Clustering

Luca Martino*

* Universidad Rey Juan Carlos, Madrid (Spain).

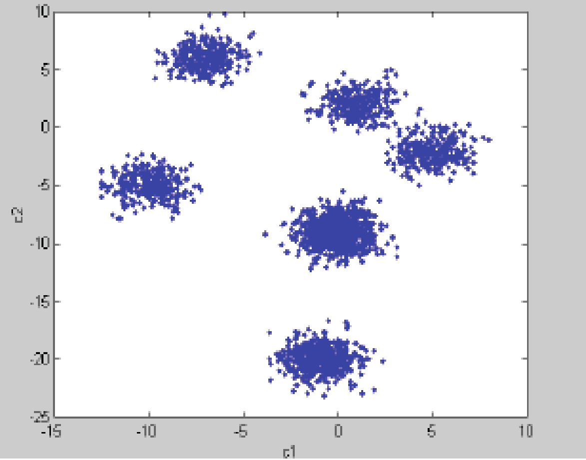
MAGICMOTORSPORT - COURSE

Clustering

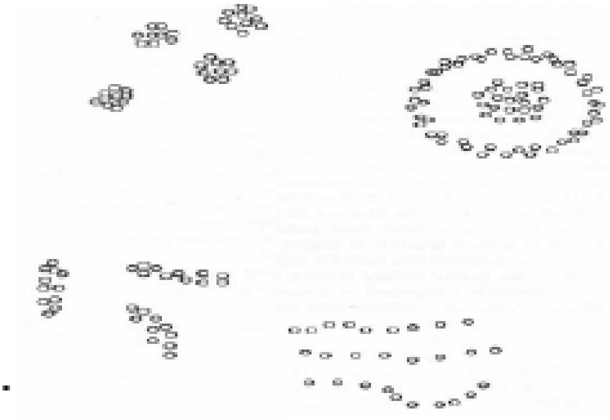
- Now, an **unsupervised task**.
- we have just $\mathbf{x}_i$...
- We will see the k-**means algorithm** which is, in my opinion, the unsupervised version of the Nearest Neighbors (NNs) method.
- in this slides: number of data N.

What is clustering?

Find the groups of samples and, if it is possible, **The number of groups.**

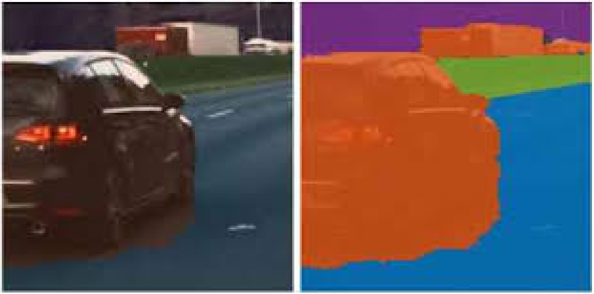


Type of clusters.... infinite....



Examples of application

Segmentation

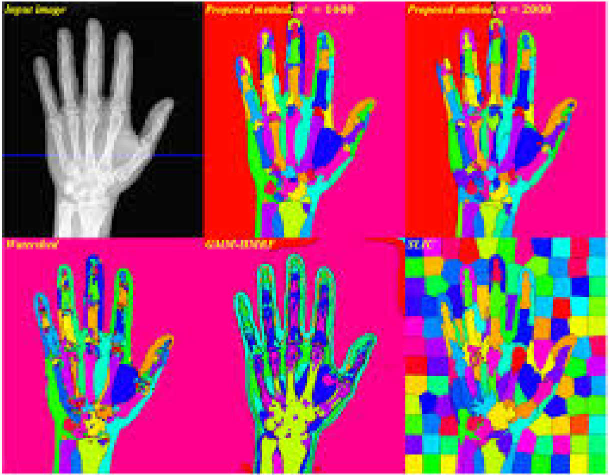


Examples of application

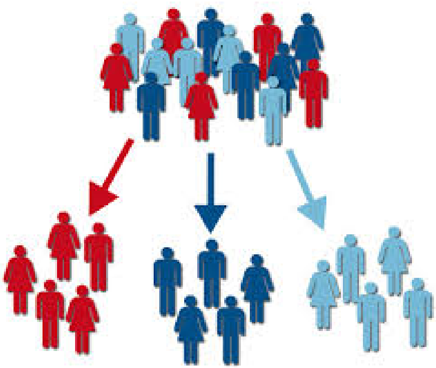
Segmentation

Examples of application

Segmentation

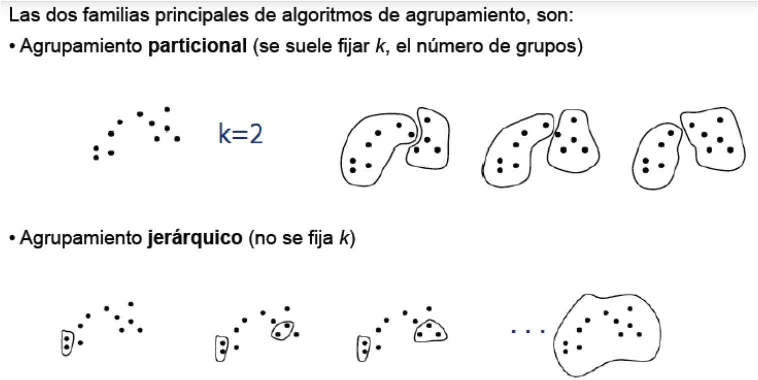


Examples of application



Main families of clustering algorithms

(a) partitional (partitioning) clustering
(b) hierarchical clustering etc...



The most famous clustering algorithm

The k-means algorithm.

k is the number of clusters....

Note that

$$1 \leq k \leq N$$

If we set $k = N$, each data should be a cluster....

k-means algorithm

- Fix a number k (of clusters).

- Start randomly, choosing the positions of k **centroids.**

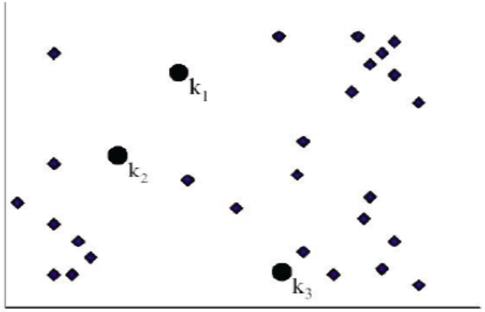
- Assign samples/data to each centroid as function of some distance.

- Move the centroids, doing the arithmetic means of the assigned samples/data....

k-means algorithm: starting ....

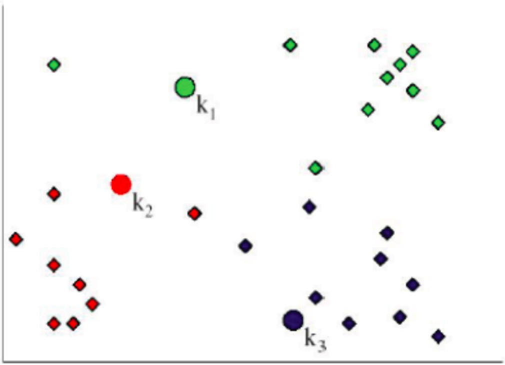
Fix a number k (of clusters).

Ejemplo del algoritmo *k*-medias (*k-means*) con *k*=3



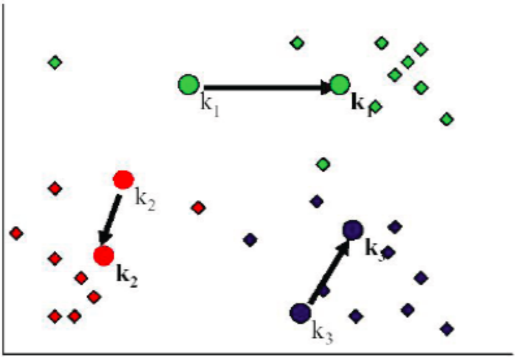
k= 3 centroides: k1, k2, k3

k-means algorithm: assignment/distribution step according to the distances....



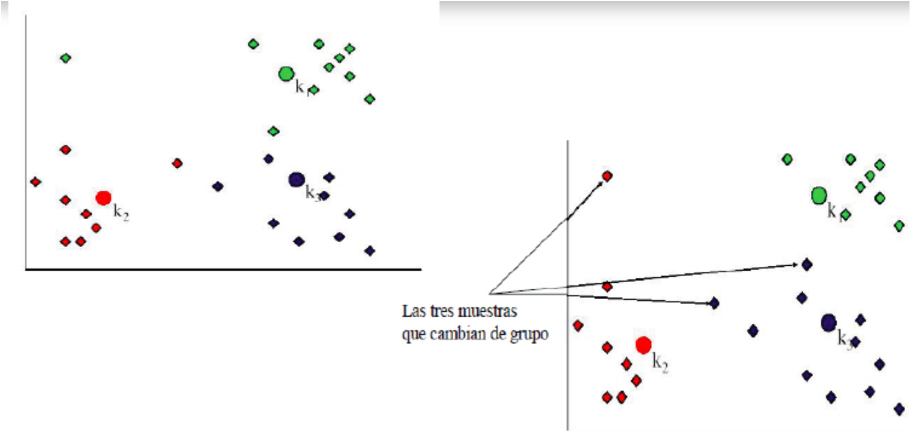
Se asigna cada muestra al centroide más cercano.
Cada color representa un *cluster* distinto.

k-means algorithm: moving step according to the means of the assigned samples...



Se recalcula la posición de cada centroide como el promedio de las muestras/observaciones/ejemplos de cada *cluster*.

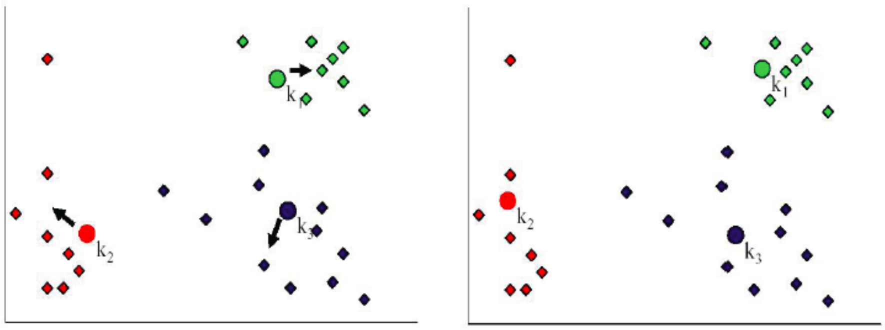
k-means algorithm: again, assign ...



Se reasignan las muestras al centroide más cercano

k-means algorithm: and move... and repeat...

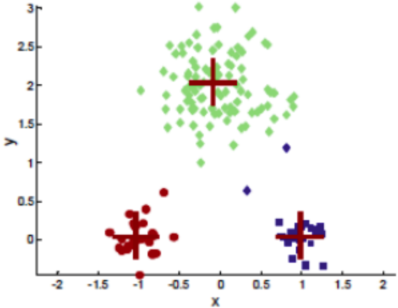
Recalcular los centros de los *clusters*



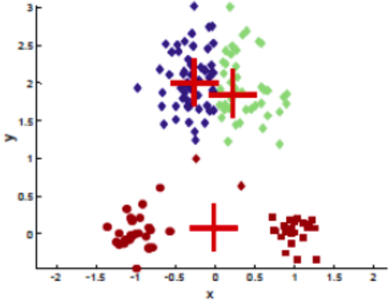
Reasignar las muestras al cluster más cercano ...

El criterio para detener el algoritmo puede ser un número máximo de iteraciones, la estabilización en la posición de los centroides, ...

Possible results



Solución óptima



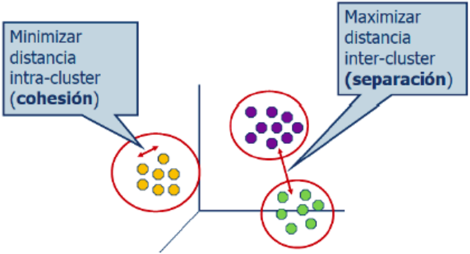
Posible resultado proporcionado por k-means

Are we minimizing a cost function? Yes...

We are minimizing the sum of the distances "inside each cluster"

Al considerar la distancia Euclídea y la media aritmética, se está **minimizando** la **variación *intra-cluster***. Esta medida se usa, por tanto, como criterio del grado de ajuste (cohesión y separación) de los centroides.

Ejemplo con observaciones de tres dimensiones



Función a optimizar

$$\sum_{i=1}^{k} \frac{1}{|C_i|} \sum_{\underline{x}\in C_i} d^2\left(\underline{x}, \underline{c}_i\right)$$

|.| Cardinalidad del conjunto (número de observaciones)

C_i: *cluster* i-ésimo

$\underline{c}_i$: *centroide* i-esimo

Relationship with density estimation

We are minimizing the sum of the distances "inside each cluster"

This is related to density estimation, and variances (of components within a mixture of densities)....

It is can be shows that we are looking for a "good" mixture of Gaussians describing the data....

“Variance” inside the cluster

If the distance is Euclidean, we are minimizing the “variance” inside the cluster.

“... the basic idea behind partitioning methods, such as k-means clustering, is to define clusters such that the total intra-cluster variation (or total within-cluster sum of square) is minimized.”

"Variance" inside the cluster

If the distance is Euclidean, we are minimizing the "variance" inside the cluster.

j-th centroid $\mathbf{c}_j$ of the j-th cluster C_j, then we want to minimize

$$\mathrm{Var}[\mathbf{x} \in C_j] = \text{variance inside } C_j \approx \frac{1}{|C_j|} \sum_{i \in C_j} ||\mathbf{x}_i - \mathbf{c}_j||^2.$$

"Law of Total Variance"....

Clear the mean of all the centroids is the mean of the data

$$
\begin{aligned}
\boldsymbol{\mu} &= \frac{1}{k}\sum_{j=1}^{k}\mathbf{c}_j = \frac{1}{k}\sum_{j=1}^{k}\left(\frac{1}{|C_j|}\sum_{i\in C_j}\mathbf{x}_i\right), \\
&= \frac{1}{N}\sum_{i=1}^{M}\mathbf{x}_i,
\end{aligned}
$$

Law of Total Variance:

$$
\begin{aligned}
\mathrm{Var}[\mathbf{x}] &= \left(\sum_{j=1}^{k}\mathrm{Var}[\text{inside the cluster } C_j]\right) + \mathrm{Var}[\text{of "centroids"}], \\
&= \sum_{j=1}^{k}\underbrace{\mathrm{Var}[\mathbf{x}\in C_j]}_{\text{internal}} + \underbrace{\sum_{j=1}^{k}||\mathbf{c}_j - \boldsymbol{\mu}||^2}_{\text{among the centroids}} .
\end{aligned}
$$

"Law of Total Variance"....

as a consequence:

$$\mathrm{Var}[\mathbf{x}] \geq \sum_{j=1}^{k} \mathrm{Var}[\mathbf{x} \in C_j]$$

and it is also valid for the approximations:

$$\frac{1}{N} \sum_{i=1}^{N} ||\mathbf{x}_i - \boldsymbol{\mu}||^2 \geq \sum_{j=1}^{k} \left(\frac{1}{|C_j|} \sum_{i \in C_j} ||\mathbf{x}_i - \mathbf{c}_j||^2 \right)$$

Recall that

$$\boldsymbol{\mu} = \frac{1}{k} \sum_{j=1}^{k} \mathbf{c}_j = \frac{1}{k} \sum_{j=1}^{k} \left(\frac{1}{|C_j|} \sum_{i \in C_j} \mathbf{x}_i \right)$$

as k grows...

Considering that k grows approaching N:

$$\sum_{j=1}^{k} \mathrm{Var}[\mathbf{x} \in C_j] \to 0,$$

and

$$\sum_{j=1}^{k} ||\mathbf{c}_j - \boldsymbol{\mu}||^2 \to \mathrm{Var}[\mathbf{x}].$$

When $k = N$ (with a proper clustering: each data is a cluster),

$$\sum_{j=1}^{k} \mathrm{Var}[\mathbf{x} \in C_j] = 0, \quad \sum_{j=1}^{k} ||\mathbf{c}_j - \boldsymbol{\mu}||^2 = \mathrm{Var}[\mathbf{x}].$$

when $k = 1$...

When $k = 1$: all the data in one unique cluster,

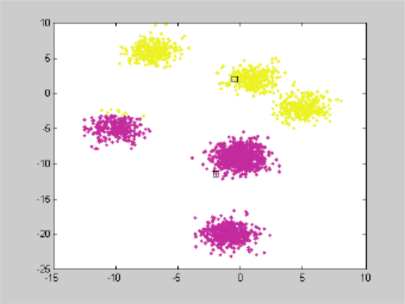
$$\sum_{j=1}^{k} \mathrm{Var}[\mathbf{x} \in C_j] = \mathrm{Var}[\mathbf{x} \in C_1] = \mathrm{Var}[\mathbf{x}],$$

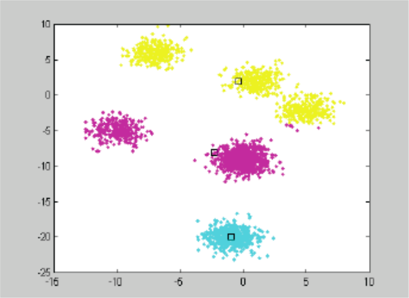
$$\sum_{j=1}^{k} ||\mathbf{c}_j - \boldsymbol{\mu}||^2 = ||\mathbf{c}_1 - \boldsymbol{\mu}||^2 = 0.$$

The k-means algorithm is a parametric method

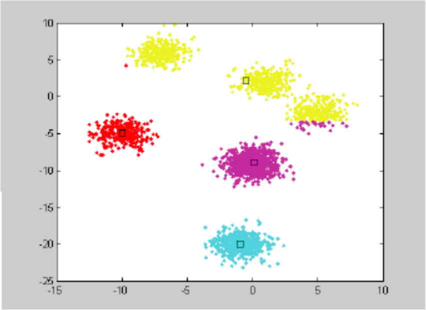
- Note that the k-means algorithm is a **parametric** method.
- We fix k and then decide....

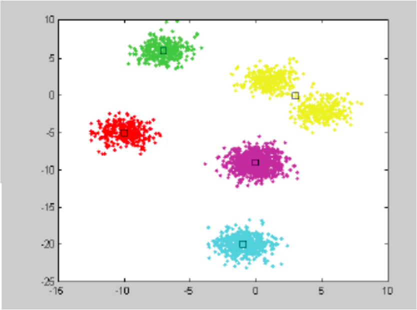
How many clusters? starting with $k = 2$ and then increase k...



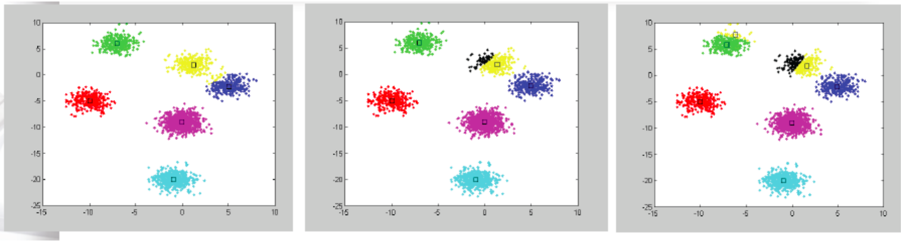


How many clusters? starting with $k = 2$ and then increase k...





How many clusters? starting with $k = 2$ and then increase k...



and then use some criterium for the "optimal performance"...

Note that

$$1 \leq k \leq N$$

With $k = N$, each data is a cluster....
($k = 1$ could/should be underfitting)
($k = N$ could/should be overfitting)

However, in this unsupervised case, it is not "easy", straightforward, to apply Cross-Validation (CV)

and then use some criterium for the “optimal performance”...

- We could use “marginal likelihood with probabilistic approaches.....”
- Now we will see two methods:
  - Elbow method + AIC (Akaike information criterion)
  - Silhouette method

Elbow method + AIC (for deciding k...)

Find k^* which minimizes the following cost function

$$\mathrm{Cost}(k) = \sum_{j=1}^{k} \mathrm{Var}[\mathbf{x} \in C_j] + k$$

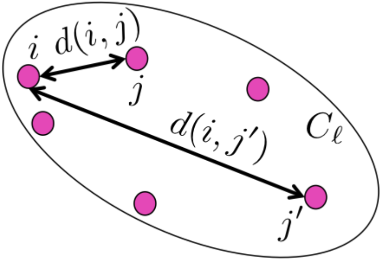
$$\mathrm{Cost}(k) = \underbrace{\sum_{j=1}^{k} \mathrm{Var}[\mathbf{x} \in C_j]}_{\texttt{fitting}} + \underbrace{k}_{\texttt{model penalty (AIC)}}$$

When k grows: the first term $\sum_{j=1}^{k} \mathrm{Var}[\mathbf{x} \in C_j]$ decreases, and the model penalty grows (k)

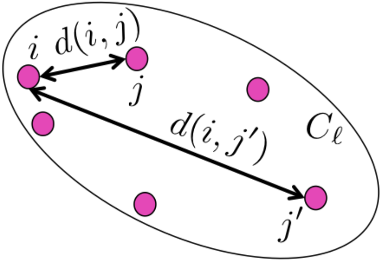
Silhouette method (for deciding k...)

- Consider the ℓ-th cluster C_ℓ.
- For each i-th point in the ℓ-th cluster C_ℓ, then $i = 1, ..., |C_\ell|$, we compute

$$a(i) = \frac{1}{|C_\ell| - 1} \sum_{j \in C_\ell; j \neq i} d(i, j).$$



Silhouette method (for deciding k...)



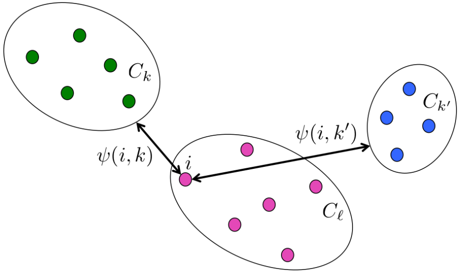
- $a(i)$ **measures how dissimilar is i-th data to its own cluster....**
- high $a(i)$ $==>$ great dissimilarity

Silhouette method (for deciding k...)

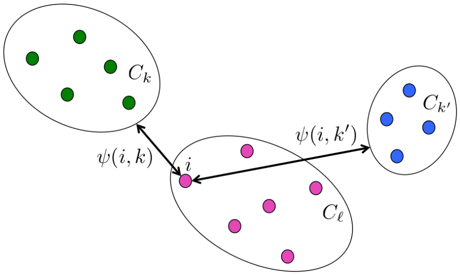
- For each i-th point in the ℓ-th cluster C_ℓ, then $i = 1, ..., |C_\ell|$, we also compute

$$\psi(i,k) = \frac{1}{|C_k|}\sum_{j\in C_k} d(i,j), \qquad \text{with } k \neq \ell.$$

$$b(i) = \min_k \psi(i,k), \qquad \text{with } k \neq \ell.$$



Silhouette method (for deciding k...)



- high $b(i)$ $==>$ the other clusters are different and do not explain the i-th data; the i-th data is not similar to other cluster ....

Silhouette method (for deciding k...)

We now define a *silhouette* (value) of one data point i

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \text{ if } |C_i| > 1$$

and

$$s(i) = 0, \text{ if } |C_i| = 1$$

Which can be also written as:

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases}$$

From the above definition it is clear that

$$-1 \leq s(i) \leq 1$$

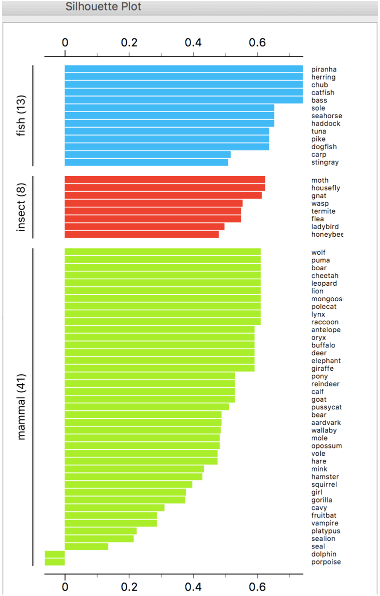
Silhouette method (for deciding k...)

Also, note that score is 0 for clusters with size = 1. This constraint is added to prevent the number of clusters from increasing significantly.

For $s(i)$ to be close to 1 we require $a(i) \ll b(i)$. As $a(i)$ is a measure of how dissimilar i is to its own cluster, a small value means it is well matched. Furthermore, a large $b(i)$ implies that i is badly matched to its neighbouring cluster. Thus an $s(i)$ close to one means that the data is appropriately clustered. If $s(i)$ is close to negative one, then by the same logic we see that i would be more appropriate if it was clustered in its neighbouring cluster. An $s(i)$ near zero means that the datum is on the border of two natural clusters.

- $s(i) \approx 1$ then the i-th data has been properly clustered.
- $s(i) \approx 0$ then the i-th data could belong to different clusters....
- $s(i) \approx -1$ then the i-th data should belong to another cluster....

Silhouette method (for deciding *k*...)



Silhouette method (for deciding k...)

Consider the mean $s(i)$ inside one cluster C_ℓ,

$$\bar{s}_\ell = \frac{1}{|C_\ell|} \sum_{i \in |C_\ell|} s(i).$$

if $\bar{s}_\ell$ is high $==>$ the points in the cluster are well-grouped (i punti del cluster C_ℓ sono effettivamente simili tra loro)

Silhouette method (for deciding k...)

Consider now the mean of $s(i)$ over all the clusters,

$$\begin{aligned}\widetilde{s}(k) &= \frac{1}{k}\sum_{\ell=1}^{k}\bar{s}_\ell = \frac{1}{k}\sum_{\ell=1}^{k}\left(\frac{1}{|C_\ell|}\sum_{i\in|C_\ell|} s(i)\right). \\ &= \frac{1}{N}\sum_{i=1}^{N} s(i),\end{aligned}$$

is a measure of how our clustering is “good”.

Note that $\widetilde{s}(k)$ depends on the number of clusters that we choose at the beginning.

Silhouette method (for deciding k...)

Choose the number of clusters such that

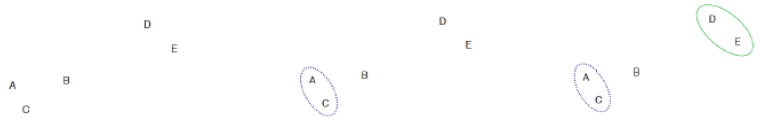
$$k^* = \arg\max_k \widetilde{s}(k).$$

Recall that $\widetilde{s}(k)$ represents the mean of the $s(i)$ over all the data of the entire dataset for a specific number of clusters k.

(agglomerative) Hierarchical clustering

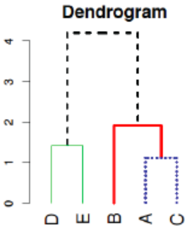
we always consider the two closest “clusters” or “super-clusters”.

Hierarchical clustering starts by treating each observation as a separate cluster. Then, it repeatedly executes the following two steps: (1) identify the two clusters that are closest together, and (2) merge the two most similar clusters. This iterative process continues until all the clusters are merged together. This is illustrated in the diagrams below.

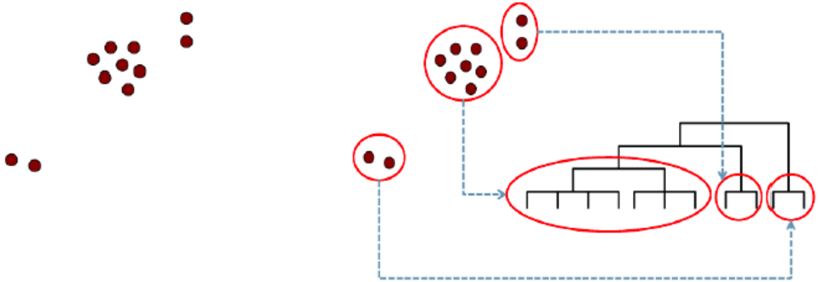


Hierarchical clustering

different resolutions....

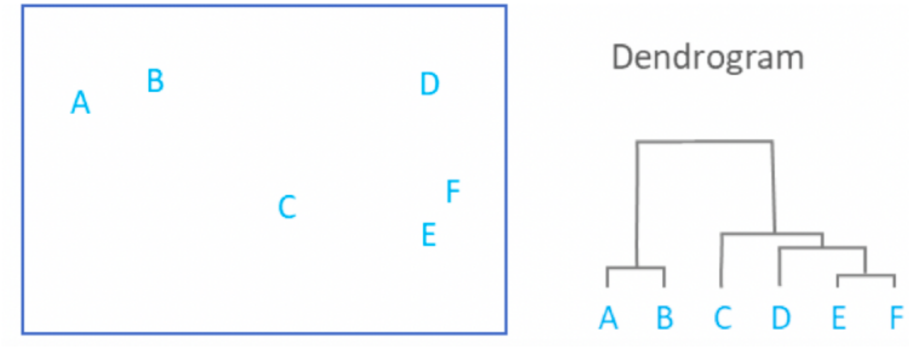


Hierarchical clustering

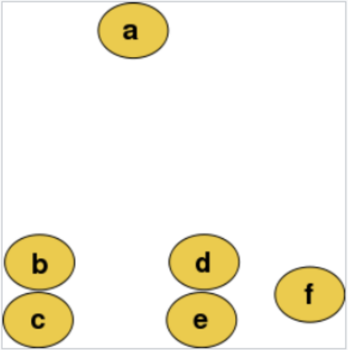


Hierarchical clustering

The main output of Hierarchical Clustering is a *dendrogram*, which shows the hierarchical relationship between the clusters:

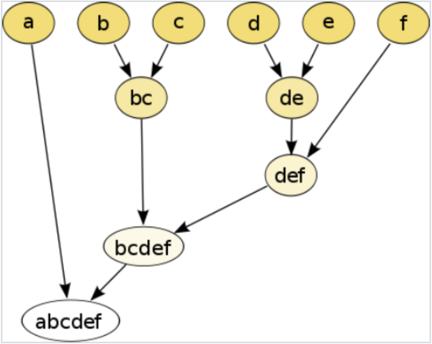


Hierarchical clustering



Raw data

The hierarchical clustering dendrogram would be as such:



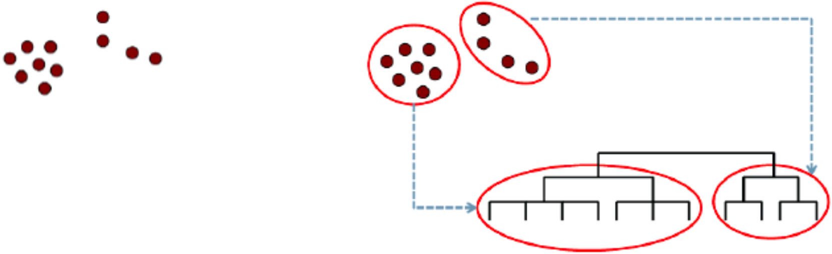
Traditional representation

Hierarchical clustering

Agglomerative versus divisive algorithms

Hierarchical clustering typically works by sequentially merging similar clusters, as shown above. This is known as *agglomerative hierarchical clustering*. In theory, it can also be done by initially grouping all the observations into one cluster, and then successively splitting these clusters. This is known as *divisive hierarchical clustering*. Divisive clustering is rarely done in practice.

Hierarchical clustering

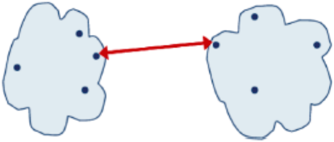


Hierarchical clustering

How do you consider a “distance” between clusters? (linkage)

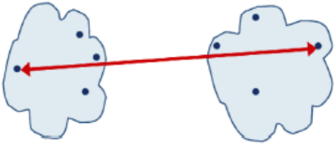
Single-link

Mínima distancia/disimilitud *inter-cluster*



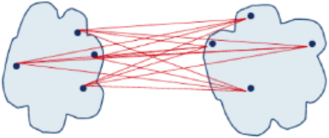
Complete-link

Máxima distancia/disimilitud *inter-cluster*



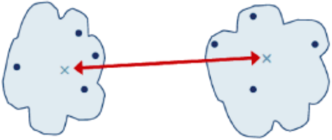
Average

Distancia/disimilitud media *inter-cluster*

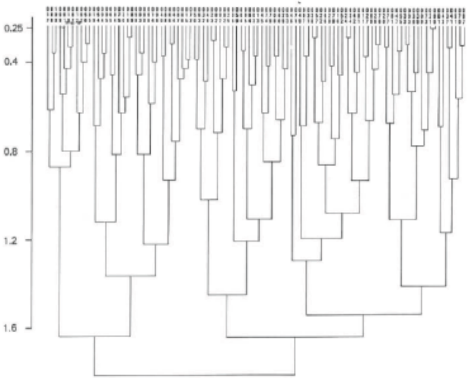


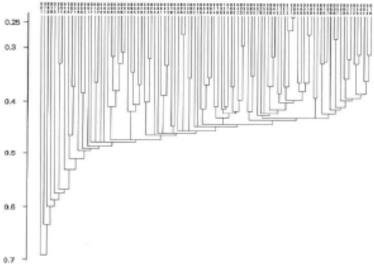
Centroids

Disimilitud o distancia entre centroides

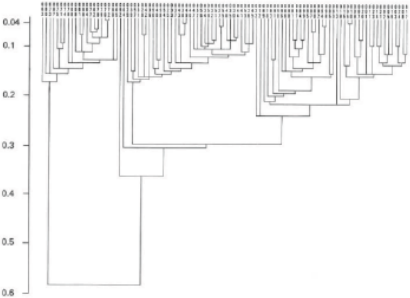


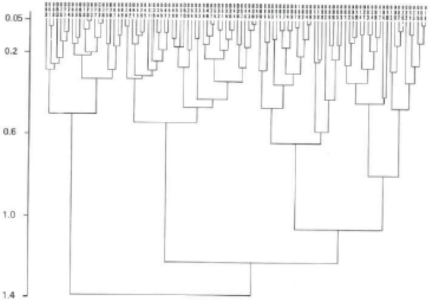
Hierarchical clustering





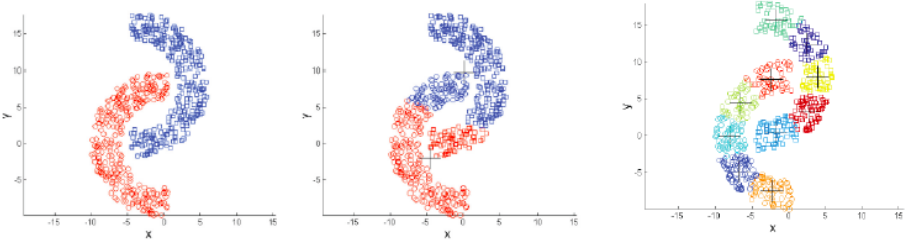
Hierarchical clustering





Hierarchical clustering

Better with non-convex stuffs....



Agrupamiento jerárquico 2-medias 10-medias