CLUSTER ANALYSIS

Definition of clustering

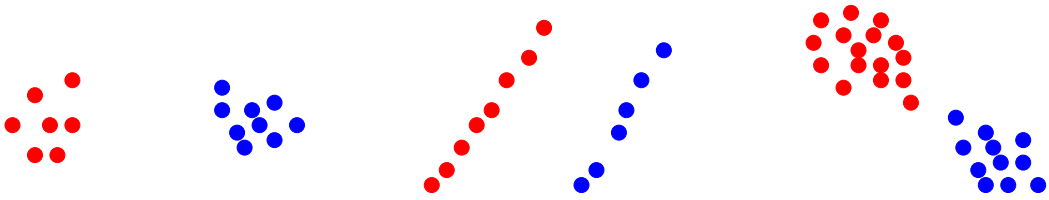
Given a collection of N objects, $X = \{x_1, \ldots, x_N\}$, one seeks a partition of X into K nonempty disjoint sets (the *clusters*),

$$X = \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_K$$

such that, *given the notion of resemblance considered*, it

- maximizes the internal homogeneity or cluster cohesion, or equivalently, it minimizes the intra-cluster variability - objects belonging to the same cluster should share the similar features
- it maximizes the external heterogeneity or cluster separation, i.e., it maximizes the inter-cluster separability - objects belonging to distinct clusters should be very dissimilar and have clear distinguished features

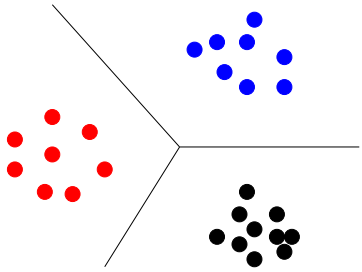
Examples



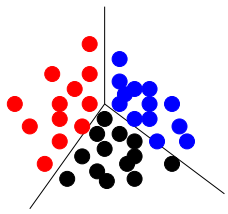
strong internal cohesion
strong separation

weak internal cohesion
strong separation

strong internal cohesion
weak separation



clear clustering structure



artificial clustering structure

Clustering

- Clustering always imposes some kind structure on the data, even when no special structure or discontinuities are present!

  For instance, many clustering techniques tend to form globular clusters, e.g., with elliptical or spherical shapes

- How to choose the best partition ?

Huge solution space...

The number of distinct partitions of N elements into K clusters $(1 \leq K \leq N)$ equals

$$\xi(N, K) = \frac{1}{K!} \sum_{j=1}^{K} \binom{K}{j} (-1)^{K-j} j^N,$$

which is a huge number, known as Stirling number of second kind, even for relatively small values of N and K, making impossible to to find the best partition by exhaustive search.

For instance, the number of partitions of a set with 25 elements into 8 clusters equals

$$\xi(25, 8) = 69022372111836858$$

For N large and K fixed we have

$$\xi(N, K) \approx \frac{K^N}{K!}$$

In the previous example, one gets $\xi(25, 8) \approx \frac{8^{25}}{8!} = 9.369775e{+}17$

Common steps in a cluster analysis

Variables/features selection

- Which type of variables (continuous,categorical, ordinal,binary, . . .), encode as much as possible the information concerning the task, avoiding redundancy (i.e., highly correlated variables) ?
- Standardize/normalize the variables to balance their importance ?

Clustering model

- Which combination of a clustering method with a distance/dissimilarity is more appropriate?

Cluster validation

- **Internal**: How many groups and how to assess the quality of the clusters ?
- **External**: How the clustering results compare with the outcomes obtained using different clustering models or how they compare with known information ?

Interpretation of the results

- Are the outcomes interpretable in the context of the problem ?
- Which variables/features (active/supplementary) are more important to characterize the clusters ?

Cluster model

A **cluster model** is build upon two concepts:

- the notion of distance/dissimilarity between individuals and clusters should be adequate to the type of variables involved and to the type of results sought
- the clustering method should take into account the type of structure/shape of the clusters sought (rounded shape/arbitrary shape/...) and characteristics of the method itself (sensitivity to outliers/noise/ldots), computational issues (scalability for large datasets), etc...

When several cluster models are appropriate one should compare the outputs of such models to seek for common patterns that emerge from these clustering models - **robust solutions**

Example of numerical dataset - iris flower dataset

The well known iris flower dataset contains the sepal and petal lengths and widths (in cm) of 150 iris flowers

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5.0	3.6	1.4	0.2
6	5.4	3.9	1.7	0.4
7	4.6	3.4	1.4	0.3
8	5.0	3.4	1.5	0.2
9	4.4	2.9	1.4	0.2
10	4.9	3.1	1.5	0.1

- How to measure the distance between each pair of iris flowers ?
- Standardize (z-score normalization) or normalize (min-max scaling) the variables in order that the differences between all variables contribute equally ?

Example - a freshwater fish dataset in West Africa

In the biogeography it is common to use biological markers (e.g., river fish species) to distinguish between sites (e.g., river basins)

	annectens	ansorgi	bichir	endlicheri
GAMBIE	1	0	1	0
GEBA	0	1	1	1
CRUBAL	0	1	0	0
KONKOURE	0	0	0	0
KOLENTE	0	0	0	0
LSCARC	0	0	0	0
ROKEL	0	0	0	0

- Which type of variable/feature is more appropriate to encode this type data ?
- How to assess the similarity between river basins given the distribution of fish species ?
- How to assess the similarity between fish species given their distribution by the sites ?

Example of a categorical dataset

The following two-way contingency table encodes the country of residence and primary language spoken by 1000 inhabitants in 5 countries

	English	French	Spanish	German	Italian	Total
Canada	688	280	10	11	11	1000
USA	730	31	190	8	41	1000
England	798	74	38	31	59	1000
Italy	17	13	11	15	944	1000
Switzer.	15	222	20	648	95	1000
Total	2248	620	269	713	1150	5000

Source: `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3718710/`

- How to assess the similarity between countries given the languages spoken in these countries ?
- How to assess the similarity between the spoken languages given their distribution by the countries ?

Properties of a dissimilarity measure / distance

In order to tackle the previous questions we first need to establish which properties a dissimilarity/distance notion should have.

A **dissimilarity measure** on a set X is a real function

$$d : X \times X \to \mathbb{R},$$

such that, for all $x, y \in X$, we have

- $d(x, y) \geq 0$
- $d(x, y) = 0$ if and only if $x = y$
- $d(x, y) = d(y, x)$

We call d a **distance** if moreover d verifies the triangle inequality

- $d(x, z) \leq d(x, y) + d(y, z)$ for all $x, y, z \in X$,

Three important distances

Consider $\mathbf{x} = (x_1, \ldots, x_N)$ and $\mathbf{y} = (y_1, \ldots, y_N)$ of $\mathbf{R}^n$

- The usual euclidean distance:

$$d(x, y) = \sqrt{\sum_{i=1}^{N} |x_i - y_i|^2}$$

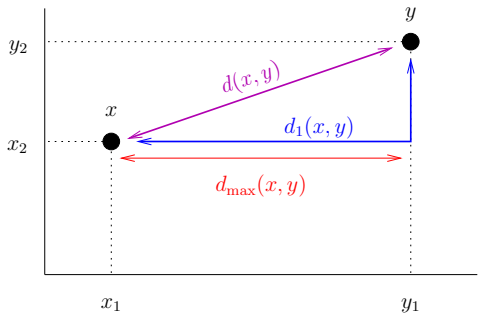
- The Manhattan distance (also called city block or taxicab distance):

$$d_1(x, y) = \sum_i |x_i - y_i|.$$

- The so-called maximum distance (also called Chebyshev distance):

$$d_{\max}(x, y) = \max_i |x_i - y_i|$$

Relation among the 3 distances



For all $x, y \in \mathbb{R}^N$ we have $d_1(x,y) \geq d(x,y) \geq d_{\max}(x,y)$

- For the taxi-cab and euclidean distances all differences $|x_i - y_i|$, $i = 1, \ldots, N$, have approximately the same relative weight in the computation of the overall distance
- For the maximum distance only the variable(s) i yielding the largest difference $|x_i - y_i|$ accounts for the overall distance

Canberra distance

If $\mathbf{x}, \mathbf{y}$ are N-dimensional vectors with positive components, one can define the so-called **Canberra distance**

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N} \frac{|x_i - y_i|}{x_i + y_i}$$

- This distance isa weighted version of the Manhattan distance that is sensitive to differences between values x_i and y_i of small amplitudes.
- It is invariant under differentiated changes of scale in each variable but not under variables centering. Only the relative proportion between the differences of the coordinates and their sum are importante.

When to standardize the data ?

- Usually, the euclidean distance between original numerical variables is employed if all variables are expressed in the same units and similar scales of measurement. Otherwise, it is usually better to standardize the data to give the same weight to all variables.

- It could also be interesting to explore if other types of dissimilarities (for instance, the Canberra or Mahalanobis distance), could be more appropriate...

Dissimilarity measures for binary data

Consider binary vectors $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$ and define

a: nr components where both variables take value 1 (positive agreement)

b: nr of components where $\mathbf{x}$ take value 1 and $\mathbf{y}$ value 0 (disagreement)

c: nr of components where $\mathbf{x}$ take value 0 and $\mathbf{y}$ value 1 (disagreement)

d: nr components where both variables take value 0 (negative agreement)

- **Simple matching** (counts double-zeroes, is suitable if 0-1 represent equally valued attributes like male-female):

$$S(\mathbf{x}, \mathbf{y}) = \frac{a+d}{a+b+c+d} \implies D(\mathbf{x}, \mathbf{x}) = 1 - S(\mathbf{x}, \mathbf{x}) = \frac{b+c}{a+b+c+d}$$

- **Jaccard coefficient** (does not count double zeroes. Suitable if 0-1 representunequal valued attributes, like species presences-absences):

$$J(\mathbf{x}, \mathbf{y}) = \frac{a}{a+b+c} \implies D(\mathbf{x}, \mathbf{y}) = 1 - J(\mathbf{x}, \mathbf{y}) = \frac{b+c}{a+b+c}$$

Example

Assume that we have two binary variables $\mathbf{x}$ and $\mathbf{y}$ representing presences (1) and absences (0) of two species at 16 spots:

$$\mathbf{x} = (0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0)$$

$$\mathbf{y} = (0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1)$$

We want to determine how similar are the two species with regard to their distribution in the 16 spots. Computing the positive and negative agreements/disagreements, we get $a = 1$, $b = 3$, $c = 3$ and $d = 9$ $(a + b + c + d = 16)$ and we have

- Simple matching: $\frac{a+d}{a+b+c+d} = 10/16$
- Jaccard coefficient: $\frac{a}{a+b+c} = 1/7$

The asymmetrical Jaccard's coefficient seems to the be a more suitable similarity to create homogeneous groups of species with respect to their distribution in the spots

R code

```
# The R function dist with the method ‘‘binary’’ computes the
dissimilarity as d(x,y) = 1 − S(x,y), where S is the Jaccard coefficient
d = dist(cbind(x,y),method=‘‘binary’’,diag=FALSE,upper=FALSE,p=2)
```

Several other dissimilarity measures well suited for binary data in the framework of ecology and community composition data are available via the function `dist.ldc` from the ADESPATIAL package

χ^2-distance for nominal data

- Let $\mathbf{X} = [x_{ij}]$ be a contingency table, where x_{ij} is the observed frequency in category a_i of a nominal variable A and category b_j of a nominal variable B (assuming nonzero row and column sums). Let I and J be the number of categories of A and B and $N = \sum_{i,j} x_{ij}$ the total number of observations.
- Dividing each row i by the corresponding row total, $x_{i\cdot} = \sum_j x_{ij}$, we obtain the so-called ith row-profile, $\left(\frac{x_{i1}}{x_{i\cdot}}, \ldots, \frac{x_{iJ}}{x_{i\cdot}}\right)$, which corresponds to the conditional distribution of variable B assuming category a_i of A.
- Theset of the I row-profiles defines a cloud of I points in $\mathbb{R}^J$ and the centroid of this cloud, $\frac{1}{I}\sum_i \left(\frac{x_{i1}}{x_{i\cdot}}, \ldots, \frac{x_{iJ}}{x_{i\cdot}}\right) \in \mathbb{R}^J$, is called is the mean row-profile.
- If variables A and B are independent, i.e., $x_{ij} = \frac{x_{i\cdot}x_{\cdot j}}{N}$ $\forall i, j$,

  $$\left(\frac{x_{i1}}{x_{i\cdot}}, \ldots, \frac{x_{iJ}}{x_{i\cdot}}\right) = \left(\frac{x_{\cdot 1}}{N}, \ldots, \frac{x_{\cdot J}}{N}\right) = (f_{\cdot 1}, \ldots, f_{\cdot J}),$$

  where $f_{\cdot j} = \sum_i f_{ij}$ are the column marginals of the relative frequencies $f_{ij} = \frac{x_{ij}}{N}$, and thus all row-profiles are equal to the mean row-profile. If A and B are not independent, the row-profiles spread away from the mean row-profile.
- The squared χ^2-distance between the ith and ℓth row-profiles is defined as,

  $$d^2_{\chi^2}(i, \ell) = \sum_{j=1}^{J} \frac{1}{f_{\cdot j}} \left(\frac{x_{ij}}{x_{i\cdot}} - \frac{x_{\ell j}}{x_{\ell\cdot}}\right)^2 = \sum_{j=1}^{J} \frac{1}{f_{\cdot j}} \left(\frac{f_{ij}}{f_{i\cdot}} - \frac{f_{\ell j}}{f_{\ell\cdot}}\right)^2$$

  (the weights in the inverse proportion of the column marginal frequencies $f_{\cdot j}$ increase the importance of the small differences between rare categories).

Example

Consider again the two-way contingency table slide 10, that contains the distribution by country of residence of the primary language spoken of 5000 inhabitants:

	English	French	Spanish	German	Italian	Total
Canada	688	280	10	11	11	1000
USA	730	31	190	8	41	1000
England	798	74	38	31	59	1000
Italy	17	13	11	15	944	1000
Switz.	15	222	20	648	95	1000
Total	2248	620	269	713	1150	5000

Source: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3718710/

χ^2-distance between the row-profiles

- The corresponding 5 row-profiles and mean row-profile are given below

	English	French	Spanish	German	Italian	Totals
Canada	0.688	0.280	0.010	0.011	0.011	1.000
USA	0.730	0.031	0.190	0.008	0.041	1.000
England	0.798	0.074	0.038	0.031	0.059	1.000
Italy	0.017	0.013	0.011	0.015	0.944	1.000
Switz.	0.015	0.222	0.020	0.648	0.095	1.000
mean	0.4496	0.124	0.0538	0.1426	0.230	1.000
f.j	0.4496	0.124	0.0538	0.1426	0.230	1.000

- The 5 row-profiles define a a cloud of $I = 5$ points in $\mathbb{R}^J$, with $J = 5$ (number of columns) with centroid given by the mean row-profile
- The squared χ^2-distance between the row profiles of *Canada* and *Switzerland* is

$$d^2_{\chi^2}(1,5) = \frac{(0.688-0.015)^2}{0.4496} + \frac{(0.280-0.222)^2}{0.124} + \frac{(0.010-0.020)^2}{0.0538} + \frac{(0.011-0.648)^2}{0.1426} + \frac{(0.011-0.095)^2}{0.230} = 3.912575$$

- We define similarly the set of 5 column-profiles, which can be regarded as a cloud of $J = 5$ points in $\mathbb{R}^I$, with $I = 5$ and the corresponding pairwise squared χ^2-distances (left as an exercise).
- The correspondence analysis (CA) allows to study and visualize the relationships of a contingency table when the number of categories is high.

Corresponding R code

The R function `dist.ldc` from the package ADESPATIAL computes the χ^2-distance matrix between every pair of row-profiles

R code

```
library(adespatial)

tab<-matrix(c( 688, 280, 10 , 11 , 11, 730, 31, 190, 8 , 41, 798, 74,
38, 31, 59, 17, 13, 11, 15, 944, 15, 222, 20, 648, 95),
nrow=5, byrow = TRUE)

colnames(tab)<-c("English", "French", "Spanish", "German", "Italian")

rownames(tab)<-c("Canada","USA","England","Italy","Switz.")

tab

d.chisqr<-dist.ldc(tab,method="chisquare")

d.chisqr
```

We obtain the following distance matrix (d_{χ^2}) between row-profiles

Countries	Canada	USA	England	Italy
USA	1.0536310			
England	0.6297091	0.6780536		
Italy	2.3154271	2.2966246	2.1925680	
Switzerland	1.9780231	2.2030640	2.0546442	2.5094977

For instance, $d^2_{\chi^2}(r_1, r_5) = (1.9780231)^2 = 3.912575$, as in the previous slide

Dissimilarity measures for variables

- An usual similarity notion between two variables x and y is Pearson's correlation coefficient

$$r = \frac{s^2_{xy}}{s_x\, s_y}$$

  This similarity can be transformed into a dissimilarity using the transformation $d = \sqrt{1 - r^2}$, which take values in the interval $[0, 1]$
- Highly linearly correlated variables (positively or negatively) will have $d \approx 0$ while for uncorrelated variables $d \approx 1$
- Alternatively, we can define $d = (1 - r)/2$. In this case the strength of the linear relationship and the direction are both accounted
- We can use the above dissimilarity measures to cluster variables. Each cluster will consist of a set of variables highly correlated. This can be useful to detect redundancies and can give an idea of the number of principal dimensions of data

Clustering methods

- Distance-based models rely only on pairwise dissimilarities between individuals
- Density-based clustering seeks for high density regions of points (clusters) separated by low density of points (noise)
- Model-based clustering assumes that the data in each cluster is drawn from some probabilistic distribution (the standard model is a finite mixture of multivariate gaussians) and assign a degree of membership (probability) to each element to belong to a cluster. Can be considered as generalizations of some distance-based clustering methods
- Constrained-clustering methods, are clustering methods that also account for other type of information, like spatial relationships between observations (for instance, contiguity relationships between cells in a map)
- ...

Two important types of clustering

- Hierarchical clustering - produces a *nested* structure of partitions and do not requires that the number of clusters is known *a priori*:
  - *Hierarchical agglomerative (or ascending) clustering algorithm* (HAC) - starts from the partition consisting of N clusters with one individual per cluster (*singletons*) and proceeds until a unique group is obtained.
  - *Divisive clustering algorithm* - proceeds in the opposite way and are usually more computacional demanding, being more seldom used (not considered in this course)
- Partitional clustering - produces *flat* (non-nested) partition and requires that the number of clusters is known *a priori*. Usually seeks to maximize some criterion like the intra-cluster homogeneity or the inter-cluster heterogeneity.

Hierarchical ascending clustering algorithm

Algorithm

Input: *the proximity matrix containing the pairwise dissimilarities between N individuals $x_1, \ldots, x_N$.*

- *Starts with N clusters containing a single object each (singletons);*
- *Merges the least dissimilar pair of clusters into a new cluster, according to the given definition of distance between clusters, and updates the proximity matrix (reducing its order by one);*
- *Repeats step 2 $N - 1$ steps, until only the cluster containing all individuals is obtained.*

Output: *the sequence (of length $N - 1$) of the clusters aggregated during the clustering algorithm along with pairwise distances between the merged clusters.*

Remark

Once two individuals are grouped together they cannot be split apart at a posterior stage.

Dissimilarity between clusters

The dissimilarity $d_{i,j} = D(\mathcal{C}_i, \mathcal{C}_j)$, between clusters C_i and C_j with n_i and n_j elements, respectively, depends on the aggregation method:

- Single-linkage or nearest-neighbor:

$$d_{i,j} = \min_{x \in \mathcal{C}_i, y \in \mathcal{C}_j} d(x, y)$$

- Complete-linkage or furthest-neighbor:

$$d_{i,j} = \max_{x \in \mathcal{C}_i, y \in \mathcal{C}_j} d(x, y)$$

- Average:

$$d_{i,j} = \frac{1}{n_i n_j} \sum_{x \in C_i} \sum_{y \in C_j} d(x, y)$$

- Centroid
- Median
- Ward or minimum-variance clustering
- ...

Recurrence formula for HAC

- For all aggregation methods that we are going to consider, the dissimilarity between two merged clusters, say $\mathcal{C}_i \cup \mathcal{C}_j$, and each one of the remaining clusters $\mathcal{C}_k$,

  $$d_{ij,k} = D(\mathcal{C}_i \cup \mathcal{C}_j, \mathcal{C}_k),$$

  can be determined in terms only of the pairwise dissimilarities,

  $$d_{i,j} = D(\mathcal{C}_i, \mathcal{C}_j), \quad d_{i,k} = D(\mathcal{C}_i, \mathcal{C}_k), \quad d_{j,k} = D(\mathcal{C}_j, \mathcal{C}_k)$$

- In other words, the proximity matrix containing the pairwise distances between the clusters at a given step $\ell + 1$ can be determined in terms of the proximity matrix containing the pairwise distances between the clusters at the previous step ℓ, via a convenient recurrence formula

- Therefore and unlike many other statistical methods like PCA, the HAC algorithm does not require the knowledge of the original data matrix $\mathbf{X}$, but only the knowledge of the proximity matrix containing the pairwise distances between the elements of $\mathbf{X}$.

Example of updating formulas

- Single-linkage or nearest-neighbor:

  $$d_{ij,k} = \min\{d_{i,k}, d_{j,k}\}$$

- Complete-linkage or furthest-neighbor:

  $$d_{ij,k} = \max\{d_{i,k}, d_{j,k}\}$$

- Average:

  $$d_{ij,k} = \frac{n_i d_{i,k} + n_j d_{j,k}}{n_i + nj}$$

The recurrence formula for other linkages schemes, like the examples below, will be obtained via Lance-Williams Chart (see slide 57).

- Centroid
- Ward

Dendrogram

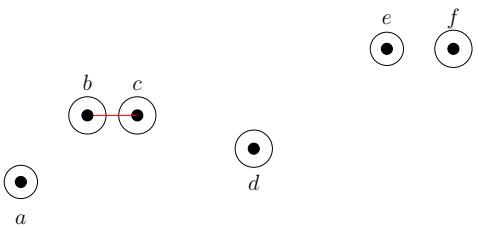
The sequence of length $N-1$ of the merged clusters and the corresponding fusion costs (i.e., the distance between the merged clusters) can be graphically represented by a special tree graph called dendrogram

- Dendrograms are tree-like diagrams made of branches that join terminal nodes called leaves.
- The branches represent clusters and the heights at which the branches are connected represent fusion costs.
- The leaves represent the objects to be clustered.
- The lifetime of a branch is the difference of fusion costs between the step in which it appears and the step in which it is aggregated.

Example: step -1 (initial step)

As an example we are going to performing the single-linkage clustering algorithm to a set of 6 points, only knowing its proximity matrix

$X = \{a, b, c, d, e, f\}$



PROXIMITY MATRIX

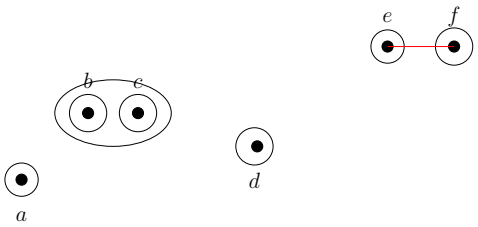
	a	b	c	d	e
b	0.7				
c	1.0	0.3			
d	1.8	.1.3	0.9		
e	2.9	2.4	1.9	1.3	
f	3.4	2.8	2.4	1.7	.5

At the initial step all clusters are singletons

Next step merges the clusters $\{b\}$ and $\{c\}$ with fusion cost 0.3 (the least dissimilar pair) and in each dashed box the minimum value is chosen, reducing the proximity matrix order by one, and defining the dissimilarities between each one of the singletons and the new formed cluster $\{b, c\}$

Step -2

$X = \{a, b, c, d, e, f\}$

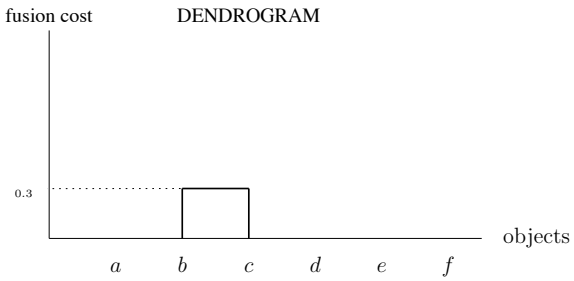


PROXIMITY MATRIX

	a	$\{b,c\}$	d	e
$\{b,c\}$	0.7			
d	1.8	0.9		
e	2.9	1.9	1.3	
f	3.4	2.4	1.7	0.5

Next step merges the singletons $\{e\}$ and $\{f\}$

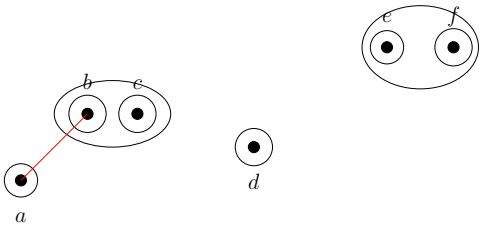
with fusion cost 0.5



Step - 3

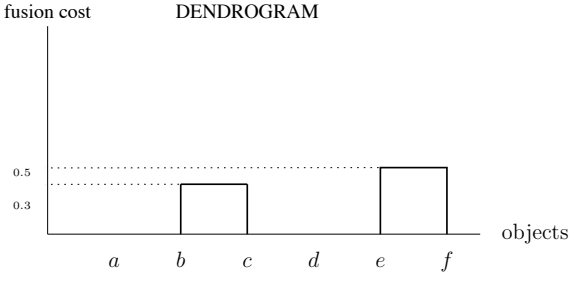
PROXIMITY MATRIX

	a	$\{b,c\}$	d
$\{b,c\}$	0.7		
d	1.8	0.9	
$\{e,f\}$	2.9	1.9	1.3

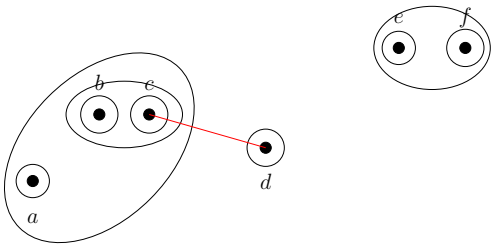


Next step merges the pair of clusters $\{a\}$ and $\{b,c\}$

with fusion cost 0.7



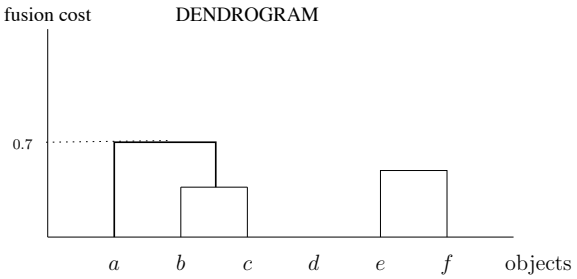
Step - 4



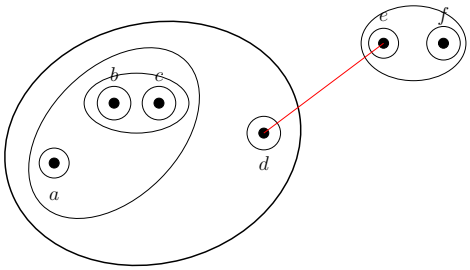
PROXIMITY MATRIX

	$\{a,b,c\}$	d
d	0.9	
$\{e,f\}$	1.9	1.3

Next step merges the clusters $\{a, b, c\}$ and $\{d\}$ with fusion cost 0.91



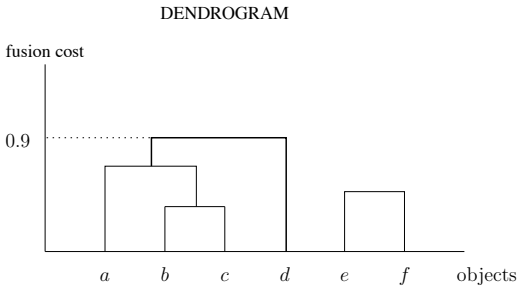
Step - 5



PROXIMITY MATRIX

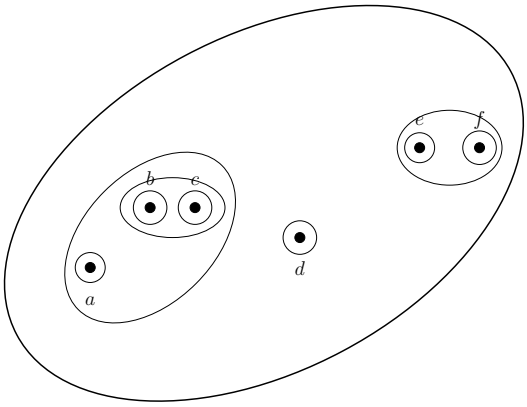
	$\{a,b,c,d\}$
$\{e,f\}$	1.3

Next step is the final one and merges the clusters $\{a, b, c, d\}$ and $\{e, f\}$ with fusion cost 1.3



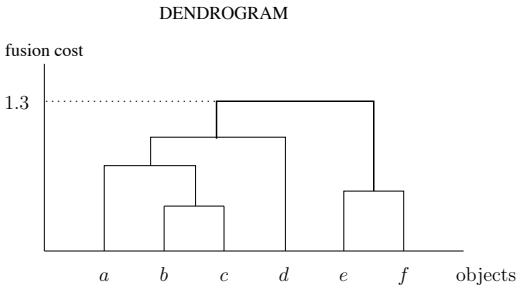
step - 6 (final step)

The final structure of nested clusters and the dendrogram encoding the clustering procedure are the following



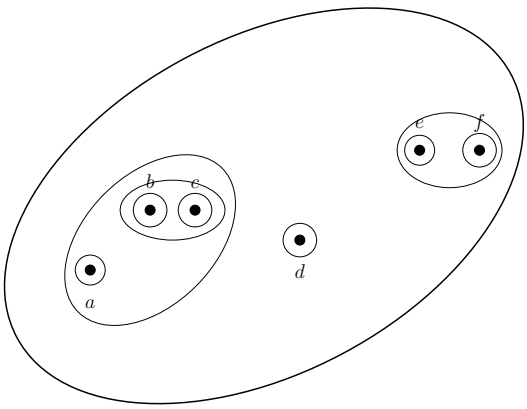
PROXIMITY MATRIX

EMPTY



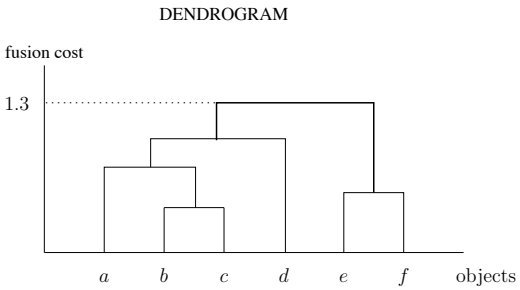
step - 6 (final step)

The final structure of nested clusters and the dendrogram encoding the clustering procedure are the following



PROXIMITY MATRIX

EMPTY



The R function hclust

It performs hierarchical agglomerative clustering using several aggregation criterion methods and it admits an arbitrary dissimilarity matrix as input

input: a *dissimilarity matrix* d and the clustering *method* among the options, "ward", "single", "complete" (default), "average", "mcquitty", "median" or "centroid".

value: the function returns an object of the class *hclust*, which consists of a list including, among others, the following elements:
merge - a $(n-1) \times 2$ matrix indicating the clusters being merged
heigth - the list of fusion costs

R code

```
hc<-hclust(d, method=''complete'', members=NULL)

plot(hc)

# to plot all leaves at the same height do instead
plot(hc, hang=-1)
```


R code for a single-linkage example with output

```
X<-matrix(c(0,0,0.5,0.5,0.85,0.5,1.75,0.25,2.75,1,3.25,1),nrow=6,byrow=TRUE)
# the set of 6 points {a,b,c,d,e,f} in two variables
     [,1]   [,2]
[1,] 0.00 0.00 point "a"
[2,] 0.50 0.50 point "b"
[3,] 0.85 0.50 point "c"
[4,] 1.75 0.25 point "d"
[5,] 2.75 1.00 point "e"
[6,] 3.25 1.00 point "f"
d<-dist(X) # by default uses the euclidean distance
SL<-hclust(d, method="single")
SL$height
[1] 0.375 0.5 0.707 0.91 1.25
SL$merge
[,1] [,2]
[1,] -2 -3 (merges singletons {b} and {c})
[2,] -5 -6 (merges singletons {e} and {f})
[3,] -1 1 (merges singleton {a} with cluster {b,c})
[4,] -4 3 (merges singleton {d} with cluster {a,b,c})
[5,] 2 4 (merges cluster {e,f} with cluster {a,b,c,d})
(The number with minus sign refers to a singleton ID, otherwise refers
to the step number where the cluster was aggregated)
```

Where to cut the dendrogram?

A cut in a dendrogram at a given height τ produces the (flat) partition into the clusters whose fusion cost is smaller than τ

Usually one seeks cuts in the dendrogram such that:

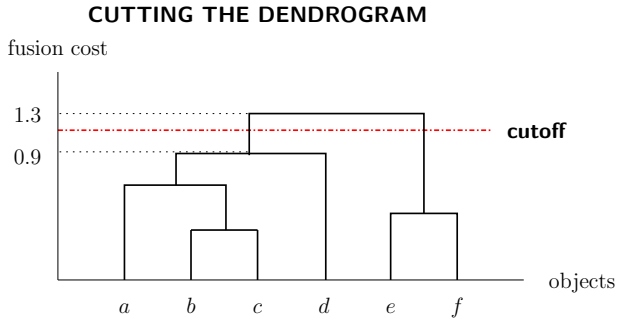
- split high height branches (high lifetimes) to get high inter-cluster heterogeneity
- as close as possible to the leaves to get high intra-class homogeneity

Some caution has to be applied regarding the decision where to cut the dendrogram (and what is the "best" number of clusters). With some methods (for instance, the Ward method), the dendrogram lifetimes tend to increase when the larger clusters are merged, due to the way the fusion costs are defined

Several internal validity indices can be used to estimate the optimal number of clusters

Example

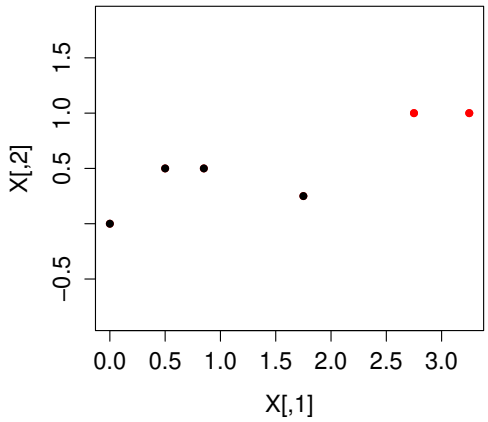
For instance, to obtain a partition into 2 clusters we have to cut the dendrogram at some height in the interval $]0.9, 1.3[$, yielding the clusters $\mathcal{C} = \{a, b, c, d\}$ and $\mathcal{C}' = \{e, f\}$



- The cluster $\{e, f\}$ is relatively well separate from the cluster $\{a, b, c, d\}$ since the fusion cost (1.3) between these groups is relatively high
- But cluster $\{a, b, c, d\}$ is not very homogeneous since the fusion cost (0.9) of aggregating all of its elements is also relatively high

Cutting the dendrogram in R

The resulting partition into two clusters $\{a, b, c, d\}$ and $\{e, f\}$ (depicted using distinct colors)



R code

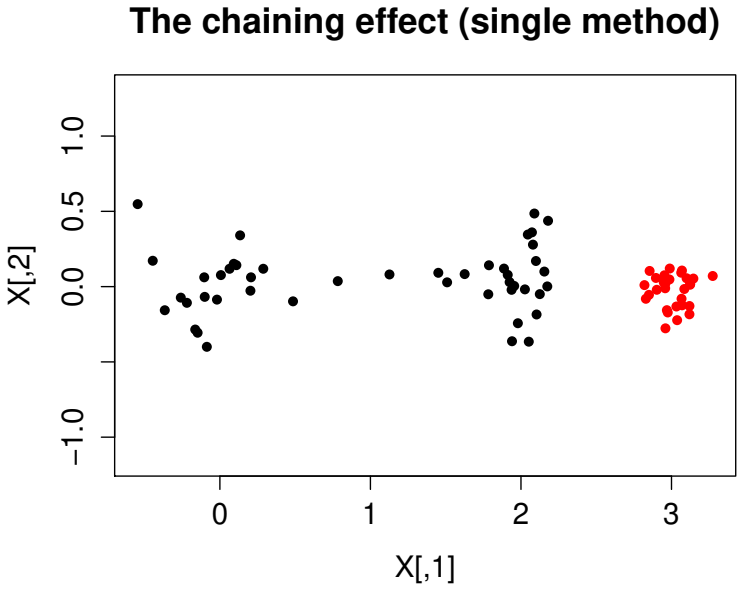
```
SL<-hclust(X,method="single")
part<-cutree(SL,2) # 2 clusters
# or do instead
part<-cutree(SL,h=1.1) # h is the height where to cut
plot(X,type="p",cex=0.8,pch=16, col=part,asp=TRUE)
```


Chaining effect

- In single-linkage if two clusters are merged at a fusion cost τ, every pair of objects, one in each cluster, have pairwise distance greater than or equal to τ.
- As the clusters growth it becomes more and more easier to incorporate new elements in the cluster since the distances between these elements and the cluster is the distance to the nearest point in the cluster
- As a consequence, the singletons tend to aggregate to the larger clusters, often producing elongated clusters (chaining effect) and/or very unbalanced partitions

Chaining effect

The chaining effect is usually produced by the existence of intermediate points between clusters, giving rise to elongated clusters connecting distant points



Single-linkage emphasizes clusters separation

The nearest neighbor distance can be used to measure of separability between clusters. More precisely, we can measure the separability of a partition $X = C_1 \cup \cdots \cup C_k$ as the distance between the closest pair of clusters for the nearest neighbor criterion, i.e., as

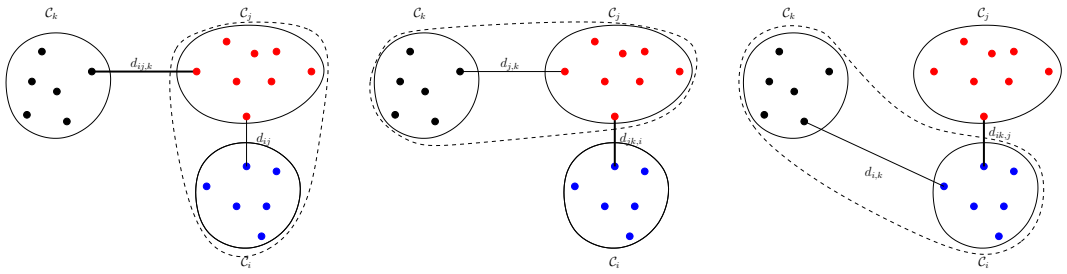
$$\min_{i \neq j} D(C_i, C_j) = \min_{i \neq j} \left(\min_{x \in C_i,\, y \in C_j} d(x, y) \right).$$

In each step the single-linkage algorithm merges the pair of closest clusters, which amounts to say that it merges the pair of clusters that maximizes the separability of the resulting partition

Remark

The single-linkage clustering algorithm tends to produce well separate partitions but not necessarily homogeneous!

Single-linkage and clusters separation



Remark

The aggregation of the closest clusters in the leftmost case yield the better separated 2-partition among the 3 possible 2-partitions:

$$\{C_{ij}, C_k\}, \qquad \{C_{jk}, C_i\}, \qquad \{C_{ik}, C_j\}$$

Single-linkage clustering - summary

Pros

- Can detect arbitrary cluster shapes
- Can be applied to large datasets since it is computationally efficient, i.e., there are polynomial-time clustering algorithms
- Emphasizes clusters separation, i.e., tends to form well separated clusters
- Invariant under monotonic transformations of the proximity matrix, since it only depends on the rank orders of the pairwise distances between the points of the dataset
- Insensitive to ties in the proximity matrix

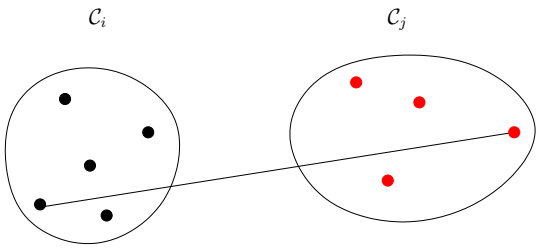
Cons

- Suffers from the chaining effect - often produces elongated clusters with very distinct sizes
- Sensitive to observation errors and noise
- The decision of aggregate two clusters relies only on a pair of elements, one in each cluster

Complete-linkage

The complete-linkage or furthest neighbor is the opposite of nearest-neighbor clustering algorithm The fusion cost between two clusters $\mathcal{C}_i$ and $\mathcal{C}_j$ in this method is defined as the distance between the furthest pair of points, one in each cluster, that is,

$$d_{i,j} = D(\mathcal{C}_i, \mathcal{C}_j) = \max_{x \in \mathcal{C}_i, y \in \mathcal{C}_j} d(x, y)$$



Updating formula for the complete-linkage:

$$\boxed{d_{ij,k} = \max\{d_{i,k}, d_{j,k}\}}$$

Complete-linkage method

- In complete-linkage two clusters are merged at a height τ only if all elements of one cluster are at a distance inferior than or equal to τ with respect to the elements of the other cluster.
- As the cluster growths it becomes more and more difficult to incorporate new elements in a cluster. Therefore the aggregations tend to occur between clusters with few elements.
- The complete method tend to be sensitive to the presence of outliers.

Exercise

- Perform a clustering analysis with the complete-linkage method on the set of points of the real line $X = \{0.2, 3, 4.2, 5, 5.9\}$ and represent the respective dendrogram.
- Cut the dendrogram in order to obtain two clusters. What you conclude?

Complete-linkage emphasizes clusters homogeneity

The diameter of a set C is the largest dissimilarity between pairs of elements of C, i.e.,

$$\text{diam}(C) = \max_{x,y \in C} d(x, y)$$

We can measure the cohesion of a partition $X = C_1 \cup \ldots \cup C_k$, as the partition diameter, i.e., as the largest value among the diameters of $C_1, \ldots, C_k$:

$$\max_i \text{diam}(C_i) = \max_i \left(\max_{x,y \in C_i} d(x, y) \right).$$

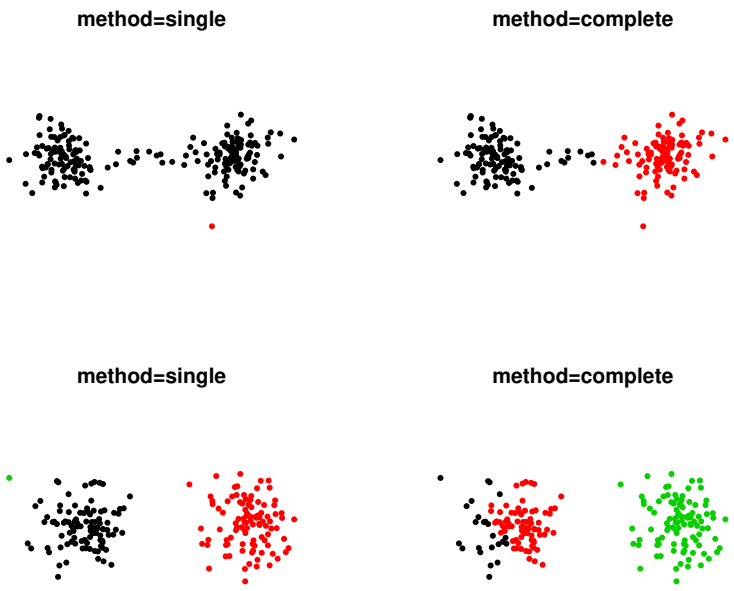
In each step the complete-linkage, also called diameter clustering method, seeks to aggregate the clusters that produce the smallest increase in the partition diameter, i.e., such that the resulting partition has the smallest possible diameter.

Remark

The complete-linkage clustering algorithm tends to produce compact clusters, but not necessarily well separated!.

Noise and outliers: single vs complete linkage methods

The following examples illustrates that the single clustering method is more sensitive to noise than complete, whereas the opposite occurs with outliers (the partitions on the top row have two clusters each and partitions on bottom row 3 clusters)



Complete-linkage clustering - summary

Pros

- Emphasizes cluster compactness - tend to form tight spherical clusters with small diameters, i.e., homogenous clusters
- Invariant under monotonic transformations of the proximity matrix - only the ranks of the pairwise dissimilarities are important.

Cons

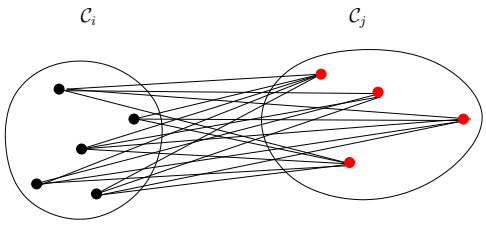
- Sensitive to outliers
- Cannot detect arbitrary cluster shapes
- The decision to aggregate two cluster relies only on a pair of individuals, one in each cluster

Average clustering method

In-between the single-linkage and the complete-linkage clustering methods, we have the average method, also known as *unweighted pair group method average* (UPGMA) The merging cost between two clusters $\mathcal{C}_i$ and $\mathcal{C}_j$ is defined as thearithmetic mean of the distances between every point of $\mathcal{C}_i$ and eevery point of $\mathcal{C}_j$, i.e., equals

$$d_{i,j} = \frac{\sum_{x \in \mathcal{C}_i} \sum_{y \in \mathcal{C}_j} d(x,y)}{n_i \, n_j},$$

where $n_i = |C_i|$ and $n_j = |C_j|$.



The recurrence formula is given by (left as an exercise),

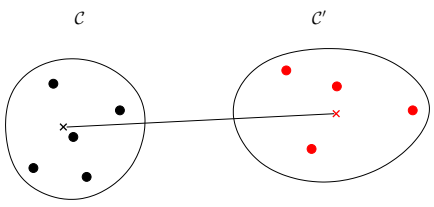
$$\boxed{d_{ij,k} = \frac{n_i d_{i,k} + n_j d_{j,k}}{n_i + n_j}}$$

This method often outperforms single-linkage and complete linkage but it is not invariant under monotonic transformations of the proximity matrix

Centroid clustering model

This method, also known as UPGMC (unweighted pair group method centroid) implements the very natural idea that the clusters are represented by their centroids and thus define distance $d_{i,j}$ between two clusters $\mathcal{C}_i$ and $\mathcal{C}_j$ as the distance between the respective centroids m_i and m_j:

$$d_{i,j} = \left\| \frac{1}{|\mathcal{C}_i|} \sum_{x_i \in \mathcal{C}_i} x_i - \frac{1}{|\mathcal{C}_j|} \sum_{x_j \in \mathcal{C}_j} x_j \right\| = \|m_i - m_j\|$$



The centroid of the group obtained by merging the clusters C_i and C_j is given by

$$m_{ij} = \frac{n_i m_i + n_j m_j}{n_i + n_j}$$

The updating formula is more complicated in this case. We shall resort to a general procedure to define the updating formula for the centroid method.

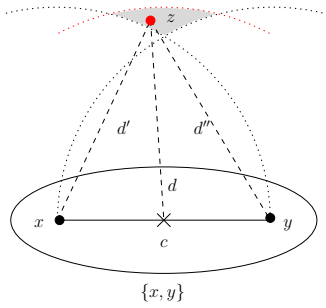
Exercise

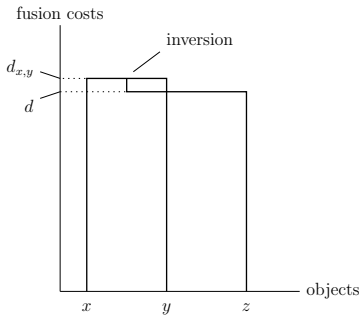
- Perform a clustering analysis using the centroid method on the set of 3 points of $\mathbb{R}^3$, $X = \{(0,0),(8,0),(4,7.5)\}$ and represent the respective dendrogram
- What happened ?

Centroid clustering model - inversions

In the centroid method the merging cost can be non-monotonic, giving rise crossovers (also called inversions) in the dendrogram

All circles have radii equal to the distance between x and y, $d_{x,y}$.





Since z (red point) lie in the grey area, ouside the black circles, $d_{x,y} < d', d''$. Hence x and y are the first pair of objects to be merged. Since z lie inside the red circle centred at the centroid c of x and y,

$$D(\{x,y\},z) = d_{c,z} < d_{x,y} = D(\{x\},\{y\})$$

Lance-Williams recurrence formula

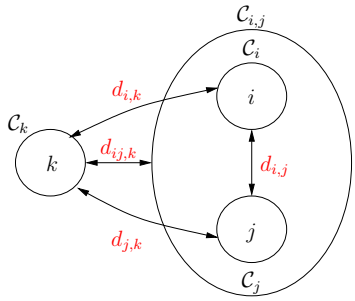
Given clusters $\mathcal{C}_i$, $\mathcal{C}_j$, $\mathcal{C}_k$ and $\mathcal{C}_{ij} = \mathcal{C}_i \cup \mathcal{C}_j$ we will define updating formulas for a family of clustering methods

$$\boxed{d_{ij,k} = \alpha_i d_{i,k} + \alpha_j d_{j,k} + \beta d_{i,j} + \gamma |d_{i,k} - d_{j,k}|}$$

or

$$\boxed{d^2_{ij,k} = \alpha_i d^2_{i,k} + \alpha_j d^2_{j,k} + \beta d^2_{i,j} + \gamma |d^2_{i,k} - d^2_{j,k}|}$$

depending on the method considered, where α_i, α_j, β and γ are convenient parameters that may depend only on the clusters cardinality $n_i = |C_i|$, $n_j = |C_j|$, $n_k = |C_k|$ and $n_i + n_j = |C_{ij}|$:



Lance-Williams Chart

	α_i	α_j	β	γ	dissimilarity matrix	reversals
single	$\frac{1}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$	d_{ij}	NO
complete	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	d_{ij}	NO
average (UPGMA)	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	0	0	d_{ij}	NO
McQuitty (WPGMA)	$\frac{1}{2}$	$\frac{1}{2}$	0	0	d_{ij}	NO
centroid (UPGMC)	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	$\frac{-n_i n_j}{(n_i+n_j)^2}$	0	d^2_{ij}	can occur
median (WPGMC)	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{4}$	0	d_{ij}	can occur
Ward	$\frac{n_i+n_k}{n_i+n_j+n_k}$	$\frac{n_j+n_k}{n_i+n_j+n_k}$	$-\frac{n_k}{n_i+n_j+n_k}$	0	d^2_{ij}	NO

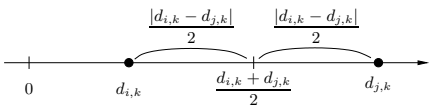
Derivation of Lance-Williams parameters in 2 examples

Let us see how to obtain the recurrence parameters of Lance-Williams Chart for the single-linkage and complete-linkage methods. We know that

$$\begin{aligned} d_{ij,k} &= \min\{d_{i,k}, d_{j,k}\} \quad \text{(single-linkage)}, \\ d_{ij,k} &= \max\{d_{i,k}, d_{j,k}\} \quad \text{(complete-linkage)}. \end{aligned}$$

Assuming, without loss of generality, $d_{i,k} \leq d_{j,k}$ we have

$$\begin{aligned} d_{ij,k} &= \min(d_{i,k}, d_{j,k}) = d_{i,k} = \frac{d_{i,k} + d_{j,k}}{2} - \frac{1}{2}|d_{i,k} - d_{j,k}|, \\ d_{ij,k} &= \max(d_{i,k}, d_{j,k}) = d_{j,k} = \frac{d_{i,k} + d_{j,k}}{2} + \frac{1}{2}|d_{i,k} - d_{j,k}|. \end{aligned}$$



Hence Lance-Williams parameters for single-linkage and complete-linkage are, resp.,

$$\alpha_i = \alpha_j = \frac{1}{2},\ \gamma = -\frac{1}{2} \text{ and } \beta = 0 \text{ (single-linkage)},$$
$$\alpha_i = \alpha_j = \frac{1}{2},\ \gamma = \frac{1}{2} \text{ and } \beta = 0 \text{ (complete-linkage)}.$$

Example: recurrence formula for the centroid method

Using the previous Lance-Williams table we obtain the following updating formula for the centroid method:

$$\boxed{d^2_{ij,k} = \frac{n_i}{n_i+n_j} d^2_{i,k} + \frac{n_j}{n_j+n_j} d^2_{j,k} - \frac{n_i n_j}{(n_i+n_j)^2} d^2_{i,j}}$$

Note that the distances are squared!

Repeat the clustering performed on the set $\mathbf{X}$ of slide 54 and using the update formula given here

Monotonic condition and inversions

We say that a clustering method satisfies the monotonic condition if whenever two clusters $\mathcal{C}_i$ and $\mathcal{C}_j$ are merged into a cluster $\mathcal{C}_{ij}$ we have

$$d_{ij,k} \geq d_{i,j} \qquad \forall k \neq i, j, ij$$

This implies that the dendrogram cannot have inversions

Proposition

If in the Lance-Williams's formula the parameters α_i, α_j are nonnegative, $\alpha_i + \alpha_j + \beta \geq 1$, and either $\gamma \geq 0$ or $\max\{-\alpha_i, -\alpha_j\} \leq \gamma \leq 0$, the clustering method satisfies the monotonic condition(*)

(*) A stronger condition is given by Batagelj : the Lance-Williams clustering algorithm is monotonic if and only if,

$$\gamma \geq -\min(\alpha_1, \alpha_2), \quad \alpha_1 + \alpha_2 \geq 0, \quad \alpha_1 + \alpha_2 + \beta \geq 1$$

From the Lance-Williams table we deduce immediately that *single*, *complete*, *average*, *McQuitty* and *Ward* verify the conditions of the proposition above and therefore satisfy the monotonic condition.
In particular, their dendrograms cannot have inversions.

Ward's method

Let $\mathbf{X}$ be a dataset containing the observations of p variables across N individuals $\mathbf{x}^1, \ldots, \mathbf{x}^N$. Let $\mathbf{x}^G = (\bar{x}_1, \ldots, \bar{x}_p)$ be the mean vector of $\mathbf{X}$. Given a partition of set of individuals into K clusters

$$\{\mathbf{x}^1, \ldots, \mathbf{x}^N\} = \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_K$$

we define,

- $\mathbf{SSQ}_t = \sum_{i=1}^{N} \|\mathbf{x}^i - \mathbf{x}^G\|^2 = \sum_{i=1}^{N}\sum_{j=1}^{p}(x_{ij} - \bar{x}_j)^2$ (total inertia)

  Note that $\mathbf{SSQ}_t = \text{tr}((\mathbf{X}^*)^T\mathbf{X}^*) = (N-1)\text{tr}(\mathbf{S})$

- $\mathbf{SSQ}_b = \sum_{k=1}^{K} n_k \|\mathbf{m}_k - \mathbf{x}^G\|^2$ (between-clusters inertia)

- $\mathbf{SSQ}_w = \sum_{k=1}^{K}\sum_{\mathbf{x}\in\mathcal{C}_k} \|\mathbf{x} - \mathbf{m}_k\|^2$ (total within-clusters inertia),

where $\mathbf{m}_k$ is the centroid of cluster $\mathcal{C}_k$ and n_k the number of its elements

Ward's method

- The between-clusters inertia $\mathbf{SSQ}_b$ represents the inertia of the dataset assuming that each cluster $\mathcal{C}_k$ is represented by n_k copies of the cluster centroid $\mathbf{m}_k$ (is equal to $(N-1)\text{tr}(\mathbf{S}_b)\dots$).
- The total within-clusters inertia $\mathbf{SSQ}_w$ represents the information that is lost by replacing the n_k elements of each cluster $\mathcal{C}_k$ by n_k copies of the cluster centroid (is equal to $(N-1)\text{tr}(\mathbf{S}_w)\dots$).
- By Huygens theorem, $\mathbf{SSQ}_t = \mathbf{SSQ}_b + \mathbf{SSQ}_w$, which is a constant.
- Ward's clustering method, also called minumum variance criterion, tries to minimize the total within-clusters inertia $\mathbf{SSQ}_w$, i.e., the clusters heterogeneity/variability, which, by Huygens theorem, amounts to maximize the between-clusters inertia $\mathbf{SSQ}_b$, i.e., the clusters separation
- Hence Ward's method seeks to simultaneously optimize two criteria: maximize the clusters separation and minimize the clusters variability

Increase in the sum of within-cluster inertia

- At beginning all clusters have a unique element and therefore,

$$\mathbf{SSQ}_t = \mathbf{SSQ}_b, \qquad \mathbf{SSQ}_w = 0$$

- At each step, Ward's method merges the pair of clusters $\mathcal{C}_i$, $\mathcal{C}_j$ yielding the smallest increase in the total within-cluster inertia $\mathbf{SSQ}_w$
- We shall write $\mathbf{SSQ}_w$ as

$$\mathbf{SSQ}_w = \sum_{k=1}^{K} \mathbf{e}_k^2,$$

where $\mathbf{e}_k^2$ is the inertia of cluster k in, i.e.,

$$\mathbf{e}_k^2 = \sum_{\mathbf{x}\in\mathcal{C}_k} \|\mathbf{x}-\mathbf{m}_k\|^2 = \frac{\sum_{\mathbf{x},\mathbf{y}\in\mathcal{C}_k} \|\mathbf{x}-\mathbf{y}\|^2}{2n_k}$$

(note that the later expression only depends on the pairwise distances between elements of C_k).

Increase in the sum of within-cluster inertia

- When two clusters $\mathcal{C}_i$ and $\mathcal{C}_j$ are merged into a cluster $\mathcal{C}_{ij}$, the increase in the total within-cluster inertia $\mathbf{SSQ}_w$ reduces to the following statistic,

$$\Delta_{ij}\mathbf{SSQ}_w = \mathbf{e}_{ij}^2 - \mathbf{e}_i^2 - \mathbf{e}_j^2,$$

since all other within-group inertias are not affected. After $N-1$ aggregation steps (assuming $|X| = N$) the sum of the successive increases $\Delta_{ij,k}$ is equal to the total inertia $\mathbf{SSQ}_t$.

- It can be proved that

$$\Delta_{ij}\mathbf{SSQ}_w = \frac{n_i n_j}{n_i + n_j}\|\mathbf{m}_i - \mathbf{m}_j\|^2,$$

which represents a weighted distance between the cluster centroids (cf. with centroid method).

- In particular, $\Delta_{ij}\mathbf{SSQ}_w$ is always nonnegative (i.e., the $\mathbf{SSQ}_w$ is increasing) and only depends on the squared distance between the cluster centroids $\mathbf{m}_i$ and $\mathbf{m}_j$ and on the cluster sizes n_i and n_j.

A better recurrence formula for Ward's method with LW

- The fusion cost between the clusters $\mathcal{C}_{ij} = \mathcal{C}_i \cup \mathcal{C}_j$ and $\mathcal{C}_k$ is

$$\Delta_{ij,k}\mathbf{SSQ}_w = \frac{(n_i + n_j)n_k}{n_i + n_j + n_k}\|m_{ij} - m_k\|^2,$$

which can be used as an updating formula for Ward's clustering method but has the disadvantage that it requires the knowledge of the original dataset to compute the centroids.

- Using the Lance-Williams table we can derive an alternative updating formula for Ward's method that only requires the (squared) proximity matrix at previou step:

$$\boxed{d_{ij,k}^2 = \frac{(n_i + n_k)d_{i,k}^2 + (n_j + n_k)d_{j,k}^2 - n_k d_{i,j}^2}{n_i + n_j + n_k}}$$

- The above expression actually returns twice the value of $\Delta_{ij,k}\mathbf{SSQ}_w$ and corresponds to the square of the dendrogram height computed with R function `hclust` and the ward.D2 method.

Example

Consider the univariate dataset $X = \{a, b, c, d\} = \{1, 2, 4, 8\}$
The pairwise distances and squared pairwise distances between elements of X are given, respectively, by

$$\left[\begin{array}{c|ccc} D & a & b & c \\ \hline b & 1 & & \\ c & 3 & 2 & \\ d & 7 & 6 & 4 \end{array}\right] \quad \text{and} \quad \left[\begin{array}{c|ccc} D^2 & a & b & c \\ \hline b & 1 & & \\ c & 9 & 4 & \\ d & 49 & 36 & 16 \end{array}\right]$$

The minimum of the squared distances is attained for $D^2(a, b)$ so the first pair to be clustered will be $a \cup b$ with squared fusion cost equal to 1

Example (cont.)

$$\begin{aligned} D^2(a \cup b, c) &= \frac{2\,D^2(a, c) + 2\,D^2(b, c) - D^2(a, b)}{3} \\ &= \frac{2 \cdot 9 + 2 \cdot 4 - 1}{3} = \frac{25}{3} \end{aligned}$$

and

$$\begin{aligned} D^2(a \cup b, d) &= \frac{2\,D^2(a, d) + 2\,D^2(b, d) - D^2(a, b)}{3} \\ &= \frac{2 \cdot 49 + 2 \cdot 36 - 1}{3} = \frac{169}{3} \end{aligned}$$

$D^2(c, d)$ is not affected. Thus the new squared dissimilarity matrix is

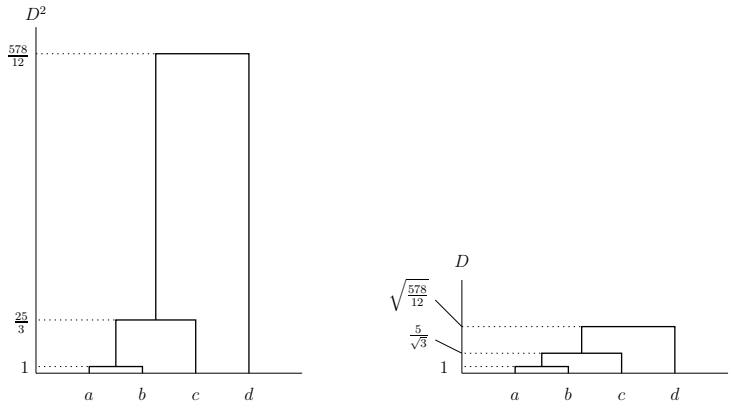
$$\left[\begin{array}{c|cc} D^2 & a \cup b & c \\ \hline c & \frac{25}{3} & \\ d & \frac{169}{3} & 16 \end{array}\right]$$

The minimum of the squared distances is attained for $D^2(a \cup b, c)$ so the next pair to be clustered will be $(a \cup b) \cup c$ with squared fusion cost $\frac{25}{3}$.

Ward clustering using LW recurrence formula (concl.)

$$D^2((a \cup b) \cup c, d) = \frac{3\,D^2(a \cup b, d) + 2\,D^2(c, d) - D^2(a \cup b, c)}{4}$$
$$= \frac{3 \cdot \frac{169}{3} + 2 \cdot 16 - \frac{25}{3}}{4} = \frac{578}{12}.$$

The dendrogram can be presented either using squared or not squared fusion costs. Its topology however does not change



Ward's aggregation method

The dendrogram of the previous slide with Ward's aggregation method can also be computed using the R software as follows:

R code

```
X<-c(1,2,4,8)
N<-length(X)
d<-dist(X) # (euclidean) distance matrix
h.ward<-hclust(d,method="ward.D2")
h.ward$height
sum(h.ward$height**2)/2
SSQt=var(X)*(N-1)
plot(h.ward, hang=-1)
```

Ward's clustering method - summary

Pros

- Tend to form hyperspherical shape clusters, with approximately the same number of elements each (balanced)
- No crossovers
- It is regarded by some authors as a natural hierarchical method to be used with the factorial analysis, such as, PCA, MCA (multiple correspondence analysis), etc, since it seeks to optimize the same type of variance criterion

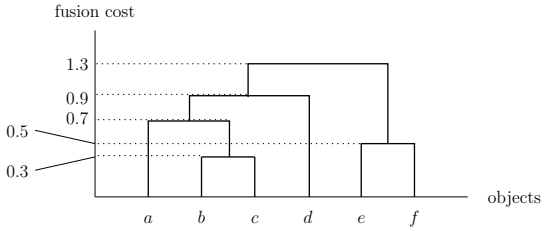
Cons

- Computationally intensive
- Cannot detect arbitrary cluster shapes
- Sensitive to outliers since it relies on a sum of squares measure.

Cophenetic distances

The cophenetic distance between two individuals x and y with respect to a given HAC is the merging cost at which x and y become members of the same cluster, during the course of the hierarchical clustering.

Any dendrogram can be represented by its matrix of cophenetic distances (up to permutation of the order of the leaves), which can be used to compare distinct classifications.



Dendrogram of the example of slide 30

	a	b	c	d	e
b	0.7	·	·	·	·
c	0.7	0.3	·	·	·
d	0.9	0.9	0.9	·	·
e	1.3	1.3	1.3	1.3	·
f	1.3	1.3	1.3	1.3	0.5

Two elements x, y belong to the same cluster of a partition obtained cutting the dendrogram at height τ if and only if their cophenetic distance is less than τ

Distortion measures - Cophenetic Pearson's Coefficient

Cophenetic Pearson's correlation coefficient (CPCC) is Pearson's correlation between the original distances (d_{ij}), $i < j$, and the cophenetic distances (c_{ij}), $i < j$, (using half of the proximity matrix), i.e.,

$$CPCC = \frac{\text{cov}(D, C)}{s_D s_C} = \frac{\sum_{i<j}(d_{ij} - \bar{d})(c_{ij} - \bar{c})}{\sqrt{\sum_{i<j}(d_{ij} - \bar{d})^2}\sqrt{\sum_{i<j}(c_{ij} - \bar{c})^2}}$$

- CPCC is considered an internal validation criterion for hierarchical clustering that can be used to evaluate and compare different hierarchical clustering methods, although should be used with caution
- A high value of the CPCC means that the cophenetic distances are a good portray of the original distances
- The cophenetic correlation usually ranges between 0.6 and 0.95.
- Cophenetic correlations between 0.7 and 0.8 are considered reasonable good, between 0.8 and 0.9 good and above 0.9 very good.

Distortion measures - Cophenetic Spearman's Coefficient

Another distortion measure is cophenetic Spearman's rank order correlation coefficient (CSCC), which only depends on the ranks of the variables and corresponds to Pearson's correlation coefficient between the respective ranked variables $rk(C) = (c'_{ij})$ and $rk(D) = (d'_{ij})$ defined by the vectors of original and cophenetic distances,

$$CSCC = \frac{\text{cov}(rk(D), rk(C))}{s_{rk(D)} s_{rk(C)}} = \frac{\sum_{i<j}(d'_{ij} - \bar{d})(c'_{ij} - \bar{c'})}{\sqrt{\sum_{i<j}(d'_{ij} - \bar{d'})^2}\sqrt{\sum_{i<j}(c'_{ij} - \bar{c'})^2}}.$$

- Unlike the Pearson correlation coefficient, Spearman's rank order correlation coefficient can be applied to compare original and cophenetic dissimilarities even if no linear relation between both dissimilarities exists
- A Spearman's rank order correlation close to 1 means that we have a strong correlation between the ranks of original and the ranks of the cophenetic distances, suggesting monotonic relationship between the original distances and the corresponding cophenetic distances

Cophenetic correlations of example of slide 30

The original d_{ij} distances of the example of slide 30 and the corresponding cophenetic distances c_{ij} for the single, complete and avarage methods are

d_{ij}	a	b	c	d	e
b	0.7	·	·	·	·
c	1	0.3	·	·	·
d	1.8	1.3	0.9	·	·
e	2.9	2.4	1.9	1.3	·
f	3.4	2.8	2.4	1.7	5

c_{ij}^s	a	b	c	d	e
b	0.7	·	·	·	·
c	0.7	0.3	·	·	·
d	0.9	0.9	0.9	·	·
e	1.3	1.3	1.3	1.3	·
f	1.3	1.3	1.3	1.3	0.5

Computing the cophenetic Pearson and Spearman correlation coefficients we obtain:

$$\begin{aligned} CPCC &= r(d_{ij}, c_{ij}) = 0.82 \\ CSCC &= r(rk(d_{ij}), rk(c_{ij})) = 0.84 \end{aligned}$$

HAC - summary

Pros

- The number of clusters does not need to be defined *a priori*
- Many methods rely on a proximity matrix allowing almost any kind of resemblance notion

Cons

- The aggregation of a point in a group at a given step cannot be revised, even if the point is misplaced in that group
- Computationally demanding for large datasets since keeps track of a square matrix of order n (number of individuals): the time and space complexity of most algorithms are not better than $O(n^2 \log(n))$
- Dendrogram difficult to visualize and interpret for large datasets
- Most HAC algorithms are greedy and produce suboptimal solutions

The average and Ward methods are often considered among the best overall HAC methods

Nonhierarchical clustering

To find a single partition into K clusters of a set of N objects in a p dimensional space. Two types of criteria are commonly found:

- Global criterion such as to represent each cluster by a *type-object* (e.g., centroid, medoid) and to assign each object to the nearest *type-object*, optimizing some global criterion of internal homogeneity and/or external heterogeneity, such as, minimizing the within cluster inertia.
  Usually requires a prior estimate of the number of clusters.

  Examples: k-means and k-medoids (PAM) algorithms.
- Local criterion such as to seek for regions of higher density in data. May require to set some parameters.

  Example: DBSCAN.

k-means criterion and algorithm

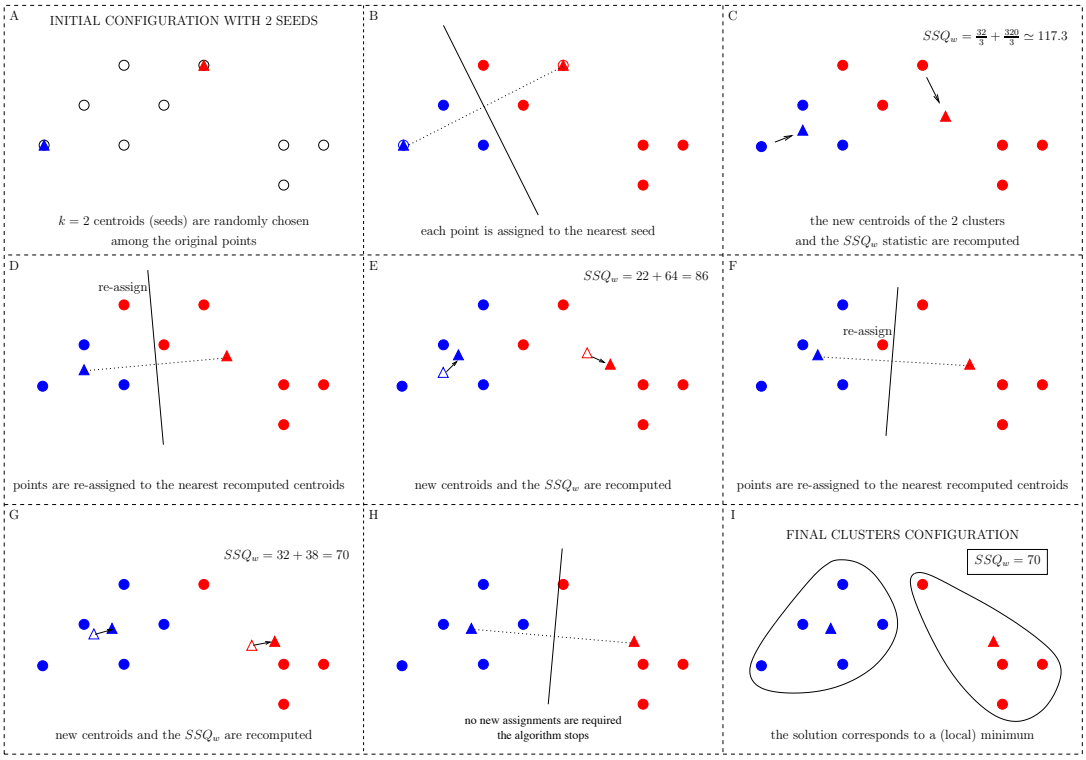
Shares the same global criterion with Ward's method:

> To minimize the total within-clusters sum of squares $\mathbf{SSQ}_w$ of a set of points partitioned into K clusters in a d-dimensional space

Algorithm

1. *Starts with K randomly chosen initial cluster seeds representing initial candidates to centroids;*
2. *Assigns each object to the nearest centroid*
3. *Recomputes the centroids of the K groups and use them as the new seeds*
4. *Repeat the steps 2 and 3 until no new reassignments occur.*

k-means example



Convergence of the k-means algorithm

The k-means algorithm consists essentially of a sequence of two steps that are repeatedly iterated:

- Reassignment of the points of X to the closest centroid - this step clearly lowers the statistic $\mathbf{SSQ}_w = \sum_{k=1}^{K} \sum_{x \in \mathcal{C}_k} \|x - m_k\|^2$
- Recalculation of the centroids of the K groups to use as the new seeds - this step also lowers the $\mathbf{SSQ}_w$ statistic, since it is a well known fact that the minimum of the quadratic function

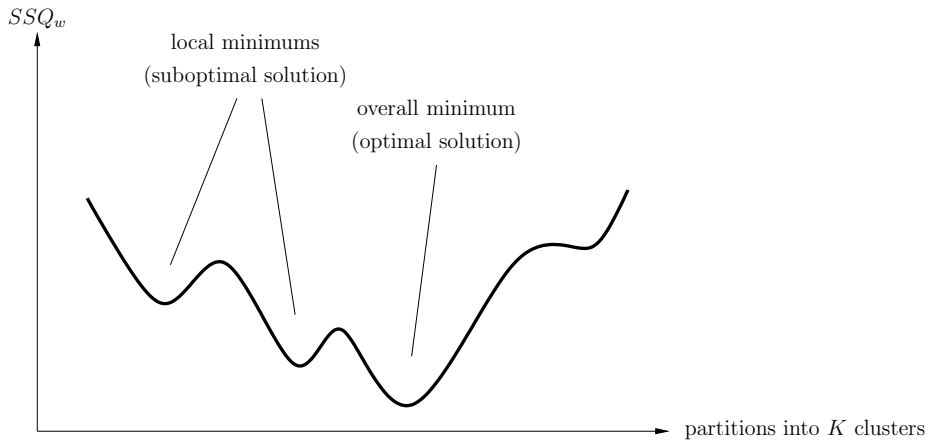
$$f(y) = \sum_{x \in G} \|x - y\|^2,$$

with G a finite subset of $\mathbb{R}^d$, is attained at the centroid of G, i.e., when $y = m_G$

Since there are only finite number of partitions of X into K clusters, the algorithm cannot continue indefinitely strictly lowering the $\mathbf{SSQ}_w$ statistic and therefore has to converge to a (possibly local) minimum.

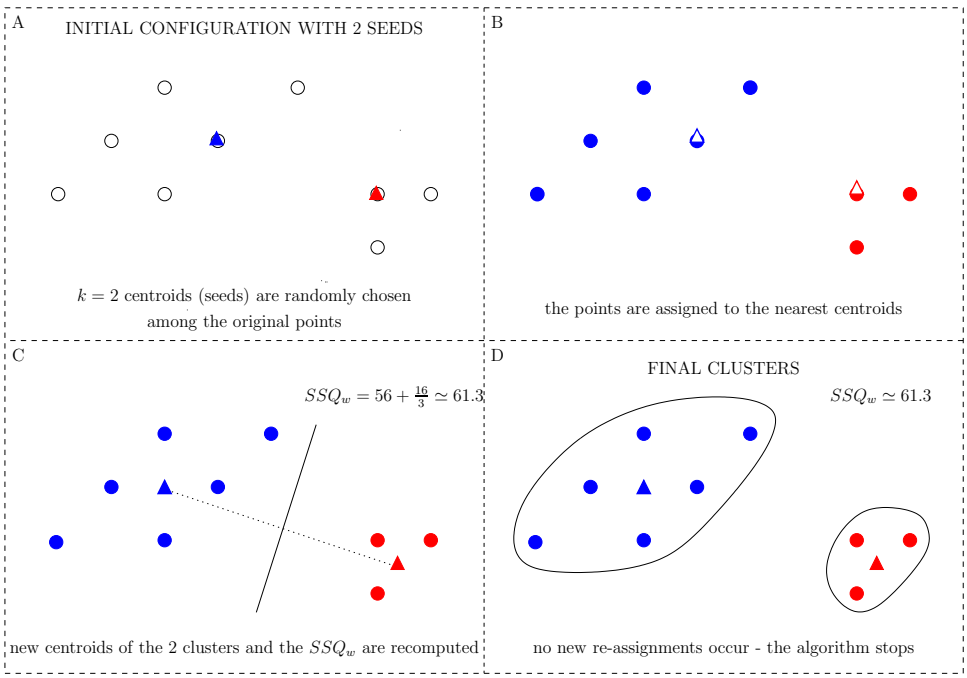
k-means: local minimum problem

The clustering solution can behighly depend on the choice of the initial position of the centroids (seeds) and may converge to a local minimum (suboptimal solution):



k-means example recomputed with new seeds

The solution found by the k-means algorithm in the example of slide 78 is not a global minimum. Actually, with new seeds the algorithm can converge to a solution that improves (i.e., lowers) the $\mathbf{SSQ}_w$ statistic



Possible strategies to improve the local minimum?

- To repeat the algorithm several times with randomized sets of K seed points and keep the configuration giving the smallest $\mathbf{SSQ}_w$ value of the within-cluster inertia
- To provide an initial configuration of K seed points close to the final solution relying on some real hypothesis
- To provide an initial configuration of seed points issued from some hierarchical aggregation method (e.g., Ward), using for instance, their clusters centroids - this is sometimes called the consolidation of the hierarchical clustering

k-means in the plane and the Voronoi diagram

Given a set of N points in the plane,

$$\{c_1, \ldots, c_K\}$$

the Voronoi diagram is defined as the partition of the plane into K convex regions, called Voronoi cells,

$$R_1, \ldots, R_K$$

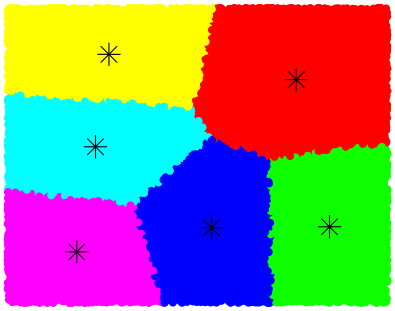
such that each cell R_i consists of the set points of the plane closest to c_i

In each step of the k-means algorithm each cluster corresponds to the set of points of X belonging to one of the Voronoi cells defined be the K centroids c_1,..., c_K, which is called Lloyd's algorithm or Voronoi iteration

The above construction can generalized to a set of K points in the N-dimensional space

The Voronoi partition and its centroids

The partition below into 6 clusters was obtained applying the k-means algorithm to a highly dense set of points in the plane with 6 seeds, to give an approximated idea of the Voronoi cells defined by the final centroids



Each cluster arising from a k-means clustering algorithm lies inside the Voronoi cell containing the respective cluster centroid.
In particular, the convex hulls of the clusters don't overlap, i.e., each pair of clusters can be linearly separated.

Computing k-means with R

The k-means clustering can be performed using the R function

kmeans(x, centers, iter.max $= 10$, nstart $= 1$, ...)

x: numeric matrix of data

centers: the number of clusters or a set of initial (distinct) cluster centres. If a number, a random set of (distinct) rows in x is chosen as the initial centres

nstart: if centers is a number, how many random sets should be chosen (repeat)
Returns a list with components:

cluster: A vector of integers (from 1:k) indicating the number of the cluster where each point is assigned

centers: A matrix of cluster centers.

totss: The total sum of squares, i.e., $\mathbf{SSQ}_t$

withinss: Vector of within-cluster sum of squares, one component per cluster

tot.withinss: Total within-cluster sum of squares, i.e., $\mathbf{SSQ}_w$

betweenss:The between-cluster sum of squares, i.e., $\mathbf{SSQ}_b$

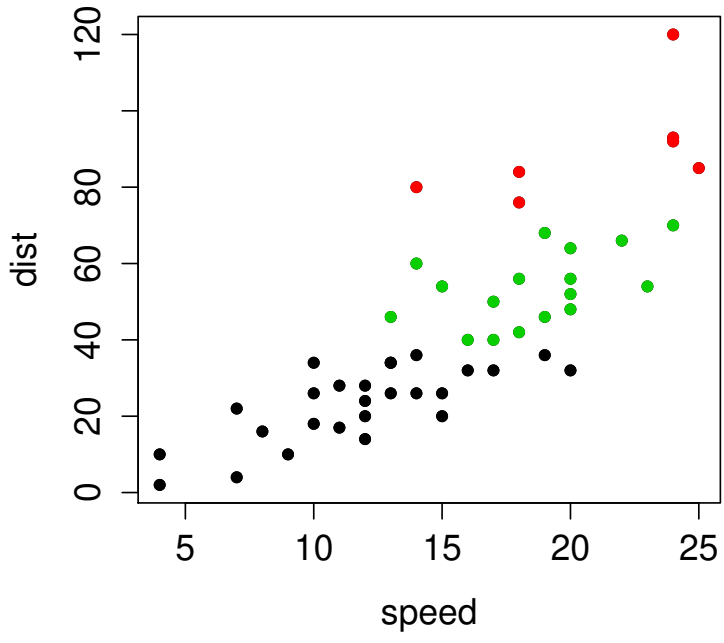
size: The number of points in each cluster

Example

R code

```
require(datasets)

data(cars)

?cars # get information about the cars dataset

head(cars)

# 3 centers randomly chosen repeated 100 times
cars.cl<-kmeans(cars, 3, nstart=100)

cars.cl

plot(cars,type=‘‘p’’,pch=16,cex=.5)

for(i in 1:50){points(cars[i,1],
cars[i,2],col=cars.cl$cluster[i], pch=16,type=‘‘p’’)}
```


Clustering output



k-means: summary

- The optimizing function $\mathbf{SSQ}_w$ is always monotonic decreasing, i.e., the intra-group inertia decreases in each step, converging to some (possibly local) optimum
- The number of iterations required to converge is usually small (≈ 10 iterations are enough)
- Finding an optimal solution is NP-hard. Actually the time complexity is $O(n^{dK+1} \ln d)$, where K denotes the number of clusters, d the dimension and N the number of points)
- Tends to form rounded shaped clusters that can be linearly separated (since each cluster is contained in a Voronoi cell). In particular, it cannot detect arbitrarily shaped clusters
- Nearby points can end in distinct classes. Groups can end empty.
- Sensitive to noise and outliers
- Requires some geometric notion of centroid and cannot be applied to categorical data, since it assumes the points lying in some euclidean space

The model-based clustering as a generalization of k-means

- The standard model-based clustering is a finite mixture of multivariate Gaussians, i.e., it is assumed that each cluster C_i is generated by a multivariate Gaussian distribution with pdf

$$\phi(x|\mu_i, \Sigma_i)$$

  where μ_i and Σ_i are the mean and covariance matrix of C_i.
- One seeks a partition of X into clusters C_i and a mixture of Gaussians with pdf given by a convex combination of the form

$$\phi = \sum_i \eta_i \phi(x|\mu_i, \Sigma_i),$$

  with nonnegative weights η_i, $i = 1, \ldots, K$, such that $\sum_i \eta_i = 1$. To determine the parameters uses the so-called expectation-maximization algorithm.
- In the model-based clustering the partition can have clusters with different covariance matrices i.e., with distinct ellipsoidal shapes, volumes and orientations, that account with distinct weights to the pdf of the finite mixture.
- The k-means clustering can be considered a particular case of the model-based clustering, with all weights η_i equal to $\frac{1}{K}$ and identical isotropic covariance matrices $\Sigma_i = \sigma^2 I$ (I denotes the identity matrix).

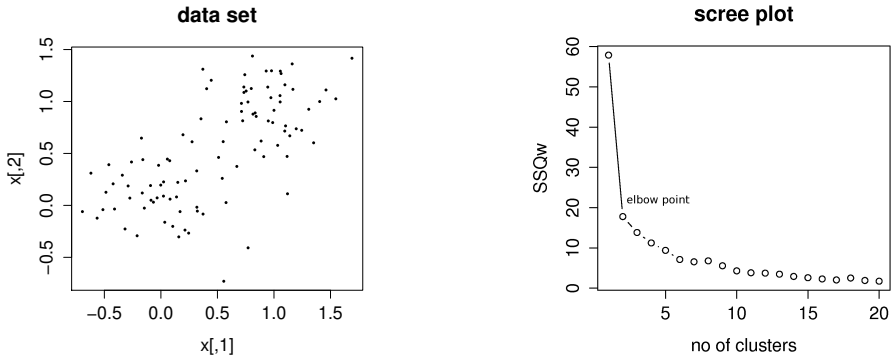
Best number of clusters and internal cluster quality

- To estimate the optimal number of clusters we usually look for a good trade-off between a relatively small number of clusters (parsimony principle) and the minimization of the information (variability) loss due to replacing the observations in each cluster by some cluster representative (for instance, the cluster centroid).
- This is one of the most difficult tasks in clustering analysis and no definitive answer can usually be given.
- Several internal cluster validity indices can be used to estimate the optimal number of clusters and/or to assess the cluster quality. Among the most well-known indices we have:
  - $\mathbf{SSQ}_w$.
  - Calinski-Harabasz index.
  - Silhouette coefficient.
  - Davies-Boudin.
  - Duhn index.
  - Several other indices can be computed with the R functions `clustCrit` and `NbClust`.

For a more detailed account on validity indices, see, for instance, O. Arbelaitz et al. *An extensive comparative study of cluster validity indices*, Pattern Recognition 46 (2013) 243–256

Scree plot of $\mathbf{SSQ}_w$ statistic

- A simple method to estimate the best number of clusters consists to study the variation of $\mathbf{SSQ}_w$ with number of clusters in a scree plot, which essentially amounts, by Huygens's theorem, to study the variation of thepercentage of total inertia retained by the clusters, i.e., explained by the partition, $\frac{\mathbf{SSQ}_b}{\mathbf{SSQ}_t}$
- An elbow point in the scree plot indicating high decrease in the $\mathbf{SSQ}_w$ statistic while further increments in the number of clusters will only marginally improves this statistic, could suggest a good estimate for the optimal number of clusters



- Although the statistic $\mathbf{SSQ}_w$ depends on the number of clusters, it can be used to compare partitions of a given dataset X with the same number of clusters. Partitions yielding smaller $\mathbf{SSQ}_w$ values are preferable for this criterion.

Calinski-Harabaz index

- The Calinski-Harabaz index also known as variance ratio criterion is defined as

$$CH(K) = \frac{\mathbf{SSQ}_b/(K-1)}{\mathbf{SSQ}_w/(N-K)}$$

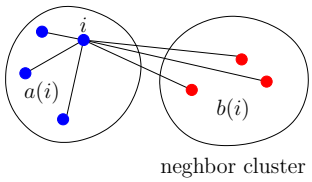
  with the optimal number of clusters being estimated as the number yielding the largest value for $CH(K)$. (Inspired in the F-ratio test of one-way ANOVA)
- Since we have

$$\begin{aligned} CH(K) &= \frac{\mathbf{SSQ}_b/(K-1)}{\mathbf{SSQ}_w/(N-K)} = \frac{N-K}{K-1} \times \frac{\mathbf{SSQ}_b}{\mathbf{SSQ}_w} \\ &= \frac{N-1+1-K}{K-1} \times \frac{\mathbf{SSQ}_b}{\mathbf{SSQ}_w} = \left(\frac{N-1}{K-1} - 1\right) \frac{\mathbf{SSQ}_b}{\mathbf{SSQ}_w}, \end{aligned}$$

  high values of $CH(K)$ are obtained with well separated and homogeneous clusters, i.e., with large values of $\mathbf{SSQ}_b$ and small values of $\mathbf{SSQ}_w$, keeping at the same time, the number of clusters K small, i.e., $\frac{N-1}{K-1}$ relatively large.
- Particularly well adapted when clusters tend to have spherical shapes due to its definition based on the variance
- Some studies suggest Calinski-Harabaz as being one of the best internal cluster validity indices.
  (E.g., Milligan GW, Cooper MC (1985) An Examination of Procedures for Determining the Number of Clusters in a Data Set. Psychometrika 50:159–179.)
- Can be computed using the R function `calinhara` of the package `fpc`

Silhouette coefficient

- For each observation i we compute the average dissimilarity $a(i)$ between i and the remaining points in its cluster
- For each one of the other clusters we compute the average dissimilarity from point i to the points of that cluster and take the minimum $b(i)$ of these average dissimilarities
- The cluster for which the minimum $b(i)$ is attained, i.e., the cluster with lowest average dissimilarity w.r.t to observation i, is called the **neighbor cluster** of i



The silhouette coefficient of observation i is defined as

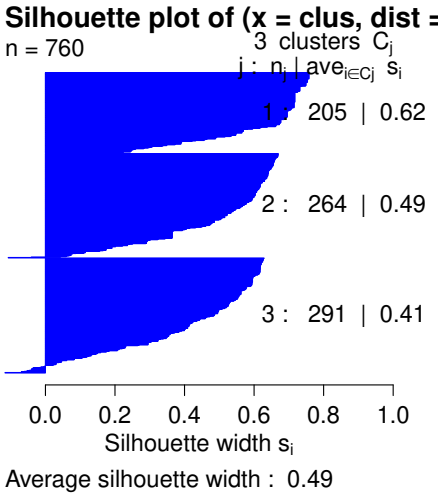
$$s(i) = \frac{b(i)-a(i)}{\max\{a(i),b(i)\}} = \begin{cases} 1 - \frac{a(i)}{b(i)}, & a(i) < b(i) \\ 0, & a(i) = b(i) \\ \frac{b(i)}{a(i)} - 1, & a(i) > b(i) \end{cases}$$

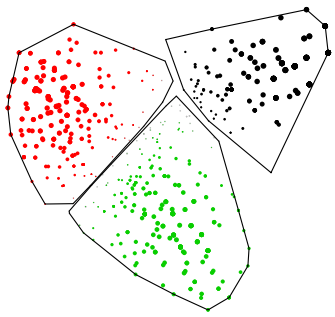
and gives an indication of how well an element is classified in its cluster.

Interpretation of silhouette coefficients

- The denominator $\max\{a(i), b(i)\}$ is a normalization term allowing that the index vary in the range $[-1, 1]$
- Small values of $a(i)$ along with large values of $b(i)$ yield a silhouette coefficient close to one
- Likewise, large values of $a(i)$ along with small values of $b(i)$ yield a silhouette coefficient close to minus one
- Observations with silhouette coefficients close to one are very well classified
- Observations with silhouette coefficients close to zero probably lie between clusters
- Observations with negative silhouette coefficients are probably misplaced in their clusters

Silhouette plot





In the figure on the right the dot sizes are proportional to their silhouette coefficients. Larger dots lie in core regions of the clusters whereas smaller dots lie in border regions or between clusters

Average silhouette width - an internal validity criterion

The average silhouette width (ASW) is defined as the average of the silhouette coefficients for all observations

- It assess both cluster cohesion and cluster separation
- It increases with a strong cluster separation (higher $b(i)$ values) and cluster tightness (small values of $a(i)$)

Range of ASW It is common to consider that

- between 0.71 and 1.0: a strong structure has been found
- between 0.5 and 0.7: a reasonable structure has been found
- between 0.26 and 0.5: the structure is weak and can be artificial
- below 0.25: no substantial structure has been found

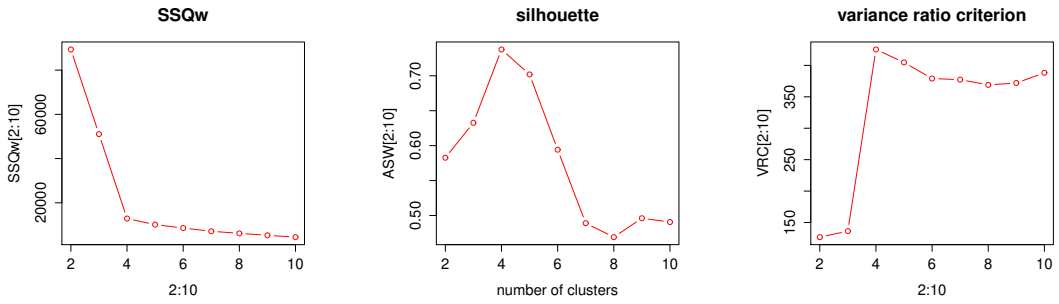
The optimal number of clusters can be estimated maximizing ASW A closely related internal validation criterion is Davies-Bouldin index

$$DB = \frac{1}{K}\sum_{i=1}^{K}\max_{j\neq i}\frac{S_i + S_j}{m_{ij}}$$

Here S_i denotes some internal cohesion measure of cluster C_i and m_{ij} a separation measure between clusters C_i and C_j, verifying certain properties...
For instance, S_i can be the average distance of the points of C_i to its centroid and m_{ij} the distance between the centroids of C_i and C_j

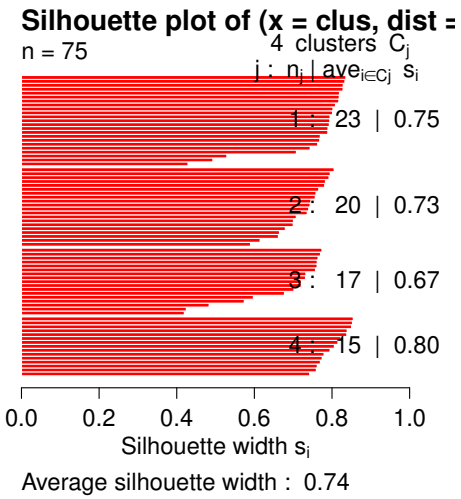
Number of clusters?

Applying the criteria $\mathbf{SSQ}_W$ statistic, ASW and CH to the Ruspini data, a popular dataset in clustering analysis, all criteria agree on 4 clusters

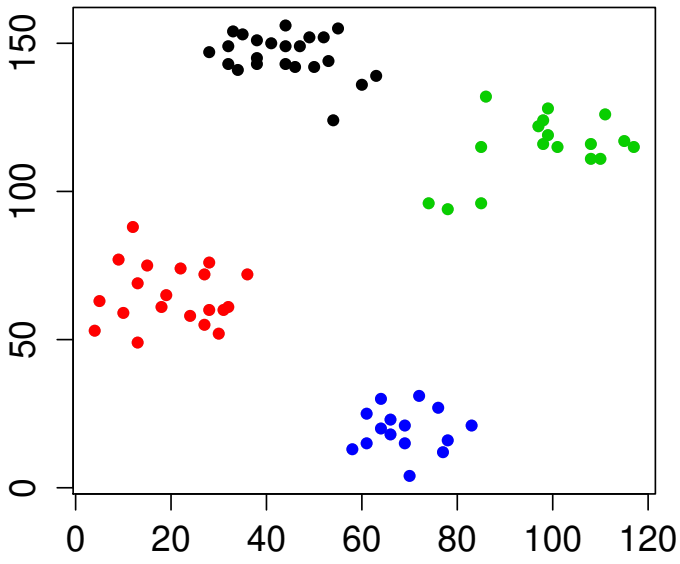


An (internal) cluster validity criterion

The average of the silhouette widths of the previous example is close to .75 suggesting that a strong clustering structure was found in Ruspini data. Since all silhouette coefficients are above .4 no points are misplaced in their clusters



Ruspini plot into 4 clusters using the k-means algorithm



Ruspini plot into 4 clusters using the k-means algorithm

R code

```
library(cluster)
ch.res<-rep(NA,10)
si.res<-rep(NA,10)
ssqw.res<-rep(NA,10)
plot(ruspini)
for (n in 2:10){
   km <- kmeans(ruspini,n,nstart=500)
   ch.res[n]<-round(calinhara(ruspini,km$cluster),digits=2)
   si.res[n]<-mean(silhouette(km$cluster,dist(ruspini))[,3])
   ssqw.res[n]<-km$tot.withinss
   # (or ssqw.res[n]<-km$betweenss/km$tot.withinss)
}
par(mfrow=c(2,2))
plot(ssqw.res,type="b",col="black",main="SSQw")
plot(si.res,type="b",col="blue",main="SIL")
plot(ch.res,type="b",col="red",main="CH")
km <- kmeans(ruspini,4,nstart=500)
plot(ruspini, col=km$cluster)
```


Comparing Partitions

Motivation

- Several clustering analyses of the same data can be done using distinct meaningful combinations of clustering methods and resemblance notions;
- Clustering analyses having a high degree of agreement may suggest that the common patterns produced by these methods is robust;
- If the clustering structure is known *a priori* and it is important to assess how well the clustering method was able to reproduce this structure;
- It is very difficult (if not impossible or meaningless) to match each cluster of a partition with the correct cluster of the other partition
- The usual way is to compute the number of pairs of individuals that both clustering methods agree to assign in the same/distinct class

Rand index

- Assume that N individuals are classified by two distinct clustering methods. The total number of pairs of individuals is $\binom{N}{2} = \frac{N(N-1)}{2}$. Denote by:

  A: number of pairs classified in the same class in both partitions

  B: number of pairs classified in the same [distinct] class in the first [second] partition

  C: number of pairs classified in the distinct [same] class in the first [second] partition

  D: number of pairs classified in distinct classes in both partitions
- The above quantities can be represented in a contingency table as follows:

	Part. 2		
Part. 1	Classif. in the same group	Classif. in distinct groups	
Classif. in the same group	A	B	A+B
Classif. in distinct groups	C	D	C+D
	A+C	B+D	$\binom{N}{2}$

Rand index

- Rand index (RI) is a simple concordance index used as an external validity index to compare partitions and is defined as,

$$RI = \frac{A+D}{\binom{N}{2}} = \frac{A+D}{A+B+C+D},$$

  where $A+D$ is the number of agreements for both partitions
- It ranges from 0 (total disagreement) to 1 (total agreement).
- To each partition of a set of N individuals, $x_1, \ldots, x_N$ we associate a binary vector of length $\binom{N}{2}$, where the component corresponding to pair (i, j) is equal 1 if x_i and x_j are assigned in the same class and 0 otherwise
- The Rand index of two partitions is just the simple matching index between the binary vectors associated to these partitions
- Note that the number of groups in each partition can be distinct

Rand index: example

$X = \{a, b, c, d, e, f\}$

Partition 1: $a\ b\ e \mid c \mid d\ f$ Partition 2: $a\ c \mid b\ d \mid e\ f$

	a	b	c	d	e
b	1	·	·	·	·
c	0	0	·	·	·
d	0	0	0	·	·
e	1	1	0	0	·
f	0	0	0	1	0

	a	b	c	d	e
b	0	·	·	·	·
c	1	0	·	·	·
d	0	1	0	·	·
e	0	0	0	0	·
f	0	0	0	0	1

The contingency table between partition 1 and partition 2 is

	1	0	
1	A	B	$A+B$
0	C	D	$C+D$
	$A+C$	$B+D$	$\binom{N}{2}$

$=$

	1	0	
1	**0**	4	4
0	3	**8**	11
	3	12	**15**

Hence

$$RI = \frac{\mathbf{0+8}}{\mathbf{15}} = 0.53333\ldots$$

Computing the Rand index in R

To compute the Rand index of the two partitions in 3 classes,

$$\mathcal{P}_1: \quad a\ b\ e \mid c \mid d\ f \qquad\qquad \mathcal{P}_1: \quad a\ c \mid b\ d \mid e\ f,$$

we encoded these partitions as vectors as

$$(1,1,2,3,1,3), \qquad (1,2,1,2,3,3),$$

representing the classes of the elements a, b, c, d, e, f.

R code

```
# Codigo da funcao do Professor Cadima
rand <- function(class1,class2){
    n <- length(class1)
    c <- as.dist(outer(class1,class1,"=="))
    d <- as.dist(outer(class2,class2,"=="))
    rand <- sum(c == d)/(n*(n-1)/2)
    return(rand) }

rand(c(1,1,2,3,1,3),c(1,2,1,2,3,3)) # 0.5333333

# 2 random samples of length 1000 with elements extracted from 1,...,10
p1<-sample(1:10,1000,replace=TRUE)
p2<-sample(1:10,1000,replace=TRUE)
rand(p1,p2) # 0.8196997
```


Correction for chance: adjusted Rand index

The expected value of Rand index between random partitions is not constant (e.g., 0). To overcome this issue Hubert and Arabie proposed the so-called adjusted Rand index

$$ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]} = \frac{RI - E[RI]}{1 - E[RI]},$$

assuming the Permutation Model as the null model for random clusterings i.e., each partition $\mathcal{P}_i$, $i = 1, 2$, is drawn at random, subject to having a prescribed number of classes K_i and a prescribed number of elements $N_{i,j}$ in each class $j = 1, \ldots, K_i$.

It can be proved that,

$$E[RI] = \frac{2Q_1\,Q_2 - \binom{N}{2}(Q_1 + Q_2) + \binom{N}{2}}{\binom{N}{2}^2},$$

where $Q_i = \sum_{j=1}^{K_i} \binom{N_{ij}}{2}$, $i = 1, 2$, yielding

$$ARI = \frac{\binom{N}{2}(A + D) - U}{\binom{N}{2}^2 - U},$$

where $U = (A + B)(A + C) + (D + B)(D + C)$ and $\binom{N}{2} = \frac{N(N-1)}{2}$.

$ARI \in [-1, 1]$ with $ARI \approx 0$ for independent random partitions, $ARI = 1$ for identical partitions and $ARI < 0$ if the partitions have a low agreement.

More difficult to interpret than the more simple Rand index

Computing the adjusted Rand index

Consider the two partitions in 3 classes of slide 141,

$$\mathcal{P}_1: \quad a\ b\ e \mid c \mid d\ f \qquad\qquad \mathcal{P}_2: \quad a\ c \mid b\ d \mid e\ f,$$

By the results of slide 141, we have $U = (A+B)(A+C) + (D+B)(D+C) = 144$, $\binom{N}{2} = A+B+C+D = \frac{N(N+1)}{2} = 15$ and we therefore we get

$$ARI = \frac{\binom{N}{2}(A+D) - U}{\binom{N}{2} - U} = -0.2962963$$

We can recompute this index using the `adjustedRandIndex` function of the MCLUST package. In order to accomplish that we consider the vectors v_1 e v_2 representing the classes of the elements a, b, c, d, e, f, in the two partitions, $\mathcal{P}_1$ and $\mathcal{P}_2$, that is,

$$v_1 = (1, 1, 2, 3, 1, 3), \qquad v_2 = (1, 2, 1, 2, 3, 3)$$

R code

```
require(mclust)

# with the partitions above we get,
adjustedRandIndex(c(1,1,2,3,1,3),c(1,2,1,2,3,3))= -0.2962963

# with the same random samples of slide 106 we get,
adjustedRandIndex(p1,p2)=0.0002526569 ≈ 0
```


CLUSTERS INTERPRETATION

Main idea

- To interpret and describe a group obtained from a cluster analysis we can rely on variables that participate in the clustering process, called active variables, and/or on context variables added at a posterior stage, called supplementary or illustrative variables.

- Intuitively a variable is considered more or less characteristic of a group if it is more or less *unlikely that the values of the variable in the group has be drawn at random among the set of observed values*.

The case of continuous variables - notations

- In the following $\mathbf{X}$ denotes a $N \times (p+q)$ dataset containing $p+q$ columns $\mathbf{x}_1, \ldots, \mathbf{x}_p, \mathbf{y}_1, \ldots, \mathbf{y}_q$ corresponding, respectively, to the observed values across N individuals of p continuous variables and q categorical variables.
- The set of N individuals is decomposed into k disjoint groups $C_1, \ldots, C_k$, with $n_1, \ldots, n_k$ elements, respectively. In particular, $n_1 + \cdots + n_k = N$.
- For each continuous variable j, we denote by $\bar{x}_j$ and s_j^2 be the (sample) mean and (sample) variance of the vector of observations $\mathbf{x}_j$.
- Given a cluster C_ℓ containing n_ℓ individuals we denote by $\mathbf{x}_{j,\ell}$ the vector containing the values of variable j w.r.t. to the individuals belonging to cluster C_ℓ and by

$$\bar{x}_{[\ell],j} = \frac{1}{n_\ell} \sum_{i \in C_\ell} x_{i,j}, \qquad s^2_{[\ell],j} = \frac{1}{n_\ell - 1} \sum_{i \in C_\ell} (x_{i,j} - \bar{x}_{[\ell],j})^2,$$

the corresponding (sample) mean and (sample) variance.

Statisitic description of a cluster by continuous variables

- To describe a cluster by means of the continuous variables we compare the mean of each variable in the cluster and the overall mean of the variable in the sample.
- Let $X_{j,\ell}$ be the random variable, mean of n_k values drawn from the n values of x_j.
- Under the null hypothesis that the n_k values are randomly drawn without replacement we have,

$$E[X_{j,k}] = \bar{x}_j \quad \text{and} \quad Var(X_{j,k}) = \frac{s_j^2}{n_k} \frac{n - n_k}{n - 1}.$$

- If n and n_k are not too small we have, by the central limit theorem, that the random variable

$$\frac{\bar{X}_{j,k} - \bar{x}_j}{\sqrt{\frac{s_j^2}{n_k} \frac{n-n_k}{n-1}}}$$

follows, approximately, the standard normal distribution $\mathcal{N}(0, 1)$ (even in this case where the draws are not independent).

V-test (supplementary variables)

- Under the null hypothesis, the probability (p-value) that the absolute value of the difference between $X_{j,k}$ and overall mean x_j, is more extreme than the absolute value of the difference between of observed means is the same that a variable following a standard Gaussian distribution attains an absolute a value greater than $|t_{j,k}|$, where $t_{j,k}$ denotes the quantity, called v-test,

$$\frac{\bar{x}_{j,k} - \bar{x}_j}{\sqrt{\frac{s_j^2}{n_k} \frac{n-n_k}{n-1}}}.$$

- The greater the absolute value of the v-test, the smaller the p-value and hence the more unlikely is that the n_k values of x_j in C_k were randomly drawn from the population, i.e., the more the variable characterizes the group C_k.
- For instance, when $|t_k| > 1.96$, the p-value is smaller than 0.05, meaning than the variable characterizes the group at a significance level 5%.
- The variables are then sorted in decreasing order of their absolute v-test values, i.e., in decreasing order of importance to characterize the group C_k.
- If the sign of v-test is positive [negative], the group is characterized by strong [weak] values, ie. values above [below] the overall mean.

Extension to active variables and remarks

- The results of the previous slides can only be interpreted probabilistically, i.e., in termos of tests of hypothesis, for supplementary variables, since these variables didn't participate in the construction of the clusters.
- Nevertheless, the v-test can be still be used as a simple similarity measure between active variables and classes, allowing to range the variables by order of importance for the class.
- The v-test,

$$t_k = \frac{\bar{x}_k - \bar{x}_j}{\sqrt{\frac{s^2}{n_k}\frac{n-n_k}{n-1}}} = \sqrt{n_k}\frac{\bar{x}_k - \bar{x}_j}{\sqrt{s^2\frac{n-n_k}{n-1}}},$$

  is sensitive to the number of observations in the sample: if we multiply the size of a sample by M then the v-test is (approximately) multiplied by $\sqrt{M}$. As a consequence for large groups all variables tend to be became significant and hence it is very difficult to establish a threshold.

The case of nominal variables

- The description of a cluster by modalities of nominal variables relies on the comparison between the proportion of a each modality in the group and the proportion of the same modality in the population.
- A modality ℓ of a nominal variable characterizes a cluster C_k if theproportion in the group of elements having that modality is considered significantly greater [smaller] than the proportion of the modality in the population.
- If $N_{\ell k}$ is a random variable representing the number of elements having modality ℓ in cluster C_k then, under the null hypothesis, $N_{\ell k}$ follows a hypergeometric distribution with parameters n,m_ℓ and n_k/n, where m_ℓ is the proportion of elements with modality ℓ in the population.
- As before, we call v-test is the quantile of a standard Gaussian variable corresponding to the probability $P(N \geq n_{kj}$.
- We can range the modalities by their v-test values.
- The same remarks apply with respect to active and supplementary variables.

Some bibliography

C J. Cadima, Introduction to Multivariate Statistics MMA course 2021/22

LL P. Legendre, L. Legendre, *Numerical Ecology* (2003)

E B. S. Everitt, *Cluster Analysis* (1993)

TK S. Theodoridis and K. Koutroumbas, *Pattern Recognition* (2009)

R A. C Rencher, *Methods of Multivariate Analysis* (2002)

MRS C. Manning, P. Raghavan, H. Shutze, *An Introduction to Information Retrieval* (2009)

G B. Grün, *Model-based Clustering*, arXiv: 1807.01987 (2018)

HA L. Hubert and P. Arabie, *Comparing Partitions*, Journal of Classification 2 (1985)

LMP L. Lebart, A. Morineau and M. Piron, *Statistique exploratoire multidimensionnelle*, Dunod (1995)

M A. Morineau, *Note sur la caractérisation statistique d'une classe et les valeurs-tests*, Bull. Techn. du Centre de Statist. et d'Infor. Appl., 2, P 20-27 (1984)