HW4

Raphael Nash

2/21/2017

Problem Set 1

1. 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 $\mathbf{A}$

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

write code in R to compute $\mathbf{X} = \mathbf{A}\mathbf{A}^{\mathbf{T}}$ and $\mathbf{Y} = \mathbf{A}^{\mathbf{T}}\mathbf{A}$. Then, compute the eigenvalues and eigenvectors of $\mathbf{X}$ and $\mathbf{Y}$ using the built-in commans in R.

Figure 1:

$X = AA^T$

```
A<- matrix(c(1,2,3,-1,0,4), ncol = 3  )

X <- A %*% t(A)
X
```

```
##      [,1] [,2]
## [1,]   10   -1
## [2,]   -1   21
```

Calculate Eigen Vectors and Values of X

```
eigen_values_X <- eigen(X)$values
eigen_values_X
```

```
## [1] 21.09017  9.90983
```

```
eigen_vectors_X <- eigen(X)$vectors
eigen_vectors_X
```

```
##            [,1]       [,2]
## [1,] -0.0898056 -0.9959593
## [2,]  0.9959593 -0.0898056
```

$Y=AˆTA

```
Y <-t(A) %*% A
Y
```

```
##      [,1] [,2] [,3]
## [1,]    5    1    8
## [2,]    1   10   -4
## [3,]    8   -4   16
```

Calculate Eigen Vectors and Values of Y

```r
eigen_values_Y <- eigen(Y)$values
eigen_values_Y
```

```
## [1] 21.09017  9.90983  0.00000
```

```r
eigen_vectors_Y <- eigen(Y)$vectors
eigen_vectors_Y
```

```
##            [,1]       [,2]       [,3]
## [1,] -0.4141868 -0.3734355  0.8300574
## [2,]  0.2755368 -0.9206109 -0.2766858
## [3,] -0.8674842 -0.1141117 -0.4842001
```

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**.

Figure 2:

$svd(A) = U\Sigma V^*$

```r
svd_A <- svd(A)
```

Left Vectors (U) = Eigen Vectors of AA^T

```r
eigen_vectors_X[,1:2]
```

```
##            [,1]       [,2]
## [1,] -0.0898056 -0.9959593
## [2,]  0.9959593 -0.0898056
```

```r
svd_A$u
```

```
##            [,1]      [,2]
## [1,] -0.0898056 0.9959593
## [2,]  0.9959593 0.0898056
```

Right Vectors(V) = Eigen Vectors of $A^TA

```r
svd_A$v
```

```
##            [,1]      [,2]
## [1,]  0.4141868 0.3734355
## [2,] -0.2755368 0.9206109
## [3,]  0.8674842 0.1141117
```

```r
eigen_vectors_Y[,1:2]
```

```
##            [,1]       [,2]
## [1,] -0.4141868 -0.3734355
## [2,]  0.2755368 -0.9206109
## [3,] -0.8674842 -0.1141117
```

Singular Values = sqrt of the Eigen Values for either A^TA or AA^T, with the extra values of A^TA being zero or close to zero

```
svd_A$d
```

```
## [1] 4.592404 3.147988
```

```
sqrt(eigen_values_X)
```

```
## [1] 4.592404 3.147988
```

```
sqrt(eigen_values_Y)
```

```
## [1] 4.592404 3.147988 0.000000
```

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 $\mathbf{A}$ is a matrix and $\mathbf{B}$ is its inverse and $\mathbf{A}\times\mathbf{B} = \mathbf{I}$. The off-diagonal elements of $\mathbf{I}$ 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 $\mathbf{A}$ to compute the inverse.

Please submit PS1 and PS2 in an R-markdown document with your first initial and last name.

Figure 3:

```
myinverse <- function(A) {
  C <- diag (nrow(A))


  for (i in 1:nrow ( A )) {
    for (j in 1:ncol( A )) {
    C[i,j] <- ( det(A[(-1*i), (-1*j) ]) * ((-1)^(i+j) )  )

    }
  }
  T <- diag(nrow(A))

  for ( i in 1:nrow(A)) {
    row <- C[i,]
    for ( j in 1:length(row)) {
      T [ j,i ] <- row[j]
    }

  }
  inv <- 1/(det(A) ) * T
  inv
}
```

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

B <- myinverse(A)

A %*% B
```

```
##               [,1] [,2] [,3]
## [1,]  1.000000e+00    0    0
## [2,] -2.775558e-17    1    0
## [3,] -1.110223e-16    0    1
```