Canonical Correlations

Goals: Learn how to perform canonical correlations analysis

Assignments:
Use the data set `Lab3_close.csv` from the course web site; it consists of daily closing prices for ten stocks. (Alternatively, you can use a data set of your choice.) The goal of this assignment is to find the best possible pair of “duplicate portfolios,” that is two groups of non-overlapping stocks with the maximal possible first canonical correlation.

Reports: Printed reports are due on Tuesday, May 8, 2018

Report preparation: Consider each report as a mini-paper. It should NOT be long, but it should provide a reader with all background information about the data, problem, and methods you are using. Review the necessary theoretical material, describe the data. Do not insert the R-output in your report; instead, summarize it in tables or text in a nice readable form. If you feel some parts of the output should be included, put them in Appendix. Put your name on the title page. Illustrations should support your conclusions and make it easier to read a report.
# Install libraries ...
library(MASS)  # ... for Multivariate Normal Distribution
library(car)   # ... for ellipse plots

# Example 1: X=N1+N2+2*N3+2*N4
Sigma=diag(c(1,1,1,1))
N<-mvrnorm(n=1000,c(0,0,0,0),Sigma)
a=as.matrix(c(1,1,2,2))
X<-N%*%a
cor(X,N)
c<-=cancor(X,N)
# Notice four combinations in Y and only one significant correlation

X1<-X%*%c$xcoef[,1]
Y<-N%*%c$ycoef[,3]
cor(X1,Y)
SA(cbind(X1,Y))

# Example 2: X1=4*N1+3*N2+2*N3+N4, X2=4*N1+3*N2+2*N3+N4
Sigma=diag(c(1,1,1,1))
N<-mvrnorm(n=1000,c(0,0,0,0),Sigma)
a=matrix(c(1,2,3,4,4,3,2,1),4,2)
X<-N%*%a
Y<-N[,1:4]
cor(X,Y)
cc<-=cancor(X,Y)
cc
X1<-X%*%cc$xcoef[,1]
Y1<-Y%*%cc$ycoef[,1]
X2<-X%*%cc$xcoef[,2]
Y2<-Y%*%cc$ycoef[,2]
```r
SA(cbind(X1,Y1))
SA(cbind(X2,Y2))

# Real data example
# (Phychological type vs academic performance)
# For data description and analysis detail, see
# https://stats.idre.ucla.edu/rdae/canonical-correlation-analysis

mm <- read.csv("https://stats.idre.ucla.edu/stat/data/mmreg.csv")
colnames(mm) <- c("Control", "Concept", "Motivation", "Read", "Write", "Math", "Science", "Sex")

psyc <- ss[,c(1,2,3)]
test <- ss[,c(4,5,6,7)]
I = Sex == 0

c <- cancor(psyc[I,], test[I,])
c$xcoef / c$xcoef[1,1]
c$ycoef / c$ycoef[1,1]

X <- psyc * * c$xcoef[,1]
Y <- test * * c$ycoef[,1]
SA(cbind(X,Y))

# Real data example
# (Closing prices on 10 stocks)

# read the data table
T <- read.table("Lab3_close.csv", sep=',', header=TRUE)
len <- dim(T)[1] # length
Itime <- seq(len,1,by=-1) # inverse index for plots

time <- 1989 + (31+28+31+30+4)/365.25 + seq(1,len)/250

names(T) # names of variables
P <- T[,seq(2,11)] # remove the dates
P <- log10(P)

# Remove long-term trend

PD <- P * 0
```
for (i in seq(1,10))
{
t<-ts(P[,i], frequency=365)
s<-stl(t,s.window=250)
PD[,i]=t-s$time.series[,2]
}
X<-PD[,c(1,2,3,4,5)]
Y<-PD[,c(6,7,8,9,10)]
cor(X,Y)
cancor(X,Y)

#===================================================
# Function that illustrates spectral decomposition
# and statistical distance ellipses
#===================================================

SA <- function(X,add=FALSE,data.plot=TRUE)
{
  # Vector of means
  #================================
  n<dim(X)[1]
one<-matrix(rep(1,n),ncol=1)
mu<-as.vector(t(X) %*% ones / n)
  # Variance
  #================================
  Sigma<-var(X)
e<-eigen(Sigma)
par(bg='yellow')
ellipse(mu,Sigma,3,add=add,xlim=range(X),ylim=range(X))
ellipse(mu,Sigma,2,add=TRUE)
ellipse(mu,Sigma,1,add=TRUE)
if (data.plot)
  points(X[,1],X[,2],pch=20,col=4)
  arrows(mu[1],mu[2],mu[1]+e$vectors[1,1]*sqrt(e$values[1]),
         mu[2]+e$vectors[2,1]*sqrt(e$values[1]),length=.1,col='green',lwd=2)
  arrows(mu[1],mu[2],mu[1]+e$vectors[1,2]*sqrt(e$values[2]),
         mu[2]+e$vectors[2,2]*sqrt(e$values[2]),length=.1,col='green',lwd=2)
e}
ellipse <-
function (mu, Sigma, R, col = 'red', add = FALSE, xlim = NULL, ylim = NULL, N = 1000)
{
    # Find coordinates of a circle
    t <- seq(0, 2*pi, length.out = N)
    x <- R*cos(t)
    y <- R*sin(t)

    # Spectral decomposition of Sigma
    e <- eigen(Sigma)  # spectral decomposition
    P <- e$vectors    # eigenvectors
    L <- e$values

    # Square root matrix
    S05 <- P%*%sqrt(diag(L))%*%t(P)

    # Ellipse cordinates
    vec <- cbind(x, y)
    vec <- t(vec%*%S05)
    x <- vec[1,] + mu[1]
    y <- vec[2,] + mu[2]

    if (add)
    {
        points(x, y, type = 'l', col = col)
    }
    else
    {
        plot(x, y, type = 'l', col = col, xlim = xlim, ylim = ylim)
    }
}