PCA and SVD

2023-01-24

#To reduce the dimension using Principal Component Analysis  
#and Singular Value Decompositionsa  
#iris  
data = iris[,1:4]  
data\_scaled = scale(data, center = T,scale = T)  
head(data\_scaled)

## Sepal.Length Sepal.Width Petal.Length Petal.Width  
## [1,] -0.8976739 1.01560199 -1.335752 -1.311052  
## [2,] -1.1392005 -0.13153881 -1.335752 -1.311052  
## [3,] -1.3807271 0.32731751 -1.392399 -1.311052  
## [4,] -1.5014904 0.09788935 -1.279104 -1.311052  
## [5,] -1.0184372 1.24503015 -1.335752 -1.311052  
## [6,] -0.5353840 1.93331463 -1.165809 -1.048667

cov\_mat = cov(data\_scaled); cov\_mat

## Sepal.Length Sepal.Width Petal.Length Petal.Width  
## Sepal.Length 1.0000000 -0.1175698 0.8717538 0.8179411  
## Sepal.Width -0.1175698 1.0000000 -0.4284401 -0.3661259  
## Petal.Length 0.8717538 -0.4284401 1.0000000 0.9628654  
## Petal.Width 0.8179411 -0.3661259 0.9628654 1.0000000

eig\_decomp = eigen(cov\_mat)  
eig = eig\_decomp$values ; eig

## [1] 2.91849782 0.91403047 0.14675688 0.02071484

eigvect = eig\_decomp$vectors ; eigvect

## [,1] [,2] [,3] [,4]  
## [1,] 0.5210659 -0.37741762 0.7195664 0.2612863  
## [2,] -0.2693474 -0.92329566 -0.2443818 -0.1235096  
## [3,] 0.5804131 -0.02449161 -0.1421264 -0.8014492  
## [4,] 0.5648565 -0.06694199 -0.6342727 0.5235971

proj\_data = cov\_mat %\*% eigvect  
proj\_data

## [,1] [,2] [,3] [,4]  
## Sepal.Length 1.5207297 -0.34497120 0.10560131 0.005412503  
## Sepal.Width -0.7860899 -0.84392037 -0.03586471 -0.002558482  
## Petal.Length 1.6939344 -0.02238608 -0.02085802 -0.016601890  
## Petal.Width 1.6485326 -0.06118702 -0.09308389 0.010846229

var1 = eig[1]/sum(eig) ;var1

## [1] 0.7296245

var2 = eig[2]/sum(eig) ;var2

## [1] 0.2285076

var3 = eig[3]/sum(eig) ;var3

## [1] 0.03668922

var4 = eig[4]/sum(eig) ;var4

## [1] 0.005178709

for(x in eig) print(x/sum(eig))

## [1] 0.7296245  
## [1] 0.2285076  
## [1] 0.03668922  
## [1] 0.005178709

data\_new = data\_scaled %\*% eigvect[,1:2]; head(data\_new)

## [,1] [,2]  
## [1,] -2.257141 -0.4784238  
## [2,] -2.074013 0.6718827  
## [3,] -2.356335 0.3407664  
## [4,] -2.291707 0.5953999  
## [5,] -2.381863 -0.6446757  
## [6,] -2.068701 -1.4842053

#using built-in function  
iris\_pca = prcomp(iris[,1:4], center = T, scale = T, rank = 2)  
iris\_pca

## Standard deviations (1, .., p=4):  
## [1] 1.7083611 0.9560494 0.3830886 0.1439265  
##   
## Rotation (n x k) = (4 x 2):  
## PC1 PC2  
## Sepal.Length 0.5210659 -0.37741762  
## Sepal.Width -0.2693474 -0.92329566  
## Petal.Length 0.5804131 -0.02449161  
## Petal.Width 0.5648565 -0.06694199

summary(iris\_pca)

## Importance of first k=2 (out of 4) components:  
## PC1 PC2  
## Standard deviation 1.7084 0.9560  
## Proportion of Variance 0.7296 0.2285  
## Cumulative Proportion 0.7296 0.9581

head(iris\_pca$x)

## PC1 PC2  
## [1,] -2.257141 -0.4784238  
## [2,] -2.074013 0.6718827  
## [3,] -2.356335 0.3407664  
## [4,] -2.291707 0.5953999  
## [5,] -2.381863 -0.6446757  
## [6,] -2.068701 -1.4842053

#SVD  
A = cbind(c(2,1,3), c(1,3,2))  
A

## [,1] [,2]  
## [1,] 2 1  
## [2,] 1 3  
## [3,] 3 2

A\_svd = svd(A); A\_svd

## $d  
## [1] 5.000000 1.732051  
##   
## $u  
## [,1] [,2]  
## [1,] -0.4242641 0.4082483  
## [2,] -0.5656854 -0.8164966  
## [3,] -0.7071068 0.4082483  
##   
## $v  
## [,1] [,2]  
## [1,] -0.7071068 0.7071068  
## [2,] -y0.7071068 -0.7071068

A\_svd$u %\*% diag(A\_svd$d) %\*% t(A\_svd$v)

## [,1] [,2]  
## [1,] 2 1  
## [2,] 1 3  
## [3,] 3 2

#Detailed SVD  
Au =A\_svd$u; Au

## [,1] [,2]  
## [1,] -0.4242641 0.4082483  
## [2,] -0.5656854 -0.8164966  
## [3,] -0.7071068 0.4082483

Bu = as.matrix(Au[,1]); Bu

## [,1]  
## [1,] -0.4242641  
## [2,] -0.5656854  
## [3,] -0.7071068

Ad = diag(A\_svd$d);Ad

## [,1] [,2]  
## [1,] 5 0.000000  
## [2,] 0 1.732051

Bd = as.matrix(Ad[1,1]); Bd

## [,1]  
## [1,] 5

Av = A\_svd$v; Av

## [,1] [,2]  
## [1,] -0.7071068 0.7071068  
## [2,] -0.7071068 -0.7071068

AvT = t(Av); AvT

## [,1] [,2]  
## [1,] -0.7071068 -0.7071068  
## [2,] 0.7071068 -0.7071068

BvT = as.matrix(AvT[1,]); BvT

## [,1]  
## [1,] -0.7071068  
## [2,] -0.7071068

B = Bu %\*% Bd %\*% t(BvT); B

## [,1] [,2]  
## [1,] 1.5 1.5  
## [2,] 2.0 2.0  
## [3,] 2.5 2.5

#forbenius norm  
f = sqrt(sum((A-B)^2));f

## [1] 1.732051