Partitioning subjects based on high-dimensional fMRI data

Jeffrey Durieux, MSc

6/26/2019

Introduction

This document contains a brief tutorial on how apply the two-step procedure in R. For more information about this procedure please read my paper: https://lnkd.in/gmRyK-5

Note that a Juypter notebook of this tutorial can be found on my GitHub repo

Download an example dataset and load it into R

First download an example dataset via this link: https://surfdrive.surf.nl/files/index.php/s/j0RGzTWwYo2l6GS This dataset is stored as a .Rdata object and it contains a large list with 60 elements (+/- 46 Mb). Each element is a matrix with 1000 rows and 100 columns. This is a very common way for me to store three-way data in R. (Note that in R it is also entirely possible to store three-way data in an array)

```
load("~/Downloads/ExampleData.Rdata") # object is named Xe
# size of the list
length(Xe)
```

```
## [1] 60
```

```
# show first two objects of the list
str(Xe, list.len = 2)
```

```
## List of 60
##  $ SimCluster_1: num [1:1000, 1:100] -4.8 14.18 15.33 -3.9 -5.38 ...
##  $ SimCluster_1: num [1:1000, 1:100] -4.235 -2.367 6.978 -5.228 0.504 ...
##   [list output truncated]
```

```
# names of list
names(Xe)
```

```
##  [1] "SimCluster_1" "SimCluster_1" "SimCluster_1" "SimCluster_1"
##  [5] "SimCluster_1" "SimCluster_2" "SimCluster_2" "SimCluster_2"
##  [9] "SimCluster_2" "SimCluster_2" "SimCluster_2" "SimCluster_2"
## [13] "SimCluster_2" "SimCluster_2" "SimCluster_2" "SimCluster_3"
## [17] "SimCluster_3" "SimCluster_3" "SimCluster_3" "SimCluster_3"
## [21] "SimCluster_3" "SimCluster_3" "SimCluster_3" "SimCluster_3"
## [25] "SimCluster_3" "SimCluster_3" "SimCluster_3" "SimCluster_3"
## [29] "SimCluster_3" "SimCluster_3" "SimCluster_3" "SimCluster_3"
## [33] "SimCluster_3" "SimCluster_3" "SimCluster_3" "SimCluster_4"
## [37] "SimCluster_4" "SimCluster_4" "SimCluster_4" "SimCluster_4"
## [41] "SimCluster_4" "SimCluster_4" "SimCluster_4" "SimCluster_4"
## [45] "SimCluster_4" "SimCluster_4" "SimCluster_4" "SimCluster_4"
## [49] "SimCluster_4" "SimCluster_4" "SimCluster_4" "SimCluster_4"
## [53] "SimCluster_4" "SimCluster_4" "SimCluster_4" "SimCluster_4"
## [57] "SimCluster_4" "SimCluster_4" "SimCluster_4" "SimCluster_4"
```

Source some functions from my GitHub page

```
suppressPackageStartupMessages(library(RCurl))

s1 <- getURL("https://raw.githubusercontent.com/jeffreydurieux/Tutorials/master/modRV.R",
                  ssl.verifypeer = FALSE)
s2 <- getURL("https://raw.githubusercontent.com/jeffreydurieux/Tutorials/master/computeRVmat.R",
                  ssl.verifypeer = FALSE)

eval(parse(text = s1))
eval(parse(text = s2))
rm(s1, s2)
```

Step 1: apply a data reduction to the example dataset

In the first step, ICA's are performed on each element of your list (the example dataset) and we store the estimated component matrices S in a list object named `ICAList`. Note that for this example we select for 20 components, since this is the true underlying components of the example dataset. In practice, it is recommended to apply a model selection procedure to the data in order to choose the optimal number of components present in each dataset.

```
library(ica)

ICAlist <- lapply(X = Xe, FUN = icafast, nc = 20)
ICAlist <- lapply(X = ICAlist, function(anom) anom$S)
```

Step 2: Calculate all pairwise modified RV coefficients

Use the two functions sourced from my GitHub page to compute all modified RV coefficients between all subject pairs. Depending on the argument settings, the fuction returns a `dist` object or a similarity matrix. Also note that when the argument `verbose == TRUE` a progressbar is added to the console so that you can monitor the computation time.

```
RVmat <- computeRVmat(DataList = ICAlist, dist = TRUE, verbose = TRUE)
```

```
## ================================================================================
```

Apply a clustering procedure

After computing the modified RV matrix, you can apply a clustering procedure to this matrix. In this example we use hierarchical clustering with Ward's method. In order to see whether the clustering result is estimated correctly you can compare the colour of the branches with the symbols on the leaf nodes. The symbols represent the true clustering and the colour of the branches represent the estimated clustering.

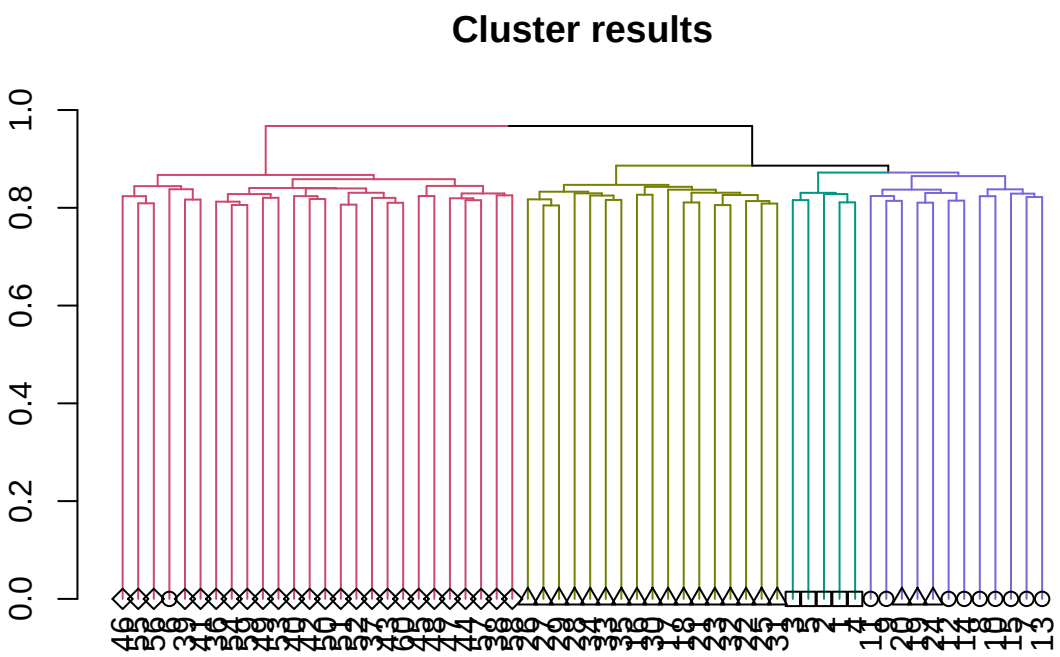
```
suppressPackageStartupMessages(library(dendextend))

res <- hclust(RVmat, method = 'ward.D2')
res <- as.dendrogram(res)

nameres <- names( cutree(res, k = 4, order_clusters_as_data = F))
nn <- rep(NA, 60)
nn[nameres %in% paste(1:5)] <- 0
```

```r
nn[nameres %in% paste(6:15)] <- 1
nn[nameres %in% paste(16:35)] <- 2
nn[nameres %in% paste(36:60)] <- 5

res %>% set("branches_k_color", k = 4) %>%
  set("leaves_pch", nn) %>%
  plot(main = 'Cluster results')
```



Final notes

Note that this tutorial document assumed that you installed the `RCurl`, `ica` and `dendextend` R packages. Moreover, the ICA and modified RV computations are done in a sequential manner. However both can be done in parallel on a computer cluster if necessary; applying single subject ICA to each matrix can be done in parallel and all modified-RV coefficients between all subject pairs (total of $\frac{N(N-1)}{2}$) can also be computed in parallel. (Both are embarrassingly parallel computing problems).

Hope you liked this very short tutorial and if you have any question or suggestions please contact me via: j.durieux@fsw.leidenuniv.nl