CLUSTER ANALYSIS

Definition of clustering

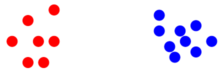
Given a collection of N objects, $X = \{x_1, \ldots, x_N\}$, one seeks a partition of X into K nonempty disjoint sets (the *clusters*),

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

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

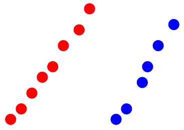
- maximizes the internal homogeneity or cluster cohesion, or equivalently, it minimizes the intra-cluster variability - objects belonging to the same cluster should share the similar features
- it maximizes the external heterogeneity or cluster separation, i.e., it maximizes the inter-cluster separability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

Examples



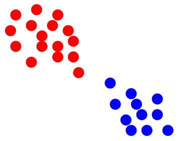
strong internal cohesion

strong separation



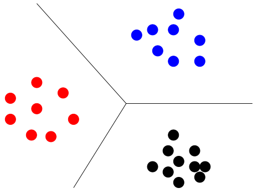
weak internal cohesion

strong separation

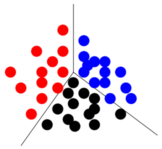


strong internal cohesion

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

$$\xi(25,8) = 690223721118368580$$

For N large and K fixed, $\xi(N,K) \approx \frac{K^N}{K!}$

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 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

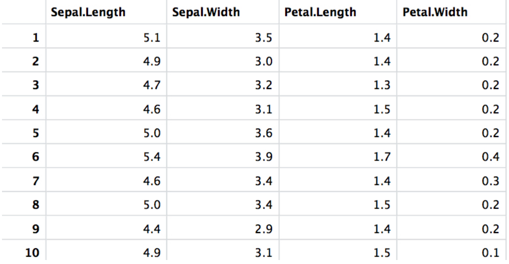
A cluster model is build upon two concepts:

- 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

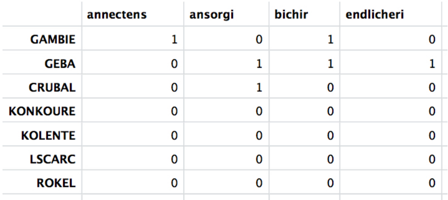


	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5.0	3.6	1.4	0.2
6	5.4	3.9	1.7	0.4
7	4.6	3.4	1.4	0.3
8	5.0	3.4	1.5	0.2
9	4.4	2.9	1.4	0.2
10	4.9	3.1	1.5	0.1

- 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 categorical dataset

```
The following two-way contingency table encodes  the  country of residence and
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

({\bf 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:

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

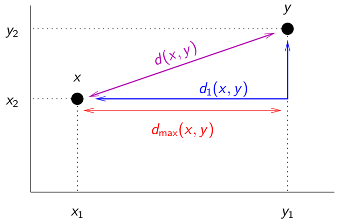
- The Manhattan distance (also called city block or taxicab distance):

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

- 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

The 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 is a 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 represent unequal valued attributes, like species presences-absences):

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

Example

Assume that we have two binary variables **x** and **y** representing presences (1) and absences (0) of two species at 16 spots:

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

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 $d = 9$ ($a+b+c+d = 16$). Therefore we have.

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

The asymmetrical character of 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

# 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, $x_{i.} = \sum_j x_{ij}$, we obtain the so-called ith row-profile, $\left(\frac{x_{i1}}{x_{i.}}, \ldots, \frac{x_{iJ}}{x_{i.}}\right)$, which corresponds to the conditional distribution of variable B assuming category a_i of A.
- The set of the I row-profiles defines a cloud of I points in $\mathbb{R}^J$ and the centroid of this cloud, $\frac{1}{I}\sum_i \left(\frac{x_{i1}}{x_{i.}}, \ldots, \frac{x_{iJ}}{x_{i.}}\right) \in \mathbb{R}^J$, is called is the mean row-profile.
- If variables A and B are independent, i.e., $x_{ij} = \frac{x_{i.}x_{.j}}{N}$ $\forall i,j$, ith row-profile verifies
$$\left(\frac{x_{i1}}{x_{i.}}, \ldots, \frac{x_{iJ}}{x_{i.}}\right) = \left(\frac{x_{.1}}{N}, \ldots, \frac{x_{.J}}{N}\right) = (f_{.1}, \ldots, f_{.J}),$$
where $f_{.j} = \sum_i f_{ij}$ are the column marginals of the relative frequencies $f_{ij} = \frac{x_{ij}}{N}$. In particular, 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,
$$d^2_{\chi^2}(i,\ell) = \sum_{j=1}^{J} \frac{1}{f_{.j}}\left(\frac{x_{ij}}{x_{i.}} - \frac{x_{\ell j}}{x_{\ell .}}\right)^2 = \sum_{j=1}^{J} \frac{1}{f_{.j}}\left(\frac{f_{ij}}{f_{i.}} - \frac{f_{\ell j}}{f_{\ell .}}\right)^2$$
(the weights in the inverse proportion of the column marginal frequencies $f_{.j}$ increase the importance of the small differences between rare categories).

Example

Consider again the two-way contingency table containing the distribution by **country of residence** of the **primary language spoken** of 5000 inhabitants (see slide 13)

	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	(verify)
f.j	0.4496	0.124	0.0538	0.1426	0.230	1.000	(verify)

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

The 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

```
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 computed 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 d take values in $[-1, 1]$ and both the strength of the linear relationship and its direction are 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 $x_1, \dots, x_N$

- *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 $N - 1$ times, until only one 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 these merged clusters

Once two individuals are grouped together they cannot be separate 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
- ...

Updating 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 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 updating 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 $\mathbf{X}$.

Examples 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

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

Dendrogram

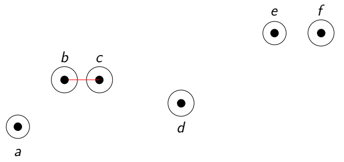
The sequence of length $N-1$ of the merged clusters and the corresponding fusion costs (i.e., the distance between the merged clusters) can be graphically represented by a special tree graph called dendrogram

- Dendrograms are tree-like diagrams made of branches that join terminal nodes (*leaves*)
- The branches represent clusters and the heights at which the branches are connected represent fusion costs. The leaves represent the objects
- 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 apply the single-linkage clustering algorithm to a set of 6 points

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



PROXIMITY MATRIX

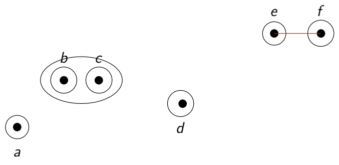
	a	b	c	d	e
b	0.7				
c	1.0	0.3			
d	1.8	.1.3	0.9		
e	2.9	2.4	1.9	1.3	
f	3.4	2.8	2.4	1.7	.5

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

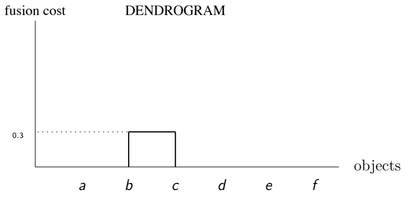
$X = \{a, b, c, d, e, f\}$



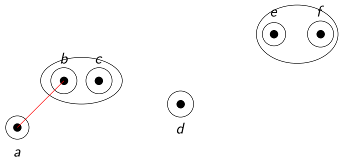
PROXIMITY MATRIX

	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



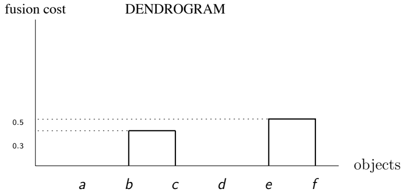
Step - 3



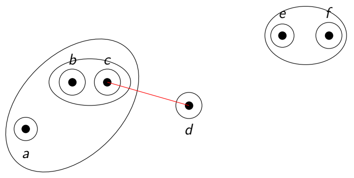
PROXIMITY MATRIX

	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

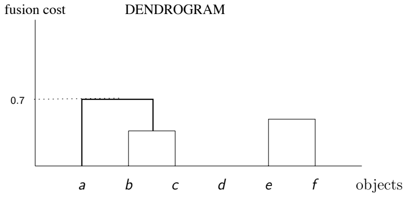


PROXIMITY MATRIX

	$\{a,b,c\}$	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

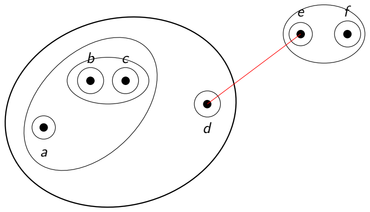
DENDROGRAM



Step - 5

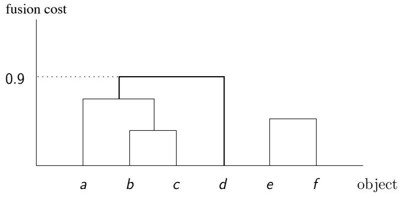
PROXIMITY MATRIX

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



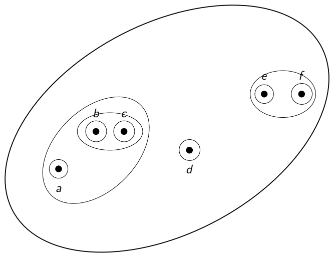
Next step is the final one and merges the clusters $\{a,b,c,d\}$ and $\{e,f\}$ with fusion cost 1.3

DENDROGRAM



step - 6 (final step)

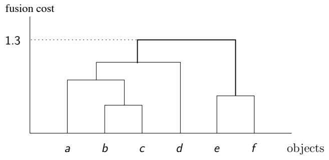
The final structure of nested clusters and the dendrogram encoding the clustering procedure are the following

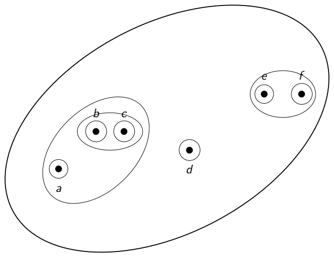


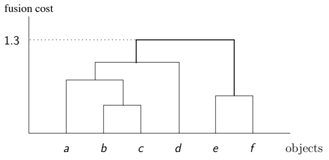
PROXIMITY MATRIX

EMPTY

DENDROGRAM







step - 6 (final step)

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

PROXIMITY MATRIX

EMPTY

DENDROGRAM

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 (hclust function)

```
hc<-hclust(d, method=‘‘complete’’, members=NULL)
plot(hc) or plot(hc, hang=-1) to plot the dendrogram with all
leaves at the same height
```

Example

R (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

plot(SL, hang=-1) # plot the dendrogram
```

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

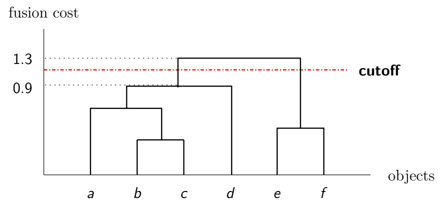
Some caution has to be applied regarding the decision where to cut the dendrogram (and what is the “best” number of clusters). With some methods (for instance, the Ward method), the dendrogram lifetimes tend to increase when the larger clusters are merged, due to the way the fusion costs are defined

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 $\mathcal{C}' = \{e, f\}$

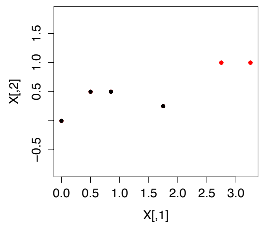
CUTTING THE DENDROGRAM



- 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 (cutree function)

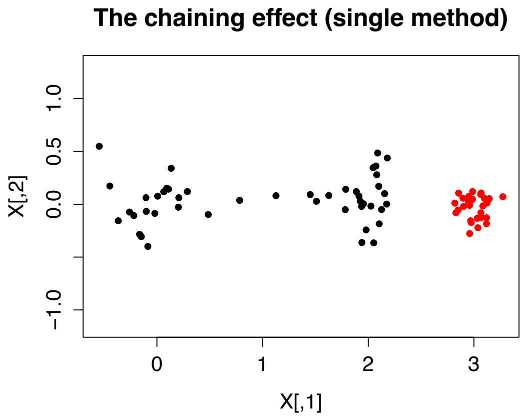
```
SL<-hclust(X,method="single")
part<-cutree(SL,2)  # 2 clusters
# # or
part<-cutree(SL,h=1.1)  # h is the height
part
plot(X,type="p",cex=0.8,pch=16, col=part,asp=TRUE)
```

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 (chain 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

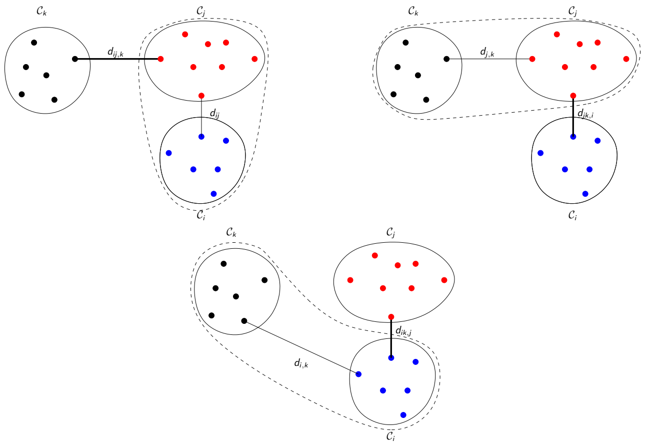
$$\min_{i \neq j} D(C_i, C_j) = \min_{i \neq j} \left(\min_{x \in C_i,\, y \in C_j} d(x, y) \right) .$$

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.

Therefore we have the following.

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

Single-linkage and clusters separation



The aggregation of the pair of closest clusters (top row, on the left) 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
- It is 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

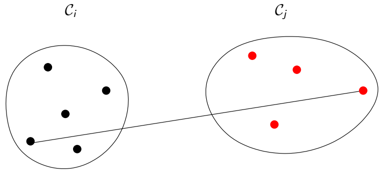
Cons

- 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.,

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

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 $C_1, \ldots, C_k$:

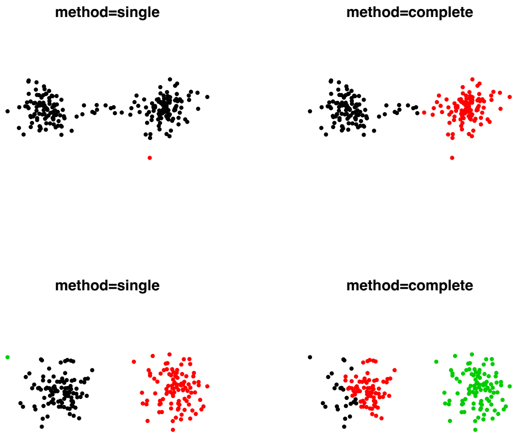
$$\max_i \operatorname{diam}(C_i) = \max_i \left(\max_{x,y \in C_i} d(x,y) \right).$$

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. Hence we have

> *The complete-linkage clustering algorithm tends to produce compact clusters (but not necessarily well separated!)*

Noise and outliers: single vs complete aggregation 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
- It is 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 of aggregate two cluster only relies on a pair of individuals, one in each cluster

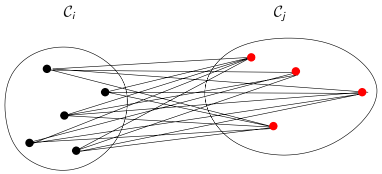
Average clustering method

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 the arithmetic 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 updating formula is given by (left as an exercise),

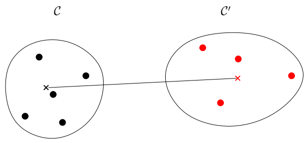
$$\boxed{d_{ij,k} = \frac{n_i d_{i,k} + n_j d_{j,k}}{n_i + n_j}}$$

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.

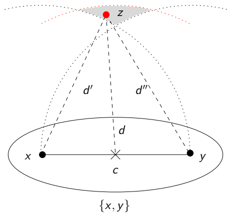
Exercise

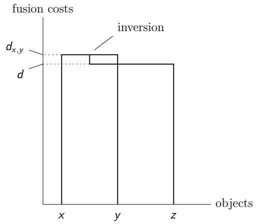
- Perform a clustering analysis using the centroid method on the set of 3 points of $\mathbb{R}^3$, $X = \{(0,0),(8,0),(4,7.5)\}$ and represent the respective dendrogram
- What happened ?

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 general updating formula

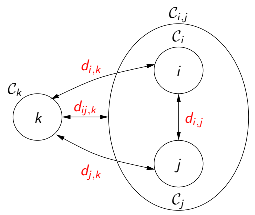
Given clusters $\mathcal{C}_i$, $\mathcal{C}_j$, $\mathcal{C}_k$ and $\mathcal{C}_{ij} = \mathcal{C}_i \cup \mathcal{C}_j$ we will define updating formulas for a family of clustering methods

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

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 $n_i + n_j = |C_{ij}|$:



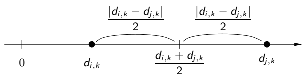
Lance-Williams updating formula - examples

Let us see how to obtain the updating formulas for the single-linkage and complete linkage of slides 39 and 56 (verificar!)

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

from the Lance-Williams table. We can assume $d_{i,k} \leq d_{j,k}$. Therefore

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


$\frac{|d_{i,k} - d_{j,k}|}{2}$
$\frac{|d_{i,k} - d_{j,k}|}{2}$
0
$d_{i,k}$
$\frac{d_{i,k} + d_{j,k}}{2}$
$d_{j,k}$

Hence the Lance-Williams coefficients for the single-linkage and complete-linkage, are:

$\alpha_i = \alpha_j = \dfrac{1}{2}$, $\gamma = -\dfrac{1}{2}$ and $\beta = 0$ (single-linkage)

$\alpha_i = \alpha_j = \dfrac{1}{2}$, $\gamma = \dfrac{1}{2}$ and $\beta = 0$ (complete-linkage)

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

Example: updating 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 124 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 the clustering aggregation methods, *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 with N individuals, $\mathbf{x}^1, \ldots, \mathbf{x}^N$ in p (observed) variables with mean vector $\mathbf{x}^G = (\bar{x}_1, \ldots, \bar{x}_p)$. Given a partition of $\mathbf{X}$ into K clusters

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

we define,

- $SSQ_t = \sum_{i=1}^{N} \|\mathbf{x}^i - \mathbf{x}^G\|^2 = \sum_{i=1}^{N} \sum_{j=1}^{p} (x_{ij} - \bar{x}_j)^2$ (total inertia)
- $SSQ_b = \sum_{k=1}^{K} n_k \|\mathbf{m}_k - \mathbf{x}^G\|^2$ (between-clusters inertia)
- $SSQ_w = \sum_{k=1}^{K} \sum_{\mathbf{x} \in \mathcal{C}_k} \|\mathbf{x} - \mathbf{m}_k\|^2$ (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 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 $\mathbf{m}_k$.
- The total within-clusters inertia 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.
- By Huygens theorem, $SSQ_t = SSQ_b + SSQ_w$, which is a constant.
- Ward's clustering method, also called minumum variance criterion, tries to minimize the total within-clusters inertia SSQ_w, i.e., the clusters heterogeneity/variability, which, by Huygens theorem, amounts to maximize the between-clusters inertia 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,

$$SSQ_t = SSQ_b, \qquad SSQ_w = 0$$

- 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 SSQ_w
- We shall write SSQ_w as

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

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 SSQ_w reduces to the following statistic,

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

  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 SSQ_t.

- It can be proved that

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

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

- In particular, $\Delta_{ij} SSQ_w$ is always nonnegative (i.e., the 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 updating formula for Ward's method using LW

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

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

  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 previous 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 $\Delta_{ij,k} SSQ_w$ 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

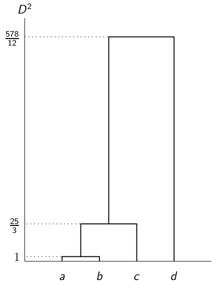
$$\left[\begin{array}{c|cc} D^2 & a \cup b & c \\ \hline c & \frac{25}{3} & \\ d & \frac{169}{3} & 16 \end{array}\right]$$

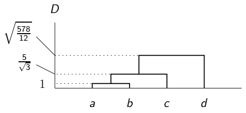
The minimum of the squared distances is attained for $D^2(a \cup b, c)$ so the next pair to be clustered will be $(a \cup b) \cup c$ with squared fusion cost $\frac{25}{3}$

Ward clustering using LW updating formula (concl.)

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

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





Computing using the R function *hclust*

The previous dendrogram can also be computed using the R software in the following way:

R (Ward's method)

```
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 variance criterion
- The sum of all dendrogram heights is equal to $2\times SSQ_t$.

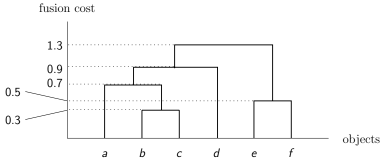
Cons

- Computationally intensive
- Cannot detect arbitrary cluster shapes
- Sensitive to outliers since it uses centroids

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. This matrix can be used to compare distinct classifications



$$\left[\begin{array}{c|ccccc} & a & b & c & d & e \\ \hline b & 0.7 & \cdot & \cdot & \cdot & \cdot \\ c & 0.7 & 0.3 & \cdot & \cdot & \cdot \\ d & 0.9 & 0.9 & 0.9 & \cdot & \cdot \\ e & 1.3 & 1.3 & 1.3 & 1.3 & \cdot \\ f & 1.3 & 1.3 & 1.3 & 1.3 & 0.5 \end{array}\right]$$

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

The **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.,

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

- 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 the **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,

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

- 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 100

The original d_{ij} distances of the example of slide 100 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^s_{ij}	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,

$$CPCC = r(d_{ij}, c_{ij}) = 0.82, \qquad CSCC = r(rk(d_{ij}), rk(c_{ij})) = 0.84$$