Inverse of a Matrix & Singular Value Decomposition IS605

Sharon Morris

2/21/2017

Problem Set 1 In this problem, we'll verify using R that SVD and Eigenvalues are related as worked out in the weekly module. Given a 3×2 matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ -1 & 0 & 4 \end{bmatrix}$$

Write code in R to compute $X = AA^T$ and $Y = A^T A$. Then, compute the eigenvalues and eigenvectors of $X and Y$ using the built-in commans in R. Then, compute the left-singular, singular values, and right-singular vectors of A using the svd command. Examine the two sets of singular vectors and show that they are indeed eigenvectors of X and Y. In addition, the two non-zero eigenvalues (the 3rd value will be very close to zero, if not zero) of both X and Y are the same and are squares of the non-zero singular values of A.

Your code should compute all these vectors and scalars and store them in variables. Please add enough comments in your code to show me how to interpret your steps. https://www.r-bloggers.com/quick-review-of-matrix-algebra-in-r/

```
A <- matrix(c(1,2,3,-1,0,4), nrow = 2, byrow = TRUE)
A
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]   -1    0    4
```

```
X <- A%*%t(A) #Compute X=AA^T
X
```

```
##      [,1] [,2]
## [1,]   14   11
## [2,]   11   17
```

```
Y <- t(A)%*%A #Compute Y=A^TA
Y
```

```
##      [,1] [,2] [,3]
## [1,]    2    2   -1
## [2,]    2    4    6
## [3,]   -1    6   25
```

```
# Compute eigenvalues, eigenvectors of X and Y
eignX <- eigen(X)
eignX
```

```
## $values
## [1] 26.601802  4.398198
##
## $vectors
##           [,1]       [,2]
## [1,] 0.6576043 -0.7533635
## [2,] 0.7533635  0.6576043
```

```r
eignY <- eigen(Y)
eignY
```

```
## $values
## [1] 2.660180e+01 4.398198e+00 1.058982e-16
##
## $vectors
##             [,1]       [,2]       [,3]
## [1,] -0.01856629 -0.6727903  0.7396003
## [2,]  0.25499937 -0.7184510 -0.6471502
## [3,]  0.96676296  0.1765824  0.1849001
```

```r
# Compute singular vectors
svdA <- svd(A)
(svdA$d)^2
```

```
## [1] 26.601802  4.398198
```

```r
eignX$vectors  # Compute left singular vector U to the eigen vectors of X
```

```
##           [,1]       [,2]
## [1,] 0.6576043 -0.7533635
## [2,] 0.7533635  0.6576043
```

```r
svdA$u
```

```
##            [,1]       [,2]
## [1,] -0.6576043 -0.7533635
## [2,] -0.7533635  0.6576043
```

```r
round(abs(eignX$vectors)) == round(abs(svdA$u))
```

```
##      [,1] [,2]
## [1,] TRUE TRUE
## [2,] TRUE TRUE
```

```r
eignY$vectors #eigenvectors of Y with the right singular vectors
```

```
##             [,1]       [,2]       [,3]
## [1,] -0.01856629 -0.6727903  0.7396003
## [2,]  0.25499937 -0.7184510 -0.6471502
## [3,]  0.96676296  0.1765824  0.1849001
```

```r
svdA$v
```

```
##             [,1]       [,2]
## [1,]  0.01856629 -0.6727903
## [2,] -0.25499937 -0.7184510
## [3,] -0.96676296  0.1765824
```

```r
round(abs(eignY$vectors[,-3])) == round(abs(svdA$v))
```

```
##      [,1] [,2]
## [1,] TRUE TRUE
## [2,] TRUE TRUE
## [3,] TRUE TRUE
```

```r
eignX$values   #compare the eigenvalues and the squares of singular values
```

```
## [1] 26.601802  4.398198
```

```r
eignY$values
```

```
## [1] 2.660180e+01 4.398198e+00 1.058982e-16
```

```r
svdA$d^2
```

```
## [1] 26.601802  4.398198
```

```r
round(eignX$values) == round(svdA$d^2)
```

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

```r
round(eignY$values[-3]) == round(svdA$d^2)
```

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

Problem Set 2 Using the procedure outlined in section 1 of the weekly handout, write a function to compute the inverse of a well-conditioned full-rank square matrix using co-factors. In order to compute the co-factors, you may use built-in commands to compute the determinant. Your function should have the following signature: $B = myinverse(A)$ where A is a matrix and B is its inverse and $A\ddot{O}B = I$. The off-diagonal elements of I should be close to zero, if not zero. Likewise, the diagonal elements should be close to 1, if not 1. Small numerical precision errors are acceptable but the function myinverse should be correct and must use co-factors and determinant of A to compute the inverse.

https://rpubs.com/aaronsc32/inverse-matrices

```r
#Matrix A.
myinverse <- function(A) {

  if (det(A) == 0) {          #if the determinant is 0 the matrix is not invertable
    stop('No inverse')
  }

  m <- nrow(A)
  n <- ncol(A)
  C <- diag(0, nrow = m, ncol = n)

  #compute the cofactors
  #- loop thru m and n of matrix A, and remove the current row/col, and calc the determinant
  for(i in 1:m)
  {
    for(j in 1:n)
    {
        C[i, j] <- (-1)^(i+j) * det(A[-i, -j])
    }
  }

  return((t(C) / det(A)))  #Inverse of A = t(C)/det(A)
}

set.seed(40)
A <- matrix(sample(10:20, 9, replace=TRUE),byrow=T, nrow =3)

B <- myinverse(A)

all.equal(solve(A), B)  # R inverse function
```

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

```r
I <- round(A %*% B, 10); I
```

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