Solutions for ST340 Lab 6

2019–20

Validation

The dataset `SmokeCancer.csv` shows lung cancer rates by U.S. state in 2010, with a number of covariates such as Federal Year 2010 cigarette sales per 100,000.

(a) Read the data file on lung cancer and create a data frame with variables of interest.

```
X = read.table("SmokeCancer.csv", header=TRUE,sep=",",row.names=1)
LungCancer = data.frame(CigSalesRate=100000*X[,"FY2010Sales"]/X[,"Pop2010"],
                        X[,c("CigYouthRate","CigAdultRate","LungCancerRate")])
```

(b) Fit a linear model for LungCancerRate (`?lm` for a reminder about `lm`):

```
summary(lm(LungCancerRate~CigSalesRate+CigYouthRate+CigAdultRate,data=LungCancer))
```

```
##
## Call:
## lm(formula = LungCancerRate ~ CigSalesRate + CigYouthRate + CigAdultRate,
##     data = LungCancer)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.8367  -3.1471   0.9165   3.4358   9.9253
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   15.5139     5.1039   3.040  0.00386 **
## CigSalesRate   1.2641     0.5241   2.412  0.01983 *
## CigYouthRate  -0.3191     0.2831  -1.127  0.26545
## CigAdultRate   1.9013     0.3575   5.318 2.85e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.323 on 47 degrees of freedom
## Multiple R-squared:  0.6883, Adjusted R-squared:  0.6684
## F-statistic: 34.59 on 3 and 47 DF,  p-value: 5.907e-12
```

(c) Write a function that takes a formula and does LOOCV (leave one out cross validation) with respect to the squared error of the linear model for the given formula. Use it to find a good linear model for `LungCancerRate` in terms of `CigSalesRate`, `CigYouthRate` and `CigAdultRate`. You could also try using transformations of the covariates by adding terms such as `I(CigSalesRate^2)` and `I(CigSalesRate*CigAdultRate)` to your formulae.

(By good, we mean that it is the optimal, in terms of cross-validation error, linear model using some or all of these covariates.)

```
loocv<-function(formula) {
  s=0
  for (i in 1:dim(LungCancer)[1]) {
    l=lm(formula,LungCancer[-i,])
    s=s+(predict(l,LungCancer[i,])-LungCancer$LungCancerRate[i])^2
```

```r
  }
  s/dim(LungCancer)[1]
}

loocv("LungCancerRate~CigSalesRate+             CigAdultRate")
```

```
##       AL
## 30.43823
```

```r
loocv("LungCancerRate~CigSalesRate+CigAdultRate             +I(CigSalesRate^2)")
```

```
##       AL
## 31.92415
```

```r
loocv("LungCancerRate~             CigAdultRate             ")
```

```
##       AL
## 32.19693
```

```r
loocv("LungCancerRate~CigSalesRate+CigYouthRate+CigAdultRate")
```

```
##       AL
## 32.24044
```

```r
loocv("LungCancerRate~CigSalesRate+CigAdultRate             +I(CigSalesRate*CigAdultRate)")
```

```
##      AL
## 32.9402
```

```r
loocv("LungCancerRate~             CigYouthRate+CigAdultRate")
```

```
##       AL
## 34.33167
```

```r
loocv("LungCancerRate~CigSalesRate+CigAdultRate             +I(CigAdultRate^2)")
```

```
##       AL
## 36.71572
```

```r
loocv("LungCancerRate~CigSalesRate                          ")
```

```
##       AL
## 46.12886
```

```r
loocv("LungCancerRate~CigSalesRate+CigYouthRate             ")
```

```
##      AL
## 50.2469
```

(d) The Akaike Information criterion (AIC) and Bayesian Information criterion (BIC) are analytic approximations to the validation step. They are (different) ways of quantifying the trade-off between model complexity (in terms of, e.g. the number of parameters) and the fit to the training data (in terms of likelihood), defined as follows:

- Akaike Information criterion (AIC) = $(2 \times \#\text{parameters} - 2 \times \log(\text{likelihood}))$, and
- Bayesian information criterion (BIC) = $(\log(\text{amount of data}) \times \#\text{parameters} - 2 \times \log(\text{likelihood}))$

Write a function that takes a formula and then calculates AIC and BIC. Use your function to find a *good* linear model for `LungCancerRate`, as in (b).

```r
aic<-function(formula) {
  AIC(lm(formula,data=LungCancer))
#   #Equivalent to
#   l=lm(formula,data=LungCancer)
#   p=length(l$coefficients)+1
#   logLik(l)
#   2*p-2*logLik(l) # or:
#   2*p-2*sum(log(dnorm(l$residuals,sd=summary(l)$sigma)))
}

bic<-function(formula) {
  BIC(lm(formula,data=LungCancer))
}

aic("LungCancerRate~CigSalesRate+CigAdultRate             ")
```

```
## [1] 320.4682
```

```r
aic("LungCancerRate~CigSalesRate+CigYouthRate+CigAdultRate")
```

```
## [1] 321.1082
```

```r
aic("LungCancerRate~CigSalesRate+CigAdultRate             +I(CigAdultRate^2)")
```

```
## [1] 321.5652
```

```r
aic("LungCancerRate~CigSalesRate+CigAdultRate             +I(CigSalesRate^2)")
```

```
## [1] 322.3432
```

```r
aic("LungCancerRate~CigSalesRate+CigAdultRate             +I(CigSalesRate*CigAdultRate)")
```

```
## [1] 322.2933
```

```r
aic("LungCancerRate~             CigAdultRate             ")
```

```
## [1] 323.2598
```

```r
aic("LungCancerRate~             CigYouthRate+CigAdultRate")
```

```
## [1] 325.0596
```

```r
aic("LungCancerRate~CigSalesRate                          ")
```

```
## [1] 341.5566
```

```r
aic("LungCancerRate~CigSalesRate+CigYouthRate             ")
```

```
## [1] 343.1325
```

```r
bic("LungCancerRate~CigSalesRate+             CigAdultRate")
```

```
## [1] 328.1955
```

```r
bic("LungCancerRate~CigSalesRate+CigYouthRate+CigAdultRate")
```

```
## [1] 330.7673
```

```r
bic("LungCancerRate~             CigYouthRate+CigAdultRate")
```

```
## [1] 332.7869
```

```
bic("LungCancerRate~              CigAdultRate              ")
```

```
## [1] 329.0553
```

```
bic("LungCancerRate~CigSalesRate                            ")
```

```
## [1] 347.3521
```

```
bic("LungCancerRate~CigSalesRate+CigYouthRate               ")
```

```
## [1] 350.8598
```

The curse of dimensionality

Suppose N points are chosen uniformly at random in the D-dimensional hypercube $[0,1]^D$. Consider a smaller hypercube $H = [0,r]^D$ in the "corner" of $[0,1]^D$.

(a) How big does r have to be for there to be approximately one of the N points lying in H?

$(1/N)^{1/D}$.

(b) How big does r have to be for there to be approximately 10 of the N points lying in H?

$(10/N)^{1/D}$.

(c) How big does r have to be for there to be approximately $\frac{N}{2}$ of the N points lying in H?

$(1/2)^{1/D}$ which is approximately 1 for large D.

Check each of your answers by simulation.

```
a1 = vector(); a2 = vector(); a3 = vector()
N = 10000
for (D in 1:10) {
  p = matrix(runif(N*D),nrow = N, ncol = D)
  r1 = (1/N)^(1/D)
  r2 = (10/N)^(1/D)
  r3 = (1/2)^(1/D)
  a1[D] = sum(rowSums(p < r1) == D) # Should average 1
  a2[D] = sum(rowSums(p < r2) == D) # Should average 10
  a3[D] = sum(rowSums(p < r3) == D) # Should average N/2
}
a1
```

```
##  [1] 3 3 2 1 0 1 0 1 2 1
```

```
a2
```

```
##  [1]  9 12 18 12  6 17  4  8  9 11
```

```
a3
```

```
##  [1] 4977 4934 5033 5038 4925 5017 5016 4993 4925 5003
```

Distance functions

(a) Write a function to calculate the ℓ_1 distances between pairs of row vectors in two matrices:

```r
distances.l1 <- function(x,y) {
  apply(y,1,function(p) apply(x,1,function(q) sum(abs(p-q))))
}
```

(b) Write a similar function to calculate a matrix of pairwise ℓ_2 distances:

```r
distances.l2 <- function(x,y)
  apply(y,1,function(p) apply(x,1,function(q) sqrt(sum((p-q)^2))))
```

(c) Write a similar function to calculate the Mahalanobis distance between the row vectors, given a $D \times D$ covariance matrix S:

```r
distances.maha <-function(x,y) {
  C=cov(x)
  C.inv=solve(C)
  apply(y,1,function(p) apply(x,1,function(q) sqrt( (p-q) %*% C.inv %*% (p-q) )))
}
```