Multivariate Statistical Methods

Assignment 4 Canonical correlation analysis

Ahmet Hakan Akdeve(ahmak554), Jooyoung Lee(joole336), Weng Hang Wong(wonwo535), Zhixuan Duan(zhidu838)

2019-12-13

Question: Canonical correlation analysis

a)

Codes as shown below.

```
# import the dataset
bbb <- read.table('/Users/darin/Desktop/multi-var/P10-16.DAT')
bbb <- as.matrix(bbb)

# calculate the corr matrix
aaa <- matrix(0,5,5)

for(i in 1:nrow(bbb))
{
  for(j in 1:ncol(bbb))
  {
    aaa[i,j]<-bbb[i,j]/(sqrt(bbb[i,i])*sqrt(bbb[j,j]))
  }
}

# specific four matrix
s11 <- aaa[1:3,1:3]
s12 <- aaa[1:3,4:5]
s22 <- aaa[4:5,4:5]
s21 <- aaa[4:5,1:3]

# produce the eigenvalues and eigenvectorsof s11
eigen_x <- eigen(s11)

# produce the eigenvalues and eigenvectorsof s22
eigen_y <- eigen(s22)

#  diagonalization
sqrex <- (eigen_x$vectors)%*%diag(eigen_x$values^-.5)%*%solve(eigen_x$vectors)
sqrey <- (eigen_y$vectors)%*%diag(eigen_y$values^-.5)%*%solve(eigen_y$vectors)

# estimate the eigenvalues and eigenvectors of sqrex and sqrey
m1 <- sqrex%*%s12%*%solve(s22)%*%t(s12)%*%sqrex
m2 <- sqrey%*%t(s12)%*%solve(s11)%*%(s12)%*%sqrey
m1e <- eigen(m1)
m2e <- eigen(m2)


# get a and b
```

```
a <- sqrex%*%m1e$vectors
b <- sqrey%*%m2e$vectors

sqrt(0.26764579)
```

Correct.

```
## [1] 0.5173449
```

```
sqrt(0.01575231)
```

```
## [1] 0.1255082
```

We find two non-zero eigenvectors: The highest eigenvalue is 0.268, that is the canonical correlation is 0.518. The second eigenvalue is 0.016, that is the canonical correlation is 0.126.

b)

```
corcoef.test<-function(r, n, p, q, alpha=0.05){
  m<-length(r)
  Q<-rep(0, m)
  lambda <- 1

  for (k in m:1){
    lambda<-lambda*(1-r[k]^2);
    Q[k]<- -log(lambda)
  }

  s<-0
  i<-m

  for (k in 1:m){
    Q[k]<- (n-k+1-1/2*(p+q+3)+s)*Q[k]
    chi<-1-pchisq(Q[k],(p-k+1)*(q-k+1))
    if (chi>alpha){
      i<-k-1
      break
    }
    s<-s+1/r[k]^2
  }
  return(i)
}

corcoef.test(r=sqrt(m2e$values),n=46,p=3,q=2)
```

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

From the result, we choose the first pair of canonical correlation.

c)

The "significant" squared canonical correlation is 0.268 from the result above, which means the correlation between U and V is 0.268, they groups the variables in such way that the correlation between them is maximized.

d)

We use the first eigenvectors as weights for the first canonical correlation (X_1 means glucose intolerance, X_2 means insulin response to oral glucose, X_3 means insulin resistance, X_4 means relative weight, X_5 means fasting plasma glucose):

$$U = 0.436 * X_1 - 0.705 * X_2 + 1.082 * X_3$$

$$V = -1.020 * X_4 + 0.161 * X_5$$

These coefficients should be multipled by a negative sign. Note that each eigen value of a matrix corresponds to infinitely many eigen vectors, and given an arbitrary eigen vector, the negative of it is still an eigen vector. So when picking an eigen vector for CCA, it is your responsibility to pick an eigen vector that makes U and V positively correlated. Your choice of this particular eigen vector makes U and V negatively correlated.

e)

The canonical correlation coefficient cannot explain how much variance each root contribute to the variables, but by checking the significancy of each variable weights in canonical variate, we can assume that this pair of canonical correlated factor is a good summary of the whole dataset.

f)

There are a few things worth noting, the first is that the correlation between the two sets of variables is reflected by the linear combination of the variables. Second, the Lagrangian equation is used to solve the maximum problem. Third, the canonical correlated variables cannot directly explain the variables.

I guess you can use that to prove that the eigen values/vectors maximises the correlations between U and V, but there may well be other ways of proving that.

Appendix

```
knitr::opts_chunk$set(echo = FALSE,
                      warning = FALSE,
                      message = FALSE)
# import the dataset
bbb <- read.table('/Users/darin/Desktop/multi-var/P10-16.DAT')
bbb <- as.matrix(bbb)

# calculate the corr matrix
aaa <- matrix(0,5,5)

for(i in 1:nrow(bbb))
{
  for(j in 1:ncol(bbb))
  {
    aaa[i,j]<-bbb[i,j]/(sqrt(bbb[i,i])*sqrt(bbb[j,j]))
  }
}

# specific four matrix
s11 <- aaa[1:3,1:3]
s12 <- aaa[1:3,4:5]
s22 <- aaa[4:5,4:5]
s21 <- aaa[4:5,1:3]

# produce the eigenvalues and eigenvectorsof s11
eigen_x <- eigen(s11)

# produce the eigenvalues and eigenvectorsof s22
eigen_y <- eigen(s22)

#  diagonalization
sqrex <- (eigen_x$vectors)%*%diag(eigen_x$values^-.5)%*%solve(eigen_x$vectors)
```

```r
sqrey <- (eigen_y$vectors)%*%diag(eigen_y$values^-.5)%*%solve(eigen_y$vectors)

# estimate the eigenvalues and eigenvectors of sqrex and sqrey
m1 <- sqrex%*%s12%*%solve(s22)%*%t(s12)%*%sqrex
m2 <- sqrey%*%t(s12)%*%solve(s11)%*%(s12)%*%sqrey
m1e <- eigen(m1)
m2e <- eigen(m2)


# get a and b
a <- sqrex%*%m1e$vectors
b <- sqrey%*%m2e$vectors

sqrt(0.26764579)
sqrt(0.01575231)
corcoef.test<-function(r, n, p, q, alpha=0.05){
  m<-length(r)
  Q<-rep(0, m)
  lambda <- 1

  for (k in m:1){
    lambda<-lambda*(1-r[k]^2);
    Q[k]<- -log(lambda)
  }

  s<-0
  i<-m

  for (k in 1:m){
    Q[k]<- (n-k+1-1/2*(p+q+3)+s)*Q[k]
    chi<-1-pchisq(Q[k],(p-k+1)*(q-k+1))
    if (chi>alpha){
      i<-k-1
      break
    }
    s<-s+1/r[k]^2
  }
  return(i)
}

corcoef.test(r=sqrt(m2e$values),n=46,p=3,q=2)
```