Data Analysis Using R: Chapter11

罗智超 (ROKIA.ORG)

1 通过本章你将学会

- 回归分析

1.1 比较三个概念

- 回归分析：着重寻求变量之间近似的函数关系
- 相关分析：致力于寻求一些数量性指标，以刻画有关变量之间关系深浅的程度
- 方差分析：着重考虑一个或一些变量对一特定变量的影响有无及大小，由于其方法是基于样本方差的分解，故称为方差分析。

1.2 模型拟合

- 不需要任何假定的拟合。如决策树、boosting、随机森林等
- 不需要分布假定但需要模型假定的拟合。如，最小二乘回归
- 需要对分布和模型做假定的拟合。最大似然法需要对数据的分布和模型形式作出假定。

1.3 模型评价

- 交叉验证可以在任何模型之间做客观的比较
- 对于参数模型，再对模型和数据的各种数学假定下的评估

1.4 回归类型

- 简单线性用一个解释变量（数量）预测响应变量（数量）
- 多项式用一个解释变量（数量）预测响应变量（数量），模型的关系是 n 阶多项式
- 多层用拥有等级结构的数据预测一个响应变量。也称为分层模型、嵌套模型或混合模型
- 多元线性用两个或者多个解释变量（数量）预测一个响应变量（数量）
- 多变量用一个或者多个解释变量预测多个响应变量（数量）
- logistic 用一个或者多个解释变量预测一个响应变量（二值类别）
- 柏松用一个或者多个解释变量预测一个代表频数的响应变量
- Cox 比例风险用一个或者多个解释变量预测一个事件（死亡、失败或者旧病复发）发生的时间
- 时间序列对误差项相关的时间序列数据建模
- 非线性用一个或者多个解释变量预测一个响应变量，不过模型是非线性的
- 非参数用一个或者多个解释变量预测一个响应变量，模型的形式源自数据形式，不事先设定，如 KNN
- 稳健 Robust 用一个或者多个解释变量预测一个响应变量，能抵御强影响点的干扰。

1.5 lm()

```
fit<-lm(weight ~height,data = women)
summary(fit)
```

```
# myfit <- lm(formula, data)
# formula: y~x1+x2+..+xk
# : 预测变量交互项
# * 表示所有可能的交互项目 x*y*z= x+y+z+x:y+x:z+z:y+x:y:z
# ^ 表示交互项大道某个次数 (x+y+z)^2=x+y+z+x:y+x:z+z:y
# . 表示除了因变量意外的所有自变量 y~.
# - 表示从等式中移除某个变量。(x+y+z)^2-x:z
# -1 删除截距项 y~x-1
# I() y~x+I((x+z)^2)
# function 可以再表达式中使用的数学函数 log(y)~x+z
```

```
# 常用的函数

summary(fit) # 拟合模型的详细结果
coefficients(fit) # 拟合模型的模型参数
confint(fit) # 模型参数的置信区间
fitted(fit) # 拟合模型的预测值
residuals(fit) # 拟合模型的残差值
anova(fit) # 拟合模型的方差分析表
vcov(fit) # 模型参数的协方差矩阵
AIC(fit) # 赤词信息统计量
plot(fit) # 生成评价拟合模型的诊断图
plot(women$height,women$weight)
abline(fit)
predict(fit) # 用你和模型对新数据集预测响应变量值
```

```
# 简单线性回归
fit<-lm(weight ~height,data = women)
summary(fit)
plot(fit)
fit2<-lm(weight~height+I(height^2),data=women)
summary(fit2)
plot(fit2)
```

```
# 多元线性回归
states<-as.data.frame(state.x77[,c("Murder",
    "Population","Illiteracy","Income","Frost")])
# 检查变量之间的相关性
cor(states)
library(car)
scatterplotMatrix(states,spread=FALSE,
              soother.args=list(lty=2),
              main="Scatter Plot Matrix")
# 多元线性回归

fit<-lm(Murder~Population+Illiteracy+
          Income+Frost,data = states)
summary(fit)

# 带有交互项的多元线性回归

fit<-lm(mpg~hp+wt+hp:wt,data=mtcars)
summary(fit)
library(effects)
plot(effect("hp:wt",fit,,
            list(wt=c(2.2,3.2,4.2))),
     multiline=TRUE)
```

1.6 如何评价模型拟合效果

- 残差标准误

$$RSE = \sqrt{\frac{1}{n-2}RSS} = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

- R^2 统计量

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}$$

$$TSS = \sum (y_i - \bar{y})^2$$

- F 统计量

$$F = \frac{(TSS - RSS)/p}{RSS/(n-p-1)}$$

- 交叉验证（Cross Validation）

交叉验证的方法是再计算机时代发展起来的，它用训练数据集来训练模型，然后用未参与建模的测试数据集来评价模型预测功能的优劣。交叉验证不用对模型做任何假定。最常用的是 N 折交叉验证。把数据随机分为 N 份，轮流把其中 1 份作为测试集，其余 $N-1$ 份合起来做训练集。从而得到 N 个标准化均方误差（Normalized Mean Squared Error, NMSE）

$$NMSE = \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y}_i)^2}$$

1.7 OLS 回归的统计假设

- 正态性当预测变量值固定时，因变量成正太分布，而残差也应该时一个均值为 0 的正态分布。
- 独立性只能从收集数据的背景信息判断因变量是否互相独立
- 线性因变量与自变量线性相关
- 同方差性残差具有零均值、同方差
- 无序列相关残差不序列相关

1.8 离群点和高杠杆点

- 离群点（Outlier）指对于给定的预测值 x_i 来说，响应值 y_i 异常的点。
- 高杠杆（high leverage）值表示观测值 x_i 是异常的，可以通过 Cook's D 统计量来甄别。

1.9 回归诊断的内容

- 正态性

```
library(car)
fit<-lm(Murder~Population+Illiteracy+
          Income+Frost, data = states)
qqPlot(fit,labels=row.names(states),
       id.method="identify",
       simulate=TRUE,
       main="Q-Q Plot")
```

- 独立性

```
durbinWatsonTest(fit)
```

- 线性

```
crPlots(fit)
```

- 同方差性

```
ncvTest(fit)
spreadLevelPlot(fit)
```

- 综合验证

```
library(gvlma)
gvmodel <- gvlma(fit)
summary(gvmodel)
```

- 多重共线性

是指解释变量之间存在多重共线性时的方差与不存在多重共线性时的方差之比。容忍度的倒数，VIF 越大，显示共线性越严重。经验判断方法表明：当 $0 < VIF < 10$，不存在多重共线性；当 $10VIF < 100$，存在较强的多重共线性；当 $VIF100$，存在严重多重共线性。

$$VIF = \frac{1}{1-(R_i)^2}$$

$(R_i)^2$ 指的是用第 i 个自变量与剩下所有自变量回归的 R^2

```
vif(fit)
```

- 离群点

```
outlierTest(fit)
```

- 高杠杠点

```
p<-length(coefficients(fit))
n<-length(fitted(fit))
plot(hatvalues(fit),main="Index Plot of Hat Values")
abline(h=c(2,3)*p/n,col="red",lty=2)
identify(1:n,hatvalues(fit),names(hatvalues(fit)))
```

- 强影响点

一般说来 Cook's D 值大于 $4/(n-k-1)$，则表示是强影响点

```
cutoff<-4/(nrow(states)-length(fit$coefficients)-2)
plot(fit,which=4,cook.levels = cutoff)
abline(h=cutoff,lty=2,col="red")
#
avPlots(fit,ask = FALSE,id.method="identify")

#
influencePlot(fit,id.method = "identify",
              main="Influence Plot",
              sub="Circle size is proportional
              to Cook's distance")
```