

Econometrics (I) homework 02

張文誠
m136040030

October 2024

```
#set matrix
A = matrix(c(1,2,3,
             2,9,6,
             3,6,7),3,3)

#find eigenvalues of A
eigen_A = eigen(A)
lambda1 = eigen_A$values[1] #第1個特徵值
lambda2 = eigen_A$values[2] #第2個特徵值
lambda3 = eigen_A$values[3] #第3個特徵值
cat(lambda1,lambda2,lambda3) #顯示三個特徵值
14.94834 2.337817 -0.286152

#find eigenvectors of A
eigenvtA = eigen(A)$vectors #所有特徵向量的矩陣
eigenvt1 = eigen(A)$vectors[,1] #第1個特徵向量
eigenvt2 = eigen(A)$vectors[,2] #第2個特徵向量
eigenvt3 = eigen(A)$vectors[,3] #第3個特徵向量
print(eigenvt1)
[1] -0.2423047 -0.7281726 -0.6411342
print(eigenvt2)
[1] 0.3863702 -0.6785878 0.6246893
print(eigenvt3)
[1] 0.88994748 0.09634994 -0.44576919

#verify tr(A)=sum(lambda)
tr_A = sum(diag(A)) #計算矩陣A的對角線元素之和
sum_lambda = lambda1+lambda2+lambda3 #計算三個特徵值之和
if (round(tr_A) == round(sum_lambda)) {
  print("tr(A) equals sum of eigenvalues when rounded.")
} else {
  print("tr(A) does not equal sum of eigenvalues when rounded.")
}
```

```

[1] "tr(A) equals sum of eigenvalues when rounded."

#verify det(A)=pi(lambda)
det_A = det(A) #計算A的行列式
pi_lambda = lambda1*%*%lambda2*%*%lambda3 #計算特徵值的乘積
if (round(det_A) == round(pi_lambda)) {
  print("det_A equals pi_lambda when rounded.")
} else {
  print("det_A does not equal pi_lambda when rounded.")
}
[1] "det_A equals pi_lambda when rounded."

#verify eigenvectors are orthonormal
round(t(eigenvta)%*%eigenvta) #檢查特徵向量矩陣的轉置與其自身的乘積，應為單位矩陣

      [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1

#verify A is full rank
rank_A = rankMatrix(A) #計算矩陣A的rank
print(rank_A)
[1] 3

# eigenvalues of M
n = 1200
I = diag(1, n) #單位矩陣
i = matrix(1, nrow = n, ncol = 1) #全為1的列向量
iT = t(i) #i的轉置
M1 = iT %*% i #(i*i)
M = I - i %*% solve(M1) %*% iT # M = I - i(i'i)^(-1)i'
eigen_M = eigen(M) #計算矩陣M的特徵值和特徵向量
eigenvalues_M <- eigen_M$values #提取矩陣M的特徵值
num_ones <- sum(round(eigenvalues_M) == 1) #計算特徵值中有多少個等於1
cat("Number of eigenvalues equal to 1:", num_ones)
Number of eigenvalues equal to 1: 1199

```