Week15 - IS622

James Quacinella

11/22/2015

Exercise 11.1.6

For the matrix of Exercise 11.1.4:

(a) Starting with a vector of three 1's, use power iteration to find an approximate value of the principal eigenvector.

(b) Compute an estimate the principal eigenvalue for the matrix.

(c) Construct a new matrix by subtracting out the effect of the principal eigenpair, as in Section 11.1.3.

(d) From your matrix of (c), find the second eigenpair for the original matrix of Exercise 11.1.4.

(e) Repeat (c) and (d) to find the third eigenpair for the original matrix.

Answer

```
frobenius <- function(M) {
  return(sqrt(sum(apply(M, c(1, 2), function(x) {x*x}))));
}

principal_eigen_vector <- function(M, x, tolerance) {
  # print(M)
  diff = 1
  x_new <- x
  count <- 0;
  while(diff > tolerance && count < 10000) {
    #print(x_new)

    # update x_old to next value
    x_old <- x_new

    # Calc new x
    x_new <- (M %*% x_old) / frobenius(M %*% x_old)

    if(x_new[which.max(abs(x_new))] < 0) {
      x_new <- -1 * x_new;
    }

    # update diff from new x to old x
    diff <- frobenius(x_new - x_old)
    #print(diff)

    count <- count + 1;
  }
```

```
  return(x_new);
}

eigen_value <- function(M, eigenvec) {
  return(as.double(t(eigenvec) %*% M %*% eigenvec))
}

M <- matrix(c(1,1,1,1,2,3,1,3,5), ncol=3, byrow=TRUE)
x <- c(1,1,1)

# Calculate first eigenvector
first_evector <- principal_eigen_vector(M, x, tolerance = 0.0007)
first_evalue <- eigen_value(M, first_evector)

# Deflate M
M_1 <- M - first_evalue * first_evector %*% t(first_evector)

# Calculate second eigenvector
second_evector <- principal_eigen_vector(M_1, first_evector, tolerance = 0.0007)
second_evalue <- eigen_value(M_1, second_evector)

# Deflate M again
## This has a spurious eigenvector that the next step finds instead
M_2 <- M_1 - second_evalue * second_evector %*% t(second_evector)

# Calculate third eigenvector
third_evector <- principal_eigen_vector(M_2, second_evector, tolerance = 0.0007)
third_evalue <- eigen_value(M_2, third_evector)
```

Correctly find the first eigen vector and value:

```
# Print results
print(first_evector)
print(first_evalue)
print(eigen(M))
```

```
##           [,1]
## [1,] 0.2185605
## [2,] 0.5216310
## [3,] 0.8247014
## [1] 7.162278
## $values
## [1]  7.162278e+00  8.377223e-01 -2.220446e-16
##
## $vectors
##           [,1]       [,2]       [,3]
## [1,] 0.2184817  0.8863403  0.4082483
## [2,] 0.5216090  0.2475023 -0.8164966
## [3,] 0.8247362 -0.3913356  0.4082483
```

Correctly find the second eigen vector and value:

```
print(second_evector)
print(second_evalue)
print(eigen(M_1))
```

```
##            [,1]
## [1,]  0.8861741
## [2,]  0.2471062
## [3,] -0.3919616
## [1] 0.8377228
## $values
## [1]  8.377228e-01 -2.672823e-16 -3.765475e-07
##
## $vectors
##            [,1]       [,2]      [,3]
## [1,]  0.8861741  0.4082483 0.2191546
## [2,]  0.2471062 -0.8164966 0.5217967
## [3,] -0.3919616  0.4082483 0.8244388
```

As you can see below, there is a spurious eigenvector in the new deflated M value. I am not sure how to handle this.

```
print(third_evector)
print(third_evalue)
print(eigen(M_2))
```

```
##           [,1]
## [1,] 0.2185605
## [2,] 0.5216310
## [3,] 0.8247014
## [1] -3.765477e-07
## $values
## [1]  1.693000e-13 -1.711611e-16 -3.765477e-07
##
## $vectors
##            [,1]       [,2]       [,3]
## [1,]  0.8863608  0.4081616 -0.2185605
## [2,]  0.2473762 -0.8165208 -0.5216310
## [3,] -0.3913689  0.4082866 -0.8247014
```

```
# Still doesn't help
library(MASS)
third_evector <- principal_eigen_vector(ginv(M), x, tolerance = 0.0007)
third_evector
```

```
##            [,1]
## [1,]  0.8863503
## [2,]  0.2475264
## [3,] -0.3912975
```

Exercise 11.3.2

Use the SVD from Fig. 11.7. Suppose Leslie assigns rating 3 to Alien and rating 4 to Titanic, giving us a representation of Leslie in "movie space"" of [0, 3, 0, 0, 4]. Find the representation of Leslie in concept space. What does that representation predict about how well Leslie would like the other movies appearing in our example data?

Answer

```
M <- matrix(c(1,1,1,0,0,
              3,3,3,0,0,
              4,4,4,0,0,
              5,5,5,0,0,
              0,0,0,4,4,
              0,0,0,5,5,
              0,0,0,2,2), ncol=5, byrow=TRUE)

U <- matrix(c(0.14, 0,
              0.42, 0,
              0.56, 0,
              0.70, 0,
              0, 0.60,
              0, 0.75,
              0, 0.30), ncol=2, byrow=TRUE)

Sigma <- matrix(c(12.4, 0, 0, 9.5), ncol=2, byrow=TRUE)

Vt <- matrix(c(0.58, 0.58, 0.58, 0, 0,
              0, 0, 0, 0.71, 0.71), ncol=5, byrow=TRUE)

leslie <- c(0, 3, 0, 0, 4)

leslie %*% t(Vt) # 1.74 2.84
```

```
##      [,1] [,2]
## [1,] 1.74 2.84
```

This shows that Leslie would tend to like group 2 of movies (romance) moreso than group 1, sci-fi movies. This makes sense since she rated the one romance movie higher than the one sci-fi movie.

We can confirm this by looking at a possible set of ratings Leslie would have for all movies:

```
leslie %*% t(Vt) %*% Vt
```

```
##        [,1]   [,2]   [,3]   [,4]   [,5]
## [1,] 1.0092 1.0092 1.0092 2.0164 2.0164
```

So, it does seem that Leslie would rate the romance movies higher than the sci-fi movies.