Ex MDS

Vicent Caselles Ballester

2024-04-17

Exercici 1

```
dat <- matrix(c(0, 7, 5, 9, 5, 7, 9,
                7, 0, 4, 6, 4, 6, 7,
                5, 4, 0, 3, 4, 5, 6,
                9, 6, 3, 0, 3, 2, 2,
                5, 4, 4, 3, 0, 5, 4,
                7, 6, 5, 2, 5, 0, 4,
                9, 7, 6, 2, 4, 4, 0), ncol=7)
all(dat == t(dat)) # to make sure I haven't made mistakes
```

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

a) Construir la matriz B $= -\frac{1}{2}HD^{(2)}H$, donde $D^{(2)}$ es la matriz de distancias al cuadrado y H es la matriz de centrado, y calcular sus valores propios. Observar si la matriz de distancias es euclídea.

From [1] :

> **Theorem**: Let D be a distance matrix and define K by (2). Then D is Euclidean if and only if K is positive semi-definite.

K is the Kernel matrix, refered to as B in Everitt, 2011. Therefore, let's use some of the functions I've created to check if a symmetric matrix is positive definite.

```
source('../../funcs/sym.R')
require(ade4); require(matrixcalc)
```

```
## Loading required package: ade4
```

```
## Loading required package: matrixcalc
```

```
positive_definite(dat)
```

```
## [1] FALSE
```

```
is.positive.semi.definite(dat)
```

```
## [1] FALSE
```

```
is.euclid(as.dist(dat))
```

```
## [1] FALSE
```

With this proof, I think we can be somewhat sure that the distances are not euclidean.

[1] https://www.math.uwaterloo.ca/ aghodsib/courses/f10stat946/notes/lec10-11.pdf

About the first part, we can calculate B using some functions I've written.

```r
source('../../funcs/MDS.R')
D <- dat
B <- KernelMatrix(D**2)
B2 <- B_from_D(D)

max(abs(B-B2)) # We see that both functions using different approaches accomplish the same
```

```
## [1] 7.105427e-15
```

Eigenvalues of B:

```r
evB <- eigen(B)$values
sum(evB < 0)
```

```
## [1] 2
```

We see that one eigenvalue of B is < 0, therefore B is not positive semi-definite, therefore D is not euclidean.

b) Obtener la representación con las dos primeras coordenadas principales e indicar el grado de bondad de esta representación. Se puede hacer a partir de la descomposición de la matriz B o con la función `cmdscale`.

I'm gonna choose the first option. We select fist two eigen{values, vectors} pairs of B.

```r
evecs_2 <- eigen(B)$vectors[,1:2]
evalues_2 <- diag(sqrt(eigen(B)$values[1:2]))
new_X <- evecs_2%*%evalues_2

max(abs(new_X - cmdscale(dat, k=2)))
```

```
## [1] 5.107026e-15
```

Let's check how different the new euclidean distances are using the reduced data matrix (`new_X`).

```r
max(abs(dat - as.matrix(dist(new_X))))
```

```
## [1] 2.934933
```

We see that there is a sufficiently enough large distance difference between the points in the original space and the ones in the new reduced space.

Using the criteria from Everitt 2011:

```r
sum(eigen(B)$values[1:2])/sum(eigen(B)$values[eigen(B)$values>0])
```

```
## [1] 0.7881461
```

They also mention the following criteria in case B is not positive definite:

```r
sum(abs(eigen(B)$values[1:2]))/sum(abs(eigen(B)$values))
```

```
## [1] 0.7062874
```

```r
sum(eigen(B)$values[1:2] ** 2)/sum(eigen(B)$values ** 2)
```

```
## [1] 0.9135125
```

We see that the results differ a lot.