpha, x_{new} > +b) = \alpha sgn(< w', x_{new} > +b)$$

考虑到若$\alpha > 0$, $f(x_{new}) = +1$则判定为类别1, $f(x_{new}) = +1$则判定为类别2, 否则判定相反.

90.5 例子

程序包 MASS: 函数 lda(Linear discriminant analysis), qda(Quadratic discriminant analysis.)

mda 程序包提供 mixture and flexible discriminant analysis with mda() and fda()

lda() method 参数有四种方法

- "moment": for standard estimators of the mean and variance, 标准的均值与方差估计
- "mle": for MLEs, 最大似然估计
- "mve": to use 'cov.mve' (minimum volume ellipsoid)
- "t" for robust estimates based on a t distribution. 基于 t 分布的估计

下面是一个一维的例子. lda() 自带的例子是 4 维的

```
> library(MASS)
> d=data.frame(x=(1:20),g=c(rep(1,10),rep(2,10)))
> d
   x g
1   1 1
2   2 1
3   3 1
4   4 1
5   5 1
6   6 1
7   7 1
8   8 1
9   9 1
10 10 1
11 11 2
12 12 2
13 13 2
```

```
14 14 2
15 15 2
16 16 2
17 17 2
18 18 2
19 19 2
20 20 2

> l=lda(g~x,data=d)
> l
Call:
lda(g ~ x, data = d)

Prior probabilities of groups:
  1   2
0.5 0.5

Group means:
    x
1  5.5
2 15.5

Coefficients of linear discriminants:
       LD1
x 0.3302891

# 新数据必须使用data.frame 且 x 标记,
# 即与 lda() 函数使用的变量名称一样
> new = data.frame(x=(5:15))
> predict(l,new)
> predict(l,new)
$class
 [1] 1 1 1 1 1 1 2 2 2 2 2
Levels: 1 2

$posterior
          1         2
1  0.99752738 0.002472623
2  0.99267486 0.007325140
3  0.97850450 0.021495499
4  0.93861689 0.061383107
```

```
5  0.83703953 0.162960471
6  0.63308037 0.366919631
7  0.36691963 0.633080369
8  0.16296047 0.837039529
9  0.06138311 0.938616893
10 0.02149550 0.978504501
11 0.00732514 0.992674860

$x
         LD1
1  -1.8165902
2  -1.4863011
3  -1.1560120
4  -0.8257228
5  -0.4954337
6  -0.1651446
7   0.1651446
8   0.4954337
9   0.8257228
10  1.1560120
11  1.4863011
```

Chapter 91

聚类分析

此部分主要参考"统计建模与R软件" [21] 及"生物数学" [11] 相关部分.

聚类分析中, 大多数据往往不能直接参加运算, 需要先中心化或标准化.

91.1 系统聚类(hierarchical clustering method)

使用最多. 设开始有 n 个样本,

$$X_1, \cdots, X_n$$

其中每个样本 p 维 (p 个性状), 第 k 个样本为

$$X_k = x_{1k}, \cdots, x_{pk}$$

基本思想是,

1. 开始把每个样本作为一类, 计算 n 个样本之间的 $n*n$ 距离矩阵

2. 取距离最短的两个样本合并成为一个新类.

3. 然后计算此新类与其它样本的距离, 产生 $(n-1)*(n-1)$ 距离矩阵.

4. 取此矩阵中距离最短的两个样本又合并成为一个新类.

5. 然后计算此新类与其它样本的距离, 产生 $(n-2)*(n-2)$ 距离矩阵.

6. $\cdots$

依次进行, 直到最后只有一类, 结束计算.

按照计算新类与其它样本的距离的方法不同可以分为最短距离法, 最长距离法, 中间距离法等.

91.1.1 最短距离法(the shortest distance method)

计算新类与其它样本的距离的方法为: 设第一步 1,2 被合并为 $n+1$, 下面要计算 $n+1$ 与 $3,4,\cdots,n$ 的距离,

$$\begin{aligned} d_{n+1,3} &= min(d_{1,3}, d_{2,3}) \\ d_{n+1,4} &= min(d_{1,4}, d_{2,4}) \\ d_{n+1,5} &= min(d_{1,5}, d_{2,5}) \\ &\cdots \\ d_{n+1,n} &= min(d_{1,n}, d_{2,n}) \end{aligned}$$

91.1.2 最长距离法(the longest distance method)

计算新类与其它样本的距离的方法为: 设第一步 1,2 被合并为 $n+1$, 下面要计算 $n+1$ 与 $3,4,\cdots,n$ 的距离

$$\begin{aligned} d_{n+1,3} &= max(d_{1,3}, d_{2,3}) \\ d_{n+1,4} &= max(d_{1,4}, d_{2,4}) \\ d_{n+1,5} &= max(d_{1,5}, d_{2,5}) \\ &\cdots \\ d_{n+1,n} &= max(d_{1,n}, d_{2,n}) \end{aligned}$$

91.1.3 中间距离法(median method)

计算新类与其它样本的距离的方法为: 设第一步 1,2 被合并为 $n+1$, 下面要计算 $n+1$ 与 $3,4,\cdots,n$ 的距离

$$d_{n+1,3}=\sqrt{\frac{1}{2}(d_{1,3}^2+d_{2,3}^2)-\frac{1}{4}d_{1,2}^2}$$

$$d_{n+1,4}=\sqrt{\frac{1}{2}(d_{1,4}^2+d_{2,4}^2)-\frac{1}{4}d_{1,2}^2}$$

$$d_{n+1,5}=\sqrt{\frac{1}{2}(d_{1,5}^2+d_{2,5}^2)-\frac{1}{4}d_{1,2}^2}$$

$$\cdots$$

$$d_{n+1,n}=\sqrt{\frac{1}{2}(d_{1,n}^2+d_{2,n}^2)-\frac{1}{4}d_{1,2}^2}$$

实际上采用的是 k 与 1,2 连线中线的距离作为新距离

91.1.4 中间距离法的推广

计算新类与其它样本的距离的方法为: 设第一步 1,2 被合并为 $n+1$, 下面要计算 $n+1$ 与 $3,4,\cdots,n$ 的距离

$$d_{n+1,3}=\sqrt{\frac{1-\beta}{2}(d_{1,3}^2+d_{2,3}^2)-\beta d_{1,2}^2}$$

其中 $\beta<1$. 当 $\beta=0$, 称为 Mcquitty 相似分析法.

91.1.5 类平均法(average linkage method)

两种定义. 一种是把类与类之间的距离定义为所有样本之间的平均距离.

设合并后的类 G_K, G_L 分别有 n_K, n_L 个样本, 定义 G_K, G_L 之间的距离为

$$d_{KL} = \frac{1}{n_K n_L} \sum_{i \in G_K, j \in G_L} d_{ij}$$

例如 G_K 为 1,2 合并得到, G_L 为 3,4 合并得到, 那么

$$d_{KL} = \frac{1}{2 * 2}(d_{1,3} + d_{1,4} + d_{2,3} + d_{2,4})$$

另一种定义为样本对之间平方距离的平均作为距离的平方

$$d^2_{KL} = \frac{1}{n_K n_L} \sum_{i \in G_K, j \in G_L} d^2_{ij}$$

递推公式为

$$d^2_{MJ} = \frac{n_K}{n_M} d^2_{KJ} + \frac{n_L}{n_M} d^2_{LJ}$$

类平均法较好利用了所有样本的信息, 很多时候被认为是一种较好的距类方法.

进一步推广为

$$d^2_{MJ} = (1 - \beta)(\frac{n_K}{n_M} d^2_{KJ} + \frac{n_L}{n_M} d^2_{LJ}) + \beta d^2_{KL}$$

其中 $\beta < 1$ 称为可变类平均法.

91.1.6 重心法

类与类(每个类大于 3 个样本, 2 个样本的重心在中点, 1 个就是其本身)之间的距离定义为它们重心之间的 Euclide 距离.

此方法处理异常值比其它方法更稳健, 但是在别的方面一般不如类平均法或离差平方和法效果好.

91.1.7 离差平方和法(Ward 法)

Ward (1936) 提出此方法. 基于方差分析的思想, 如果类分的正确, 则同类内部的离差平方和应较小, 不同类之间的离差平方和应较大.

与重心法比较差别在于一个常数系数(类内一般个数), 那么结果两个大类的距离倾向于比较大, 不易合并, 更符合对聚类的实际要求. 故很多情况下优于重心法.

但是对异常值敏感.

设 G_K, G_L 合并为新类 G_M, 则内部的离差平方和为

$$W_K = \sum_{i \in G_K} (x_{(i)} - \bar{x}_K)^T (x_{(i)} - \bar{x}_K)$$

其中 $\bar{x}_K$ 为 G_K 的重心. 类似的有

$$W_L = \sum_{i \in G_L} (x_{(i)} - \bar{x}_L)^T (x_{(i)} - \bar{x}_L)$$

$$W_M = \sum_{i \in G_M} (x_{(i)} - \bar{x}_M)^T (x_{(i)} - \bar{x}_M)$$

如果 G_K, G_L距离较近, 则合并后增加的离差平方和 $W_M - W_K - W_L$ 应较小. 否则, 应较大. 定义 G_K, G_L 平方距离为

$$d^2_{KL} = W_M - W_K - W_L$$

这种方法称为离差平方和法或 Ward 方法(Ward's minimum variance method)

递推公式为

$$d^2_{MJ} = \frac{n_J + n_K}{n_J + n_M} d^2_{KJ} + \frac{n_J + n_L}{n_J + n_M} d^2_{LJ} - \frac{n_J}{n_J + n_M} d^2_{KL}$$

G_K, G_L 之间的平方距离也可以写成

$$d^2_{KL} = \frac{n_K n_L}{n_M} (\bar{x}_K - \bar{x}_L)^T (\bar{x}_K - \bar{x}_L)$$

91.1.8 其它方法

采用各种平均, 例如使用所有联系两个类群之间的分类单位(样本)距离的平方和再取平均值, 作为距离的平方.

91.2 例子

hclust() 函数执行系统分类.

```
> x<-c(1,2,6,8,11); dim(x)<-c(5,1);x
     [,1]
[1,]    1
[2,]    2
[3,]    6
[4,]    8
[5,]   11
> d<-dist(x); d
   1  2  3  4
2  1
3  5  4
4  7  6  2
5 10  9  5  3
> h1<-hclust(d,"single");
> h2<-hclust(d,"complete")
> h3<-hclust(d,"median")
> h4<-hclust(d,"mcquitty")

> par(mfrow=c(2,2))
> plot(h1,hang=-1)
> plot(h2,hang=-1)
> plot(h3,hang=-1)
> plot(h4,hang=-1)

# hang =-1 表示线画到底. 查看 hang 的用法
> par(mfrow=c(2,1))
> plot(h4,hang=-1)
> plot(h4)
```

```
# as.dendrogram 可以绘制更好的图
> d1<-as.dendrogram(h1)
> str(d1)
--[dendrogram w/ 2 branches and 5 members at h = 4]
  |--[dendrogram w/ 2 branches and 2 members at h = 1]
  |  |--leaf 1
  |  `--leaf 2
  `--[dendrogram w/ 2 branches and 3 members at h = 3]
     |--leaf 5
     `--[dendrogram w/ 2 branches and 2 members at h = 2]
        |--leaf 3
        `--leaf 4

par(mfrow=c(2,2))
plot(d1)
plot(d1, nodePar=list(pch = c(1,NA), cex=0.8, lab.cex=0.8),
    type = "t", center=TRUE)
plot(d1, edgePar=list(col = 1:2, lty = 2:3),
     dLeaf=1, edge.root = TRUE)
plot(d1, nodePar=list(pch = 2:1, cex=.4*2:1, col=2:3),
     horiz=TRUE)
```

下面是一个比较实际的例子 ([21] page 488). 数据为全国 31 个省级地区居民的 8 项基本消费支出, 我们要把 31 个样本分类.

```
X<-data.frame(
x1=c(2959.19, 2459.77, 1495.63, 1046.33, 1303.97, 1730.84,
     1561.86, 1410.11, 3712.31, 2207.58, 2629.16, 1844.78,
     2709.46, 1563.78, 1675.75, 1427.65, 1783.43, 1942.23,
     3055.17, 2033.87, 2057.86, 2303.29, 1974.28, 1673.82,
     2194.25, 2646.61, 1472.95, 1525.57, 1654.69, 1375.46,
     1608.82),
x2=c(730.79, 495.47, 515.90, 477.77, 524.29, 553.90, 492.42,
     510.71, 550.74, 449.37, 557.32, 430.29, 428.11, 303.65,
     613.32, 431.79, 511.88, 512.27, 353.23, 300.82, 186.44,
     589.99, 507.76, 437.75, 537.01, 839.70, 390.89, 472.98,
     437.77, 480.99, 536.05),
```

```
x3=c(749.41, 697.33, 362.37, 290.15, 254.83, 246.91, 200.49,
    211.88, 893.37, 572.40, 689.73, 271.28, 334.12, 233.81,
    550.71, 288.55, 282.84, 401.39, 564.56, 338.65, 202.72,
    516.21, 344.79, 461.61, 369.07, 204.44, 447.95, 328.90,
    258.78, 273.84, 432.46),
x4=c(513.34, 302.87, 285.32, 208.57, 192.17, 279.81, 218.36,
    277.11, 346.93, 211.92, 435.69, 126.33, 160.77, 107.90,
    219.79, 208.14, 201.01, 206.06, 356.27, 157.78, 171.79,
    236.55, 203.21, 153.32, 249.54, 209.11, 259.51, 219.86,
    303.00, 317.32, 235.82),
x5=c(467.87, 284.19, 272.95, 201.50, 249.81, 239.18, 220.69,
    224.65, 527.00, 302.09, 514.66, 250.56, 405.14, 209.70,
    272.59, 217.00, 237.60, 321.29, 811.88, 329.06, 329.65,
    403.92, 240.24, 254.66, 290.84, 379.30, 230.61, 206.65,
    244.93, 251.08, 250.28),
x6=c(1141.82, 735.97, 540.58, 414.72, 463.09, 445.20, 459.62,
    376.82, 1034.98, 585.23, 795.87, 513.18, 461.67, 393.99,
    599.43, 337.76, 617.74, 697.22, 873.06, 621.74, 477.17,
    730.05, 575.10, 445.59, 561.91, 371.04, 490.90, 449.69,
    479.53, 424.75, 541.30),
x7=c(478.42, 570.84, 364.91, 281.84, 287.87, 330.24, 360.48,
    317.61, 720.33, 429.77, 575.76, 314.00, 535.13, 509.39,
    371.62, 421.31, 523.52, 492.60, 1082.82, 587.02, 312.93,
    438.41, 430.36, 346.11, 407.70, 269.59, 469.10, 249.66,
    288.56, 228.73, 344.85),
x8=c(457.64, 305.08, 188.63, 212.10, 192.96, 163.86, 147.76,
    152.85, 462.03, 252.54, 323.36, 151.39, 232.29, 160.12,
    211.84, 165.32, 182.52, 226.45, 420.81, 218.27, 279.19,
    225.80, 223.46, 191.48, 330.95, 389.33, 191.34, 228.19,
    236.51, 195.93, 214.40)
)

# 距离矩阵
> d<-dist(scale(X))
> h<-hclust(d)
# 绘图
> plclust(h)
# 31 个样本分为 5 个大类
> r<-rect.hclust(h,5)
```

91.3 类个数的确定

分成多少个类是合适的? 这个问题没有统一的答案. 一般根据需要确定. 基本方法大概是

1. 给定你认为的距离的最小阈值
2. 观测散点图, 查看类大概的个数
3. 使用某种统计量确定类的个数
4. 根据初步分类的图再次分类确定个数

Bemirman (1972) 提出了根据研究目的来确定的分类方法, 并提出了根据谱系图来分析的准则

- 各类重心距离必须较大
- 各类包含的元素不要太多
- 类的个数需符合使用目的
- 多采用几种方法, 应该结果差不多

rect.hclust() 函数绘出指定类的框.

```
# 需先 plot, plclust() 也可以
> plot(h1)
# 同时在图上绘出框
> re<-rect.hclust(h1, k=3)
> re
[[1]]
[1] 1 2

[[2]]
[1] 5

[[3]]
[1] 3 4
```

91.4 k-均值动态聚类

系统聚类需要计算距离矩阵, 当样本很多时, 需占用很多内存和计算时间. 基于此, 产生了动态聚类方法.

动态聚类的基本思想是, 开始粗略的分一下, 然后按照某种最优原则修改不合理的分类, 直到分类比较合理为止. 此方法计算量较小, 内存较小, 方法简单, 适用于大样本.

算法: 任何多元分析教科书均有

91.4.1 k means 算法

参考 http://en.wikipedia.org/wiki/K-means_algorithm

k-means 算法最早由 MacQueen 1967 年提出, 后来经许多人多次修改.

k 个聚类具有以下特点: 各聚类本身尽可能的紧凑, 而各聚类之间尽可能的分开.

用途:

1. 资料压缩: 以少数的资料点来代表大量的资料, 达到资料压缩的功能
2. 资料分类: 以少数代表点来代表特点类别的资料, 可以降低资料量及计算量

k-means 算法的工作过程说明如下:

1. 从 c 个数据对象任意选择 k 个对象作为初始聚类中心
2. 循环 (3)到 (4) 直到每个聚类不再发生变化为止(收敛)
3. 根据每个聚类对象的均值(中心对象), 计算每个对象与这些中心对象的距离; 并根据最小距离重新对相应对象进行划分

4. 重新计算每个(有变化)聚类的均值(中心对象)

一般都采用均方差作为标准测度函数. 下面是伪代码

```
var m = initialCentroids(x, K);
var N = x.length;
while (!stoppingCriteria) {
  var w = [][];
  // calculate membership in clusters
  for (var n = 1; n <= N; n++) {
    v = arg min (v0) dist(m[v0], x[n]);
    w[v].push(n);
}
  // recompute the centroids
  for (var k = 1; k <= K; k++) {
    m[k] = avg(x in w[k]);
}
}
return m;
```

91.4.2 k-means++方法

2006 年提出了一个关于初始值的选择的新的改进, 叫做"k-means++".[1] 基本思想是选择尽量接近大的数量的点的作为初始点, 然后开始聚类. 作者使用 L^2 范数来选择聚类中心. 虽然初始直到计算量比较大, 但是对减少误差很有帮助, 而且后来的聚类过程会很快, 从而整个计算过程会加快 2-10 倍. 大的数据量会减少 1000 倍的误差. 几乎与 vanilla k-means 方法的速度和误差一样.

kmeans() 函数使用 k-均值方法, 采用逐个修改方法, 方法有"Hartigan-Wong", "Lloyd", "Forgy", "MacQueen" 三种, 具体见帮助.

[1] D. Arthur, S. Vassilvitskii: "k-means++ The Advantages of Careful Seeding" 2007 Symposium on Discrete Algorithms (SODA).

下面对 31 个省级地区居民的 8 项基本消费支出使用 k-均值方法分类. 帮助里有绘图的例子

```
> km <- kmeans(scale(X), 5, nstart = 20); km
K-means clustering with 5 clusters of sizes 16, 10, 1, 3, 1

Cluster means:
        x1         x2        x3        x4         x5        x6
1 -0.7008593 -0.33291790 -0.5450901 -0.2500165 -0.54749319 -0.6131804
2  0.2646918  0.04585518  0.2487958 -0.3405821 -0.01812541  0.2587437
3  1.1255255  2.91079330 -1.0645632 -0.4082114  0.53291392 -1.0476079
4  1.8790347  1.02836873  2.1203833  2.1727806  1.49972764  2.2232050
5  1.8042004 -1.12776493  0.9368961  1.2959544  3.90904835  1.6014419
        x7         x8
1 -0.5420723 -0.57966702
2  0.2874133 -0.02413414
3 -0.9562089  1.66126641
4  0.9583064  1.94532737
5  3.8803141  2.01876530

Clustering vector:
 [1] 4 2 1 1 1 1 1 1 4 2 4 1 2 1 2 1 2 2 5 2 1 2 2 1 2 3 1 1 1 1 1

Within cluster sum of squares by cluster:
[1] 30.14432 22.12662  0.00000 10.19134  0.00000

Available components:
[1] "cluster"  "centers"  "withinss" "size"
```

91.5 k 邻 近 法(K Nearest Neighbors, knn)算法

参考

Teknomo, Kardi. K-Nearest Neighbors Tutorial.
http://people.revoledu.com/kardi/tutorial/KNN/index.html

http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm

knn 是有监督的分类方法. 已经用于数据挖掘, 模式识别图像处理等方面. 成功的例子包括手写字体, 卫星图像和 EKG 模式识别.

91.5.1 knn 算法

knn 基于训练样本和新数据的属性分类. knn 不依赖于任何模型, 只依赖于记忆.

给出新的样本, 我们发现在它周围 k 个样本中属于某类最多的样本, 那么这些最多的样本的归类就是新样本的类别. 这实际上是一个它周围样本对新样本的投票过程.

下面是一个例子, 我们知道两个属性 $X_1 = (x_{1,1}, x_{2,1}, \cdots), X_2 = (x_{1,2}, x_{2,2}, \cdots)$ 的24个样本的分类 g, 然后对新样本$s_{25} = (x_{25,1} = 5, x_{25,2} = 6)$分类.

	x1	x2	g
s_1	4.00	3.00	1.00
s_2	1.00	3.00	1.00
s_3	3.00	3.00	1.00
s_4	3.00	7.00	1.00
...	...	...	...
s_{23}	7.00	4.00	2.00
s_{24}	8.00	8.00	2.00
s_{25}	5	6	???

```
 X=data.frame(
   x1=c(4,1,3,3,7,4,6,5,3,6,4,4,5,7,5,10,7,4,9,5,8,6,7,8),
   x2=c(3,3,3,7,4,1,5,6,7,2,6,4,8,8,6,5,6,10,7,4,5,6,4,8),
   g=c(rep(1,10),rep(2,14)) )
> X
  x1 x2 g
1  4  3 1
```

```
2   1  3 1
3   3  3 1
4   3  7 1
5   7  4 1
6   4  1 1
7   6  5 1
8   5  6 1
9   3  7 1
10  6  2 1
11  4  6 2
12  4  4 2
13  5  8 2
14  7  8 2
15  5  6 2
16 10  5 2
17  7  6 2
18  4 10 2
19  9  7 2
20  5  4 2
21  8  5 2
22  6  6 2
23  7  4 2
24  8  8 2

# 下面画图看看
# red 为类1, blue 为类2.
> plot(x2~x1,col=c("red","blue")[g],data=X)
# 新样本(5,6)为 "*" 标记
> points(5,6,pch=8,cex=3)
```

已知的样本为训练样本. 下面是步骤

- 确定 k 值
- 计算新样本与所有训练样本的距离
- 排序计算出的距离
- 收集前 k 个最小距离和它们属的类别

- 判别新样本的类别

假设 $k = 8$, 我们使用 8 个最邻近点来确定新样本的分类. 首先计算新样本 s_{25} 与所有 24 个训练样本的距离(此处使用 Euclidean 距离), 然后从小到大排序, 取前 8 个最小值, 查看这 8 个训练样本的分类, 其中类 1 有 3 个, 类 2 有 5 个. 我们判断新样本的分类为类 2.

```
# 两个向量 euclidean 距离
dist.euclidean <- function(x,y){
   res <- sqrt(sum((x-y)^2))
   res
}

# 计算训练样本与新样本的距离
> s=data.frame(x1=X$x1,x2=X$x2)
> apply(s,1,dist.euclidean,y=c(6,5))
 [1] 2.828427 5.385165 3.605551 3.605551 1.414214 4.472136 0.000000 1.414214
 [9] 3.605551 3.000000 2.236068 2.236068 3.162278 3.162278 1.414214 4.000000
[17] 1.414214 5.385165 3.605551 1.414214 2.000000 1.000000 1.414214 3.605551

# 实际上距离的平方更好看一些, 计算也容易(但是我们不准
备编写新的函数了)
> apply(s,1,dist.euclidean,y=c(6,5))^2
 [1]  8 29 13 13  2 20  0  2 13  9  5  5 10 10  2 16  2 29 13  2  4  1  2 13

# [排序并查看类别
> d <- cbind(apply(s,1,dist.euclidean,y=c(6,5))^2, X$g)
> o<-order(d[,1])
> d1<-d[o,]
> d1
     [,1] [,2]
 [1,]    0    1
 [2,]    1    2
 [3,]    2    1
 [4,]    2    1
 [5,]    2    2
 [6,]    2    2
 [7,]    2    2
 [8,]    2    2
```

```
 [9,]    4    2
[10,]    5    2
[11,]    5    2
[12,]    8    1
[13,]    9    1
[14,]   10    2
[15,]   10    2
[16,]   13    1
[17,]   13    1
[18,]   13    1
[19,]   13    2
[20,]   13    2
[21,]   16    2
[22,]   20    1
[23,]   29    1
[24,]   29    2
```

91.5.2 预测

使用 knn 来预测(extrapolation, 外推)

X Y 是按照时间排列的数值型数据. 第六个 $X = 6.5$, 我们要预测 Y 的值.

	X	Y
1	1.00	23.00
2	1.20	17.00
3	3.20	12.00
4	4.00	27.00
5	5.10	8.00
6	6.5	???

- 首先确定 $k = 2$
- 计算新样本 6.5 与其它 X 的距离

	X	Y	6.5 与 X 的距离	标记最近的 2 个 Y
1	1.00	23.00	5.5	
2	1.20	17.00	5.3	
3	3.20	12.00	3.3	
4	4.00	27.00	2.5	y
5	5.10	8.00	1.4	y
6	6.5	???		

- 计算最近的 2 个 Y 值的平均 $\frac{27+8}{2} = 17.5$, 即是 $X = 6.5$ 的预测值

91.5.3 平滑

使用同样的方法可以平滑(interpolation, 内插), 只要取 $X \in [1, 5]$, 然后计算每个 X 值的预测, 即可得到平滑.

```
d<-data.frame(
   X=c(1,1.2,3.2,4,5.1),
   Y=c(23,17,12,27,8) )
> d
   X  Y
1 1.0 23
2 1.2 17
3 3.2 12
4 4.0 27
5 5.1  8

> attach(d)
# 我们想从 0 到 6 平滑 x
> x<-seq(0,6,by=0.5)
> x
 [1] 0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0

# 先用 0.1 做例子
> a=0.1
# abs(X-a) 为距离,
# 取 k=2, 即前两个距离最小的对应的 Y 值
```

```
> Y[order(abs(X-a))[1:2]]
[1] 23 17
# 预测值为对应 Y 值的平均
> pre<-mean(Y[order(abs(X-a))[1:2]]); pre
[1] 20

# 编写内插/平滑(预测)函数
# A 相当于 X, B 相当于 Y
> pre<-function(A,B,x,k){
   p<-mean(B[order(abs(A-x))[1:k]]);
   p
}
> x<-seq(0,6,by=0.5)
> x
 [1] 0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0
# 对 x 中每个值预测(内插)
> apply(X=as.matrix(x),1,pre,A=X,B=Y,k=2)
 [1] 20.0 20.0 20.0 20.0 20.0 14.5 19.5 19.5 19.5 17.5 17.5 17.5 17.5
```

91.5.4 优点与缺点

优点

- 对噪音不敏感(robust), 尤其使用加权距离
- 若训练数据很大, 算法是比较有效的

缺点

- 需要确定 k 值
- 距离的种类难于确定
- 计算量比较大, 因为要计算所有训练数据与新数据的距离. 使用例如 K-D tree 等其它数据结构可能会好些

91.5.5 knn() 函数用法

class 包的 knn() 函数用于通常的分类, 使用 Euclidean 距离, 判别方法为投票法. 如果有个数一样的类, 则随机选择一个.

对于数值型数据及其平均(此处的预测与平均)并不能实现.

```
# 使用内插的例子数据
X=c(1,1.2,3.2,4,5.1),
Y=c(23,17,12,27,8)
> library(class)
> knn(train=X,test=as.matrix(seq(0,6,by=0.5)),cl=Y,k=2)
 [1] 17 17 17 17 17 17 12 27 27 8  8  8  8
Levels: 8 12 17 23 27

# 下面使用算法中的数据
 X=data.frame(
   x1=c(4,1,3,3,7,4,6,5,3,6,4,4,5,7,5,10,7,4,9,5,8,6,7,8),
   x2=c(3,3,3,7,4,1,5,6,7,2,6,4,8,8,6,5,6,10,7,4,5,6,4,8),
   g=c(rep(1,10),rep(2,14)) )

# 新数据
> test<-matrix(rnorm(20)*10,ncol=2)
> test
           [,1]       [,2]
 [1,]  14.497537  26.6676358
 [2,] -19.116549   1.9244647
 [3,]  -3.845555  -6.2873504
 [4,]  -6.286773 -11.1151424
 [5,]  -9.735719  -4.0103464
 [6,]   2.039382   5.3554126
 [7,] -14.485949  14.5503272
 [8,] -19.374097   9.2758593
 [9,]   8.656341  -0.1229066
[10,] -17.240399  -5.9442139

> cl <- X[,3]
> train<-X[,1:2]
> knn(train,test,cl,k=8)
 [1] 2 1 1 1 1 1 1 1 2 1
```

Chapter 92

核方法概要与支持向量机

核方法部分参考文献 [9] chapter 2

支持向量机部分参考文献 [9] chapter 7.2 7.3

92.1 原始线性回归(线性插值)

92.1.1 描述

给定训练集合$S = ((x_1, y_1), \cdots, (x_l, y_l))$, 其中$x_i$来自$R^n$维空间, 是个n维向量. 对应标签$y_i$来自$R^n$维空间, 也是个n维向量. 试图寻找齐次线性实值函数

$$g(x) =< w, x >= w'x = \sum_{i=1}^{n} w_i x_i$$

使得其为S的最优插值.

此关系建立了下面应该近似为0的模式函数

$$\xi f((x, y)) = |y - g(x)| = |y - < w, x > | \approx 0$$

这个任务也叫做线性插值(linear interpolation)

92.1.2 精确的情况

精确的情况下, 数据应该是以$(x, g(x))$的形式出现(且$l = n$), 通过解线性方程组

$$Xw = g(x)$$

92.1.3 存在误差

如果点数少于维数, 即$l < n$, 那么就存在许多可能的w, 都可以准确的描述数据. 在这些w之间做出选择需要一个标准. 我们倾向于选择范数最小的向量w.

如果点数多于维数且数据存在噪声, 那么不存在精确的模式, 即前面说的≈ 0, 或线性方程组的等号不成立, 变成≈ 0. 这种情况下, 我们希望选择具有最小误差的模式.

一般, 如果是小的数据集, 并且有噪声, 那么需要将这两个策略结合, 即求w使得范数较小, 误差也较小

(这里对噪声做一些解释, 除了普通的噪声外, 真实的模型不知道, 且模式比较复杂的时候, 即使数据没有噪声, 我们也很难理解这些数据. 如果我们给出的模式不完全适合数据, 此时我们也会把这种不准确看作噪声, 实际上可能不是噪声, 而是我们的模型有问题.)

92.1.4 最小二乘逼近

我们想找到度量误差的函数, 并且求得误差比较小的情况. 记误差为

$$|\xi| = f((x, y)) = |y - g(x)| = |y - < w, x >| \approx 0$$

误差平方和是最常用的度量尺度, 来度量训练数据与特定函数(预测, 分类等)之间的总体偏差. 这个特定函数表示为

$$L(g, S) = L(w, S) = \sum_{i=1}^{l} (y_i - g(x_i))^2 = \sum_{i=1}^{l} \xi_i^2 = \sum_{i=1}^{l} L(g, (x_i, y_i))$$

这里我们使用$L(g,(x_i,y_i))=\xi_i^2$表示第i个样本的平方误差, 或者损失. $L(g,S)$, 或 $L(w,S)$表示总体损失.

此时, 学习问题变为选择向量w使得总体损失最小. 人们深入研究了这个问题, 并把它应用到几乎每个学科领域. 它由高斯(Gauss)引入, 被称为最小二乘逼近(least squares approximation).

这样, 误差可以写作

$$\xi = y - Xw$$

从而总体损失函数写为

$$L(w,S) = ||\xi||_2^2 = (y-Xw)'(y-Xw)$$

92.1.5 正态方程

对损失函数求w的偏导, 并令其为0向量,

$$\frac{\partial L(w,S)}{\partial w} = -2X'y + 2X'Xw = 0$$

我们可以找到最优的w, 从而得到所谓"正态方程"(normal equation)

$$X'Xw = X'y$$

若$X'X$的逆存在, 可以把最小二乘法的解表示为

$$w = (X'X)^{-1}X'y$$

因此, 为了使线性差值的平方损失最小, 需要保留和维数一样多的参数$((X'X)^{-1}_{n*n}, X'y_{n*n})$, 解$n*n$的线性方程组, 计算代价为$O(n^3)$, 意思是运算次数$t(n)$的界

$$t(n) <= Cn^3$$

C为某个常数.

92.1.6 预测

在新的数据点上预测输出, 可以使用下式计算

$$g(x) =< w, x >$$

92.1.7 对偶表示

若$X'X$的逆存在, 我们可以把w表示为

$$w = (X'X)^{-1}X'y = X'X(X'X)^{-2}X'y = X'[X(X'X)^{-2}X'y] = X'\alpha$$

即

$$w = \sum_{i=1}^{l} \alpha_i x_i$$

92.1.8 伪逆

若$X'X$是奇异矩阵, 则可以使用伪逆, 这就找到了满足正态方程的具有最小范数的w. 另一方面, 我们可以在范数大小和损失之间做出权衡. 这种方法被称为岭回归.

92.2 岭回归(ridge regression)

(另外参考维基 http://en.wikipedia.org/wiki/Ridge_regression, 其中的A即此处的X, b即此处的y, Γ即此处的λI_n, 或Γ可以为对角元素不同的对角矩阵, 这样样本每列的权重就不同. 另外还有Bayesian解释, 及其它广义正则化)

92.2.1 原始解法(primal solution)

有些情况下, 可能无法准确拟合数据, 原因或者是没有足够的数据保证$X'X$可逆, 或者是数据中存在噪声. 这时, 不可能准确匹配目标输出. 这种问题被称为不适定的(ill-posed)问题. 在这种情况下, 人们常常用某种方法限制函数的选择, 这种限制或者是偏置, 称为正则化(regularisation).

最简单的正则化是采用范数比较小的函数. 对于最小二乘回归情况, 这给出著名的岭回归最优化标准.

$$minL_\lambda(w,S) = min[\lambda||w||^2 + \sum_{i=1}^{l}(y_i - g(x_i))^2]$$

其中λ是一个正数, 定义范数和损失之间的相对权衡, 从而控制正则化程度.

再次取关于w的导数, 得到方程

$$X'Xw + \lambda w = (X'X + \lambda I_n)w = X'y$$

这种情况下, 如果$\lambda > 0$, 矩阵$(X'X + \lambda I_n)$总是可逆的. 故该方程的解由

$$w = (X'X + \lambda I_n)^{-1}X'y$$

给出. 这项任务的复杂度为$O(n^3)$.

得到的预测函数为

$$g(x) =< w, x >= y'X(X'X + \lambda I_n)^{-1}x$$

92.2.2 岭迹图确定λ

目前还没有一个完美的方法确定λ. 在实际中, 我们首先以λ为自变量, w为因变量, 绘制出若干曲线, 当曲线开始平稳, 对应的λ就是我们需要的. 绘制的图就是"岭迹图".

92.2.3 例子

MASS包的lm.ridge()函数计算岭回归. 下面是在线的例子.

```
> data(longley) # not the same as the S-PLUS dataset
> longley
       y    GNP Unemployed Armed.Forces Population Year Employed
1947  83.0 234.289     235.6       159.0    107.608 1947   60.323
1948  88.5 259.426     232.5       145.6    108.632 1948   61.122
1949  88.2 258.054     368.2       161.6    109.773 1949   60.171
...

> names(longley)[1] <- "y"
> lm.ridge(y ~ ., longley) # 和普通回归一样, lambda=0
                    GNP   Unemployed  Armed.Forces   Population
2946.85636017   0.26352725   0.03648291   0.01116105  -1.73702984
       Year     Employed
  -1.41879853   0.23128785
# 计算每个lambda时的回归系数, 并绘岭迹图
> plot(lm.ridge(y ~ ., longley, lambda = seq(0,0.1,0.001)))
# 选择好的lambda
> select(lm.ridge(y ~ ., longley, lambda = seq(0,0.1,0.0001)))
modified HKB estimator is 0.006836982
modified L-W estimator is 0.05267247
smallest value of GCV  at 0.0057
```

92.2.4 对偶方法(dual solution)

上面的原始解法为

$$X'Xw + \lambda w = (X'X + \lambda I_n)w = X'y$$

w的解由

$$w = (X'X + \lambda I_n)^{-1}X'y$$

给出. 另外, 我们可以根据w重写第一个方程为

$$X'Xw + \lambda I_n w = X'y$$

得到

$$w = \lambda^{-1}X'(y - Xw) = X'\alpha$$

其中$\alpha = \lambda^{-1}(y - Xw)$

再次表明, w可以写作训练点的线性组合, 即

$$w = \sum_{i=1}^{l} \alpha_i x_i$$

因此我们有

$$\begin{aligned} \alpha &= \lambda^{-1}(y - Xw) \\ \Rightarrow \quad \lambda\alpha &= (y - XX'\alpha) \\ \Rightarrow \quad (XX' + \lambda I_l)\alpha &= y \\ \Rightarrow \quad \alpha &= (G + \lambda I_l)^{-1}y \end{aligned}$$

其中$G = XX'$, 即$G_{i,j} =< x_i, x_j >$. 得到

$$w = \sum_{i=1}^{l} \alpha_i x_i = X'\alpha = X'(G + \lambda I_l)^{-1}y$$

设新数据为x, 原来的数据为$x_i, i = 1, \cdots, l$, 预测函数为

$$g(x) =< w, x >=< \sum_{i=1}^{l} \alpha_i x_i, x >= \sum_{i=1}^{l} \alpha_i < x_i, x >= y'(G + \lambda I_l)^{-1}k$$

其中$k_i =< x_i, x >$

对偶解法的一个很重要的事实是, 来自训练集的信息由矩阵$G = XX'$, 即训练点之间的内积给出. 关于预测, 只是训练集和新数据之间的内积.

矩阵G被称为Gram矩阵(Gram matrix). Gram矩阵和$G + \lambda I_l$都是$l * l$维的. 若$l < n$, 那么对偶解法的复杂度小于原始解法.

92.3 核定义的非线性特征映射

92.3.1 特征映射

考虑把x嵌入映射, 即对x做一个变换, 线性或非线性, 原来的x属于空间R^n, 映射后的空间为R^N, 记映射为

$$\phi : x \in R^n \longmapsto \phi(x) \in F \subseteq R^N$$

映射后的训练集编码为

$$S = (\phi(x_1), y_1), \cdots, (\phi(x_l), y_l)$$

在此训练集上寻找回归关系, 如果N非常大$N > l$, 那么会出现对偶解法的复杂度小于原始解法的情况.

现在Gram矩阵为

$$G_{ij} =< \phi(x_i), \phi(x_j) >$$

对于新数据x, 向量k为

$$k_i =< \phi(x_i), x >, \quad i = 1, \cdots, l$$

此时计算向量α的复杂度为

$$O(l^3 + l^2 N)$$

计算新例子的复杂度为

$$O(lN)$$

92.3.2 核函数/有效核函数

有时候, 直接计算内积比显式的计算映射ϕ更加高效. 即可以跳过计算特征映射. 我们把直接计算内积的函数称为核函数, 即核函数κ满足

$$\kappa(x, z) =< \phi(x), \phi(z) >$$

我们称满足一个特征空间内积的函数为有效核函数. 即倘若可以高效的计算函数, 并且这个函数相当于计算它的两个自变量的合适映射的内积, 它就构成一个潜在有用的核.

否则, 如果定义了一个函数, 但是不能表示任何空间的内积, 这个函数就是非有效的核函数.

因此, 核函数并不能唯一确定映射后的内积. 下面我们将会看到.

92.3.3 核函数与特征映射非一一对应

例如, 考虑二维输入(训练集)空间$X \subseteq R^2$, 同时有特征映射

$$\phi : x = (x_1, x_2) \longmapsto \phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1, x_2) \in F = R^3$$

即此映射把二维数据映射到三维空间. 将特征映射与内积结合, 可以计算如下

$$\begin{aligned} < \phi(x), \phi(z) > &= < (x_1^2, x_2^2, \sqrt{2}x_1, x_2), (z_1^2, z_2^2, \sqrt{2}z_1, z_2) > \\ &= x_1^2 z_1^2 + x_2^2 z_2^2 + 2x_1 x_2 z_1 z_2 \\ &= (x_1 z_1 + x_2 z_2)^2 \\ &= < x, z >^2 \end{aligned}$$

因此, 核函数为

$$\kappa(x, z) =< x, z >^2$$

这意味着我们可以计算两个点在特征空间的投影的内积而不用显式的求出它们的坐标(即映射). 注意到, 同一个核可以对于四维特征映射的内积

$$\phi : x = (x_1, x_2) \longmapsto \phi(x) = (x_1^2, x_2^2, x_1, x_2, x_1, x_2) \in F = R^4$$

92.3.4 核函数如何改进特征空间内积的计算效率

实际上在上面的描述中我们已经看到核函数如何改进特征空间内积的计算效率. 此处再举一个例子, 仿照四维特征映射的例子, 考虑n维空间, 核函数

$$\kappa(x, z) =< x, z >^2$$

可以表示一个n^2维的特征映射的内积. 可以设特征映射为

$$\phi : x \longmapsto \phi(x) = (x_i x_j)_{i,j=1}^{n} \in F = R^{n^2}$$

内积

$$\begin{aligned} < \phi(x), \phi(z) > &= < (x_i x_j)_{i,j=1}^{n}, (z_i z_j)_{i,j=1}^{n} \\ &= \sum_{i,j=1}^{n} x_i x_j z_i z_j = \sum_{i=1}^{n} x_i z_i \sum_{j=1}^{n} x_j z_j \\ &= < x, z >^2 \end{aligned}$$

92.3.5 核的选择

可以证明, 从很严谨的意义上讲, 核的选择相当于把关于数据和我们期望识别的模式类型的先验知识进行编码. 通过研究如何从生成数据的过程的概率模型推导核, 就可以探索这一关系.

不同的核函数可以构造不同类型的非线性决策面, 从而导致不同的支持向量算法. 实际问题中, 通常直接给出核函数. 常用的核函数有(kernlab 包有更多的核函数(7种))

1. 线性核函数(linear kernel)

$$k(x, x_i) = (x\dot{x}_i)$$

2. 多项式核函数(polynomial kernel)

$$k(x, x_i) = (s(x\dot{x}_i) + c)^d$$

其中s,c,d为参数. 线性核函数可以看作多项式核函数的特例

3. 径向基核函数(radical basis function, rbf)

$$k(x, x_i) = exp(-\gamma|x - x_i|^2)$$

其中s,c为参数.

4. Sigmoid 核函数(Sigmoid tanh)

$$k(x, x_i) = tanh(s(x\dot{x}_i) + c)$$

其中γ为参数.

5.高斯核

$$k(x, x_i) = exp(-|x - x_i|^2/(2\sigma^2))$$

其中s,c为参数.

92.3.6 结论

核函数可以高效的应用于高维特征空间, 这是通过避免计算特征映射而做到的. 核方法一般表示的是非线性映射的内积, 故可以区分任何的模式, 但是一旦核函数确定, 其复杂度就确定了, 因此可以有效的控制其复杂度, 不像神经网络或其它的方法常常会出现过拟合的情况.

此处有一个小小的问题是核函数不能唯一确定特征映射.

92.3.7 核模式分析的过程

- 指定核函数,
- 先使用训练数据构造核矩阵,
- 然后用模式分析算法处理核矩阵, 得到一个模式函数,
- 再使用这个模式函数处理新的数据以获得预测.

此处的模式分析算法包括(不全):

- Fisher判别,
- 主成分分析,
- 典型相关分析,

- 线性回归(核偏最小二乘法),
- 用于分类的支持向量机,
- 用于回归的支持向量机,
- 在线分类和回归,
- 排列, 聚类和数据可视化,
- 其它任何可以使用核函数的模式分析方法

92.4 TODO: 新颖性检测

新颖性检测十分类似异常值检测.

svm方法通过建立一个超球体包含大部分的目标点, 并且给定损失函数来评估异常检测的结果.

92.4.1 最小封闭超球体

92.4.2 新颖性检测的稳定性

92.4.3 包含大部分点的超球体

92.5 用于分类的支持向量机

92.5.1 硬间隔(最大间隔)分类器

对于两个线性可分离类, 存在几何间隔(两个超平面构成的)将其分离. 我们的任务是寻找使几何间隔最大化的线性函数. 这个函数经常称为最大间隔超平面(maximal margin hyperplane)或者硬间隔支持向量机(hard margin support vector machine).

给定训练集合$S = ((x_1, y_1), \cdots, (x_l, y_l))$, 其中$x_i$来自$R^n$维空间, 是个n维向量. 对应标签$y_i$来自$R^n$维空间, 也是个n维向量. 存在一个由权向量w和阈值b确定的范数为1的线性函数

$$g(x) =< w, \phi(x_i) > +b$$

并且存在$\gamma > 0$, 使得

$$\xi_i = (\gamma - y_i g(x_i)) = 0, \quad i = 1, \cdots, l$$

表明训练集S的间隔满足

$$m(S, g) = min_{1<=i<=l} y_i g(x_i) >= \gamma$$

表明两个类可以用间隔为γ的超平面分开.

硬间隔SVM: 把选择超平面转化为解下面的最优化问题

$$\begin{array}{rcl} max_{w,b} & : & \gamma \\ 约束条件为 & : & y_i(< w, \phi(x_i) > +b) >= \gamma, \quad i = 1, \cdots, l \\ & & ||w||^2 = 1 \end{array}$$

TODO: 对偶形式, 把最优化视为最小化$-\gamma$, 引入拉格朗日乘子....

92.5.2 软间隔分类器(C,v-SVM分类)

硬间隔分类器是一个很重要的概念. 但是只能用于数据线性可分的情况. 否则我们会得到非常复杂的核, 从而导致过度拟合.

我们放松对ξ的要求, 即允许$\xi > 0$的情况存在, 这时间隔与ξ的一范数组合最优化. 我们经常称ξ为间隔松弛变量(margin slack vector), 因为它允许间隔约束条件之间存在冲突, 即有些类可以错分.

一范数软间隔SVM:

$$max_{w,b,\gamma,\xi} \quad : \quad -\gamma + C\sum_{i=1}^{l}\xi_i$$

$$\text{约束条件为} \quad : \quad y_i(<w,\phi(x_i)>+b) >= \gamma - \xi_i, \quad \xi_i >= 0, i = 1,\cdots,l$$
$$||w||^2 = 1$$

参数C控制间隔与松弛变量大小之间的权衡. 有可能使用ξ的2-范数取代1-范数.

实际中, 参数C在一个很广的范围内变动. 随着C的变化, 间隔γ沿着对应的区域平稳变化. 所以, 对于一个给定的问题, 选择具体的C相当于选择γ, 然后把$||\xi||_1$最小化.

参数C的意义并不直观. 但是与新颖性检测一样, 也就是$C >= 1/l$, 可以利用

$$C = 1/(vl)$$

其中$v \in (0,1]$, 这导致对异常值个数的控制.

这种形式的支持向量机被称为 v-支持向量机, 或者新支持向量机.

92.6 用于回归的支持向量机

我们选择平方损失函数

$$f(z) = f(x,y) = (y - g(x))^2$$

作为评估算法的输出与理想输出的偏差. (另一种评估方法是使用相对互信息)

92.6.1 TODO: ε-不敏感回归和v-svm回归

ε-不敏感损失函数为

$$L(x,y,g) = |y - g(x)|_\varepsilon = max(0, |y - g(x)| - \varepsilon)$$

即只要误差小于ε, 就记为0. 这相当于在目标点的附近ε建立一个带, 对误差强制为0, 称为不敏感带.

92.7 R的svm()函数

参考文献 [49]

92.7.1 libsvm介绍与特性

libsvm是台湾大学林智仁(Lin Chih-Jen)副教授等开发设计的一个简单、易于使用和快速有效的SVM模式识别与回归的软件包, 他不但提供了编译好的可在Windows系列系统的执行文件, 还提供了源代码, 方便改进、修改以及在其它操作系统上应用; 该软件还有一个特点, 就是对SVM所涉及的参数调节相对比较少, 提供了很多的默认参数, 利用这些默认参数就可以解决很多问题.

特性:

C- and ν-classification: C,v分类

one-class-classification (novelty detection): 新颖性检测

ε- and ν-regression: ε- and ν-回归

包括:

linear, polynomial, radial basis function, and sigmoidal kernels

formula interface

k-fold cross validation(k-折交叉验证)

另外两个概念:

Multi-class classification: 理论上, SVM只能用于2-分类问题.

libsvm 使用 one-against-one(一对一)技术, 借助多数投票机制进行多类别分类. 一对一技术为任意两个类构建超平面, 共需训练$k*(k-1)/2$个2值svm分类器. 这种情况下, 对k个分类的训练集进行两两区分. 最后常常使用投票法, 得票最多的(max wins)类为样本所在类. 缺点显而易见, 当类别数增加时, 要训练的分类器数目迅速增加(平方), 导致检测时速度很慢.

92.7.2 用法

R 的 libsvm 接口在 e1071 包内, 函数 svm()

如果y是因子, 执行分类任务, 否则执行回归任务. 如果忽略y, 执行新颖性检测

用法为

```
## S3 method for class 'formula':
    svm(formula, data = NULL, ..., subset, na.action =
    na.omit, scale = TRUE)
## Default S3 method:
    svm(x, y = NULL, scale = TRUE, type = NULL, kernel =
    "radial", degree = 3, gamma = if (is.vector(x)) 1 else 1 / ncol(x),
    coef0 = 0, cost = 1, nu = 0.5,
    class.weights = NULL, cachesize = 40, tolerance = 0.001, epsilon = 0.1,
    shrinking = TRUE, cross = 0, probability = FALSE, fitted = TRUE,
    ..., subset, na.action = na.omit)
```

type=可以是如下分类或回归. 亦可以由y来自行判断.

- 'C-classification'
- 'nu-classification'
- 'one-classification' (for novelty detection)
- 'eps-regression'
- 'nu-regression'

kernel: 核函数

linear: u'*v

polynomial: (gamma*u'*v + coef0)^degree

radial basis: exp(-gamma*|u-v|^2)

sigmoid: tanh(gamma*u'*v + coef0)

其它参数:
gamma: 除linear之外的其它核函数参数, 默认为 1/(data dimension)

详细参考在线帮助

下面例子来自参考文献 [49]. svm()与rpart()分类与回归树做了比较.

分类的例子

```
# 导入包
library(e1071)
library(rpart)
# 准备数据
data(Glass, package = "mlbench")
index <- 1:nrow(Glass)
testindex <- sample(index, trunc(length(index)/3))
testset <- Glass[testindex, ]
trainset <- Glass[-testindex, ]
# svm与rpart分类, 预测
svm.model <- svm(Type ~ ., data = trainset, cost = 100, gamma = 1)
svm.pred <- predict(svm.model, testset[, -10])

rpart.model <- rpart(Type ~ ., data = trainset)
rpart.pred <- predict(rpart.model, testset[, -10], type = "class")

# 查看分类结果与理论值的列联表
table(pred = svm.pred, true = testset[, 10])
table(pred = rpart.pred, true = testset[, 10])
```

```
# 对角线是正确的个数. 非对角线的是错分的个数
> table(pred = svm.pred, true = testset[, 10])
    true
pred  1  2  3  5  6  7
   1 13  6  2  0  0  0
   2  4 21  1  1  1  4
   3  5  2  2  0  0  0
   5  0  0  0  3  0  0
   6  0  0  0  0  0  0
   7  0  0  0  0  0  6
> table(pred = rpart.pred, true = testset[, 10])
    true
pred  1  2  3  5  6  7
   1 13 10  3  0  0  2
   2  9 18  2  0  0  0
   3  0  0  0  0  0  0
   5  0  1  0  4  1  0
   6  0  0  0  0  0  0
   7  0  0  0  0  0  8
```

下面是回归的例子

```
library(e1071)
library(rpart)
data(Ozone, package = "mlbench")
index <- 1:nrow(Ozone)
testindex <- sample(index, trunc(length(index)/3))
testset <- na.omit(Ozone[testindex, -3])
trainset <- na.omit(Ozone[-testindex, -3])

svm.model <- svm(V4 ~ ., data = trainset, cost = 1000, gamma = 1e-04)
svm.pred <- predict(svm.model, testset[, -3])
crossprod(svm.pred - testset[, 3])/length(testindex)

rpart.model <- rpart(V4 ~ ., data = trainset)
rpart.pred <- predict(rpart.model, testset[, -3])
crossprod(rpart.pred - testset[, 3])/length(testindex)
```

```
# 平均残差
> crossprod(svm.pred - testset[, 3])/length(testindex)
       [,1]
[1,] 8.137294

> crossprod(rpart.pred - testset[, 3])/length(testindex)
       [,1]
[1,] 15.08934
```

92.7.3 注意事项

- svm 可能对参数的选择很敏感. 因此建议总是测试一组参数的取值范围, 如果数据集很大的话, 至少对数据集的一部分
- 对于分类, 很可能你会选择 C-分类方法和RBF核函数, 因为它们的推广性能很好, 参数也比较少(只有两个: C, γ). libsvm 的作者建议首先测试大的和小的C值, 例如, 1-1000, 使用交叉验证来决定哪个比较好. 最后对选择的C值测试γ.
- 好的结果通常需要在整个参数空间内搜索才得到. 所以建议首先使用 tune.svm() 来调试
- 大的数据集训练时间会增长很快
- 对数据集缩放/归一化有时候会显著增加正确率, 所以svm()默认缩放数据

92.8 R的kernlab包

参考文献为 kernlab 包附带的文献《kernlab – An S4 Package for Kernel Methods in R》[39]

包括: dot product primitives (kernels), svm, relevance vector machine,高斯过程, 一个ranking算法, kernel PCA, kernel CCA, kernel feature analysis, online kernel methods, spectral clustering algorithm.

92.8.1 核函数

kernlab 包含7种核函数.

- linear vanilladot kernel: 最简单的核函数, 实际上就是普通的内积

$$k(x, x') =< x, x' >$$

- Gaussian radial basis function rbfdot: 一般意义的核函数, 经常在没有先验知识的时候使用

$$k(x, x') = exp(-\sigma||x - x'||^2)$$

- polynomial kernel polydot: 常常用于图像分类

$$k(x, x') = (scale\dot{<}x, x' > +offset)^{degree}$$

- hyperbolic tangent kernel tanhdot: 主要用于对神经网络的模拟

$$k(x, x') = tanh(scale\dot{<}x, x' > +offset)$$

- Bessel function of the first kind kernel besseldot: 一般意义的核函数, 经常在没有先验知识的时候使用, 主要在高斯过程社区流行

$$k(x, x') = \frac{Bessel^n_{(v+1)}(\sigma||x - x'||)}{(||x - x'||)^{-n(v+1)}}$$

- Laplace radial basis kernel laplacedot: 一般意义的核函数, 经常在没有先验知识的时候使用

$$k(x, x') = exp(-\sigma||x - x'||)$$

- ANOVA radial basis kernel anovadot: 在多元回归问题中表现良好

$$k(x, x') = (\sum_{k=1}^{n} exp(-\sigma(x^k - x'^k)^2)^d$$

x^k为第k个x元素

下面是例子

```
>  rbf <- rbfdot(sigma = 0.05)
>  rbf
Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.05
> x <- rnorm(10)
> y <- rnorm(10)
> rbf(x, y)
         [,1]
[1,] 0.2681125
```

92.8.2 核函数相关的方法

- kernelMatrix: 计算Gram矩阵. 用法为

  ```
  kernelMatrix(kernel, x, y = NULL)
  ```

  计算Gram矩阵$K_{ij} = k(x_i, x_j)$, x_i为X的列. 当y不为null时, 计算$K_{ij} = k(x_i, y_j)$.

  kernel是核函数, 可以自己定义, 只要接受两个向量并返回一个标量就可以.

- kernelFast:

  ```
  K = kernelFast(kernel, x1, x2, a)
  ```

  a为x1的范数的平方.

  对于rbfdot, besseldot, and the laplacedot核与kernelMatrix有区别, 这些都是RBF核. 需要第一个参数的范数的平方作为参数a, 在计算RBF核的时候, 重复调用kernelMatrix会出现溢出.

- kernelMult: 计算核展开式. 返回

$$f = (f(x_1), \cdots, f(x_m))$$

其中

$$f(x_i) = \sum_{j=1}^{m} k(x_i, x_j)\alpha_j, \ \ hence \ f = K\alpha$$

有时候gram矩阵K大于内存. 所以逐步计算. 参数blocksize决定逐步计算的行数(rows in the stripes).

```
f <- kernelMult(kernel, x, alpha)
```

计算

$$f_i = \sum_{j=1}^{m} k(x_i, x_j)\alpha_j$$

```
f <- kernelMult(kernel, x1, x2, alpha)
```

计算

$$f_i = \sum_{j=1}^{m} k(x1_i, x2_j)\alpha_j$$

- kernelPol: 非常类似函数kernelMatrix, 唯一的区别为计算

$$K_{ij} = y_i y_j k(x_i, x_j)$$

意味着

```
f <- kernelPol(kernel, x, y)
```

计算

$$K_{ij} = y_i y_j k(x_i, x_j)$$

x_i为x的列, y_i为向量y的元素.

```
f <- kernelPol(kernel, x1, x2, y1, y2)
```

计 算

$$K_{ij} = y1_i y2_j k(x1_i, x2_j)$$

x1,x2为 矩 阵, y1,y2为 向 量

下 面 是 例 子

```
> poly <- polydot(degree = 2)
> x <- matrix(rnorm(60), 6, 10)
> y <- matrix(rnorm(40), 4, 10)
> kx <- kernelMatrix(poly, x)
> kxy <- kernelMatrix(poly, x, y)
> kx
An object of class \kernelMatrix"
           [,1]      [,2]        [,3]        [,4]        [,5]      [,6]
[1,] 1.342487e+02 11.0388205  2.601413301 16.875461655 1.652125e-03   0.476504
[2,] 1.103882e+01 61.5710254  1.071795276  0.376449067 4.527327e+00   0.992037
[3,] 2.601413e+00  1.0717953 72.653642793  0.001752811 1.452268e+01  16.516585
[4,] 1.687546e+01  0.3764491  0.001752811 63.307511898 1.858641e+01   2.600207
[5,] 1.652125e-03  4.5273269 14.522679132 18.586407221 1.614202e+02   3.974663
[6,] 4.765040e-01  0.9920370 16.516585265  2.600207450 3.974663e+00 139.825836
> kxy
An object of class \kernelMatrix"
           [,1]      [,2]      [,3]       [,4]
[1,] 1.665327e-04  6.5463234  7.2422275 18.76852572
[2,] 1.243804e+01  0.5761263  5.9159828  4.67826648
[3,] 2.589982e+00  0.2339365  8.0780966  6.67319043
[4,] 8.328725e-02  2.0450171  0.4364127  0.02057895
[5,] 3.963183e+00  8.9034184 11.2254889  1.41871430
[6,] 1.417424e+01 64.1995024 34.5418277 10.21745485
```

92.8.3 核 方 法: svm

参 考 (http://www.cnblogs.com/zgw21cn/archive/2009/03/16/1413656.html, 作 者 zgw21cn)

包里函数ksvm()通过.Call接口, 使用bsvm和libsvm库中的优化方法, 得以实现svm算法. 对于分类, 有C-SVM分类算法和v-SVM分类算法, 同时还包括C分类器的有界约束的版本. 对于回归, 提供了ε-SVM回归算法和v-SVM回归算法. 对于多类分类, 有一对一(one-against-one)方法和原生多类分类方法.

下面是帮助中的部分例子

```
> library("kernlab")  #导入包

> data("iris")  #导入数据集iris

> irismodel <- ksvm(Species ~ ., data = iris,

+ type = "C-bsvc", kernel = "rbfdot",

+ kpar = list(sigma = 0.1), C = 10,

+ prob.model = TRUE) #训练
```

其中, type表示是用于分类还是回归, 还是检测, 取决于y是否是一个因子。缺省取C-svc或eps-svr。可取值有

- C-svc C classification
- nu-svc nu classification
- C-bsvc bound-constraint svm classification
- spoc-svc Crammer, Singer native multi-class
- kbb-svc Weston, Watkins native multi-class
- one-svc novelty detection
- eps-svr epsilon regression
- nu-svr nu regression
- eps-bsvr bound-constraint svm regression

Kernel设置核函数。可设核函数有

- rbfdot Radial Basis kernel "Gaussian"
- polydot Polynomial kernel
- vanilladot Linear kernel
- tanhdot Hyperbolic tangent kernel
- laplacedot Laplacian kernel
- besseldot Bessel kernel
- anovadot ANOVA RBF kernel
- splinedot Spline kernel
- stringdot String kernel

```
> irismodel
Support Vector Machine object of class "ksvm"
SV type: C-bsvc (classification)
parameter : cost C = 10
Gaussian Radial Basis kernel function.
Hyperparameter : sigma = 0.1
Number of Support Vectors : 32
Training error : 0.02
Probability model included.
>predict(irismodel, iris[c(3, 10, 56, 68, 107, 120), -5], type = "probabilities")
```

```
setosa     versicolor   virginica

[1,] 0.986432820 0.007359407 0.006207773

[2,] 0.983323813 0.010118992 0.006557195

[3,] 0.004852528 0.967555126 0.027592346

[4,] 0.009546823 0.988496724 0.001956452

[5,] 0.012767340 0.069496029 0.917736631

[6,] 0.011548176 0.150035384 0.838416441

# Ksvm支持自定义核函数。如

>k <- function(x, y) { (sum(x * y) + 1) * exp(0.001 * sum((x - y)^2)) }

> class(k) <- "kernel"

> data("promotergene")

> gene <- ksvm(Class ~ ., data = promotergene, kernel = k, C = 10, cross = 5)#训
练

> gene

Support Vector Machine object of class "ksvm"

SV type: C-svc (classification)

parameter : cost C = 10

Number of Support Vectors : 66

Training error : 0

Cross validation error : 0.141558

# 对于分类问题，可以对结果用plot()进行可视化。例子如下
> set.seed(123)
```

```
> x <- rbind(matrix(rnorm(120), , 2), matrix(rnorm(120,
+    mean = 3), , 2))
> y <- matrix(c(rep(1, 60), rep(-1, 60)))
> svp <- ksvm(x, y, type = "C-svc")
Using automatic sigma estimation (sigest) for RBF or laplace kernel
> plot(svp, data = x)
```

92.8.4 核方法: Relevance vector machine

Relevance vector machine(RVM) 是一个概率稀疏核模型, 与 svm 函数形式的预测一致

$$y(x) = \sum_{n=1}^{N} \alpha_n K(x, x_n) + a_0$$

α_n为权重.

它采用贝叶斯方法学习, 引入权重的先验为

$$p(\alpha, \beta) = \prod_{i=1}^{m} N(\beta_i|0, a_i^{-1}) Gamma(\beta_i|\beta_\beta, \alpha_\beta)$$

稀疏是因为很多权重的后验概率在0附近有尖锐的峰值. 非0权重不与决策边界的样本结合, 而与"prototypical"样本一起出现, 这些样本称为 relevance vectors.

kernlab 函数 rvm() 基于 II 型最大似然方法, 可以用于回归.

```
> x <- rbind(matrix(rnorm(120), , 2), matrix(rnorm(120,
+    mean = 3), , 2))
> y <- matrix(c(rep(1, 60), rep(-1, 60)))
>  rvmm <- rvm(x, y, kernel = "rbfdot", kpar = list(sigma = 0.1))
>  rvmm
Relevance Vector Machine object of class "rvm"
Problem type: regression
```

```
Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.1

Number of Relevance Vectors : 5
Variance :  0.06528009
Training error : 0.062690743

# 可以使用str(rvmm)查看
> str(rvmm)
Formal class 'rvm' [package "kernlab"] with 20 slots
  ..@ tol     : num 2.22e-16
  ..@ nvar    : num 0.0653
  ..@ mlike   : num 90.4
  ..@ RVindex : int [1:5] 10 44 92 94 104
  ..@ coef    : NULL
  ..@ nRV     : int 5
  ..@ alpha   : num [1:5] 1.81 1.523 0.779 -2.918 -1.019
  ..@ type    : chr "regression"
......

> ytest <- predict(rvmm, x)
> ytest
           [,1]
  [1,]  1.1255553
  [2,]  1.0993850
  [3,]  0.9690448
  [4,]  1.0627147
  [5,]  1.0610187
  [6,]  0.6401644
  [7,]  0.9392122
  [8,]  1.0585391
  [9,]  1.1007576
 [10,]  0.7673066
 ...
```

92.8.5 核方法: Gaussian processes

Gaussian processes (Williams and Rasmussen 1995) 基于对先验的假设: 邻近的观测应该互相之间有信息传递. 特别的, 它假设观测样本为正态分布, 它们之间的联合为正态分布协方差矩阵. 使用核矩阵作为协方差矩阵是一个简便的途径, 扩展贝叶斯线性预测模型到非线性模型. 进一步, 它对方法中 "kernel trick"的对应部分最小化正则风险.(it represents the counterpart of the "kernel trick" in methods minimizing the regularized risk)

对于回归预测, 我们观察不到 $t(x_i)$, 观察的是 $y_i = t(x_i) + \xi_i$. x的后验概率为

$$p(y|t) = [\prod_i p(y_i - t(x_i))] \frac{1}{\sqrt{(2\pi)^m det(K)}} exp(\frac{1}{2} t^T K^{-1} t)$$

替换$t = K\alpha$, 并取对数

$$\ln p(\alpha|y) = -\frac{1}{2\sigma^2}||y - K\alpha||^2 - \frac{1}{2}\alpha^T K\alpha + c$$

最大化$\ln p(\alpha|y)$, 对α, 我们得到最大后验近似

$$\alpha = (K + \sigma^2 1)^{-1} y$$

类似的, 高斯过程可以用于分类.

gausspr()函数实现此高斯过程.

92.8.6 核方法: Ranking

Google 的成功表明一个好的排序算法对于现实世界的用途. kernlab 包含一个排序算法, 基于 (Zhou, Weston, Gretton, Bousquet, and Sch¨¨lkopf, 2003). 此算法考虑数据的几何结构, 与使用欧氏距离或内积来比较, 它复杂一些. 实际数据很长是高度结构化的, 这个算法比简单的方法(欧氏距离或内积)工作的要好.

算法: 略

下面是一个例子

```
data(spirals)
ran <- spirals[rowSums(abs(spirals) < 0.55) == 2, ]
ranked <- ranking(ran, 54, kernel = "rbfdot", kpar = list(sigma = 100),
   edgegraph = TRUE)
ranked[54, 2] <- max(ranked[-54, 2])
c <- 1:86
op <- par(mfrow = c(1, 2), pty = "s")
plot(ran)
plot(ran, cex = c[ranked[, 3]]/40)
```

92.8.7 TODO: 核方法: Online learning with kernels

92.8.8 核方法: Spectral clustering

Spectral clustering (Ng, Jordan, and Weiss 2001) 是一个有希望作为普通聚类方法的替代的方法.

此方法使用由某些相似性测度产生的最大的矩阵特征向量来对数据聚类. 与排序(ranking)类似, 相似性矩阵由下式给出

$$K_{ij} = exp(-\sigma||x_i - x_j||^2$$

并且归一化为$L = D^{-1/2}KD^{-1/2}$, 其中$D_{ij} = \sum_{j=1}^{m} K_{ij}$.

相似性矩阵的最大的k个特征向量(k为要识别的类别数目)用来产生$n * k$矩阵Y, 列再次归一化. Y的每一行为一个样本点, 最后使用kmeans方法来聚类.

下面是例子, 将两个螺旋分布的数据分类.

```
data(spirals)
sc <- specc(spirals, centers = 2)
plot(spirals, pch = (23 - 2 * sc))
```

92.8.9 核方法: Kernel principal components analysis

Kernel principal components analysis(PCA)

Kernel PCA (Scholkopf, Smola, and Muller 1998)执行一个坐标系的非线性变换, 来发现非线性相关的主成分.

算法: 略

kpca() 函数实现此功能.

92.8.10 核方法: Kernel feature analysis

核特征分析. 虽然核PCA方法很好, 但是有几个问题需要讨论.

首先, kpca 算法复杂度为$O(m^3)$.

其次, resulting feature extractors are given as a dense expansion in terms of the training patterns. 在有监督的学习中, 使用一个l_1惩罚expansion coefficients, 常常可以得到稀疏解. 可以使用相同的算法用于特征分析, 只使用n个基本函数近似前n个特征. Kernel feature analysis(Smola, Mangasarian, and Scholkopf 2000) 计算简单, 算法复杂度比kpca低一个等级.

公式: 略

kfa() 执行此 Kernel feature analysis

92.8.11 核方法: Kernel canonical correlation analysis

Kernel canonical correlation analysis(CCA, 典型相关分析)

典型相关分析中

$$y_1 = w_1 x_1 = \sum_j w_1 x_{1j}$$

$$y_2 = w_2 x_2 = \sum_j w_2 x_{2j}$$

类似kpca, cca可以扩展为

$$y_1 = w_1 \Phi(x_1) = \sum_j w_1 \Phi(x_{1j})$$

$$y_2 = w_2 \Phi(x_2) = \sum_j w_2 \Phi(x_{2j})$$

Φ为内积函数.

```
data(spam)
train <- sample(1:dim(spam)[1], 400)
kpc <- kpca(~., data = spam[train, -58], kernel = "rbfdot",
   kpar = list(sigma = 0.001), features = 2)
kpcv <- pcv(kpc)
plot(rotated(kpc), col = as.integer(spam[train, 58]),
   xlab = "1st Principal Component", ylab = "2nd Principal Component")

data(promotergene)
f <- kfa(~., data = promotergene, features = 2, kernel = "rbfdot",
   kpar = list(sigma = 0.013))
plot(predict(f, promotergene), col = as.numeric(promotergene[,
   1]), xlab = "1st Feature", ylab = "2nd Feature")
```

92.8.12 TODO: Interior point code quadratic optimizer

92.8.13 TODO: Incomplete cholesky decomposition

矩阵的不完全cholesky分解

Chapter 93

HMM

参考文献: 一本电子书. 无封面.

93.1 介绍

隐马尔可夫模型(Hiden Markov Models, HMM), 作为语音信号的一种统计模型, 今天在语音处理各个领域中获得广泛的应用. 其理论基础, 是在1970年前后由Baum等建立起来. 随后由CMU的Baker和IBM的Jelinek等人将其应用到语音识别之中. 由于Bell实验室Rabiner等人在80年代中期对HMM的深入浅出的介绍, 才逐渐使HMM为世界各国语音处理的研究人员所了解和熟悉, 进而称为公认的一个研究热点.

93.1.1 一个通俗的例子

来自: 一亩二分地 http://www.ohehlium.com/journal/2007/01/07/232/

隐马尔可夫模型(HIDDEN MARKOV MODEL - HMM)

这又是一个和统计有关的话题, 自然还是以赌场的例子开头。这个例子叫“Fair Bet Casino”（公平赌博的赌场？）。赌具十分简单：硬币, 只赌两面, 头(H)还是尾(T)：(Head or Tail)。

当然赌场的庄家留着一手，那就是准备了两枚做了手脚的硬币，一枚叫多头(D)，也就是出现H(Head)的几率要大于T(Tail)，比如H占0.7的机会，T只占0.3；另一枚则是少头(S)，出现H的机会要小于T，比如H占0.3的机会，T只占0.7。不过幸好这位庄家老千技术不过硬，怕被人看出破绽，不敢时时的换硬币，也就大概每十次才敢动手换一下手中的硬币(0.1的几率)。知道了这些，就让你来想办法怎样能赢。当然首先内线提供的这些情报得准确，其次，得想办法找出在哪个阶段庄家在使用哪枚硬币，然后就可以使劲的对症下药，猛押几率大的，当然再祈祷一下，庄家不要突然换硬币（赢面还是很大的，毕竟很大的可能性还是继续用同一枚硬币的(0.9 vs 0.1)）。

这个例子从某种角度说明了，我们对隐藏的信息更感兴趣，也就是庄家究竟在使用哪枚硬币，但我们能直接观察到的只是H和T。比如连续出现三个H (HHH)，直觉就可以判断很有可能是在用D硬币（具体的数值则是比较0.7*0.7*0.7¿0.3*0.3*0.3），当然前提还必须是只使用了其中一种硬币。如果要考虑每扔一次硬币后，有机会选择下一枚硬币（0.9的概率继续使用，0.1的概率换另一枚硬币），那仍然一直使用D硬币来出现三个H的可能性就为 0.9*0.7*0.9*0.7*0.9*0.7（这其中还是有一步假设，那就是第一步之前所使用的硬币还是D硬币；如果第一步使用之前的硬币为S硬币，则一直使用D硬币来出现三个H的概率就变为0.1*0.7*0.9*0.7*0.9*0.7）。不管怎样，只是在连续3个H的这种情况下，很大的可能性是在一直使用着D硬币，但如果是HTHHTTTHH..，或者更随机的情况呢，它究竟是D硬币还是S硬币产生的呢，或者其间还换了硬币？

除了上述这个赌场问题，其实隐马尔可夫模型(HMM)更十分广泛使用于语音识别，因为电脑或机器直接接受的是音频音调，然后来猜测实际要表达的词汇。它也广泛应用于生物信息学（bioinformatics），比较著名的例子就是探测DNA序列中的CG岛的存在。众所周知，A、T、C、G四个碱基交替出现组成DNA链，尽管只有四个组员，C和G却一般不常碰头，据说两个碰在一起了，比较容易被甲基化，把基因的遗传信息给弄坏了可是件大事，所以两者也就拘束着。但在一些地方，可能由于某些保护机制，这两者在一起的概率要远远大于其他区域，这些CG可以自由碰面的地区也就被称为CG岛。这些CG岛也就被猜测为对DNA遗传信息的控制有着一定的作用，所以被scorp称为游手好闲的家伙们就会来计算一下，究竟基因的那些区域属于CG岛。同样他们用的也是HMM的原

理。本来还想编个小程序来展示一下，如何解那个赌场问题，可是想不通怎样很好的记录最有可能的隐蔽态（Hidden state）。所以也就不在这儿献丑了。

93.2 HMM中的三个经典问题

HMM有三个经典(canonical)问题:

- 已知模型参数，计算某一特定输出序列的概率.通常使用forward算法解决.
- 已知模型参数，寻找最可能的能产生某一特定输出序列的隐含状态的序列.通常使用Viterbi算法解决.
- 已知输出序列，寻找最可能的状态转移以及输出概率.通常使用Baum-Welch算法以及Reversed Viterbi算法解决.

另外,最近的一些方法使用Junction tree算法来解决这三个问题。

93.3 模型与定义

93.3.1 球和缸(Ball and Urn)实验模型

由于实际问题比Markov链模型描述的更为复杂, 观察到的事件并不是与状态一一对应, 而是通过一组概率分布相联系, 这样的模型就称为HMM. 它是一个双重随机过程. 其中之一是Markov链, 描述状态的转移. 另一个随机过程描述状态和观察值之间的统计对应关系.

这样, 站在观察者的角度上, 只能看到观察值, 不像Markov链模型中的观察值和状态一一对应. 而是通过一个随机过程去感知状态的存在机器特征. 因此称为隐马尔可夫链, 即HMM.

下面是一个著名的说明HMM概念的例子, 球和缸(Ball and Urn)实验.

设有N个缸, 每个缸中有很多彩球, 其颜色由一组概率分布描述. 实验这样进行, 根据某个初始概率分布, 随机选择一个

Table 93.1: 球和缸(Ball and Urn)实验

缸1	缸2	缸N
P(红)=b_{11}	P(红)=b_{21}	P(红)=b_{N1}
P(蓝)=b_{12}	P(蓝)=b_{22}	P(蓝)=b_{N2}
P(绿)=b_{13}	P(绿)=b_{23}	P(绿)=b_{N3}
$\cdots$	$\cdots$	$\cdots$
P(黄)=b_{1M}	P(黄)=b_{2M}	P(黄)=b_{NM}

缸, 例如第i个缸, 再根据这个缸中的彩球颜色的概率分布, 随机选择一个球, 记下球的颜色, 记为o_1, 再把球放回缸中, 又根据描述缸的转移的概率分布, 随机选择下一个缸, 例如第j个缸, 再从缸中随机选择一个球, 记下球的颜色, 记为o_2, 一直进行下去.

这样我们得到一个描述求的颜色的序列$o_1, o_2, \cdots$. 称为观察值序列. 但是缸之间的转移及每次选取的缸被隐藏起来了, 并不能直接观察到. 而且从每个缸中选取的球的颜色并不是与缸一一对应, 而是由该缸中彩球颜色概率分布随机决定的. 每次选取哪个缸由一组转移概率决定.

93.3.2 HMM定义

一个HMM可以由下列参数描述:

- N: 模型中Markov链状态数目. 记N个状态为$\theta_1, \cdots, \theta_N$. t时刻Markov链所处的状态为$q_t$, 则$q_t \in (\theta_1, \cdots, \theta_N)$. 球缸模型中, 缸就相当于状态. 赌场模型中, 不同的骰子就是不同的状态.
- M: 每个状态对应的可能的观察值数目. M个观察值为$V_1, \cdots, V_M$, t时刻观察到的观察值为o_t, 其中$o_t \in (V_1, \cdots, V_M)$. 球缸模型中, 所选的彩球的颜色就是观察值. 每个骰子的不同面就是观察值.

- 初始状态概率矢量$\pi = (\pi_1, \cdots, \pi_N$, 其中$\pi_i = P(q_1 = \theta_i)$, $1 <= i <= N$, 为实验开始的时候选取某个缸的概率. 开始选择某个骰子的概率.
- A: 状态转移概率矩阵, $A = (a_{ij})_{N*N}$, 其中$a_{ij} = P(q_{t+1} = \theta_j | q_t = \theta_i)$, $1 <= i, j <= N$. 球缸模型中指每一次选取下一个缸的概率. 下一次选择每个骰子的概率.
- B: 观察值概率矩阵, $B = (b_{jk})_{N*M}$, 其中 $b_{jk} = P(o_t = V_k | q_t = \theta_j$. 球缸模型中$b_{jk}$就是第j个缸中球的颜色k出现的概率. 此次出现面x的概率.

这样, 可以记一个HMM为

$$\lambda = (N, M, \pi, A, B)$$

或简写为

$$\lambda = (\pi, A, B)$$

形象的说, HMM可以分为两部分. 一个是Markov链, 由π, A描述, 输出为状态序列, 另一个是一个随机过程, 由B描述, 输出为观察值序列. 我们只能看到观察值, 如下图所示.

```
        q_1,q_2,...,q_T
Markov链------------------>随机过程----->观察值序列
(\pi,A)                  (B)       o_1,....,o_T
```

93.4 前向-后向算法

93.4.1 解决的问题(对应模型产生指定序列的概率)

我们想求得对应模型$\lambda = (\pi, A, B)$产生输出$O = o_1, o_2, \cdots, o_T$的概率$P(O|\lambda)$.

93.4.2 直接计算

$P(O|\lambda)$直接的计算方法为: 对一个固定的状态序列$S = q_1, \cdots, q_T$, 有

$$P(O|S,\lambda) = \prod_{t=1}^{T} P(O_t|q_t,\lambda) = b_{q_1}(o_1)b_{q_2}(o_2)\cdots b_{q_T}(o_T)$$

其中

$$b_{q_t}(o_t) = b_{jk}|_{q_t=\theta_j, o_t=V_k},\ 1 <= t <= T$$

对于给定λ, 产生S的概率为

$$P(S|\lambda) = \pi_{q_1} a_{q_1,q_2} \cdots a_{q_{T-1},q_T}$$

让S取遍所有可能的组合, 得到所求概率为

$$\begin{aligned} P(O|\lambda) &= \sum_{all\ \ S} P(O|S,\lambda)P(S|\lambda) \\ &= \sum_{q_1,\cdots,q_T} \pi_{q_1} b_{q_1}(o_1) a_{q_1,q_2} b_{q_2}(o_2) \cdots a_{q_{T-1},q_T} b_{q_T}(o_T) \end{aligned}$$

其计算量是十分惊人的, 大约为$2TN^T$. 当$N = 5, T = 100$, 计算量为10^{72}, 这是完全不能接受的.

93.4.3 前向算法

已知模型参数, 计算某一特定输出序列的概率.通常使用forward算法解决.

计算量变为: $N(N+1)(T-1) + N$次乘法和$N(N-1)(T-1)$次加法.

是一种典型的格形算法, 是一种动态规划算法(dynamic programming algorithm).

定义前向变量, 即前$t-1$步观察值为$o_1, \cdots, o_{t-1}$, 且第t步选择第i个状态, 观察值为o_t的概率

$$\alpha_t(i) = P(o_1, \cdots, o_t, q_t = \theta_i|\lambda), \quad 1 <= t <= T$$

那么从开始有

- 初始化: 第一步选择第i个状态, 然后其观察值为o_1的概率为

$$\alpha_1(i) = \pi_i b_i(o_1), \ 1 <= i <= N$$

- 递归: 第$t+1$步选择第j个状态, 且观察值为o_{t+1}的概率就是第t步选择第i个状态, 然后第$t+1$步选择第j个状态, 且观察值为o_{t+1}的概率的乘积对所有i求和. 实际上是全概率公式的运用[1].

$$\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i)a_{ij}]b_j(o_{t+1}), \ 1 <= t <= T-1, 1 <= j <= N$$

- 终结:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

其中

$$b_j(o_{t+1}) = b_{jk}|o_{t+1} = V_k$$

[1]再复习一下全概率公式: 设$A_1, ..., A_n$为一般空间的一个分割, 则

$$B = \sum_{i=1}^{\infty} A_i B$$

由完全可加性和乘法定理得

$$P(B) = \sum_{i=1}^{\infty} P(A_i)P(B|A_i)$$

93.4.4 前向算法的例子

例如, 观察值序列为”红,红,蓝,黄”, 设1,2,3分别代表颜色”红,黄,蓝”, 那么$o = (1, 1, 3, 2)$, 下面计算给定的$\lambda = (\pi, A, B)$下出现此观察值序列的概率

```
# Markov转移矩阵A
T=matrix(c(.7,.3,.3,.7),nc=2,
       dimnames=list(c("缸1","缸2"),c("缸1","缸2")))
# 观察值概率矩阵
B=matrix(c(.2,.3,.5,.1,.7,.2),nc=2,
       dimnames=list(c("红","黄","蓝"), c("缸1","缸2")))
# 初始状态概率向量
pi=c(0.6,0.4)
# 观察值序列
o=c(1,1,3,2)
# 保存前向概率的向量为
Alpha.now=c(0,0)
Alpha.new=c(0,0)
```

初始化: t=1时,o[1]=1, 即观察值为红色, 首先计算选择第一个缸且出现红色的概率$\alpha_{t=1}(1) = \pi_1 b_1(o1)$

```
# t=1时, 选择缸1且观察到o[1]=1(红色)的概率
Alpha.now[1]=pi[1]*B[o[1],1]
# t=1时, 选择缸2且观察到o[1]=1(红色)的概率
Alpha.now[2]=pi[2]*B[o[1],2]

# 结果
> Alpha.now
[1] 0.12 0.04
```

下面是递归部分

```
# 分步计算
```

```
# t=2时, 选择缸1且观察到o[2]=1(红色)的概率
Alpha.new[1]=sum(Alpha.now*T[,1])*B[o[2],1]
# t=2时, 选择缸2且观察到o[2]=1(红色)的概率
Alpha.new[2]=sum(Alpha.now*T[,2])*B[o[2],2]
> Alpha.new
[1] 0.0192 0.0064
# t=3时, Alpha.now=Alpha.new,
# 然后一直更新Alpha.new即可
```

终结部分很简单

```
# 最终的概率为
P=sum(Alpha.new)
```

下面将完整的循环形式写出,

```
# Markov转移矩阵A
T=matrix(c(.7,.3,.3,.7),nc=2,
       dimnames=list(c("缸1","缸2"),c("缸1","缸2")))
# 观察值概率矩阵
B=matrix(c(.2,.3,.5,.1,.7,.2),nc=2,
       dimnames=list(c("红","黄","蓝"), c("缸1","缸2")))
# 初始状态概率向量
pi=c(0.6,0.4)
# 观察值序列
o=c(1,1,3,2)

forword<-function(pi,T,B,o){
   tt=length(o) # 观察值的个数, 即步数
   N=length(pi) # 状态个数
   # 保存前向概率的向量为
   Alpha.now=rep(0,N)
   Alpha.new=rep(0,N)

   # 初始化
   for (i in 1:N){
```

```
        Alpha.now[i]=pi[i]*B[o[1],i]}

    cat('time',1,':','o:',o[1],'P:',Alpha.now,'\n')

    # 递归部分
    for (t in 2:tt){
        for (j in 1:N){
            Alpha.new[j]=sum(Alpha.now*T[,j])*B[o[t],j]}
        Alpha.now=Alpha.new
        cat('time',t,':','o:',o[t],'P:',Alpha.now,'\n')}

    # 最终部分
    P=sum(Alpha.new)
    P
}

# 观察值多一点
o=c(1,2,1,3,3,2,1)
> forword(pi,T,B,o)
time 1 : o: 1 P: 0.12 0.04
time 2 : o: 2 P: 0.0288 0.0448
time 3 : o: 1 P: 0.00672 0.004
time 4 : o: 3 P: 0.002952 0.0009632
time 5 : o: 3 P: 0.00117768 0.000311968
time 6 : o: 2 P: 0.0002753899 0.0004001771
time 7 : o: 1 P: 6.256522e-05 3.627410e-05
[1] 9.883931e-05

# o的长度为112的时候
time 1 : o: 1 P: 0.12 0.04
time 2 : o: 2 P: 0.0288 0.0448
time 3 : o: 1 P: 0.00672 0.004
...
time 110 : o: 3 P: 6.995718e-64 1.858370e-64
time 111 : o: 2 P: 1.636354e-64 2.379702e-64
time 112 : o: 1 P: 3.718717e-65 2.156698e-65
[1] 5.875415e-65
```

93.4.5 后向算法

已知模型参数，计算某一特定输出序列的概率.通常使用forward算法解决.

计算量变为: $N(N+1)(T-1)+N$次乘法和$N(N-1)(T-1)$次加法. 也是一种典型的格形算法.

定义后向变量

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \cdots, o_T, q_t = \theta_i|\lambda), \quad 1 <= t <= T-1$$

其中

$$\beta_T(i) = 1$$

类似有步骤

- 初始化:

$$\beta_T(i) = 1, \quad 1 <= i <= N$$

- 递归(由于原文看不清楚, 此步公式可能有错误):

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), \ t = T-1, T-2, \cdots, 1, \quad 1 <= i <= N \alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i) a$$

- 终结:

$$P(O|\lambda) = \sum_{i=1}^{N} \beta_1(i)$$

算法结束[2]

[2]怀疑有点问题, π没有用到? 终结步骤可能为这样

$$P(O|\lambda) = \sum_{i=1}^{N} \beta_1(i)\pi_i$$

93.5 Viterbi算法

93.5.1 解决的问题(给定模型和序列, 最可能的状态序列)

此算法解决的问题是, 给定一个模型$\lambda = (\pi, A, B)$和观察值序列$O = o_1, o_2, \cdots, o_T$, 求最可能(最佳意义上)产生此观察值序列的状态的序列$Q = q_1, \cdots, q_T$.

由于不同的状态序列可能产生一样的观察值序列, 此处最佳的意义是指使得$P(Q, O|\lambda)$最大的状态序列.

93.5.2 算法描述

定义$\delta_t(i)$为时刻t沿一条路径$q_1, q_2, \cdots, q_t$,且$q_t = \theta_i$,产生出$o_1, o_2, \cdots, o_t$的最大概率. 即

$$\delta_t(i) = max_{q_1, q_2, \cdots, q_{t-1}} P(q_1, q_2, \cdots, q_t, q_t = \theta_i, o_1, o_2, \cdots, o_t|\lambda)$$

那么求取最佳状态序列的过程为

- 初始化:

$$\begin{aligned} \delta_1(i) &= \pi_i b_i(o_1), 1 <= i <= N \\ \varphi_1(i) &= 0, 1 <= i <= N \end{aligned}$$

- 递归:

$$\begin{aligned} \delta_t(j) &= max_{1<=i<=N}[\delta_{t-1}(i)a_{ij}]b_j(o_t), 2 <= t <= T, 1 <= j <= N \\ \varphi_t(i) &= argmax_{1<=i<=N}[\delta_{t-1}(i)a_{ij}], 2 <= t <= T, 1 <= j <= N \end{aligned}$$

- 终结:

$$\begin{aligned} P &= max_{1<=i<=N}[\delta_T(i)] \\ q_T &= argmax_{1<=i<=N}[\delta_T(i)] \end{aligned}$$

- 状态序列求取:

$$q_t = \varphi_{t+1}(q_{t+1}), \quad t = T-1.T-2, \cdots, 1$$

Viterbi算法的一个副产品

$P = max_{q_1,q_2,\cdots,q_{t-1}} P(q_1, q_2, \cdots, q_t, q_t = \theta_i, o_1, o_2, \cdots, o_t|\lambda) = max_Q P(Q, O|\lambda)$

是前向后向计算出的$P(O|\lambda) = \sum_Q P(Q, O|\lambda)$中举足轻重的唯一成分.

对于语音处理, $P(Q, O|\lambda$动态范围很大, 或者说不同的Q使得$P(Q, O|\lambda$的值差别很大. 因此常常等价的使用$max_Q P(Q, O|\lambda)$和$P(O|\lambda)$.

同样, 由后向算法的思想出发, 也可以推导出 Viterbi 算法的另外一种实现方式.

93.6 Baum-Welch算法

93.6.1 解决的问题(给定序列, 参数估计)

解决HMM训练, 即HMM参数估计问题, 给定一个观察值序列$O = o_1, o_2, \cdots, o_T$,该算法能够确定一个$\lambda = (\pi, A, B)$, 使得$P(O|\lambda)$最大. 即估计初始状态概率矢量, 状态转移概率矩阵,观察值概率矩阵.

93.6.2 算法描述

显然, 由前面定义的前向和后向变量, 有

$$P(O|\lambda) = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \ 1 <= t <= T-1$$

此处求取λ, 使得$P(O|\lambda)$最大, 是一个泛函极值问题. 但是, 由于给定的训练序列有限, 因而不存在一个最佳的方法来估计λ, 在这种情况下, Baum-Welch算法利用递归的思想, 使$P(O|\lambda)$局部最大, 最后得到模型参数$\lambda = (\pi, A, B)$. 此外, 用梯度方法也可以达到类似的目的.

定义$\xi_t(i,j)$为给定训练训练序列O和模型λ时, 时刻t时Markov链处于θ_i状态和时刻$t+1$为θ_j状态的概率, 即

$$\xi_t(i,j) = P(O, q_t = \theta_i, q_{t+1} = \theta_j|\lambda)$$

可以推导出

$$\xi_t(i,j) = [\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)]P(O|\lambda)$$

那么, 时刻t时Markov链处于θ_i状态的概率为

$$\xi_t(i) = P(O, q_t = \theta_i|\lambda) = \sum_{j=1}^{N} \xi_t(i,j) = \alpha_t(i)\beta_t(i)/P(O|\lambda)$$

因此, $\sum_{t=1}^{T-1}\xi_t(i)$表示从θ_i状态转移除去的次数的期望值, 而$\sum_{t=1}^{T-1}\xi_t(i,j)$表示从$\theta_i$状态转移到$\theta_j$状态的次数的期望值. 由此导出了Baum-Welch算法中著名的重估(reestimation)公式

$$\begin{aligned}
\bar{\pi}_i &= \xi_1(i) \\
\bar{a}_{ij} &= \sum_{t=1}^{T-1}\xi_t(i,j) / \sum_{t=1}^{T-1}\xi_t(i) \\
\bar{b}_{jk} &= \sum_{t=1,and\ \ o_t=V_k}^{T}\xi_t(j) / \sum_{t=1}^{T}\xi_t(j)
\end{aligned}$$

推导过程略.

HMM参数$lambda = (\pi, A, B)$的求取过程为: 根据观察值序列O和选取的初始模型$lambda = (\pi, A, B)$, 由重估公式, 求得一组新参数$\bar{\pi}_i, \bar{a}_{ij}, \bar{b}_{jk}$, 亦即得到了一个新的模型$\bar{lambda} = (\bar{\pi}, \bar{A}, \bar{B})$. 可以证明, $P(O|\bar{\lambda}) > P(O|\lambda)$, 即由重估公式得到的$\bar{\lambda}$比在$\lambda$在表示观察值序列O方面要好. 那么, 重复这个过程, 逐步改善模型参数, 直到$P(O|\bar{\lambda})$收敛, 即不再明显增大, 此时的$\bar{\lambda}$即为所求的模型.

93.6.3 讨论

应当指出, HMM训练, 或称参数估计问题, 是HMM在语音处理中应用的关键问题, 也是最困难的问题. Baum-Welch算法只是得到广泛应用的解决这个问题的经典方法, 但并不是唯一的, 也远不是最完善的方法.

93.7 R包

hmm.discnp: 使用离散非参数观测到的数据分布拟合HMM. 并使用训练模型模拟数据

RHmm: 离散单/多元变量高斯, 混合高斯HMM函数, 用于模拟和预测.

stochmod: 多个概率模型的学习与推断.

mhsmm: HMM和semi-Markov models参数估计与预测. 数据可以是多个观测序列. 时间间隔必须一致, 缺失数据是允许的. 观测数据可以是多元的. 使用 EM算法估计参数. 核心使用C. 允许用户设置初始分布.

93.7.1 HMMFit()-估计HMM参数

HMMFit(): 已知观测序列和状态数目, 使用 Baum-Welch 算法估计HMM的参数. 给定一个观察值序列$O = o_1, o_2, \cdots, o_T$,该算法能够确定一个$\lambda = (\pi, A, B)$, 使得$P(O|\lambda)$最大. 即估计初始状态概率矢量, 状态转移概率矩阵,观察值概率矩阵.

```
library(RHmm)
data(n1d_3s)
ResFit <- HMMFit(obs_n1d_3s, nStates=3) # 估计HMM参数
VitPath <- viterbi(ResFit, obs_n1d_3s) # 使用估计的参数估计新序列最优隐状态序列
```

```
# n1d_3s 有10条序列. 每条观察100次试验.(可以不同)
# 相当于骰子100次为1个观察序列, 10次.
> data(n1d_3s)

# 对观察序列估计参数. 状态数为3. 先验为高斯分布.
> ResFit=HMMFit(obs_n1d_3s, nStates=3); ResFit
... Computing the asymptotic covariance matrix ...

Call:
----
HMMFit(obs = obs_n1d_3s, nStates = 3)

Model:
------
3 states HMM with univariate gaussian distribution

Baum-Welch algorithm status:
----------------------------
Number of iterations : 25
Last relative variation of LLH function: 0.000001

Estimation:
-----------
# 开始选择3个状态的概率
Initial probabilities:
      Pi1       Pi2       Pi3
  0.4823016 0.2000004 0.3176981

# 3个状态之间转移概率, 即下一次选择另一个状态的概率.
Transition matrix:
         State 1   State 2    State 3
State 1 0.4248688 0.3207284 0.25440277
State 2 0.4076902 0.5014643 0.09084547
State 3 0.1231243 0.1770507 0.69982504

# 条件分布的参数. 即状态观察值的概率. 此处为正态分布. 以均值和方差标志.
Conditionnal distribution parameters:

Distribution parameters:
           mean     var
```

```
State 1 -1.082003 0.960587
State 2 10.038389 4.538559
State 3 -4.879551 2.069071

# 对数似然.
Log-likelihood: -2654.07
BIC criterium: -2792.22
Warning messages:
1: In nlmeHessian(par = Teta0, fn = NumLLH, HMM = HMM, obs = obs, nSample = nSample
  non-finite finite-difference value while computing the Hessian matrix ...
2: In asymptoticCovMat(Res2, obs, paramAlgo$asymptMethod) :
  Hessian matrix is not inversible
```

93.7.2 viterbi()-估计隐状态序列

viterbi(): 已知HMM, 和新观测序列, 使用 viterbi 算法估计最优隐状态序列(能够产生此序列的概率最大的隐状态序列).

```
> VitPath <- viterbi(ResFit, obs_n1d_3s) # 使用估计的参数估计新序列最优隐状态序列
> VitPath
$states
$states[[1]]
  [1] 3 3 1 2 2 2 2 2 2 2 3 3 3 3 1 1 2 2 2 1 3 1 1 2 1 3 1 1 3 3 1 1 1 1 3 1 1
 [38] 3 3 3 3 1 3 1 2 2 1 1 1 1 3 1 3 1 2 2 1 2 2 3 3 1 1 1 1 1 1 1 3 1 1 1 2 2
 [75] 3 1 1 3 1 1 3 3 3 3 3 1 3 1 1 3 1 1 1 3 2 1 1 2 2 2

$states[[2]]
  [1] 2 3 3 1 3 1 1 3 1 2 2 2 2 2 2 2 2 2 2 3 3 3 1 1 2 1 3 1 3 1 3 1 1 1 1 1 3
 [38] 1 1 1 2 3 2 2 1 3 3 3 1 2 3 3 1 3 3 1 1 3 1 2 2 2 2 2 2 2 2 1 2 2 2 2 3 1
 [75] 1 3 2 2 2 2 3 3 3 3 3 1 3 1 1 1 3 3 3 1 3 2 3 3 3 2

$states[[3]]
  [1] 1 1 3 2 2 2 2 2 1 2 2 2 2 3 3 3 1 2 2 2 3 3 1 3 3 1 1 3 1 3 1 3 1 3 1 3 1
 [38] 1 1 1 1 1 1 1 1 1 1 3 3 3 3 2 2 2 2 2 2 3 1 2 2 1 1 1 1 1 2 2 2 2 2 2 1 1
 [75] 1 3 1 1 1 3 1 3 1 3 3 1 3 3 2 2 2 2 2 3 3 3 3 3 3 3
```

```
...10条序列的隐状态...

# 每个隐状态对应的对数 ViterbiScore
$logViterbiScore
$logViterbiScore[[1]]
[1] -267.5053

$logViterbiScore[[2]]
[1] -276.6716

$logViterbiScore[[3]]
[1] -272.6233

...10个对数 ViterbiScore...


# 每个隐状态对应的对数 ProbSeq
$logProbSeq
$logProbSeq[[1]]
[1] -264.2509

$logProbSeq[[2]]
[1] -274.1634

$logProbSeq[[3]]
[1] -270.1086

...10个对数 ProbSeq...

attr(,"class")
[1] "viterbiClass"
```

93.7.3 forwardBackward()-某指定观测序列的概率

已知序列和HMM, 求得到此序列的概率. o为观测序列

$$\begin{aligned} o &= o(1), \cdots, o(T) \\ \rho(t) &= P(O_1 = o(1), \cdots, O_t = o(t)|HMM) \end{aligned}$$

结果中 LLH 即为HMM下出现观测序列的概率的对数

$$LLH = \ln \rho[T]$$

其它结果参考帮助.

```
> fb <- forwardBackward(ResFit, obs_n1d_3s[[1]])
> str(fb)
List of 6
 $ Alpha: num [1:100, 1:3] 2.82e-33 3.50e-37 1.78e-04 1.58e-09 6.47e-08 ...
 $ Beta : num [1:100, 1:3] 6.92e+65 1.54e+65 3.38e+64 7.22e+63 1.58e+63 ...
 $ Gamma: num [1:100, 1:3] 7.76e-32 1.20e-34 9.99e-01 1.27e-04 5.18e-02 ...
 $ Xsi  :List of 100
  ..$ : num [1:3, 1:3] 6.64e+32 8.91e+38 8.22e+63 5.78e+32 1.64e+39 ...
  ..$ : num [1:3, 1:3] 2.10e+28 8.80e+34 1.71e+62 1.66e+28 1.47e+35 ...
  ..$ : num [1:3, 1:3] 1.96e+60 1.01e+57 8.69e+55 2.29e+60 2.50e+57 ...
.........................
  ..$ : num [1:3, 1:3] 1.29e-121 6.91e-115 3.21e-128 1.37e-121 1.54e-114 ...
  ..$ : num NaN
 $ Rho  : num [1:100] 3.55e-02 2.90e-03 1.78e-04 1.17e-05 1.15e-06 ...
 $ LLH  : num -3.25
```

Chapter 94

TODO: 神经网络

94.1 包介绍

nnet: S-plus 的 VR 包: Venables and Ripley, 'Modern Applied Statistics with S' (4th edition).包括 MASS, class, nnet, spatial. 没有单独的介绍.

neural: RBF, MLP neural networks, 带图形界面.

neuralnet: 使用下面方法训练神经网络 1. Resilient Backpropagation with (Riedmiller, 1994) or 2. without Weightbacktracking (Riedmiller, 1993) or 3. the modified globally convergent version by Anastasiadis et. al. (2005). 允许灵活的设置误差与激活函数. 而且可以计算全局权值(generalized weights (Intrator O and Intrator N, 1993))

AMORE: (A MORE flexible neural network package) 用来实现 TAO robust (TAO 鲁棒的)神经网络算法. 已经修改, 应该可以适用于用户自定义的训练算法和其它用户自定义的要求.

grnnR: (A Generalized Regression Neural Network)由训练数据(P(atterns) and T(argets))产生一个广义回归神经网络.

94.2

94.3 参考文献: 进化BP神经网络的围岩位移预测

权值, 结构, 学习参数 3 个方面进化.

初始种群: 隐层限制在1-100个节点之间. 速率与动量因子在0.1-0.9之间.

适应度: 网络误差, 训练时间, 结构复杂度.

94.4 参考文献: 基于进化神经网络的模拟电路故障诊断

网络编码: 改进的Miller矩阵编码, 结构进化中采用二进制编码. 权值采用实值编码.

适应度函数: 结构进化的适应度. 学习精度, 泛化能力, 学习效率, 网络复杂度.

权值适应度: 不考虑网络复杂度等和网络结构相关的参数.

遗传算法: 排序选择.

自适应进化速率.

94.5 参考文献: 一种基于多进化神经网络的分类方法

神经网络模型的训练:

个体表示方法: 进化中的个体为单个神经网络的所有权值和阈值

初始群体: 随机产生P个父本, 即P个神经网络, 结构相同.

变异: 对任意父本X_i, 其第k代为X_i^k.

适应度: 直接设置为优化问题的目标函数值.

Chapter 95

TODO: 遗传算法(Genetic Algorithm)

95.1 包介绍

genalg: 二元与浮点数作为染色体的遗传算法. 也包含一个函数用于多元函数的优化.

gafit: 基于用户定义的表达式计算一组采样点, 采样点是一列可能替换表达式的有值的参数. 遗传算法目标是使表达式的结果最小化(通常是残差平方和). 实际上它属于优化的包.

mco: 用于多约束下的优化的包, 使用遗传算法.

Part XI

随机数与MCMC

Chapter 96

随机数的产生及检验

参考文献 [17] 14.2

96.1 随机数的产生

从现在起, 我们将$[0,1]$区间均匀分布的随机数简称为随机数, 用r_i表示.

随机数的产生递推公式有如下形式

$$r_{n+k} = T(r_n, r_{n+1}, \cdots, r_{n+k-1})$$

T是某个函数. 给定初值$r_1, \cdots, r_k$, 按上式可确定r_{n+k}构成随机数列. 经常用$k=1$的情况, 递推公式简化为

$$r_{n+1} = T(r_n)$$

递推公式产生的随机数列存在两个问题

1. 初值确定后, 随机数列就确定了.不满足随机数之间相互独立的要求.
2. 随机数列是按确定算法计算处来的. 计算机能够表示的$[0,1]$区间的数是有限多个, 由计算机字长决定. 故递推到

一定时候, 同一个数字总会出现第二次. 此后就出现周期性的重复现象. 这样的数列不符合随机性(对于均匀分布就是均匀性)要求.

故递推产生的随机数常常称为伪随机数. 这两个缺点不可能根本解决. 但是, 只要递推公式比较好, 独立性可以近似满足. 周期如果足够长, 使得在使用的随机数列长度小于周期,

产生伪随机数的方法有很多, 常用的有

96.1.1 乘同余法

$$x_{n+1} = a * x_n (mod(M)), \qquad r_{n+1} = \frac{x_{n+1}}{M}$$

其中a为乘因子, 一般是一个大的素数. M为正整数(称为模数), 通常为2^k, k是计算机的最大有效位数. mod是modula, 求模(求余数).

数列的最大容量为$L <= M$.

独立性取决于x_0, a的选择.

96.1.2 乘加同余法

$$x_{n+1} = a * x_n + C(mod(M)), \qquad r_{n+1} = \frac{x_{n+1}}{M}$$

C是非负数. 适当选取C可改善随机数列的统计性质.

数列的最大容量为$L <= M$.

96.1.3 加同余法

$$x_{n+2} = x_n + x_{n+1}(mod(M)), \qquad r_{n+2} = \frac{x_{n+2}}{M}$$

数列的最大容量在一般情况下考虑是很困难的. 对于$M = 2^k, x_0 = x_1 = 1$, 最大容量为$1.5M$.

96.2 随机数的统计检验

检验是否服从$[0, 1]$均匀分布.

96.2.1 参数检验

产生的N个随机数可以构成各阶子样矩,

$$m'_k = \frac{1}{N}\sum_{i=1}^{N} r_i^k, \quad k = 1, 2, \cdots$$

期望和方差的理论值为

$$m_k = \frac{1}{k+1} \qquad \sigma^2_{k,N} = (\frac{1}{2k+1} -$$

根据中心极限按定理

$$Z_{k,N} = \frac{m'_k - m_k}{\sigma_{k,N}} \sim N(0, 1)$$

由此可以检验$Z_{k,N} > Z_{alpha/2}$则拒绝零假设, 不服从正态分布.

96.2.2 均匀性检验

通过频率和累积频率分布可以检验.

频率检验: 假设划分k个区间, 每个区间理论个数为$m_j = N * \frac{1}{k}$, 实际观测的为n_j, 那么

$$\chi^2 = \sum_{j=1}^{k} \frac{(n_j - m_j)^2}{m_j} \sim \chi^2_{k-1}$$

累积频率检验: 略, 使用 ks 方法.

96.2.3 独立性检验

a 相关系数独立性检验.

$$\hat{\rho}_j = [\frac{1}{N-j} \sum_{i=1}^{N-j} r_i r_{j+i} - \bar{r}^2]/s^2$$

$$s^2 = \frac{1}{N} \sum (r_i - \frac{1}{2})^2$$

$$u = \hat{\rho}_j \sqrt{N-j} \sim N(0,1)$$

u可以作为独立性的检验标准

b 列联表独立性检验

略

c 多维频率检验

略.

96.2.4 TODO: 连贯性检验

将随机数按某种规律分为2类, 按随机数出现的顺序排列其对应类别, 此为一个游程检验.

略

Chapter 97

蒙特卡洛方法的随机抽样

参考文献 [17] 14.3

97.1 介绍

蒙特卡洛方法解题时, 经常遇到各种不同分布的随机变量, 要求产生对应该随机变量的随机子样. 这一步骤称为对此随机变量的随机模拟或随机抽样.

任意分布的随机变量可由[0, 1]区间均匀分布的随机数经过变换或舍选得到. 只要r_i满足均匀和独立, 则由它产生的任何分布的随机数列(简单随机子样)相互独立且与总体同分布.

97.1.1 直接抽样的优点和缺点

优点(Pros): 简单易行. 不象非独立抽样(mcmc)易陷入局部最优.

缺点(Cons): 独立的抽样可能忽略感兴趣的总体区域(重要抽样可以避免此问题). 收敛性检验不如非独立方法那样强. 不易产生统计性质一致的样本.

97.2 直接抽样方法

97.2.1 离散随机变量

设离散随机变量X取值$x_1, \cdots, x_n$的概率为$p_1, \cdots, p_n$. 其累积概率为

$$F(x) = \sum_{x_i<x} p_i$$

归一性

$$F(x >= x_n) = \sum_{i=1}^{n} p_i = 1$$

当

$$F(x) \in \sum_{i=1}^{i^*-1} p_i, \quad \sum_{i=1}^{i^*} p_i$$

随机变量取值为x_{i^*}

因此当

$$\sum_{i=1}^{i^*-1} p_i < r <= \sum_{i=1}^{i^*} p_i$$

随机变量应该为x_{i^*}

这就是离散随机变量的直接抽样方法.

例如抽样1,2,3的概率分别为$0.5, 0.2, 0.3$, 产生20个子样, 为

```
> sample(1:3,20,prob=c(0.5,0.2,0.3),replace=T)
 [1] 2 1 2 3 1 1 2 1 1 1 1 2 1 2 2 3 2 1 2 2
```

其它函数的抽样类似.

二项分布, 贝努里分布, 泊松分布等有专门的产生此分布的函数, 如$rbinom, rpoisson$等等.

97.2.2 连续随机变量(反函数法)

设函数f在定义域A值域B定义, 则其反函数f^{-1}在定义域B值域A定义

$$f^{-1}(y) = x \Longleftrightarrow f(x) = y$$

概率分布累积函数(CDF)的反函数方法. 因为CDF是单调的, 故其与反函数一一对应. CDF的值域为[0,1], 故反函数的定义域为[0,1]. 故为了模拟概率分布 , 我们可以产生[0,1]分布的均匀随机数x, 利用反函数映射到f(x)即得到需要的概率分布的随机数.

例如指数分布的CDF

$$F(x) = 1 - e^{-\lambda x}$$

令

$$u = F(x) = 1 - e^{-\lambda x}$$

解得

$$x = -\frac{1}{\lambda} log(1 - u)$$

] 此处, 若u为均匀分布, x就是指数分布的

下面我们就利用均匀分布随机数产生指数分布的随机数

```
# 产生n个指数分布的随机数
simExp<-function(lambda, n){
u<-runif(n) # 均匀分布
x<-(-1/lambda)*log(1-u) # 指数分布
}
```

```
y1=simExp(2,1000)
y2=rexp(1000,2)

par(mfrow=c(2,1))
hist(y1,30,xlab='simExp')
hist(y2,30,xlab='rexp')
```

97.3 TODO: 直接抽样方法的推广–变换抽样

反函数法就是变换抽样的一种. 可以推广到一般情况下.

略.

97.4 舍选抽样方法(rejection sampling)

参考文献 [17] 14.3.3

http://en.wikipedia.org/wiki/Rejection_sampling

这个方法可以产生任意分布$f(x)$的随机数. 当$f(x)$的表达式不容易得到的时候常常使用. 这种方法相当于$g(x)$为均匀分布时乘抽样方法。

设随机变量x取值范围$[a, b]$, 概率密度函数$f(x)$为有界函数, 极大值为M

$$f(x) <= M$$

下面是步骤

1. 产生均匀分布的随机数$r_1, r_2 \sim U(0, 1)$

2. $\eta = r_1(b-a)+a$

3. 如果$r_2 <= f(\eta)/M$, 接受η作为一个服从分布$f(x)$的随机数, 否则拒绝.

4. 继续前面直到产生足够的随机数.

上面假定$[a,b]$为有限区间. 对于无限区间, 可以截尾处理, 即选择有限区间$[a,b]$, 满足

$$\int_a^b f(x)dx >= 1-\epsilon$$

只要ϵ足够小, 就可以使得计算误差足够小. 这种方法对于任何抽样方法都适合.

对于抽样效率, 如果$f(x)$比较尖锐(M值越大), 效率就低, 越平坦(M值越小)效率越高.

```
x=1:10
fx=table(sample(x,30,replace=T)) # f(x)  为 fx[x]
> fx # 注意, f(x)要小心. 例如此处缺少 5 的频率(0)

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

eta=round(runif(5000,1,10)) # 从1:10的随机数, eta=r1*(b-a)+a
r2=runif(5000)
a=min(x)
b=max(x)
M=max(fx) # M=5, M越大, 效率越低
s=eta[r2<=(fx[eta]/M)] # 接受的随机数
s1=eta[r2<=(fx[eta]/3)] #  M<max(fx) 结果不正确. 分布不是 fx
s2=eta[r2<=(fx[eta])] # M<max(fx) 结果不正确. 分布不是 fx
s3=eta[r2<=(fx[eta]/10)] # 效率低, 相当于直接先舍弃了一部分.

# 绘图. 产生的数据的分布, 若连续数据使用 hist 函数, 并去掉 table
```

```
op<-par(mfrow=c(5,1))
barplot(fx,main="Original dist") # 指定的分布
barplot(table(s),main="Generated random number, M=5")
barplot(table(s1),main="Generated random number, M=3")
barplot(table(s2),main="Generated random number, M=1")
barplot(table(s3),main="Generated random number, M=10")
par(op)

# 另外一个例子. 模拟 beta 分布, 参数 2,2
x <- runif(5000,0,1)
u <- runif(5000,0,1)
fx=dbeta(x,2,2)
M=1.5
b <- x[fx > M*u]
o=order(x)
hist(b,prob=T)
lines(fx[o]~x[o])
```

97.5 TODO: 利用极限定理采样

n个独立同分布的随机变量有如下分布

$$Y = \frac{\sum_{i=1}^{n} X_i - n\mu}{\sqrt{n}\sigma} \sim N(0,1)$$

利用这一性质可以从$[0,1]$均匀分布的随机数构成标准正态分布.

$$\xi = (\sum_{i=1}^{n} r_i - \frac{n}{2}) / \sqrt{\frac{n}{12}}$$

n应该足够大, 一般$n >= 10$已经足够好. 取$n = 12$的形式特别简单

$$\xi = (\sum_{i=1}^{n} r_i - 6$$

ξ的取值为

$$-\sqrt{3n} <= \xi <= \sqrt{3n}$$

当$n = 6$, 有$-6 <= \xi <= 6$, 在± 6, 密度已经小于10^{-8}. 大多数场合已经足够.

根据德莫佛-拉普拉斯定理, 正态分布是二项分布的极限形式.

$$\frac{B(n;p) - np}{\sqrt{np(1-p)}} \sim N(0,1)$$

那么

$$\eta = \xi\sqrt{np(1-p)} + np \sim B(n;p)$$

凡是存在极限分布的随机变量都可以利用极限分布产生随机数.

97.6 TODO: 复合分布的抽样方法

97.6.1 加抽样方法

97.6.2 乘抽样方法

$$f(x) = H(x)g(x)$$

当$g(x)$为均匀分布时, 就是舍选抽样法.

步骤为

1. 抽取r, ξ
2. 若$r <= H(\xi)/M$, 接受ξ为一个合格的随机数. 否则抛弃.
3. 重复直到产生足够的随机数.

97.6.3 加乘抽样方法

97.7 TODO: 近似抽样方法

97.8 TODO: 多维分布的抽样

一维分布的抽样方法可以应用到多维的情况.

Chapter 98

马尔可夫链蒙特卡洛模拟采样

参考 http://en.wikipedia.org/wiki/Markov_chain_Monte_Carlo

98.1 介绍

98.1.1 马尔可夫链的性质

一致性(homogeneous): 转移概率不随状态的改变而改变.

遍历性(ergodicity):

MCMC(Markov chain Monte Carlo) method, (马尔可夫链蒙特卡洛方法), 包括random walk Monte Carlo method(随机漫步蒙特卡洛方法), 是概率分布抽样的方法. 它构造一个具有与期望概率分布一致的马尔可夫链, 经过一个很长的循环(long number of steps), 然后用来对希望的概率分布采样. 采样的质量证明是循环步数的函数.

mcmc采样方法容易实现和分析, 但是需要经过相当长的时间来遍历整个空间. 遍历时经常会返回已经遍历过的空间.

一般构造具有希望的(目标)概率分布的马尔可夫链不难, 困难的是判断需要多少循环才能收敛到稳定的概率分布(即具有

可以接受的误差). 好的链有一个快速混合(rapid mixing)的特点: 从任何位置开始都会迅速到达平稳分布.

MCMC只能近似目标分布, 因为在开始位置有一个残差效应. 更复杂的基于MCMC的方法, 例如coupling from the past, 可以产生精确的分布, 代价是更多的计算和完成时间不确定(虽然期望是有限的).

最经常的用途是计算多元积分. 多元积分经常出现在贝叶斯统计, 计算物理学和计算生物学中.

这种方法具有采样所期望的所有性质.

不需要借助另外一个分布和舍选采样(rejecting samples)就可以产生与目标分布统计一致的样本.

得到的样本是有依赖性的(非独立的), 并服从马尔可夫性, 允许更加鲁棒的收敛性检验方法, 例如自相关函数, Gelman-Rubin 统计等.

很多mcmc抽样方法需要随机漫步, 例如

- Metropolis-Hastings algorithm
- Gibbs sampling
- Slice sampling
- Multiple-try Metropolis

有些方法更复杂, 但可以避免随机漫步. 例如

- Successive over-relaxation
- Hybrid Monte Carlo (HMC), or Hamiltonian Monte Carlo
- Some variations on slice sampling also avoid random walks

有些还可以改变维数. 例如 Reversible Jump method(is a variant of Metropolis-Hastings). 在模拟空间中的分子的数目发生改变的时候会出现这种情况.

98.1.2 推荐包

coda: 分析诊断马尔可夫链蒙特卡洛模拟的结果. 很多包依赖于它.

mcmc: 使用random-walk Metropolis方法产生任意连续分布的d维随机数. 使用用户指定的非标准化的密度函数.

MCMCpack: 贝叶斯推断, 各种线性或非线性回归模型参数估计. 返回coda包的mcmc对象, 便于分析. 另外包括一个 general purpose Metropolis sampling algorithm 的采样器

gibbs.met: 简单 gibbs 采样

98.2 随机漫步的例子

下面是一个二维随机漫步的例子

```
x=rnorm(100)
y=rnorm(100)
x=cumsum(x)
y=cumsum(y)

# 静态查看
op<-par(mfrow=c(2,2))
plot(y[1:25]~x[1:25],t='l',xlim=c(-6,6),ylim=c(-6,6))
plot(y[1:50]~x[1:50],t='l',xlim=c(-6,6),ylim=c(-6,6))
plot(y[1:75]~x[1:75],t='l',xlim=c(-6,6),ylim=c(-6,6))
plot(y~x,t='l',xlim=c(-6,6),ylim=c(-6,6))
par(op)

# 动态模拟二维随机漫步
walk<-function(step,xstart=0,ystart=0,xlim=c(-15,15),ylim=c(-15,15),sleep=0.1){
   x=xstart
   y=ystart
   plot(x,y,t='l',xlim=xlim+xstart,ylim=ylim+ystart)
   for (i in 1:step){
```

```
        Sys.sleep(sleep)
        x.old=x
        y.old=y
        x=x+rnorm(1)
        y=y+rnorm(1)
        lines(c(x.old,x),c(y.old,y))}
}
walk(100)
```

98.3 Gibbs 采样

Gibbs Sampling 是 Metropolis-Hastings sampling 的特例. 主要就是给出一个多维分布, 从条件分布来采样总比对联合分布的积分采样要容易.

背景:

- 1974, Besag 用来对空间数据分析
- 1984, Geman and Geman 基于贝叶斯思想引入到图像重构
- 1990, Gelfand and Smith 首次在贝叶斯统计中介绍.

此后, 大概没有其它方法比Gibbs Sampler更受关注和流行. 它相当简单且强大.

假设我们要从联合分布为$p(x, y)$的分布中采样k个x值. 从某个值y_0开始, 采样x使得$x_i \sim p(x|y = y_{i-1})$. 一旦x值计算出来, 则继续采样下一个y: $y_i \sim p(y|x = x_i)$.

两个参数A,B的分布, 其联合分布和边际分布如下

下面是一个简单例子

```
# 联合分布
jion=matrix(c(0.3,.2,.2,.1,.1,.1),nc=3,
```

	$A=1$	$A=2$	$A=3$	B的边际分布
$B=1$	0.3	0.2	0.1	0.6
$B=2$	0.2	0.1	0.1	0.4
A的边际分布	0.5	0.3	0.2	1

```
       dimnames=list(c("B=1","B=2"),c("A=1","A=2","A=3")))

> jion
   A=1 A=2 A=3
B=1 0.3 0.2 0.1
B=2 0.2 0.1 0.1

# B条件分布, P(B|A)
> B=apply(jion,2,function(x) x/sum(x));B
   A=1      A=2 A=3
B=1 0.6 0.6666667 0.5
B=2 0.4 0.3333333 0.5

# A条件分布, P(A|B)
> A=t(apply(jion,1,function(x) x/sum(x)));A
   A=1       A=2       A=3
B=1 0.5 0.3333333 0.1666667
B=2 0.5 0.2500000 0.2500000

# 从第一行第一列开始
a<-1 # a表示列
b<-1 # b表示行
margA<-NULL
margB<-NULL
gibbs<-matrix(0,nrow=2,ncol=3) # 采样个数
for(i in 1:50){
   b<-sample(c(1,2),1,prob=B[,a]))
   a<-sample(c(1,2,3),1,prob=A[b,]))
   margA<-append(margA,a)
   margB<-append(margB,b)
   gibbs[b,a]<-gibbs[b,a]+1
}
```

下面是一个二维正态分布的例子. 设 (X,Y) 服从均值为 $(0,0)$, 协方差矩阵为

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

的二维正态分布. 其条件分布为

$$P(X|Y=y) \sim N(\rho y, 1-\rho^2)$$
$$P(Y|X=x) \sim N(\rho x, 1-\rho^2)$$

下面是Gibbs采样的函数

```
# 参数x,y为第一个采样的值.
# n为采样个数
# rho为相关系数
gibbs2<-function(x,y, n, rho){
   m<-matrix(ncol=2,nrow=n)
   m[1,]<-c(x,y)
   for (i in 2:n){
      x<-rnorm(1,rho*y,sqrt(1-rho^2))
      y<-rnorm(1,rho*x,sqrt(1-rho^2))
      m[i,]<-c(x,y)
   }
   m
}
```

98.4 Metropolis-Hastings 方法

98.4.1 介绍

Metropolis-Hastings 方法产生一个马尔可夫链序列, 服从指定的分布(一般此分布很难直接采样). 这个序列可以用来逼近指定的分布(i.e., 直方图)或计算积分.

开始是针对Boltzmann distribution的方法, 叫做Nicholas Metropolis. W.K. Hastings 1970将其推广. Gibbs sampling是它的一个特例. 比较快, 应用方便, 但一般性不强.

Metropolis-Hastings方法可以从任一分布$P(x)$采样, 仅仅要求x处的密度可计算. 贝叶斯方法中归一化因子(分母的多重积分)很难计算, 故Metropolis-Hastings方法的最大优点之一就是不需要知道比例常数就可以采样.

Metropolis-Hastings方法产生一个马尔可夫链, 其x^{t+1}依赖于它的前一个值x^t. 此方法使用一个建议的密度函数$Q(x';x^t)$, 依赖于当前状态x^t, 来产生一个新的可能的采样值x'. 这个值被接受为下一个值$x^{t+1}=x'$, 如果α(服从[0,1]均匀分布)满足

$$\alpha < \frac{P(x')Q(x^t|x')}{P(xt)Q(x'|x^t)}$$

如果当前可能的样本x'被拒绝, 那么保留当前状态$x^{t+1}=x^t$.

例如, 建议的密度可以是高斯函数, 以当前状态x^t为其均值

$$Q(x';x^t) \sim N(x^t, \sigma^2 I)$$

给定x^t读取$Q(x';x^t)$作为x'的密度函数. 这个建议的密度会产生一个以当前状态x^t均值的, 方差为$\sigma^2 I$的正态分布随机数. 最初的Metropolis方法要求密度函数是对称的($Q(x;y)=Q(y;x)$). Hastings一般化此方法后去除了这个要求. 它还允许密度函数$Q(x';x^t)$完全不依赖于x^t, 这种情况下称为"Independence Chain Metropolis-Hastings"(相对于"Random Walk Metropolis-Hastings"). 给出一个合适的密度函数, 独立M-H方法可以产生比随机行走方法更高精度的结果, 但需要知道分布的一些先验信息.

98.4.2 步骤

假设最近的采样为x^t. 下一步使用$Q(x';x^t)$产生来产生新的备选随机数x', 并计算

$$\alpha = \alpha_1 \alpha_2$$

其中

$$\alpha_1 = \frac{P(x')}{P(x^t)}$$

为x'与x^t的似然比.

$$\alpha_2 = \frac{Q(x^t; x')}{Q(x'; x^t)}$$

为两个方向的建议密度比. 若密度函数对称, 此值为1. 新的状态x^{t+1}由下面的规则产生

$$\begin{aligned} If \quad \alpha >= 1: \quad x^{t+1} &= x' \\ else\ If \quad \alpha < 1: \quad x^{t+1} &= x'\ with\ probability\ \alpha \\ &= x^t\ with\ probability\ 1-\alpha \end{aligned}$$

马尔可夫链从一个随机数x^0开始. 算法首先计算很多步, 直到这个初始状态被忘记. 这些样本被抛弃, 此过程叫做"burn-in". 后面的值就可以作为采样值了.

当建议的密度函数形状与目标函数的形状一致时, 即$Q(x'; x^t) \approx P(x')$, 采样的效率最高(舍弃的最少). 但大部分情况下目标函数的形状是未知的. 当使用正态分布作为$Q(x'; x^t)$建议的分布时, 其方差参数σ^2需要在"burn-in"过程中调整. 这一般通过计算接受率来调整(最后的N个样本的接受率). 理论上一维高斯分布接受率接近50%, 多维的下降到大概23%. 如果σ^2太小, 链就会mix的很慢(接受率很高, 但在空间中移动很慢, 很难覆盖整个空间, 收敛就很慢). 反之, σ^2太大, 接受率就会很低.

98.5 例子

另外参考各函数的在线帮助.

```
#------MCMCpack包-----------
```

```
# 模拟正态分布采样. logfun=FALSE 表示使用密度函数. 否则使用对数密度函数.
library(MCMCpack)
ff2<-function(x) return(exp(-x^2))
m=MCMCmetrop1R(ff2,theta=0.2,logfun=F)
plot(m)
```

打印的接受率信息: The Metropolis acceptance rate was 0.70117

```
ff3<-function(x) return(-x^2)
m=MCMCmetrop1R(ff3,theta=0.2,logfun=T)
plot(m)
```

打印的接受率信息: The Metropolis acceptance rate was 0.70117

```
# error. Not run. 均匀分布好像不行. 对数与非对数函数均不可
ff<-function(x)if(x>=0&&x<=1)return(0)else return(-Inf)
m=MCMCmetrop1R(ff,theta=0.2)
ff1<-function(x)if(x>=0&&x<=1)return(1)else return(0)
m=MCMCmetrop1R(ff1,theta=0.2,logfun=F)
```

错误信息: Hessian from call to optim() not negative definite.
Sampling (as specified) cannot proceed.
错误: Check data and fun() and call MCMCmetrop1R() again.

```
#---mcmc包-------------------------
# 接受率在0.2-0.7之间都可以. 过高过低不行.
# 若过高过低, 使用参数 scale 来调整.

# 模拟均匀分布采样. 函数为对数密度函数
ff<-function(x)if(x>=0&&x<=1)return(0)else return(-Inf)

> r=metrop(ff,initial=0.2,nbatch=100)
> r$accept # 接受率还可以
[1] 0.38
> r=metrop(r,nbatch=100)
> plot(hist(r$batch)) # 看看是否均匀分布

# 模拟正态分布采样. 函数为对数正态分布函数
ff3<-function(x) return(-x^2)
```

```
> r=metrop(ff3,initial=0.2,nbatch=100)
> r$accept # 接受率还可以
[1] 0.64
> r=metrop(r,nbatch=1000)
> plot(hist(r$batch)) # 看看是否正态分布

#-------gibbs.met包--------
# log_f: 对数密度函数. no_var: 变量的个数. 即随机数的维数. ini_value: 初始值.
# iters: 模拟的次数. iters_met: 更新维数的迭代数.
# stepsizes_met: proposal分布的方差, 即随机漫步的速度.

# 模拟正态分布采样. 对数正态分布函数
ff3<-function(x) return(-x^2)

> library(gibbs.met)
> g=gibbs_met(log_f=ff3,no_var=1,ini_value=0.2,
          iters=400,iters_met=2,stepsizes_met=0.5)
> plot(hist(g))
```

Chapter 99

蒙特卡洛法计算积分

参考文献 [17] 14.4

对于形式复杂的被积函数, 解析方法往往很难, 一般的数值方法也困难. 蒙特卡洛方法总能比较简单的求出近似解及误差.

99.1 基本思想

考虑积分

$$I = \int_0^1 g(x)dx \qquad 0 <= g(x) <= 1$$

定义域和值域都是$[0,1]$. 但是$g(x)$很不规则. 积分I等于曲线$g(x)$下面的面积G.

设想均匀的往正方形$0 <= x <= 1, 0 <= y <= 1$内随机地投掷一个点. 该点的两个坐标在$[0,1]$均匀分布, 且相互独立. 这样, 点落入正方形内任意位置有相等的可能性. 于是落入区域G内的概率等于G的面积, 即积分I.

如果用某种方法产生两个均匀分布而又相互独立的随机变

量ξ,η的N组样本.对于每组(ξ_i,η_i), 若有

$$\eta_i < f(\xi_i)$$

点ξ_i,η_i落入区域G内. 计数落入区域G内的点的个数n, 根据大数定理, 当$N\to\infty$,

$$I = p = \lim_{N\to\infty}\frac{n}{N} \tag{99.1}$$

由此例子看到, 蒙特卡洛法解题有三个基本步骤

1. 构造或描述概率过程
2. 实现对已知概率过程的随机抽样
3. 建立与问题对应的估计量

蒙特卡洛法的理论基础是大数定理. 因此其应用理论上不受任何限制. 但是需要大量的试验, 故总是使用计算机完成.

99.2 频率法

99.2.1 方法简介

前面介绍了最简单的积分的例子. 现在讨论

$$I = \int_a^b g(x)dx$$

$g(x)$是区间$[a,b]$非负, 有界可积函数. 其极大值为M. 曲线下面面积为A.

抽取n个随机数对

$$(\xi_i,\eta_i) \qquad i = 1,2,\cdots,n$$

其中ξ是$[a,b]$均匀分布的随机数. η是$[0,M]$均匀分布的随机数

$$\xi_i = a + (b-a)r_i \qquad \eta_i = Mr_j$$

r_i, r_j是不同的$[0,1]$均匀分布的随机数. 若满足

$$\eta_i < g(\xi_i)$$

则点落入曲线下面面积A内.

设共m个点落入A, 当n充分大时, 有近似公式

$$I = \int_a^b g(x)dx \approx \frac{m}{n}M(b-a) \equiv I_n \tag{99.2}$$

可计算得积分I的近似I_n

事实上, 该算法相当于在框入曲线面积A的矩形$abcd$中均匀独立的投点, 假定落入A为成功, 则每次成功的概率p为面积A与矩形面积$abcd$之比

$$p = \frac{A}{(b-a)M}$$

大数定律知

$$\lim_{n \to infty} \frac{m}{n} = p$$

99.2.2 积分值均值和方差(误差)的估计

n次投点可以看作n次贝努里试验, 成功的概率为p, 成功的次数m服从参数n, p的二项分布. 其均值和方差为

$$E(m) = np \qquad \sigma^2(m) = np(1-p)$$

从而有

$$E(I_n) = E[m\frac{M(b-a)}{n}] = \frac{M(b-a)}{n}E(m) = \frac{M(b-a)}{n}np = A = I$$

$$\begin{aligned}\sigma^2(I_n) &= \sigma^2[m\frac{M(b-a)}{n}] = \frac{M^2(b-a)^2}{n^2}\sigma^2(m) \\ &= M^2(b-a)^2p(1-p)/n\end{aligned}$$

可见, I_n是积分值I的无偏估计, 误差是

$$\sigma(I_n) = M(b-a)\sqrt{\frac{p(1-p)}{n}}$$

p的真值是不知道的, n充分大时, $p \approx m/n$, 于是有

$$\sigma(I_n) \approx \frac{M(b-a)}{n}\sqrt{m(1-\frac{m}{n}}$$

99.2.3 误差与样本量

一个重要的问题是, 对于任意的给定的正数ε, 怎样才能保证积分近似值I_n与真实值之差的绝对值Δ小于ε的概率不小于$\alpha(0 < \alpha < 1)$. Δ可以表示为

$$\Delta = |I_n - I| = M(b-a)|\frac{m}{n} - p|$$

根据中心极限定理, 当n充分大, 随机变量

$$\frac{m-np}{\sqrt{np(1-p)}} \sim N(0,1)$$

因此, 有

$$\begin{aligned}P(\Delta < \varepsilon) &= P[M(b-a)|\frac{m}{n} - p| < \varepsilon] = P[\frac{|m-np|}{\sqrt{np(1-p)}} < t] \\ &= \Phi(t) - \Phi(-t) = 2\Phi(t) - 1\end{aligned}$$

其中

$$t = \frac{\varepsilon}{M(b-a)}\sqrt{\frac{n}{p(1-p)}}$$

于是问题转化为求出适当的t值, 满足

$$\alpha <= P(\Delta < \varepsilon) = 2\Phi(t) - 1$$

对于给定的α, 可以从累积标准正态分布表中查出满足

$$\alpha = 2\Phi(t_\alpha) - 1$$

的t_α值. 令$t = t_\alpha$, 代入t的表达式求得

$$\varepsilon_\alpha = t_\alpha M(b-a)\sqrt{\frac{p(1-p)}{n}}$$

这表示在置信水平α上, 积分计算值与真实值之差的上限为ε_α. 它与投点数n的根号成反比. 上式也可以写为

$$\varepsilon_\alpha = t_\alpha \sigma(I_n)$$

反过来, 为保证差小于给定正数ε, 应有$t >= t_\alpha$, 即投点数n必须满足

$$n >= \frac{t_\alpha^2 M^2 (b-a)^2 p(1-p)}{\varepsilon_\alpha^2}$$

其中p是未知的, 当n充分大的时候可用m/n作为p的近似值.

我们还可用利用被积函数的性质减小积分估计值的误差.

如果被积函数$g(x)$在积分区间$[a,b]$内存在大于0的极小值M', 那么M'下面的面积是一个矩形, 十分容易计算, 无须运用蒙特卡洛技巧. 积分可用分为两部分

$$\begin{aligned} I &= I' + \int_a^b [g(x) - M']dx \equiv I' + I'' \\ I' &= M'(b-a) \end{aligned}$$

前面求得的结果直接可用, 唯一的更动是将原式中的M改为$M - M'$, 由误差公式知道, 积分误差变小.

99.2.4 多重积分

均匀投点法原则上也适用于多重积分. 例如

$$I = \int_a^b g(x)dx = \int_{a_1}^{b_1}\int_{a_2}^{b_2}\cdots\int_{a_l}^{b_l} g(x_1,\cdots,x_l)dx_1\cdots dx_l$$

令$\xi = (\xi_1,\cdots,\xi_l)$为$[a,b]$区间的均匀分布的随机数, $\eta = Mr$是$[0,M]$区间内均匀分布的随机数, M是函数在区间$[a,b]$内的极大值. 选取n组ξ,η, 若其中满足

$$\eta < g(\xi)$$

的共有n组, 则当n充分大时,

$$I = \int_a^b g(x)dx \approx \frac{m}{n}M\prod_{i=1}^{l}(b_i - a_i) \equiv I_n$$

关于I_n的方差, I_n与积分真实值之差小于给定正数ε所要求的n值, 可用类似一维的方式讨论. 只要将一维公式中的$b-a$改为$\prod_{i=1}^{l}(b_i-a_i)$即可.

99.2.5 相对误差(精度)

实验工作者关心的通常是相对误差. 即积分值的误差与积分值的比. 给定置信水平α上, 积分的相对误差δ可表示为

$$\delta = t_\alpha\sqrt{\frac{1-p}{pm}} = \frac{t_\alpha}{\sqrt{n}}(\frac{1}{p}-1)^{1/2}$$

该公式对于一维或多维积分都正确. 它说明利用均匀投点法求积分的精度与两个因素有关.

1. 与投点数n的平方根成反比, n越大, 精度越好, 而且与积分的维数无关

2. 精度与被积函数的行为有关. $g(x)$变化越平坦, 投点成功率越大, p越接近1, 精度越好.

一般说来, 如果对于函数的行为无所了解, 尤其在高维积分的情况下, 投点成功的概率p往往很小, 这样精度就很差, 所以均匀投点法一般不适合求高维积分.

99.3 期望值估计法

任何积分都可用表示为某个随机变量的数学期望. 因此, 可用该随机变量的子样平均值作为积分的近似值.

设欲求积分为

$$I = \int_{V_s} g(x)dx$$

其中$x = (x_1, \cdots, x_s)$表示S维空间的点. 令$f(x)$为V_s上的任一随机变量ξ的概率密度函数

$$\int_{V_s} f(x)dx = 1$$

则积分可表示为随机变量$h(x) = g(x)/f(x)$的数学期望

$$I = \int_{V_s} g(x)dx = \int_{V_s} \frac{g(x)}{f(x)} f(x)dx = E[\frac{g(x)}{f(x)}] = E[h(x)]$$

当从随机变量ξ抽取容量为n的随机子样, 可求得随机变量$h(x)$的子样

$$h_i = h(\xi_i) = g(\xi_i)/f(\xi_i)$$

根据大数定律, 子样平均的期望与总体的数学期望相等. 当n充分大时, 有

$$I = E[h] = E(\hat{h}) \approx \frac{1}{n} \sum_{i=1}^{n} h_i \equiv I_n$$

选取$f(x)$最简单的方法是V_s内的均匀分布

$$f(x) = 1/V_s$$

这样, 上式简化为

$$I \approx I_n \equiv \frac{V_s}{n} \sum_{i=1}^{n} g(\xi_i)$$

对于一维的特殊情况, 若积分上下限为b,a, 则立即有[1]

$$I \approx I_n = \frac{b-a}{n} \sum_{i=1}^{n} g(\xi_i)$$

期望和方差为

$$E(I_n) = E(h) = I$$

$$V(I_n) = V[\frac{1}{n} \sum_{i=1}^{n} h_i] = \frac{1}{n} V_h$$

其中V_h为随机变量$h(x)$的方差

$$V_h = \int_{V_s} [h - E(h)]^2 f(x) dx = \int_{V_s} (h - I)^2 f(x) dx$$

根据同分布中心极限定理, n充分大时有

$$\frac{I_n - I}{\sqrt{V_h/n}} \sim N(0, 1)$$

利用此性质可用求得误差条件.

[1]文献中为$g(r_i)$, 怀疑应该为$g(\xi_i)$

99.4 TODO: 重要抽样方法

目标函数密度大的地方抽样多, 密度小的地方抽样少, 这种方法叫做重要抽样方法.

将期望值估计法中的均匀分布换为与目标函数形状接近的分布就可以实现重要抽样.

略.

99.5 TODO: 半解析法

某些被积函数的主要部分可以使用解析方法得到, 然后对剩余部分使用蒙特卡洛方法求积分, 可以大大降低方差.

略.

99.6 TODO: 自适应蒙特卡洛积分

99.7 例子

利用蒙特卡洛法计算积分

$$I = \int_0^1 e^x dx$$

此积分是解析可积的, 积分值为$I = 2.718281828$

产生10个随机数, 利用这10个随机数使用不同方法积分

```
n=100
M=exp(1) # M=2.71828为最大值
```

```
a=0
b=1
r=matrix(runif(2*n),nc=2)
xi=r[,1]*(b-a)+a
eta=r[,2]*M
g<-function(x) exp(x)

# 频率法
m=sum(eta<g(xi))# 合格的个数
I1=m*M*(b-a)/n; I1

# 期望值法
I2=sum(g(xi))/n;I2

# 重要抽样方法, 需要选择适当的概率密度函数f(x).
# 由于 e^x=1+x+\cdots , 选择 f(x)=(1+x)*3/2,
# 使用反函数法由U[0,1]求得服从f(x)分布的随机数
xi1=sqrt(1+3*r[,1])-1
f<-function(x) 2*(1+x)/3
I3=sum(g(xi1)/f(xi1))/n; I3

# 半解析法, 略

# 结果看到, 重要抽样方法的效果比较好. 半解析方法也比较好.
> I1=m*M*(b-a)/n; I1
[1] 1.685335
> I2=sum(g(xi))/n;I2
[1] 1.732034
> I3
[1] 1.723359
```

Chapter 100

马尔可夫链与生物学

参考 [47] chapter 10, 11, 12

参考 [11] 第五章 马尔可夫链数学模型

随机过程包括各种不同的过程. 两个主要的类别为到达时间过程和马尔可夫过程. 前者包括Bernoulli过程和Poisson过程.

100.1 马尔可夫过程

定义: 请参考相关教科书

下面是基因($5' - 3'$)随时间变化的情况([47] figure 10-2), 这就是一个马尔可夫链, 因为每一个的状态只与前一个有关, 而与前一个之前的所有状态无关. 虽然碱基之间可能互相影响, 但是我们的模型一般会忽略, 并假设碱基之间的变化互不影响(独立性).

```
ATCGCCATCGAATACTCTAGCATG t=0
ATCcCCATCGAATACTCTAGCATG t=1
ATCcCCAaCGAATACTCTAGCATG t=2
ATCcCCAaCGAATACcCTAGCATG t=3
ATCcCCATaGAATACgCTAcCATG t=4
```

100.2 转移图

略. 很有用, 尤其分析马尔可夫链的吸收性!!!

100.3 几个例子

100.3.1 动物健康

下面是一个马尔可夫链的例子([11] Page 163). 某动物群体, 患病是唯一死亡原因. 经验表明一日内健康个体患病的概率是0.01, 患病个体恢复健康的概率是0.9, 死亡的概率是0.01. 下面是单位时间(一天)转移概率矩阵

开始状态/到达状态	健康	患病	死亡
健康	0.99	0.01	0
患病	0.9	0.09	0.01
死亡	0	0	1

100.3.2 豌豆杂交(Aa基因型)

下面是杂交的例子([11] Page 165). 使用三种基因型AA,Aa,aa同杂交型Aa杂交, 其马尔可夫转移矩阵(即得到后代基因型的概率矩阵)为 表示为

状态/后代基因型	AA	Aa	aa
AA与Aa杂交	0.5	0.5	0
AA与Aa杂交	0.25	0.5	0.25
AA与Aa杂交	0	0.5	0.5

```
> Aa=matrix(c(0.5,0.25,0,0.5,0.5,0.5,0,0.25,0.5),nc=3,
      dimnames=list(c("AA*Aa","Aa*Aa","aa*Aa"),c("AA","Aa","aa")))
> Aa
       AA   Aa   aa
AA*Aa 0.50 0.25 0.00
Aa*Aa 0.25 0.25 0.25
aa*Aa 0.00 0.25 0.50
```

100.3.3 豌豆杂交(AA基因型)

如果三种基因型与AA杂交, 其转移矩阵具体值会改变为

```
# 三种基因型与AA杂交的转移矩阵
> AA=matrix(c(1,0.5,0,0,0.5,1,0,0,0),nc=3,
      dimnames=list(c("AA*AA","Aa*AA","aa*AA"),c("AA","Aa","aa")))
> AA
      AA  Aa aa
AA*AA 1.0 0.0  0
Aa*AA 0.5 0.5  0
aa*AA 0.0 1.0  0
```

假设开始群体基因型的比例为0.2,0.3,0.5

```
> x=matrix(c(0.2,0.3,0.5),nr=1,
      dimnames=list(c("ratio"),c("AA","Aa","aa")))
> x
      AA  Aa  aa
ratio 0.2 0.3 0.5
```

与基因型Aa杂交, 下一代的基因型分配比例为

```
# 注意 * 是不对的. 矩阵相乘应该使用 %*% 符号
```

```
# 下一代的基因型分配比例
> x2<-x%*%Aa; x2
       AA  Aa   aa
ratio 0.175 0.5 0.325
```

若继续与Aa杂交,则

```
> x3<-x2%*%Aa; x3 # 第三代基因型分配比例
        AA  Aa    aa
ratio 0.2125 0.5 0.2875
> x4<-x3%*%Aa; x4 # 第4代基因型分配比例
         AA  Aa     aa
ratio 0.23125 0.5 0.26875

......

> x9<-x8%*%Aa; x9 # 第9代基因型分配比例
           AA  Aa       aa
ratio 0.2494141 0.5 0.2505859
```

实际上, 可以看到, Aa 是转移一代的概率, $Aa\%*\%Aa$ 是转移2代的概率矩阵, 自乘 n 次的结果就是转移 n 代的概率矩阵. 有

```
# 效率非常低的矩阵连乘函数!!!
mulprod<-function(X,n){
   tmp<-X
   if(n>=2){
   for (i in 2:n){
   tmp<-tmp%*%X
}
}
   tmp
}
# 转移8代, x到第九代 x9
> x%*%mulprod(Aa,8)
```

```
          AA  Aa       aa
ratio 0.2494141 0.5 0.2505859
```

看看转移很多代的结果, 分配比例稳定下来了. 实际上其n代转移矩阵随着n的增大而平稳.

```
# 第21代基因型分配比例
> x%*%mulprod(Aa,20)
          AA  Aa       aa
ratio 0.2499999 0.5 0.2500001
# 第51代基因型分配比例
> x%*%mulprod(Aa,50)
      AA  Aa   aa
ratio 0.25 0.5 0.25

# 看看n代转移矩阵
# x %*% 此矩阵就是转移20代后的基因型分配比例
# x %*% 两次就是转移40代的基因型分配比例(顺便猜想一个提高mulprod()函数的方法?)
> mulprod(Aa,20)
           AA  Aa       aa
AA*Aa 0.2500005 0.5 0.2499995
Aa*Aa 0.2500000 0.5 0.2500000
aa*Aa 0.2499995 0.5 0.2500005
> mulprod(Aa,50)
       AA  Aa   aa
AA*Aa 0.25 0.5 0.25
Aa*Aa 0.25 0.5 0.25
aa*Aa 0.25 0.5 0.25
> mulprod(Aa,100)
       AA  Aa   aa
AA*Aa 0.25 0.5 0.25
Aa*Aa 0.25 0.5 0.25
aa*Aa 0.25 0.5 0.25
```

100.4 正则马尔可夫链(极限分布)

100.4.1 定理

对于马尔可夫链, 若存在正整数k使得其转移概率矩阵乘幂P^k的所有元素值都大于0, 则称该马尔可夫链是正则的(regular).

乘幂计算非常不方便, 一般利用转移图来查看. 如果任何状态经有限步可以到达任何其它状态, 则该马尔可夫链是正则的.

定理([11] Page 171, 有证明): 对于正则马尔可夫链的转移矩阵P, 有以下结论

1. 当 $t \to \infty$, $P^t \to W$(随机矩阵)
2. W的行向量均相同, 即

$$W = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \\ v_1 & v_2 & \cdots & v_n \\ \vdots & \vdots & \vdots & \vdots \\ v_1 & v_2 & \cdots & v_n \end{bmatrix}$$

3. W的所有分量都大于零

其中定理2说明, 随着时间延续, 无论初始状态分布如何, 最终将以一定的概率分布到达某一状态. 定理2的推论告诉我们, 分配比例将趋近稳定.

定理2的推论([11] Page 173, 有证明): 如果 $V = [v_1, \cdots, v_n]$ 是矩阵W的行向量, 那么

1. 对于任何随机向量 $X = [x_1, \cdots, x_n]$, 当 $t \to \infty$, 有 $XP^t \to V$
2. 存在唯一的随机向量V使得 $VP = V$, V亦称作随机矩阵P的不动点向量(stationary vector)

100.4.2 不动点向量的计算

推论给出了计算不动点向量的公式.

$$VP = V$$

方程组形式为

$$\begin{aligned} v_1P_{11} + v_2P_{21} + \cdots + v_nP_{n1} &= v_1 \\ v_1P_{12} + v_2P_{22} + \cdots + v_nP_{n2} &= v_2 \\ &\cdots \\ v_1P_{1n} + v_2P_{2n} + \cdots + v_nP_{nn} &= v_n \\ v_1 + \cdots + v_n &= 1 \end{aligned}$$

可以变换为

$$\begin{aligned} v_1(P_{11} - 1) + v_2P_{21} + \cdots + v_nP_{n1} &= 0 \\ v_1P_{12} + v_2(P_{22} - 1) + \cdots + v_nP_{n2} &= 0 \\ &\cdots \\ v_1P_{1n} + v_2P_{2n} + \cdots + v_n(P_{nn} - 1) &= 0 \\ v_1 + \cdots + v_n &= 1 \end{aligned}$$

前n个方程式是奇异的, 从其中去掉一个, 例如, 选择去掉第一个方程式, 然后与最后一个约束构成方程组, 解此方程组即得到不动点向量.

根据上面的公式及思路, 我们编写函数求解不动点向量

```
# 求解不动点向量的函数, P为转移矩阵(不需要转置)
SV<-function(P){
   d<-dim(P)[1]
   for (i in 1:d){
      P[i,i]<-P[i,i]-1
}
   P1<-t(cbind(P[,2:d],rep(1,d)))
   b<-c(rep(0,d-1),1)
   v<-solve(P1,b)
   v
}
```

现在看与Aa杂交的例子. 其不动点为

```
# 再次写出转移矩阵
Aa=matrix(c(0.5,0.25,0,0.5,0.5,0.5,0,0.25,0.5),nc=3,
       dimnames=list(c("AA*Aa","Aa*Aa","aa*Aa"),c("AA","Aa","aa")))

> SV(Aa)
[1] 0.25 0.50 0.25
```

说明多次与Aa杂交, 最终基因型分配比例将稳定在$\frac{1}{4},\frac{1}{2},\frac{1}{4}$

100.5 Hardy-Weiberg定理

参考文献 [22] Page 25 [11] Page 176-178

英国数学家Hardy与德国医生Weiberg于1908年分别发现. 并由Punett于1950年与Stem于1943年的论文中分别介绍. 现在已经被公认为群体遗传学的创始理论

100.5.1 定理

设某群体有n个个体. 其中三种基因型AA,Aa,aa的个体数量分别为x,y,z, $x+y+z=n$. 基因型频率的估计可以是

$$d=\frac{x}{n}$$

$$h=\frac{y}{n}$$

$$r=\frac{z}{n}$$

$$d+h+r=1$$

群体中等位基因A与a的频率分别用p, q表示. (其中p=A的个数/(A的个数+a的个数), q=a的个数/(A的个数+a的个数)), 那么

$$p = d + \frac{h}{2}$$

$$q = r + \frac{h}{2}$$

$$p + q = 1$$

那么随机交配第二代产生基因型AA的频率为p^2, Aa为$2pq$, aa为q^2

如果我们描述为: 群体(继代)随机交配, 在不产生选择, 突变和迁移的情况下, 基因频率与基因型频率每代保持不变, 合子系列频率等于配子系列频率的二项式平方

$$AA\ p^2 + Aa\ 2pq + aa\ q^2 = (Ap + aq)^2$$

这就是Hardy-Weiberg定理.

实际上可以这样列表多个基因亦可用平方式展开.

交配/后代基因型	AA	Aa	aa
$AA * AA$	d^2		
$AA * Aa$	$dh/2$	$dh/2$	
$AA * aa$		dr	
$Aa * AA$	$dh/2$	$dh/2$	
$Aa * Aa$	$h^2/4$	$h^2/2$	$h^2/2$
$Aa * aa$		$hr/2$	$hr/2$
$aa * AA$		dr	
$aa * Aa$		$hd/2$	$hd/2$
$aa * aa$			r^2
	$(d + h/2)^2$ p^2	$2(d + h/2)(h/2 + r)$ $2pq$	$(h/2 + r)^2$ q^2

转移矩阵可以写作

$$P = \begin{bmatrix} d + h/2 & r + h/2 & 0 \\ d/2 + h/4 & d/2 + h/2 + r/2 & r/2 + h/4 \\ 0 & d + h/2 & r + h/2 \end{bmatrix} = \begin{bmatrix} p & q & 0 \\ p/2 & 1/2 & q/2 \\ 0 & p & q \end{bmatrix}$$

解得不动点

$$V = [p^2, 2pq, q^2]$$

100.5.2 复等位基因

复等位基因的随机交配下的合子的频率使用下式

$$A_1p_1 + \cdots + A_np_n)^2 = A_1A_1p_1^2 + A_1A_2 2p_1p_2 + \cdots + A_nA_np_n^2$$

纯合体基因型频率为p_i^2, 杂合体基因型频率为$2p_ip_j$

100.5.3 例子

设基因型频率$d = 0.3, h = 0.2, r = 0.5$, A的基因频率为$p = d + h/2 = 0.4$, a的基因频率为$q = r + h/2 = 0.6$, 则转移矩阵

```
# 转移矩阵
P=matrix(c(0.4,0.2,0,.6,.5,.4,0,.3,.6),nc=3)

> P
     [,1] [,2] [,3]
[1,]  0.4  0.6  0.0
[2,]  0.2  0.5  0.3
[3,]  0.0  0.4  0.6
# 不动点向量为
> SV(P)
[1] 0.16 0.48 0.36

# 初始分配
x=c(0.3,0.2,0.5)


> x%*%P  # 一次转移(第一代)后已经达到不动点
     [,1] [,2] [,3]
[1,] 0.16 0.48 0.36
> x%*%P%*%P # 二次转移
```

```
     [,1] [,2] [,3]
[1,] 0.16 0.48 0.36
```

100.6 吸收马尔可夫链

(证明及其它证明参考[11] 第五章相关部分)

100.6.1 吸收状态

马尔可夫链中若状态i(转移矩阵第i行)满足

1. $p_{ii} = 1$
2. $p_{ik} = 0, \quad k \neq i$

称该状态为吸收状态(absorbing state). 即不能离开的状态, 例如死亡, 称为吸收状态. 在转移图中表现就是到自身的值为1的弧.

100.6.2 吸收马尔可夫链

满足下面条件的马尔可夫链称吸收马尔可夫链.

1. 至少存在一个状态为吸收状态
2. 从任何状态经有限步可以到达吸收状态

实际上转移图中表示就是从任何其它状态可以到达某个吸收状态.

非吸收状态称转移状态(transient state)

100.6.3 规范的转移矩阵写法

一般将有r个吸收状态, k个转移状态的吸收马尔可夫链转移矩阵写作

$$P = \begin{bmatrix} E & O \\ R & Q \end{bmatrix}$$

其中

- E: $r * r$单位矩阵
- O: $r * k$零矩阵
- R: $k * r$矩阵, 表示从转移状态一步就达到吸收状态的概率
- O: $k * k$矩阵, 从一个转移状态到另一个转移状态的概率

100.6.4 定理: 最终进入吸收状态的概率

对于吸收马尔可夫链, 从任何状态出发最终进入吸收状态的概率为1

100.6.5 转移矩阵的幂

转移矩阵的t次幂写作

$$P = \begin{bmatrix} E & O \\ R_t & Q^t \end{bmatrix}$$

其中 $R_t = (E + Q + Q^2 + \cdots + Q^{t-1})R$

对于转移矩阵的幂有以下结论

1. $t \to \infty$时, $Q^t \to O$, O为零矩阵

2. 矩阵$E-Q$可逆. 此处 E 为k阶单位矩阵, 与Q阶数相同

3. $N=(E-Q)^{-1}=E+Q+Q^2+\cdots$, 为k阶方阵, 称为该吸收马尔可夫链的基本矩阵(fundamental matrix)

100.6.6 定理: 进入次数的数学期望

具有r个吸收状态的吸收马尔可夫链, 从转移状态$i_1=r+i$开始, 到达吸收状态前, 进入指定转移状态$j_1=r+j$的次数的数学期望是基本矩阵$N=(E-Q)^{-1}$的第i行第j列.

推论: 具有r个吸收状态的吸收马尔可夫链, 从转移状态$i_1=r+i$开始, 到达吸收状态前, 在所有转移状态之间传递的步数的数学期望是基本矩阵$N=(E-Q)^{-1}$的第i行之和.

100.6.7 例子: 豌豆杂交

我们使用与AA基因型杂交的例子[1], 那么它就是一个吸收马尔可夫链. 再次写出转移矩阵

```
# 三种基因型与AA杂交的转移矩阵, Q就是右下4个值
AA=matrix(c(1,0.5,0,0,0.5,1,0,0,0),nc=3,
      dimnames=list(c("AA*AA","Aa*AA","aa*AA"),c("AA","Aa","aa")))
Q=AA[2:3,2:3]
```

```
> AA
      AA  Aa aa
AA*AA 1.0 0.0  0
Aa*AA 0.5 0.5  0
aa*AA 0.0 1.0  0
> Q
      Aa aa
Aa*AA 0.5  0
aa*AA 1.0  0
```

[1]例子描述见前面. 更多例子参考[11]

```
# 再次引用乘幂函数. 效率非常低的矩阵连乘函数!!!
mulprod<-function(X,n){
   tmp<-X
   if(n>=2){
   for (i in 2:n){tmp<-tmp%*%X }}
   tmp}

# 转移矩阵乘幂
> mulprod(AA,5)
         AA     Aa aa
AA*AA 1.00000 0.00000  0
Aa*AA 0.96875 0.03125  0
aa*AA 0.93750 0.06250  0
> mulprod(AA,10)
           AA          Aa aa
AA*AA 1.0000000 0.0000000000  0
Aa*AA 0.9990234 0.0009765625  0
aa*AA 0.9980469 0.0019531250  0
> mulprod(AA,100)
     AA          Aa aa
AA*AA  1 0.000000e+00  0
Aa*AA  1 7.888609e-31  0
aa*AA  1 1.577722e-30  0
> mulprod(AA,1000)
     AA           Aa aa
AA*AA  1  0.000000e+00  0
Aa*AA  1 9.332636e-302  0
aa*AA  1 1.866527e-301  0
> mulprod(AA,10000)
     AA Aa aa
AA*AA  1  0  0
Aa*AA  1  0  0
aa*AA  1  0  0

# Q乘幂
> mulprod(Q,10)
              Aa aa
Aa*AA 0.0009765625  0
aa*AA 0.0019531250  0
> mulprod(Q,100)
              Aa aa
```

```
Aa*AA 7.888609e-31  0
aa*AA 1.577722e-30  0
> mulprod(Q,1000)
              Aa aa
Aa*AA 9.332636e-302  0
aa*AA 1.866527e-301  0
> mulprod(Q,10000)
     Aa aa
Aa*AA  0  0
aa*AA  0  0
```

下面计算基本矩阵N并分析之. 可以看到:

- 从状态Aa开始进入吸收状态AA在非吸收状态停留的总次数为2+0=2次, 即从Aa开始, 经过2步大多就纯化了.
- 从状态aa开始进入吸收状态AA在非吸收状态停留的总次数为2+1=3次, 即从aa开始, 经过3步大多就获得显性性状.

```
# 计算基本矩阵N
> E=diag(c(1,1)); E
     [,1] [,2]
[1,]    1    0
[2,]    0    1
# 基本矩阵.

> N=solve(E-Q); N
   Aa*AA aa*AA
Aa     2     0
aa     2     1
```

100.6.8 例子: 动物健康

我们写出转移矩阵[2]. 从基本矩阵可以看到

[2]例子描述见前面

- 由good状态出发的寿命(即到dead)平均为$9100+100=9200$(天)
- 由ill状态出发的寿命平均为$9000+100=9100$(天)

```
P=matrix(c(1,0.,0.01,0,0.99,0.9,0,0.01,0.09),nc=3,
       dimnames=list(c("dead","good","ill"),c("dead","good","ill")))

# 基本矩阵
> N=solve(diag(c(1,1))-P[2:3,2:3]);N
     good ill
good 9100 100
ill  9000 100
```

100.6.9 多个吸收状态

具有$r(r>1)$个吸收状态的吸收马尔可夫链, 从转移状态$i_1=r+i$开始, 最终进入第j个吸收状态的概率是矩阵$B=NR$的第i行第j列元素值.

下面是一个多吸收状态的例子. 其中

- "dead1" 其它死亡
- "dead2" 呼吸疾病死亡
- "dead3" 循环疾病死亡
- "good" 健康
- "ill1" 呼吸疾病
- "ill2" 循环疾病

```
P=matrix(c(1,0.,0, 0.001,0,0,
          0,1,0,0,0.2,0,
          0,0,1,0,0,0.1,
          0,0,0,0.889,0.7,0.8,
```

```
          0,0,0,0.01,0.1,0,
          0,0,0,0.1,0,0.1),nc=6,
       dimnames=list(
          c("dead1","dead2","dead3","good","ill1","ill2"),
          c("dead1","dead2","dead3","good","ill1","ill2")))

> R=P[4:6,1:3];R
     dead1 dead2 dead3
good 0.001   0.0   0.0
ill1 0.000   0.2   0.0
ill2 0.000   0.0   0.1
> Q=P[4:6,4:6];Q
      good ill1 ill2
good 0.889 0.01  0.1
ill1 0.700 0.10  0.0
ill2 0.800 0.00  0.1
> E=diag(c(1,1,1))
> N=solve(E-Q); N
        good     ill1    ill2
good 69.76744 0.7751938 7.751938
ill1 54.26357 1.7140396 6.029285
ill2 62.01550 0.6890612 8.001723
> B=N%*%R; B
         dead1     dead2     dead3
good 0.06976744 0.1550388 0.7751938
ill1 0.05426357 0.3428079 0.6029285
ill2 0.06201550 0.1378122 0.8001723
```

对于B的分析表明, 从健康开始, 疾病3的死亡概率最大. (应该引起谁的注意?)

对疾病矩阵N的分析表明, 从健康开始, 其寿命期望为$69.77 + 0.78 + 7.75 = 78.29$岁

若考虑生育增长, 则需要带输入的马尔可夫链

100.7 带输入的马尔可夫链

100.7.1 水塘氮循环的例子

考虑一个水塘, 鱼吃水藻, 水藻从水中吸收氮, 鱼排泄与水藻生长中排除部分氮. 鱼可能捕捞卖掉, 水藻可能溢出池塘(或打捞). 那么此系统的氮的转移矩阵为

```
 \begin{verbatim}
P=matrix(c(1,0,0, 0,0.75,
           0,1,0,0.2,0,
           0,0,0.5,0.1,0.125,
           0,0,0.5,0.2,0,
           0,0,0,0.5,0.125),nc=5,
       dimnames=list(
           c("catch","out","water","plant","fish"),
           c("catch","out","water","plant","fish")))
```

```
> P
      catch out water plant  fish
catch  1.00 0.0 0.000   0.0 0.000
out    0.00 1.0 0.000   0.0 0.000
water  0.00 0.0 0.500   0.5 0.000
plant  0.00 0.2 0.100   0.2 0.500
fish   0.75 0.0 0.125   0.0 0.125
```

每单位时间在转移状态上补充氮肥, 设每年向水中投入有效氮肥80kg, 则输入向量为 $F = [80, 0, 0]$

我们有如下定理

100.7.2 定理: 转移向量的极限

若带有输入的马尔可夫链, 输入向量为F, 则其状态向量序列的转移部分存在极限FN, 其中N为转移矩阵的基本矩阵

```
> Q=P[3:5,3:5]; Q
      water plant  fish
water 0.500   0.5 0.000
plant 0.100   0.2 0.500
fish  0.125   0.0 0.125
> E=diag(c(1,1,1))
> N=solve(E-Q); N # 基本矩阵, 转入吸收前停留的时间
         water    plant     fish
water 2.5454545 1.5909091 0.9090909
plant 0.5454545 1.5909091 0.9090909
fish  0.3636364 0.2272727 1.2727273
> R=P[3:5,1:2]
> B=N%*%R; B # 最终三个转移状态转入吸收状态的概率
         catch       out
water 0.6818182 0.31818182
plant 0.6818182 0.31818182
fish  0.9545455 0.04545455
> F=c(80,0,0)
> F%*%N # 最终三个状态稳定的氮含量
       water   plant    fish
[1,] 203.6364 127.2727 72.72727

# 若每条鱼的含氮量为0.02kg, 则每年投入80kg的有效氮最多能够养的鱼为3636条
> (F%*%N)[3]/0.02
[1] 3636.364
```

Part XII

Bayes方法

Chapter 101

总论

参考 [2]

贝叶斯统计推断理论源于英国作者贝叶斯(Thomas Bayes)于1763年在英国皇家学会哲学学报上发表的论文《论机会学说中一个问题的求解》(An Essay Toward Solving a Problem in the Doctrine of Chances), 该文提出了一种归纳推理的理论, 被一些统计学家发展为系统的统计推断方法, 简称贝叶斯方法.

20世纪50年代后, 通过De Finetti, Savaga, Raiffa, Schlaifer, Jefferys, Good等统计学家大量的开拓性的工作, 贝叶斯统计推断理论获得了迅速发展和完善. 目前, 在欧美等西方国家, 贝叶斯统计推断已经成为与经典统计学派并驾齐驱的当今两大学派之一. 并且, 英国统计学家Lindely认为21世纪是贝叶斯统计的世界. 目前贝叶斯统计理论在可靠性工程, 风险管理工程, 经济预测决策及生物统计学等诸多领域均获得广泛应用.

实际上贝叶斯方法是一个很大的论题. 大部分的统计学论题可以使用贝叶斯的思想来描述. HMM, 神经网络等可以涵盖在一种叫做概率图的广义的贝叶斯推断与贝叶斯网络的概念下.

其它部分也分散一些关于贝叶斯的讨论.

101.1 介绍

经典的频率理论: 参数为常数. 使用点估计和区间估计

贝叶斯理论: 参数为随机变量. 先验分布+模型=后验分布. 所有信息在后验分布中.

如果θ为待估计参数/参数向量, 有先验分布$\pi(\theta)$(根据经验或共轭理论), 观测数据y, 认为来自条件分布$p(y|\theta)$. 我们想通过y推测最可能产生观测y的参数, 即已知y后参数的后验分布. 那么θ的后验分布为

$$\pi(\theta|y) = \frac{p(y|\theta)\pi(\theta)}{\int_{all\ \theta} p(y|\theta)\pi(\theta)d\theta}$$

贝叶斯统计的一般步骤: 选择待估计参数的先验分布, 计算与先验分布与观测数据结合后的后验概率, 选择使后验概率最大的参数值作为估计, 并由后验分布估计其区间.

如何选择先验分布: 1. 无信息先验分布: $p(\theta)$正比与某常数. 2. 先验分布和后验分布也该有相同的形式, 即属于同一分布族. 称为后验分布的共轭分布.

如何计算后验分布: 1. 直接计算. 如果后验概率的形式明显的话. 2. 否则使用采样技术模拟. 最常用的是 MCMC Gibbs 采样和 Metropolis- Hasting Algorithm

101.2 R的贝叶斯相关包介绍

101.2.1 一般模型

arm包: 包括使用lm,glm,mer,polr等对象进行贝叶斯推断的R函数

BACCO: 随机函数的贝叶斯分析. 包含3个子包: emulator, calibrator, and approximator, 进行贝叶斯估计和评价计算机程序.

bayesm: 市场与微经济分析模型的许多贝叶斯推断函数. 模型包括线性回归, 多项式logit, 多项式probit, 多元probit, 多元混合normals(包括聚类), 密度估计-使用有限混合正态模型与Dirichlet先验过程, 层次线性模型, 层次多元logit, 层次负二项回归模型, 线性工具变量模型(linear instrumental variable models).

bayesSurv: 生存回归模型的贝叶斯推断.

DPpackage: 贝叶斯非参数和半参数模型. 现在还包括密度估计, ROC曲线分析, 区间一致数据, 二项回归模型, 广义线性模型和IRT类型模型的半参数方法.

MCMCpack: 特定模型的MCMC模拟算法, 广泛用于社会和行为科学. 拟合很多回归模型的R函数. 生态学模型推断. 还包括一个广义Metropolis采样器, 适合任何模型.

mcmc: 随机行走Metropolis算法, 对于连续随机向量.

101.2.2 特殊模型和方法

AdMit: 拟合适应性混合t分布拟合目标密度使用核函数.

bark: 实现(Bayesian Additive Regression Kernels)

BayHaz: 贝叶斯估计smooth hazard rates, 通过 Compound Poisson Process (CPP) 先验概率.

bayesGARCH: 贝叶斯估计 GARCH(1,1) 模型, 使用t分布.

BAYSTAR: 贝叶斯估计 threshold autoregressive models

BayesTree: implements BART (Bayesian Additive Regression Trees) by Chipman, George, and McCulloch (2006).

BCE: 从生物注释数据中估计分类信息.

bcp: a Bayesian analysis of changepoint problem using the Barry and Hartigan product partition model.

BMA:

BPHO: 贝叶斯预测高阶相互作用, 使用slice 采样技术.

bqtl: 拟合 quantitative trait loci (QTL) 模型.可以估计多基因模型, 使用拉普拉斯近似. 基因座内部映射(interval mapping of genetic loci).

bim: 贝叶斯内部映射, 使用MCMC方法.

bspec: 时间序列的离散功率谱贝叶斯分析

cslogistic: 条件特定的logistic回归模型(conditionally specified logistic regression model)的贝叶斯分析.

deal: 逆运算网络分析: 当前版本覆盖离散和连续的变量, 在正态分布下.

dlm: 贝叶斯与似然分析动态信息模型. 包括卡尔曼滤波器和平滑器的计算, 前向滤波后向采样算法.

EbayesThresh: thresholding methods 的贝叶斯估计. 尽管最初的模型是在小波下开发的, 当参数集是稀疏的, 用户也可以受益.

eco: 使用MCMC方法拟合贝叶斯生态学推断 in two by two tables

evdbayes: 极值模型的贝叶斯分析.

exactLoglinTest: log-linear models 优度拟合检验的条件P值的MCMC估计.

HI: transdimensional MCMC 方法几何途径, 和随机多元 Adaptive Rejection Metropolis Sampling.

G1DBN: 动态贝叶斯网络推断.

Hmisc内的gbayes()函数, 当先验和似然都是正态分布, 导出后验(且最优)分布, 且当统计量来自2-样本问题.

geoR包的krige.bayes()函数地理统计数据的贝叶斯推断, 允许

不同层次的模型参数的不确定性.

geoRglm 包的 binom.krige.bayes() 函数进行贝叶斯后验模拟, 二项空间模型的空间预测.

MasterBayes: MCMC方法整合家谱数据(由分子和形态数据得来的)

lme4包的mcmcsamp()函数信息混合模型和广义信息混合模型采样.

lmm: 拟合信息混合模型, 使用MCMC方法.

MNP: 多项式probit模型, 使用MCMC方法.

MSBVAR: 估计贝叶斯向量自回归模型和贝叶斯结构向量自回归模型.

pscl: 拟合item-response theory模型, 使用MCMC方法, 且计算beta分布和逆gamma分布的最高密度区域

RJaCGH: CGH微芯片的贝叶斯分析, 使用hidden Markov chain models. 正态数目的选择根据后验概率, 使用 reversible jump Markov chain Monte Carlo Methods 计算.

sna: 社会网络分析, 包含函数用于从Butt's贝叶斯网络精确模型, 使用MCMC方法产生后验样本.

tgp: 实现贝叶斯 treed 高斯过程模型: 一个空间模型和回归包提供完全的贝叶斯MCMC后验推断, 对于从简单线性模型到非平稳treed高斯过程等都适合.

Umacs: Gibbs采样和Metropolis algorithm的贝叶斯推断.

vabaye1Mix: 高斯混合模型的贝叶斯推断, 使用多种方法.

101.2.3 Post-estimation tools

BayesValidate: 实现了对贝叶斯软件评估的方法.

boa: MCMC序列的诊断, 描述分析与可视化. 导入BUGS格式的绘图. 并提供 Gelman and Rubin, Geweke, Heidelberger and Welch, and Raftery and Lewis 诊断. Brooks and Gelman 多元收缩因子.

coda: (Convergence Diagnosis and Output Analysis) MCMC的收敛性分析, 绘图等. 可以轻松导入 WinBUGS, OpenBUGS, and JAGS 软件的MCMC输出. 亦包括 Gelman and Rubin, Geweke, Heidelberger and Welch, and Raftery and Lewis 诊断.

mcgibbsit: 提供 Warnes and Raftery MCGibbsit MCMC 诊断. 作用于mcmc对象上面.

ramps: 高斯过程的贝叶斯几何分析, 使用重新参数化和边际化的后验采样算法.

rv: 基于模拟的随机变量类, 后验模拟对象可以方便的作为随机变量来处理.

scapeMCMC: 处理年龄和时间结构的人群模型贝叶斯工具. 提供多种MCMC诊断图形, 可以方便的修改参数.

101.2.4 学习贝叶斯的包

BaM: Jeff Gill's book, "Bayesian Methods: A Social and Behavioral Sciences Approach, Second Edition" (CRC Press, 2007). 伴随的包

Bolstad: 此书的包. Introduction to Bayesian Statistics, by Bolstad, W.M. (2007). 的包

LearnBayes: 学习贝叶斯推断的很多的函数. 包括1个,2个参数后验分布和预测分布, MCMC算法来描述分析用户定义的后验分布. 亦包括回归模型, 层次模型. 贝叶斯检验, Gibbs采样的实例.

101.2.5 其它软件与R的接口

bayesmix: JAGS 软件, 贝叶斯混合模型.

BRugs: windows 系统下的 OpenBUGS 接口.

R2WinBUGS 提供windows和linux的 WinBUGS 的接口. linux下安装openbugs, 设置 bugs() 函数参数 program=”openbugs”.

rbugs: 支持 OpenBUGS 的linux接口(LinBUGS)

rjags, R2jags, and runjags: 都提供 Just Another Gibbs Sampler (JAGS) 接口

gR: BUGS引擎的图形接口部分.

Chapter 102

几个后验概率形式可以推导的例子

参考文献: Peng Ding *Bayesian Statistics and R* December 16, 2008. 一个ppt

使用包 MCMCpack

102.1 二项分布

推测二项分布的参数p. n为试验次数.

模型为

$$Pr(y|p) \propto p^y(1-p)^{n-y}$$

p的先验概率为

$$Pr(p) \propto p^{\alpha-1}(1-p)^{\beta-1}$$

后验概率为

$$Pr(p|y) \propto p^{\alpha+y-1}(1-p)^{\beta+n-y-1}$$

例如, 试验12次, 成功3次. 估计参数p

```
library(MCMCpack)
p <-MCbinomialbeta(y=3,n=12,alpha=1,beta=1,mc=5000)
summary(p)
plot(p)

# 从p的后验概率分布中抽样5000次, 计算其均值与百分位数作为其点和区间估计
> p
Iterations = 1:5000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 5000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

        Mean             SD       Naive SE Time-series SE
    0.288060       0.116299       0.001645       0.001727

2. Quantiles for each variable: # 作为其95%区间估计

   2.5%     25%     50%     75%   97.5%
0.09177 0.20232 0.27752 0.36524 0.53607
```

102.2 泊松分布

推测泊松分布的参数λ.

模型为

$$Pr(y|\lambda) \propto \prod_{i=1}^{n} \lambda^{y_i} e^{-\lambda}$$

λ先验概率为

$$Pr(\lambda) \propto e^{\beta\lambda} \lambda^{\alpha-1}$$

后验概率为

$$\lambda|y \propto Gamma(\alpha + n\bar{y}, \beta + n)$$

```
y<-rpois(1000,lambda=2)
posterior <- MCpoissongamma(y, 15, 1, 5000)
summary(posterior)
plot(posterior)

> posterior
Iterations = 1:5000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 5000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

        Mean             SD       Naive SE Time-series SE
   2.0322322      0.0461749      0.0006530      0.0006996

2. Quantiles for each variable:

 2.5%   25%   50%   75% 97.5%
1.942 2.001 2.032 2.063 2.124

# 频率理论的估计
> mean(y)
[1] 2.019
```

102.3 正态分布-方差已知

推测正态分布参数均值μ

模型为

$$Pr(y|\mu) \propto exp(-\frac{1}{2\sigma^2}\sum(y_i - \mu)^2)$$

μ先验概率为

$$Pr(\mu) \propto exp(-\frac{1}{2\tau_0^2(\mu-\mu_0)^2})$$

后验概率为

$$\mu|y \sim N(\mu_1, \tau_1^2)$$

其中

$$\mu_1 = \frac{\mu_0/\tau_0^2 + n\bar{y}/\sigma^2}{1/\tau_0^2 + 1/\sigma^2}, \frac{1}{\tau_1^2} = \frac{1}{\tau_0^2} + \frac{1}{\sigma^2}$$

```
y<-rnorm(1000,5,1)
posterior <-MCnormalnormal(y, sigma2=1, mu0=0,tau20=100, mc=5000)
summary(posterior)
plot(posterior)

> posterior
Iterations = 1:5000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 5000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

        Mean             SD       Naive SE Time-series SE
   5.0168337      0.0313200      0.0004429      0.0004538

2. Quantiles for each variable:

 2.5%   25%   50%   75% 97.5%
4.956 4.996 5.016 5.038 5.078

> mean(y)
[1] 5.017568
```

102.4 正态分布-方差未知

推测正态分布参数均值和方差.

模型为

$$Pr(y|\mu) \propto exp(-\frac{1}{2\sigma^2}\sum(y_i-\mu)^2)$$

先验概率为(semi-congugate)

$$\mu \sim N(b_0, B_0^2)$$
$$\sigma^2 \sim Inverse \quad \chi^2(2c_0, 2d_0)$$

```
y<-rnorm(1000,2,1)
posterior<-
MCMCregress(y~1, b0 = 0, B0 =0, c0= 0.001, d0 = 0.001)
summary(posterior)
plot(posterior)

> summary(posterior)

Iterations = 1001:11000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 10000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

            Mean     SD  Naive SE Time-series SE
(Intercept) 2.0080 0.03084 0.0003084      0.0003228
sigma2      0.9633 0.04324 0.0004324      0.0004045

2. Quantiles for each variable:

            2.5%    25%    50%    75% 97.5%
(Intercept) 1.948 1.9870 2.0080 2.0288 2.069
sigma2      0.882 0.9334 0.9624 0.9915 1.051
```

```
> mean(y)
[1] 2.007813
> var(y)
[1] 0.9617382
```

102.5 多元dirichlet分布

模型为

$$Pr(y|\theta) \propto \prod_{i=1}^{n} \theta_i^{y_i}$$

先验概率为

$$Pr(\theta|\alpha) \propto \prod_{i=1}^{n} \theta_i^{\alpha_i - 1}$$

```
posterior <-MCmultinomdirichlet(c(727,583,137), c(1,1,1), mc=10000)
summary(posterior)
plot(posterior)

> summary(posterior)

Iterations = 1:10000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 10000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

       Mean      SD  Naive SE Time-series SE
pi.1 0.50194 0.013016 1.302e-04      1.353e-04
pi.2 0.40293 0.012778 1.278e-04      1.207e-04
```

```
pi.3 0.09513 0.007713 7.713e-05      8.031e-05

2. Quantiles for each variable:

       2.5%     25%     50%    75%  97.5%
pi.1 0.47699 0.49316 0.50178 0.5106 0.5280
pi.2 0.37717 0.39424 0.40289 0.4117 0.4278
pi.3 0.08072 0.08981 0.09491 0.1002 0.1106
```

102.6 广义线性模型

此时函数对回归系数做估计.

- $E(y|x) = g^{-1}(\beta^T x)$
- logistic 回归: $\log(\frac{Pr(y=1|x)}{1-Pr(y=1|x)}) = \beta^T x$
- probit 回归 $\Phi^{-1}(Pr(y=1|x)) = \beta^T x$
- 泊松回归: $\log(E(y|x)) = \beta^T x$

```
# 线性模型
X<-rnorm(100,2,1)
Y<-1+2*X+rnorm(100,0,1)
posterior <- MCMCregress(Y~X,b0 = 0, B0 = 0,
       c0 = 0.001, d0 = 0.001,verbose=1000)
plot(posterior)
summary(posterior)

> summary(posterior)

Iterations = 1001:11000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 10000
```

```
1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:
# 下面是模型的系数
           Mean     SD  Naive SE Time-series SE
(Intercept) 1.234 0.22190 0.0022190      0.002355
X          1.877 0.09667 0.0009667      0.001113
sigma2      1.111 0.16325 0.0016325      0.001823

2. Quantiles for each variable:
# 百分位数
            2.5%    25%   50%   75% 97.5%
(Intercept) 0.8022 1.0867 1.237 1.382 1.668
X          1.6878 1.8133 1.876 1.940 2.067
sigma2      0.8369 0.9964 1.095 1.210 1.468

# logistic模型
x<-rnorm(1000)
y<-rbinom(1000,1,exp(1-x)/(1+exp(1-x)))
posterior <-MCMClogit(y~x, b0=0, B0=.001)
plot(posterior)
summary(posterior)

> summary(posterior)

Iterations = 1001:11000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 10000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:
# 下面是模型的系数
            Mean     SD  Naive SE Time-series SE
(Intercept)  1.029 0.08118 0.0008118      0.002634
x          -1.077 0.09005 0.0009005      0.002665

2. Quantiles for each variable:

             2.5%    25%    50%    75%   97.5%
(Intercept)  0.8722  0.9719  1.030  1.085  1.1885
```

```
x          -1.2524 -1.1371 -1.077 -1.017 -0.8974

# probit模 型
y<-rbinom(1000,1,pnorm(1-x))
posterior <- MCMCprobit(y~x, b0=0,B0=.001)
plot(posterior)
summary(posterior)

# 泊 松 模 型
x<-rnorm(100)
y<-rpois(100,exp(1+x))
posterior <- MCMCpoisson(y ~x)
plot(posterior)
summary(posterior)
```

Chapter 103

Book: Bayesian Computation with R

这主要是一本书的笔记, 参考文献: Jim Albert *Bayesian Computation with R* [36]

使用包 LearnBayes

另外参考"机器学习", "算法" 一章86. 或见于参考文献[18]第4章. 有一个对贝叶斯推断和贝叶斯网络的简单明了的介绍.

103.1 使用MCMC估计显著性水平的例子

```
#␣一个教科书上面的t值计算函数，与R的函数t.test()结果相同
#␣sp为方差的合并
tstatistic=function(x,y)
{
m=length(x)
n=length(y)
sp=sqrt(((m-1)*sd(x)^2+(n-1)*sd(y)^2)/(m+n-2))
t=(mean(x)-mean(y))/(sp*sqrt(1/m+1/n))
return(t)
```

```
}

#␣数 据
x=c(1,4,3,6,5)
y=c(5,4,7,8,10)

>␣tstatistic(x,y)
[1]␣-1.937926

>␣t.test(x,y)$statistic␣␣#␣R的 结 果
␣␣␣␣␣␣␣␣t
-1.937926

␣␣␣␣
```

若 我 们 对 真 实 的 显 著 性 水 平 感 兴 趣 ， 而 数 据 不 服 从 标 准 正 态 分 布 ， 方 差 也 不 相 同 。 那 么 ， 真 实 的 显 著 性 水 平 依 赖 于

- 指 定 的 显 著 性 水 平α
- 数 据 的 分 布 形 状
- 方 差
- 样 本 数

给 定 显 著 性 水 平α， 分 布 形 状 ， 方 差 ， 样 本 数 ， 那 么 真 实 的 显 著 性 水 平 为

$$\alpha^T = P(|T| >= t_{n+m-2,\alpha/2})$$

计 算 真 实 显 著 性 水 平 的 步 骤 为

1. 从 第 一 个 样 本 和 第 二 个 样 本 分 别 采 样m，n个
2. 计 算 两 个 样 本 的t值T
3. 判 断 是 否$|T| > t_{n+m-2,\alpha/2}$， 即 是 否 拒 绝H_0

重复上面的步骤N次，真实的显著性水平为

$$\hat{\alpha^T} = \frac{number\ of\ rejections\ of\ H_0}{N}$$

```
alpha=.1; m=10; n=10
N=10000
n.reject=0
for (i in 1:N)
{
x=rnorm(m,mean=0,sd=1)
y=rnorm(n,mean=0,sd=1)
t=tstatistic(x,y)
if (abs(t)>qt(1-alpha/2,n+m-2))
    n.reject=n.reject+1
}
true.sig.level=n.reject/N

> true.sig.level
[1] 0.0969
```

固定$\alpha = 0.1, m = n = 10$，看不同分布的真实显著性情况

```
# 两个正态分布，方差不同
n.reject=0
for (i in 1:N)
{
x=rnorm(m,mean=0,sd=1)
y=rnorm(n,mean=0,sd=10)
t=tstatistic(x,y)
if (abs(t)>qt(1-alpha/2,n+m-2))
    n.reject=n.reject+1
}
true.sig.level=n.reject/N

> true.sig.level
[1] 0.1149
```

```
#␣两 个t分 布
n.reject=0
for␣(i␣in␣1:N)
{
x=rt(m,df=4)
y=rt(n,df=4)
t=tstatistic(x,y)
if␣(abs(t)>qt(1-alpha/2,n+m-2))
␣␣␣␣n.reject=n.reject+1
}
true.sig.level=n.reject/N

>␣true.sig.level
[1]␣0.0992

#␣两 个 指 数 分 布
n.reject=0
for␣(i␣in␣1:N)
{
x=rexp(m,rate=1)
y=rexp(n,rate=1)
t=tstatistic(x,y)
if␣(abs(t)>qt(1-alpha/2,n+m-2))
␣␣␣␣n.reject=n.reject+1
}
true.sig.level=n.reject/N

>␣true.sig.level
[1]␣0.0966

#␣一 个 正 态 分 布 , 一 个 指 数 分 布 , 均 值 都 为10
n.reject=0
for␣(i␣in␣1:N)
{
x=rnorm(m,mean=10,sd=2)
y=rexp(n,rate=1/10)
t=tstatistic(x,y)
if␣(abs(t)>qt(1-alpha/2,n+m-2))
␣␣␣␣n.reject=n.reject+1
}
```

```
true.sig.level=n.reject/N

>␣true.sig.level
[1]␣0.1555
```

␣␣␣␣

结果是，若两个分布的形状与方差都不同，那么其真实的显著性水平比指定的要显著高。

若分布相同，方差不同，则真实的显著性水平比指定的略高10%。

对于最后一种情况，正态分布与指数分布，我们可能想知道其t值的分布情况。下面绘制标准t值与模拟的t值

```
alpha=.1;␣m=10;␣n=10;␣N=10000;␣n.reject=0
tval<-rep(0,N)␣#␣保存t值
for␣(i␣in␣1:N)
{
x=rnorm(m,mean=10,sd=2)
y=rexp(n,rate=1/10)
t=tstatistic(x,y)
tval[i]=t
if␣(abs(t)>qt(1-alpha/2,n+m-2))
␣␣␣␣n.reject=n.reject+1
}
true.sig.level=n.reject/N

plot(density(tval),xlim=c(-5,8),ylim=c(0,.4),lwd=3)
x<-seq(-5,8,len=N)
lines(x,dt(x,df=18))
legend(4,.3,c("exact","t(18)"),lwd=c(3,1))

#␣得到真实的t值95%的上下阈值
>␣quantile(tval,c(0.025,0.5,0.975))
␣␣␣␣␣␣␣2.5%␣␣␣␣␣␣␣␣␣50%␣␣␣␣␣␣␣97.5%
-1.6000410␣␣␣0.1176278␣␣␣3.5945912
```

␣␣␣␣

下面是第一章练习题4 设y服从二项分布，采样n，成功率p，则p的90%置信区间为

$$C(y) = (\hat{p} - z_{0.95}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}, \hat{p} + z_{0.95}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}})$$

从下面的结果看出，真实p的置信区间依赖于n和p的选择

```
#␣计算理论成功率置信区间
binomial.conf.interval=function(y,n)
{
z=qnorm(.95)
p=y/n
se=sqrt(p*(1-p)/n)
return(c(p-z*se,p+z*se))
}

n=20;␣p=0.5;␣N=10000
conf<-c()
for(i␣in␣1:N){
␣␣␣␣y=rbinom(1,size=n,prob=p)
␣␣␣␣conf<-rbind(conf,binomial.conf.interval(y,n))
}

quantile(conf[,1],c(0.025,0.5,0.975))
quantile(conf[,2],c(0.025,0.5,0.975))

>␣conf[1:5,]
␣␣␣␣␣␣␣␣␣␣[,1]␣␣␣␣␣␣␣[,2]
[1,]␣0.2670216␣0.6329784
[2,]␣0.2198153␣0.5801847
[3,]␣0.1314527␣0.4685473
[4,]␣0.3160998␣0.6839002
[5,]␣0.2670216␣0.6329784

#␣p真实值的90%下界估计和下界的95%区间
>␣quantile(conf[,1],c(0.025,0.5,0.975))
␣␣␣␣␣2.5%␣␣␣␣␣␣␣␣50%␣␣␣␣␣97.5%
0.1314527␣0.3160998␣0.5314527
```

```
#␣p真实值的90%上界估计和上界的95%区间
>␣quantile(conf[,2],c(0.025,0.5,0.975))
␣␣␣␣␣2.5%␣␣␣␣␣␣␣␣50%␣␣␣␣␣97.5%
0.4685473␣0.6839002␣0.8685473

#-------n=5------------
>␣n=5;␣p=0.5;␣N=10000
>␣conf<-c()
>␣for(i␣in␣1:N){
+␣␣␣␣␣␣y=rbinom(1,size=n,prob=p)
+␣␣␣␣␣␣conf<-rbind(conf,binomial.conf.interval(y,n))
+␣}
>
>␣quantile(conf[,1],c(0.025,0.5,0.975))
␣␣␣␣␣␣␣2.5%␣␣␣␣␣␣␣␣␣␣50%␣␣␣␣␣␣␣97.5%
-0.09424036␣␣0.23963063␣␣1.00000000
>␣quantile(conf[,2],c(0.025,0.5,0.975))
␣␣␣␣␣2.5%␣␣␣␣␣␣␣␣50%␣␣␣␣␣97.5%
0.0000000␣0.9603694␣1.0942404

#-------n=100------------
>␣n=100;␣p=0.5;␣N=10000
>␣conf<-c()
>␣for(i␣in␣1:N){
+␣␣␣␣␣␣y=rbinom(1,size=n,prob=p)
+␣␣␣␣␣␣conf<-rbind(conf,binomial.conf.interval(y,n))
+␣}
>
>␣quantile(conf[,1],c(0.025,0.5,0.975))
␣␣␣␣␣2.5%␣␣␣␣␣␣␣␣50%␣␣␣␣␣97.5%
0.3194190␣0.4177573␣0.5194190
>␣quantile(conf[,2],c(0.025,0.5,0.975))
␣␣␣␣␣2.5%␣␣␣␣␣␣␣␣50%␣␣␣␣␣97.5%
0.4805810␣0.5822427␣0.6805810

#------p=0.2------------------
n=20;␣p=0.2;␣N=10000
conf<-c()
for(i␣in␣1:N){
␣␣␣␣y=rbinom(1,size=n,prob=p)
```

```
␣␣␣␣conf<-rbind(conf,binomial.conf.interval(y,n))
}

#␣p真实值的90%下界估计和下界的95%区间
>␣quantile(conf[,1],c(0.025,0.5,0.975))
␣␣␣␣␣␣␣2.5%␣␣␣␣␣␣␣␣␣␣50%␣␣␣␣␣␣␣97.5%
-0.03016025␣␣␣0.05287982␣␣␣0.21981531

#␣p真实值的90%上界估计和上界的95%区间
>␣quantile(conf[,2],c(0.025,0.5,0.975))
␣␣␣␣␣2.5%␣␣␣␣␣␣␣␣50%␣␣␣␣␣97.5%
0.1301603␣0.3471202␣0.5801847
␣␣␣␣
```

103.2 贝叶斯思想

先验概率是观察到数据之前对数据参数分布的估计。观察到数据后，更新其分布，叫做后验概率。

也可以对新样本的产生概率做出估计。

103.2.1 睡眠情况研究

大学生睡眠至少8小时的比例p是多少？

全美的大学生为总体。

研究者搜集到两个资料，1个说大部分大学生每天睡眠6小时。另外一个通过100名大学生的平时睡眠（工作日，weekdays）调查得到，70%睡眠5-6小时，28%睡眠7-8个小时，2%睡眠9个小时。

基于此，研究者认为睡眠8小时的比例小于0.5。经过考虑，p可能为0.3. 但是p有可能在0-0.5之间的任何值。

研究者搜集了27名大学生，其中11名报告昨天晚上睡眠至

少8小时。基于此数据，研究者想估计p的后验概率。并且，如果再调查20名大学生，她希望得到其p值的估计。

假设p的先验概率为$g(p)$。睡眠超过8小时为成功。随机采样成功s次，失败（睡眠少于8小时）f次，那么似然函数（即L(data—p）为

$$L(p) \propto p^s(1-p)^f, \quad 0 < p < 1$$

后验概率与先验概率乘以似然函数成比例

$$g(p|data) \propto g(p)L(p)$$

103.2.2 离散先验概率

p的取值可能为

.05, .15, .25, .35, .45, .55, .65, .75, .85, .95

她赋予这些p值一些权重（先验概率）

1, 5.2, 8, 7.2, 4.6, 2.1, 0.7, 0.1, 0, 0,

```
p=seq(0.05,0.95,by=0.1)
prior=c(1,␣5.2,␣8,␣7.2,␣4.6,␣2.1,␣0.7,␣0.1,␣0,␣0)
prior=prior/sum(prior)␣#␣权重归一化为概率值
plot(p,␣prior,␣type␣=␣"h",␣ylab="Prior␣Probability")
␣␣␣␣
```

注意似然函数为beta分布，参数为$s+1=12, f+1=17$

$$L(p) \propto p^{11}(1-p)^{16}, \quad 0 < p < 1$$

LearnBayes包的函数pdisc计算其后验概率

```
>␣pdisc
function␣(p,␣prior,␣data)
```

```
{
    s = data[1]
    f = data[2]
    # 对先验概率的0和1值校正
    p1 = p + 0.5 * (p == 0) - 0.5 * (p == 1)
    # 计算对数似然函数
    like = s * log(p1) + f * log(1 - p1)
    # p>0 p<1 的置0, p==0,p==1的对数似然减去999,防止其过大
    like = like * (p > 0) * (p < 1) -
        999 * ((p == 0) * (s > 0) + (p == 1) * (f > 0))
    # 似然值
    like = exp(like - max(like))
    # 似然值×先验概率得到未归一化的后验概率
    product = like * prior
    # 归一化
    post = product/sum(product)
    return(post)
}

# 下面计算观察数据后p的后验概率
> data = c(11, 16)
> post = pdisc(p, prior, data)
> round(cbind(p, prior, post),2)
          p prior post
 [1,] 0.05  0.03 0.00
 [2,] 0.15  0.18 0.00
 [3,] 0.25  0.28 0.13
 [4,] 0.35  0.25 0.48
 [5,] 0.45  0.16 0.33
 [6,] 0.55  0.07 0.06
 [7,] 0.65  0.02 0.00
 [8,] 0.75  0.00 0.00
 [9,] 0.85  0.00 0.00
[10,] 0.95  0.00 0.00

看到p落入0.25-0.45之间的概率为0.94
# 绘制先验概率和后验概率hist图
library(lattice)
PRIOR=data.frame("prior",p,prior)
POST=data.frame("posterior",p,post)
names(PRIOR)=c("Type","P","Probability")
```

```
names(POST)=c("Type","P","Probability")
data=rbind(PRIOR,POST)
xyplot(Probability~P|Type,data=data,layout=c(1,2),
type="h",lwd=3,col="black")
```

␣␣␣␣

103.2.3 先验概率为Beta分布

先验分布为beta分布，则

$$g(p) \propto p^{a-1}(1-p)^{b-1}, \quad 0 < p < 1$$

beta分布的均值为$m = a/(a+b)$，方差为$v = m(1-m)/(a+b+1)$。对于一般用户，可能估计a，b比较困难。但是，一般用户会给出两个百分位点的估计，一个为50%的p值小于0.3，即p的中位数为0.3，90%的p值会小于0.5. 即p的90百分位点为0.5.那么使用下面的函数估计参数a，b

```
quantile2=list(p=.9,x=.5)
quantile1=list(p=.5,x=.3)
beta.select(quantile1,quantile2)
```

```
>␣␣beta.select(quantile1,quantile2)
[1]␣3.26␣7.19
```

␣␣␣␣

beta.select()函数使用迭代的方法逼进给定两个百分位点的beta参数a，b。

结合先验概率与似然函数，后验概率为

$$g(p|data) \propto g(p)L(p) = p^{a+s-1}(1-p)^{b+f-1} = p^{3.26+11}(1-p)^{7.19+16}$$

则先验概率，似然函数，后验概率都为beta分布，下面绘制三个曲线

```
a␣=␣3.26
b␣=␣7.19
s␣=␣11
f␣=␣16
curve(dbeta(x,a+s,b+f),␣from=0,␣to=1,
xlab="p",ylab="Density",lty=1,lwd=4)
curve(dbeta(x,s+1,f+1),add=TRUE,lty=2,lwd=4)
curve(dbeta(x,a,b),add=TRUE,lty=3,lwd=4)
legend(.7,4,c("Prior","Likelihood","Posterior"),
␣␣␣␣␣␣␣␣lty=c(3,2,1),lwd=c(3,3,3))

#␣p值的95%置信区间
>␣qbeta(c(0.025,0.975),a+s,b+f)
[1]␣0.2343206␣0.5392949

#␣p值>0.5的概率
>␣1␣-␣pbeta(0.5,␣a␣+␣s,␣b␣+␣f)
[1]␣0.0690226
␣␣␣␣
```

另外一个获得区间的方法是随机模拟，结果与精确的值基本一致。

```
>␣␣ps␣=␣rbeta(1000,␣a␣+␣s,␣b␣+␣f)
>␣␣quantile(ps,␣c(0.05,␣0.95))
␣␣␣␣␣␣␣␣5%␣␣␣␣␣␣␣␣95%
0.2602180␣0.5155249
␣␣␣␣
```

103.2.4 Using a Histogram Prior(任意先验概率离散化)

类似第一个离散方法

- 选择p的区间可以覆盖后验概率密度
- 计算每个区间的先验概率与似然函数

- 先验与似然的乘积归一化
- 使用sample函数从离散分布中采样

模拟的结果近似了后验概率。

我们把这种方法叫做 histogram prior，有可能更好的反应先验概率（例如在先验概率的表达式很难给出时）

这里midpt称为睡眠超过8小时的比例p的离散化的中间值（midpoint of the intervals），向量prior为p的先验权重，归一化后为先验概率。使用histprior()函数来计算先验概率

```
midpt␣=␣seq(0.05,␣0.95,␣by␣=␣0.1)
prior␣=␣c(1,␣5.2,␣8,␣7.2,␣4.6,␣2.1,␣0.7,␣0.1,␣0,␣0)
prior␣=␣prior/sum(prior)
curve(histprior(x,midpt,prior),␣from=0,␣to=1,
␣␣␣␣ylab="Prior␣density",ylim=c(0,.3))

>␣x=seq(0,1,by=0.05)
>␣x
␣[1]␣0.00␣0.05␣0.10␣0.15␣0.20␣0.25␣0.30␣0.35␣0.40␣0.45␣0.50␣0.55␣0.60␣0.65␣0.70
[16]␣0.75␣0.80␣0.85␣0.90␣0.95␣1.00
>␣histprior(x,midpt,prior)␣#␣计算指定位置x的先验概率
␣[1]␣0.034602076␣0.034602076␣0.179930796␣0.179930796␣0.276816609␣0.276816609
␣[7]␣0.249134948␣0.249134948␣0.159169550␣0.159169550␣0.072664360␣0.072664360
[13]␣0.024221453␣0.024221453␣0.003460208␣0.003460208␣0.000000000␣0.000000000
[19]␣0.000000000␣0.000000000␣0.000000000

#␣计算后验概率，并归一化
>␣post=histprior(x,midpt,prior)␣*␣dbeta(x,s+1,f+1)
>␣post=post/sum(post)
>␣round(post,2)
␣[1]␣0.00␣0.00␣0.00␣0.00␣0.02␣0.07␣0.16␣0.27␣0.20␣0.19␣0.06␣0.03␣0.00␣0.00␣0.00
[16]␣0.00␣0.00␣0.00␣0.00␣0.00␣0.00

#␣以指定后验概率post对x随机采样
>␣␣ps␣=␣sample(x,␣replace␣=␣TRUE,␣prob␣=␣post)
>␣ps
␣[1]␣0.45␣0.45␣0.30␣0.45␣0.30␣0.45␣0.45␣0.30␣0.35␣0.35␣0.30␣0.30␣0.30␣0.35␣0.55
```

```
[16] 0.45 0.40 0.45 0.50 0.25 0.30

# 绘制直方图查看
> ps2 = sample(x, size=1000,replace = TRUE, prob = post)
> hist(ps2)
> quantile(ps2,c(0.025,0.5,0.975))
 2.5%   50% 97.5%
 0.25   0.40   0.55
```

103.2.5 预测

我们想估计下一个20个样本中睡眠好的人数y。那么

$$f(y) = \int f(y|p)g(p)dp$$

当p为先验概率，f称为先验预测密度，g若为后验概率，f称为后验预测密度。

```
> p=seq(0.05, 0.95, by=.1)
> prior = c(1, 5.2, 8, 7.2, 4.6, 2.1, 0.7, 0.1, 0, 0)
> prior=prior/sum(prior)

# 根据公式计算，例如成功5次的概率
> sum(dbinom(5,20,p)*prior)
[1] 0.1124487

# 下面批量计算，看到可能性最大的是y=5,6
> m=20
> ys=0:20 #感兴趣的成功次数
> pred=pdiscp(p, prior, m, ys)
> round(cbind(0:20,pred),3)
          pred
 [1,]   0 0.020
 [2,]   1 0.044
 [3,]   2 0.069
 [4,]   3 0.092
```

```
␣[5,]␣␣4␣0.106
␣[6,]␣␣5␣0.112
␣[7,]␣␣6␣0.110
␣[8,]␣␣7␣0.102
␣[9,]␣␣8␣0.089
[10,]␣␣9␣0.074
[11,]␣10␣0.059
[12,]␣11␣0.044
[13,]␣12␣0.031
[14,]␣13␣0.021
[15,]␣14␣0.013
[16,]␣15␣0.007
[17,]␣16␣0.004
[18,]␣17␣0.002
[19,]␣18␣0.001
[20,]␣19␣0.000
[21,]␣20␣0.000
␣␣␣␣
```

设先验概率为beta分布，那么

$$f(y) = \int f_B(y|m,p)g(p)dp = \binom{m}{y}\frac{B(a+y, b+m-y)}{B(a,b)}, \quad y = 0, \cdots, m$$

使用pbetap函数计算

```
>␣␣ab=c(3.26,␣7.19)
>␣␣m=20;␣ys=0:20
>␣␣pred=pbetap(ab,␣m,␣ys)
>␣round(pred,2)
␣[1]␣0.02␣0.05␣0.07␣0.09␣0.11␣0.11␣0.11␣0.10␣0.09␣0.07␣0.06␣0.04␣0.03␣0.02␣0.01
[16]␣0.01␣0.00␣0.00␣0.00␣0.00␣0.00
```

#␣直接按照公式计算

```
y=0;␣a=3.26,b=7.19;
>␣choose(m,y)*beta(a+y,b+m-y)/beta(a,b)
[1]␣0.01812205
```

```
> y=1
> choose(m,y)*beta(a+y,b+m-y)/beta(a,b)
[1] 0.04511485
> y=2
> choose(m,y)*beta(a+y,b+m-y)/beta(a,b)
[1] 0.07248106

# 小技巧，应用gamma函数与阶乘的关系
> choose(20,10)
[1] 184756
> exp(lgamma(21)-lgamma(11)-lgamma(11)) # 此式相当于choose(20,10)
[1] 184756
```

对于任意的先验分布求预测密度的一个方法是使用随机模拟。为了得到成功次数y，首先从先验分布g(p)采样p，然后从二项分布采样y。设先验分布服从beta(3.26,7.19)分布。

```
> p=rbeta(1000, 3.26, 7.19) # 随机模拟1000个p值
> y = rbinom(1000, 20, p) # 随机模拟1000个y值，成功概率为p
```

然后就可以统计y各种值的概率

```
> freq=table(y)
> ys=as.integer(names(freq)) # 横轴坐标
> predprob=freq/sum(freq) # 归一化
> plot(ys,predprob,type="h",xlab="y",
+ ylab="Predictive Probability")

> freq
y
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
 17  39  72  94 119 100 113 114  81  65  56  43  33  22  12  12   4   3   1

# 使用discint函数计算百分位点
> dist=cbind(ys,predprob)
> dist
```

```
␣␣␣ys␣predprob
0␣␣␣␣0␣␣␣␣␣0.017
1␣␣␣␣1␣␣␣␣␣0.039
2␣␣␣␣2␣␣␣␣␣0.072
3␣␣␣␣3␣␣␣␣␣0.094
4␣␣␣␣4␣␣␣␣␣0.119
5␣␣␣␣5␣␣␣␣␣0.100
6␣␣␣␣6␣␣␣␣␣0.113
7␣␣␣␣7␣␣␣␣␣0.114
8␣␣␣␣8␣␣␣␣␣0.081
9␣␣␣␣9␣␣␣␣␣0.065
10␣10␣␣␣␣␣0.056
11␣11␣␣␣␣␣0.043
12␣12␣␣␣␣␣0.033
13␣13␣␣␣␣␣0.022
14␣14␣␣␣␣␣0.012
15␣15␣␣␣␣␣0.012
16␣16␣␣␣␣␣0.004
17␣17␣␣␣␣␣0.003
18␣18␣␣␣␣␣0.001
>␣covprob=.9
>␣discint(dist,covprob)
$prob
␣␣␣12
0.929

$set
␣1␣␣2␣␣3␣␣4␣␣5␣␣6␣␣7␣␣8␣␣9␣10␣11␣12
␣1␣␣2␣␣3␣␣4␣␣5␣␣6␣␣7␣␣8␣␣9␣10␣11␣12


␣␣␣␣
```

103.3 单参数模型

103.3.1 已知均值未知方差的正态分布

Gelman et al (2003) 考虑一个已知均值的正态分布估计方差的问题.

对于均值为0的，方差未知的正态分布, 回忆正态分布函数为

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_a^b e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} dx$$

那么方差的似然函数为

$$L(\sigma^2) = (\sigma^2)^{-n/2} exp(-\sum_{i=1}^{n} d_i^2/(2\sigma^2)), \sigma^2 > 0$$

考虑无信息先验概率密度为$p(\sigma^2) = 1/\sigma^2$. 后验密度为

$$g(\sigma^2|data) \propto (\sigma^2)^{-1/2-1} exp(-v/(2\sigma^2))$$

其中$v = \sum_{i=1}^{n} d_i^2$. 如果定义精确参数$P = 1/\sigma^2$, 那么可以证明，P的分布为U/v, U为自由度n的卡方分布. 假设我们对方差的点估计和95%置信区间感兴趣.

下面使用LearnBayes包的数据footballscores, 估计$d = favorite - underdog - spread$的方差. 已经知道方差的倒数P服从的分布为卡方分布/v. 那么根据这个结论直接计算即可.

```
library(LearnBayes)
data(footballscores)
attach(footballscores)
d = favorite - underdog - spread
n = length(d)
v = sum(d^2)
```

```
# 计算方差的倒数 P=1/sigma^2 ~ U/v, 其中U服从卡方分布, v是观测的方差和.
P = rchisq(1000, n)/v
s = sqrt(1/P) # 计算标准差 sigma
hist(s,main="") # 查看标准差的分布情况

# 估计标准差的95%置信区间, 和中位数点估计. 中位数通常是一个稳健的估计, 当然这里与均值相差很小.
quantile(s, probs = c(0.025, 0.5, 0.975))

> quantile(s, probs = c(0.025, 0.5, 0.975))
   2.5%     50%   97.5%
13.17012 13.85135 14.56599
```

对于其它熟悉的密度函数的情况, 象指数族分布, 并且已经选择了一个共轭先验分布, 可以使用R的随机数函数, 譬如, rnorm, rbeta, rgamma, 来直接计算后验概率.

103.3.2 估计心脏移植手术的成活率

考虑某医院的心脏移植手术的成活率。统计到心脏移植手术共n例，30天内死亡y例。根据医院的各种情况，期望的死亡数为e[1]，标准的模型假设死亡数y服从泊松分布，均值为$e\lambda$，目标是估计每例手术的危险率λ。

对危险率标准的估计是最大似然估计$\lambda = y/e$。当y接近0的时候，这个估计就会很差。此时我们希望使用贝叶斯方法，应用一个先验的危险率分布。一个合适的选择是gamma分布族。

$$p(\lambda) \propto \lambda^{\alpha-1} exp(-\beta\lambda), \quad \lambda > 0$$

合适的先验概率信息来源是其他条件相似的医院的心脏移植手术的危险率，我们相信他们的危险率差不

[1]此处e似乎应该为手术例数，下文也提示e为手术例数，泊松分布的定义中也应该是这样

多。假设我们从10家医院观察到死亡为z_j，观察手术例数为$o_j, j = 1, \cdots, 10$。那么z_j的均值为$o_j\lambda$。我们定义λ为标准无信息先验概率$p(\lambda) \propto \lambda^{-1}$，那么更新后的先验分布为

$$p(\lambda) \propto \lambda^{\sum_{j=1}^{1} 0z_j - 1} exp(-(\sum_{j=1}^{1} 0o_j)\lambda)$$

使用此信息，λ的先验概率为$gamma(\alpha, \beta)$分布，$\alpha = \sum_{j=1}^{1} 0z_j$，$\beta = \sum_{j=1}^{1} 0o_j$，此例中，

$$\sum_{j=1}^{1} 0z_j = 16, \quad \sum_{j=1}^{1} 0o_j = 15174$$

那么λ的先验概率为$gamma(16, 15174)$。后验概率分布为$gamma(\alpha + y, \beta + e)$。先验预测y的密度为

$$f(y) = \frac{f(y|\lambda)g(\lambda)}{g(\lambda|y)}$$

$f(y|\lambda)$为泊松分布$Poisson(e\lambda)$,即条件概率，y的似然函数。

考虑医院A，66个手术1个死亡，即$y = 1, e = 66$

```
alpha=16;beta=15174
yobs=1;␣ex=66
y=0:10
lam=alpha/beta
py=dpois(y,␣lam*ex)*dgamma(lam,␣shape␣=␣alpha,
␣␣␣␣␣rate␣=␣beta)/dgamma(lam,␣shape=␣alpha␣+␣y,
␣␣␣␣␣rate␣=␣beta␣+␣ex)
#␣dpois(y,␣lam*ex)：␣为y的似然函数，条件分布
#␣dgamma(lam,␣shape␣=␣alpha,rate␣=␣beta)：␣lambda的先验概率
#␣dgamma(lam,␣shape=␣alpha␣+␣y,rate␣=␣beta␣+␣ex)：lambda的后验概率

>␣cbind(y,␣round(py,␣3))␣#␣未来66例手术死亡次数的估计为
```

```
␣␣␣␣␣␣␣y
␣[1,]␣␣␣0␣0.933
␣[2,]␣␣␣1␣0.065
␣[3,]␣␣␣2␣0.002
␣[4,]␣␣␣3␣0.000
␣[5,]␣␣␣4␣0.000
␣[6,]␣␣␣5␣0.000
␣[7,]␣␣␣6␣0.000
␣[8,]␣␣␣7␣0.000
␣[9,]␣␣␣8␣0.000
[10,]␣␣␣9␣0.000
[11,]␣␣10␣0.000
␣␣␣␣
```

既然知道后验概率，我们可以使用随机模拟λ的分布1000次来描述其密度

```
lambdaA␣=␣rgamma(1000,␣shape␣=␣alpha␣+␣yobs,␣rate␣=␣beta␣+␣ex)
␣␣␣␣
```

考虑若1767例手术死亡4例，即$y = 4, e = 1767$，那么

```
␣ex␣=␣1767;␣yobs=4
␣y␣=␣0:10
␣py␣=␣dpois(y,␣lam␣*␣ex)␣*␣dgamma(lam,␣shape␣=␣alpha,
␣␣␣␣rate␣=␣beta)/dgamma(lam,␣shape␣=␣alpha␣+␣y,
␣␣␣␣rate␣=␣beta␣+␣ex)

>␣␣cbind(y,␣round(py,␣3))␣#␣未来1767例手术死亡次数的估计为
␣␣␣␣␣␣␣␣y
␣[1,]␣␣␣0␣0.172
␣[2,]␣␣␣1␣0.286
␣[3,]␣␣␣2␣0.254
␣[4,]␣␣␣3␣0.159
␣[5,]␣␣␣4␣0.079
␣[6,]␣␣␣5␣0.033
␣[7,]␣␣␣6␣0.012
␣[8,]␣␣␣7␣0.004
```

```
␣[9,]␣␣8␣0.001
[10,]␣␣9␣0.000
[11,]␣10␣0.000

>␣lambdaB␣=␣rgamma(1000,␣shape␣=␣alpha␣+␣yobs,␣rate␣=␣beta␣+␣ex)
␣␣␣␣
```

绘图查看先验概率的影响

```
par(mfrow␣=␣c(2,␣1))
plot(density(lambdaA),␣main="HOSPITAL␣A",
␣␣␣␣xlab="lambdaA",␣lwd=3)
curve(dgamma(x,␣shape␣=␣alpha,␣rate␣=␣beta),␣add=TRUE)
legend("topright",legend=c("prior","posterior"),lwd=c(1,3))
plot(density(lambdaB),␣main="HOSPITAL␣B",
␣␣␣␣xlab="lambdaB",␣lwd=3)
curve(dgamma(x,␣shape␣=␣alpha,␣rate␣=␣beta),␣add=TRUE)
legend("topright",legend=c("prior","posterior"),lwd=c(1,3))
␣␣␣␣
```

103.3.3 贝叶斯方法的鲁棒性

103.3.3.1 先验概率：正态分布

假设要考察Joe的智商IQ θ。你相信Joe的智商应该在平均水平。先验估计的中位数为100，而且90%置信区间落入80-120之间。使用函数normal.select，可以估计符合上面两个先验信息的正态分布的均值和标准差。

```
quantile1=list(p=.5,x=100);␣quantile2=list(p=.95,x=120)
normal.select(quantile1,␣quantile2)

$mu
[1]␣100

$sigma␣#␣此处我们称为␣tau，观测值的方差为␣sigma，se
[1]␣12.15914
```

␣␣␣␣

Joe 进 行 了4次IQ测 试 ， 结 果 为$y1, y2, y3, y4$, 其 分 布 服 从$N(\theta = 100, \sigma = 15)$. 那 么

$$P(\theta|y) proptop P(\theta)P(y|\theta)$$

此 时 观 测 到 一 个 试 验 结 果$y1 = 110$。 如 果 认 为$\theta \in (60, 180)$， 我 们 可 以 这 样 来 计 算θ的 后 验 概 率 和 后 验 期 望 ,

```
ybar=110␣␣␣␣#␣观 察 到 一 个 值
theta=seq(60,␣180,␣length␣=␣500)␣␣␣#␣theta的 可 能 的 取 值 范 围
sigma=15
tau=12.16
sd=sigma/sqrt(4)␣␣␣#␣根 据 观 测 值 重 新 估 计 方 差
prior␣=␣dnorm(theta,mean=100,sd=tau)␣␣␣#␣先 验 概 率 服 从N(100,12.16)
␣␣␣␣#␣似 然 概 率 服 从␣N(mean=theta,sd=重 新 估 计 的 方 差)
like␣=␣dnorm(ybar,mean=theta,sd=sd)
post=prior␣*␣like␣␣␣␣␣#␣后 验 概 率=先 验×似 然
post=post/sum(post)␣␣␣#␣归 一 化(theta的)后 验 概 率
m␣=␣sum(theta␣*␣post)␣␣␣␣␣#␣theta␣的 期 望

>␣m
[1]␣107.2442
```

注： 由 于 正 态 分 布 的 对 称 性 ,

```
like␣=␣dnorm(ybar,mean=theta,sd=sd)
```

相 当 于

```
like␣=␣dnorm(theta,mean=ybar,sd=sd)
```

␣␣␣␣

若 观 察 值 的 方 差 未 知 ， 那 么 可 以 由 先 验 方 差$tau = 12.16$代 替 , 最 后 结 果 稍 微 不 同 。

```
like␣=␣dnorm(ybar,mean=theta,sd=tau)

>␣m
[1]␣105
```

␣␣␣␣

若使用前3个值推测，假设每次测验是独立的。记$Y=(y1,y2,y3)$，那么

$$P(\theta|Y)proptopP(\theta)P(Y|\theta)=P(\theta)P(y1|\theta)P(y2|\theta)P(y3|\theta)$$

下面来具体计算

```
y1=110;␣y2=125;␣y3=140␣␣␣␣#␣观察到3个值
theta=seq(60,␣180,␣length␣=␣500)␣␣␣␣#␣theta的可能的取值范围
sigma=15
tau=12.16
sd=sigma/sqrt(4)␣␣␣␣␣␣␣#␣根据观测值重新估计方差
prior␣=␣dnorm(theta,mean=100,sd=tau)␣␣␣␣␣␣#␣先验概率服从N(100,12.16)
␣␣␣␣␣␣#␣似然概率服从␣N(mean=theta,sd=重新估计的方差)
like␣=␣dnorm(y1,mean=theta,sd=sd)␣*
␣␣␣␣␣␣␣␣dnorm(y2,mean=theta,sd=sd)␣*
␣␣␣␣␣␣␣␣dnorm(y3,mean=theta,sd=sd)

post=prior␣*␣like␣#␣后验概率=先验×似然
post=post/sum(post)␣#␣归一化(theta的)后验概率
m␣=␣sum(theta␣*␣post)␣#␣theta␣的期望

>␣␣m
[1]␣122.1866
␣␣␣␣
```

由于我们已经知道先验分布和似然分布都是正态分布的后验分布也是正态分布。且后验分布的方差

$$\tau_1=1/\sqrt{4/\sigma^2+1/tau^2}$$

后验估计

$$\mu_1=[4/\sigma^2+\mu(1/tau^2)]tau_1^2$$

我们可以这样计算

```
mu␣=␣100
tau␣=␣12.16
sigma␣=␣15
```

```
n␣=␣4
se␣=␣sigma/sqrt(4)
ybar␣=␣c(110,␣125,␣140)
tau1␣=␣1/sqrt(1/se^2␣+␣1/tau^2)
mu1␣=␣(ybar/se^2␣+␣mu/tau^2)␣*␣tau1^2
summ1=cbind(ybar,␣mu1,␣tau1)

>␣summ1
␣␣␣␣␣ybar␣␣␣␣␣␣␣mu1␣␣␣␣␣␣tau1
[1,]␣␣110␣107.2442␣6.383469
[2,]␣␣125␣118.1105␣6.383469
[3,]␣␣140␣128.9768␣6.383469
␣␣␣␣
```

103.3.3.2 先验概率：t分布

我们使用另外一个对称分布t分布来代替正态分布的先验概率分布。

均值为$\mu = 100$, 为了满足95百分位点是120的条件，我们计算其尺度（方差）τ

```
>␣␣tscale␣=␣20/qt(0.95,␣2)
>␣␣tscale
[1]␣6.849349

#␣绘制t分布与正态分布的曲线
mu=100
par(mfrow=c(1,1))
curve(1/tscale*dt((x-mu)/tscale,2),
from=60,␣to=140,␣xlab="theta",␣ylab="Prior␣Density")
curve(dnorm(x,mean=mu,sd=tau),␣add=TRUE,␣lwd=3)
legend("topright",legend=c("t␣density","normal␣density"),
lwd=c(1,3))
␣␣␣␣
```

后验分布为

$$g(\theta|data)proptog_T(\theta|v,\mu,\tau)L(\bar{(}y)|\theta,\sigma/sqrtn)$$

似然函数$L(\bar{(y)}|\theta,\sigma/sqrtn)$与前面的似然函数完全一样，服从$N(\theta,\sigma/sqrtn)$

下面具体计算后验概率与后验估计

```
norm.t.compute=function(ybar)␣{
␣␣␣␣theta␣=␣seq(60,␣180,␣length␣=␣500)
␣␣␣␣like␣=␣dnorm(theta,mean=ybar,sd=sigma/sqrt(n))
␣␣␣␣prior␣=␣dt((theta␣-␣mu)/tscale,␣2)
␣␣␣␣post␣=␣prior␣*␣like
␣␣␣␣post␣=␣post/sum(post)
␣␣␣␣m␣=␣sum(theta␣*␣post)
␣␣␣␣s␣=␣sqrt(sum(theta^2␣*␣post)␣-␣m^2)␣#␣估计后验方差
␣␣␣␣c(ybar,␣m,␣s)
}
summ2=t(sapply(c(110,␣125,␣140),norm.t.compute))
dimnames(summ2)[[2]]=c("ybar","mu1␣t","tau1␣t")
summ2

>␣summ2
␣␣␣␣␣ybar␣␣␣␣␣mu1␣t␣␣␣tau1␣t
[1,]␣␣110␣105.2921␣5.841676
[2,]␣␣125␣118.0841␣7.885174
[3,]␣␣140␣135.4134␣7.973498
␣␣␣␣
```

从结果看出，当值比较大的时候，即远离先验均值，先验概率对结果的影响还是比较大。

```
#␣两个结果比较
>␣cbind(summ1,summ2)
␣␣␣␣␣ybar␣␣␣␣␣␣␣mu1␣␣␣␣␣␣tau1␣ybar␣␣␣␣␣mu1␣t␣␣␣tau1␣t
[1,]␣␣110␣107.2442␣6.383469␣␣110␣105.2921␣5.841676
[2,]␣␣125␣118.1105␣6.383469␣␣125␣118.0841␣7.885174
[3,]␣␣140␣128.9768␣6.383469␣␣140␣135.4134␣7.973498
␣␣␣␣
```

103.3.4 混合先验概率

103.3.4.1 理论计算

要推广先验概率的范围，一个直接的方法是使用离散分布。

假设一个硬币是有偏的，一面为0.7, 另一面为0.3, 但是不知道是正面还是反面的概率大，现在要估计哪一面的概率是0.7. 没有数据的时候，正面反面的可能性各一半。这个猜测可以使用下面的模型表达

$$g(p) = \gamma g_1(p) + (1-\gamma)g_2(p)$$

$g_1 = beta(6,14), g_2 = beta(14,6), \gamma = 0.5$(正反的概率各一半)。

这种情况下，先验概率和后验概率都是混合beta分布，即它们是共轭分布（分布形式一致），后验分布为

$$g(p|data) = \gamma(data)g_1(p|data) + (1-\gamma(data))g_2(p|data)$$

$g_1 = beta(6+s, 14+f), g_2 = beta(14+s, 6+f)$

$$\gamma(data) = \frac{\gamma f_1(s,f)}{\gamma f_1(s,f) + (1-\gamma)f_2(s,f)}$$

$f_i(s,f)$为，当p的先验概率为g_i, n次试验有s次正面的概率。我们使用函数 binomial.beta.mix() 来计算后验γ的值。

```
probs=c(.5,.5)
beta.par1=c(6,␣14)
beta.par2=c(14,␣6)
betapar=rbind(beta.par1,␣beta.par2)
data=c(7,3)
post=binomial.beta.mix(probs,betapar,data)
```

```
>␣post
$probs
␣beta.par1␣␣beta.par2
0.09269663␣0.90730337
```

```
$betapar
␣␣␣␣␣␣␣␣␣␣[,1]␣[,2]
beta.par1␣␣␣13␣␣␣17
beta.par2␣␣␣21␣␣␣␣9

#␣绘图查看p的先验，后验概率
curve(post$probs[1]*dbeta(x,13,17)+post$probs[2]*dbeta(x,21,9),
from=0,␣to=1,␣lwd=3,␣xlab="P",␣ylab="DENSITY")
curve(.5*dbeta(x,6,12)+.5*dbeta(x,12,6),0,1,add=TRUE)
legend("topleft",legend=c("Prior","Posterior"),lwd=c(1,3))
␣␣␣␣
```

后验概率的形式为

$$g(p|data) = 0.093 * beta(13, 17) + 0.907 * beta(21, 9)$$

103.3.4.2 模拟方法

下面我们使用模拟的方法来计算p的后验概率

```
p=seq(0,1,len=500)␣#␣p␣的可能取值
ga=0.5␣#␣gamma
prior=ga*dbeta(p,6,14)+(1-ga)*dbeta(p,14,6)
#␣prior=prior/sum(prior)␣#␣没有变换到对数计算的时候不需要
归一化
like=dbinom(x=7,size=10,prob=p)
post=prior*like
post=post/sum(post)

plot(p,post)␣#␣p␣的极大值点在0.714左右。
lines(p,prior/sum(prior))␣#␣先验概率

#␣p的后验平均值
m=sum(post*p)
>␣m
[1]␣0.6752809
␣␣␣␣
```

103.3.5 TODO: 硬币均匀性检验

抛硬币试验服从二项分布。我们要检验硬币是否均匀，即出现正面的概率

$$H_0: \ p = 0.5 \quad vs \quad H_1: \ p \neq 0$$

若n次试验观测到y次正面，那么其p值为

$$2 * min(P(Y <= y), P(Y >= y))$$

若p值很小，则拒绝零假设，否则接受。

下面是20次试验出现5次正面的概率和小于等于5次的概率及其p值

```
>␣␣pbinom(5,␣20,␣0.5)
[1]␣0.02069473
>␣␣sum(pbinom(0:5,␣20,␣0.5))
[1]␣0.02811432
>␣␣sum(pbinom(0:5,␣20,␣0.5))*2␣#␣p值
[1]␣0.05622864
␣␣␣␣
```

此处我们来考虑贝叶斯估计。若试验之前你认为硬币很有可能是均匀的，那么p值限定在0.5, 如果硬币非均匀，那么你可能会赋给一个介于0,1之间的先验概率，称为$g_1(p)$，表示你对硬币非均匀性的先验估计。假设$g_1 = beta(a, a)$，这个beta分布关于0.5对称。那么综合上面的描述，先验概率为

$$g(p) = 0.5I(p = 0.5) + 0.5I(p \neq 0.5)g_1(p)$$

$I(A)$是一个指示性函数，若A为真，$I = 1$,否则$I = 0$.

p的后验概率为

$$g(p|y) = \lambda(y)I(p = 0.5) + (1 - \lambda(y))g_1(p|y)$$

$g_1(p|y) = beta(a + y, a + n - y)$, $\lambda(y)$为模型的后验概率

$$\lambda(y) = \frac{0.5p(y|0.5)}{0.5p(y|0.5) + 0.5m_1(y)}$$

$m_1(y)$为使用beta分布对y的先验估计，$\lambda(y), p(y|0.5)$是$p = 0.5$时的二项分布的y的概率密度。

在R中，。。。。

103.4 TODO: 多参数模型

103.5 贝叶斯计算

在前面有两种方法来计算后验概率。若样本密度的函数形式比较熟悉，那么直接使用R的随机数函数（rnorm，rbeta，rgamma等），然后可以基于此做出统计。另外一种方法是，叫做“brute-force“方法，后验概率的形式不是熟悉的形式，那么将其离散化然后逼进其连续值。

103.5.1

假设观测到样本$y_1, \cdots, y_n$，那么待估计参数的后验概率的对数为

$$\log g(\theta|y) = \log g(\theta) + \sum_{i=1}^{n} \log f(y_i|\theta)$$

例如我们从正态分布$N(\mu, \sigma)$中采样，参数为$\theta = (\mu, \log \sigma)$，$\mu$的先验概率为$N(10, 20)$，$\log \sigma$的先验概率为平坦函数。对数后验概率为

$$\log g(\theta|y) = \log \phi(\mu; 10, 20) + \sum_{i=1}^{n} \log \phi(y_i; \mu, \sigma)$$

为了简化代码我们写出一个简单的函数计算y的对数似然概率密度

```
logf␣<-␣function(y,␣mu,␣sigma)
␣␣␣␣{dnorm(y,mean=mu,sd=sigma,log=TRUE)}
␣␣␣␣
```

若数据为$data = (y_1, \cdots, y_n)$，我们就可以计算对数似然的和$\sum_{i=1}^{n} \log \phi(y_i; \mu, \sigma)$

```
sum(logf(data,mu,sigma))
␣␣␣␣
```

对数后验概率的计算为

```
mylogposterior<-function(theta,data)
{
␣␣␣␣n=length(data)
␣␣␣␣mu=theta[1];␣sigma=exp(theta[2])
␣␣␣␣logf␣<-␣function(y,␣mu,␣sigma)
␣␣␣␣␣␣␣␣{dnorm(y,mean=mu,sd=sigma,log=TRUE)}
␣␣␣␣val=dnorm(mu,␣mean=10,
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣sd=20,log=TRUE)+sum(logf(data,mu,sigma))
␣␣␣␣return(val)
}
␣␣␣␣
```

103.5.2 平坦分布的 Beta-Binomial 模型

前面一个数字是死亡数，后面一个是胃癌患病数

```
(0,␣1083)
(0,␣527)
(1,␣680)
(3,␣588)
(0,␣855)
(2,␣3461)
(0,␣657)
(1,␣1208)
```

```
(1,␣1025)
(2,␣1668)
(1,␣583)
(3,␣582)
(0,␣917)
(1,␣857)
(1,␣917)
(54,␣53637)
(0,␣874)
(0,␣395)
(1,␣581)
(0,␣383)
␣␣␣␣
```

二项分布好像不太合适。就使用一个beta-binomial 模型来拟合

$$f(y_i|\eta,K)=\binom{n_j}{y_i}\frac{B(K\eta+y_i,K(1-\eta)+n_j-y_j)}{B(K\eta,K(1-\eta))}$$

假设我们指定先验分布为

$$g(\eta,K)\propto\frac{1}{\eta(1-\eta)}\frac{1}{(1+K)^2}$$

后验概率为

$$g(\eta,K|data)\propto\frac{1}{\eta(1-\eta)}\frac{1}{(1+K)^2}\prod_{j=1}^{20}\frac{B(K\eta+y_i,K(1-\eta)+n_j-y_j)}{B(K\eta,K(1-\eta))}$$

$0<\eta<1,K>0$

下面是计算对数后验概率的函数

```
betabinexch0<-function␣(theta,␣data)
{
␣␣␣␣eta␣=␣theta[1]
␣␣␣␣K␣=␣theta[2]
␣␣␣␣y␣=␣data[,␣1]
```

```
␣␣␣␣n␣=␣data[,␣2]
␣␣␣␣N␣=␣length(y)
␣␣␣␣logf␣<-␣function(y,␣n,␣K,␣eta)
␣␣␣␣␣␣␣␣{lbeta(K␣*␣eta␣+␣y,
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣K␣*␣(1␣-␣eta)␣+␣n␣-␣y)␣-
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣lbeta(K␣*␣eta,␣K␣*␣(1␣-␣eta))}
␣␣␣␣val␣=␣sum(logf(y,␣n,␣K,␣eta))
␣␣␣␣val␣=␣val␣-␣2␣*␣log(1␣+␣K)␣-␣log(eta)␣-␣log(1␣-␣eta)
␣␣␣␣return(val)
}
␣␣␣␣
```

103.5.3 MC方法计算积分

设θ的后验概率为$g(\theta|y)$，我们对θ的某函数$h(\theta)$的平均值感兴趣（一般$h(\theta)=\theta$），那么$h(\theta)$的后验平均值为

$$E(h(\theta)|y)=\int h(\theta)g(\theta|y)d\theta$$

假设我们可以从后验概率密度模拟独立的样本$\theta^1,\cdots,\theta^m$，那么 Monte Carlo 估计后验平均值为

$$\bar{(h)}=\frac{\sum_{j=1}^m h(\theta^j)}{m}$$

标准差为

$$se_{\bar{h}}=\sqrt{\frac{\sum_{j=1}^m(h(\theta^j)-\bar{h})^2}{(m-1)m}}$$

当后验概率的精确样本点可以得到，那么MC方法是很有效的方法。

例如睡眠的例子中p的后验概率形式为$beta(14.26,23.19)$. 假设我们对p^2的均值感兴趣（两个学生有充足睡眠的概率）。我们采样1000次。p_j表示采样点，那么p_j^2均值就是MC模拟的均值。

```
p=rbeta(1000,␣14.26,␣23.19)␣#␣MC采样1000次
est=mean(p^2)␣#␣计算p^2的均值
se=sd(p^2)/sqrt(1000)␣#␣p^2均值的标准差
>␣c(est,se)
[1]␣0.150293014␣0.002080788
␣␣␣␣
```

MC模拟的估计为$E(p^2|data) = 0.149, se = 0.002$

103.5.4 Rejection Sampling

在前面几章的例子中，我们直接对后验概率分布采样，因为后验概率的函数形式是我们熟悉的。但是在很多时候，函数形式我们并不熟悉，例如混合分布 beta-binomial 分布。所以我们需要另外的方法来产生模拟样本。

Rejection Sampling（舍选法）是一个通用的采样方法。假设我们需要产生后验概率密度函数$g(\theta|y)$的独立样本。舍选法的第一步是寻找一个概率密度函数$p(\theta)$，满足

- 容易计算
- 函数$p(\theta)$与$g(\theta|y)$的分布形状差不多
- 对于所有θ和常数c，$g(\theta|y) <= cp(\theta)$

我们使用下面的方法来产生$g(\theta|y)$的独立随机样本

1. 从分布$p(\theta)$中产生独立的随机数θ，从均匀分布$U(0,1)$中产生随机数u
2. 若$u <= \frac{g(\theta|y)}{cp(\theta)}$，接受$\theta$作为一个合格的采样，服从$g(\theta|y)$
3. 重复步骤1,2直到产生足够的服从$g(\theta|y)$分布的随机数为止

舍选法是最重要的采样方法之一。

设计舍选法抽样的时候，主要任务是寻找合适的概率分布$p(\theta)$和常数c。有效的舍选法的接受概率比较高。

103.6 MCMC方法

103.6.1 Metropolis-Hastings 算法

MCMC模拟后验分布，实际上是离散马尔柯夫链的连续化。MCMC采样产生一个不可复原/不可简化（irreducible），非周期（aperiodic）的马尔柯夫链，其平稳分布等于后验分布。

通常的方法是使用Metropolis-Hastings方法构建马尔柯夫链。

设后验概率分布为$g(\theta|y)$，我们简化称为$g(\theta)$。Metropolis-Hastings算法从一个初始的值θ^0开始，并确定由θ^{t-1}产生θ^t值的规则。此规则包括一个候选的密度来模拟θ^*，一个接受概率P指明候选的θ^*是否作为下一个θ^t值。算法可以描述为

- 由期望的概率密度$p(\theta^*|\theta^{t-1})$得到一个θ^*
- 计算比例

$$R = \frac{g(\theta^*)p(\theta^{t-1}|\theta^*)}{g(\theta^{t-1})p(\theta^*|\theta^{t-1})}$$

- 计算接受概率$P = min(R, 1)$
- 以概率P接受$\theta^t = \theta^*$，否则拒绝

不同的Metropolis-Hastings方法构建由期望的概率密度$p(\theta^*|\theta^{t-1})$的方法不同。如果期望概率密度不依赖于当前值

$$p(\theta^*|\theta^{t-1}) = p(\theta^*)$$

那么算法的结果叫做独立链，其他的定义有

$$p(\theta^*|\theta^{t-1}) = h(\theta^* - \theta^{t-1})$$

h关于初始值对称。在随机行走链中，R有简单的形式

$$R = \frac{g(\theta^*)}{g(\theta^{t-1})}$$

LearnBayes 包的函数 rwmetrop and indepmetrop 分别使用随机行走和独立的方法采样。

rwmetrop 使用随机行走的方法采样。其期望密度函数为

$$\theta^* = \theta^{t-1} + scale * Z$$

Z为多元正态分布，均值为0,协方差矩阵为V，scale为尺度参数。

indepmetrop 使用独立的方法采样。θ^*服从多元正态分布，均值为μ，协方差矩阵为V。

要使用 Metropolis-Hastings 方法，首先需要确定期望密度函数，然后得到采样值。

103.6.2 Gibbs Sampling

设我们的参数为$\theta = (\theta_1, \cdots, \theta_p)$.联合后验分布为$[\theta|data]$，我们定义下面的条件分布

```
␣␣␣␣␣\left\[␣\begin{array}{l}
\theta_1|\theta_2,\cdots,\theta_p,data␣\\
\theta_2|\theta_1,\theta_3,\cdots,\theta_p,data␣\\
\cdots␣\\
\theta_p|\theta_1,\cdots,\theta_{p-1},data
␣␣␣␣\end{array}␣\right␣\]
␣␣␣␣␣
```

Gibbs 采样的思想是可以从条件分布来采样最后得到联合分布的样本。最后可以收敛到目标联合分布。

对于条件分布难以直接采样的情况，可以使用 Metropolis 方法例如随机行走来对每个分布采样。假设θ_i^t为当前的θ_i的采样值，令$g(\theta_i)$表示θ_i的条件概率分布函数，以便去除其他θ值的依赖。那么一个备选的下一个θ_i^{t+1} 值为

$$\theta_i^* = \theta_i^t + c_i Z$$

Z为标准正态分布，c_i为固定的尺度参数。我们以概率

$$P = min(1, g(\theta_i^*)/g(\theta_i^t))$$

接受$\theta_i^{t+1} = \theta_i^*$，否则不变，即$\theta_i^{t+1} = \theta_i^t$。

103.7 模型比较

103.7.1 比较假设

某人从分布$f(y|\theta)$中观察到数据Y，想推断

$$H_0 : \theta \in \Theta_0, \qquad H_1 : \theta \in \Theta_1$$

若先验概率为$g(\theta)$，那么先验优势比（prior odds ratio）为

$$\frac{\pi_0}{\pi_1} = \frac{P(\theta \in \Theta_0)}{P(\theta \in \Theta_0)} = \frac{\int_{\Theta_0} g(\theta)d\theta}{\int_{\Theta_1} g(\theta)d\theta}$$

观察到数据$Y = y$后，后验概率为

$$g(\theta|y) \propto L(\theta)g(\theta)$$

后验优势比为

$$\frac{p_0}{p_1} = \frac{P(\theta \in \Theta_0|y)}{P(\theta \in \Theta_0|y)} = \frac{\int_{\Theta_0} g(\theta|y)d\theta}{\int_{\Theta_1} g(\theta|y)d\theta}$$

贝叶斯因子(BF)为后验优势比与先验优势比的比值

$$BF = \frac{p_0/p_1}{\pi_0/\pi_1}$$

BF的含义是数据Y提供的支持H_0的证据的度量。

注意到$p_1 = 1 - p_0, \pi_1 = 1 - \pi_0$，我们有

$$p_0 = \frac{\pi_0 BF}{\pi_0 BF + 1 - \pi_0}$$

103.7.2 单边检验

某人体重称了10次，结果为 182, 172, 173, 176, 176, 180, 173, 174, 179, and 175. 为简化起见，他知道自己的体重标准差为$\sigma = 3$。令μ为他的真实体重，假设我们对他的体重是否超过175感兴趣

$$H_0 : \mu <= 175 \qquad H_1 : \mu > 175$$

假设他对自己体重的先验知识很少，那么就假设

$$\mu \sim N(170, 5)$$

先验优势比为

$$\frac{\pi_0}{\pi_1} = \frac{P(\mu <= 175)}{P(\mu > 175)}$$

我们来计算先验优势比

```
pmean=170; pvar=25
probH=pnorm(175,pmean,sqrt(pvar))
probA=1-probH
prior.odds=probH/probA
> prior.odds
[1] 5.302974
```

那么先验优势比表示零假设的可能性是备择假设的5倍多。

下面计算后验优势比，根据正态分布的均值和标准差更新公式，我们有

```
weights=c(182, 172, 173, 176, 176, 180, 173, 174, 179, 175)
ybar=mean(weights)
sigma2=3^2/length(weights)
post.precision=1/sigma2+1/pvar
post.var=1/post.precision
post.mean=(ybar/sigma2+pmean/pvar)/post.precision
```

```
>␣c(post.mean,sqrt(post.var))
[1]␣175.7915058␣␣␣0.9320546

post.odds=pnorm(175,post.mean,sqrt(post.var))/
(1-pnorm(175,post.mean,sqrt(post.var)))
>␣post.odds
[1]␣0.2467017
␣␣␣␣
```

BF支持零假设的力度为

```
>␣BF␣=␣post.odds/prior.odds
>␣BF
[1]␣0.04652139
␣␣␣␣
```

根据上一节的公式，零假设的后验概率为

```
>␣␣postH=probH*BF/(probH*BF+probA)
>␣␣postH
[1]␣0.1978835
␣␣␣␣
```

p值为

```
>␣␣z=sqrt(length(weights))*(mean(weights)-175)/3
>␣␣1-pnorm(z)
[1]␣0.1459203
␣␣␣␣
```

Chapter 104

附: 一个例子的Bayes方法(不全)

参考 [8] 第三章

104.1 全概率公式

设$A_1, ..., A_n$为一般空间的一个分割, 则

$$B = \sum_{i=1}^{\infty} A_i B$$

由完全可加性和乘法定理得

$$P(B) = \sum_{i=1}^{\infty} P(A_i)P(B|A_i)$$

104.2 Bayes公式

例如, 设计一个能够在传送带上分开两种鱼的分类器. 记w表示鱼的状态, $w = w1$为鲈鱼, $w = w2$为鲑鱼. 出现w1和w2是随机的, 即w是一个由概率来描述其特征的随机变量.

我 们 假 定 下 一 条 鱼 是w1的 先 验 概 率 为p(w1), 是w2的 概 率 为p(w2). 先 验 概 率 可 能 来 自 于 以 往 的 捕 捞 经 验, 取 决 于 季 节 或 捕 捞 地 点. 假 设 没 有 其 它 的 种 类, 则

$$p(w1) + p(w2) = 1$$

如 果 在 观 测 之 前(即 得 到 新 的 信 息 之 前)必 须 判 断 下 一 条 鱼 的 种 类, 那 么 下 面 的 判 决 规 则 是 比 较 合 理 的: 如 果$p(w1) > p(w2)$, 判 断 为w1, 否 则 判 断 为w2. 如 果 我 们 一 直 得 不 到 任 何 新 的 信 息, 那 么 我 们 会 一 直 这 样 判 断 下 去.

但 是, 绝 大 部 分 情 况 我 们 多 少 总 可 以 获 取 一 些 新 的 信 息, 例 如 在 传 送 带 上 安 装 一 个 摄 像 头 捕 捉 鱼 的 颜 色. 不 同 的 鱼 会 呈 现 不 同 的 颜 色. 这 里 颜 色 划 分 为 白(white), 灰(gray), 黑(dark)三 种. 这 里 我 们 把 三 种 颜 色 分 别 称 为x1, x2, x3. 例 如 我 们 根 据 经 验 得 到 条 件 概 率$p(x_i|w_j)$, $i = 1, 2, 3 \quad j = 1, 2$, 又 称 为w关 于x的 似 然 函 数

Table 104.1: 条 件 概 率(似 然 函 数)

鱼 的 状 态/颜 色 状 态	x1(white)	x2(gray)	x3(dark)
x—w1	0.2	0.3	0.5
x—w2	0.8	0.1	0.1

通 过 条 件 概 率 和 先 验 概 率 我 们 可 以 得 到 联 合 概 率$p(x_i, w_j)$, 即 鱼w_j颜 色 为x_i的 概 率. 且$p(x_i, w_j) = p(w_j, x_i)$

下 面 我 们 就 要 通 过 摄 像 头 捕 捉 的 颜 色 深 浅 来 判 断 鱼 的 种 类, 例 如 某 次 测 量 得 到 颜 色x_i,

注 意 到, 联 合 概 率 可 以 写 作 两 种 形 式

$$p(w_j, x_i) = p(w_j|x_i)p(x_i) = p(x_i|w_j)p(w_j) = p(x_i, w_j)$$

重 新 组 织 上 式 前 半 部 分(或 后 半 部 分, 但 是 我 们 这 里 感 兴 趣 的 只 是 前 半 部 分)我 们 就 得 到 "贝 叶 斯 公 式"

$$p(w_j|x_i) = \frac{p(x_i|w_j)p(w_j)}{p(x_i)}$$

其中$p(x_i)$可以由下式计算得到

$$p(x_i) = \sum_{j=1}^{2} p(x_i|w_j)p(w_j)$$

这就是问题的答案. 称$p(w_j|x_i)$为后验概率,是结合先验概率, 条件概率, 和得到新的信息x_i后对鱼的种类的重新认识.

一旦x_i确定, 就可以计算出所有$p(w_j|x_i),\ \ j = 1, 2$. 此时判决规则为: 选择概率值最大的对应的鱼的种类j做出判断. 值先验概率, 条件概率

```
# 先验概率(prior, p(w_j))
Pp=matrix(c(0.4,0.6),nc=2,
      dimnames=list(c('prior'),c('w1','w2')))

# 条件概率(condition, p(x_i|w_j)(行的和为1)
# 又称为w关于x的似然函数
Pc=matrix(c(.2,.8,.3,.1,.5,.1),nr=2,
      dimnames=list(c('x|w1','x|w2'),c('x1','x2','x3')))

# 联合概率密度(jion, p(x_i,w_j)=p(xi|wj)*p(wj)) 所有和为1
# p(x_i)就是第i列的和
Pj=matrix(,2,3,
      dimnames=list(c('w1','w2'),c('x1','x2','x3')))
Pj[1,]=Pc[1,]*Pp[1]
Pj[2,]=Pc[2,]*Pp[2]

# 颜色x的概率 p(x_i)
Px=colSums(Pj)

# 查看一下
> Pp # 先验概率 p(w_j)
      w1  w2
prior 0.4 0.6
> Pc # 条件概率 p(x_i|w_j)
     x1  x2  x3
x|w1 0.2 0.3 0.5
```

```
x|w2 0.8 0.1 0.1
> Pj # 联合概率密度 p(x_i,w_j)
    x1   x2   x3
w1 0.08 0.12 0.20
w2 0.48 0.06 0.06
> Px # 颜色x的概率 p(x_i)
  x1   x2   x3
0.56 0.18 0.26

# 计算后验概率的函数
after<-function(w,x,Pp,Pc,Px){
   a<-Pc[w,x]*Pp[w]/Px[x]
   a }

# 后验概率(posterior) p(w|x)
Pa=matrix(,2,3, dimnames=list(c('p(w1|x)','p(w2|x)'),c('x1','x2','x3')))

# 计算后验概率
for (w in 1:2){
   for (x in 1:3){
      Pa[w,x]<-after(w,x,Pp,Pc,Px)}}

> Pa # 后验概率 p(w|x)
             x1       x2       x3
p(w1|x) 0.1428571 0.6666667 0.7692308
p(w2|x) 0.8571429 0.3333333 0.2307692
```

结果说明, 若

- 检测到颜色为x1(white), 因为$p(w1|x1) < p(w2|x1)$那么我们判断鱼的种类为w2
- 检测到颜色为x2(gray), 判断为w1,
- 检测到x3, 判断为w1

104.3 误差

虽然$p(w1|x1) < p(w2|x1)$我们判断鱼的种类为w2, 但是它是w1的概率是0.143, 即我们错判的可能性为0.143. 一般的, 无论我们何时观测到特定的x, 都有

$$p(error|x) = \begin{cases} p(w1|x), & if\ decision\ is\ w2 \\ p(w2|x), & if\ decision\ is\ w1 \end{cases}$$

$$\begin{aligned} ab &= (a_y b_z - a_z b_y)i + (a_z b_x - a_x b_z)j + (a_x b_y - a_y b_x)k \\ &= \begin{vmatrix} i & j & k \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{vmatrix} \end{aligned}$$

使得误差最小化等价于我们的贝叶斯判决规则:

$$if \quad p(w1|x) > p(w2|x) \quad decision: \quad w1 \qquad else: \quad w2$$

即

$$p(error|x) = min(p(w1|x), p(w2|x))$$

总的误差为

$$p(error) = \int_{-\infty}^{\infty} p(error, x)dx = \int_{-\infty}^{\infty} p(error|x)p(x)dx$$

离散形式为

$$p(error) = \sum p(error, x)dx = \sum p(error|x)p(x)$$

接前面的例子

```
# 条件误差p(error|x), 即取每列的最小值
Pex=apply(Pa,2,min)
> Pex
      x1       x2       x3
0.1428571 0.3333333 0.2307692

# 误差p(error,x)=p(error|x)p(x)
Pe=Pex*Px
> Pe
  x1   x2   x3
0.08 0.06 0.06
```

104.4 损失函数

现在我们将前面的讨论推广. 即

- 每个观测的特征x允许是多维的, 即观测$x_i = x_{i1}, \cdots, x_{id}$是一个d维向量. x所在的d维空间称为特征空间
- 类别(class)个数$c >= 2$
- 允许判别之外的其它行为(Action), $A_1, \cdots, A_a$表示a种可能的行为. 简单化起见, 这里A_k表示类别判别为k, 此时行动的总个数等于类别个数, 即$a = c$.
- 使用损失函数(loss)代替误差函数, $l(A_k|w_j)$表示类别状态为j时采取行动k的损失. 简单化起见此处

$$l(A_k|w_j) = \begin{cases} 0, & k = j1, \quad k \neq j \quad k, j = 1, \cdots, c \end{cases}$$

这种损失函数叫做"对称损失", 或"0-1损失"函数[1]

那么, 后验概率

$$p(w_j|x_i) = \frac{p(x_i|w_j)p(w_j)}{p(x_i}$$

[1]其它损失函数, 例如二次型或线性差分可能对回归任务更好, 因为这种情况下有一个序的概念, 更加错误的惩罚更加重

其中

$$p(x_i) = \sum_{j=1}^{c} p(x_i|w_j)p(w_j)$$

104.5 最小误差分类

假设我们观测到x_i并采取行为A_k, 那么其损失的期望为

$$R(A_k|x_i) = \sum_{j=1}^{c} l(A_k|w_j)p(w_j|x_i)$$

决策理论中, 一个预期的损失被称为一次风险, $R(A_i|x_i)$称为条件风险. 我们的贝叶斯决策实际上提供了一个风险最小化的过程.

总的风险为

$$R = \sum_i \sum_k R(A_k|x_i)p(x_i) = \sum_i \sum_k \sum_{j=1}^{c} l(A_k|w_j)p(w_j|x_i)p(x_i) = \sum_i \sum_k \sum_{j=1}^{c} l(A_k|w_j)p(w_j, x_i)$$

连续形式为

$$R = \int R(A|x)p(x)dx$$

如果选择A_k使得$R(A_k|x_i)$对每个x_i都最小, 那么风险将被最小化. 最小化后的风险被称为贝叶斯风险. 这种分类方法称为最小误差分类.

接前面的例子

```
# 损失函数
l=matrix(c(0,1,1,0),nr=2,
```

```
      dimnames=list('Action'=c('w1','w2'),'Truth'=c('w1','w2')))

> l
     Truth
Action w1 w2
   w1  0  1
   w2  1  0

# 观测到x_i, 采取行为j的损失的期望
# 因为我们的损失函数是'0-1损失', 故其期望损失恰好是1-Pa(后验概率)
# 采用其它的损失函数不会是这样
Rx=l%*%Pa
> Rx

Action       x1       x2       x3
   w1 0.8571429 0.3333333 0.2307692
   w2 0.1428571 0.6666667 0.7692308
> 1-Pa
            x1       x2       x3
p(w1|x) 0.8571429 0.3333333 0.2307692
p(w2|x) 0.1428571 0.6666667 0.7692308

# 总的风险(损失)为
R=Rx*rbind(Px,Px)
> R
Action   x1   x2   x3
   w1 0.48 0.06 0.06
   w2 0.08 0.12 0.20
> sum(R)
[1] 1
```

我们取最小风险, 结果与贝叶斯判决一致

- 检测到颜色为x1(white), 因为$R(w1|x1) > R(w2|x1)$那么我们判断鱼的种类为w2
- 检测到颜色为x2(gray), 判断为w1,
- 检测到x3, 判断为w1

104.6 极小化极大准则

先验概率可能并不确定, 或根本不知道. 我们希望在先验概率不知道的情况下使用此分类器, 此时设计分类器使得先验概率无论取什么值所引起的最大的总风险最小. 即最小化最大风险. 这就是极小化极大准则.

风险最大的情况就是使得贝叶斯风险最大的先验概率. 只要得到这个先验概率, 我们就可以依据其它条件概率等求得后验概率, 进而计算出最小总风险所对应的行动(或判决).

寻找使得贝叶斯风险最大的先验概率的决策边界可能比较困难, 尤其在分布复杂的时候.

104.7 判别函数

有很多形式可以描述模式分类器, 使用最多的是一种判别函数$g_i(x),\ i = 1, \cdots, c$如果

$$g_i(x) > g_j(x), \quad j \neq i$$

此分类器将此特征向量x判断为w_i.

因此, 此分类器可以看作一个计算c个判别函数, 然后选择最大判别值对应的类别的机器(或某种网络).

例如, 我们可以令$g_i(x) = -R(A_i|x)$来使用它.

如果我们将$g_i(x)$替换为$f(g_i(x))$, 其中f是单调函数, 这并不影响判别结果. 因此

$$\begin{aligned} g_i(x) &= p(w_i|x) = \frac{p(x_i|w_j)p(w_j)}{p(x)} \\ g_i(x) &= lnp(x|w_i) + lnp(w_i) - lnp(x) \end{aligned}$$

结果是一样的. 显然后面的式子计算要简单一些.

104.8 二分分类器(dichotomizer)

两类问题是多类问题的特殊形式, 但通常都拿出来单独讨论. 划分两类问题的分类器有一个专门的名称: 二分分类器.

一般, 只有两个类别时, 会定义一个简单的判别函数

$$g(x) = g_1(x) - g_2(x)$$

若$g(x) > 0$判为w1, 否则为w2.

下面是两个经常使用的形式

$$g(x) = p(w1|x) - p(w2|x)$$

$$g(x) = ln\frac{p(x|w1)}{p(x|w2)} + ln\frac{p(w1)}{p(w2)}$$

104.9 多元正态分布

多元正态分布的密度函数为

$$p(x) = \frac{1}{(2\pi)^{d/2}|\sum|^{1/2}} exp[-\frac{1}{2}(x-\mu)^t \sum^{-1}(x-\mu)] \sim N(\mu, \sum)$$

x为d维列向量, μ为d维均值向量, $\sum$是$d*d$的协方差矩阵, $|\sum|, \sum^{-1}$分别是其行列式的值和逆.

由常用形式的后面一个我们有, 最小误差分类器的判别函数是

$$g_i(x) = ln[p(x|w_i)] + ln[p(w_i)]$$

若密度函数$p(x|w_i) \sim N(\mu_i, \sum_i$为多元正态分布, 则此表达式可以较任意的估计出来

$$p(x) = -\frac{1}{2}(x-\mu_i)^t \sum_i^{-1}(x-\mu_i)] - \frac{d}{2}ln[2\pi] - \frac{1}{2}ln[|\sum_i|] + ln[p(w_i)]$$

104.9.1 最简单的情况

个特征统计独立, 且方差相同, 设为σ^2, 那么$\sum_i = I\sigma^2$, 为常数. 我们把判别式的常数项省略得到

$$g_i(x) = -\frac{||x-\mu_i||^2}{2\sigma^2} + ln[p(w_i)]$$

展开得

$$g_i(x) = -\frac{1}{2\sigma^2}[x^t x - 2\mu_t x + \mu_i^t \mu_i] + ln[p(w_i)]$$

实际上$x^t x$对所有i相等, 省略后我们得到等价的线性判别函数

$$g_i(x) = u_i^t x + u_{i0}$$

其中

$$u_i = \frac{1}{2\sigma^2}\mu_i$$

$$u_{i0} = -\frac{1}{2\sigma^2}\mu_i^t \mu_i + ln[p(w_i)]$$

称u_{i0}为第i个方向的阈值或偏置.

使用线性判别函数的分类器称为"线性机器". 其判别面是一些超平面.

104.9.2 TODO:复杂情况

104.10 二维高斯分布的例子

设两个类别的先验概率相等, $p(w1) = p(w2) = 0.5$, 条件概率$p(x|w1) \sim N(\mu_1, \sum_1)$, $p(x|w2) \sim N(\mu_2, \sum_2)$. 具体数值为

```
# 先验概率
p1=0.5
p2=0.5
# 条件概率的参数
mu1=matrix(c(3,6),nc=1)
mu2=matrix(c(3,-2),nc=1)
sigma1=matrix(c(.5,0,0,2),nc=2)
sigma2=matrix(c(2,0,0,2),nc=2)
> mu1
     [,1]
[1,]    3
[2,]    6
> mu2
     [,1]
[1,]    3
[2,]   -2
> sigma1
     [,1] [,2]
[1,]  0.5    0
[2,]  0.0    2
> sigma2
     [,1] [,2]
[1,]    2    0
[2,]    0    2
```

代入判别方程式$g_1(x) = g_2(x)$, 得到判别边界为

$$x_2 = 3.514 - 1.125x_1 + 0.1875x_1^2$$

是顶点为$[3, 1.83]$的抛物线.

```
g<-function(x,mu,sigma,p){
   d=length(x)
   res=-0.5 *t(x-mu)%*%solve(sigma)%*%(x-mu)-d/2*log(2*pi)-.5*log(det(sigma))+lo
   res}

# g_1(x)>g_2(x)故属于类别1
x=matrix(c(3,6),nc=1)
g(x,mu1,sigma1,p1)
```

```
g(x,mu2,sigma2,p2)

> g(x,mu1,sigma1,p1)
         [,1]
[1,] -2.531024
> g(x,mu2,sigma2,p2)
         [,1]
[1,] -19.22417

# 绘制判别曲面
# x_2=3.514-1.125x_1+0.1875x_1^2
b=3.514
a1=-1.125
a2=0.1875
x1=seq(-3,10,by=0.1)
x2=b+a1*x1+a2*x1^2

> plot(x2~x1,xlim=c(-3,10),ylim=c(-3,10))
> text(mu1,"mu1")
> text(mu2,"mu2")
```

104.11 贝叶斯置信网络

参考 [8], Page 47-49

104.11.1 描述

下面是一个简单的置信网络,

```
季节          光泽度
  A--->    --->C
      X(鱼)
  B--->    --->D
位置          厚度
```

其 中A,B是X的 父 节 点, C,D是X的 子 节 点. X确 实 依 赖 于A,B, 而C,D依 赖 于X. 假 设A,B,C,D互 相 没 有 关 系.

- A表 示 季 节, 可 以 是1=冬, 2=春, 3=夏, 4=秋.
- B表 示 捕 捞 位 置, 1=北 方, 2=南 方
- X代 表 鱼, 仅 有 两 种, 1=鲑 鱼, 2=鲈 鱼
- C表 示 光 泽 度, 1=亮, 2=中, 3=暗
- D表 示 厚 度, 1=厚, 2=薄

```
# 条 件 概 率: P(x|A),P(x|B),P(C|x),P(D|x),
# 各 个 季 节 发 现 两 种 鱼 的 概 率
PA=matrix(c(.9,.3,.4,.8,.1,.7,.6,.2),nc=2,
      dimnames=list(c("冬","春","夏","秋"),c("鲑 鱼","鲈 鱼")))
# 各 个 位 置 发 现 鱼 的 概 率
PB=matrix(c(.65,.25,.35,.75),nc=2,
      dimnames=list(c("北","南"),c("鲑 鱼","鲈 鱼")))
# 两 种 鱼 的 光 泽 度 概 率
PC=matrix(c(.33,.8,.33,.1,.34,.1),nc=3,
      dimnames=list(c("鲑 鱼","鲈 鱼"),c("亮","中","暗")))
# 两 种 鱼 的 厚 度 概 率
PD=matrix(c(.4,.95,.6,.05),nc=2,
      dimnames=list(c("鲑 鱼","鲈 鱼"),c("宽","窄")))
```

```
> PA
   鲑 鱼 鲈 鱼
春  0.9  0.1
夏  0.3  0.7
秋  0.4  0.6
冬  0.8  0.2
> PB
   鲑 鱼 鲈 鱼
北 0.65 0.35
南 0.25 0.75
> PC
      亮   中   暗
鲑 鱼 0.33 0.33 0.34
鲈 鱼 0.80 0.10 0.10
```

```
> PD
      宽   窄
鲑鱼 0.40 0.60
鲈鱼 0.95 0.05
```

104.11.2 计算父节点条件下的概率

```
################################
# 通过父节点的条件概率计算条件概率, Ppar=P(x|A,B)的概
率, 每一行都归一化了.
# 此处我们知道P(x|A)和P(x|B)
# 假设A,B独立的话, 我们可以这样来估计Ppar=P(x|A,B)
N<-t(PB[1,]*t(PA))/colSums(PB[1,]*t(PA))
S<-t(PB[2,]*t(PA))/colSums(PB[2,]*t(PA))
Ppar=array(c(N,S),dim=c(4,2,2),dimnames=list("季节"=c("冬","春","夏","秋"),"种
类"=c("鲑鱼","鲈鱼"),"位置"=c("北","南")))
```

```
> Ppar
, , 位置 = 北

   种类
季节      鲑鱼       鲈鱼
  冬 0.9435484 0.05645161
  春 0.4431818 0.55681818
  夏 0.5531915 0.44680851
  秋 0.8813559 0.11864407

, , 位置 = 南

   种类
季节      鲑鱼       鲈鱼
  冬 0.7500000 0.2500000
  春 0.1250000 0.8750000
  夏 0.1818182 0.8181818
  秋 0.5714286 0.4285714
```

```
################################
# 父节点的已知情况
```

```
# A: 季节的先验概率, 已知现在为冬季
Ps=matrix(c(1,0,0,0),nc=1,dimnames=list(c("冬","春","夏","秋"),c("季节")))
# B: 位置的先验概率, 已知渔民喜欢在南面捕鱼,概率为0.8, 背面为0.2
Ppos=matrix(c(0.2,0.8),nc=1,dimnames=list(c("北","南"),c("位置")))

# 下面计算父节点下x1(鲑鱼)的概率P(x|A,B)*Ps*Ppos
# 查看鲑鱼的概率
> Ppar[,1,]
   位置
季节       北       南
  冬 0.9435484 0.7500000
  春 0.4431818 0.1250000
  夏 0.5531915 0.1818182
  秋 0.8813559 0.5714286

# 产生先验概率矩阵
> a=Ps[,1]%o%Ppos[,1]
   北  南
冬 0.2 0.8
春 0.0 0.0
夏 0.0 0.0
秋 0.0 0.0

# 父节点下x1(鲑鱼)的概率 P(x1|A,B)*Ps*Ppos
# 可以把它们作为计算子节点后验概率时的先验概率
> sum(Ppar[,1,]*a)
[1] 0.7887097
# 顺便计算父节点下x2(鲈鱼)的概率
> sum(Ppar[,2,]*a)
[1] 0.2112903
```

104.11.3 计算子节点下的后验概率

将父节点下的x的概率作为先验概率, 那么子节点下的后验概率就是最终全部条件下x出现的概率.

```
################################
现在已知x1出现的概率为0.788
# 子节点的测量结果
# C: 观测结果比较亮, 亮的概率为0.75,中的概率为0.25
Pcol=matrix(c(0.75,0.25,0),nc=1,dimnames=list(c("亮","中","暗"),c("颜
色")))
# D: 观测宽度因为遮挡,无法判断, 故宽窄的概率均为0.5
Plen=matrix(c(0.5,0.5),nc=1,dimnames=list(c("宽","窄"),c("位置")))

# x1时子节点的概率P(C,D|x1)=P(C|x1)P(D|x1)
> sum(PC[1,]*Pcol)*sum(PD[1,]*Plen)
[1] 0.165
> sum(PC[2,]*Pcol)*sum(PD[2,]*Plen)
[1] 0.3125
```

104.11.4 总结

Part XIII

图论

Chapter 105

图算法(graph algorithm)

105.1 参考文献

参考文献 Robert Sedgewick *ALGORITHMS IN C++, PART 5–Graph Algorithms(Third Edition)* 影印版. Pearson Education 出版集团. 高等教育出版社. 2002.10 [31]对图论算法有很好的描述, 并且使用C++实现.

C++ 准标准库 boost 库中的 BGL 是图算法的工业级实现. 且有一个很好的文档介绍图算法.

graph包页面有几个 vignette 介绍 graph 包如何使用.

RBGL 也有 vignette 介绍.

105.2 包

参考网页 CRAN Task View: gRaphical Models in R

graph: 处理图数据结构. 提供基本的图定义和函数.

RBGL: boost 图论包, 依赖于 graph. 提供图算法. 例如最短路

径, 最小连接等.

Rgraphviz: 提供渲染功能, 绘制图节点与连接. 提供不同层的算法和节点绘制, 线类型, 颜色等参数可以控制.(已经从CRAN库删除, 早期版本还可以获得.) 转移到了 bioconductor 项目下面了. 安装的时候需要 graphviz-dev 库支持. 还要将 libdotneato.so.0 映射到 local/lib/ 下. 可能这是系统路径设置的问题

```
sudo apt-get install graphviz-dev
sudo ln -s /usr/lib/graphviz/libdotneato.so.0 /usr/local/lib/
```

network: 建立和修改网络对象. 可以描述很多关系数据类型, 支持任意顶点, 边, 图属性.

105.3 基本概念

G: 图, 由点V和边E构成.

walk: 一系列点$v_1, \cdots, v_k$, 对于所有的i, $[v_i, v_{i+1}]$在E中.

path: 一个walk, 没有重复的点.

cycle: path, 但是开始和结束点为同一个点.

directed graph: 边有方向. 即$[v_j, v_k]! = [v_k, v_j]$

directed acyclic graph(DAG): directed graph, 没有 directed cycle.

in-degree of vector v: 所有到达点v的边的数目.

out-degree of vector v: 所有从v出发的点的个数.

network N: 有向图, 并且: 有一个原点s, 其 in-degre=0. 有一个终点t, out-degree=0, 可以到达每个边.

flow in N: 对network N每条边赋值, 但是不超过其最大限制值.

所有的内部点的流入和流出相同(在点上没有截留), s只有流出, t只有流入.

其它概念随例子定义.

105.4 graph包-基本图操作

105.4.1 graph类

graph 类是 S4 类, 为包的基础虚类. 所有其它的相关类都继承自它. 不能使用它创建实例.

具体的slot变量和相关的method请看帮助.

```
> library("graph")
> getClass("graph")
Virtual Class \graph" [package "graph"]

Slots:

Name:    edgeData   nodeData renderInfo  graphData
Class:   attrData   attrData renderInfo       list

Known Subclasses: "graphNEL", "graphAM", "distGraph", "clusterGraph"
```

slot变量说明

- edgemode: 是否有向
- graphData: 最近加入, 可以保存任意的图的属性
- edgeData, nodeData: 保存边与顶点.

graphNEL: 使用列表保存节点之间的连接. 节点为vector, 边为list. list的每个元素对应一个节点, 值为从那个节点出发的边. 若是有向图, 所有的边出现两次.

graphAM: 使用邻近矩阵保存边, 矩阵为正方, 行名称与列名称必须一致. 若是无向图, 矩阵必须对称. 对于不考虑边的长度的图, 一般矩阵的i,j=0表示两个节点i,j无边, 1表示有边. 如果考虑边的距离, 使用 distGraph, 是此类的一个特例.

distGraph: 特例化的类, 直接使用距离矩阵, 并有特殊的阈值能力(special thresholding capabilities).[1] 不清楚是否属于 graphAM 类.

clusterGraph: 特例化的类, 可以表示聚类算法的结果作为图. 一本作为节点, 在同一类内的样本有边, 不同的聚类的样本没有. 其实例必须是无向图(edgemode=undirected). 如果重置edgemode, 就会强制转换为其它的类.

105.4.2 Multi-graphs类

Multi-graphs 的定义并不是很清晰. 对于生物计算来讲, 可能有用. 产生此类的一个重要原因是表示蛋白质相互作用的需要.

Multi-graphs 的定义: 包括两部分. 一个是节点集合, 一个是一系列的边集合. 每个边集合对应一个可能的节点间的连接方式. 使用$G = (V, E_L)$表示. V是节点集合, $E_L = (E_1, \cdots, E_L)$为L个边集合. 每个表示不同的节点关系. 边可以是有向和无向的, 指向自己的边也允许.

不清楚是否有必要区分Multi-graphs和graphs. 但是可以肯定的是, Multi-graphs对边集合的支持更灵活, 允许有不同的结构. 当前的定义没有扩展图的概念. 定义为

```
> getClass("multiGraph")
Class \multiGraph" [package "graph"]

Slots:

Name:     nodes     edgeL  nodeData graphData
Class:    vector     list  attrData     list
```

[1]可能是超过某阈值的距离就不计算了

nodes: 节点向量

edgeL: 可能有名称的list, 元素为边集合类edgeSet.

edgeSet 类是一个虚类, 有几个不同的扩展. 包括 edgeSetNEL, edgeSetAM

边属性在 edgeSet 类的 edgeData 变量里. 这样可以使 Multigraphs 的 edgeSet 有完全不同的属性. 另一个方法是拥有一个list, 保证 edgeSet list 包含对于所有边相同的属性.

105.4.3 Bipartite Graphs

Bipartite Graphs 是图的节点可以分为两类, 例如V1, V2. 只有V1,V2之间有边连接, V1,V2内部的节点之间没有连接.

105.4.4 获取图的信息

参考文献 graph.pdf graphAttributes.pdf

其它函数请参考 graph-class, graphNEL,graphAM 的帮助.

inEdges: 返回指向节点的边

numNodes: 图中的节点数

```
library(graph)
set.seed(123)
# 创建随机连接的图. 节点15个, 边100条.
# 节点名称必须是字符串.
# 返回 graphNEL
g1 = randomEGraph(LETTERS[1:15], edges = 100)
g2 = randomEGraph(LETTERS[1:15], edges = 10)
```

```
#----图相关函数----------------
> g1
A graphNEL graph with undirected edges
Number of Nodes = 15
Number of Edges = 100

> nodes(g1) # 所有节点的向量
 [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O"

> degree(g1) # 每个节点的连接数
 A  B  C  D  E  F  G  H  I  J  K  L  M  N  O
14 14 12 13 12 12 13 14 14 13 14 14 14 13 14
> degree(g2)
A B C D E F G H I J K L M N O
2 1 0 0 2 1 2 2 1 3 2 1 0 1 2

> sum(degree(g1)) # 拥有的边数
[1] 200

> adj(g1,"A") # 节点A的邻居, 即与此节点有边的节点
$A
 [1] "H" "D" "I" "M" "C" "O" "G" "N" "E" "F" "J" "K" "B" "L"
> adj(g2,"A")
$A
[1] "J" "I"

# acc() 函数: 返回带有名称的list.
# 可以用于判断两个节点是否连通
# 名称为从当前节点可以到达的节点, 值为经过多少条边
> acc(g1, c("E", "G"))
$E
A B C D F G H I J K L M N O
1 1 1 1 1 2 1 1 2 1 1 1 1 1

$G
A B C D E F H I J K L M N O
1 1 1 1 2 1 1 1 1 1 1 1 1 1
> acc(g2,"A")
$A
E F H I J K N O
2 3 3 1 1 4 5 2
```

```
#-------子图-------------
# 子图. 选择一个图的部分节点构成此图的子图.
> sg2 = subGraph(c("A", "E", "F", "L"), g2)
> sg2
A graphNEL graph with undirected edges
Number of Nodes = 4
Number of Edges = 0 # 此子图节点之间无连接
> adj(sg2,"A") # 果然没有邻居
$A
character(0)

# 从原图g2到其子图sg2的边. 每个sg2的节点的边构成一个向量.
> boundary(sg2, g2)
$A
[1] "I" "J"

$E
[1] "H" "J"

$F
[1] "O"

$L
[1] "G"

# 只能子图到原图, 反过来不行
> boundary(g2,sg2)
错误于boundary(g2, sg2) : some nodes are not in the graph

# g1 的子图. 查看子图的边和边的权值
> sg1 = subGraph(c("A", "E", "F", "L"), g1)
> edges(sg1)
$A
[1] "E" "F" "L"

$E
[1] "A" "F" "L"

$F
```

```
[1] "E" "A" "L"

$L
[1] "A" "F" "E"

> edgeWeights(sg1)
$A
E F L
1 1 1

$E
A F L
1 1 1

$F
E A L
1 1 1

$L
A F E
1 1 1
```

105.4.5 手工创建图,增加—删除节点和边

```
#------创建图--------------
# 手工创建 NEL 图

V <- LETTERS[1:4]
edL1 <- vector("list", length = 4)
names(edL1) <- V
# 设置第i个节点连接到第c(2, 1, 4, 3)[i] 节点
for (i in 1:4) edL1[[i]] <- list(edges = c(2, 1, 4, 3)[i], weights = sqrt(i))
gR <- new("graphNEL", nodes = V, edgeL = edL1)

# 下面查看结果

> edL1 # 用于创建的边列表
$A
```

```
$A$edges
[1] 2

$A$weights
[1] 1


$B
$B$edges
[1] 1

$B$weights
[1] 1.414214


$C
$C$edges
[1] 4

$C$weights
[1] 1.732051


$D
$D$edges
[1] 3

$D$weights
[1] 2

> gR # NEL 图
A graphNEL graph with undirected edges
Number of Nodes = 4
Number of Edges = 2

> edges(gR) # 边的情况.
$A
[1] "B"

$B
[1] "A"
```

```
$C
[1] "D"

$D
[1] "C"

> edgeWeights(gR) # 边的权值.
$A
      B
1.414214

$B
A
1

$C
D
2

$D
      C
1.732051

# 再创建一个 NEL 图
edL2 <- vector("list", length = 4)
names(edL2) <- V
for (i in 1:4) edL2[[i]] <- list(edges = c(2, 1, 2, 1)[i], weights = sqrt(i))
gR2 <- new("graphNEL", nodes = V, edgeL = edL2, edgemode = "directed")


#----增加|删除边和节点----------------
# 增加节点
>  gX = addNode(c("E", "F"), gR)
>  gX
A graphNEL graph with undirected edges
Number of Nodes = 6
Number of Edges = 2

# 增加边. from  to
# 最后的参数是边的权值.
```

```
> gX2 = addEdge(c("E", "F", "F"), c("A", "D", "E"), gX, c(1, 2,
+    3))
> gX2
A graphNEL graph with undirected edges
Number of Nodes = 6
Number of Edges = 5

# 重复增加, 给出警告. 或许需要修改权值.
> gX= addEdge(c("E", "F", "F"), c("A", "D", "E"), gX)
Warning message:
In .local(from, to, graph) :
  The following edges already exist and will be replaced:
E|A, F|D, F|E

#----------删除节点和边使用----------------
removeEdge, removeNode


#------合并节点-----------
# 将被合并的节点的所有in和out的边合并到一个节点
# 参数 collapseFun=sum 默认将权值求和
# 下面是将节点 A,B 合并, 新的名称为W
>  gR3 = combineNodes(c("A", "B"), gR, "W")
>  gR3
A graphNEL graph with undirected edges
Number of Nodes = 3
Number of Edges = 1

> edges(gR3) # 查看边的情况
$C
[1] "D"

$D
[1] "C"

$W
character(0)

#-----去除节点所有的边, 包括in和out的----------
> gX2=clearNode("A", gX)
> edges(gX2)
```

```
$A
character(0)

$B
character(0)

$C
[1] "D"

$D
[1] "C" "F"

$E
[1] "F"

$F
[1] "D" "E"
```

105.4.6 underlying graph

underlying graph: 一个有向图, 当忽略其边的方向后得到的图叫做 underlying graph. 即相反的方向也成为一个连接. 例如, 当一个有向图 A与B连接, 但是B不与A连接, 变为无向图后, B与A是连接的.

当图本身是 undirected, 此函数简单的返回. 当是 multi-graph, 错误.

```
edL2 <- vector("list", length = 4)
names(edL2) <- V
for (i in 1:4) edL2[[i]] <- list(edges = c(2, 1, 2, 1)[i], weights = sqrt(i))
gR2 <- new("graphNEL", nodes = V, edgeL = edL2, edgemode = "directed")

> gR2u=ugraph(gR2)
> gR2u
A graphNEL graph with undirected edges
Number of Nodes = 4
```

```
Number of Edges = 3

> isDirected(gR2)
[1] TRUE
> isDirected(gR2u)
[1] FALSE

# gR2 的B与C不连接, A与D也不连接, 变为无向图 gR2u 后,
> edges(gR2)
$A
[1] "B"

$B
[1] "A"

$C
[1] "B"

$D
[1] "A"

> edges(gR2u)
$A
[1] "B" "D"

$B
[1] "A" "C"

$C
[1] "B"

$D
[1] "A"
```

105.4.7 jion,union,intersection,complement

join: 两个图联合. 相同的节点的边合并, 返回的图的所有权值重置为1. 若需要保留原来的权值, 请在新图上执行 addEdge 来设置边的权值.

union: 两个图合并. 必须节点相同. 返回 graphNEL 类.

intersection: 两个图的交集部分. 两个图节点必须相同. 若两个图的某节点都有边, 则有边. 即边也取and操作.

complement: 计算图的补集. 如果提供的图没有边, 则返回的图此处有边.

```
# 创建图

V <- LETTERS[1:4]
edL1 <- vector("list", length = 4)
names(edL1) <- V
# 设置第i个节点连接到第c(2, 1, 4, 3)[i] 节点
for (i in 1:4) edL1[[i]] <- list(edges = c(2, 1, 4, 3)[i], weights = sqrt(i))
gR <- new("graphNEL", nodes = V, edgeL = edL1)

gX = addNode(c("E", "F"), gR)
gX2 = addEdge(c("E", "F", "F"), c("A", "D", "E"), gX, c(1, 2, 3))

> gR
A graphNEL graph with undirected edges
Number of Nodes = 4
Number of Edges = 2
> gX
A graphNEL graph with undirected edges
Number of Nodes = 6
Number of Edges = 5

#-----join--------
# join 联合两个图.
# 此处因为gR是gX的子图, 所以gJ与gX一样
gJ=join(gR,gX)
```

```
> gJ
A graphNEL graph with undirected edges
Number of Nodes = 6
Number of Edges = 5
> edges(gJ)
$A
[1] "B" "E"

$B
[1] "A"

$C
[1] "D"

$D
[1] "C" "F"

$E
[1] "A" "F"

$F
[1] "D" "E"

#---union---------
> union(gR,gX)
错误于union(gR, gX) : graphs must have the same nodes
此外: Warning message:
In xN != yN : 长的对象长度不是短的对象长度的整倍数

# 创建新的图. 节点与gR一样. 边不同
V <- LETTERS[1:4]
edL1 <- vector("list", length = 4)
names(edL1) <- V
for (i in 1:4) edL1[[i]] <- list(edges = c(1,2, 4, 3)[i], weights = sqrt(i))
gR3 <- new("graphNEL", nodes = V, edgeL = edL1)

> edges(gR)
$A
[1] "B"
```

```
$B
[1] "A"

$C
[1] "D"

$D
[1] "C"

> edges(gR3)
$A
[1] "A"

$B
[1] "B"

$C
[1] "D"

$D
[1] "C"

> gU=union(gR,gR3)
> gU
A graphNEL graph with undirected edges
Number of Nodes = 4
Number of Edges = 4
> edges(gU)
$A
[1] "B" "A"

$B
[1] "A" "B"

$C
[1] "D"

$D
[1] "C"

#------intersection---------
```

```
> gI=intersection(gR,gR3)
> gI
A graphNEL graph with undirected edges
Number of Nodes = 4
Number of Edges = 1
> edges(gI)
$A
character(0)

$B
character(0)

$C
[1] "D"

$D
[1] "C"

#-----complement-------------
> gC=complement(gR)
> gC
A graphNEL graph with undirected edges
Number of Nodes = 4
Number of Edges = 4
> edges(gC)
$A
[1] "C" "D"

$B
[1] "C" "D"

$C
[1] "A" "B"

$D
[1] "A" "B"
```

105.4.8 随机创建图

randomEGraph: 随机边的图, 可以指定边出现的概率

randomGraph: 节点的个数是指定的, 层的数目是固定的. 边随机. ???

randomNodeGraph: 指定节点分布.???

```
set.seed(23)
V <- LETTERS[1:20]
M <- 1:4
g1 <- randomGraph(V, M, 0.2)

set.seed(123)
c1 <- c(1, 1, 2, 4)
names(c1) <- letters[1:4]
g2 <- randomNodeGraph(c1)
```

105.4.9 subGraph,connComp

```
set.seed(123)
c1 <- c(1, 1, 2, 4)
names(c1) <- letters[1:4]
g2 <- randomNodeGraph(c1)

> edges(g2)
$a
character(0)

$b
character(0)

$c
[1] "c"

$d
```

```
[1] "a" "b" "d"

#---最大连接子图-----------
# g2 的c单独自己连接. abd连接. 构成两个分隔的子图.
> g2cc<-connComp(g2)
> g2cc
[[1]]
[1] "a" "b" "d"

[[2]]
[1] "c"

#--------子图----------
> g2.sub <- subGraph(g2cc[[2]], g2)
> g2.sub
A graphNEL graph with directed edges
Number of Nodes = 1
Number of Edges = 1

> g2.sub2<-subGraph(g2cc[[1]], g2)
> g2.sub2
A graphNEL graph with directed edges
Number of Nodes = 3
Number of Edges = 3
> edges(g2.sub2)
$a
character(0)

$b
character(0)

$d
[1] "a" "b" "d"

> g2.sub3<-subGraph("a", g2)
> edges(g2.sub3)
$a
character(0)

> g2.sub3<-subGraph(c("a","b"), g2)
> edges(g2.sub3)
```

```
$a
character(0)

$b
character(0)

> g2.sub3<-subGraph(c("a","d"), g2)
> edges(g2.sub3)
$a
character(0)

$d
[1] "a" "d"
```

105.4.10 DFS(深度优先算法)

此函数要求图是全连接的. 更好的算法在 RBGL 包里.

105.4.11 其它函数

ftM2adjM: converts a from-to matrix into an adjacency matrix.

An aM is an affiliation matrix which is frequently used in social networks analysis. The rows of aM represent actors, and the columns represent events. A one, 1, in the ith row and jth column represents the affiliation of the ith actor with the jth event. The function aM2bpG coerces a aM into an instance of the graphNEL where the nodes are both the actors and the events (there is currently no bipartite graph representation, although one could be added).

The two functions sparseM2Graph and graph2SparseM provide coercion between graphNEL in- stances and sparse matrix representations. Currently we rely on the SparseM of Koncker and Ng for the sparse matrix implementation.

105.5 RBGL包-图算法

具体算法参考文献[31]

参考文献: RBGL 页面的 vignette RBGL.pdf:

L. Long, VJ Carey, and R. Gentleman *RBGL: R interface to boost graph library* April 21, 2009

105.5.1 使用的数据

下面是例子使用的数据. 后面使用的时候不再导入. 使用XML 包解析数据.

```
library(RBGL)
library(Rgraphviz) # 用于绘制渲染图
data(FileDep)
con <- file(system.file("XML/bfsex.gxl", package = "RBGL"))
bf <- fromGXL(con)
close(con)

con <- file(system.file("XML/dfsex.gxl", package = "RBGL"))
df <- fromGXL(con)
close(con)

con <- file(system.file("XML/dijkex.gxl", package = "RBGL"))
dijk <- fromGXL(con)
close(con)

con <- file(system.file("XML/conn.gxl", package = "RBGL"))
coex <- fromGXL(con)
close(con)

con <- file(system.file("XML/conn2.gxl", package = "RBGL"))
coex2 <- fromGXL(con)
close(con)
```

```
con <- file(system.file("XML/conn2iso.gxl", package = "RBGL"))
coex2i <- fromGXL(con)
close(con)

con <- file(system.file("XML/kmstEx.gxl", package = "RBGL"))
km <- fromGXL(con)
close(con)

con <- file(system.file("XML/biconn.gxl", package = "RBGL"))
bicoex <- fromGXL(con)
close(con)

con <- file(system.file("XML/ospf.gxl", package = "RBGL"))
ospf <- fromGXL(con)
close(con)

con <- file(system.file("dot/joh.gxl", package = "RBGL"))
joh <- fromGXL(con)
close(con)

con <- file(system.file("XML/hcs.gxl", package = "RBGL"))
hcs <- fromGXL(con)
close(con)

con <- file(system.file("XML/snacliqueex.gxl", package = "RBGL"))
kclex <- fromGXL(con)
close(con)

con <- file(system.file("XML/snacoreex.gxl", package = "RBGL"))
kcoex <- fromGXL(con)
close(con)
```

105.5.2 深度优先搜索(DFS)

深度优先搜索(Depth First Search).

返回两个向量. 一个是搜索发现节点的顺序. 一个是搜索停止的顺序(注: 大概就是递归返回的顺序).

```
# df 图见文档.
# 从 y 开始搜索.
> print(dfs.res <- dfs(df, "y"))
$discovered # 发现的顺序
[1] "y" "x" "v" "w" "z" "u"

$finish # 停止的顺序.
[1] "v" "x" "y" "z" "w" "u"

# 从u开始搜索
> print(dfs.res <- dfs(df, "u"))
$discovered
[1] "u" "v" "y" "x" "w" "z"

$finish
[1] "x" "y" "v" "u" "z" "w"
```

105.5.3 广度优先搜索(BFS)

广度优先搜索(Breadth First Search).

返回一个向量, 包含搜索的顺序.

```
> print(bfs.res <- bfs(bf, "s"))
[1] "s" "w" "r" "t" "x" "v" "u" "y"
```

105.5.4 最短路径(shortest paths)

边长可以是距离, 概率等. 最短路径就是最小距离, 最小概率.

当有负边长的环时, 距离为$-Inf$

有两类算法可以使用

1. 第一类算法: 寻找其中一个节点, 例如s, 到其它所有节点的最短路径. 可用算法为 Dijkstra's, Bellman-Ford's and DAG,

   - Dijkstra 算法解决第一类问题的(有向图或无向图), 边的长度不能为负. 如果所有边的长度相同, 使用DFS搜索代替.

     dijkstra.sp() 函数计算某节点(默认从第一个节点)到所有其它节点的最短路径

     返回两个向量. 1. distance: 到其它节点的最短路径 2. penilt: 从节点到初始节点A回溯的最近的祖先节点的下标(在nodes(g)中 的). (penult: A vector of indices (in 'nodes(g)') of predecessors corresponding to each node on the path from that node back to 'start')
   - sp.between() 计算指定两个节点的最短路径. 并给出详细的路径.
   - Bellman-Ford's 算法也解决第一类问题的(有向图或无向图), 边的长度可以为负.

     bellman.ford.sp() 函数计算某节点(默认从第一个节点)到所有其它节点的最短路径
   - DAG 算法适用于有权值的, 有向非环图(DAG). 比 Dijkstra 和 Bellman-Ford 效率都高. 当权值一样, 使用深度优先算法.

     dag.sp() 函数.

2. 第二类算法: 寻找所有点对之间的最小的距离. 可用算法为: Johnson's and Floyd Warshall's.

   johnson.all.pairs.sp() 函数返回一个矩阵. 元素为两个节点的最短路径. 可用为负.

   floyd.warshall.all.pairs.sp() 函数使用 Floyd Warshall's 算法, 从一个致密(dence graph)图寻找点对之间的最小距离. 返回距离矩阵.

```
> nodes(dijk)
[1] "A" "B" "C" "D" "E"
> edgeWeights(dijk)
$A
C
1
```

```
$B
B D E
2 1 2

$C
B D
7 3

$D
E
1

$E
A B
1 1

> dijkstra.sp(dijk)
$distances
A B C D E
0 6 1 4 5


$penult
A B C D E
1 5 1 3 4

$start # 开始节点
A
1

> nodes(ospf)[6]
[1] "RT3"

# 计算某节点(默认从第一个节点)到所有其它节点的最短路径
> dijkstra.sp(ospf, nodes(ospf)[6])
$distances
 RT1   N1   N3  RT2   N2  RT3  RT6   N4  RT4  RT5  RT7  N12  N13  N14 RT10   N6
   1    4    1    1    4    0    8    2    1    9   15   17   17   17   15   16
 N15  RT8   N7  RT9   N9  N11   N8 RT11 RT12  N10   H1
  24   16   20   19   19   22   18   18   19   21   29
```

```
$penult
 RT1   N1   N3  RT2   N2  RT3  RT6   N4  RT4  RT5  RT7  N12  N13  N14 RT10   N6
   3    1    6    3    4    6    6    6    3    9   10   10   10   10    7   15
 N15  RT8   N7  RT9   N9  N11   N8 RT11 RT12  N10   H1
  11   16   18   21   24   20   15   23   21   25   25

$start
RT3
  6

# 计算指定两个节点的最短路径. 并给出详细的路径.
> sp.between(ospf, "RT6", "RT1")
$`RT6:RT1`
$`RT6:RT1`$length
[1] 7

$`RT6:RT1`$path_detail
[1] "RT6" "RT3" "N3"  "RT1"

$`RT6:RT1`$length_detail
$`RT6:RT1`$length_detail[[1]]
RT6->RT3  RT3->N3  N3->RT1
       6        1        0
```

105.5.5 最小展开树

最小展开树(Minimum spanning tree, MST), 发现边的一个子集, 包含所有节点, 不包括环, 其边长和最小.

两个算法可用. Kruskal's algorithm and Prim's algorithm. 两个都针对无向图, 都返回边, 权值, 节点的列表

```
# Kruskal's algorithm
# km 是有向图, 但是被当做无向图处理.
> mstree.kruskal(km)
$edgeList
```

```
    [,1] [,2] [,3] [,4]
from "A"  "E"  "E"  "B"
to   "C"  "D"  "A"  "D"

$weights
      [,1] [,2] [,3] [,4]
weight    1    1    1    1

$nodes
[1] "A" "B" "C" "D" "E"

> mstree.prim(coex2)
$edgeList
    [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
from "A"  "A"  "A"  "A"  "C"  "C"  "D"  "H"
to   "A"  "B"  "C"  "D"  "E"  "G"  "H"  "F"

$weights
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
weight    0    1    1    1    1    1    1    1

$nodes
[1] "A" "B" "C" "D" "E" "G" "H" "F"
```

105.5.6 连通子图(Connected components)

Connected components: 连通图. 无向图, 图中子图, 可以连通的.

strongly connected component: 强连通图. 有向图, 子图, 其中任意两个节点u,v, 即可以从u到v, 也可以从v到u.

biconnected graph: 双向连通图, 去除任何一个节点并不会使其分为两个字图. 如果去除一个节点, 会增加连通子图的数目(即这个节点是这个图的腰, 树的任何非叶子节点都是这种点), 此节点叫做关节点(articulation point).

```
km1 <- km
```

```
km1 <- graph::addNode(c("F", "G", "H"), km1)
km1 <- addEdge("G", "H", km1, 1)
km1 <- addEdge("H", "G", km1, 1)
connectedComp(ugraph(km1))

$`1`
[1] "A" "B" "C" "D" "E"

$`2`
[1] "F"

$`3`
[1] "G" "H"

km2 <- km
km2 <- graph::addNode(c("F", "G", "H"), km2)
km2 <- addEdge("G", "H", km2, 1)
km2 <- addEdge("H", "G", km2, 1)
strongComp(km2) # 强连通子图.

$`1`
[1] "D"

$`2`
[1] "A" "B" "C" "E"

$`3`
[1] "F"

$`4`
[1] "G" "H"

# 发现所有双向连通子图
> biConnComp(bicoex)
[[1]]
[1] "B" "C" "D"

[[2]]
[1] "A" "B" "F" "E"

[[3]]
```

```
[1] "G" "H" "I"

[[4]]
[1] "A" "G"

# 寻找关节点
> articulationPoints(bicoex)
[1] "B" "G" "A"
```

当添加一个边到无向图中, 想更新其连通图信息, 可以这样.

```
> jcoex <- join(coex, hcs)
> x <- init.incremental.components(jcoex)
> incremental.components(jcoex)
[[1]]
no. of connected components
                     2

[[2]]
[1] "G" "F" "H" "E" "D" "C" "B" "A"

[[3]]
 [1] "X"  "Z"  "B4" "B3" "B2" "B1" "A3" "Y"  "A5" "A4" "A2" "A1"

# A与F是否在同一个连通图中.
> same.component(jcoex, "A", "F")
[1] TRUE
> same.component(jcoex, "A", "A1")
[1] FALSE
> jcoex <- addEdge("A", "A1", jcoex)
> incremental.components(jcoex)
[[1]]
no. of connected components
                     1

[[2]]
 [1] "X"  "Z"  "B4" "B3" "B2" "B1" "A3" "Y"  "A5" "A4" "A2" "A1" "G"  "F"  "H"
[16] "E"  "D"  "C"  "B"
```

```
# R 异 常 退 出. 函 数 有bug.
> same.component(jcoex, "A", "A1")

 *** caught segfault ***
address 0x8, cause 'memory not mapped'

Possible actions:
1: abort (with core dump, if enabled)
2: normal R exit
3: exit R without saving workspace
4: exit R saving workspace
4
```

105.5.7 Maximum Flow

, edmunds.karp.max.flow() and push.relabel.max.flow() 两 个 函 数 实 现 此 功 能.

```
> edgeWeights(dijk)
$A
C
1

$B
B D E
2 1 2

$C
B D
7 3

$D
E
1
```

```
$E
A B
1 1

> edmunds.karp.max.flow(dijk, "B", "D")
$maxflow
[1] 2

$edges
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
from "A"  "B"  "B"  "B"  "C"  "C"  "D"  "E"  "E"
to   "C"  "B"  "D"  "E"  "B"  "D"  "E"  "A"  "B"

$flows
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
flow    1    0    1    1    0    1    0    1    0

> push.relabel.max.flow(dijk, "C", "B")
$maxflow
[1] 8

$edges
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
from "A"  "B"  "B"  "B"  "C"  "C"  "D"  "E"  "E"
to   "C"  "B"  "D"  "E"  "B"  "D"  "E"  "A"  "B"

$flows
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
flow    0    0    0    0    7    1    1    0    1
```

105.5.8 Sparse Matrix Ordering

三个函数可用: cuthill.mckee.ordering(), minDegreeOrdering() and sloan.ordering().

```
> dijk1 <- ugraph(dijk)
> cuthill.mckee.ordering(dijk1)
```

```
$‘reverse cuthill.mckee.ordering‘
[1] "A" "B" "E" "C" "D"

$‘original bandwidth‘
[1] 4

$‘new bandwidth‘
[1] 3

> minDegreeOrdering(dijk1)
$inverse_permutation
[1] "B" "A" "C" "E" "D"

$permutation
[1] "B" "A" "C" "E" "D"

> sloan.ordering(dijk1)
$sloan.ordering
[1] "A" "E" "C" "B" "D"

$bandwidth
[1] 3

$profile
[1] 13

$maxWavefront
[1] 4

$aver.wavefront
[1] 2.6

$rms.wavefront
[1] 2.792848
```

105.5.9 Edge connectivity and minimum disconnecting set

Edge connectivity: 对于无向图, 它是需要把一个无向图变为两个图需要去除的边的最小数目. 函数输出为需要去除的边.

很类似 minCut 算法. 此算法考虑边权值. 返回两个分隔的图的节点.

```
> edgeConnectivity(coex)
$connectivity
[1] 2

$minDisconSet
$minDisconSet[[1]]
[1] "D" "E"

$minDisconSet[[2]]
[1] "D" "H"
```

105.5.10 Topological sort

tsort() 返回DAG按照拓扑排序顺序的节点

```
> tsort(coex) # 非有向图
错误于tsort(coex) : requires directed graph
> tsort(dijk) # 有环
character(0)
Warning message:
In tsort(dijk) : not a DAG.

> tsort(coex2)
[1] "A" "B" "C" "D" "E" "F" "G" "H"

> tsort(FileDep)
 [1] "zow_h"      "boz_h"      "zig_cpp"    "zig_o"      "dax_h"
```

```
 [6] "yow_h"      "zag_cpp"    "zag_o"      "bar_cpp"    "bar_o"
[11] "foo_cpp"    "foo_o"      "libfoobar_a" "libzigzag_a" "killerapp"
> FD2 <- FileDep
> FD2 <- addEdge(from = "bar_o", to = "dax_h", FD2)
> tsort(FD2) # 有一个环. 不是DAG
character(0)
Warning message:
In tsort(FD2) : not a DAG.
```

105.5.11 Layout

参考包 Rgraphviz.

有下面几种layout方法

The randomGraphLayout: 随机放置.

The circleLayout: 排列为正方的多边形

The kamadaKawaiSpringLayout: 针对连通无向图. 把边作为弹簧, 试图使系统能量最小化.

The fruchtermanReingoldForceDirectedLayout: 针对非权值无向图, 可能是非连通的. 把边看作压力使得节点靠在一起, 没有边使得节点分离. 最初的节点位置随机放置(使用randomGraphLayout), "width" and "height" 值的选择对行为有明显影响.

```
> randomGraphLayout(coex)
           A         B         C         D         E         H         F
x 2.247747e-05 0.6013526 0.9679557 0.5149758 0.2629062 0.08954777 0.5822297
y 8.503245e-02 0.8916113 0.1896898 0.3980084 0.7435125 0.56038993 0.8095667
         G
x 0.5919188
y 0.5117126
> circleLayout(coex)
```

```
  A         B           C         D           E         H           F
x 1 0.7071073 1.326795e-06 -0.7071054 -1.00000e+00 -0.7071091 -3.980385e-06
y 0 0.7071063 1.000000e+00 0.7071082 2.65359e-06 -0.7071044 -1.000000e+00
          G
x 0.7071035
y -0.7071101
> kamadaKawaiSpringLayout(coex)
         A         B          C          D         E         H         F
x 0.9812284 0.7105882 -0.04072935 0.04913392 -0.8469084 -0.1890417 -0.6743637
y 0.2309024 0.9317890 0.93750275 -0.02487002 -0.6190274 -1.0732776 -1.8284108
         G
x -1.380900
y -1.340714
> fruchtermanReingoldForceDirectedLayout(coex, 10, 10)
        A        B        C         D         E        HF          G
x 3.192958 0.9889099 -1.428881 -0.1548334 -0.8784356 -3.792289 -5 -2.724398
y 5.000000 5.0000000 5.000000 1.4771074 -3.2574547 -1.783194 -5 -5.000000

# 基于上面layout方法的绘制图需要自己编写函数.
crudeGraphPlot <- function(g, alg = circleLayout, ...) {
   layout <- alg(g)
   plot(layout[1, ], layout[2, ], pch = nodes(g), axes = FALSE,
      xlab = "", ylab = "", main = substitute(g), cex = 1.4)
   ee <- edges(g)
   src <- names(ee)
   ds <- function(nn1, nn2, lob) segments(lob[1, nn1], lob[2,
      nn1], lob[1, nn2], lob[2, nn2], ...)
   for (s in src) sapply(ee[[s]], function(x) ds(s, x, layout))
   invisible(NULL)
}
crudeGraphPlot(coex)
crudeGraphPlot(coex, alg = kamadaKawaiSpringLayout, col = "green")
```

105.5.12 Isomorphism

同构图: 一个图的节点和边可以一一映射到另外一个图, 称两个图为同构图.

```
> isomorphism(dijk, coex2)
$isomorphism
[1] FALSE

> isomorphism(coex2i, coex2)
$isomorphism
[1] TRUE
```

105.5.13 Vertex Coloring

sequential.vertex.coloring() 将图的每个节点对应一种颜色, 使得一个边的两个节点颜色不同. 此函数不保证使用的颜色数目最少.结果依赖于图中节点输入的顺序

```
> sequential.vertex.coloring(coex)
$`no. of colors needed`
[1] 4

$`colors of nodes`
A B C D E H F G
0 1 2 3 0 1 2 3
```

105.5.14 Transitive Closure

Transitive Closure: 有向图可以表示一个有限集合V的关系, 记为R.

Transitive digraph: 如果是双向图, 其关系是可传递的, 称为.... 例如, 有edge(u,v), edge(v,w), 那么必有有 edge(u,w).

如果D是有向图, 表示关系R, 那么有向图D^*表示R^*称为D的Transitive Closure.

即, 两个图g1, g2, 节点一一对应. 如果g1有一个路径u,v, 那么g2的u,v之间有一条边. g2就是g1的Transitive Closure.

```
> dijk.tc = transitive.closure(dijk)
> dijk.tc
A graphNEL graph with directed edges
Number of Nodes = 5
Number of Edges = 25
> dijk
A graphNEL graph with directed edges
Number of Nodes = 5
Number of Edges = 9
```

105.5.15 Wavefront, Profiles

```
ss <- 1
ith.wavefront(dijk, ss)
maxWavefront(dijk)
aver.wavefront(dijk)
rms.wavefront(dijk)
```

```
> ss <- 1
> ith.wavefront(dijk, ss)
$ith.wavefront
[1] 3
```

```
> maxWavefront(dijk)
$maxWavefront
[1] 4
```

```
> aver.wavefront(dijk)
$aver.wavefront
[1] 2.6
```

```
> rms.wavefront(dijk)
$rms.wavefront
[1] 2.792848
```

105.5.16 Betweenness Centrality and Clustering

Betweenness Centrality: 一个节点或边的 Betweenness Centrality 指它在图中的重要性. 即在所有的节点对的最短路径中, 有多少必须经过它们. Relative betweenness centrality 使用因子$2/((n-1)(n-2))$来校正.

brandes.betweenness.centrality() 算法实现 Brandes' 算法计算 calculating betweenness centrality.

betweenness.centrality.clustering() 基于 edge betweenness centrality 对图聚类

```
> brandes.betweenness.centrality(coex)
$betweenness.centrality.vertices
    A B C  D E H F G
[1,] 0 0 0 12 4 4 0 0

$edges
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
[1,] "A"  "A"  "A"  "B"  "B"  "C"  "D"  "D"  "E"  "E"   "E"   "H"   "H"   "F"
[2,] "B"  "C"  "D"  "C"  "D"  "D"  "E"  "H"  "G"  "H"   "F"   "F"   "G"   "G"

$betweenness.centrality.edges
          [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
centrality   1    1    5    1    5    5    8    8    3     1     3     3     3
          [,14]
centrality     1

$relative.betweenness.centrality.vertices
    A B C       D       E       H F G
[1,] 0 0 0 0.5714286 0.1904762 0.1904762 0 0

$dominance
[1] 0.5170068

> betweenness.centrality.clustering(coex, 0.1, TRUE)
$no.of.edges
[1] 12
```

```
$edges
    [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
from "A"  "A"  "A"  "B"  "B"  "C"  "E"  "E"  "E"  "H"   "H"   "F"
to   "B"  "C"  "D"  "C"  "D"  "D"  "G"  "H"  "F"  "F"   "G"   "G"

$edge.betweenness.centrality
           [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
centrality    1    1    1    1    1    1    1    1    1     1     1     1
```

105.5.17 基于RBGL的算法

Min-Cut: 一个cut是将节点分隔为两个不为空的子集的操作. cost是连接两个分隔的子集的边的权值和. minCut() 寻找cost最小的cut.

简单起见, S是较小的子集

```
> minCut(coex)
$mincut
[1] 2

$S
[1] "A" "B" "C" "D"

$`V-S`
[1] "E" "H" "F" "G"
```

highlyConnSG(): 如果一个图的连接$k(G) > n/2$, 称为高度连接的. 此函数将一个图分隔为高度连接的子图.

```
> highlyConnSG(coex)
$clusters
$clusters[[1]]
[1] "A" "B" "C" "D"
```

```
$clusters[[2]]
[1] "E" "H" "F" "G"
```

105.6 独立于RBGL的算法

105.6.1 maxClique

clique: 完全子图称为 clique. 即此子图任何两个节点之间都有一条边.

Maximum Clique: 寻找最大的完全子图. 此问题是 NP-完全问题.

算法来源: Function maxClique implements the algorithm from Finding all cliques of an undi-rected graph, by C. Bron and J. Kerbosch (CACM, Sept 1973, Vol 16, No. 9.), which finds all the cliques in a graph.

```
> maxClique(coex)
$maxCliques
$maxCliques[[1]]
[1] "D" "B" "C" "A"

$maxCliques[[2]]
[1] "D" "E" "H"

$maxCliques[[3]]
[1] "F" "E" "H" "G"


> maxClique(hcs)
$maxCliques
$maxCliques[[1]]
[1] "B1" "B2" "B3" "B4"

$maxCliques[[2]]
[1] "B1" "Y"
```

```
$maxCliques[[3]]
[1] "B1" "A5"

$maxCliques[[4]]
[1] "A2" "A4" "A3"

$maxCliques[[5]]
[1] "A2" "A4" "A1"

$maxCliques[[6]]
[1] "A4" "A5" "A3"

$maxCliques[[7]]
[1] "A4" "A5" "A1"

$maxCliques[[8]]
[1] "A1" "Y"

$maxCliques[[9]]
[1] "Z" "Y" "X"

$maxCliques[[10]]
[1] "Z"  "B4"
```

105.6.2 is.triangulated

triangulated: 一个图为 triangulated, 如果所有的长度大于4环都有一个弦(chord)

算法来源: We implemented the following algorithm from Combinatorial Optimization: algo-rithms and complexity (p. 403) by C. H. Papadimitriou, K. Steiglitz: G is chordal ieither G is an empty graph, or there is a v in V such that (i) the neighborhood of v, i.e.,v and its adjacent vertices, forms a clique, and (ii) recursively, G-v is chordal.

```
> is.triangulated(coex)
```

```
[1] TRUE
> is.triangulated(hcs)
[1] FALSE
```

105.6.3 separates

separates() 判断是否一个节点子集分隔其它两个节点子集.

```
> separates("B", "A", "E", km)
[1] TRUE
> separates("B", "A", "C", km)
[1] FALSE
```

105.6.4 kCores

k-Core: 子图的每个节点至少有k个同一子图的邻居

kCores() 寻找所有的k-core子图. 返回c所有节点的core的数目. 一个图的 k-core 不必是图的连接子图

算法基于 V. Batagelj and M. Zaversnik, 2002.

```
> kCores(kcoex)
A C B E F D G H J K I L M N O P Q R S T U
1 2 1 2 3 3 3 3 3 3 3 3 2 2 1 1 2 2 2 2 0
> kcoex2 <- coex2
> kCores(kcoex2)
A B C D E G H F
3 3 3 3 3 3 3 3
> kCores(kcoex2, "in")
A B C D E G H F
0 0 0 0 0 0 0 0
> kCores(kcoex2, "out")
A B C D E G H F
```

```
0 0 0 0 0 0 0 0
> g1 <- addEdge("C", "B", kcoex2)
> kCores(g1, "in")
A B C D E G H F
0 1 1 1 1 1 1 1
> g2 <- addEdge("C", "A", kcoex2)
> kCores(g2, "out")
A B C D E G H F
1 1 1 0 0 0 0 0
```

105.6.5 kCliques

在社会网络分析中, k-cliques 指其任何两个节点的最短连接不超过 k 的最大子图,

```
> kCliques(kclex)
$`1-cliques`
$`1-cliques`[[1]]
[1] "1" "2" "3"

$`1-cliques`[[2]]
[1] "2" "4"

$`1-cliques`[[3]]
[1] "3" "5"

$`1-cliques`[[4]]
[1] "4" "6"

$`1-cliques`[[5]]
[1] "5" "6"


$`2-cliques`
$`2-cliques`[[1]]
[1] "1" "2" "3" "4" "5"
```

```
$`2-cliques`[[2]]
[1] "2" "3" "4" "5" "6"


$`3-cliques`
$`3-cliques`[[1]]
[1] "1" "2" "3" "4" "5" "6"
```

105.7 Rgraphviz包-绘制图

绘制, 渲染图. 使用 graphviz 库.

105.7.1 排列

参考帮助: GraphvizLayouts

不同的排列方式有: dot, neato, twopi,

```
library(Rgraphviz)
set.seed(123)
V <- letters[1:10]
M <- 1:4
g1 <- randomGraph(V, M, .2)
if (interactive()) {
  op <- par()
  on.exit(par=op)
  par(ask=TRUE)
  plot(g1, "dot")
  plot(g1, "neato")
  plot(g1, "twopi")
#  plot(g1, "circo") 有错误, 系统异常退出
#   plot(g1, "fdp")
}
```

105.7.2 线的单双

```
rEG <- new("graphNEL", nodes = c("A", "B"), edgemode = "directed")
rEG <- addEdge("A", "B", rEG, 1)
rEG <- addEdge("B", "A", rEG, 1)
plot(rEG) # 单线, 两边有箭头
plot(rEG, recipEdges = "distinct") # 双线, 单箭头表示有向图
```

105.7.3 子图

```
library("Rgraphviz")
set.seed(123)
V <- letters[1:10]
M <- 1:4
g1 <- randomGraph(V, M, 0.2)
sg1 <- subGraph(c("a", "d", "j", "i"), g1)
sg2 <- subGraph(c("b", "e", "h"), g1)
sg3 <- subGraph(c("c", "f", "g"), g1)
subGList <- vector(mode = "list", length = 3)
subGList[[1]] <- list(graph = sg1)
subGList[[2]] <- list(graph = sg2, cluster = FALSE)
subGList[[3]] <- list(graph = sg3)
plot(g1, subGList = subGList)

# 不同的子图
sg1 <- subGraph(c("a", "c", "d", "e", "j"), g1)
sg2 <- subGraph(c("f", "h", "i"), g1)
plot(g1, subGList = list(list(graph = sg1), list(graph = sg2)))
```

105.7.4 控制颜色

```
plot(g1, attrs = list(node = list(label = "foo", fillcolor = "lightgreen"),
    edge = list(color = "cyan"), graph = list(rankdir = "LR")))
```

105.7.5 节点标记

```
nAttrs <- list()
eAttrs <- list()
z <- strsplit(packageDescription("Rgraphviz")$Description, " ")[[1]]
z <- z[1:numNodes(g1)]
names(z) = nodes(g1)
nAttrs$label <- z
eAttrs$label <- c( `a~h`="Label 1", `c~h`="Label 2")
attrs <- list(node = list(shape = "ellipse", fixedsize = FALSE))
plot(g1, nodeAttrs = nAttrs, edgeAttrs = eAttrs, attrs = attrs)
```

105.7.6 使用边权值作为标记

```
# 将边的权值作为字符串
ew <- as.character(unlist(edgeWeights(g1)))
ew <- ew[setdiff(seq(along = ew), removedEdges(g1))]
names(ew) <- edgeNames(g1)
eAttrs$label <- ew
plot(g1, nodeAttrs = nAttrs, edgeAttrs = eAttrs, attrs = attrs)
```

```
> ew
a~b a~d a~e a~f a~h b~f b~d b~e b~h c~h d~e d~f d~h e~f e~h f~h
"1" "1" "1" "1" "1" "2" "1" "1" "1" "1" "1" "1" "1" "1" "1" "1"
> edgeNames(g1)
 [1] "a~b" "a~d" "a~e" "a~f" "a~h" "b~f" "b~d" "b~e" "b~h" "c~h" "d~e" "d~f"
[13] "d~h" "e~f" "e~h" "f~h"
```

105.7.7 TODO: 增加颜色

Part XIV

信息理论

Chapter 106

信息熵与信息理论

参考文献”生物数学”[11] 第七章 生物信息论. 其中证明记录不全, 详细请参考文献.

106.1 函数介绍

信息熵的计算:

参考包 entropy, 其中函数 entropy() 计算各种熵. freq() 计算频率, mi.plugin() 计算互信息.

另外 seewave 包是时间, 波形数据分析和可视化的包. 里面有函数计算熵.

离散量的计算, 下面是自己编的:

```
# 计算离散量的函数
my.div<-function(x){
   n=sum(x)
   d=n*log(n)-sum(log(x)*x)
   d
}
```

```
# X,Y离散增量的函数
my.incdiv<-function(x,y){
    z=x+y
    dz=my.div(z)
    dx=my.div(x)
    dy=my.div(y)
    d=dz-dx-dy
    d
}
# 矩阵S离散增量=D(X)+D(Y)-D(XY),根据公式可以这样计算
my.div(colSums(S))+my.div(rowSums(S))-my.div(S)
```

106.2 信息的度量

信源的一般表示为

$$[X,P]=\begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ p_1 & p_2 & \cdots & p_n \end{bmatrix}$$

对应信息符号x, 出现的概率为p, 如果p很小, 那么当x出现, 则其携带的信息应该很多. 即信息度量应该是p的单调递减函数$f(p)$.

若另外有符号y, 出现的频率为q, 那么同时出现xy, 信息量应该为两个符号信息量$f(p), f(q)$的和, 另外, 两个符号相继出现的概率为pq, 信息量应该是$f(pq)$, 那么信息度量函数应该满足

$$f(pq)=f(p)+f(q)$$

下面证明满足该条件的连续可导函数只能是对数函数.

令$u=pq$,有

$$f(u)=f(p)+f(q)$$

将p固定, 对q求导, 得

$$\frac{df(u)du}{dudq}=\frac{df(q)}{dq}$$

即

$$f'(u)p = f'(q)$$

若$q = 1$时, $u = p$, 代入前式有

$$f'(p)p = f'(1)$$

根据$f(p)$单调递减, 可以设

$$f'(1) = d <= 0$$

解方程

$$f'(p) = \frac{d}{p}$$

得

$$f(p) = d \ln p + c$$

其中c为积分常数. 若$p = 1$时, 必然出现的信息符号不确定性为0, 信息量亦为0. 得$f(1) = 0$, 求得积分常数$c = 0$, 且$d \neq 0$, 得解

$$f(p) = d \ln p \qquad (d < 0)$$

取对数的底为$b = e^{-\frac{1}{d}}, d = -1/\ln b$, 最后得到信息量度量函数

$$f(p) = -log_b p$$

其中$b > 0$为待定系数.

106.3 Shannon信息量

106.3.1 定义

对应信源每个信息符号的概率, 信源的信息量为

$$H(X) = H(p_1, \cdots, p_n) = -\sum_{i=1}^{n} p_i log_b p_i = \sum_{i=1}^{n} p_i log_b \frac{1}{p_i}$$

$p_i = 0$时, 为了保证形式的一致, 规定$log_b 0 = 0$. (实际上概率为0的可以认为不发生, 其信息量为0, 处理之前可以删除)

对数的底取2时, 信息量的单位为 bit(binary unit). 取自然对数, 底为$e = 2.7182818\cdots$, 信息量的单位为nat(nature unit). 取常用对数, 10为底, 信息量的单位为hart(Hartley). 一般默认取自然对数, 单位即为nat.

此信息量也称为信息熵.

106.3.2 连续非负性

$H(p_1, \cdots, p_n) >= 0$, 且是$p_1, \cdots, p_n$的连续函数

106.3.3 对称性

对于$p_1, \cdots, p_n$任意两项互换, 信息量不变. 从公式可以看出

106.3.4 扩展性

即

$$\lim_{\varepsilon \to 0} H_{n+1}(p_1, \cdots, p_n - \varepsilon, \varepsilon) = H(p_1, \cdots, p_n)$$

将信息量按定义展开后取极限即得

106.3.5 可加性

对于信源

$$X : \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} & x_{21} & \cdots & x_{2n} & \cdots & \cdots & x_{m1} & \cdots & x_{mn} \\ p_{11} & p_{12} & \cdots & p_{1n} & p_{21} & \cdots & p_{2n} & \cdots & \cdots & p_{m1} & \cdots & p_{mn} \end{bmatrix}$$

假设

$$p_k = \sum_{i=1}^{n} p_{ki}, \quad k = 1, \cdots, m$$

满足信源条件

$$\sum_{k=1}^{m} p_k = 1$$

可加性指下面的式子成立

$$\begin{aligned} H(X) &= H_{mn}(p_{11}, p_{12}, \cdots, p_{mn}) \\ &= H_m(p_1, p_2, \cdots, p_m) + \sum_{k=1}^{m} p_k H_n(\frac{p_{k1}}{p_k}, \frac{p_{k2}}{p_k}, \cdots, \frac{p_{kn}}{p_k}) \end{aligned}$$

只要我们将求和公式展开, 然后做简单的合并就可以得到证明

对于3个值的例子来说,当$p+q+r=1$

$$H(p,q,r) = H(p,q+r) + (q+r)\dot{H}(\frac{q}{q+r}, \frac{r}{q+r}) = H(p,q+r) + (q+r)\dot{H}(q,r)$$

106.3.6 极值性

即不等式

$$H(p_1, \cdots, p_n) <= H_n(\frac{1}{n}, \frac{1}{n}, \cdots, \frac{1}{n}) = log(n)$$

证明需要引用Jessen不等式, 略

	草本	灌木	乔木
低山阔叶林区	120	80	40
高山草甸区	36	9	0

Table 106.1: 两个地区的植被构成

106.3.7 例子: 植被调查

统计两个地区的植被构成

entropy()函数默认计算 shannon 信息熵

```
# 低山阔叶林区的植被数据
> x=c(rep(1,120),rep(2,80),rep(3,40))
> table(x)
x
  1   2   3
120  80  40
> freqs(table(x))
x
       1        2        3
0.5000000 0.3333333 0.1666667
> entropy(table(x))
[1] 1.011404
> entropy(c(120,80,40))
[1] 1.011404

# 手工计算, x为频率, 原始数据请使用freqs(table())转换为频率
my.entropy<-function(x){
   y<-apply(x,1,function(x) x*log(x))
   res<- -sum(y)
   res}
> my.entropy(freqs(table(x)))
[1] 1.011404
```

106.4 相对信息量和信源剩余度

106.4.1 定义

由于信息量的极值性, 信息符号概率相同的信源具有最大的信息量$\log n$, 可以定义相对率

$$\eta = \frac{H_n(X)}{\log n}$$

与相对率对应的概念是信源剩余度

$$r = 1 - \eta$$

当信源每个符号出现概率相等, 信息量最大, 相对率为1, 剩余度为0. 当信源仅有一个符号时信息量为0, 相对率为0, 剩余度为1.

106.4.2 例子

例如我们可以对不同的蛋白质根据氨基酸的组成比例计算信息量和相对率, 作为其营养价值的评估.

也有生物学家把相对率作为蛋白质活性的指标, 提出相对率指标, 只有相对率大于一定指标的蛋白质才具有活性.

106.5 互信息(mutual information)

106.5.1 例子: 中国豆科植物花冠类型与植株类型

我们用一个例子说明. 下面是中国豆科植物花冠类型x与植株类型y的统计数据. 其中花冠类型为x1为辐射对称, x2为左右

对称复瓦状花, x3为蝶形花. y1为草本, y2为灌木, y3为乔木或木质藤本.

```
# 两个信源的统计数据
s=matrix(c(0.0025,0.0076,0.3839,
           0.0101,0.0202,0.3409,
           0.0707,0.0530,0.1111),
         nc=3,
        dimnames=list(c("辐射对称花","左右对称花","蝶形花"),
                      c("草本","灌木","乔木")))
> s
           草本   灌木   乔木
辐射对称花 0.0025 0.0101 0.0707
左右对称花 0.0076 0.0202 0.0530
蝶形花     0.3839 0.3409 0.1111

> sum(s) # 总概率为1
[1] 1
```

它们分别构成了两个信源, 植株类型信源和花冠类型信源. 下面是信源符号出现的概率

```
# s.y
> colSums(s) # 植株类型出现的概率
   y1    y2    y3
0.3940 0.3712 0.2348
# s.x
> rowSums(s) # 花冠类型出现的概率
   x1    x2    x3
0.0833 0.0808 0.8359
```

上述两个信源的信息量为

```
# H(x): 花冠类型信源的信息量
> h.x=entropy(rowSums(s)); h.x
[1] 0.5601329
```

```
# H(y): 植株类型的信息量
> h.y=entropy(colSums(s)); h.y
[1] 1.075068
```

106.5.2 联合信息量

联合信息量(s的信息量), 即

$$H(xy) = H(p_{11}, \cdots, p_{33})$$

计算得

```
#
> h.s=entropy(s);h.s
[1] 1.498812
```

106.5.3 条件信息量

先写出公式. X在Y下的条件信息量为

$$\begin{aligned} H(X|Y) &= \sum_{y\in Y} p(y)H(X|y) \\ &= \sum_{y\in Y}\sum_{x\in X} p(x,y)\log\frac{p(y)}{p(x,y)} \end{aligned}$$

类似, Y在X下的条件信息量为

$$\begin{aligned} H(Y|X) &= \sum_{x\in X} p(x)H(Y|x) \\ &= \sum_{x\in X}\sum_{y\in Y} p(x,y)\log\frac{p(x)}{p(x,y)} \end{aligned}$$

我们发现

$$H(X) \neq H(X|Y) \neq H(Y|X)$$

进一步想到$H(X) - H(X|Y)$是否可以作为Y对X关联性的度量呢?

利用信息量的可加性有

$$H(XY) = H(X) + H(Y|X)$$
$$H(XY) = H(Y) + H(X|Y)$$

两式相减有

$$H(X) - H(X|Y) = H(Y) - H(Y|X) = H(X) + H(Y) - H(XY) = I(X,Y) = I(Y,X)$$

$I(X,Y), I(Y,X)$为平均互信息, 可以解释为X,Y可以互相解释的程度.

下面是例子.

先考虑花冠类型x, 按照植株类型又分为3个部分, 如果除以$p(y1)$, 则其和变为1, 构成随机向量

$$\sum_{k=1}^{3} \frac{p_{k1}}{p(y1)} = 1$$

我们可以对此随机向量求得信息量$H(x/y1)$. y1把x分为一个条件信源, 类似, 3种植株类型把x分为3个条件信源. 其信息量记为$H(x/y1), H(x/y2), H(x/y3)$

```
# x被不同的y划分为3个条件信源
> s.y=apply(s,2,function(x) x/sum(x));s.y
          y1        y2        y3
x1 0.006345178 0.02720905 0.3011073
x2 0.019289340 0.05441810 0.2257240
x3 0.974365482 0.91837284 0.4731687
> colSums(s.y)
y1 y2 y3
 1  1  1
```

```
# 求3个列的信息量: H(x/y1),H(x/y2),H(x/y3)
> h.yy=apply(s.y,2,entropy);h.yy
      y1       y2       y3
0.1335683 0.3346824 1.0514665

# 实际上不做变换的结果是一样的
> h.yy=apply(s,2,entropy);h.yy
      y1       y2       y3
0.1335683 0.3346824 1.0514665
```

这样我们取得了从不同植株类型的条件概率来衡量花冠类型的信息量, 为此, 将3个信息量相加. 当然更加合理的做法是分别乘以其出现的概率$p(y_i)$, 结果为

```
# H(x/y): x被y划分后的总加权信息量
> h.xy=sum(h.yy*colSums(s));h.xy
[1] 0.4237444
```

类似的, y也可以被x划分为3个不同的条件信源, 求得信息量分别是

```
> h.xx=apply(s,1,entropy);h.xx
      x1       x2       x3
0.5002464 0.8455143 0.9913762
# H(y/x): y被x划分后的加权的总信息量
> h.yx=sum(h.xx*rowSums(s));h.yx
[1] 0.9386795
```

106.5.4 关联性

我们发现x的信息量0.5601, 按照y划分后信息量为0.4237. $H(x) > H(x/y)$. 多出的部分是由于花冠被植株类型划分而出现的, 说明植株类型对花冠类型的信息量产生了影响.

```
# 差值
> h.x-h.xy
[1] 0.1363885

# 实际上这就是互信息. 后面会看到
> mi.plugin(s)
[1] 0.1363885
```

如果没有影响, 假设x被y划分后线性相关, 可以看到其信息量保持不变.

```
> tt=rowSums(s)
> s1=cbind(tt*0.2,tt*0.3,tt*0.5); s1
     [,1]    [,2]    [,3]
x1 0.01666 0.02499 0.04165
x2 0.01616 0.02424 0.04040
x3 0.16718 0.25077 0.41795

> entropy(rowSums(s)) # H(x)
[1] 0.5601329

> sum(apply(s1,2,entropy)*rowSums(s1)) # H(x/y)
[1] 0.5601329
```

那么$H(x) - H(x/y)$应该可以作为y对x关联性的一种度量.

106.5.5 平均互信息及其性质

定义X,Y的平均互信息为

$$\begin{aligned} I(X,Y) &= H(X) - H(X/Y) \\ I(Y,X) &= H(Y) - H(Y/X) \\ I(X,Y) &= I(Y,X) = H(X) + H(Y) - H(XY) \end{aligned}$$

(下面参考 http://en.wikipedia.org/wiki/Mutual_information)

离散公式为

$$I(X;Y) = \sum_{y\in Y}\sum_{x\in X} p(x,y)\log\left(\frac{p(x,y)}{p_1(x)\,p_2(y)}\right)$$

连续变量

$$I(X;Y) = \int_Y\int_X p(x,y)\log(\frac{p(x,y)}{p_1(x),p_2(y)})dxdy$$

下面是平均互信息的性质

- 对称性
- 非负
- 若X,Y独立, 则
  - $I(X,Y) = 0$
  - $H(XY) = H(X) + H(Y)$
  - $H(X) = H(X/Y), \quad H(Y) = H(Y,X)$

实际上, x,y独立时, $p(x,y) = p(x)p(y)$, 因此

$$\log\left(\frac{p(x,y)}{p(x)\,p(y)}\right) = \log 1 = 0$$

互信息与信息量(熵)有以下关系式

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X,Y) \\ &= H(X,Y) - H(X|Y) - H(Y|X) \end{aligned}$$

$H(X), H(Y)$为边际信息量(熵), $H(X|Y), H(Y|X)$为条件信息量. $H(X,Y)$为联合信息量.

平均互信息给出了多种状态多种因素之间复杂的关系的度量. 这些因素可以是数值的, 也可以是非数值的. 因此更加能够满足生物学的需要.

如果$H(X)$给出了X的不确定程度, 那么$H(X|Y)$就是X能够给出的但是Y给不出的信息量, 即已知Y后X增加的信息量.

106.5.6 信息增量

(参考 http://en.wikipedia.org/wiki/Mutual_information)

类似后面的离散增量, 由互信息与信息量(熵)的关系式, 定义X,Y的信息增量为

$$\begin{aligned} d(X,Y) &= H(X,Y)`I(X;Y) = H(X|Y) + H(Y|X) \\ D(X,Y) &= d(X,Y)/H(X,Y) <= 1 \end{aligned}$$

满足三角关系式, 非负, 对称. 此信息增量可以作为信息系数使用, 度量两个变量的信息距离.

106.5.7 函数计算

下面计算中国豆科植物花冠类型x与植株类型y的统计数据的互信息

```
# 为方便再次写出数据
s=matrix(c(0.0025,0.0076,0.3839,
         0.0101,0.0202,0.3409,
          0.0707,0.0530,0.1111),
       nc=3,
       dimnames=list(c("辐射对称花","左右对称花","蝶形花"),
                c("草本","灌木","乔木")))
# 建立独立样本s1
tt=rowSums(s)
s1=cbind(tt*0.2,tt*0.3,tt*0.5); s1

# 计算互信息
```

```
> mi.plugin(s)
[1] 0.1363885
# 对称性
> mi.plugin(t(s))
[1] 0.1363885

# 独立则互信息为0
> mi.plugin(t(s1))
[1] 0
```

106.5.8 条件互信息(Conditional mutual information)

(参考 http://en.wikipedia.org/wiki/Mutual_information)

有时候需要计算某条件下的两个变量的互信息. 定义为

$$I(X;Y|Z) = \mathbb{E}_Z\big(I(X;Y)|Z\big) = \sum_{z\in Z}\sum_{y\in Y}\sum_{x\in X} p_Z(z)p_{X,Y|Z}(x,y|z)\log\frac{p_{X,Y|Z}(x,y|z)}{p_{X|Z}(x|z)p_{Y|Z}(y|z)},$$

可以简化为

$$I(X;Y|Z) = \sum_{z\in Z}\sum_{y\in Y}\sum_{x\in X} p_{X,Y,Z}(x,y,z)\log\frac{p_Z(z)p_{X,Y,Z}(x,y,z)}{p_{X,Z}(x,z)p_{Y,Z}(y,z)}.$$

进一步, 可以写为

$$I(X;Y|Z) = H(X,Z) + H(Y,Z)`H(X,Y,Z)`H(Z)$$

下面我们来推导一下

$$\begin{aligned} I(X;Y|Z) &= \sum_{z\in Z}\sum_{y\in Y}\sum_{x\in X} p_Z(z)p_{X,Y|Z}(x,y|z)\log\frac{p_{X,Y|Z}(x,y|z)}{p_{X|Z}(x|z)p_{Y|Z}(y|z)} \\ &= \sum_{z\in Z}\sum_{y\in Y}\sum_{x\in X} p(x,y,z)[\log p(x,y|z) - \log p(x|z) - \log p(y|z)] \end{aligned}$$

其中第二项, 根据条件概率公式$p(x|z)=\frac{p(x,z)}{p(z)}$和条件信息量的公式

$$-\sum_{z\in Z}\sum_{y\in Y}\sum_{x\in X}p(x,y,z)\log p(x|z)=-\sum_{z\in Z}\sum_{x\in X}p(x,z)\log\frac{p(x,z)}{p(z)}=H(X|Z)=H(XZ)-H(Z)$$

同理, 第三项

$$-\sum_{z\in Z}\sum_{y\in Y}\sum_{x\in X}p(x,y,z)\log p(y|z)=-\sum_{z\in Z}\sum_{y\in Y}p(y,z)\log\frac{p(y,z)}{p(z)}=H(Y|Z)=H(YZ)-H(Z)$$

第一项

$$\sum_{z\in Z}\sum_{y\in Y}\sum_{x\in X}p(x,y,z)\log p(x,y|z)=\sum_{z\in Z}\sum_{y\in Y}\sum_{x\in X}p(x,y,z)\log\frac{p(x,y,z)}{p(z)}=-H(XY|Z)=H(Z)-H$$

综合起来得到

$$\begin{aligned}I(X;Y|Z) &= H(Z)-H(XYZ)+H(XZ)-H(Z)+H(YZ)-H(Z)\\ &= H(X,Z)+H(Y,Z)`H(X,Y,Z)`H(Z)\end{aligned}$$

条件Z下的互信息量可能增加, 也可能减小. 但是对于离散随机变量X,Y,Z下式总是正确的

$$I(X;Y|Z)\geq 0$$

此结果是证明其它信息理论不等式的基础.

其链式递推规则为.

$$I(X_1,X_2,\cdots,X_n;Y)=\sum_{i=1}^{n}I(X_i;Y|X_{i-1},X_{i-2},\cdots,X_1)$$

证 明 略 (见 T. M. Cover and J. A. Thomas. Elements of Information Theory. John Wiley and Sons, Inc., 1991.)

例如

$$\begin{aligned}I(X_1,X_2;Y) &= I(X_1;Y)+I(X_2;Y|X_1)\\ I(X_1,X_2,X_3;Y) &= I(X_1;Y)+I(X_2;Y|X_1)+I(X_3;Y|X_2,X_1)\\ &= I(X_1,X_2;Y)+I(X_3;Y|X_2,X_1)\end{aligned}$$

106.5.9 多元互信息(Multivariate mutual information)

已经提出了多个多元互信息的推广. 例如total correlation 和 interaction information. 后者也称为 co-information 理论.

如果 shannon 熵可以看作符号测度的信息度量的方法, 那么唯一有意义的推广到多变量互信息的定义为

$$I(X1) = H(X1)$$

对于$n > 1$

$$I(X_1; \ldots; X_n) = I(X_1; \ldots; X_{n-1}) - I(X_1; \ldots; X_{n-1}|X_n),$$

其中

$$I(X_1; \ldots; X_{n-1}|X_n) = \mathbb{E}_{X_n}\big(I(X_1; \ldots; X_{n-1})|X_n\big).$$

此定义与交互信息(interaction information, or co-information) 理论的唯一不同在于变量个数为奇数时 co-information 为负, 而此为正.

106.5.10 TODO: co-information

基于集合理论的运算来计算其信息量的理论. 有些难以解释的现象.

106.6 离散信道矩阵

106.6.1 定义

考虑DNA复制, 假设被复制的信源(输入)为$X = T, C, A, G$,复制出的新DNA的信宿(输出)为$Y = T, C, A, G$. 信源信宿的对应为

```
T-->A
C-->G
A-->T
G-->C
```

其信息传递的简单信道为P. 应该使得$a * P = b$, 那么

```
P=matrix(c(0,0,1,0,0,0,0,1,1,0,0,0,0,1,0,0),nc=4,
       dimnames=list(c("T","C","A","G"),c("T","C","A","G")))
```

```
> P
  T C A G
T 0 0 1 0
C 0 0 0 1
A 1 0 0 0
G 0 1 0 0
```

那么有

$$XP = Y$$

DNA转录为RNA的信道是一样的.

另外RNA翻译为蛋白质的信道见参考文献 [11] p275

106.6.2 讨论

可以看到, 如果信道是确定的, 即没有噪声, 那么由X,P就决定Y, 回忆互信息的定义, 实际上X,P就决定了互信息的大小.

问题1: 以信道容量来分析遗传信息的传递, 在信道固定的情况下, 问何种信源的分布(核苷酸的分布)可以使得遗传信息的传递达到最佳状态?

问题2: 信源确定的情况下, 问何种信道可以最大限度的利用信源和能量?

问题3: 假设信道有噪声(随机变量), 那么以上问题如何回答?

问题4: 假设信道在不同时间或不同地点是变化的, 如何分析?

106.7 离散量

106.7.1 描述

与shannon信息量平行的一个概念是对离散性(diversity, 也称为多样性)的度量, 称为离散量(measure of diversity).

很多重要的生物学概念和应用都基于离散量, 最突出的是生物地理学和生物信息分类.

- 生物地理学中物种的地理分布就是典型的离散源, 可以直接引入离散量的概念并分析.
- 数量分类学中的信息分类一直使用离散量作为其基础.
- 分支系统研究中, 离散量和俭约性原理称为重构生物演化关系的依据.
- 生物群落结构, 生物多样性指标和生物关联性分析中都需要引入离散量的概念

106.7.2 定义

回到前面植被调查的例子. 低山阔叶林区的植物构成为草本120, 灌木80, 乔木40, 物种总数为$N = 120 + 80 + 40 = 240$. 称低山阔叶林区的植物构成了下面的离散源

$$X : [120, 80, 40]$$

每种植物类型的信息量为

$$-\log\frac{120}{240} \quad -\log\frac{80}{240} \quad -\log\frac{40}{240}$$

所有植物信息量的总和称为离散量, 记为$D(X)$

$$\begin{aligned} D(X) &= -120\log\frac{120}{240} - 80\log\frac{80}{240} - 40\log\frac{40}{240} \\ &= 240\log 240 - 120\log 120 - 80\log 80 - 40\log 40 \end{aligned}$$

其单位与信息量完全一样, e为底就是nat.

推广到一般情况, 假设s个状态的状态空间, 第i个状态出现的个数为n_i, 离散源为

$$X : [n_1, \cdots, n_s], \quad N = \sum_{i=1}^{s} n_i$$

那么所有符号不确定性度量的总和, 即离散量为

$$D(X) = N\log(N) - \sum_{i=1}^{s} n_i \log n_i$$

106.7.3 离散量函数

下面是计算离散量的函数

```
my.div<-function(x){
   n=sum(x)
   d=n*log(n)-sum(log(x)*x)
   d
}
```

106.7.4 性质

- 非负性
- 对称性, 即变换顺序后离散量不变

- 扩 展 性, $D(n_1, \cdots, n_s) = D(n-1, \cdots, n_s, 0)$
- 可 加 性,

$$\begin{aligned} & D(n_{11}, \cdots, n_{1s},\ n_{21}, \cdots, n_{2s},\ \cdots,\ n_{r1}, \cdots, n_{rs}) \\ = \ & D(m_1, \cdots, m_r) + \sum_{i=1}^{r} D(n_{i1}, \cdots, n_{is}) \end{aligned}$$

其 中

$$m_i = \sum_{i=1}^{k} n_{ik} \quad i = 1, \cdots, s, \quad N = \sum_{i=1}^{r} m_i$$

- 极 值 性, 若$n * s = N$, 那 么n_i相 等 时 离 散 量 最 大, 为

$$D(n_1, \cdots n_s) <= D(n, \cdots, n) = ns \log ns - sn \log n = sn \log s$$

- 等 倍 增 性, 即

$$D(kX) = kD(X), \quad k >= 0$$

106.7.5 其 它 定 义

1958年, D. R. Margalef 提 出 使 用 Brillouin 函 数 表 示 离 散 量, 称 为 Brillouin 离 散 量 定 义

$$D(X) = \log_b \frac{N!}{n_1! n_2! \cdots n_s!}$$

Laxton(1978)在5个 条 件 下, 证 明 了 上 式 对 离 散 量 表 示 的 唯 一 性. 因 为 阶 乘 的 计 算 很 容 易 导 致 溢 出, 计 算 量 也 比 较 大. 由 于 计 算 的 原 因, 此 公 式 实 际 中 应 用 并 不 普 遍

106.7.6 与信息量的区别

信息量是对符号不确定性的度量, 也是对状态不确定性或紊乱性的一种描述.

离散量是对整体不确定性多少的度量, 也是离散多少的度量.

信息量大, 不确定性大, 但是离散量不一定多. 离散量多不一定信息量大.

106.8 两个数据的离散增量

106.8.1 离散增量的定义

定义两个离散源X,Y的和为

$$X+Y:[n_1+m_1,n_2+m_2,\cdots,n_s+m_s]$$

那么和的离散量总是大于离散量的和, 即

$$D(X+Y)>=D(X)+D(Y)$$

从此引入一个十分重要的量, 离散增量(increament of diversity)

$$\triangle(X,Y)=D(X+Y)-D(X)-D(Y)$$

离散增量的另一个数学表达式为

$$\begin{aligned}\triangle(X,Y)=D(M,N)-\sum_{i=1}^{s}D(m_i,n_i)&\\ =\ &(M+N)\log(M+N)-M\log MN\log N\\ &-\sum_{i=1}^{s}[(m_i+n_i)\log(m_i+n_i)-m_i\log m_i-n_i\log n_i]\end{aligned}$$

其中

$$M = \sum_{i=1}^{s} m_i \qquad N = \sum_{i=1}^{s} n_i$$

利用此表达式可以估计离散增量的最大值. 当M,N一定时, 其第二项$D(m_i, n_i) = 0$, 离散增量为最大. 而当$Y = kX$时取0.

因此离散增量的取值范围为

$$0 <= \triangle <= D(M, N)$$

106.8.2 离散增量函数

下面是计算离散增量的函数

```
# 计算离散量的函数
my.div<-function(x){
   n=sum(x)
   d=n*log(n)-sum(log(x)*x)
   d
}

# 离散增量的函数
my.incdiv<-function(x,y){
   z=x+y
   dz=my.div(z)
   dx=my.div(x)
   dy=my.div(y)
   d=dz-dx-dy
   d
}
```

106.8.3 性质

离散增量满足相似性关系的两个基本条件

- 非负性, $\triangle(X,Y) >= 0$
- 对称性, $\triangle(X,Y) = \triangle(Y,X)$

106.9 基于离散量的信息系数

106.9.1 信息系数

至此, 除距离系数, 相关系数, 联合系数外, 又获得一种基于信息理论的系数, 称为信息系数(information coefficient). 我们可以定义各种信息系数

1. 相亲有限系数, $I_1 = \sum_{i=1}^{s} D(m_i, n_i)/D(M,N), \in [0,1]$
2. 相亲无限系数, $I_2 = \sum_{i=1}^{s} D(m_i, n_i)/[(D(M,N) - \sum_{i=1}^{s} D(m_i, n_i)] = \sum_{i=1}^{s} D(m_i, n_i)/\triangle(X,Y) \in [0,\infty]$
3. 相亲相关系数, $I_3 = 2\sum_{i=1}^{s} D(m_i, n_i)/D(M,N) - 1, \in [-1,1]$
4. 相异有限系数, $I_4 = 1 - \sum_{i=1}^{s} D(m_i, n_i)/D(M,N), \in [0,1]$
5. 相异无限系数, $I_5 = [(D(M,N) - \sum_{i=1}^{s} D(m_i, n_i)]/\sum_{i=1}^{s} D(m_i, n_i), \in [0,\infty]$
6. 相异相关系数, $I_6 = 1 - 2\sum_{i=1}^{s} D(m_i, n_i)/D(M,N), \in [-1,1]$

类型4为经常使用的系数, 也可以表示为

$$I(X,Y)_4 = \frac{\triangle}{D(M,N)}$$

106.9.2 使用信息系数进行分类

我们可以使用信息系数来分类, 就好像使用相似系数一样.

106.10 离散增量的推广

106.10.1 两种方法计算离散增量

前面讨论的是两个离散源的离散增量, 我们把它推广到多个.

下面是中国豆科植物花冠类型x与植株类型y的统计数据,

```
S=matrix(c(1,3,152,4,8,135,28,21,44),nc=3,
      dimnames=list(c("辐射对称花","左右对称花","蝶形花"),
                  c("草本","灌木","乔木")))
```

```
> S
         草本 灌木 乔木
辐射对称花    1    4   28
左右对称花    3    8   21
蝶形花      152  135   44
```

现在我们试图使用离散增量的方法来分析.

从花冠类型角度分析, 其离散源和离散量表示为

$X:[33,32,331]$ $D(X)=221.8505X/y_i:[1,3,152]$ $D(X/y_1)=20.85186X/y_2:[4,8,135]$

计算过程为

```
> my.div(S[,1]) # D(X/y_1)
[1] 20.85186
```

```
> my.div(S[,2]) # D(X/y_2)
[1] 49.20079
> my.div(S[,3]) # D(X/y_3)
[1] 97.79071
> my.div(rowSums(S)) # D(X)
[1] 221.8505
```

考虑离散增量

$$D(X) - D(X/y_1) - D(X/y_2) - D(X/y_3) = 54.00719$$

然后, 我们从植株类型考虑离散增量

$Y : [156, 147, 93]$ $D(Y) = 425.73715$ $Y/x_1 : [1, 4, 28]$ $D(Y/x_1) = 16.53785$ $Y/x_2 : [3, 8, 21]$

离散增量为

$$D(Y) - D(Y/x_1) - D(Y/x_2) - D(Y/x_3) = 54.00719$$

106.10.2 统一的公式

注意到, 两种方法的离散增量完全相等. 我们得到同一的公式

$$\triangle(X, Y) = D(X) - D(X/Y)$$
$$\triangle(Y, X) = D(Y) - D(Y/X)$$

两种方法计算离散增量

```
# 离散增量(X,Y)
> my.div(rowSums(S))-sum(apply(S,2,my.div))
[1] 54.00719

# 离散增量(Y,X)
> my.div(colSums(S))-sum(apply(S,1,my.div))
[1] 54.00719
```

还有一种方法计算离散增量, 考虑前面离散量的性质中的可加性, 有

$$\triangle(X,Y) = \triangle(Y,X) = D(X) + D(Y) - D(XY)$$

下面是计算

```
# D(X)+D(Y)-D(XY)
> my.div(colSums(S))+my.div(rowSums(S))-my.div(S)
[1] 54.00719
```

106.10.3 性质

- 对称性, $\triangle(X,Y) = \triangle(Y,X)$
- 非负性,
- 若X,Y独立, 则

$$\begin{aligned} \triangle(X,Y) &= 0, \\ D(XY) &= D(X) + D(Y), \\ D(X) = D(X/Y), D(Y) &= D(Y/X) \end{aligned}$$

106.10.4 离散量系数

对应信息系数, 将相应的信息量换成离散量, 就得到不同的离散量系数

Part XV

杂项

Chapter 107

动态规划

主要参考如下资料：

http://baike.baidu.com/view/28146.htm
http://en.wikipedia.org/wiki/Dynamic_programming
http://zh.wikipedia.org/wiki/动态规划
http://student.csdn.net/space.php?uid=123638&do=blog&id=13697

107.1 概述

动态规划(dynamic programming)是运筹学的一个分支，是求解决策过程(decision process)最优化的数学方法。20世纪50年代初美国数学家R.E.Bellman等人在研究多阶段决策过程(multistep decision process)的优化问题时，提出了著名的最优化原理(principle of optimality)，把多阶段过程转化为一系列单阶段问题，利用各阶段之间的关系，逐个求解，创立了解决这类过程优化问题的新方法——动态规划。1957年出版了他的名著Dynamic Programming，这是该领域的第一本著作。

动态规划问世以来，在经济管理、生产调度、工程技术和最优控制等方面得到了广泛的应用。例如最短路线、库存管理、资源分配、设备更新、排序、装载等问题，用动态规划方法比用其它方法求解更为方便。

虽然动态规划主要用于求解以时间划分阶段的动态过程的优化问题，但是一些与时间无关的静态规划(如线性规划、非线性规划)，只要人为地引进时间因素，把它视为多阶段决策过程，也可以用动态规划方法方便地求解。

动态规划程序设计是对解最优化问题的一种途径、一种方法，而不是一种特殊算法。不象前面所述的那些搜索或数值计算那样，具有一个标准的数学表达式和明确清晰的解题方法。动态规划程序设计往往是针对一种最优化问题，由于各种问题的性质不同，确定最优解的条件也互不相同，因而动态规划的设计方法对不同的问题，有各具特色的解题方法，而不存在一种万能的动态规划算法，可以解决各类最优化问题。因此读者在学习时，除了要对基本概念和方法正确理解外，必须具体问题具体分析处理，以丰富的想象力去建立模型，用创造性的技巧去求解。我们也可以通过对若干有代表性的问题的动态规划算法进行分析、讨论，逐渐学会并掌握这一设计方法。

动态规划只能应用于有最优子结构的问题。最优子结构的意思是局部最优解能决定全局最优解(对有些问题这个要求并不能完全满足，故有时需要引入一定的近似)。简单地说，问题能够分解成子问题来解决。

107.2 步骤

1. 最优子结构性质。如果问题的最优解所包含的子问题的解也是最优的，我们就称该问题具有最优子结构性质（即满足最优化原理）。最优子结构性质为动态规划算法解决问题提供了重要线索。

2. 子问题重叠性质。子问题重叠性质是指在用递归算法自顶向下对问题进行求解时，每次产生的子问题并不总是新问题，有些子问题会被重复计算多次。动态规划算法正是利用了这种子问题的重叠性质，对每一个子问题只计算一次，然后将其计算结果保存在一个表格中，当再次需要计算已经计算过的子问题时，只是在表格中简单地查看一下结果，从而获得较高的效率。

107.3 例1-斐波那契数列

斐波那契数列(Fibonacci series)定义为$F_1 = F_2 = 1,\ F_i = F_{i-1} + F_{i-2},\ i > 2.$

例如我们需要得到F_5，有两种方法计算此值。

107.3.1 Top-down approach

这种方法可以应用到动态规划的思想。

示Top-down 方法

```
F5--->F4--->F3--->F2
  |  |    |
  |  | ->F1
  |   ->F2
   ->F3--->F2
     |
      ->F1
```

受过训练的程序员会想到这个答案。将F_5逐级分解，逐个求得。从上面的示意图看出，在此过程中，重复计算了F3。下面是此递归算法的R程序。注意当n增大的时候重复计算的量是指数增长的。

```
# 计算斐波那契数Fn（do not run!! it's very slow for big n）
fib<-function(n){
   if( n == 1|n==2) {return( 1)}
return(fib(n-1)+fib(n-2))
}

# 计算斐波那契数列1-n
fibs<-function(n){
x=rep(0,n)
for(i in 1:n){
x[i]=fib(i)
```

```
}
x
}

# do not run!! it's very slow for big n
> fibs(10)
 [1]  1  1  2  3  5  8 13 21 34 55
```

我们可以对此做出改进。使用一个容器将已经计算好的结果存储起来，用到的时候查询即可，不需要重新计算。这种思想就是动态规划的思想。下面是使用了动态规划的算法。（注意：由于R没有dict或map的数据结构，所以实现起来语句和效率有一点点问题，但是并不明显）

```
# 先看看伪代码
  var m := map(0 → 0, 1 → 1)
  function fib(n)
     if map m does not contain key n
        m[n] := fib(n  1) + fib(n  2)
     return m[n]

#---------------------------------
# 计算斐波那契数Fn，使用动态规划思想，速度快
m<-vector() # 存储已有结果
m[1:2]<-1
# 递归函数，但是加入了一个判断语句来使用已有结果
fibr<-function(n){
   l=length(m)
   if(n<=l){
    return(m[n])
   }
   else{
m[n]=fibr(n-1)+fibr(n-2)
}
m[n]
}

> fibr(10)
[1] 55
```

107.3.2 Bottom-up approach

这种方法就是普通的方法，依次计算，与动态规划的思想无关，但是实质是一样的。

```
# also fast
fibcommon<-function(n){
if(n>2){
Fibo<-vector()
Fibo[1:2]<-1
for (i in 3:n) Fibo[i]<-Fibo[i-1]+Fibo[i-2]
return(Fibo[n])
}
else{
return(1)
}
}
```

107.4 例2-最短路径问题

下图表示城市之间的交通路网，线段上的数字表示费用，单向通行由A-¿E。求A-¿E的最省费用(最短路径)。一个简单的方法就是计算所有路径的距离，选择最小的一个。所有路径的数目为指数增长。而且许多路径的和重复计算。

很明显，当某阶段的起点给定时，它直接影响着后面各阶段的行进路线和整个路线的长短，而后面各阶段的路线的发展不受这点以前各阶段的影响。故此问题的要求是：在各个阶段选取一个恰当的决策，使由这些决策组成的一个决策序列所决定的一条路线，其总路程最短。例如，从A到D1的最短距离可以由min(A到C1的最短距离+C1D1,A到C2的最短距离+C2D1,A到C3的最短距离+C2D1)得到

我们由A向E推进。

首先计算A-B的最短距离

```
B1=2  B2=5  B3=1
```

下一步，由A-C的最短距离

```
min(A,C1)=min((A,B1,C1),(A,B2,C1),(A,B3,C1))
 =min(2+12,5+6,1+13)
 =11
min(A,C2)=min((A,B1,C2),(A,B2,C2),(A,B3,C2))
 =min(2+14,5+10,1+12)
 =13
min(A,C3)=min((A,B1,C3),(A,B2,C3),(A,B3,C3))
 =min(无路径,5+4,1+11)
 =9
```

下一步，由A到D的最短路径

```
min(A,D1)=min((A,C1,D1),(A,C2,D1),(A,C3,D1))
 =min(11+3,13+6,无路径)
 =14
min(A,D2)=min((A,C1,D2),(A,C2,D2),(A,C3,D2))
 =min(11+9,13+5,无路径)
 =18
```

下一步，由A到E的最短路径

```
min(A,E)=min((A,D1,E),(A,D2,E))
 =min(14+5,18+2)
 =19
```

同时，我们可以获得最短的路径为19：从E回溯，寻找最短路径即可。D1到E为最短路径，C1到D1为最短路径，B3到C1为最短路径，最后的最短路径结果为 E-D1-C1-B3-A。

同理，我们还可以得到A到任意点的最短路径。例如，A到D2的最短路径为18：C2到D2为最短路径，B3到C2为最短路径，最后的最短路径结果为 D2-C2-B3-A。

从上例的求解结果中，我们不仅得到由A点出发到终点E的最短路线及最短距离，而且还得到了从所有各中间点到终点的最短路线及最短距离，这对许多实际问题来讲是很有用的。

动态规划的最优化概念：是在一定条件下，得到一种途径，在对各阶段的效益经过按问题具体性质所确定的运算以后，使得全过程的总效益达到最优。

应用动态规划要注意阶段的划分是关键，必须依据题意分析，寻求合理的划分阶段(子问题)方法。而每个子问题是一个比原问题简单得多的优化问题。而且每个子问题的求解中，均利用它的一个后部子问题的最优化结果，直到最后一个子问题所得最优解，它就是原问题的最优解。

(1)状态必须满足最优化原理；

(2)状态必须满足无后效性。

动态规划的最优化原理：是无论过去的状态和决策如何，对前面的决策所形成的当前状态而言，余下的诸决策必须构成最优策略。可以通俗地理解为子问题的局部最优将导致整个问题的全局最优。在上例中例题1最短路径问题中，A到E的最优路径上的任一点到终点E的路径也必然是该点到终点E的一条最优路径，满足最优化原理。

动态规划的无后效性原则：某阶段的状态一旦确定，则此后过程的演变不再受此前各状态及决策的影响。也就是说，“未来与过去无关”，当前的状态是此前历史的一个完整总结，此前的历史只能通过当前的状态去影响过程未来的演变。具体地说，如果一个问题被划分各个阶段之后，阶段I中的状态只能由阶段I+1中的状态通过状态转移方程得来，与其他状态没有关系，特别是与未发生的状态没有关系，这就是无后效性。

107.5 序列比对

参考《计算分子生物学导论(Introduction to Computational

Molecular Biology)》 [7] chapter 3.2 Page 35-42

三种比对：根据我们关注的是整个序列还是序列子串，导致了所谓的全局(global)序列比对和局部(local)序列比对。还存在第三种比较，不是比较任意子串而是比较给定序列的前缀和后缀，我们称为半全局比对，例如在序列的拼接问题中。所有上述问题均可以使用动态规划(动态程序设计)方法来有效解决。

比对的定义：首先定义两个序列的比对。序列可以长度不同，允许包含空格，但是不能有两个空格的对应。空格可以插入在头或尾部。包含空格使得序列长度相同。因此，我们把比对定义为在序列中任意位置插入空格使得序列长度相同。

比对的打分：给定两个序列的比对，我们可以按照下面的方法赋予其一个打分分值。比对的每列得到一个值，比对的总值是各列的值的和。若一列字符相同(匹配)，值为+1，字符不同(失配)，值为-1，其中一个为空格，值为-2。这些特定的值在实际中经常使用。当然也有很多其他的打分系统。

最佳比对：两个序列不同比对的数目是巨大的。每个比对都有一个分值。拥有最大分值的比对就是最佳比对。

```
GA-CGGATTAG
GATCGGAATAG
```

上面是一个比对，由9个相同字符的列，1个不同字符的列，1个空格列。总的打分为

```
9*1+1*(-1)+1*(-2)=6
```

107.5.1 全局比对

计算两个序列的相似性的一个办法是产生所有的比对，选择最佳的。但是两个序列的可能的比对的数目呈指数增长。可以看到，其中有很多的重复计算，这是可以利用的。

思路是给定两个序列s和t，不是直接确定s和t的整体相似性，而是从最短的前缀开始，利用先前计算的结果求解更大前缀的相似性。

令m为s的长度，n为t的长度。计算在一个$(m+1)*(n+1)$的矩阵M内进行。其第i行j列的值表示$s[1\cdots i]$和$t[1\cdots j]$的相似性。

例如

```
s=AAAC
t=AGC
```

s排列在左边，t排在上面。第一行和第一列初始化为空格罚分，因为若一个序列是完全空格，两个序列之间只有一种可能的比对，即在空序列中插入与另一个序列字符同样多的空格。其打分为$-2k$。k是非空序列的长度。因此，第一行和第一列的填写是很容易的。

观察矩阵内其它的值，我们发现，能够通过矩阵中$M(i-1,j), M(i-1,j-1), M(i,j-1)$这三项来计算$M(i,j)$，因为要得到$s[1\cdots i]$和$t[1\cdots j]$的比对，只有3种可能

- $s[1\cdots i-1]$和$t[1\cdots j]$在$s[i]$处匹配一个空格，对应从$M(i-1,j)$到$M(i,j)$的路径。
- $s[1\cdots i-1]$和$t[1\cdots j-1]$在$s[i]$与$t[j]$处匹配或失配，对应从$M(i-1,j-1)$到$M(i,j)$的路径。
- $s[1\cdots i]$和$t[1\cdots j-1]$在$t[j]$处匹配一个空格，对应从$M(i,j-1)$到$M(i,j)$的路径。

下面是动态规划相似性比对的打分矩阵,行列下标从0开始。

```
A G C
0 1 2 3
-------------
  0 0 -2 -4 -6
```

```
A 1 -2 1 -1 -3
A 2 -4 -1 0 -2
A 3 -6 -3 -2 -1
C 4 -8 -5 -4 -1
```

这样，前缀的相似性可以按照下式计算

```
sim(s[1...i],t[1...j])=max{ sim(s[1...i],t[1...j-1])-2,
sim(s[1...i-1],t[1...j-1])+p(i,j),
sim(s[1...i-1],t[1...j])-2 }
```

其中如果$s[i] = t[j], p(i,j) = +1$，否则$s[i] \neq t[j], p(i,j) = -1$。对应的矩阵的写法为

```
M(i,j)=max{ M(i-1,j)-2,
M(i-1,j-1)+p(i,j),
M(i,j-1)-2 }
```

我们只要按照顺序从左到右或从上到下逐个填写就可以。任何其他次序，只要在计算$M(i,j)$之前得到$M(i-1,j), M(i-1,j-1), M(i,j-1)$即可。

最佳比对：从$M(m,n)$开始，回溯到$M(0$fi$0)$我们就可以得到最佳比对。但是注意，最佳比对常常不止一个，可以存在很多。一种方法是按照逆时针回溯，即t中带空格的列优先于两个字母的列，后者优先于s带空格的列。例如，比对$s = ATAT, t = TATA$时，我们得到

```
-ATAT
TATA-
```

而不是

```
ATAT-
-TATA
```

这种比对有时候称为“最上”比对，因为它使用上方的箭头。

一般优化比对的数目非常大，这时候常常建议保留一部分，例如最上和最下比对。

算法的空间复杂性和时间复杂性均为$O(mn)$

107.5.2 TODO: 局部比对

将基本的算法做一个修改就可以得到局部比对算法。

将矩阵第一行和第一列初始化为0，内部值作为s与t的后缀的比对的最高分。。。。略

107.5.3 TODO: 半局部比对

也可以通过修改基本算法得到

107.5.4 TODO: 基本算法的扩展

节省空间，相似序列的比较。

107.5.5 TODO: 多个序列的比对

多个序列的比对是NP完全问题，使用动态规划方法后算法复杂性为$O(n^k)$，k为序列的条数。所以需要使用启发式方法。

Chapter 108

Bootstrapping介绍

108.1 多少bootstrap样本才够?

文献建议的样本量随计算能力的增加而增加。

增加样本量不能增加原始数据的信息量，仅仅可以减少随机采样的误差，这是bootstrap的效果。

108.2 bootstrap的类型

单变量问题中，通常使用放回式采样（case resampling）。但是，小样本的情况下，使用 parametric bootstrap 可能更合适。有些情况，smooth bootstrap 可能更合适。

108.2.1 case resampling

回归问题中，case resampling指对个体的重采样，一般是以“行”为单位。回归问题的数据通常比较大，这种采样一般是可以接受的。但是常常受到批评。

回归中，解释变量（译者注：自变量）常常是固定的，或至少比反应变量（译者注：因变量）的约束要多。并且解释变量定义了所能够获得的信息。因此，对每个单位重采样意味着每个bootstrap会损失一些信息。因此应该使用其他的bootstrap方法。

108.2.2 smooth bootstrap

每次bootstrap增加一个均值为0的噪声（通常为正态分布）。这等价于对数据的核密度（kernel density）估计进行采样。

108.2.3 parametric bootstrap

在此情况下，将数据进行一个参数模型的拟合（译者注：拟合一个有显式函数的分布，这个显式函数就是参数模型），拟合一般用最大似然法，然后从此模型拟合得到的密度函数随机采样。

一般采样和原数据的样本大小分布一致。这个采样过程象其他bootstrap方法一样重复多次。使用参数模型的采样过程与从同样的模型进行基本的统计学推断是不同的。

108.2.4 resampling residuals

另外一个回归问题的bootstrap方法是对残差重采样（resample residuals）。方法如下

1. 拟合模型得到拟合值$\hat{y}_i$,残差$\hat{\epsilon}_i = y_i - \hat{y}_i,\ (i = 1, \cdots, n)$

2. 对每一对(x_i, y_i)，其中x_i可能是多维变量，将y_i叠加一个随机采样的残差$\hat{\epsilon}_j$。换句话说，做替换$y_i^* = y_i + \hat{\epsilon}_j$，其中j是从$i = 1, \cdots, n$随机选择的。

3. 使用y_i^*重新拟合模型得到感兴趣的参数（译者注：斜

率，均值等）

4. 重复步骤2和3多次

这个方法的优势在于保留了自变量的信息。然而，有一个问题是究竟重采样那个残差？原始残差是一个选择，另一个是（线性回归中的）学生氏残差（studentized residuals）。使用学生氏残差有它的理由，实际中它经常得到较小差异的结果并且容易运行和互相比较。

108.2.5 gaussian process regression bootstrap

当数据相关的时候，直接做bootstrap会破坏其相关性。此方法使用高斯过程回归来拟合概率模型，得到哪个重复可以采样。高斯过程是贝叶斯非参数统计的一种，但是此处用来建立参数bootstrap过程，以便于允许处理时间相关的数据。

108.2.6 wild bootstrap

每个残差随机乘以1或-1.此方法假设残差对称分布，并在数据较少的时候，可能提供比原始残差更多的信息。

108.2.7 choice of statistic - pivoting

对于从数据中获取尽可能多的信息非常重要的时候，要考虑哪个参数或统计量是bootstrap的目标。假设要根据观测数据推断均值，那么有两个选择

1. 产生样本均值的bootstrap采样，来估计均值的置信区间。

2. 产生新的统计量（均值除以样本标准差）的bootstrap采样，构造此值的置信区间。然后导出均值的置信区间（乘以原数据的标准差）

结果会不同，模拟结果显式第二种好一点。此方法可以部分的由标准的正态分布的参数过程导出，但是更加一般化。其想法是利用轴变量（pivotal quantity），或发现导出的统计量逼近轴（pivotal）。

108.3 由bootstrap分布导出置信区间

有几种方法导出置信区间，没有一个在所有情况下有压倒性优势。我们需要在简洁和一般化之间寻求平衡。

108.3.1 置信区间的偏差和缺乏对称

偏差：当我们比较bootstrap的均值和原始数据的均值，要检查偏差（bias）。一旦bootstrap分布对称且无偏差，百分比置信区间是一个好方法。bootstrap的偏差会导致置信区间的偏差。不同的方法试图纠正此偏差。

缺乏对称：bootstrap的分布产生另外一个问题，即应该如何通过置信区间反应分布的非对称。

108.3.2 bootstrap导出置信区间的方法

包括：

1. 百分比bootstrap。这是最简单的方法。使用2.5和97.5百分位点来得到95%的置信区间。这种方法可以应用于任何统计量。当bootstrap的分布对称并且与原数据无偏差的时候此方法效果很好。但是不满足这些条件时，估计结果会过小（over optimistic）（see Schenker 1985)）。

Schenker 指出当样本量小的时候（小于50），方差的百分比置信区间会过小。例如，样本量为20的样本，90%置信区间一般只包含真实置信区间的78%。

2. basic bootstrap：this is a "turned around" version of the percentile bootstrap. 这是百分比bootstrap的另外一种形式。

3. 学生氏bootstrap

4. 偏差修正（bias corrected）bootstrap：对bootstrap的偏差进行修正。

5. 加速bootstrap（The bootstrap bias-corrected accelerated (BCa) (a.k.a: accelarated bootstrap)）：对bootstrap的偏差和偏斜度同时校正，其设置的很多范围内是准确的，计算量比较合理，结果也比较合理。

108.4 例子

108.4.1 中位数检测

由于bootstrap方法不需要原数据服从正态分布，并且样本量比较小的时候也比较有效（小于20），所以使用它检测中位数很流行。

但是，仍然有两种方法很流行（可能并不合适），(1) the logic of Baron and Kenny [3] or (2) the Sobel test.

108.4.2 smoothed bootstrap

使用下面的数据

`http://en.wikipedia.org/wiki/Classic_data_sets`

对每个bootstrap值叠加一个服从$N(0, \sigma^2)$的随机数，其中$\sigma^2 = 1/n$，n为样本量。

可以看到这样bootstrap采样后平滑了不少。

108.4.3 与其他推断方法的关系

1. jackknife procedure: 用于估计样本统计量的偏差和方差。

2. 交叉验证（cross-validation）：使用样本的一部分（子样本）估计的统计量应用于另一个子样本。

108.4.4 TODO: U-统计量

参考

http://en.wikipedia.org/wiki/U-statistic

Chapter 109

z-curve

109.1 解释

DNA 碱基由 ATGC 四种构成, 若某 DNA 序列有 N 个碱基 A_n, C_n, G_n, T_n 分别为从序列开始计算到第n个碱基时的 ATGC 的个数. 实际上, 此序列由 A_n, C_n, G_n, T_n 唯一确定. 下面将之映射到三维空间

$$
\begin{aligned}
x_n &= (A_n + G_n) - (C_n + T_n) \equiv R_n - Y_n \\
y_n &= (A_n + C_n) - (G_n + T_n) \equiv M_n - K_n \\
z_n &= (A_n + T_n) - (C_n + G_n) \equiv W_n - S_n
\end{aligned}
$$

其中, $A_0 = C_0 = G_0 = T_0 = 0$, 因此 $x_0 = y_0 = z_0 = 0$.

可以证明, A_n, C_n, G_n, T_n 与 x_n, y_n, z_n 互相唯一确定.

下面将 z_n 对 $n = 0, 1, 2, \cdots, N$ 做线性回归

$$z \sim kn$$

其中 k 为回归的斜率系数. 令

$$z'_n = z_n - k * n$$

将 $z'_n \sim n$ 作曲线图, 就得到所谓的 z-curve (z 曲线). z 曲线在基因分析中起关键作用.

下面是计算 z 曲线的函数. 输入为 DNA 序列, 输出 x,y,z 为坐标, a,t,g,c 为 A_n, C_n, G_n, T_n 序列. z1 为 z' 值.

```
zcurve<-function(s){
   a=cumsum(s=="a"|s=="A")
   t=cumsum(s=="t"|s=="T")
   g=cumsum(s=="g"|s=="G")
   c=cumsum(s=="c"|s=="C")
   a=append(0,a)
   t=append(0,t)
   g=append(0,g)
   c=append(0,c)
   x=a+g-(c+t)
   y=a+c-(g+t)
   z=a+t-(c+g)
   n=0:(length(z)-1)
   lm.z = lm(z~n-1)
   k=lm.z$coefficients[1]
   z1=z-k*n
   r <-list(x=x,y=y,z=z,z1=z1,a=a,t=t,g=g,c=c)
   r
}

> s='GCTTCTAGCCTGACATATTAACTCCTG'
> s<-strsplit(s,"")
> s=s[[1]]
> s
 [1] "G" "C" "T" "T" "C" "T" "A" "G" "C" "C" "T" "G" "A" "C" "A" "T" "A" "T" "T"
[20] "A" "A" "C" "T" "C" "C" "T" "G"
> r<-zcurve(s); r
$x
 [1]  0  1  0 -1 -2 -3 -4 -3 -2 -3 -4 -5 -4 -3 -4 -3 -4 -3 -4 -5 -4 -3 -4 -5 -6
[26] -7 -8 -7

$y
 [1]  0 -1  0 -1 -2 -1 -2 -1 -2 -1  0 -1 -2 -1  0  1  0  1  0 -1  0  1  2  1  2
[26]  3  2  1
```

```
$z
 [1]  0 -1 -2 -1  0 -1  0  1  0 -1 -2 -1 -2 -1 -2 -1  0  1  2  3  4  5  4  5  4
[26]  3  4  3

$z1
 [1]  0.0000000 -1.1050505 -2.2101010 -1.3151515 -0.4202020 -1.5252525
 [7] -0.6303030  0.2646465 -0.8404040 -1.9454545 -3.0505051 -2.1555556
[13] -3.2606061 -2.3656566 -3.4707071 -2.5757576 -1.6808081 -0.7858586
[19]  0.1090909  1.0040404  1.8989899  2.7939394  1.6888889  2.5838384
[25]  1.4787879  0.3737374  1.2686869  0.1636364

$a
 [1] 0 0 0 0 0 0 0 1 1 1 1 1 1 2 2 3 3 4 4 4 5 6 6 6 6 6 6 6

$t
 [1] 0 0 0 1 2 2 3 3 3 3 3 4 4 4 4 4 5 5 6 7 7 7 7 8 8 8 9 9

$g
 [1] 0 1 1 1 1 1 1 1 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4

$c
 [1] 0 0 1 1 1 2 2 2 2 3 4 4 4 4 5 5 5 5 5 5 5 6 6 7 8 8 8

# 查 看 z 曲 线
> n=0:(length(r$z1)-1)
> plot(r$z1~n,col='red',type='l')
```

Chapter 110

wu-kabat 多样性

比对之后的多个长度相同的序列, 其 wu-kabat 多样性(variability)为计算每个位点的不同氨基酸个数除以此位点出现最多的氨基酸的个数, 例如, 第一个位点的不同的氨基酸种类有10种, 出现最多的氨基酸的次数为5次, 则此位点的 wu-kabat 多样性指标为 $10/5 = 2$. 将所有位点的此指标作为向量绘制条形图, 就得到 wu-kabat plot.

http://imed.med.ucm.es/PVS/ 网站提供绘制服务. 绘图后, 有连接显示其值.

```
# 导入 read.fasta 函数
library(seqinr)
wukabat.single<-function(s){
   f<-factor(s)
   t<-table(f)
   a<-length(levels(f)) # 不同的氨基酸种类数
   b<-max(t) # 出现最多的氨基酸的个数
   res<-a/b
   res
}

my.wukabat<-function(file){
   x=read.fasta(file,seqtype="AA",set.attributes = FALSE)
   n<-length(x)
   m<-length(x[[1]])
```

```
    res<-c()
    for(i in 1:m){
        tmp<-c()
        for(j in 1:n){
            tmp<-append(tmp,x[[j]][i])
        }
        # cat(tmp,"\n")
        r<-wukabat.single(tmp)
        res<-append(res,r)
    }
    res
}

# 计 算 wukabat 系 数, 此 处 把 你 的 fasta 文 件 路 径 替 换 下 面
的 /home/xjx/tmp.fasta
w<-my.wukabat("/home/xjx/tmp.fasta")
# 网 站 把 此 系 数 归 一 化, 即 都 除 以 最 小 值, 最 小 值 变 为
为 1, (http://imed.med.ucm.es/PVS/)
w<-w/min(w)
# 绘 制 条 形 图, wukabat plot
barplot(w)
```

Part XVI

附录A-概率统计基础理论

“概率统计基础”是理论部分，不涉及R的使用。实际上是我的生物统计学的部分讲课大纲，仅供参考。参考文献除了[14]，还参考了《初等概率论附随机过程》[4]和复旦大学的《概率论第一册概率论基础》等。

这部分只写到估计和假设检验，其它部分的理论随各个命题一起讨论。

Chapter 111

条件概率与统计独立性

111.1 条件概率

111.1.1 定义

我们有时会碰到下面的情况, 第二种情况就是条件概率

- 求A事件发生的概率
- 知道B已经发生, 求A事件发生的概率

条件概率设(Ω, F, P)是一个概率空间, $B \in F$, 且$P(B) > 0$, 则对任意$A \in F$, 记

$$P(A|B) = \frac{P(AB)}{P(B)}$$

并称$P(A|B)$为在事件B发生的条件下事件A发生的条件概率

考虑两个孩子的家庭. 事件A: 随机选取的家庭有一个男孩和一个女孩这一事件, 则$P(A) = 1/2$, 若我们事先知道这个家庭至少有一个女孩, 那么上述事件的概率是多少?

甲乙两市都位于长江下游. 根据100多年的记录, 知道一年中雨天的比例甲市占20%, 乙市占18%, 两地同时下雨占12%。求甲市下雨时乙市下雨的概率和乙市下雨时甲市下雨的概率.

111.1.2 性质

三个基本性质

条件概率具有概率的三个基本性质

- $P(A|B) \geq 0$ 非负性
- $P(\Omega|B) = 1$ 规范性
- $P(\sum_{i=1}^{\infty} A_i|B) = \sum_{i=1}^{\infty} P(A_i|B)$ 完全可加性

导出性质

- $P(\Phi|B) = 0$
- $P(A|B) = 1 - P(\bar{A}|B)$
- $P(A_1 + A_2|B) = P(A_1|B) + P(A_2|B) - P(A_1A_2|B)$

当$B = \Omega$时,条件概率化为无条件概率, 故可以把一般的概率看作条件概率.

试证明导出性质的第3条

乘法定律的推广

$$P(A_1A_2 \cdots A_n) = P(A_1)P(A_2|A_1)P(A_3|A_1A_2) \cdots P(A_n|A_1A_2 \cdots A_{n-1})$$

Polya 的坛子[1] 在一只坛子里有b只黑球和r只红球. 随机取出一只, 把原球放回, 并加入与抽出的球相同眼色的球c只. 再摸第二次. 这样下去共摸了n次. 问前面的m次出现黑球, 后面的n-m次出现红球的概率是多少?(这个模型曾经被用来描述传染病模型)

在Polya的坛子模型中, 抽得前3个球依次为(黑,黑,红),(黑,红,黑),(红,黑,黑)的概率是多少?进一步, 3个球中有2个是黑球的概率是多少?n个球中有k个黑球的概率?

俄罗斯轮盘赌的时候, 很多老练的赌徒相信"若红已经连续多次出现, 则在下一次旋转中把赌本压在黑上是明智的", 你怎么看?

111.2 全概率公式

设$A_1, ..., A_n$为一般空间的一个分割, 则

$$B = \sum_{i=1}^{\infty} A_i B$$

由完全可加性和乘法定理得

$$P(B) = \sum_{i=1}^{\infty} P(A_i) P(B|A_i)$$

播种用的一等小麦种子混合有2%的二等种子, 1.5%的三等种子, 1%的四等种子. 一二三四等种子长出的穗含50个以上麦粒的概率分别为0.5,0.15,0.1, 0.05. 求这批种子长出的穗含50个以上麦粒的概率.

[1]Polya, 斯坦福名誉教授, 20世纪最著名的分析家之一. 写有多部科普著作. 推荐《数学与猜想》

111.3 Bayes公式

若事件B能且只能与两两互不相容的事件$A_1, ..., A_n$同时发生, 即

$$B = \sum_{i=1}^{\infty} BA_i$$

由于

$$P(A_i|B) = P(B)P(A_i|B) = P(A_i)P(B|A_i)$$

故

$$P(A_i|B) = \frac{P(A_i)P(B|A_i)}{\sum_{i=1}^{\infty} P(A_i)P(B|A_i)}$$

再利用全概率公式即得

$$P(A_i|B) = \frac{P(A_i)P(B|A_i)}{P(B)}$$

此公式称Bayes公式, 其中

- $P(A_i)$称先验概率
- $P(A_i|B)$称后验概率
- $P(B|A_i)$称条件概率/关于B的似然函数

此式表明了获取信息B后对A_i的新认识. 具有完整的理论依据. 对Bayes公式的应用可以参考《模式分类》

在数字通信中, 由于存在随机干扰, 因此接收到的信号与发出的信号可能不同。为确定发出的信号通常需要计算各种概率。下面是一个简单的模型–二进信道。若发报机以0.7和0.3 的概率发出信号0和1（譬如分别用低电平和高电平表

示）。由于随机干扰, 当发出信号0时, 接收机不一定收到0, 而以概率0.8和0.2收到信号0和1; 同样, 当发出信号1时, 接收机以概率0.9和0.1收到信号1和0。求当接收到信号0时发报机发出信号0的概率。

罐头厂想通过颜色自动识别鱼的种类, 例如鲤鱼和草鱼.设事件A_1为出现鲤鱼, A_2为出现草鱼. B_1为出现黑色, A_2为出现白色. 通过往年的经验(先验概率)知道$P(A_1) = 0.6$,$P(A_2) = 0.4$, 通过分析知道(条件概率), $P(B_1|A_1) = 0.2$,$P(B_2|A_1) = 0.8$, $P(B_1|A_2) = 0.1$,$P(B_2|A_2) = 0.9$. 那么怎样通过颜色来识别鱼的种类呢?

111.4 事件独立性

111.4.1 让我们来"创造"概率测度

仅对任意可数空间$\Omega = \{\omega_1, \omega_2, \cdots, \omega_n\}$, 下面我们将要看到建立概率测度是多么容易. 分两步

1. 对每个样本点ω_i, 我们指定一个任意的"权值"P_i, 使之满足

$$P_i \geq 0; \sum P_i = 1, i = 1, 2, ...n$$

2. 现在,对Ω中任意子集$A \subset \Omega$, 定义 A的概率=A中全体点上的权值的和, 即

$$P(\omega_i) = P_i$$

那么

$$P(A) = \sum_{\omega_i \in A} P_i$$

只要我们验证满足公理即可(这很容易)

可见, 我们可以得到大批的概率测度

实际上, 通过这种方法, 我们可以得到概率测度的全体(但是仅仅对可数情况而言, 对于不可数情况,需要用测度来定义)

以上构造中有一种非常特殊的情况, 即Ω只含有有限个点, 每个点给以相同的权值, 即

$$P_i = \frac{1}{n}$$

这样,我们回到了等可能的情况.(想一下, Ω为可数无穷时, 权值能不能相等? 为什么?)

111.4.2 重复独立试验

我们用一个例子来导出事件独立的概念...

对事件A与B, 若

$$P(AB) = P(A)P(B)$$

则称它们是统计独立的, 简称独立

三个事件统计独立, 若下面四个式子同时成立

$$P(AB) = P(A)P(B) \tag{111.1}$$
$$P(AC) = P(A)P(C) \tag{111.2}$$
$$P(BC) = P(B)P(C) \tag{111.3}$$
$$P(ABC) = P(A)P(B)P(C) \tag{111.4}$$

一个问题. 若$P(AB) = P(A)P(B), P(AC) = P(A)P(C), P(BC) = P(B)P(C)$, 是否有$P(ABC) = P(A)P(B)P(C)$?

n个事件独立你能写出满足的条件吗?写之前估计一下, 共需要多少式子?

几个推论(试着证明一下)

1. 若A, B独立,且$P(B) > 0$, 则

$$P(A|B) = P(A)$$

2. 若A, B独立, 则下列各对事件也相互独立: $(\bar{A}, B)$, $(A, \bar{B})$, $(\bar{A}, \bar{B})$

111.4.3 独立性与概率计算

先介绍一个常用的公式.

如果A_1, A_2相互独立, 则由于$\overline{A_1 \cup A_2} = \bar{A}_1\bar{A}_2$, 所以有

$$P(A_1 \cup A_2) = 1 - P(\bar{A}_1)P(\bar{A}_2)$$

(想一下如果直接计算3个并集的概率会怎么样? n个呢?)

若每个人血清携带某病毒的概率为0.4%, 则混合100个人的血清, 求含病毒的概率?

可靠性理论...

Chapter 112

随机变量的分布和数字特征

112.1 随机变量

112.1.1 定义

在定义域Ω上ω的一个数值函数X

$$\omega \longrightarrow X(\omega)$$

称为Ω上的一个随机变量

112.1.2 随机在哪里

随机是指对ω选择的随机性.

一旦ω确定, X的值也确定了.

112.1.3 让我们来构造随机变量

抛一枚硬币, 试着构造出现结果的随机变量

抛两枚硬币, 试着构造出现结果的随机变量

设Ω是一个包含n个人的母体, 可以记为$\Omega = \{\omega_1, \omega_2, \cdots, \omega_n\}$. 每个人有很多特征. 试着构造他们年龄, 体重, 收入的随机变量

设Ω是容器内分子的全体. 试着构造向左运动的分子的随机变量.

若X, Y是随机变量, 则X+Y, X-Y, X*Y, X/Y 也是随机变量.

若 φ 是两个变量的函数, 且X, Y是随机变量, 那么

$$\omega \longrightarrow \varphi(X(\omega), Y(\omega))$$

也是随机变量. 当不引起误会的时候,也可以简写为$\varphi(X, Y)$.

[注: 上一个定理包含在这个定理内]

112.2 分布

112.2.1 分布列

样本空间为可数有穷时, 我们可以把每个基本事件的概率一一列出, 称为分布列.

连续形式时, 设A为实数的某个子集, 例如 $A = [a, b]$. 那么, $P(\{\omega | X(\omega \in A)\}) = P(\{X \in A\}) = P(\{a <= X <= b\})$

当A缩为一点x时, 我们得到一个重要情况 $A = \{x\}$ 为单点集. 此时, $P(X = x) = P(X \in \{x\})$ 为基本事件的概率.

112.2.2 分布函数

当$A = (-\infty, x]$时, 引入一个记号

$$F_X(x) = P(X <= x)$$

称X的分布函数(就是把所有小于等于x的X值的概率捡出来相加), 又称"累积分布函数"

离散形式:

$$F_X(x) = \sum_{X<=x} P(X) = P(X <= x)$$

连续形式:

$$F_X(x) = \int_{-\infty}^{x} P(u)du$$

例如$F_X(18)$就是小于等于18岁的人的集合的概率

112.2.3 累积分布图

F_X对X作图就是累积分布图

112.3 期望

期望又称数学期望, 均值

112.3.1 离散情况

对于可数样本空间上的一个随机变量X, 定义其数学期望为

$$E(X) = \sum_{\omega \in \Omega} X(\omega)P(\{\omega\})$$

若X取值$x_1, x_2...$, 其概率分别为$p_1, p_2...$

$$E(X) = \sum_{i=1}^{\infty} x_i p_i$$

- 当$E(X)$绝对收敛, 称为数学期望(直观上是合理的, 因为顺序对期望并不是本质的)
- 当$E(X)$发散, 称数学期望不存在

期望可以解释为对X的加权平均

甲乙二射手成绩如下,问平均起来二人谁枪法好?

	甲			乙		
成绩	8	9	10	8	9	10
概率	0.3	0.1	0.6	0.2	0.5	0.3

112.3.2 连续情况

数学期望定义为

$$E(X) = \int_{-\infty}^{\infty} xp(x)dx$$

一块土地分为4块, 面积与价格如下, 问平均价格是多少?

面积	30%	15%	35%	20%
价格	5	10	10	30

112.3.3 一些定理

若X和Y是可和的, 则$X+Y$也是可和的, 并且有

$$E(X+Y)=E(X)+E(Y)$$

某彩票有100张, 其中 1 张奖金10000元, 其余皆为零. 如果我买一张彩票, 我的期望收益是多少? 两张呢? 再进一步, 如果有两种这样的彩票, 那么与其从同一种彩票买两张, 我不如从两种中各买一张, 那么我可能有机会赢得20000元, 这样做是不是对我更加有利?

一个袋子中有N张不同的票券, 我们以有放回的方式一张一张的抽取. 假设我们想收集r张不同的票券, 要期望抽取多少次才能得到它们? (这个问题同随机投弹打击目标类似)

(庞加来公式) 对任意事件$A_1, A_2, \cdots, A_n$, 我们有

$$P(\cup_{i=1}^{n} A_j)=\sum_j P(A_j)-\sum_{j,k} P(A_j,A_k)+\sum_{j,k,l} P(A_j,A_k,A_l)-\cdots+(-1)^{n-1}P(A_1,\cdots,A_n)$$

其中各下标是不同的且从1变到n.

(匹配问题) 两套各标记上从1到n的卡片被随机的匹配, 问至少出现一个匹配成对的概率是多少? 匹配成对的期望数是多少?

112.4 方差和协方差

如果X和Y是相互独立的可和随机变量, 则

$$E(XY)=E(X)E(Y)$$

对离散情况有

$$E(XY)=\sum_j \sum_k x_j y_k P(A_{jk})$$

对连续情况有

$$E(XY) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} uvf(u,v)dudv$$

112.4.1 方差

若$E(X-E(X))^2$存在, 则称之为X的方差, 并记为

$$D(X) = E(X-E(X))^2$$

根据期望的定义, $D(X) = E(X^2) - (E(X))^2$.

标准差称$\sqrt{D(X)}$为根方差或标准差

112.4.2 方差的性质

1. $D(X) = 0 \Longleftrightarrow p(X = C) = 1$ 即X为常数
2. $D(aX) = a^2D(X)$
3. 若$a \neq E(X)$则$E(X-a)^2 > D(X) = E(X-E(X))^2$
4. 若X和Y相互独立并且都有有限方差,则

$$D(X+Y) = D(X) + D(Y)$$

(柯西-施瓦茨不等式)

$$E(XY)^2 <= E(X^2)E(Y^2)$$

(这个不等式的证明方法有很多)

112.4.3 把随机变量标准化

记

$$X' = \frac{X - E(X)}{\sqrt{D(X)}}$$

为 标 准 化 的 随 机 变 量,显 然

$$E(X') = 0, D(X') = 1$$

这 就 是 标 准 化 的 理 由

112.4.4 协 方 差 与 相 关 系 数

若X和Y都 有 有 限 方 差,则

$$E[(X - E(X))(Y - E(Y))] = E(XY) - E(X)E(Y)$$

称 为X和Y的 协 方 差, 记 为$cov(X, Y)$. 量

$$\rho(XY) = \frac{cov(X, Y)}{\sqrt{D(X)}\sqrt{D(Y)}}$$

称 为 相 关 系 数.

Chapter 113

怎样描述数据

113.1 原始数据

113.1.1 收集

- 调查
  - 普查
  - 抽样调查
- 试验

113.1.2 分类

- 数量性状数据对排序有自然的意义
  - 离散变量小数部分往往无意义
  - 连续变量往往是实际情况的近似

  注: 实际测得的数据都是有限值,但离散和连续在本质上是不同的
- 质量性状(属性)数据例如: 颜色深浅, 甜味的浓淡, 叶子的种类等. 分析的时候把它们映射为数值. 排序一般无意义, 除非我们能够指定意义

113.2 位置测度

113.2.1 算术平均数(arithmetic mean)

所有观察值的和除以观察的个数.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

缺点: 对极端值敏感

113.2.2 样本中位数(sample median)

简称中位数(median). 先排序才行.

- 如果n为奇数, 则第$(n+1)/2$个最大观察值就是样本中位数
- 如果n为偶数, 则第$n/2$个最大观察值与$n/2+1$个最大观察值的平均数就是样本中位数

缺点: 对中位值以外的值不敏感

对称性: 利用平均数和中位数可以判断样本分布的对称性(你能看出来吗?)

113.2.3 众数

频数(frequency)与频数表

参考《生物统计学》–董时富编 28-29页

众数(mode)

在一个样本的所有观察值中, 发生频率最大的那个值称为样本的众数

按照众数个数分类,

- 只有一个众数的分布称单峰分布;
- 有两个众数的分布称为双峰分布;
- 有三个众数的分布称为三峰分布
- 没有众数

113.2.4 几何平均(geometric mean)

国外的定义:

$\overline{logx}$的反对数称为几何平均, 这里

$$\overline{logx} = \frac{1}{n}\sum_{i=1}^{n}\log x_i$$

国内的定义:

$$(x_1 x_2 \cdots x_n)^{1/n}$$

113.3 算术平均数的某些性质

113.3.1 改变数据的起点

如果$y_i = x_i + c, i = 1, 2, \cdots, n$,则$\bar{y} = \bar{x} + c$

113.3.2 数据伸缩

如果$y_i = cx_i, i = 1, 2, \cdots, n$,则$\bar{y} = c\bar{x}$

113.3.3 伸缩+改变起点

如果$y_i = c_1 x_i + c_2, i = 1, 2, \cdots, n$,则$\bar{y} = c_1\bar{x} + c_2$

113.4 离散性测度

113.4.1 极差(range)

一个样本中最大与最小观察值之间的差异称为极差

113.4.2 分位数(quantiles)或百分位数

第p个百分位数定义如下:

- 如果$np/100$不是一个整数, 而k是小于$np/100$的最大整数, 则第$k+1$个最大样本点即是第p个百分位数
- 如果$np/100$是一个整数, 则第$np/100$与$np/100+1$个大的观察值的平均值定义为第p个百分位数

113.4.3 偏差

- 偏差
- 绝对偏差

113.4.4 方差与标准差

113.4.4.1 偏差

一个样本中每个观察值与样本平均值偏差的总和永远为零

$$d = \frac{\sum_{i=1}^{n}(x_i - \bar{x})}{n}$$

113.4.4.2 平均偏差

$$\sum_{i=1}^{n} |x_i - \bar{x}|/n$$

113.4.4.3 样本方差(variance)

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$$

113.4.4.4 样本标准差(standard deviation)

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} = \sqrt{s^2}$$

113.4.4.5 方差与标准差的某些性质

- 假设有两样本$x_1, \cdots, x_n$及$y_1, \cdots, y_n$, 此处$y_i = x_i + c, i = 1, \cdots, n$. 若样本方差分别记为$s_x^2$及$s_y^2$, 则

$$s_x^2 = s_y^2$$

- 假设有两样本$x_1, \cdots, x_n$及$y_1, \cdots, y_n$, 此处$y_i = cx_i, i = 1, \cdots, n, c > 0$. 若样本方差分别记为$s_x^2$及$s_y^2$, 则

$$s_y^2 = c^2 s_x^2, \qquad s_y = cs_x$$

113.4.5 变异系数(coefficient variation, CV)

$$100\% * \frac{s}{\bar{x}}$$

113.5 偏斜度与峭度

113.5.1 偏斜度(skewness)

度量数据围绕众数呈不对称的程度, 为标准化了的三阶中心矩, 记为

$$g_1 = \frac{m_3}{m_2^{3/2}}$$

其中m_3为三阶中心矩(third central moment)

$$m_3 = \frac{\sum (x - \bar{x})^3}{n}$$

其中m_2为二阶中心矩(third central moment)

$$m_2 = \frac{\sum (x - \bar{x})^2}{n}$$

类似,三阶原点矩记为

$$m_3' = \frac{\sum x^3}{n}$$

二阶原点矩记为

$$m_2' = \frac{\sum x^2}{n}$$

一阶原点矩记为

$$m_1' = \bar{x}$$

一阶中心矩记为

$$m_1 = \frac{\sum (x - \bar{x})}{n} = 0$$

113.5.2 峭度(kurtosis)

表数据陡峭或平坦的程度, 记为

$$g_2 = \frac{m_4}{m_2^2} - 3$$

113.6 数据的分组

频数分布是按数值大小有序地显示数据中的每个值及出现的频数. 所谓频数即是指数值在数据中出现的次数

113.7 图示法

113.7.1 条形图(bar graph)

113.7.2 直方图(histogram)

113.7.3 茎叶图(stem-and-leaf plot)

113.7.4 盒型图(box plot)

- 异常值(outlying value) 一个观察值x如果属于下面之一, 则为异常值
  - $x >$上百分位数+1.5×（上百分位数-下百分位数）
  - $x <$下百分位数-1.5×（上百分位数-下百分位数）
- 极端异常值(extreme outlying value) 一个观察值x如果属于下面之一, 则为极端异常值
  - $x >$上百分位数+3×（上百分位数-下百分位数）
  - $x <$下百分位数-3×（上百分位数-下百分位数）

Chapter 114

离散分布

114.1 退化分布(单点分布)

当随机变量只取常数值时, 即

$$P(X(\omega) = c) \equiv 1$$

为退化分布, 其分布函数为

$$F(X) = \begin{cases} 0, & x < c \\ 1, & x >= c \end{cases} \tag{114.1}$$

或

$$F(X - c) = \begin{cases} 0, & x <= c \\ 1, & x > c \end{cases} \tag{114.2}$$

期望

$$E(X) = \sum_{\infty}^{\infty} xp(x) = c$$

方差

$$D(X) = E(X^2) - (E(X)^2 = c^2 - c^2 = 0$$

114.2 贝努里分布(两点分布)

一次试验中只有两个结果$\Omega = \{A, \bar{A}\}$, 这种试验称为贝努里试验. 其中

$$P(A) = p, \qquad P(B) = 1 - p = q$$

记X为事件A出现的次数, 则

$$X = \begin{cases} 0, & X\text{不出现} \\ 1, & X\text{出现} \end{cases} \tag{114.3}$$

概率取值为

$$\begin{cases} P(X = 1) = q \\ P(X = 0) = p \end{cases} \tag{114.4}$$

那么我们有

$$P(X) = \begin{cases} p, & X = 1 \\ q, & X = 0 \\ 0. & X = \text{其它} \end{cases} \tag{114.5}$$

期望

$$E(X) = 0 * q + 1 * p = p$$

方差

$$D(X) = E(X^2) - (E(X)^2 = p - p^2 = pq$$

(考虑一下X的取值变为A出现为2, 否则为0, 期望和方差会是什么? [1])

[1]答案为$E(X) = 2p, \quad E(X^2) = 4pq$

下面考虑贝努里分布的母函数

$$g(z) = qz^0 + pz^1 = q + pz$$

那么期望和方差可以由下面得到

$$\begin{aligned} E(X) &= g'(1) = p \\ E(X^2) &= g''(1) + g'(1) = p \\ D(X) &= p - p^2 = pq \end{aligned}$$

114.3 二项分布

在n重贝努里试验中, 记k为A出现的次数, 则k的取值为$0, 1, 2, \cdots, n$.

记A_i为第i次试验中出现事件A, $\overline{A_i}$为第i次试验中A不出现. 若记B_k为n重贝努里试验中, A出现k次这一事件, 则

$$B_k = (A_1 \cdots A_k \overline{A_{k+1}} \cdots \overline{A_n}) + (\cdots) + (\overline{A_1} \cdots \overline{A_{n-k}} A_{n-k+1} \cdots \overline{A_n})$$

右边一共有$\binom{n}{k}$项, 且两两互不相容. 由独立性得出

$$P(A_1 \cdots A_k \overline{A_{k+1}} \cdots \overline{A_n}) = P(A_1) \cdots P(A_k) P(\overline{A_{k+1}}) \cdots P(\overline{A_n})$$

利用概率的加法定理得

$$P(B_k) = \binom{n}{k} p^k q^{n-k}$$

我们常常把此概率记为

$$B(k; n, p) = \binom{n}{k} p^k q^{n-k}$$

期望 [2]

$$E(X) = \sum_{k=0}^{n} P(B_k) = np$$

方差 [3]

$$D(X) = E(X^2) - (E(X)^2 = p - p^2 = npq$$

下面考虑二项分布的母函数

$$g(z) = \sum_{k=0}^{n} P(X = k)z^k = (pz + q)^n$$

也可以这样考虑, 记$X = X_1 + X_2 + \cdots + X_n$, 其中$X_i$为第i次贝努里试验. 由于$X_i$相互独立, 则二项分布的母函数可以由$g(z) = q + pz$的n次方给出

$$g(z) = (pz + q)^n = \sum_{k=0}^{n} \binom{n}{k} q^{n-k} p^k z^k$$

由母函数的定义知, z^k的系数$\binom{n}{k} q^{n-k} p^k = P(X = k)$

那么期望和方差可以由下面得到

$$\begin{aligned} E(X) &= g'(1) = np \\ E(X^2) &= g''(1) + g'(1) = n^2p^2 - np^2 + np \\ D(X) &= npq \end{aligned}$$

[2]小提示: 可以直接计算, 也可以使用独立随机变量的和的期望等于期望的和的性质来计算. 后者更简单一点. 还有一种有点技巧但容易公式化的方法. 母函数的方法最简单

[3]小提示: 同期望一样, 也可以使用几种不同的方法

114.4 几何分布

在n重贝努里试验中, 设A的第一次出现是在第k次试验, 记此事件为W_k, 则

$$W_k = \overline{A_1 A_2} \cdots \overline{A_{k-1}} A_k$$

$$P(W_k) = P(\overline{A_1})P(\overline{A_2}) \cdots P(\overline{A_{k-1}} A_k) = q^{k-1}p$$

记

$$g(k;p) = q^{k-1}p, \quad k = 0, 1, 2, \cdots$$

$g(k;p)$是几何级数的一般项, 因此上式称为几何分布.

验证

$$\sum_{k=1}^{\infty} g(k;p) = \frac{1}{1-q}p = 1$$

期望[4]

$$E(X) = \sum_{k=1}^{\infty} kg(k;p) = \frac{1}{p}$$

而

$$E(X^2) = \sum_{k=1}^{\infty} k^2 g(k;p) = \frac{1+q}{p^2}$$

则方差

$$D(X) = \frac{q}{p^2}$$

[4]虽然有一点点复杂, 但是鼓励你尝试一下

母函数

$$g(z)=\sum_{k=0}^{n}P(X=k)z^k=\frac{pz}{1-qz}$$

期望

$$E(X)=g'(1)=\frac{1}{p}$$

$$g''(1)=\frac{2q}{p^2}$$

方差

$$D(X)=\frac{q}{p^2}$$

114.5 负二项分布(巴斯卡分布)

接着几何分布考虑, 若$T_1,T_2,\cdots,T_n$每个以几何分布的母函数为母函数(回忆一下母函数与分布函数互相唯一确定), 也就是每个都是几何分布(等待第一次成功的次数的随机变量).

记$S_n=T_1+T_2+\cdots+T_n$, 则S_n为第n次成功的等待时间(1次算一个单位时间的话).

我们先来推导两个式子.

第一个式子

$$\frac{1}{1-x} = 1 + x + x^2 + \cdots$$

$$(\frac{1}{1-x})' = \frac{1}{(1-x)^2} = 1 + 2x + 3x^2 + \cdots$$

$$(\frac{1}{1-x})'' = \frac{2!}{(1-x)^3} = 2! + 32x + 43x^2 + \cdots$$

$$(\frac{1}{1-x})^{(n)} = \frac{(n-1)!}{(1-x)^n} = (n-1)! + \frac{n!}{1!}x + \frac{(n+1)!}{2!}x^2 + \cdots + \frac{(n+j-1)!}{j!}x^j + \cdots$$

两边同除以$(n-1)!$, 由归纳法得

$$\frac{1}{(1-x)^n} = \sum_{j=0}^{\infty} \binom{n-1+j}{j} x^j$$

第二个式子–负二项分布(牛顿二项分布的推广)

$$\begin{aligned}
\binom{-n}{j} &= \tfrac{n(n+1)\cdots(n+j-1)}{j!}(-1)^j \\
&= \tfrac{(n-1+j)!}{j!(n-1)!}(-1)^j \\
&= \tbinom{n-1+j}{j}(-1)^j \\
&= \tbinom{n-1+j}{n-1}(-1)^j
\end{aligned}$$

由这两个式子得

$$\frac{1}{(1-x)^n}=\sum_{j=0}^{\infty}\binom{-n}{j}(-1)^j x^j$$

下面我们来看S_n, 由于T_i相互独立, 则S_n的母函数由下式给出(把上式代入)

$$g(z)^n=(\frac{pz}{1-qz})^n=(pz)^n\sum_{j=0}^{\infty}\binom{-n}{j}(-1)^j(qz)^j=\sum_{j=0}^{\infty}\binom{n+j-1}{n-1}p^nq^jz^{n+j}$$

设$k=n+j$, 则

$$g(z)^n=\sum_{k=n}^{\infty}\binom{k-1}{n-1}p^nq^{k-n}z^k$$

根据母函数的定义, 第n次成功出现在第$n+j$次试验的概率为

$$P(S_n=n+j)=\binom{n+j-1}{n-1}p^nq^j$$

下面的等式也是成立的

$$P(S_n=n+j)=\binom{n+j-1}{j}p^nq^j=\binom{-n}{j}p^n(-q)^j$$

由上式给出的分布叫做负二项分布.

我们再来看

$$\frac{g(z)}{z}=\sum_{j=1}^{\infty}\frac{q^{j-1}pz^j}{z}=\sum_{k=0}^{\infty}q^kpz^k$$

观察z^k的系数为T_i-1即第一次成功前失败的次数. 那么

$$(\frac{g(z)}{z})^n=(\frac{p}{1-qz})^n=\sum_{k=0}^{\infty}\binom{n+k-1}{k}p^n(qz)^k$$

就是S_n-n的母函数, 即第n次成功前失败的次数.

另外可以这样考虑, 若第n次成功发生在第$n+j$次试验, 当且仅当$n+j-1$次试验中成功$n-1$次, 失败j次, 且第$n+j$次成功, 故有

$$P(S_n=n+j)=\binom{n+j-1}{n-1}p^{n-1}q^jp=\binom{n+j-1}{j}p^nq^j=\binom{-n}{j}p^n(-q)^j$$

114.6 泊松分布

114.6.1 定义等

若随机变量X可以取一切非负整数值, 且

$$P(X=k)=\frac{a^ke^{-a}}{k!},\qquad a>0$$

则称X服从泊松分布

验证[5]

$$\sum_{k=0}^{\infty}P(X)=1$$

期望[6]

$$E(X)=a$$

[5]您可以在大部分的教科书中找到
[6]同上

而

$$E(X^2) = a^2 + a$$

方差[7]

$$D(X) = a$$

母函数[8]

$$g(z) = \sum_{k=0}^{\infty} \frac{a^k e^{-a} z^k}{k!} = e^{a(z-1)}$$

$$g'(z) = ae^{a(z-1)}$$

$$g''(z) = a^2 e^{a(z-1)}$$

由此我们又一次可以方便的得到期望与方差

114.6.2 从二项分布到泊松分布

==有时间的话推导一下==

[7]同上
[8]同上

Chapter 115

连续分布

115.1 定义

随机变量取某个区间[a,b]或$(-\infty,\infty)$的一切值. 其分布函数F(X)绝对连续, 即存在可积的函数p(x)使

$$F(X)=\int_{-\infty}^{x}p(y)dy$$

称p(x)为X的密度函数

115.2 性质

由公理知

$$p(x)>=0$$

$$\int_{-\infty}^{\infty}p(x)dx=1$$

$$p(x) = F'(x)$$

由于$p(x)$与$F(X)$相互确定, 则若$p(x)$满足上式, 推出$F(X)$也是一个分布函数.

由$F(X)$的定义知

$$P(a < x <= b) = F(b) - F(a) = \int_a^b p(x)dx$$

下面看X等于定值的概率

$$P(x = c) <= \lim_{h \to 0} \int_c^{c+h} p(x)dx = 0$$

由于$P(x = a) >= 0$, 故

$$P(X = c) = 0$$

即连续型随机变量取个别值的概率为0.(概率为0的事件不一定不可能; 概率为1的事件不一定必然发生)

由于

$$p(x)\Delta x \approx \int_x^{x+\Delta x} p(y)dy = F(x + \Delta x) - F(x)$$

故$p(x)$反映了取x邻近值的概率的大小.

115.3 均匀分布

密度函数

$$p(x) = \begin{cases} 1/(b-a) & a <= x <= b \\ 0 & x < a \quad or \quad x > b \end{cases}$$

分布函数

$$F(x) = \int_{-\infty}^{x} p(y)dy = \begin{cases} 0 & x <= a \\ (x-a)/(b-a) & a < x <= b \\ 1 & x > b \end{cases}$$

其它[1]

$$E(X) = \int_{-\infty}^{\infty} xp(x)dx = \frac{a+b}{2}$$

$$E(X^2) = \int_{-\infty}^{\infty} x^2 p(x)dx = \frac{a^2+ab+b^2}{3}$$

$$D(X) = \frac{(b-a)^2}{12}$$

115.4 正态分布

推导泊松分布的时候总是感觉有点不太正常(还记得泊松分布的条件吗?), 而且还有计算二项分布值的广泛需要.例如: n=100, p=0.5, k=50时$B(k;n,p)$的值到底是多少?

下面我们将一步一步推导出正态分布的表达式(如果时间允许的话)

[1]在几乎任何概率论教科书上都可以找到推导, 并且它们很简单

115.4.1 Stirling 公式

Stirling 公式[2] 为阶乘的近似计算公式

$\chi(n) = (e/n)^n \sqrt{2\pi n} e^{\omega(n)} = n! \quad (\ 1/(12(n+1/2)) < \omega < 1/12n\ )$

115.4.2 从二项分布到正态分布

- 首先推导当$n \to \infty$时二项系数的值趋于0
- 其次证明当$n \to \infty$时, 对于固定的区间, 二项分布的概率值之和为0
- 再次设$0 < p < 1$, $q = 1 - p$, 且

$$x = \frac{k - np}{\sqrt{npq}} \qquad 0 <= k <= n$$

设A是一个任意而固定的正常数. 于是在满足$|x| <= A$的k的范围内, 我们有

$$\binom{n}{k} p^k q^{n-k} \approx \frac{1}{\sqrt{2\pi npq} e^{-x^2/2}}$$

且收敛是一致的.

- 再次 (棣莫佛-拉普拉斯定理)对任意两个常数a和b, 我们有

$$\lim_{n\to\infty} P(a < \frac{S_n - np}{\sqrt{npq}} <= b) = \frac{1}{\sqrt{2\pi}} \int_a^b e^{-x^2/2} dx$$

115.4.3 定义

以下面的函数

$$F(x) = \frac{1}{\sqrt{2\pi}} \int_a^b e^{-x^2/2} dx$$

[2]推导见《数学分析原理》第二卷第一分册 52页. 作者: 格.马.菲赫金哥尔茨. 译者: 丁寿田

作为分布函数的概率分布称做正态分布. 概率密度函数显然(?)就为

$$p(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$

(好了, 我们该松一口气了, 我们要推导的最困难的公式终于出来了. 但是要知道, 对于分析它还只是很基础的. 有用的是技巧!)

下面来验证一下[3]

$$\int_{-\infty}^{\infty} p(x)dx = 1$$

设随机变量

$$X_j \sim N(\mu_j, \sigma_j^2) \quad j = 1, 2, \cdots, n$$

其中μ_i为均值, σ_j^2为方差. 则

$$X_1 + X_2 + \cdots + X_n \sim N(\sum_{j=1}^{n} \mu_j, \sum_{j=1}^{n} \sigma_j^2)$$

[3]其中的技巧很好. 在菲赫金哥尔茨的《数学分析原理》中至少提供了4中方法来得到这个非正常积分的结果

115.5 指数分布

115.5.1 定义

符合下述密度函数

$$p(x) = \begin{cases} ae^{-ax} & x >= \\ 0 & x < 0 \end{cases}$$

和分布函数

$$F(x) = \begin{cases} 1 - e^{-ax} & x >= \\ 0 & x < 0 \end{cases}$$

的分布称为指数分布.

115.5.2 性质

指数分布有类似几何分布的"无记忆性", 即

$$p(x > s + t | x > s) = \frac{p(x > s + t)}{p(x > s)} = \frac{e^{-a(s+t)}}{e^{-as}} = e^{-at} = p(x > t)$$

指数分布是唯一具有次性质的连续分布.

(可以这样理解, 已知寿命长于s年, 则再活t年的概率与年龄s无关.)

115.5.3 与泊松分布的关系

记X(t)为参数at的泊松分布(过程), 则

$$p(X(t) = k) = \frac{e^{-at}(at)^k}{k!}$$

当k=0时

$$p(X(t)=0)=e^{-at}\sim \text{指 数 分 布}$$

115.6 Γ 分 布

若X(t)是 服 从 参 数 为at的 泊 松 分 布(过 程). 记τ_r为 第r个 跳 跃 发 生 的 时 刻(第r个 例 子 到 来 的 时 刻). 则

$$\{\tau_r<t\}\Longleftrightarrow\{X(t)>=r\}$$

即 第r个 跳 跃 发 生 在 时 刻t之 前, 也 就 是t时 刻 之 前 发 生 至 少r次 跳 跃. 我 们 以F(x)记τ_r的 分 布 函 数, 则 有

$$F(t)=p(\tau_r<t)=p(X(t)>=r)=1-\sum_{k=0}^{r-1}\frac{(at)^k e^{-at}}{k!}$$

那 么[4]

$$p(t)=F'(t)=\frac{a^r t^{r-1}e^{-at}}{(r-1)!}=\frac{a^r t^{r-1}e^{-at}}{\Gamma(r)}$$

称

$$p(x)=\begin{cases}\frac{a^r x^{r-1}e^{-ax}}{\Gamma(r)} & x>0\\ 0 & x<=0\end{cases}$$

的 分 布 为Γ-分 布. 其 中a¿0, r¿0为 参 数.

[4]中 间 的 推 导 只 有 一 点 点 的 烦 琐. 鼓 励 大 家 推 导 出 来 以 增 加 信 心

泊松过程的第r个跳跃发生的时刻服从Γ-分布.

r=1时, Γ-分布变为指数分布

r=正整数时, Γ-分布为r个服从指数分布的随机变量之和的分布, 与负二项分布类似.

Chapter 116

从总体中抽取样本的方法

116.1 总体与样本的关系

- 已知总体, 研究样本. 一般是演绎性的.
- 已知样本, 推断总体潜在或未知的分布性质, 一般是归纳性推论. 也就是说, 拟合数据可以有很多种概率模型, 原则上我们选取一个最好的.

  (我们的内容基本集中在此)

116.2 推断的方法

- 估计用样本数据去估计指定的总体参数
- 假设检验用样本数据去检验总体参数是否等于某个指定的值

116.3 抽样

116.3.1 随机数的产生方法

均匀分布随机数的产生方法

- 真正的随机数–物理方法 抛硬币, 摸球, 粒子发生器, 白噪声...
- 伪随机数 具有类似随机表现的函数–计算机产生
  - 平方取中法
  - 线性同余法
  - 乘同余法
  - 素数模乘同余

  (如何检验伪随机数发生器的效果?)

有人已经编制成随机数表(两种方法都可以), 方便使用.

其它连续分布随机数的产生方法

由均匀分布的随机数取反函数得到.[1]

116.3.2 抽样的方法

随机选择

随机分配

随机化临床试验

- 区组随机化 比较不同组(¿1)的处理效果而随机分组. 通常

[1]参考《概率论》第一册 148页(复旦大学编)

为了更有可比性, 每组相等, 但不是必须.

- 分层在某些临床研究中, 病人可以再分为子群(层), 他们是按照某种特征来划分的. 典型的分层特征有: 年龄, 性别, 临床表现等.

116.4 临床研究中的盲法

双盲是医学临床研究中的金标准

- 双盲医生和病人都不知道每个病人的处理
- 单盲病人不知道, 但医生知道
- 非盲医生和病人都知道

(盲法的问题: 实际操作比较困难, 有时候病人或医生很容易通过某种表现猜测得到处理的类型)

Chapter 117

估计

117.1 均值的估计

问题: 如何使用一组指定的随机样本去估计潜在的总体的均值?

117.1.1 点估计

抽样分布一个 $\bar{x}$ 是从参考总体中所有可能大小为 n 的样本中的一个样本计算出来的样本均值. $\bar{x}$ 的抽样分布是指大量 $\bar{x}$ 值的分布.

关键是: 我们得到的总是一个出现的样本, 而在抽样分布中, 需要我们考虑所有可能的含量为 n 的样本. 即在不同的时候(不同的人抽样)出现的样本是不同的.

设 $x_1, x_2, \cdots, x_n$ 为从具有均值为 μ 的同一个总体出去的一个随机样本, 定义 $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$. 则所有 $\bar{x}$ 的均值为 $E(\bar{x}) = \mu$, 对所有分布成立.

无偏估计量一个参数 θ 的估计量是 $\hat{\theta}$, 如果$E(\hat{\theta}) = \theta$, 则称 $\hat{\theta}$ 是 θ 的无偏估计. 这意味着, 在大量重复的抽样中(样本量为 n), $\hat{\theta}$ 的平均值将会是 θ.

最小方差无偏估计 当 x 的潜在总体分布是正态时, 可以证明, $\bar{x}$ 的方差是最小的. 即 $\bar{x}$ 是 μ 的最小方差无偏估计.

117.1.2 均值的标准误

均值的标准误差 设 $x_1, x_2, \cdots, x_n$ 是从严格总体中抽得的一组随机样本, 总体的方差为 σ^2/n. 则在大小为 n 的样本的重复抽样中, 样本均数的集合总体中, 这个 $\bar{x}$ 集合的方差为 σ^2/n, 标准差为 $\sigma/\sqrt{n}$, 后者也常常称为均值的标准误差(standard error of mean(sem)), 或者称为标准误差(standard error).

均值标准误的计算(此处和以后用var(x)表示变量 x 的方差):

$$var(\bar{x}) = \frac{1}{n^2} var(\sum_{i=1}^{n} x_i) = \frac{1}{n^2} \sum_{i=1}^{n} var(x_i)$$

由定义知, $var(x_i) = \sigma^2$, 因此

$$var(\bar{x}) = \frac{1}{n^2}(\sigma^2 + \sigma^2 + \cdots + \sigma^2) = \frac{1}{n^2}(n\sigma^2) = \sigma^2/n$$

实际上, 总体方差常常未知. 后面会看到, σ^2 的合理估计是 s^2, 那么均值标准误差的估计量是 $s/\sqrt{n}$ –样本均值集合的标准差

(注意: 不是样本的标准差)

117.1.3 均值的区间估计

我们常常希望得到均值的严格似乎合理的区间估计. 下面的区间估计仅当未知分布是正态分布才是正确的. 若不是正态分布, 则只能近似成立.

若$\bar{x} \sim N(\mu, \sigma^2/n)$, 那么把 $\bar{x}$ 写为标准形式, 即

$$z = \frac{\bar{x} - \mu}{\sigma/\sqrt{n}}$$

则 z 应该是标准正态分布. 当重复抽样时, 95% 的 z 值落入 -1.96 到 1.96 之间. 但是 σ 在实际中很少知道.

117.1.4 t 分布

当 σ 未知时, 合理的估计是用样本的标准差 s 估计 σ 而用代替后计算的 z 来构建置信区间. 问题是, 此时的 z 已经不是正态分布了. 此时的 z 的分布是 t 分布.[1]

这个分布的形状和 n 的关系很大. 即 t 分布并不是一个分布, 而是一组分布, 依赖于称为 "自由度(degree of freedom–简写为df)" 的一个量.

如果$x_1, \cdots, x_n \sim N(\mu, \sigma^2)$ 且彼此独立, 则 $\frac{\bar{x}-\mu}{s/\sqrt{n}}$ 的分布称为具有 (n-1) 自由度的 t 分布.

具有自由度 d 的 t 分布上的第 100*u 的百分位点记为 $t_{d,u}$, 即

$$P(t_d < t_{d,u}) = u$$

正态分布中均数的置信区间具未知方差的正态分布的均值 μ 的 $100\% * (1-\alpha)$ 置信区间 (confidence interval, CI) 可以写成

$$(\bar{x} - t_{n-1,1-\alpha/2} s/\sqrt{n}, \bar{x} + t_{n-1,1-\alpha/2} s/\sqrt{n})$$

正态分布中均数的置信区间(大样本) 具未知方差的正态分布的均值 μ 的 $100\% * (1-\alpha)$ 置信区间 (confidence interval, CI) 当 n ¿ 200 时, 可以写成

$$(\bar{x} - z_{1-\alpha/2} s/\sqrt{n}, \bar{x} + z_{1-\alpha/2} s/\sqrt{n})$$

[1]这个问题首先在1908年由统计学家 Willianm Gossett 解决. 在其职业生涯中, 他在爱尔兰的应该叫做 Guinness Brewery 的酿酒厂工作. 他为自己选了一个笔名 "Student", 于是 $\frac{\bar{x}-\mu}{s/\sqrt{n}}$ 的分布就常常称为学生氏 t 分布(Student's t distribution)

置信区间的含义: μ 是一个未知但固定的值, 但是不同的样本会有不同的均值及方差, 故有不同的边界. 所以对总体可以重复抽样然后构建出上述置信区间. 即在所有的构建出的置信区间中有 95% 含有未知的参数 μ.

影响置信区间长度的因素有 n, s, α, 因为置信区间由这三个量决定.

- n—-增加时区间长度减小
- s—-反映了分散性, 它增加时区间长度增加
- α—-希望增加置信度, 即减小 α, 则置信区间长度会增加.

117.2 方差的估计

117.2.1 点估计

设 $x_1, \cdots, x_n$ 是均值为 μ 方差为 σ^2 的某总体的一组样本. 在样本量为 n 的所有随机样本中, 样本方差 s^2 是 σ^2 的无偏估计. 即 $E(s^2) = \sigma^2$.

如果我们在总体中重复样本量为 n 的抽样, 计算每一组样本的方差 s^2, 则大量的样本方差 s^2 取平均, 它就是总体方差 σ^2. 此公式对任何分布有效.

这里再次写出样本方差的公式

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x}^2)$$

如果用 n 代替 s^2 中的 n-1, 那么它将是对 σ^2 的一个偏低的估计. (大样本时(¿200)可以忽略).

117.2.2 卡方分布

要寻找方差的区间估计, 我们需要介绍一类新的分布–卡方分布(Chi-square distribution, χ^2)

设 $x_1, \cdots, x_n \sim N(0,1)$, 且彼此独立. 如果 $G = \sum_{i=1}^{n}$, 那么 G 称为自由度(df)为 n 的卡方分布. 这个分布常常被记为 χ_n^2. 同 t 分布一样, 卡方分布也是一组分布, 依赖于自由度 df. 但是它不是对称的分布, 只有正值且向右倾斜.

可以证明, χ_n^2 的期望是 n, 方差是 2n.

具有 n df 的χ_n^2 的第 u 个百分位点记为 $\chi_{n,u}^2$, 即

$$P(\chi_n^2 < \chi_{n,u}^2) = u$$

注意: 卡方分布是倾斜分布, 没有对称性. 即没有上下百分位点的对称关系.

117.2.3 区间估计

为求 σ^2 的估计, 我们需要知道 s^2 的分布. 设 $x_1, \cdots, x_n \sim N(\mu, \sigma^2)$, 则可以证明

$$s^2 \sim \frac{\sigma^2 \chi_{n-1}^2}{n-1}$$

回忆把 x 标准化后就服从标准正态分布, 从卡方分布的定义我们有

$$\sum z_i^2 = \sum \frac{(x_i - \mu)^2}{\sigma^2} \sim \chi_n^2$$

因为通常不知道 μ, 我们使用 $\bar{x}$ 代替, 损失一个自由度, 结果

就有下面的关系式

$$\sum \frac{(x_i - \bar{x})^2}{\sigma^2} \sim \chi^2_{n-1}$$

再回忆 s^2 的定义, 简单处理后我们有

$$(n-1)s^2 = \sum (x_i - \bar{x})^2$$

代入上式有

$$\frac{(n-1)s^2}{\sigma^2} \sim \chi^2_{n-1}$$

由这个方程我们就有求得 σ^2 的 $100\% * (1-\alpha)$ 的置信区间. 实际上, 可以看出

$$P(\frac{\sigma^2 \chi^2_{n-1,1-\alpha/2}}{n-1} < s^2 < \frac{\sigma^2 \chi^2_{n-1,\alpha/2}}{n-1}) = 1-\alpha$$

把不等式分成两个, 然后移项分别求得 σ^2 的表达式, 联合起来, 我们得到

$$P(\frac{(n-1)s^2}{\chi^2_{n-1,1-\alpha/2}} < \sigma^2 < \frac{(n-1)s^2}{\chi^2_{n-1,\alpha/^2}}) = 1-\alpha$$

σ^2 的 $100\% * (1-\alpha)$ 置信区间为

$$[\frac{(n-1)s^2}{\chi^2_{n-1,\alpha/2}}, \frac{(n-1)s^2}{\chi^2_{n-1,1-\alpha/2}}]$$

117.3 二项分布的估计

117.3.1 参数 p 的点估计

记 x 是二项随机变量, 其参数为 n 及 p, p 的无偏估计为事件中的样本比例 $\hat{p}$, 标准误差 $\sqrt{pq/n}$ 的精确估计为 $\sqrt{\hat{p}\hat{q}/n}$.

117.3.2 区间估计

117.3.2.1 正态近似法

回忆参数为 n 及 p, p 的二项分布近似服从正态分布 $N(np, npq)$, 若样本中发生的事件数为X, 则对应的比例 $\hat{p} = X/n$ 也是正态分布, 且参数分别为 p 和 pq/n. 即

$$\hat{p} \sim N(p, pq/n)$$

若在两边乘以 n, 则 n 次贝努里使用中成功的次数 $X = n\hat{p}$, 则有下式, 实际上与二项分布的正态近似相同

$$X \sim N(np, npq)$$

类似于样本均值的区间估计, 二项分布中参数 p 的 $100 * (1 - \alpha)$ 正态近似估计区间为(同样要求 $n\hat{p}\hat{q} \geq 5$)

$$\hat{p} \pm z_{1-\alpha/2}\sqrt{\hat{p}\hat{q}/n}$$

117.3.2.2 精确法

当需要知道参数 p 精确的置信区间时, 使用此方法. 困难之处在于如何根据下式求解, 幸好一般的统计软件和 excel 已经有了相应的函数.

二项分布中参数 p 的 $100*(1-\alpha)$ 的精确的置信区间是求得区间 (p_1,p_2) 满足下式

$$P(X \geq x|p=p_1) = \alpha/2 = \sum_{k=x}^{n} \binom{n}{k} p_1^k (1-p_1)^{n-k}$$

$$P(X \leq x|p=p_2) = \alpha/2 = \sum_{k=0}^{x} \binom{n}{k} p_2^k (1-p_2)^{n-k}$$

117.4 泊松分布的估计

下面我们考察泊松分布的参数 λ 的估计.

117.4.1 点估计

若单位时间(面积, 长度等) 的泊松分布的参数为 λ, 则时间(面积,长度等)为 T 的泊松分布的参数为 $\mu = \lambda T$. 若一事件在时间(面积,长度等) T 内观察到 X 次, 则 μ 的无偏估计 $\hat{\mu} = X$, λ 的无偏估计 $hat\lambda = X/T$. 即

$$E(\mu) = E(X)$$

$$E(\lambda) = E(X)/T$$

117.4.2 区间估计

方法类似于二项分布中求精确置信区间. 若 x 为事件的观察数. T 为观察的时间(面积,长度). 则对 λ 的精确的 $100*(1-\alpha)$ 的置信区间是 $(\mu_1/T, \mu_2/T)$ 满足

$$P(X \geq x|\mu=\mu_1) = \alpha/2 = \sum_{k=x}^{\infty} e^{-\mu_1} \mu_1^k/k!$$

$$P(X \leq x|\mu = \mu_2) = \alpha/2 = \sum_{k=0}^{x} e^{-\mu_2}\mu_2^k/k!$$

117.5 单侧置信区间

如果我们只关心置信区间的一个边界, 那么就可以构建单侧置信区间.

正态分布的 $100*(1-\alpha)$ 下单侧置信区间为

$$x_{下} = \bar{x} + z_\alpha \sigma/\sqrt{n}$$

正态分布的 $100*(1-\alpha)$ 上单侧置信区间为

$$x_{上} = \bar{x} + z_{1-\alpha} \sigma/\sqrt{n}$$

注意: $z_{1-\alpha}$ 用于构建单侧置信区间, 而 $z_{1-\alpha/2}$ 用于构建双测置信区间.

其它分布的单侧置信区间构建方法类似.

Chapter 118

假设检验: 单样本推断

118.1 一般概念

- 零假设(null hypothesis, 也叫无效假设) 常记为 H_0, 指需要检验的假设.
- 备择假设(alternative hypothesis) 常记为 H_1, 是在某种意义上与零假设相反的假设.

如果要估计某分布的均值 μ 是否等于某个值 μ_0, 我们常常写成下面的形式

$$H_0 : \mu = \mu_0 \qquad vs. \qquad H_1 : \mu \neq \mu_0$$

这样有 4 种可能的结果

1. 我们接受 H_0, 实际上 H_0 也是正确的
2. 我们接受 H_0, 实际上 H_1 是正确的
3. 我们拒绝 H_0, 实际上 H_0 是正确的
4. 我们拒绝 H_0, 实际上 H_1 是正确的

实际使用时, 我们不可能证明零假设是否正确, 这样就可能出现两类不同类型的错误

- I 型错误(概率) 当 H_0 为真时我们拒绝 H_0 (的概率). 常常用 α 来表示, 也常称为一个检验的显著性水平.
- II 型错误(概率) 当 H_1 为真时我们接受 H_0 (的概率). 常常使用 β 来表示.

功效(power)

$power = 1 - \beta = 1 - II$型错误的概率 $= P$(拒绝$H_0|H_1$是真)

假设检验中, 我们的目的是使 α β 尽可能的小. 但是两者是矛盾的. 因为 α 变小时很难拒绝接受 H_0 从而使 β 增大, 反之 β 变小则 α 会增大. 我们一般先固定 $\alpha(0.10, 0.05, 0.01, ...)$, 然后再找某个某个检验使 β 尽可能的小, 或等价的使功效尽可能的大.

118.2 正态分布均值的单样本检验: 单侧备择

对正态分布均值的最好的检验是建立在样本均值上.

接受域(acceptance region) H_0 被接受时 $\bar{x}$ 的取值范围称为接受域.

拒绝域(rejection region) H_0 被拒绝时 $\bar{x}$ 的取值范围称为拒绝域.

单尾检验(one-tailed test, 单侧检验) 如果可以确定拒绝域由较小或较大的值构成, 但是不能同时成立, 即备择假设的未知均值小于或大于零假设下的未知均值, 这种情况下的检验称为单尾检验.

其假设可以写作

$$H_0 : \mu = \mu_0 \qquad vs. \qquad H_1 : \mu > \mu_0 (\mu < \mu_0)$$

118.2.1 方差未知的正态分布均值的单样本 t 检验

118.2.1.1 备择均值¡无效均值的假设检验

检验假设

$$H_0 : \mu = \mu_0 \qquad vs. \qquad H_1 : \mu < \mu_0$$

指定的显著性水平为 α, 计算

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

如果 $t < t_{n-1,\alpha}$ 我们拒绝 H_0, 如果 $t \geq t_{n-1,\alpha}$ 我们接受 H_0.

检验统计量(test statistic) 上式中的 t 值称为检验统计量, 因为我们的检验过程是建立在这个统计量上的.

临界值(critical value) 上式中的值 $t_{n-1,\alpha}$ 称为临界值. 因为检验结果依赖于 t 值与它的比较.

临界值方法在预先指定的 I 型错误概率后, 通过比较检验统计量与临界值从而判断检验结果的方法称为假设检验的临界值方法.

- α 的选择 α 水平的选择应该依赖于 I 型及 II 型错误的相对重要性. 大多数人不喜欢 α 水平远超过 0.05. 因此传统上使用 0.05 是最普遍的.

p-值法由检验统计量(例如t)计算出来的其末端或更末端的概率值. 它可以告诉我们检验结果是如何的显著. 其显著性的常用判断标准如下

- 0.01 ≤ p ¡ 0.05, 则结果是显著的
- 0.001 ≤ p ¡ 0.01. 则结果是高度(极其)显著的

- p ¡ 0.001, 在结果是很高的显著.
- p¿0.05, 则结果被认为没有统计显著性.(有的情况下(0.05 ≤ p ¡ 0.1)被认为有弱的显著性)

(科学上的显著性与统计学上的显著性是有区别的, 二者不必一致. 一个结果在统计上有显著性, 并不表明此结果在科学上有多么重要. 这种情况特别容易发生在大样本时, 因为大样本中一个很小的差异也可以被统计学家发现. 相反, 某些统计杀光你不显著的差异可能在科学上是重要的, 因为它可以促使科学家进一步用大样本去判断结果)

118.2.1.2 备择均值¿无效均值的假设检验

检验假设

$$H_0 : \mu = \mu_0 \qquad vs. \qquad H_1 : \mu > \mu_0$$

指定的显著性水平为 α, 计算

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

如果 $t > t_{n-1,\alpha}$ 我们拒绝 H_0, 如果 $t \leq t_{n-1,\alpha}$ 我们接受 H_0.

这个检验的 p-值 为

$$p = P(t_{n-1} > t)$$

118.3 正态分布均值的单样本检验: 双侧备择

大部分情况下, 先验知识是不足以判断在无效假设被否定后备择假设的均值应该取什么方向. 此时, 应该使用双侧检验.

双侧检验(two-tailed test, two-sided test)在备择假设下做研究的参数(此处为 μ)允许大于或小于无效假设下的参数(μ_0).

检验假设

$$H_0 : \mu = \mu_0 \qquad vs. \qquad H_1 : \mu \neq \mu_0$$

指定的显著性水平为 α, 最好的检验统计量

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

判断, 如果 $|t| > t_{n-1,1-\alpha/2}$ 则拒绝 H_0, 如果 $|t| \leq t_{n-1,1-\alpha/2}$ 则接受 H_0.

p-值的计算如同单侧检验一样.

$$p = \begin{cases} 2 * P(t_{n-1} \leq t) & \text{如果} t \leq 0 \\ 2 * [1 - P(t_{n-1} \leq t)], & \text{如果} t > 0 \end{cases}$$

(一般情况下, 双侧检验总是合适的, 因为它得出的显著性结论在任何应该单侧检验中也是可以满足的. 但是如果我们能够从专业知识判断应该是单侧, 则采用单侧检验会比双侧检验有更大的功效. 另外, 决定单侧还是双侧应该在数据分析之前. 如果计算 t 值后再考虑单侧还是双侧, 会产生人为的主观偏差)

118.4 方差已知时的正态分布均值的单样本 z 检验

某些研究中, 根据过去的资料翻查可能方差是知道的. 在这种情况下, 检验统计量 t 可以由 z 代替, 临界值也由相应的标准正态分布的临界值代替. 其中

$$z = \frac{\bar{x} - \mu}{\sigma/\sqrt{n}}$$

其它的计算完全类似于方差未知时的t检验, 不论是单侧还是双侧.

118.5 检验的功效

功效可以告诉我们, 在备择假设是真时(应该否定 H_0 时), 我们可以否定 H_0 的可信程度. 若功效太低, 即使真实的 μ 与 μ_0 之间有差异, 也很难被所用的检验方法发现. 而不充分的样本量总是造成检验的低功效.

118.5.1 已知方差时正态分布均值的单样本z检验的功效

这个检验的假设是

$$H_0 : \mu = \mu_0 \quad vs. H_1 : \mu = \mu_1$$

此处已知潜在的分布是正态分布而总体方差为 σ^2, 则该检验的功效是

$$\Phi(z_\alpha + |\mu_0 - \mu_1|\sqrt{n}/\sigma) = \Phi(-z_{1-\alpha} + |\mu_0 - \mu_1|\sqrt{n}/\sigma)$$

影响功效的因素

- α 变小, 则 z_α 减小, 所以功效也减小.
- 若备择均值远离无效均值(即$|\mu_0 - \mu_1|$增加), 则功效增加.
- σ 增加, 功效减小
- 样本量 n 增加, 功效增加

118.5.2 双侧备择

双侧检验为

$$H_0 : \mu = \mu_0 \quad vs. H_1 : \mu \neq \mu_0$$

在 $\mu = \mu_1$ 的指定下,若分布是正态, 总体方差已知, 则z检验的功效的精确公式为

$$\Phi[-z_{1-\alpha/2} + \frac{(\mu_0 - \mu_1)\sqrt{n}}{\sigma}] + \Phi[-z_{1-\alpha/2} + \frac{(\mu_1 - \mu_0)\sqrt{n}}{\sigma}]$$

近似公式为

$$\Phi[-z_{1-\alpha/2} + \frac{|\mu_0 - \mu_1|\sqrt{n}}{\sigma}]$$

实际上, 若 $\mu_1 > \mu_0$, 则精确公式的第一项相对于第二项,第一项常常被忽略, 反之第二项常常被忽略.

118.6 样本量的决定

问题的描述是这样: 给出了将要进行的研究的显著性水平 α, 备择均值的期望 μ_1, 我们应该取多么大的样本才能达到希望的功效?

其实根据功效的公式,把功效当作已知而样本量 n 未知, 则可以很容易的求得需要的样本量. 从而我们有下面的公式.

118.6.1 单侧备择下的样本量

在单侧检验, 对于正态分布的均值, 显著性水平为 α, 检出有显著性差异的概率为 $power = 1 - \beta$ 时, 所求的样本量为

$$n = \frac{\sigma^2(z_{1-\beta} + z_{1-\alpha})^2}{(\mu_0 - \mu_1)^2}$$

明显, 影响样本量大小的因素有

- σ 增加时, 样本量增加

- α 变小, 样本量也增加
- power 增加时, 样本量也增加
- 无效均值与备择均值的距离($|\mu_0 - \mu_1|$)增加时, 样本量会减少. (距离增加1倍, 样本量会缩小到 1/4)

(一个问题是, 在估计样本量的时候, 如何估计这些参数? 通常无效假设 μ_0 是容易指定的, α 水平也容易指定. 而 power 却不太容易确定. 大多数研究者认为, $power < 80\%$ 是不太合适的. 备选的 μ_1 和总体方差通常是未知的. 它们可以从先前的工作, 经验, 或先验知识中得到. 在缺乏上述知识时, 有时由专业知识判断, 有时则做一些小的试验来估计. 最后要指出, 样本量的估计由于 μ_1 及 α 的不精确而通常只是提示性的, 它们通常都不精确)

118.6.2 双侧备择下的样本量

公式如下

$$n = \frac{\sigma^2(z_{1-\beta} + z_{1-\alpha/2})^2}{(\mu_0 - \mu_1)^2}$$

通常大于对应的单侧检验中的样本量, 因为 $z_{1-\alpha/2}$ 要大于 $z_{1-\alpha}$.

118.6.3 基于置信区间宽度的样本量估计

假设我们要估计正态分布中的均值, 且具有样本方差 s^2, 及要求有双侧 $100 * (1 - \alpha)$ 的置信区间, 使得 μ 的CI 的宽度不超过 L, 则样本量的近似估计为

$$n = 4z^2_{1-\alpha/2}s^2/L^2$$

118.7 假设检验与置信区间的关系

若双侧检验

$$H_0 : \mu = \mu_0 \quad vs. \quad H_1 : \mu \neq \mu_0$$

显著性水平为 α, 则对 μ 的双侧 $100 * (1 - \alpha)$ 置信区间包含了所有被 H_0 接受的值, 而置信区间之外就是拒绝 H_0 而接受 H_1 的值. 故置信区间与假设检验的结果是相同的.

区别

p-中法告诉我们此结果的统计显著性如何精细, 但是统计学的显著性在实际中往往并不是很重要. 因为大样本时, 实际差异可能是不大的,但是样本越大, 统计上就越显著, 这种差异自然不是那么重要.

置信区间往往给出均值可能存在的范围, 但不包括这个结果如何的显著.

所以, 一个好的作法是, 同时给出置信区间和p-值.

118.8 正态分布方差的估计–单样本卡方检验

在方差的置信区间估计及检验中, 正态条件特别重要. 若样本不满足正态性, 则临界值p-值及置信区间都不是有效的.

118.8.1 卡方检验

欲检验

$$H_0 : \sigma^2 = \sigma_0^2 \quad vs. \quad H_1 : \sigma^2 \neq \sigma_0^2$$

计算检验统计量

$$X^2 = (n-1)s^2/\sigma_0^2 \sim \chi_{n-1}^2$$

如果 $X^2 < \chi^2_{n-1,\alpha/2}$ 或 $X^2 > \chi^2_{n-1,1-\alpha/2}$, 则拒绝 H_0 如果 $\chi^2_{n-1,\alpha/2} \leq X^2 \leq \chi^2_{n-1,1-\alpha/2}$, 则接受 H_0

118.8.2 p-值(双侧备择)

同上计算检验统计量 X^2

如果 $s^2 \leq \sigma_0^2$, 则 p-值=2*(χ^2_{n-1} 分布曲线下从左到 X^2 的面积)

如果 $s^2 > \sigma_0^2$, 则 p-值=2*(χ^2_{n-1} 分布曲线下从右到 X^2 的面积)

118.9 二项分布的单样本检验

118.9.1 正态近似法

118.9.1.1 单样本检验

双侧备择的假设检验为

$$H_0 : p = p_0 \quad vs. \quad p \neq p_0$$

记检验统计量为

$$z = (p - p_0)/\sqrt{p_0 q_0/n}$$

如果$|z| \leq z_{1-\alpha/2}$, 则接受 H_0. 否则接受 H_1.

118.9.1.2 p-值计算

若 $p < p_0$, 则 p-值=$2 * \Phi(z)$

若 $p \geq p_0$, 则 p-值=$2 * [1 - \Phi(z)]$

118.9.2 精确的p-值计算

双侧备择下, 若 x 为 n 次试验成功的次数, 则精确p-值为

如果 $p \le p_0$, 则 p-值$=2 * P(X \le x)$

如果 $p > p_0$, 则 p-值$=2 * P(X \ge x)$

(注意: 任何时候, p-值都对应于出现在样本点末端或更末端的事件的概率)

118.10 功效及样本量的计算

双侧备择下的假设下,

$$H_0 : p = p_0 \quad vs. \quad p \ne p_0$$

在备择假设具体的指定值 $p = p_1$ 下, 正态近似法检验的功效为

$$\Phi[\sqrt{\frac{p_0 q_0}{p_1 q_1}}(z_{\alpha/2} + \frac{|p_0 - p_1|\sqrt{n}}{\sqrt{p_0 q_0}})]$$

(注意: 这个公式只在 $np_0q_0 \ge 5$ 时使用)

指定功效为 $1 - \beta$, 双侧备择下样本量的估计为

$$n = \frac{p_0 q_0 (z_{1-\alpha/2} + z_{1-\beta}\sqrt{\frac{p_1 q_1}{p_0 q_0}})^2}{(p_1 - p_0)^2}$$

118.11 泊松分布的单样本推断–小样本检验

(对于大样本的检验使用正态近似或二项近似)

如果在研究中事件很罕见(例如某些稀有疾病), 则事件的观察数可以考虑为泊松分布, 其未知期望为 μ, 我们要做的检验是 $H_0 : \mu = \mu_0 \quad vs. \quad H_1 : \mu \neq \mu_0$.

对于临界值法, 我们先要根据观察数x计算出 μ 的双侧置信区间 $(c_1, c_2$, 然后判断: 若 μ_0 在此区间内, 则接受 H_0, 否则接受 H_1.[1]

p-值法中, 在 H_0 为真时, 随机变量 X 是具有参数 μ_0 的泊松分布. 因此 μ 的精确 p-值为

$$min(2 * \sum_{k=0}^{x} \frac{e^{-\mu_0}\mu_0^k}{k!}, 1) \qquad 如果 x < \mu_0$$

$$min[2 * (1 - \sum_{k=0}^{x} \frac{e^{-\mu_0}\mu_0^k}{k!}), 1] \qquad 如果 x \geq \mu_0$$

[1]现在一般使用软件计算置信区间, 其值往往不是整数. 一些科学用表也可以查到

Chapter 119

假设检验: 两样本推断

单样本的检验都是一个样本上的检验, 建立在一个一般性的大总体上, 这个总体的参数被认为是已知的, 而吧一般所在总体的参数与一般性的总体的已知参数作比较.

两一般的假设检验问题是指两个不同总体的潜在参数都是未知的, 是需要做比较的.

119.1 匹配样本 t 检验

当第一组样本中每一个数据点都与第二组样本中的唯一数据点相联系, 这样的两个样本称为匹配(或配对)样本. (paired-sample)这样的研究称为配对研究设计.

119.1.1 匹配t检验

当两个样本是匹配样本, 服从正态分布, 均值分别为 μ 和 $\mu+\Delta$, 方差都是 σ^2. 我们想知道两个样本的均值是否相等, 即 Δ 是否为 0. 检验假设为

$$H_0 : \Delta = 0 \qquad vs. \qquad H_1 : \Delta \neq 0$$

由于是匹配样本, 每一对样本的差记为 d_i, 则 d_i 也是正态分布, 其均值为 Δ 且方差记为 σ_d^2. 于是样本均值之差 $\bar{d}$ 也具有正态分布, 其均值为 Δ 且方差为σ_d^2/n. 因此假设检验可以当作单样本t检验. 故有下面的检验方法, 称为匹配 t 检验. 记

$$t = \bar{d}/(s_d/\sqrt{n})$$

此处 $\bar{d}$ 为平均差异

$$\bar{d} = \Delta = (d_1 + \cdots + d_n)/n$$

s_d 是观察值差异的样本标准差

$$s_d = \sqrt{[\sum_{i=1}^{n} d_i^2 - (\sum_{i=1}^{n} d_i)^2/n]/(n-1)}$$

n = 匹配组数.

如果 $|t| > t_{n-1,1-\alpha/2}$, 则拒绝H_0.

如果 $|t| \leq t_{n-1,1-\alpha/2}$, 则接受H_0.

119.1.2 匹配检验的p-值计算

如果 $t \leq 0$, 则 p-值=2*[t_{n-1}分布曲线下从左到$t = \bar{d}/(s_d/\sqrt{n})$点的面积]

如果 $t > 0$, 则 p-值=2*[t_{n-1}分布曲线下从右到$t = \bar{d}/(s_d/\sqrt{n})$点的面积]

119.1.3 匹配样本均值比较的区间的估计

两匹配样本潜在均值差(Δ)的 $100\% * (1-\alpha)$ 置信区间是

$$\bar{d} \pm t_{n-1,1-\alpha/2} s_d/\sqrt{n}$$

119.2 等方差的两独立样本均值比较的t检验

当两个样本中的数据不发生关系时, 称为独立的两样本. (independent-sample)

设第一组样本量为 n_1, 每一个值都服从正态分布 $N(\mu_1, \sigma^2)$, 其均值为 x_1, 样本方差为 s_1^2

设第二组样本量为 n_2, 每一个值都服从正态分布 $N(\mu_2, \sigma^2)$, 其均值为 x_2, 样本方差为 s_2^2

(此处假定两组潜在方差相等)

119.2.1 t检验

我们要检验

$$H_0 : \mu_1 = \mu_2 \qquad vs. \qquad H_1 : \mu_1 \neq \mu_2$$

我们知道

$$\bar{X_1} - \bar{X_2} \sim N[\mu_1 - \mu_2, \sigma^2(\frac{1}{n_1} + \frac{1}{n_2})]$$

如果 H_0 为真, 则 $\mu_1 = \mu_2$, 于是上式成为

$$\bar{X_1} - \bar{X_2} \sim N[0, \sigma^2(\frac{1}{n_1} + \frac{1}{n_2})]$$

标准化后成为

$$\frac{\bar{X_1} - \bar{X_2}}{\sigma\sqrt{1/n_1 + 1/n_2}} \sim N(0, 1)$$

下一个问题是方差的合并估计, 合理的估计应该是对两个方差加权平均, 权值就是样本方差中的自由度, 于是有

$$s^2 = \frac{(n_1-1)s_1^2 + (n_2-1)s_2^2}{n_1+n_2-2}$$

此处 s_2 的自由度为 n_1+n_2-2. 代入后均值差就变成了自由度为 n_1+n_2-2 的 t 分布, 而不再是 $N(0,1)$. 从而检验统计量为

$$t = \frac{\bar{x_1} - \bar{x_2}}{s\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}}, \qquad df = n_1+n_2-2$$

如果 $|t| > t_{n_1+n_2-2,1-\alpha/2}$, 则拒绝 H_0

如果 $|t| \le t_{n_1+n_2-2,1-\alpha/2}$, 则接受 H_0

119.2.2 p-值

类似, 我们也可以得到 p-值

如果 $t \le 0$, 则 p-值=$2*$(t_{n-1+n_2-2}分布曲线下t值左边的面积)

如果 $t > 0$, 则 p-值=$2*$(t_{n-1+n_2-2}分布曲线下t值右边的面积)

119.2.3 区间估计

我们也可以计算两样本均值真实差异的置信区间.

双侧及等方差下, 两独立样本真实均值差异的双侧 $100\%*(1-\alpha)$ 的置信区间为

$$\bar{x_1} - \bar{x_2} \quad \pm \quad t_{n_1+n_2-2,1-\alpha/2} s\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}$$

119.3 两方差相等性检验-F检验

检验假设

$$H_0 : \sigma_1^2 = \sigma_2^2 \quad vs. \quad H_1 : \sigma_1^2 \neq \sigma_2^2$$

合理的方法应建立在两方差的相对度量上, 即比值. 如果比值很大或很小, 则拒绝 H_0, 若接近 1, 则接受.

首先来看σ_1^2/σ_2^2 的分布

119.3.1 F 分布

方差比的分布由统计学家 R.A.Fisher 和 G.Snedecor 研究完成. 他们在两方差相等的假设下得到上述分布, 称为 F 分布.

F 分布由两个参数, 分子的自由度和分母的自由度决定.

若我们记分子样本的样本量为 n_1, 其自由度为 $n_1 - 1$, 分母样本的样本量为 n_2, 其自由度为 $n_2 - 1$. 则F 分布由 $n_1 - 1$ 及 $n_2 - 1$ 共同决定, 记此时的分布为 F_{n_1-1,n_2-1}

自由度为 d_1, d_2 的F 分布的第 $100 * p$ 百分位点记为 $F_{d_1,d_2,p}$, 即

$$P(F_{d_1,d_2} \leq F_{d_1,d_2,p}) = p$$

具自由度 d_1, d_2 的F分布的下侧第p个百分位点, 就是具有自由度为 d_12, d_1 的F 分布的上侧第p个百分位点的倒数, 即

$$F_{d_1,d_2,p} = 1/F_{d_2,d_1,1-p}$$

119.3.2 F 检验

若要检验 $H_0 : \sigma_1^2 = \sigma_2^2$ *vs.* $H_1 : \sigma_1^2 \neq \sigma_2^2$, 而显著性水平为 α, 则

我 们 计 算 统 计 量

$$F = s_1^2/s_2^2$$

如 果$F > F_{n_1-1,n_2-1,1-\alpha/2}$ 或$F < F_{n_1-1,n_2-1,\alpha/2}$, 则 拒 绝 H_0.

如 果$F_{n_1-1,n_2-1,\alpha/2} \leq F \leq F_{n_1-1,n_2-1,1-\alpha/2}$, 则 接 受 H_0.

精 确 的p-值 由 下 面 得 到

如 果$F \geq 1$, 则 p-值$=2 * P(F_{n_1-1,n_2-1} > F)$

如 果$F < 1$, 则 p-值$=2 * P(F_{n_1-1,n_2-1} < F)$

119.4 方差不等的两个独立样本的t检验

现 假 设 有 两 个 正 态 分 布 的 样 本, 第 一 个 样 本 量 为 n_1, 服 从 $N(\mu_1, \sigma_1^2$. 第 二 个 样 本 量 为 n_2, 服 从 $N(\mu_2, \sigma_2^2$.

我 们 要 检 验$H_0 : \mu - 1 = \mu_2$ $vs.$ $H_1 : \mu_1 \neq \mu_2$. 统 计 学 家 常 称 这 个 问 题 为 Behrens-Fisher 问 题.

我 们 有

$$\bar{X_1} - \bar{X_2} \sim N(\mu_1 - \mu_2, \frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2})$$

在H_0成 立 时

$$\bar{X_1} - \bar{X_2} \sim N(0, \frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2})$$

标 准 化 后 的 检 验 统 计 量 为

$$z = (\bar{x_1} - \bar{x_2})/\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}$$

由于方差通常未知, 使用样本方差分别估计时, 因为潜在方差不同, 所以加权合并方差的方法不可用. 若使用样本方差代替后, 检验统计量变成

$$t = (\bar{x_1} - \bar{x_2}) / \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$$

在H_0成立时, 上述 t 的精确分布难于找出, 但是在合适的 I 型错误下, 已经找到了几个近似的分布.

119.4.1 不等方差下两个独立样本的t检验

此方法为 Satterthwaite 近似方法.

先计算检验统计量 t 如上.

再计算近似自由度

$$d' = \frac{(s_1^2/n_1 + s_2^/n_2)^2}{(\frac{s_1^2}{n_1})^2/(n_1 - 1) + (\frac{s_2^2}{n_2})^2/(n_2 - 1)}$$

把 d' 四舍五入到最近的整数 d'', 则

如果$|t| > t_{d'',1-\alpha/2}$, 则拒绝 H_0.

如果$|t| \leq d'', 1 - \alpha/2$, 则接受 H_0.

119.4.2 p-值

类似地,

如果$t \geq 0$, 则 p-值=$2 * (t_{d''}$分布在t值左边的面积)

如果$t < 0$, 则 p-值=$2 * (t_{d''}$分布在t值右边的面积)

119.4.3 置信区间

类似的可以证明, 不等方差下也有均值差的 $100\% * (1-\alpha)$ 置信区间

$$\bar{x_1} - \bar{x_2} \quad \pm \quad t_{d'',1-\alpha/2}\sqrt{s_1^2/n_1 + s_2^2/n_2}$$

119.5 独立样本均值比较中样本量及功效的估计

估计等样本数, 正态分布的两独立样本均值比较, 双侧检验且显著性水平为 α 功效为 $1-\beta$ 下, 样本量的估计为

$$n = \frac{(\sigma_1^2 + \sigma_2^2)(z_{1-\alpha/2} + z_{1-\beta})^2}{\Delta^2} = \text{每一组的样本量}$$

其中 $\Delta = |\mu_1 - \mu_2|$.

换言之, 每一组样本量为 n 时, 将有 $1-\beta$ 的机会发现两组中真实存在 Δ 的差异.

不等样本数所需要的样本量为

$$n_1 = \frac{(\sigma_1^2 + \sigma_2^2/k)(z_{1-\alpha/2} + z_{1-\beta})^2}{\Delta^2} = \text{第一组的样本量}$$

$$n_2 = \frac{(k\sigma_1^2 + \sigma_2^2)(z_{1-\alpha/2} + z_{1-\beta})^2}{\Delta^2} = \text{第二组的样本量}$$

其中 $k = n_2/n_1$ 为两样本事先指定的比值.

功效的估计为

$$power = \Phi[-z_{1-\alpha/2} + \frac{\sqrt{n_1}\Delta}{\sqrt{\sigma_1^2 + \sigma_2^2/k}}]$$

其 中 $k = n_2/n_1$ 为 两 样 本 事 先 指 定 的 比 值.

Chapter 120

非参数检验

之前的数据被假设来自某个潜在的分布, 这个分布的一般形式是已知的, 只是参数的具体值未知. 估计和检验方法都是基于这个分布, 来得到具体值的点或区间等. 这种方法通常被称为参数统计方法.

而如果分布的形状未知, 中心基线定理似乎又不太合适, 例如样本数太少, 这时就必须使用非参数统计方法(nonparametric statistical method). 该方法对肺部形状很少有要求.

下面介绍几个概念.

基数数据(cardinal data) 是一种有尺度的输送就, 可以用某种尺度测出任何两个数据之间的距离.

进一步, 如果该数据的零点是固定的, 称为比例尺度(ratio scale)数据. 若零点是任意的, 则称为区间尺度(interval scale)数据.

例如, 体温是一种区间尺度数据, 因为它的零点是不确定的, 例如在华氏和摄氏温度中, 零点有不同的意义.

体重及身高是比例尺度数据, 因为零点对它们有明确的意义.

比例尺度中, 任何两个数据的比值是有意义的.

区间尺度数据, 比值可能没有意义, 如温度的比值是没有意义的.

不论哪种形式的基数数据, 均值和标准差都是有意义的.

有序数据(ordinal data) 指它们之间可以排成次序但是却没有指定的数值, 因此通常的算数运算是没有意义的.

例如,颜色的深浅程度和视力的等级, 病情的恶化程度, 对每一水平, 可以用数值去代表, 但是没有一个唯一的标准. 而且它们之间的算数运算是没有意义的.

由于无法使用一组有意义的数值代表此类数据, 故计算它们的均值和标准差是不合适的. 但是我们仍然对此类数据之间的比较感兴趣. 非参数检验即适用于此类数据.

名义尺度数据(nominal sccale data) 不同的数据被分为类型或属性, 而类型是没有次序的.

例如疾病的种类, 零件的型号等, 它们都是某类事物的属性, 是没有次序的.

120.1 匹配数据的符号检验(sign test)

对于有序数据, 我们可以度量它们的相对大小, 但是不能相减, 即不能用差值的大小来衡量其相对关系. 此时使用符号检验.

例如要检验两种防晒膏的效果. 随机涂敷于左右手臂, 阳光下一小时. 假设我们只能判定手臂红色的程度

- A 防晒膏 ¿ B 防晒膏, 记为+1.
- A 防晒膏 ¡ B 防晒膏, 记为-1.
- 两者一样, 记为 0

首先去掉 0 值, 因为它对两种防晒膏的好坏不提供任何信息.

如果 +1 远多于 -1, 有理由相信, B 防晒膏的效果要好于 A. 若 -1 远多于 +1, 那么 A 的效果应该好于 B 的. 若 +1 和 -1 差不多, 那么两者效果可以认定没有显著差别.

实际上, 这是二项分布的一个特例. 此处假设

$$H_0 : p = 1/2 \quad vs. \quad H_1 : p \neq 1/2$$

此处 p 为 A 好于 B 的概率.

设 +1 和 -1 的个数有 n 个, 记 +1 的个数为 c, 那么 $E(c) = np$, $var(c) = npq$. 在零假设成立时, $c = n/2, var(c) = n/4$. 所以我们有

120.1.1 正态近似法

根据上面的描述, $c \sim N(n/2, n/4)$. 在显著性水平为 α 时, 使用双侧检验, 我们有

如果$|c| > \frac{n}{2} + \frac{1}{2} + z_{1-\alpha/2}\sqrt{n/4}$, 则拒绝 H_0

如果$|c| \leq \frac{n}{2} + \frac{1}{2} + z_{1-\alpha/2}\sqrt{n/4}$, 则接受 H_0

精确的p-值为

$$p = 2 * [1 - \Phi[\frac{c - \frac{n}{2} - 0.5}{\sqrt{n/4}}]], \text{如果} c \geq \frac{n}{2}$$

$$p = 2 * [\Phi[\frac{c - \frac{n}{2} + 0.5}{\sqrt{n/4}}]], \text{如果} c < \frac{n}{2}$$

(正态近似适用于 $npq \geq 5, \ or \ n(1/2)(1/2) \geq 5, \ or \ n \geq 20$)

120.1.2 精确方法

如果 n ¡ 20, 则需要使用精确的二项分布公式. 其 p-值计算为

$$p = 2 * \sum_{k=c} n \binom{n}{k} \frac{1}{2}^n, \quad say \quad c > n/2$$

$$p = 2 * \sum_{k=0} c \binom{n}{k} \frac{1}{2}^n, \quad say \quad c < n/2$$

$c = n/2, \quad p = 1.0$

Chapter 121

试验设计

[1]

121.1 基本原理

121.1.1 意义

- 广义指整个研究课题的设计, 包括从申请到结题评估的全部过程
- 狭义仅指试验单位的选择, 分组与排列等

121.1.2 基本要求

- 试验目的要明确
- 试验条件有代表性–可推广
- 试验结果可靠–严格试验要求与操作, 减少试验误差
- 试验结果能够重演–在不同的时间, 地域等(主要指农业试验)

[1]主要根据《生物统计学》(第二版) 第八章. 著者: 李春喜 等

121.1.3 试验设计的基本要素

- 处理因素–对对象给予的某种外部干扰(或措施), 简称处理. 例如: 温度, 压力是两个不同的因素
- 受试对象–就是被处理的对象
- 处理效应–处理完后的结果, 体现在数据上, 其中包含误差.

121.1.3.1 试验误差及控制途径

试验误差的分类

- 系统误差(可以避免)–设计不当造成的误差. 例如其它条件本应一样而不一样.
- 随机误差(不可避免)–不可控制的偶然因素造成

误差来源

- 试验材料固有的差异
- 试验条件不一致
- 操作技术不一致
- 偶然性因素的影响

控制途径

- 选择纯合一致的试验材料
- 改进操作管理制度
- 精心选择试验单位
- 采用合理的试验设计

121.1.3.2 试验设计的基本原理

- 重复
- 随机化
- 局部控制

121.2 对比设计及其统计分析

121.2.1 对比设计

- 在农作物试验上讲究地块平行
- 动植物试验就是配对试验

121.2.2 统计分析

一般使用 t 检验

121.3 随机区组设计及统计分析

121.3.1 设计

因为试验单位性质不同, 因而分为不同的组.

例如, 有 2 个组(长势好的A, 不好的B), 每个组试验 3 种温度(1个因素的 3 个水平), 设 A 组有90个个体, B组也有90个个体, 那么每个组都可以分为3组处理.

121.3.2 统计

一般使用方差分析, 可以比较A B两组的不同(两个组产量是否有显著差异), 也可以比较3种温度的不同影响(例如3种温度下产量是否有显著差异).

121.4 拉丁方设计

优点: 其误差是随机区组设计的73%.

缺点: 需要保持行, 列, 处理数三者相等. 故处理数不能太多(一般5-10, ¡=4 自由度不够, ¿10就太庞大了). 自由度至少12, 最好20以上.

步骤

1. 选择标准拉丁方
2. 行随机化
3. 列随机化
4. 处理随机化

统计分析

行之间, 列之间, 处理之间都可以当作区组, 均可比较, 故比随机化区组多了一项.

121.5 裂区设计(主要针对农业试验)

如果是多因素试验处理的组合数太多. 应用条件:

1. 若已知某因素内部差异大, 则选择作为主因素

2. 若某因素需要更多的区域(资源), 则应作为主因素

3. 若某因素要求的精度高宜作为主因素

4. 若需临时再加入一个试验因素, 可以在原设计的小区里(随机区组)再加一个因素, 这样就成了裂区设计(但是应尽量在试验前设计好, 避免中途更改试验方案).

这里只介绍二因素设计.

121.6 正交设计

对付多因素多水平的试验. 例如, 3因素3水平的完全试验组合要$3^3 = 27$个, 4因素4水平的完全试验组合要$4^4 = 256$个. D. J. Finney 倡议部分试验, 后来成为正交设计. 实质上是把具有代表性的试验做了, 其余忽略掉了.

Part XVII

参考文献

Bibliography

[1] 张奠宙. *20*世纪数学经纬. 华东师范大学出版社. 2001

[2] 朱慧明 韩玉启 著. 贝叶斯多元统计推断理论. 科学出版社. 2006

[3] 张尧庭 陈汉峰 编著. 贝叶斯统计推断. 科学出版社. 1991

[4] 钟开莱. 初等概率论附随机过程. 人民教育出版社. 1979

[5] 复旦大学. 概率论第一册概率论基础, 人民教育出版社, 1979.

[6] 中山大学数学系 梁之舜 邓集贤 杨维权 司徒荣 邓永录 概率论及数理统计*(*第二版*)*, 高等教育出版社, 1988.

[7] [巴西] J.塞图宝 , J.梅丹尼斯 著. 朱浩 等译 计算分子生物学导论*(Introduction to Computational Molecular Biology)*. 科学出版社, 2003.08.

[8] (美) Richard O. Duda, Peter E. Hart, David G. Stork. 李 宏 东 姚天翔 等译. 模式分类*(Pattern Classification)*. 机械工业出版社, 中信出版社. 2003

[9] (英) John Shawe-Taylor (美) Nello Cristianini 著. 赵玲玲 翁苏明 曾华军 等译. 模式分析的核方法*(Kernel Methods for Pattern Analysis)*. 机械工业出版社. 2006

[10] Martin T. Hagan, Howard B. demuth, Mark H. Beale 著. 戴 葵 等译, 李伯民审校. 神经网络设计. 机械工业出版社. 2002.9

[11] 徐克学. 生物数学. 科学出版社. 2002

[12] 杜荣骞. 生物统计学, 高等教育出版社.

[13] 李春喜 王志和 王文林 生物统计学 科学出版社. 2000.

[14] Bernard Rosner. 孙尚拱译. 生物统计学基础 (*Fundamentals of Biostatistics*) 第五版. 科学出版社, 2004.

[15] 徐端正.生物统计学-在实验和临床药理学中的应用. 科学出版社. 2004

[16] 茆诗松, 周纪芗, 陈颖. 试验设计. 中国统计出版社. 2004.

[17] 朱永生 著. 实验物理中的概率和统计*(*第二版*)*, 科学出版社. 2006年4月.

[18] (新西兰) Ian H. Witten, Eibe Frank 著. 董琳 邱泉 于晓峰 吴韶群 孙立骏 译数据挖掘–实用机器学习技术*(*第二版*)*, 机械工业出版社. 2006.2.

[19] [美] W.J.Conover 著, 崔恒建 译. 实用非参数统计*(*第三版*)*. 人民邮电出版社. 2006

[20] [美]A. Malcolm Campbell and Laurie J. Heyer. 孙之荣主译 *Discovering Genomics Proteomics and Bioinformatics(*探索-基因组学、蛋白质组学和生物信息学*)*. 科学出版社. 2004.07.

[21] 薛毅 陈立萍 编著. 统计建模与*R*软件. 清华大学出版社. 2006.

[22] 顾万春 著. 统计遗传学. 科学出版社. 2004

[23] [美] Albert Boggess, Francis J. Narcowich 著, 芮国胜 康健 等译. 小波与傅里叶分析基础*(A First Course in Wavelets with Fourier Analysis)*. 人民邮电出版社. 2006

[24] 王燕 编著. 应用时间序列分析. 中国人民大学出版社. 2005

[25] W. N. Venables, D. M. Smith, R 核心开发小组 (the R Development Core Team), 丁国徽译 *R* 导论–关于 *R* 语言的注解: 一个数据分析和图形显示的程序设计环境 英文版本2.3.0 (2006-04-24) 中文版本0.1 (2006-06-15). 2006.

[26] Emmanuel Paradis. 翻译: (Chap1–2: 王学枫 ; Chap3: 谢益辉 ; Chap4: 李军焘 ; Chap5–7: 丁国徽) *R for Beginners* Chinese Edition 2.0. 2006

[27] *R*语言简介–*R*语言笔记:数据分析与绘图的编程环境 (版本1.7) R Development Core Team. June 10, 2006

[28] Brian S. Everitt and Torsten Hothorn. *A Handbook of Statistical Analyses Using R* CHAPTER 9. Survival Analysis: Glioma Treatment and Breast Cancer Survival

[29] Brian S. Everitt and Torsten Hothorn. *A Handbook of Statistical Analyses Using R* CHAPTER 12, Meta-Analysis: Nicotine Gum and Smoking Cessation and the Efficacy of BCG Vaccine in the Treatment of Tuberculosis.

[30] Fabio Frascati, Elia Biganzoli and Bruno Mario Cesana *'agreement': Analyse the agreement between two measurement methods*

[31] Robert Sedgewick *ALGORITHMS IN C++, PART 5–Graph Algorithms(Third Edition)* 影印版. Pearson Education 出版集团. 高等教育出版社. 2002.10

[32] Vikneswaran *An R companion to Experimental Design*

[33] Petra Kuhnert, Bill Venables *An Introduction to R: Software for Statistical Modelling and Computing* CSIRO Australia, 2005

[34] Virasakdi Chongsuvivatwong. *Analysis of Epidemiological Data using R and Epicalc*. Prince of Songkla University.

[35] Robert Gentleman, Kurt Hornik, Giovanni Parmigiani. *Analysis Of Phylogenetics And Evolution With R Paradis*. Springer Science+Business Media, LLC. 2006.

[36] Jim Albert *Bayesian Computation with R* 2007 Springer Science+Business Media, LLC

[37] Yang ZiHeng, *Computaional Molecular Evolution (*有中文版*)*, Oxford University Press, 2006

[38] Grant V. Farnsworth *Econometrics in R*. June 26, 2006

[39] Alexandros Karatzoglou(Technische Universit¨t Wien), Alex Smola(Australian National University, NICTA),Kurt Hornik(Wirtschaftsuniversit¨t Wien) *kernlab – An S4 Package for Kernel Methods in R*

[40] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S(Fourth edition)*. Springer (mid 2002) Final 15 March 2002

[41] Paul Bliese (paul.bliese@us.army.mil) . *Multilevel Modeling in R (2.2)–A Brief Introduction to R, the multilevel package and the nlme package*. October 28, 2006

[42] John Fox. *Nonlinear Regression and Nonlinear Least Squares*. January 2002

[43] Julian J. Faraway *Practical Regression and Anova using R* July 2002.

[44] @Manual, title = R: A Language and Environment for Statistical Computing, author = R Development Core Team, organization = R Foundation for Statistical Computing, address = Vienna, Austria, year = 2008, note = ISBN 3-900051-07-0, url = http://www.R-project.org,

[45] R Development Core Team. *R Language Definition* 2006.06.01

[46] Vincent Zoonekynd (zoonek@math.jussieu.fr) *Statistics with R* 28th August 2005/2007

[47] Kim Seefeld, Ernst Linder *Statistics Using R with Biological Examples*. 2007

[48] John Verzani. *simpleR – Using R for Introductory Statistics* 2001.

[49] David Meyer. *Support Vector Machines–The Interface to libsvm in package e1071* Technische Universit¨t Wien, Austria, David.Meyer@ci.tuwien.ac.at, January 6, 2009

[50] Michael P. Fay. *Testing the Ratio of Two Poisson Rates*. June 5, 2007

[51] Brockwell, Peter J. and Davis, Richard A. *Time Series: Theory and Methods* Springer-Verlag. 1987

[52] Trevor Hastie and Robert Tibshirani. *Generalized additive models.* Statistical Science. 1986, Vol. 1, No. 3, 297-318 @Article, author = Trevor Hastie,

title = •, journal = •, year = •, OPTkey = •, OPTvolume = •, OPTnumber = •, OPTpages = •, OPTmonth = •, OPTnote = •, OPTannote = •

[53] Karline Soetaert *Using R for scientific computing.* September 2008