PCA-musk

BELFADIL Anas

10/7/2019

Find an R package that performs truncated SVD.

The svd function in the R base package performs truncated SVD and returns the tree matrices of the singular value decomposition d, u and v where:

d a vector containing the singular values of $x_{(n,p)}$, of length min(n, p), sorted decreasingly.

u a matrix whose columns contain the left singular vectors of x.

v a matrix whose columns contain the right singular vectors of x.

Create a function (or write and R script) that performs PCA based on truncated SVD.

The PCA using the *prcomp* R function is equivalent to performing an SVD on the centered data, where the centering occurs on the columns. So, to have comparable results, we must center the Data before performing the SVD. We can do this with the help of the function *scale* wich centres and scales the data column wise on the input matrix, with the argument scale = FALSE we can perform centring with this function without scaling.

```
MyPCA <- function(x){ # In: matrix - out: the Pincipal Components as columns of a matrix
  x <- scale(x, scale = FALSE) # Center the matrix as 'prcomp' does the same
  DD <- svd(x)
  d <- diag(DD$d)
  u <- DD$u
  u %*% d
}
```

Perform PCA analysis using this new R code and compare the results obtained using prcomp function on the data set musk.txt

Preparing the Data

```
dd <- read.delim("/home/anas/musk.txt")
o <- which(colnames(dd) == "musk")
d <- as.matrix(dd[,-o])
```

PCA analysis using *MyPCA*, *prcomp* and comparing the results:

the columns of the output of *MyPCA* function from the SVD methode, correspond to the principal components x in the PCA of the *prcomp* function.

```
PCA <- MyPCA(d)
PCA[1:5, 1:6]
```

```
##            [,1]     [,2]      [,3]     [,4]       [,5]      [,6]
## [1,] -118.66637 750.0974 214.33803 643.5597 -169.49701  54.14408
## [2,] -133.54577 757.1460 187.45835 644.4074  -66.09163 107.01631
## [3,]  -52.86381 730.1465 166.73382 673.5671 -161.39161  93.37711
```

```
## [4,] -202.25458 797.9864 237.67580 613.7484  -74.39267  73.60633
## [5,] -147.02514 759.8111  42.86736 608.5850 -142.52623 195.50348
```

```
pp <- prcomp(d)
pp$x[1:5, 1:6]
```

```
##             PC1      PC2       PC3      PC4        PC5       PC6
## [1,] -118.66637 750.0974 214.33803 643.5597 -169.49701  54.14408
## [2,] -133.54577 757.1460 187.45835 644.4074  -66.09163 107.01631
## [3,]  -52.86381 730.1465 166.73382 673.5671 -161.39161  93.37711
## [4,] -202.25458 797.9864 237.67580 613.7484  -74.39267  73.60633
## [5,] -147.02514 759.8111  42.86736 608.5850 -142.52623 195.50348
```

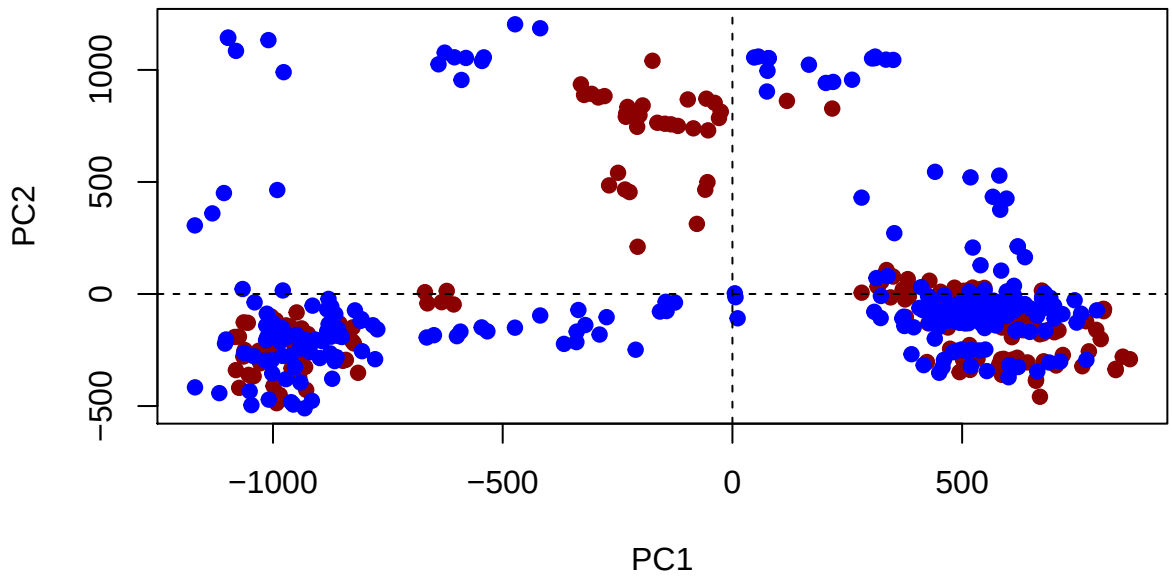
```
Diff <- abs(PCA - pp$x) # Calculate the difference to compare the resluting matrices
max(Diff) # Print the maximum difference elements wise between the matrices
```

```
## [1] 9.777068e-12
```

The maximum difference between the two methodes is 9.777068×10^{-12} wich is negligeable, we can conclude that they give the same result.

Plot the molecules (e.g. observations/rows) in the first two axes and color each dot using the information given in the column musk.

```
group <- factor(dd[,o], labels = c("non-musk", "musk"))
mycol <- ifelse(group=="musk", "darkred", "blue")
plot(PCA[,1], PCA[,2], pch = 19, col=mycol, xlab="PC1", ylab="PC2")
abline(h=0, v=0, lty = 2)
```



```r
sessionInfo()
```

```
## R version 3.6.1 (2019-07-05)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 18.04.3 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.7.1
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.7.1
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_CA.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_CA.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_CA.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_CA.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## loaded via a namespace (and not attached):
##  [1] compiler_3.6.1  magrittr_1.5    tools_3.6.1     htmltools_0.4.0
##  [5] yaml_2.2.0      Rcpp_1.0.2      stringi_1.4.3   rmarkdown_1.16
##  [9] knitr_1.25      stringr_1.4.0   xfun_0.10       digest_0.6.21
## [13] rlang_0.4.0     evaluate_0.14
```