

‘From Statistics to Data Mining’

Computer Lab Session n° 10:


Clustering (2/2)

Master 1 COSI / CPS²
Saint-Étienne, France

Fabrice Muhlenbach

Laboratoire Hubert Curien, UMR CNRS 5516
Université Jean Monnet de Saint-Étienne
18 rue du Professeur Benoît Luras
42000 SAINT-ÉTIENNE, FRANCE
<https://perso.univ-st-etienne.fr/muhlfabr/>

Outcome

The objective of this lab is to become familiar with a few clustering methods and clustering quality values by using some functions in .

1 Clustering Iris Dataset


1.1 Loading the Dataset

Iris plants dataset is commonly used in data analysis. The data can be loaded in memory with the `data` function. The four first attributes will be used for the clustering but the class (the fifth attribute) can be used to test the validity of the clustering.

```
data("iris")  
mydata <- iris[1:4]  
class <- as.matrix(iris[5])
```

1.2 The k -Means Clustering

1.2.1 Introduction

We are already familiar with k -means algorithm (we even programmed it!). We can use this clustering algorithm simply in  with the `kmeans` function with (at least) 2 parameters: the name of the dataset and a value for k (the number of expected clusters). For example, with our dataset and $k = 3$ clusters:

```
kmeans.result <- kmeans(mydata,3)
```

The `summary` function will indicate what are the results obtained with the k -means algorithm:

```
summary(kmeans.result)
```

- `cluster`: number (from 1 to k) indicating the cluster to which each point is allocated
- `centers`: coordinates of the centroids (the means)
- `totss`: total sum of squares
- `withinss`: within-cluster sum of squares (within each cluster)
- `tot.withinss`: total within-cluster sum of squares
- `betweenss`: between-cluster sum of squares
- `size`: number of points in each clusters

By printing `kmeans.result`, we will obtain the value of all these results.

1.2.2 Clustering Stability

Print the contingency table of the real classes of iris dataset (the “ground truth”) and the cluster obtained by k -means.

```
print(table(class, kmeans.result$cluster))
```

Run the k -means algorithm 10 times on the whole iris dataset. Print each time the contingency table, but the within-cluster sum of squares (WSS) and the between-cluster sum of squares (BSS) too.

```
w <- (kmeans.result$tot.withinss/kmeans.result$totss)*100
b <- (kmeans.result$betweenss/kmeans.result$totss)*100
print(paste("WSS=", round(w,2), "%"))
print(paste("BSS=", round(b,2), "%"))
print(table(class, kmeans.result$cluster))
```

Do you get every time the same results? Have you noticed a link between the results of clustering obtained by k -means and the values of WSS and BSS?

1.2.3 Clustering Validation

Issues that arise when doing clustering are:

- What is “good clustering”?
- How many clusters?
- How to find the clusters?
- A good clustering method will produce high quality clusters with:

- high intra-class similarity
- low inter-class similarity
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation.
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns.

1.2.4 Internal Validation – Cohesion and Separation

The cluster validity may be approached in three possible directions:

- the clustering is evaluated in terms of an independently drawn structure, imposed on the data a priori. The criteria used in this case are called “external criteria”.
- the clustering is evaluated in terms of quantities that involve the vectors of the attributes themselves (e.g., proximity matrix). The criteria used in this case are called “internal criteria”.
- the clustering is evaluated by comparing it with other clustering structures, resulting from the application of the same clustering algorithm but with different parameter values, or other clustering algorithms, on the dataset. Criteria of this kind are called “relative criteria”.

When we know the “ground truth”, we can apply an external criterium (e.g., Rand statistic, Jaccard statistic, Fowlkes-Mallows index, etc.)


However, most of the time in unsupervised learning, we do not have ground truth (unlike iris dataset) and we do not know the real classes (as we seek to build them), that’s why we need to use an internal validation.

The **cluster cohesion** is the sum of the weight of all links within a cluster (Figure 1). In practice, this value can be computed by the sum of squares of the distances within the clusters (WSS: within-cluster sum of squares).

The **cluster separation** is the sum of the weights between nodes in the cluster and nodes outside the cluster (Figure 1). In practice, this value can be computed by the sum of squares of the distances between the clusters (BSS: between-cluster sum of squares).

The internal validation criteria will use these values (WSS and BSS) and a clustering will be considered as “better” than another if it minimize better the within cluster variance (e.g., WSS that represents the intra-cluster variance) and maximize better the between cluster variance (e.g., BSS that represents the inter-cluster variance), see Figure 2.

1.3 Hierarchical Clustering

In , a hierarchical cluster analysis can be obtained with `hclust` function. This function needs (at least) 2 parameters: a distance matrix and the name of an agglomeration method to be used (e.g., “ward”, “single”, “complete”, “average”, “mcquitty”, “median” or “centroid”). For example, with our dataset and an average aggregation method:

```
hc <- hclust(dist(mydata), "ave")
```

The summary of the results can be obtained with the `summary` function.

```
summary(hc)
```

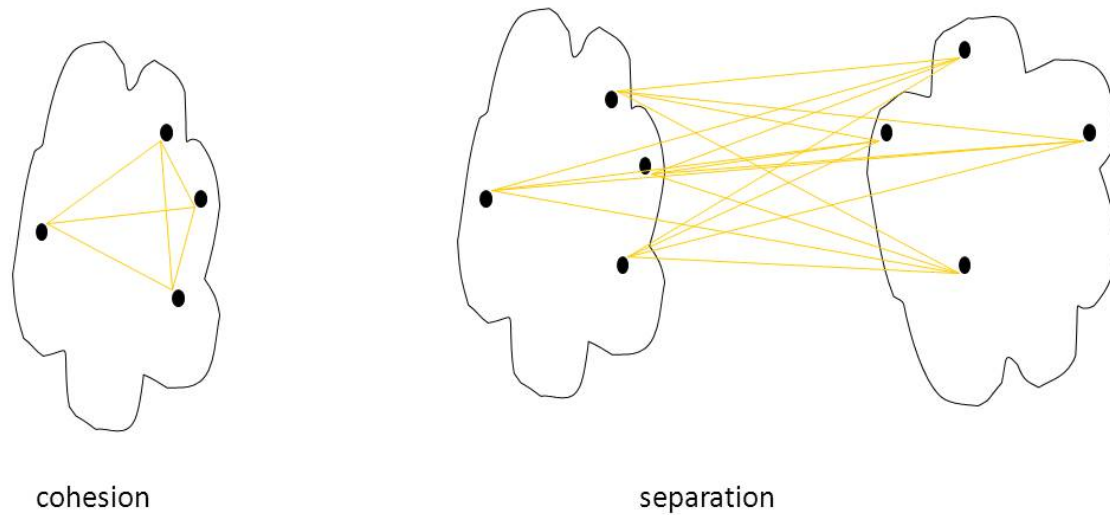


FIG. 1 – Cohesion and Separation.

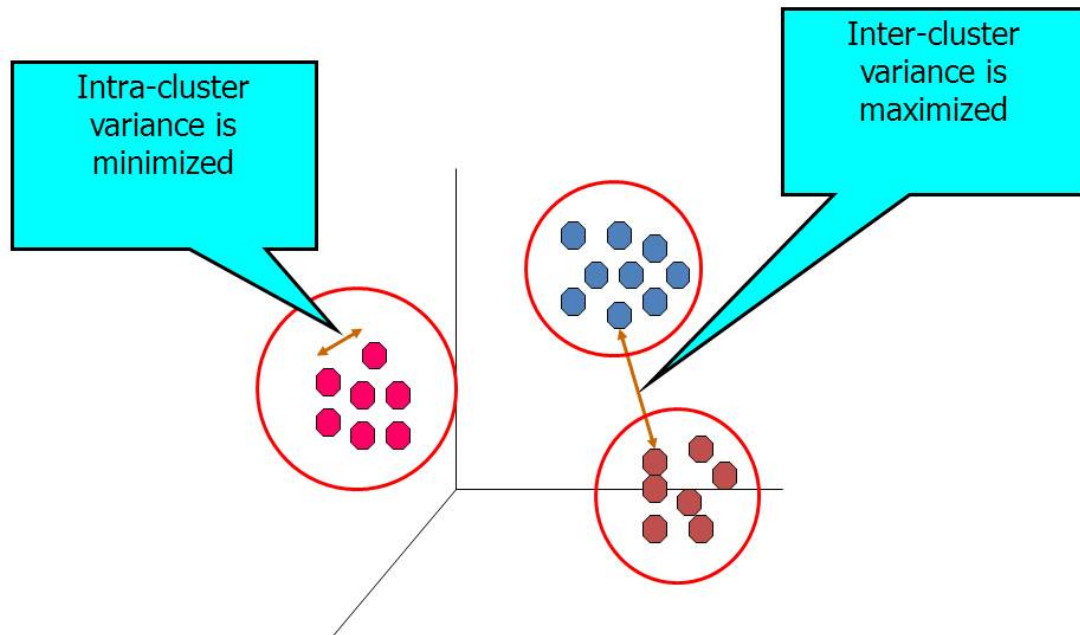
Printing the result variable `hc`, is not very interesting. The result has to be plotted for having a cluster dendrogram.

```
win.graph(800, 600, 10)
plot(hc, hang = -1, labels=class)
```

Nevertheless, there are too many data in iris dataset to see the results produced by hierarchical clustering. We can take a sample and run the hierarchical clustering algorithm on it.

```
idx <- sample(1:dim(iris)[1], 40)
irisSample <- iris[idx,]
irisSample$Species <- NULL
hc <- hclust(dist(irisSample), method="ave")
win.graph(800, 600, 10)
plot(hc, hang = -1, labels=iris$Species[idx])
```

On this plot, we can cut the dendrogram in 3:

FIG. 2 – *Inter-Cluster Variance and Intra-Cluster Variance.*

```
rect.hclust(hc, k=3)
groups <- cutree(hc, k=3)
```

2 Clustering Ruspini Dataset

2.1 Loading the Dataset

Ruspini dataset is a textfile that can be loaded in memory with the `read.table` function. First, we clean the memory to remove the previous variables and second we read the file.

```
rm(list=ls())
mydata <- read.table("~/R/data/ruspini.txt", quote="\"")
```

2.2 The k -Means Clustering and Internal Clustering Validation

Unlike for iris dataset, with Ruspini dataset we don't know how many clusters we have to find (there is no ground truth).

Run k -means algorithm on this dataset with many tries for k (from 2 to 10 clusters). For each try, print WSS and BSS and the Ball and Hall clustering quality index (1965) obtained by:

$$B_H_index = WSS/k$$

and the Calinski and Harabasz clustering quality index (1974) obtained by:

$$C_H_index = \frac{\frac{BSS}{k-1}}{\frac{WSS}{n-k}}$$

The smaller the Ball and Hall clustering quality index is, the better the clustering is, and the greater the Calinski and Harabasz clustering quality index is, the better the clustering is.

With the help of these clustering quality indices (and specially the last one), what can be the most relevant value for the number of clusters k ?

3 Recommended Readings

The main literature for this section is:

- Adler (2010), "R in a Nutshell – a Desktop Quick Reference," Chapter 22 "Machine Learning."
- Baayen (2008), "Analyzing Linguistic Data: A Practical Introduction to Statistics using R," Chapter 5 "Clustering and classification."
- Cook and Swayne (2007), "Interactive and Dynamic Graphics for Data Analysis: With R and GGobi," Chapter 5 "Cluster Analysis."
- Everitt and Hothorn (2010), "A Handbook of Statistical Analyses Using R," Chapter 9 "Recursive Partitioning" and Chapter 18 "Cluster Analysis."
- Everitt and Hothorn (2011), "An Introduction to Applied Multivariate Analysis with R," Chapter 6 "Cluster Analysis."
- Husson et al. (2010), "Exploratory Multivariate Analysis by Example Using R," Chapter 4 "Clustering."
- James et al. (2013), "An Introduction to Statistical Learning: with Applications in R," Chapter 10 "Unsupervised Learning."
- Ledolter (2013), "Data Mining and Business Analytics with R," Chapter 15 "Clustering."
- Tufféry (2010) « Data mining et statistique décisionnelle: L'intelligence des données », Chapitre « Les techniques de classification automatique » (in French).
- Wehrens (2011), "Chemometrics with R: Multivariate Data Analysis in the Natural Sciences," Chapter 5 "Self-Organizing Maps" and Chapter 6 "Clustering."

References

- Adler, J. (2010). *R in a Nutshell – a Desktop Quick Reference*. O'Reilly.
- Baayen, R. H. (2008). *Analyzing Linguistic Data: A Practical Introduction to Statistics using R*. Cambridge University Press.
- Cook, D. and D. F. Swayne (2007). *Interactive and Dynamic Graphics for Data Analysis: With R and GGobi*. Use R! Springer.
- Everitt, B. and T. Hothorn (2011). *An Introduction to Applied Multivariate Analysis with R*. Use R! Springer.
- Everitt, B. S. and T. Hothorn (2010). *A Handbook of Statistical Analyses Using R* (2nd ed.). Chapman & Hall / CRC.
- Husson, F., S. Lê, and J. Pagès (2010). *Exploratory Multivariate Analysis by Example Using R*. Computer Science & Data Analysis. Chapman & Hall / CRC.
- James, G., D. Witten, T. Hastie, and R. Tibshirani (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer.
- Ledolter, J. (2013). *Data Mining and Business Analytics with R*. John Wiley & Sons, Ltd.
- Tufféry, S. (2010). *Data mining et statistique décisionnelle: L'intelligence des données* (3^e ed.). Paris: Editions Technip.
- Wehrens, R. (2011). *Chemometrics with R: Multivariate Data Analysis in the Natural Sciences*. Use R! Springer.