Clustering

Mixture of models and EM algorithm
Spectral Clustering,
K means, and number of clusters

I. Mixture of models and the EM algorithms

The EM algorithm allows to realize a maximum likelihood type approach on mixture models. We will apply the EM algorithm for image segmentation and then regression when several linear behaviors are present in a dataset.

Mixture of Gaussian

- Load the data set `irm_thorax.txt` into the R environment using the following instruction `irm=as.matrix(read.table("irm_thorax.txt",header=F,sep=';'))`.
- Display the image using the function `image()`, then display the histogram of the pixel values? What can we observe ?
- Implement the Expectation-Maximization algorithm on this dataset to segment the color histogram (see course for parameter iteration formulas).
- Display the classification result for 2, 3, then 5 Gaussian. Does the segmentation seem relevant to you?

Mixing regressions by the EM algorithm

- Install, then download the library `mixtools`.
- Load the data set `regression_double.txt`. Display the data. What are you observing? What are the reasons for this type of behavior? Does a simple linear regression seem appropriate to you?
- Mix two linear regressions using the function `regmixEM`.
- Display the result of the regression mixture, and calculate the residuals.
- Make a mixture of two linear regressions by limiting the number of iterations to 1, 3 and then 5 (`regmixEM(y, x, maxit = k)`). Display the prediction error according to the number of iterations and view the classes calculated in each case. What are you observing?

II Spectral Clustering

This document presents the main steps to implement the *Spectral Clustering* (SC) algorithm and to evaluate the impact of the choices of the internal parameters of the results provided by the SC algorithm of simulated data. The R language is used for this practical session.

Creation of data sets

1. Write a R function, called `mysimu()` being able to generate samples of data distributed as shown in the following graphs.

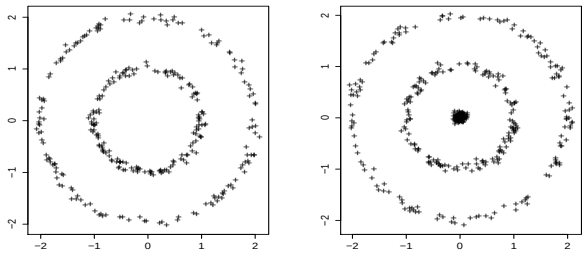

Figure 1: Samples of data to test the *Spectral Clustering* algorithm.

   The `mysimu()` function has 3 inputs:

   - the number of observations generated for each circle (same number of observations for each circle),
   - a list to define the values of the different radius for each circle, and
   - a parameter called σ_c to define the incertainty position of each observation $X_i \sim \mathcal{N}(\mu_i, \sigma_c)$ (in both direction)

2. Using `mysimu()` function, Simulate $n = 450$ observations (150 observations per circle) scattered in 3 circles with respectively $\{0,\ 1,\ 2\ \}$ radius and with $\sigma_c = 0.05$ as illustrated in Figure 1, (left).

Spectral Clustering algorithm

1. Compute a matrix, called Z, wich contains the computed values of the euclidian distances between all the n observations

$$Z = (z_{ij}) \text{ with } 1 \leq i, j \leq n$$

   Indications with R: `dist()`, `as.matrix()`.

2. Compute the affinity matrix, called W, between all the n observations: $W = (w_{ij})$, $1 \leq i, j \leq n$ such that:

$$w_{ij} = exp(\frac{-z_{ij}^2}{2\sigma^2})$$

In this application (for the example with the 3 circles), the parameter σ is first chosen equal to $\sigma = 0.1$

3. Compute the L matrix for the Laplacian graph with the appropriate normalization defined with:

$$L = I_n - D^{-1/2} W D^{-1/2}$$

with I_n the identity (n, n) matrix and
D the (n, n) diagonal matrix defined by : $D = (d_{ij})$ with $d_{ii} = \sum_j^n w_{ij}$, $1 \leq i \leq n$.

4. Implement a Singular Value Decomposition (SVD) of the L matrix such that:

$$L = U_n E_n U_n^T$$

where U_n corresponds to the n eigen vectors of L and
E_n corresponds to diagonal eigen values matrix with $E_n = diag(\lambda_1, \ldots, \lambda_n)$.
$u_j \in \mathbb{R}^n$, is the j^h eigen vector associated with the eigen value λ_j, .
$u_j = (u_{ij})$ with $1 \leq i \leq n$. Indications with R: `eig()`, `svd()`.

5. Store and normalize the first K eigen vectors $u_{(1)}, u_{(2)}, \ldots, u_{(K)}$ corresponding to the smallest eigen values $0 \leq \lambda_{(1)} \leq \lambda_{(2)} \leq \ldots \leq \lambda_{(K)}$ where K correspond to the number of groups supposed to exist in the observations:

$$u_{ij}^* = \frac{u_{ij}}{\sqrt{\sum_{j=1}^K (u_{ij})^2}}$$

6. Use the *Kmeans* algorithm to cluster the n observations u_i^*, where each observations i is characterized with its K coordinates: $u_i^* = (u_{i1}^*, \ldots, u_{iK}^*)$.

7. Using the labels of the K groups computed thanks to the *Kmeans* algorithm, visualize the initial observations and their corresponding labels as represented in Figure 2.

   In order to compare the previous clustering result, apply the *Kmeans* algorithm on the initial raw data. Conclusion .

8. Apply then the Spectral clustering algorithm as in question 2 but with different values of σ $0.05; 0.5; 1$. Conclusion?

9. The *kernlab* library in R proposes a function, `specc()`, which implements the Spectral Clustering algorithm. Study the help of the function.
   For the sample of data corresponding to the "3 circles", what is the value of the σ parameter proposed by thus function ? What is the method used to compute this value ? Conclusion

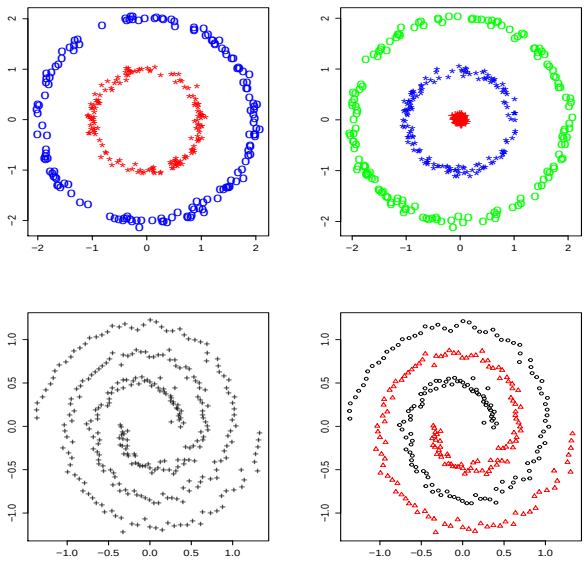

Figure 2: Examples used to cluster data with *Spectral Clustering*.

III Revision. Kmeans Clustering

The kmeans is a method to cluster a set of observations in different groups. This method needs to make an assumption on the number of groups before computing. This method is available, in R, with the `kmeans()` instruction. Before any practical work on data, read the help of the function (`help(kmeans)`).

Clustering and PCA.

- Perform a cluster analysis on the car data using the kmeans function with k=4 groups.
- Describe the following outputs of the function:

  `resk$centers ; resk$cluster ; resk$withinss ; resk$betweenbss`

- Use a PCA analysis (if possible) on this data set, first to visualize the observations on the first factorial plan then to represent the $k = 4$ clusters computed with the kmeans. Comment the graph.
- Run 20 times the kmeans algorithm with scaled data with $k = 8$. Study the value of the within variance for these 20 runs. Conclusion. Modify the call of the function to obtain stable results. Conclusion: what are the practical benefits and drawbacks of the kmeans algorithm ?

IV Revision. Hierarchical Clustering

Application

The file "cardata.txt" contains the data for a set of cars characterized by different features.

Preliminary analysis

1. Have a quick look to your data (blocnote, wordpad)? What can you say?
2. Import the data in the R environment into a data frame structure called X with the help of the instruction `read.table()`. Use the option of the `read.table()` function to define a name for your variables and for your observations. (options `row.names, header`)
3. What is the size n of the data set ? What is the number p of variables ?
   Use the instruction `plot(X)`. What can we see? Compute the correlation matrix `cor(X)` to complete your first analysis. Conclusions.

Hierarchical clustering

The Hierarchical Clustering (HC) is method for clustering data. On the opposite to the kmeans approach, the HC does not need to make any assumption on the number of groups before running the algorithm.

- What do you think about scaling your data ? if necessary use the instruction `scale()`.
- Compute all the euclidian distances between the observations using the `dist()` instruction .
- Perform a first hierarchical clustering with the "Ward" method using the `hclust()` instruction specifying appropriate parameters and represent the dendrogram on a graph.
- Analyze the impact of using other methods as (min, max, moyenne, ward) for the clustering. Represent and compared the computed clustering instances ? Use if necessary the instruction `plot()`, `plclust()` or `hclust()`.
- What does the instruction: `hsc3=cutree(hs,3)` ?
- Use if possible a PCA analysis to represent the observations on the first factorial plan and the computed clusters.

Number of clusters

The goal is here to propose a method to adaptively chose the number of clusters. This section presents several approaches all based on the evolution of the Between and Within variances computed for different number of groups.

Considering a n-sample $y_1, \ldots, y_i, \ldots, y_n$, $y_i \in \mathbb{R}$, with K labels, y_j^k denotes the observation j of group k and n_k the number of observations in group k, we have:

$$\sum_{i=1}^{n}(y_i-\bar{y})^2=\sum_{k=1}^{K}\sum_{j=1}^{n_k}(y_j^k-\bar{y}_k)^2+\sum_{k=1}^{K}n_k(\bar{y}_k-\bar{y})^2$$

We propose the following R instructions and functions to compute the Total, Between and Within variance for a scalar observation:

```
variance<-function(x)
(length(x)-1)/length(x)*var(x)}
```

```
varBetween<-function(x,c)
{m<-tapply(x,c,mean);
l<-tapply(x,c,length);
sum(l*(m-mean(x))^2/length(x))}
```

```
varWithin<-function(x,c)
{v<-tapply(x,c,variance);print(v);
v[is.na(v)]<-0;
l<-tapply(x,c,length);
sum(l*v/length(x));}
```

- Study the previous functions.
- Using the car data set, compute the evolution of the Within and Between variances, for $k = 1$ to $k = 20$.
- Represent, on a graph, the evolution of the Between and the Within variances, function of the number of groups. Use the instruction `matplot(1:n,cbind(VBetween,VWinthin),pch="o")`.
- What is the number of groups for a Between-Within variances tradeoff ? Use the `cutree` instruction to compute the groups and visualize the groups on the dendogram.
- What is the number of groups for a ratio between the Between variance over the total variance equals to 0.95? ? Use the `cutree` instruction to compute the corresponding groups and visualize the groups on the dendrogram.