

Course 5 : Mining quantitative multivariate data

-

Introduction to data partitioning

20 November 2012

Introduction

K-means method

Hierarchical clustering

Conclusion

Summary

Example

Introduction

Summary

The data partitioning (clustering data) is a statistical method of data analysis which aims to combine a set of different data into the homogeneous sets, ie that the data in each subset share common (correlated) characteristics, which usually correspond to criteria of proximity, which is defined by a measures of distance (or similarity/disimilarity measure).

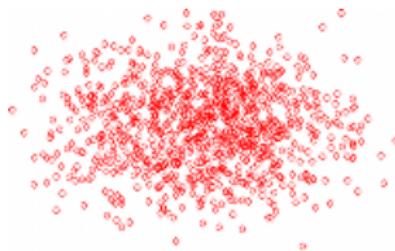
Summary

To obtain a good partitioning, the algorithm have to :

- Minimize the inertia (variance) intra-class for classes (= groups or *clusters*) as homogeneous as possible.
- Maximize the inter-class inertia to obtain well-differentiated groups.

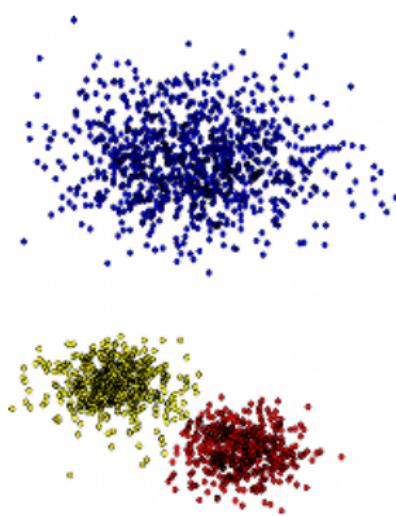
Example

We want to automatically discover groups of similar data :



Example

We want to automatically discover groups of similar data :



K-means method

Principle

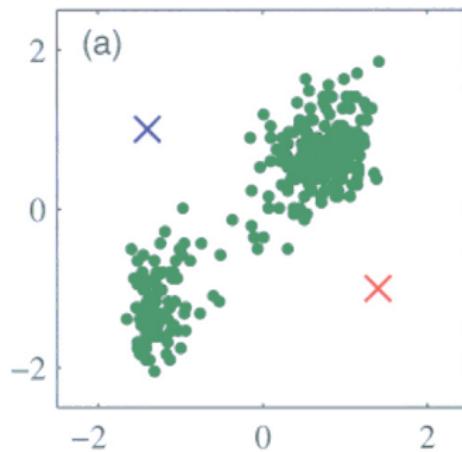
The method of K-Means is a special case of the method of mobile centers. The main objective of these methods is to select a number of representatives (= center or prototypes) in the data space. Each prototype represents a group. Thus at the end of the process we associate each data point to its closest prototype in order to obtain a segmentation of the data into different homogeneous groups.

The algorithm

- ① Randomly select k initial centers (for example, from the data).
- ② Assign each data x to the cluster (group) i whose center C_i is closest to x .
- ③ If no observation change cluster - stop.
- ④ Else, compute new centers : for all i , C_i is the center of gravity (centroid) of the group i .
- ⑤ Go to step 2.

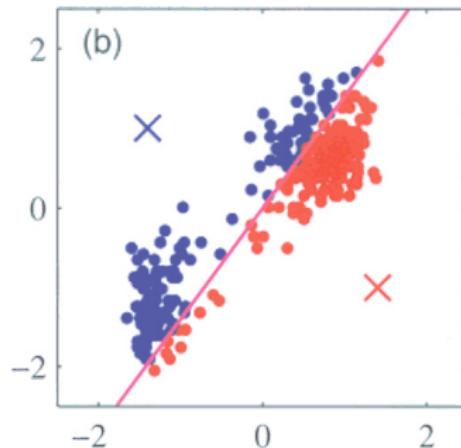
The algorithm

Example :



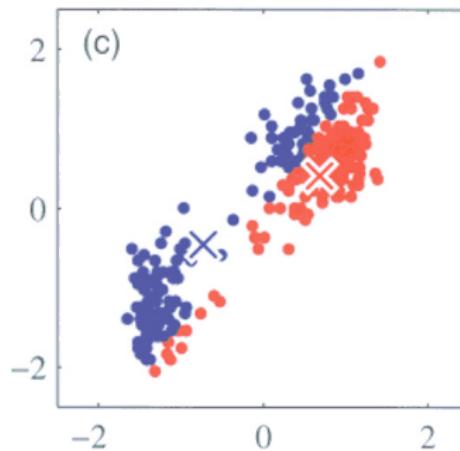
The algorithm

Example :



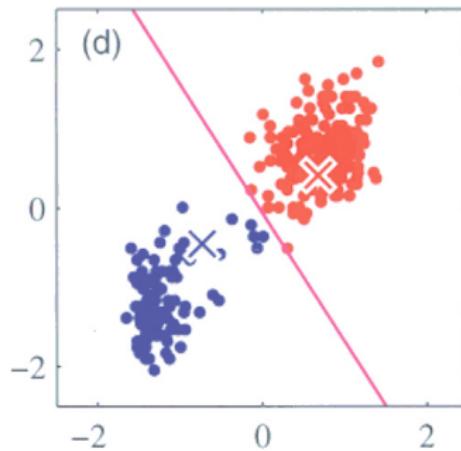
The algorithm

Example :



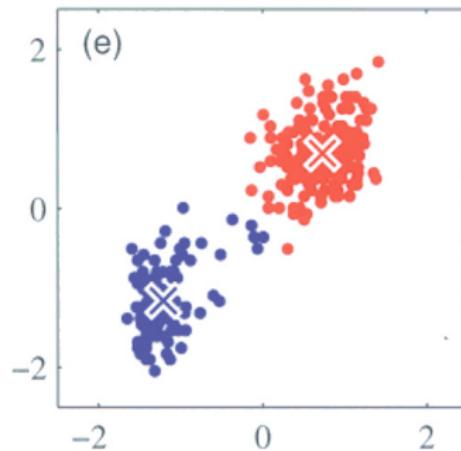
The algorithm

Example :



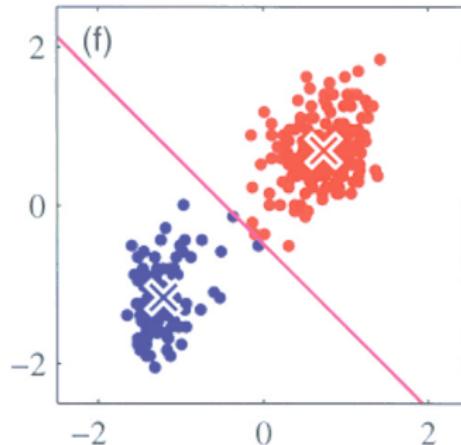
The algorithm

Example :



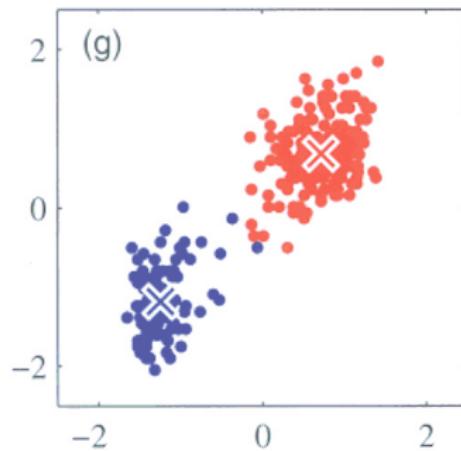
The algorithm

Example :



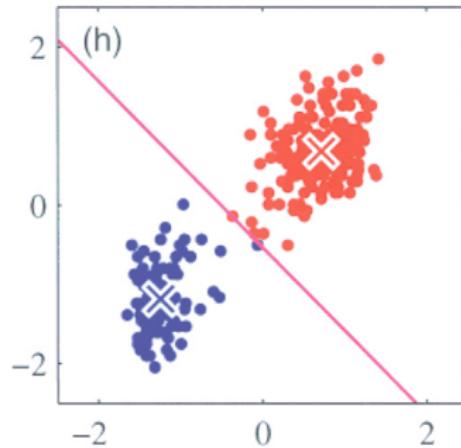
The algorithm

Example :



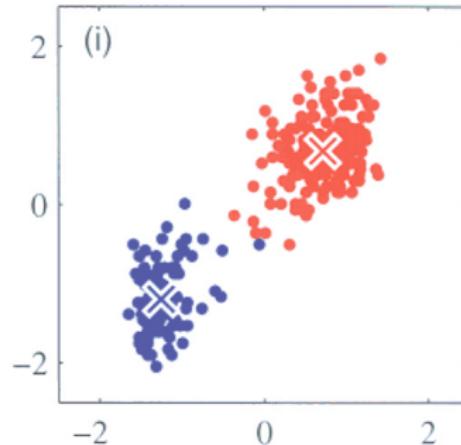
The algorithm

Example :



The algorithm

Example :



Problems

Choice of the distance measure (metric)

This choice is very important, with different distance measures are obtained different results !

Usually we use the Euclidean distance :

$$\sqrt{\sum_{i=1}^n |x_i - y_i|^2}$$

with n the number of variables.

Problems

Instability

The final result is highly dependent on the initialization of centers.
A solution of this problem is to start the algorithm several times
with random initialization and to keep only the best result.

Choosing the number of groups

The number of groups obtained at the output of the algorithm must
be selected by the user. But in general it is not known !
So usually we start the algorithm several times with different
choices for the number of groups and we keep only the best result.

Validation measures

We have seen that it is necessary to restart the K-Means algorithm many times and keep the best result. So we need an estimation index for the segmentation quality.

Davies-Bouldin index

Let S_i be the average distance of the data of group i to their prototype c_i , the DB index selects the segmentation that maximizes the distance between groups and minimizes the intra-group variance. This index is one of the most used.

$$DB(K) = \frac{1}{K} \sum_{i=1}^K \max_{j \neq i} \frac{s_i + s_j}{\text{dist}(c_i, c_j)}$$

Algorithm

- ① Choose k between 1 and k_{max} .
- ② Run $Iter$ times the basic algorithm of K-Means and keep the best segmentation S_k according to Davies-Bouldin index ($Iter = 50$ for ex.).
- ③ If all values of k have been tested, retain among S_k the best segmentation S according to Davies-Bouldin, otherwise go to 1.

Hierarchical clustering

Principle

Create a partition at each step obtained by aggregating the 2-2 closest elements (element = data or group of data). The algorithm provides a hierarchy of partitions : tree containing the history of the classification allowing to find n-1 partitions.

- Need to use a metric (Euclidean distance, ...).
- Need to set a rule for aggregating data or a data group with another group : aggregation criterion.

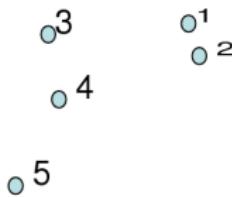
Algorithm

- ① Calculate the distance matrix between the n elements and combine the two closest elements.
- ② If all the data are not grouped into one group, return to 1.
- ③ Otherwise, construct the dendrogram (hierarchical tree) and use a quality criterion (Davies-Bouldin, ...) to select the most appropriate cut.
- ④ Obtain a segmentation of data.

Algorithm

Example :

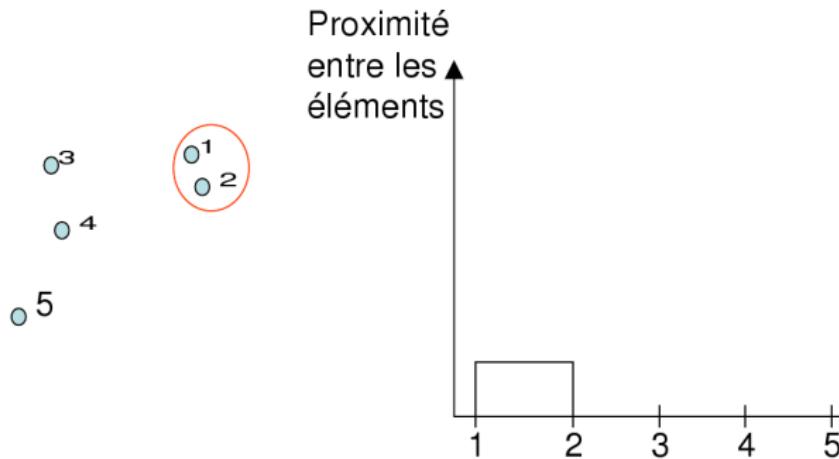
Etape 1 : n individus / n classes



Algorithm

Example :

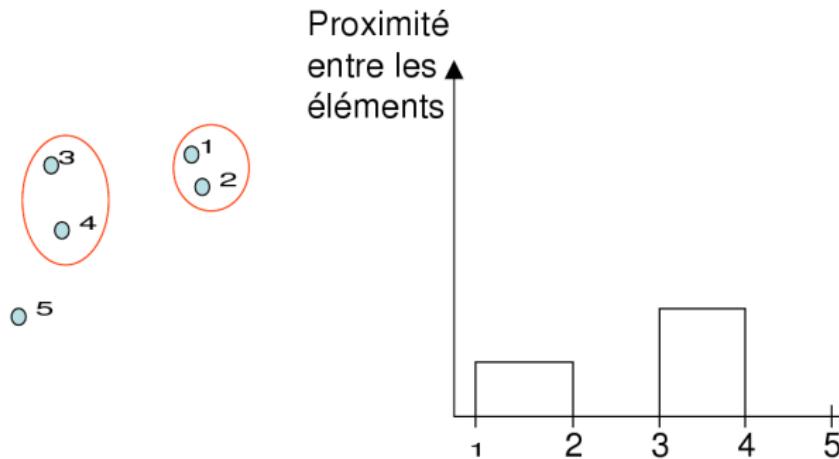
Etape 2 : $n - 1$ classes



Algorithm

Example :

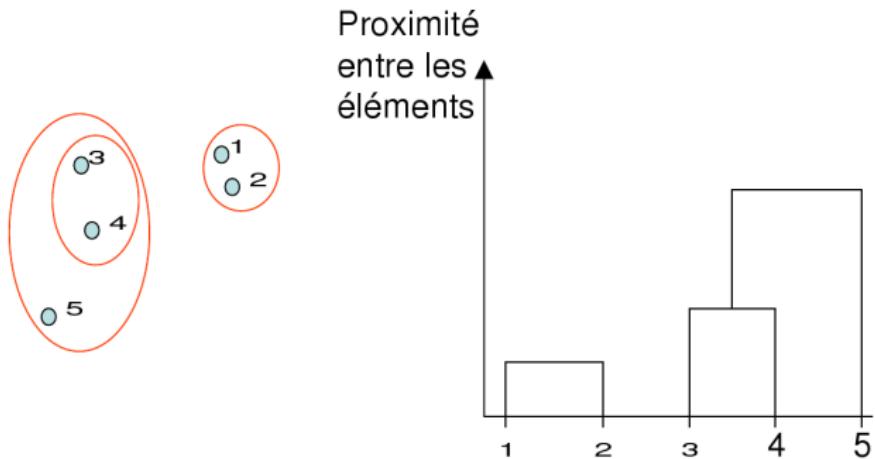
Etape 3 : $n - 2$ classes



Algorithm

Example :

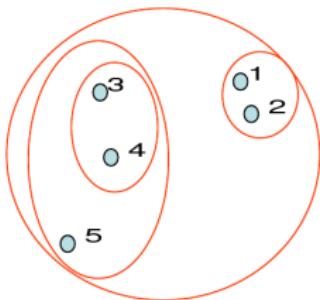
Etape 4 : n = 3 classes



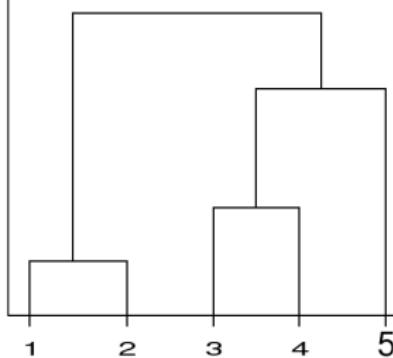
Algorithm

Example :

Etape 5 : $n - 4 = 1$ classe



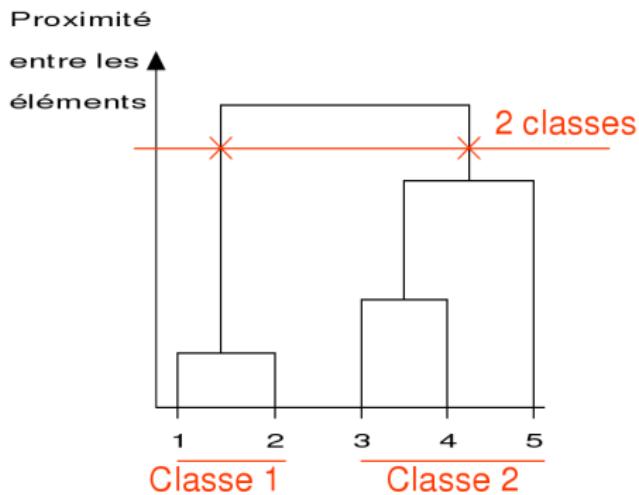
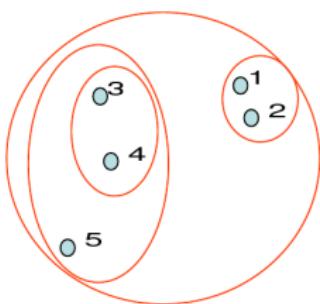
Proximité
entre les
éléments



Algorithm

Example :

Etape 5 : $n - 4 = 1$ classe



Problems

Choice of aggregation rule

This choice is very important, with different rules on different results ! There are many possible rules :

- Distance between the centroids of the two elements.
- Distance between the two nearest data of the two elements.
- Distance between the two most distant data of the two elements.
- Average distance between a data of an element and a data of the other element.

Conclusion

Conclusion

There are many methods of segmentation data. The results depend on :

- Of the algorithm (K-Means, bottom-up methods according to the rule of aggregation, top-down methods, ...).
- Metric (Euclidean distance, Manhattan distance, Minkowski distance, ...).
- The performance index (Davies-Bouldin, Calinski-Harabatz, Silhouette, ...).

However, more groups are compact and well separated, the more different methods will tend to give the same results.