Data learning 3 : Cluster Analysis

Cluster analysis is an exploratory data analysis tool for solving classifaction problems. Its object is to sort individuals (people, things, events, ...) into groups, on clusters, so that the degree of association is strong between members of the same cluster and weak between members of different clusters. Among a large unstrutured data set, it may reveal associations and structure in data which, thought not previously evident, nevertheless are sensible and usefull once found. Clustering a data set consists in finding the *best* partition of the set in the sens of a criterium that increase together when association between members increases and when association between different clusters decreases. Notice that it is impossible to look at all the partitions of the set and choose the best one because the number of partitions is much to large, even for very big computers (there exist for example more than 10 millions of partitions of 14 individuals in 4 clusters). Thus one can not hope better than finding an as good as possible partition by iterative methods. There exists mainly two kind of alogorithms, namely *hierarchical aglomeration* (that produce special hierarchical trees called *dendrogram*) and *partitioning* (which will gives us the first opportunity to see an example of *unsupervised learning algorithm*).

Distance matrix : Usually, the n individuals that we want to classify are known through the values of p variables that have been mesured on each of them. Thus the data set is usually as in lecture 1 a data matrix $X = (x_i^j)_{1\leq i\leq n, 1\leq j\leq p}$. But the first thing to do to be able to classify individuals is to choose a distance (or more generally a dissimilarity)between two indisviduals, namely $d(x_i, x_{i'})$. Than one can fill an $n \times n$ symetric matrix,

$$\Delta = (\delta_{ii'})_{1\leq i,i'\leq n} = d(x_i, x_{i'})$$

call the *distance matrix*. The choice of the distance is important because the result of the clustering analysis usually depends on it. If the individuals may be considered as elements of a p dimensional subspace (that is if the variables are non categorical) than the most commonly chosen type of distance is the euclidien distance $d(x_i, x_{i'})^2 = \sum_{j=0}^{p}(x_i^j - x_{i'}^j)^2$. But it can also be the *city-block distance* $d(x_i, x_{i'}) = \sum_{j=0}^{p}|x_i^j - x_{i'}^j|$ or more generally the *power distance* $d(x_i, x_{i'})^r = \sum_{j=0}^{p}(x_i^j - x_{i'}^j)^s$, where r and s are user-defined parameters.

Huygens' *F*ormula : Assume that the set of individuals Γ is the union of q clusters Γ_1, Γ_2, $\ldots\Gamma_q$ and denote by $\mathcal{I}(\Gamma)$ and $\mathcal{I}(\Gamma_k)$, for $k = 1, \ldots, q$, the inertia of the set Γ and the inertia of his clusters. Let us call the sum of the inertia of all clusters

$$\mathcal{I}_{within}(\Gamma) = \mathcal{I}(\Gamma_1) + \ldots + \mathcal{I}(\Gamma_q)$$

the *within-clusters inertia* of Γ. Let $\overline{x}_k$ be the centroïd of Γ_k and π_k its weight, wich is equal to the sum of the weight ω_i of the individuals belonging to Γ_k. Then the inertia of the set of the weighted points $(\overline{x}_k, \pi_k)$ is called the *between-clusters inertia* of Γ, denoted by $\mathcal{I}_{between}(\Gamma)$. We have the following result called the *Huygens' Formula :*

Proposition 1 *The total inertia $\mathcal{I}(\Gamma)$ of a cloud of points wich is the union of the clusters Γ_1, Γ_2, $\ldots\Gamma_q$ is the sum of its within-cluster inertia and its between-clusters inertia :*

$$\mathcal{I}(\Gamma) = \mathcal{I}(\Gamma_1 \dot\cup \Gamma_2 \dot\cup \ldots \dot\cup \Gamma_q) = \mathcal{I}(\Gamma_1) + \mathcal{I}(\Gamma_2) + \ldots + \mathcal{I}(\Gamma_q) + \mathcal{I}_{between}(\Gamma) = \mathcal{I}_{within}(\Gamma) + \mathcal{I}_{between}(\Gamma).$$

Verify the formula for 1-dim cloud and generalize it to any dimension using Pythagoras' formula.

One consequence of this theorem is that, as the total inertia $\mathcal{I}(\Gamma)$ does not depend on the partition of Γ into clusters, than a partition that maximize the within-clusters inertia (the association within members of a cluster been as strong as possible) will automatically minimize the between-clusters inertia (the association between members of different clusters been as low is possible).

Hierarchical clustering (HC) : One begins with every individuals representing a singleton cluster and at each step the closest two clusters are merged in a single cluster producing one less cluster at the next higher level. Therefore a measure of dissimilarity between clusters must be defined to choose which clusters

are the closest. There is a lot of possibilities, for exemple, if Γ_k and $\Gamma_{k'}$ are two clusters, $D(\Gamma_k, \Gamma_{k'}) = \text{Min}\,\{d(x_i, x_{i'}), x_i \in \Gamma_k, x_{i'} \in \Gamma_{k'}\}$ is called the *single linkage*, $D(\Gamma_k, \Gamma_{k'}) = \text{Max}\,\{d(x_i, x_{i'}), x_i \in \Gamma_k, x_{i'} \in \Gamma_{k'}\}$ is called the *complete linkage*, but one should preferably used the *Ward linkage* defined by

$$D(\Gamma_k, \Gamma_{k'}) := \frac{\pi_k \pi_{k'}}{\pi_k + \pi_{k'}} d(\overline{x}_k, \overline{x}_{k'}).$$

Indeed it can be shown that this distance measure precisely the loss of between-clusters inertia produced by the union of the two clusters in a single one (or the gain of within-clusters inertia).

Dendrogram : Usually one represents the HC by a binary tree, the root of which represent the initial data set Γ, each node represents a cluster of Γ, the terminal nodes represent the individuals. Each non terminal node (parent) has two daughter nodes that represent the two clusters that were merged to from the parent. The height of each node is proportional to the value of the distance between its two daughters. Thus with the choice of the Ward linkage the height in the dendrogram is simply proportional to the within-clusters inertia of the data set Γ. Notice that at the first step of the algorithm the within-clusters inertia is 0 and the between-clusters inertia is 1. But at the end it is the converse. At each step, the one decreases and the other increases and the algorithm chooses to merge two clusters in such a way to produce the maximal loss of between-clusters inertia (or to produce the maximal gain of within-clusters inertia). It is up to the user to decide how to cut the dendrogram, that means to decide which level (if any) actually represents the *best* clustering of the initial data set.

K-mean clustering methods : For these methods, also called *dynamic centers* methods) we have to figure out first how many clusters we want. Let q be this number of clusters. The simplest algorithm of this family starts with a ramdom sample of q points (initialisation of the *centers*) belonging to the data set Γ

$$\{c_1^0, c_2^0, \ldots, c_q^0\}$$

From these points one can defined a first partition of the set $\Gamma = \Gamma_1 \dot{\cup} \Gamma_2 \dot{\cup} \ldots \dot{\cup} \Gamma_q$, just puting in the same cluster Γ_k the individuals that are closer to c_k^0 than to the other $c_{k'}^0$. This means that, for all $k = 1, 2, \ldots, q$, the cluster Γ_k is given by

$$\Gamma_k = \{x_i / d(x_i, c_k^0) = \text{Min}\,\{d(x_i, c_{k'}^0), k' = 1, 2, \ldots, q\}.$$

Then on replace each the center c_k^0 by the centroïds of the cluster Γ_k called

$$\{c_1^1, c_2^1, \ldots, c_q^1\}.$$

For the next step of the algorithm repeat what as be done in the previous step but in begin with the new centers. Each step produces a new partition of the data set and the important point is to shows that the new one corresponds to a within-cluster inertia lower (or equal) to than the previous one. Usually the user will stop the algorithm either if two successive steps induce no new modification of the partition or if the within-clusters inertia does no longer decreas noticeably (or may be after a given number of time steps).

This algorithm belongs to the family of *iterative descent methods* because it move at each step from one partition of the set of all partitions of Γ to another in such a way that the value of some criterion (here the within-clusters inertia of the set) improve from its previous value. As the set of all partitions of Γ is finite (even if it is huge), it can be shown that this algorim converges. But its important to know that it usually converges to some local optima wich may be highly suboptimal when compared to the global optimum. For the user, this problem will probably appear when he will realize that the optimum given by the algorithm do depend on the choice of the initial sample of centers $c_1^0, c_2^0, \ldots, c_q^0$.

This is one reason to introduce a modified algorithm called *dynamic K-means method*. As for the simple K-means algorithm, one begins with q centers $c_1^0, c_2^0, \ldots, c_q^0$ chosen as a random sample and then, at each step, only one new point x_i of Γ is chosen, it is assign to one of the existing centers, say c_k^m if it is the m^{th} step, and then one replace this center $c_k^m$by the centroïd of the set $\{x_i, c_k\}$ called c_k^{m+1}. We will see in a future lecture that this modified algorithm can be considere as a *statistical learning* method and that it can be generalized in the so called *neural network approach*.