Homework4

Jonathan Hernandez

September 13, 2018

Homework Assignment 4

Problem set 1

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

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

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

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

print("Matricies X and Y: ")
```

```
## [1] "Matricies X and Y: "
```

```
list(X,Y)
```

```
## [[1]]
##      [,1] [,2]
## [1,]   14   11
## [2,]   11   17
##
## [[2]]
##      [,1] [,2] [,3]
## [1,]    2    2   -1
## [2,]    2    4    6
## [3,]   -1    6   25
```

```
# Compute eigenvalues and eigenvectors of X and Y
print("Eigenvalues and Eigenvectors of X and Y:")
```

```
## [1] "Eigenvalues and Eigenvectors of X and Y:"
```

```
eig_X <- eigen(X)
eig_Y <- eigen(Y)

list(eig_X, eig_Y)
```

```
## [[1]]
## eigen() decomposition
## $values
## [1] 26.601802  4.398198
##
## $vectors
##           [,1]       [,2]
## [1,] 0.6576043 -0.7533635
## [2,] 0.7533635  0.6576043
##
##
## [[2]]
## eigen() decomposition
## $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 SVD

svd_A <- svd(A, nu = 2, nv = 3)
svd_A
```

```
## $d
## [1] 5.157693 2.097188
##
## $u
##            [,1]       [,2]
## [1,] -0.6576043 -0.7533635
## [2,] -0.7533635  0.6576043
##
## $v
##             [,1]       [,2]       [,3]
## [1,]  0.01856629 -0.6727903 -0.7396003
## [2,] -0.25499937 -0.7184510  0.6471502
## [3,] -0.96676296  0.1765824 -0.1849001
```

Let's look at the singular vectors d, left singular u and right singular v

for A = UDV' where V' is the complex conjugate (or in this case of real numbers)

just the conjugate

For X an d Y

```r
list(eig_X$vectors, svd_A$u)
```

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

```
list(eig_Y$vectors, svd_A$v)
```

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

We can see that the values (not the sign) for u and v are the eigenvectors

of X and Y respectively.

Let's look at the non-zero eigen values of X and Y and show they are the same:

```
list(eig_X$values, eig_Y$values[1:2])
```

```
## [[1]]
## [1] 26.601802  4.398198
##
## [[2]]
## [1] 26.601802  4.398198
```

And squaring of either the eigenvalue of X or Y yields the non-zero singular values of A

```
list(eig_X$values, svd_A$d[1]^2)
```

```
## [[1]]
## [1] 26.601802  4.398198
##
## [[2]]
## [1] 26.6018
```

```r
list(eig_Y$values[1:2], svd_A$d[2]^2)
```

```
## [[1]]
## [1] 26.601802  4.398198
##
## [[2]]
## [1] 4.398198
```

“‘

```r
# create the function myinverse which will be a function that computes the inverse of a
# well-conditioned full-rank square matrix using co-factors

myinverse <- function(A){
  # check if det(A) = 0 otherwise continue
  n <- nrow(A) # number of rows
  if( det(A) == 0 || n == 1){
    return("Matrix is not invertible. Exiting")
  }

  if (n == 2){ # 2x2 matrix compute inverse manually
    return((1 / det(A)) * matrix(c(A[2, 2], -A[2, 1], -A[1, 2], A[1, 1]),
                                 nrow = 2, ncol = 2))
  }

  # create cofactor matrix set it to be initially all 0's
  # simple case a 2 x 2 matrix
  C <- matrix(0, nrow = n, ncol = n)
  # loop through whole matrix A computing minors and cofactor matrix entries
  for(i in 1:n){
      for(j in 1:n){
        minor <- det(A[-i, -j]) # determinant of minor[i,j]
        C[i, j] <- (-1)^(i + j) * minor # cofactor matrix C[i,j]
      }
  }
  # with the co-factor matrix complete, the inverse of A that is A^-1 is
  # A^-1 = 1/det(A)*t(C) where t(C) is the transpose of the conjugate matrix
  return((1 / det(A))*t(C))
}
A <- matrix(c(1, 2, 4, 7, 1,
              2, 1, -1, 3, 0,
              4, -1, -2, 6, -1,
              3, 1, 5, 2, 1,
              0, 3, 1, 1, 1), nrow = 5, ncol = 5)
B <- myinverse(A)
B
```

```
##            [,1]        [,2]        [,3]        [,4]       [,5]
## [1,] -0.1470588 -0.52941176 -0.20588235  0.20588235  1.5882353
## [2,]  0.5000000 -4.00000000 -2.50000000  0.50000000 14.0000000
## [3,] -0.1176471  2.17647059  1.23529412 -0.23529412 -7.5294118
## [4,]  0.2058824 -0.05882353  0.08823529 -0.08823529  0.1764706
## [5,] -0.1764706  2.76470588  1.35294118 -0.35294118 -8.2941176
```

Show that by multplying the matrix A with it's inverse A^-1, we get the Identity matrix

```r
A %*% B
```

```
##              [,1]          [,2]         [,3]          [,4]          [,5]
## [1,] 1.000000e+00  2.331468e-15 2.109424e-15  2.775558e-16 -2.997602e-15
## [2,] 0.000000e+00  1.000000e+00 8.881784e-16  0.000000e+00 -1.776357e-14
## [3,] 2.498002e-16 -2.664535e-15 1.000000e+00  5.551115e-17  7.105427e-15
## [4,] 3.608225e-16  4.440892e-15 3.108624e-15  1.000000e+00 -5.329071e-15
## [5,] 2.775558e-17  4.440892e-16 0.000000e+00 -5.551115e-17  1.000000e+00
```

As we can see the off-diagonals are very close to 0 and the diagonal entries equal to 1.