DISCUSSION # 8

Zifeng Zhao

Oct 24, 2014
Box-Cox Transformation (S-L Section 10.5.2)

- If the normal plot reveals evidence of nonnormality or heteroskedasticity, the standard remedy is to transform the response. A popular family of transformations is the Box-Cox transformation.

- Now our linear assumption has changed to $Y_i^{(\lambda)} = g(Y_i, \lambda) = x_i' \beta + \epsilon_i$, where

$$g(y, \lambda) = \begin{cases} 
\frac{(y^\lambda - 1)}{\lambda} & \text{if } \lambda \neq 0 \\
\log(y) & \text{if } \lambda = 0 
\end{cases}$$

- How to get the best $\lambda$? Use MLE
• Assume or pretend $\epsilon \sim iid N(0, 1)$, then the likelihood function for the original observations $Y$ is

$$(2\pi\sigma^2)^{-1/2n} \exp\{-\frac{1}{2\sigma^2} (y^{(\lambda)} - X\beta)'(y^{(\lambda)} - X\beta)\} |J|$$

, where $|J|$ is the absolute value of the determinant of the Jacobian matrix.

$|J| = \left| \prod_{i=1}^{n} \frac{dy_i^{(\lambda)}}{dy_i} \right| = \prod_{i=1}^{n} y_i^{\lambda-1}$.

• Note in Box-Cox transformation, we require $Y > 0$.

• There are also some other transformation families, see book for detail.
Box-Cox Transformation

```r
> library(MASS)
> x1=runif(100,0,10)
> x2=runif(100,0,10)
> y=10+x1+x2+rnorm(100,0,1)
> y1=sqrt(2*y+1)
> model=lm(y1~x1+x2)
> trans=boxcox(model,lambda = seq(1, 3, 1/10))
> trans$x[which.max(trans$y)]

[1] 1.888889
```
Box-Cox Transformation

> #plot(trans,type='l')
> boxcox(model,lambda = seq(1, 3, 1/10),plotit=T)
Construction of Simultaneous Confidence Intervals

- Under normality assumption, $\hat{\beta} \sim N(\beta, \sigma^2(X'X)^{-1})$, and $\hat{\beta} \perp \hat{\sigma}^2$, so we can construct t-statistic for each $\beta_i$.

- How to construct simultaneous confidence intervals?
  1. Bonferroni
  2. Scheffe’s Method
  3. Tukey’s Method
• Inclusion-Exclusion principle:

\[ P(\bigcup_{i=1}^{n} A_i) = \sum_{i=1}^{n} P(A_i) - \sum_{1 \leq i < j \leq n} P(A_i \cap A_j) + \ldots + (-1)^{n-1} P(\bigcap_{i=1}^{n} A_i) \]

• Proof: \( E(\prod_{i=1}^{n} (1 - 1_{A_i})) = 1 - P(\bigcup_{i=1}^{n} A_i) \)

• \( P(\bigcap_{i=1}^{n} A_i) = 1 - P(\bigcup_{i=1}^{n} A_i^c) \geq 1 - \sum_{i=1}^{n} P(A_i^c) \), set \( P(A_i^c) = \alpha/n \).
Scheffe’s Method

- By Theorem 4.1, we have
  
  \[(\hat{\beta} - \beta)'(A(X'X)^{-1}A')^{-1}(\hat{\beta} - \beta)/dS^2 \sim F_{d,n-p}, \text{ where } d \leq p, A\]
  
  is \(d \times p\) and is of full rank.

- Let \(L = (A(X'X)^{-1}A')^{-1}\), we have

  \[
  1 - \alpha = P((\hat{\beta} - \beta)'(A(X'X)^{-1}A')^{-1}(\hat{\beta} - \beta) \leq dS^2 F_{d,n-p})
  
  = P((\hat{\beta} - \beta)'L^{-1}(\hat{\beta} - \beta) \leq m)
  
  = P(b'L^{-1}b \leq m)
  
  = P[\sup_{h \neq 0} \frac{(h'b)^2}{h'Lh} \leq m]
  
  = P[\frac{(h'b)^2}{h'Lh} \leq m, \forall h \neq 0]
  
  = P[\frac{|h'\hat{\beta} - h'\beta|}{S(h'Lh)^{1/2}} \leq (dF^\alpha_{d,n-p})^{1/2}, \forall h \neq 0]
  
  \]
Scheffe’s Method

- Based on previous argument, we can construct a simultaneous confidence intervals using \( h' \hat{\beta} \pm (dF_{d,n-p}^\alpha)^{1/2} S(h' Lh)^{1/2} \)

- Confidence region:
  \[
  1 - \alpha = P((\hat{\beta} - \beta)'(A(X' X)^{-1} A')^{-1}(\hat{\beta} - \beta) \leq dS^2 F_{d,n-p})
  \]
Confidence Intervals

```r
> x1 = rnorm(100, 0, 1)
> x2 = rnorm(100, 0, 1)
> y = 1 + 0.5 * x1 - 0.5 * x2 + rnorm(100, 0, 1)
> model = lm(y ~ x1 + x2)
> confint(model)

2.5 % 97.5 %
(Intercept) 0.8940573 1.3060773
x1 0.2261570 0.6197188
x2 -0.5647728 -0.1607210
```
Bonferroni Confidence Intervals

```r
> library(car)
> X <- model.matrix(model)
> p <- dim(X)[2]
> confint(model, level=1-.05/p)

          0.833 % 99.167 %
(Intercept) 0.8471944 1.3529403
x1          0.1813935 0.6644824
x2         -0.6107295 -0.1147643
```
**Scheffe’s Confidence Ellipse**

```r
> library(faraway)
> temp <- confint(model, level = 1-0.05/p)
> confidenceEllipse(model, which.coef=c(1,2), Scheffe=TRUE)
> abline(v=temp[1,1], lty=2)
> abline(v=temp[1,2], lty=2)
> abline(h=temp[2,1], lty=2)
> abline(h=temp[2,2], lty=2)
```
Confidence Ellipse

```r
> library(faraway)
> temp <- confint(model, level = 1 - 0.05/p)
> confidenceEllipse(model, which.coef=c(2,3), Scheffe=TRUE)
> abline(v=temp[2, 1], lty=2)
> abline(v=temp[2, 2], lty=2)
> abline(h=temp[3, 1], lty=2)
> abline(h=temp[3, 2], lty=2)
```
**Dataset PAH**

- This data contains UV absorbance spectra obtained at 27 wavelengths from 25 different mixtures, each having varying concentrations of ten different polynuclear aromatic hydrocarbons (PAHs).
- The data were introduced in Brereton (2003) to illustrate the use of multivariate calibration techniques.
- Variables:
  - Spectrum: Identification number
  - x220-x350: Absorbances at wavelengths between 220 and 350 nm, AU (absorbance unit)
- There’s a more complicated theory on partial regression but we are going to ignore the covariates and focus only on the curves.

```r
> address <- "http://www.stat.wisc.edu/~zifeng/849/PAH.txt"
> data <- read.table(address, header=TRUE)
> absorbance <- as.numeric(data[1,12:38])  # First observation
> spectrum <- seq(220, 350, by=5)
```
**Polynomial regression**

```r
> plot(absorbance ~ spectrum, type="o")
> test4 <- lm(absorbance ~ poly(spectrum, 4))
> test8 <- lm(absorbance ~ poly(spectrum, 8))
> test16 <- lm(absorbance ~ poly(spectrum, 16))
> lines(spectrum, test4$fitted, col="Orange", lwd=2)
> lines(spectrum, test8$fitted, col="Green", lwd=2)
> lines(spectrum, test16$fitted, col="Blue", lwd=2)
```
POLYNOMIAL REGRESSION

![Graph showing absorbance vs. spectrum with polynomial regression lines.]

---

**Legend:**
- Orange line: Linear regression
- Blue line: Quadratic regression
- Green line: Cubic regression

---

**Axes:**
- X-axis: Spectrum
- Y-axis: Absorbance
The big question is: when you fit a 16-th degree polynomial, aren’t you incorporating noise as fluctuations in your data? `poly()` works up to degree 22. Let’s look at how it works.

```r
> r2=c(); ar2=c();
> for (i in 1:22){
+ myfit <- summary(lm(absorbance ~ poly(spectrum,i)))
+ r2[i] <- myfit$r.squared
+ ar2[i] <- myfit$adj.r.squared
+ }
> layout(matrix(1:2, ncol=2))
> plot(r2, type="b")
> plot(ar2, type="b")
```
$R^2$ AND ADJUSTED $R^2$
**$R^2$ AND ADJUSTED $R^2$**

- 

\[
R^2 = 1 - \frac{RSS}{SST}
\]

\[
\text{Adj} R^2 = 1 - (1 - R^2) \frac{n}{n - p} = 1 - \frac{RSS/(n - p)}{SST/(n)}
\]

- Adj $R^2$ will increase iff the corresponding *F-test* between $RSS_p$ and $RSS_{p+1}$ is bigger than 1.

- Note that the tail behavior of the adjusted $R^2$ is unusual due to the fact we’re doing a polynomial regression. Usually, as $p \to n$, the adjusted $R^2$ decreases.

- In this case, however, the fitting is approaching an interpolation, so the residual sum of squares approaches zero.

- Adj $R^2$ doesn’t work here, what else variable selection criteria we have? AIC, BIC, Mallow’ Cp etc.