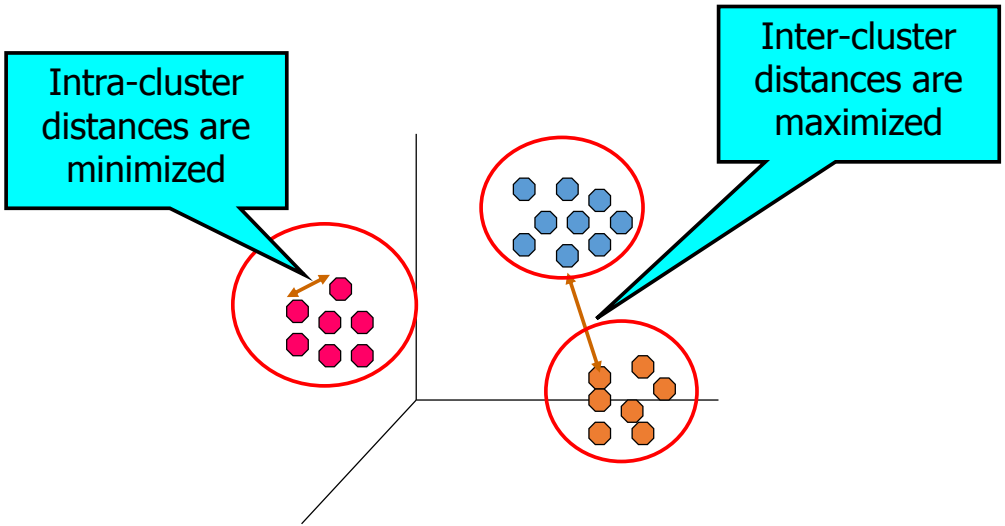
5. Cluster Analysis

Isabel M. Rodrigues

What is Cluster Analysis?

Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups



What is Cluster Analysis?

- Cluster: a collection of data objects
  Similar to one another within the same cluster
  Dissimilar to the objects in other clusters
- Cluster analysis
  Grouping a set of data objects into clusters
- Clustering is unsupervised classification:
  no predefined classes
- Typical applications:
  As a stand-alone tool to get insight into data distribution
  As a preprocessing step for other algorithms

Introduction: Clustering Applications

- Marketing: discovering of distinct customer groups
- Land use: Identification of areas of similar land use in an earth observation database
- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost
- City-planning: Identifying groups of houses according to their house type, value, and geographical location

Introduction: Clustering Applications

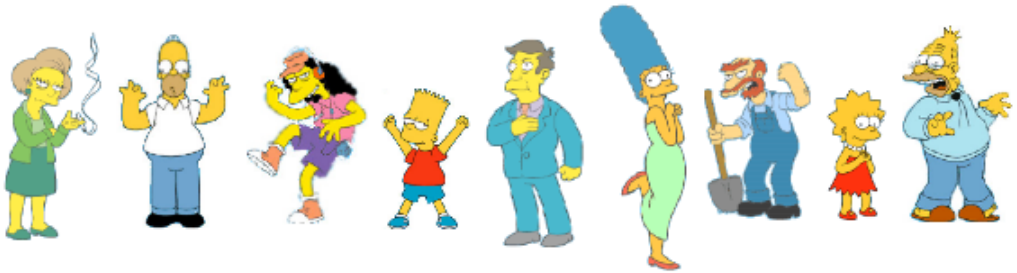
- Pattern Recognition
- Data Mining
- Image Processing
- Economic Science (especially market research)
- WWW
  - Document classification
  - Weblog clustering to identify groups of users

Introduction: Clusters analysis

- The notion of a "cluster" cannot be precisely defined. There is a common denominator: a group of data objects with homogeneity and separation principles
- A clustering method differs in the different notions of clusters and in the different notions of similarity/proximity

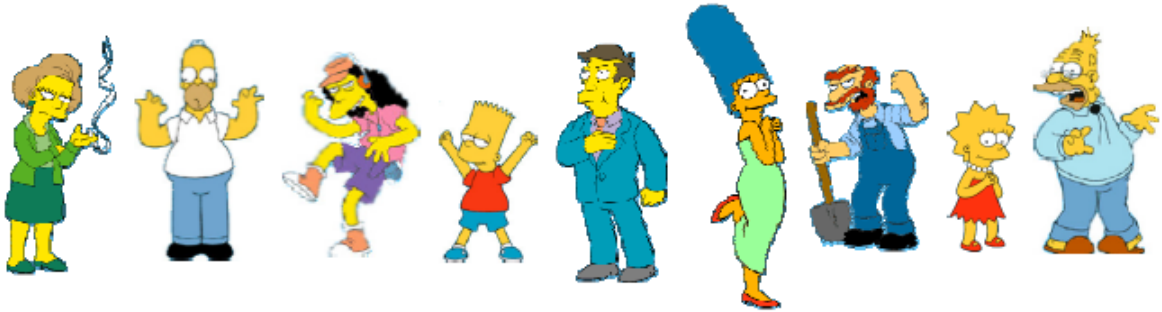
Introduction: Clustering?

What is a natural grouping among these objects?



Introduction: Clustering?

What is a natural grouping among these objects?



Clustering is subjective

Simpson's Family

School Employees

Females

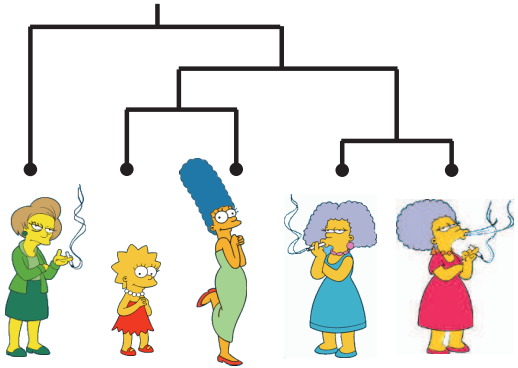
Males

Two Types of Clustering

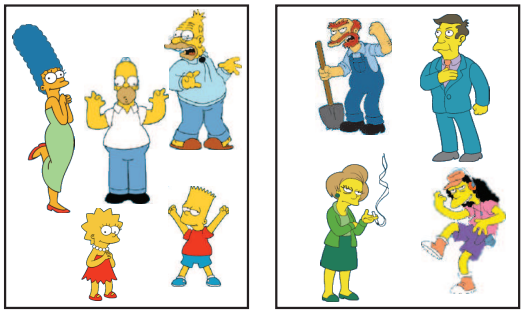
• **Partitional algorithms:** Construct various partitions and then evaluate them by some criterion

• **Hierarchical algorithms:** Create a hierarchical decomposition of the set of objects using some criterion

Hierarchical



Partitional

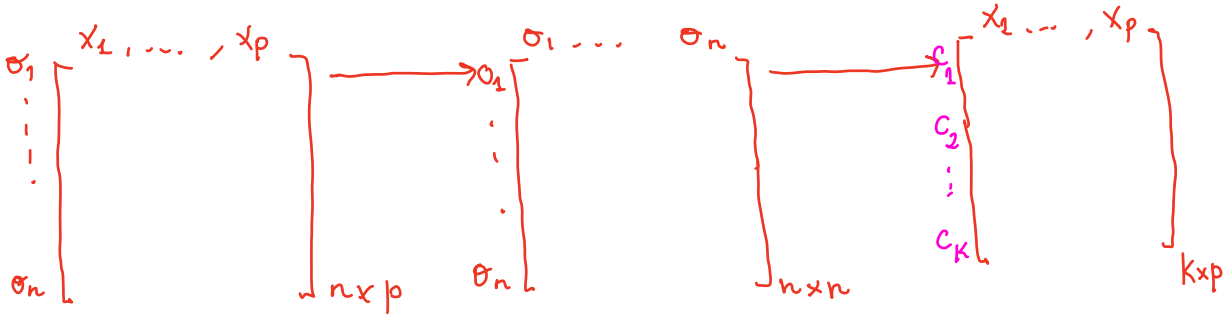


Introduction: Data Structures

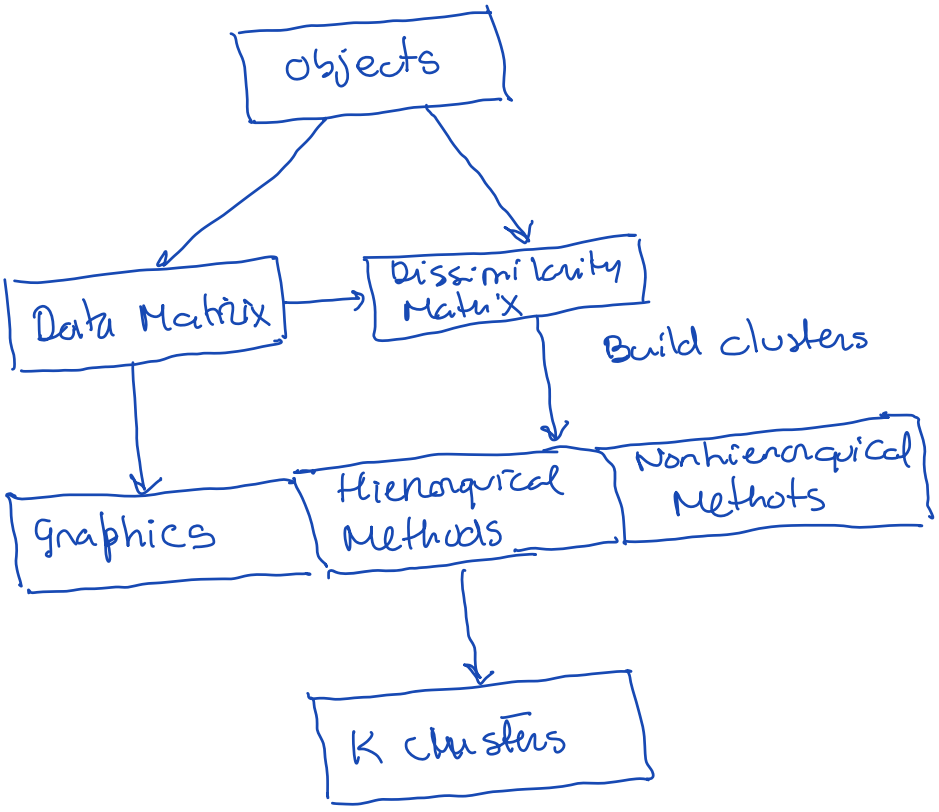
Cluster analysis operates on two kinds of data structure:

- Data Matrix (or design/profile matrix) - structure already used in previous methods: $\mathbf{X} = [x_{ij}], \quad i = 1, 2, \ldots, n; \quad j = 1, 2, \ldots, p$, where x_{ij} is the value of variable j for object i.
  This matrix may include:
  - Quantitative variables (continuous or discrete)
  - Qualitative variables (nominal or ordinal)
- Dissimilarity (or similarity) matrix - structure already mentioned previously: $\mathbf{D} = [d_{ij}] \quad i, j = 1, 2, \ldots, n$ is a square, in general symmetrical matrix, where d_{ij} element equal to the value of a chosen measure of distinction between the $i-$th and the $j-$th object. This matrix can be calculated from the data or by direct observation.

Given observations of p-variables in n objects $\tilde{x}_1, \ldots, \tilde{x}_n$, cluster Analysis transform this information to the information of K groups with $K < n$, where K is usually unknow



Phases of cluster Analysis

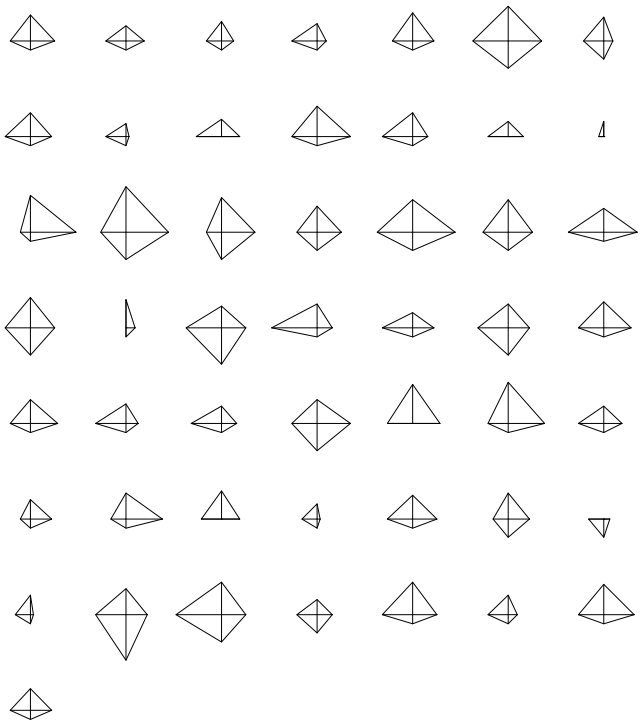


Introduction: Steps in cluster analysis

1. Object selection
2. Variable selection
3. Variable transformation
4. Create a matrix of relative dissimilarities/similarities between all objects
5. Decision on the method of combining objects into groups (graphical; hierarchic; partition; other)
6. Discussion and presentation of results (number of clusters validation/description/interpretation;)

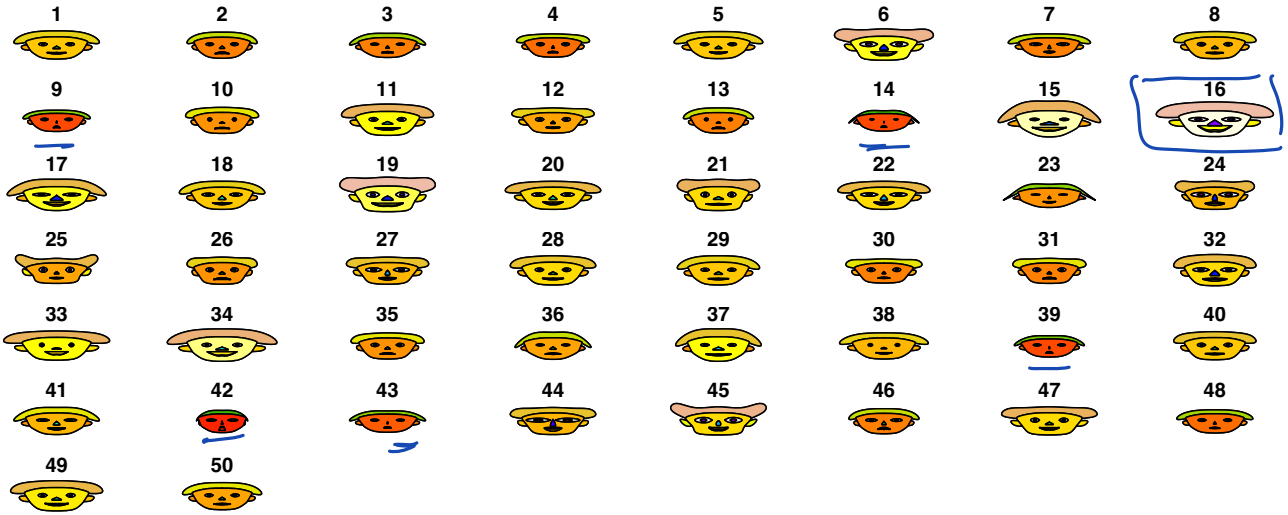
```
library(graphics)
stars(setosa)
```

Stars



Chernoff Faces

```
library(aplpack)
faces(setosa)
```



Hierarchical Methods

There are two major types of hierarchical techniques: divisive and agglomerative

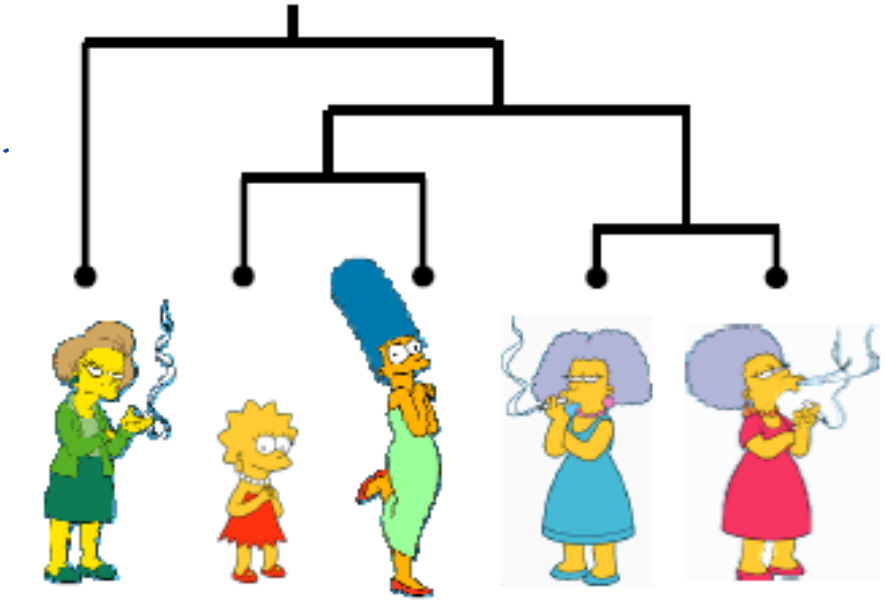
Agglomerative hierarchical techniques are the more commonly used

Agglomerative: This is a "bottom up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.

Divisive: This is a "top down" approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy

The results of hierarchical clustering are usually presented in a two-dimensional diagram known ad dendrogram.

Hierarchical Methods: Dendrogram

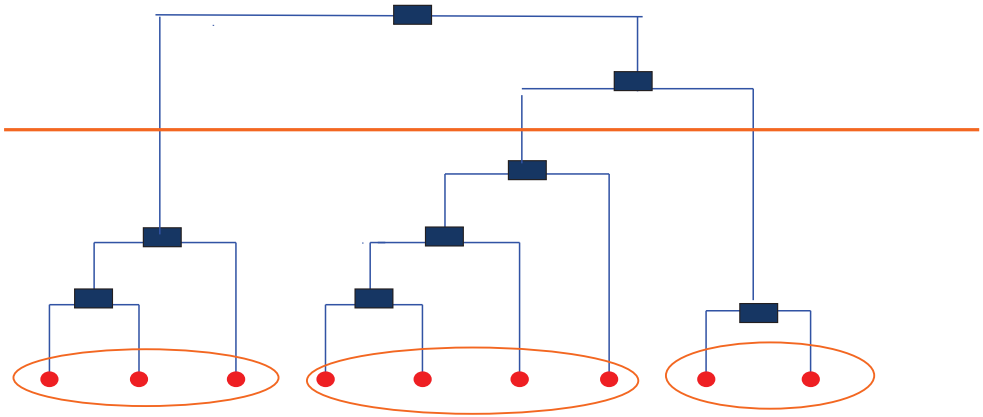


Hierarchical Methods

1. A dendrogram provides a highly interpretable complete description of the hierarchical clustering in a graphical format. This is one of the main reasons for the popularity of hierarchical clustering methods.
2. Cutting the dendrogram horizontally at a particular height we obtain a partition of the data into clusters
3. A dendrogram is often viewed as a graphical summary of the data rather than a description of the results of the algorithm
4. Different hierarchical methods, as well as small changes in the data, can lead to quite different dendrograms

Dendrogram

A clustering of the data objects is obtained by cutting the *dendrogram* at the desired level, then each connected component forms a cluster



Hierarchical Methods

Agglomerative approach

. . . start at the bottom and at each level recursively merge a selected pair of clusters into a single cluster.
This produces a grouping at the next higher level with one less cluster.
The pair chosen for merging consist of the two groups with the smallest intergroup dissimilarity

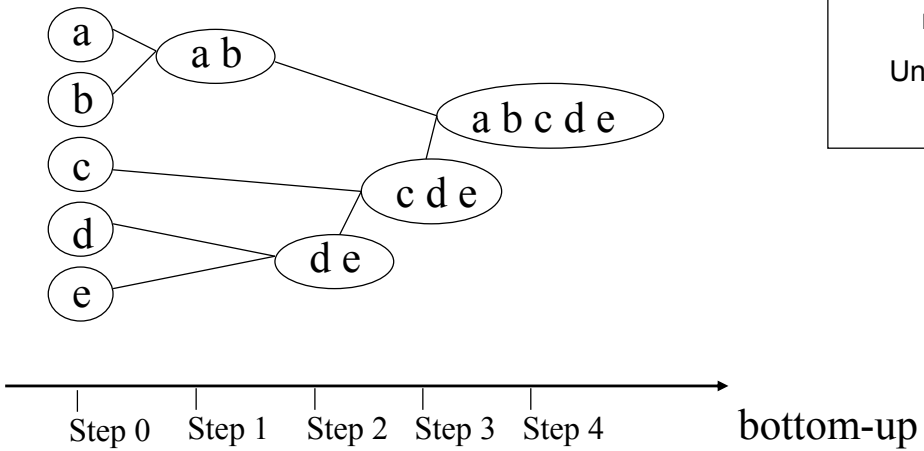
Divisive approach

. . . start at the top and at each level recursively split one of the existing clusters at that level into two new clusters.
The split is chosen to produce two new groups with the largest between-group dissimilarity.

In both approaches there are $n - 1$ levels of hierarchy

Hierarchical Clustering

Agglomerative approach



Initialization:

Each object is a cluster

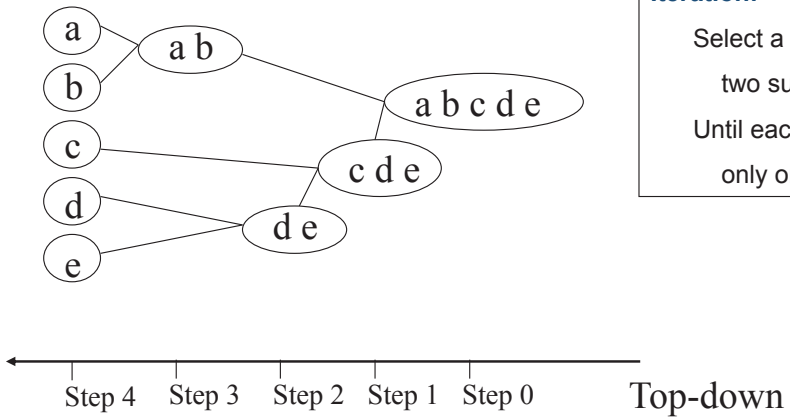
Iteration:

Merge two clusters which are most similar to each other;

Until all objects are merged into a single cluster

Hierarchical Clustering

Divisive Approaches



Initialization:

All objects stay in one cluster

Iteration:

Select a cluster and split it into two sub clusters

Until each leaf cluster contains only one object

Hierarchical Methods

- In the general case, the complexity of agglomerative clustering is $O(n^3)$, which makes them too slow for large data sets.
- Divisive clustering with an exhaustive search is $O(2^n)$, which is even worse.
- However, for some special cases, optimal efficient agglomerative methods (of complexity $O(n^2)$) are known: SLINK for single-linkage and CLINK for complete-linkage clustering

The procedure described can lead to various methods of grouping, each differing in the use of each of the following concepts:

- similarity/dissimilarity between two objects
- similarity/dissimilarity between two groups, also called linkage (or fusion)

Proximity measures: What is Similarity?

- The quality or state of being similar; likeness;...
- Similarity is hard to define, but "We know it when we see it"

Proximity measures: What is Similarity?

The real meaning of similarity is a philosophical question. We will take a more pragmatic approach

Proximity measures: Similarity/Dissimilarity between objects

- the degree of similarity measures the degree of similarity or proximity between the objects
- more similar objects $\Longrightarrow$ larger similarity
- more distinct objects $\Longrightarrow$ larger dissimilarity

Dissimilarities and distances

- dissimilarities d_{ij} between the objects i and j are measures that allows to translate quantitatively the larger or smaller differences between the objects in the set of p variables

Proximity measures: Similarity/Dissimilarity between objects

Given two objects i and j, d_{ij} is a dissimilarity measure if have the following proprieties:

d_{ij} = number of person that commute from city i to j

1. $d_{ij} \geq 0, \forall i, j = 1, 2, \ldots n$
2. $d_{ii} = 0, \forall i = 1, 2, \ldots n$
3. $d_{ij} = d_{ji}, \forall i, j = 1, 2, \ldots n$ (Symmetric)

Notes:

- Almost always requires the positivity (properties 1 and 2)
- The property of symmetry (3) sometimes is not verified, although the measure continues to be useful for defining the dissimilarity. For example, the case of dissimilarity between two cities i and j can be measured by the number of people who travel from i to j.
  - The symmetric property can be re-established if we consider $d^*_{ij} = \frac{d_{ij}+d_{ji}}{2}$

Proximity measures: Similarity/Dissimilarity between objects

If in addition, also satisfy the triangular inequality:

- $d_{ij} \leq d_{ik} + d_{kj}, \forall i, j, k = 1, 2, \ldots n$

the dissimilarity is a metric or a distance

Many dissimilarities did not satisfy the previous property. However, some dissimilarity satisfy another property, too strong, that is ultrametric, i.e.

- $d_{ij} \leq \max(d_{ik}, d_{jk}), \forall i, j, k = 1, 2, \ldots n$

$s_i \rightarrow$ similarity ; $s_{ij} \geq 0$ some times!!

$s_{ij} = s_{ji}$

Proximity measures: Similarity/Dissimilarity between objects

The dissimilarity measures depends on the characteristics (variables type) that we are observing in the objects.

Quantitative variables:

- Euclidean distance and its derivatives (weighted Euclidean or generalized, (eg. Mahalanobis, when the weighting matrix is the covariance matrix);
- Minkowski metrics (Manhattan distance);
- Canberra metric;
- Correlation coefficient (it is a similarity should be transformed into dissimilarity)

Some distances | similarities

A- Quantitative variables

1. Euclidean Distance

$$d_{ij} = \sqrt{\sum_{k=1}^{p} (x_{ik} - x_{jk})^2} = \sqrt{(\underset{\sim}{x}_i - \underset{\sim}{x}_j)^T (\underset{\sim}{x}_i - \underset{\sim}{x}_j)} \quad \text{(scale sensible)}$$

1' Mean Euclidean Distance

$$d_{ij} = \sqrt{\frac{\sum_{i=1}^{p} (x_{ik} - x_{jk})^2}{p}}$$

1'' Standardized Euclidean Distance

$$z_{ik} = \frac{x_{ik} - \bar{x}}{s_k}$$

$$d_{ij} = \sqrt{\sum_{i=1}^{p} (z_{ik} - z_{jk})^2}$$

2. Mahalanobis distance

$$d_{ij} = \sqrt{(\underset{\sim}{x}_i - \underset{\sim}{x}_j)^T \underset{\sim}{S}^{-1} (\underset{\sim}{x}_i - \underset{\sim}{x}_j)}$$

3. Manhattan distance

$$d_{ij} = \sum_{k=1}^{p} |x_{ik} - x_{jk}|$$

4. Canberra distance (non negative variables)

$$d_{ij} = \sum_{k=1}^{p} \frac{|x_{ik} - x_{jk}|}{(x_{ik} + x_{jk})} \quad \text{and} \quad d_{ij} = 0 \text{ if } x_{ik} = x_{jk} = 0$$

5. Coefficient of correlation: (similarity)

$$r_{ij} = \frac{\sum_{k=1}^{n} (x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j)}{\sqrt{\sum_{k=1}^{n} (x_{ki} - \bar{x}_i)^2 \sum_{k=1}^{n} (x_{kj} - \bar{x}_j)^2}},$$

can be transformed in a dissimilarity by:

$d_{ij} = \frac{1 - r_{ij}}{2}$, since $-1 \leq r_{ij} \leq 1$

Proximity measures: Similarity/Dissimilarity between objects

Qualitative variables: (typically measures of similarity)

- Coefficient of concordance
- Jaccard coefficient
- Gower and Legendre coefficient
- and many others ...

For mixed variables:

- Romesburg strategy - ignore the type of variables and consider them all quantitative type, encoding the qualitative;
- Perform separate analyses;
- Reduce all the variables to binary variables;
- Building coefficient of similarity combined (for example, Gower)

B- Qualitative variables

B_1 - Binary variables = $\{0, 1\}$

	variables x_1				x_p
Obj i	1	0	1	0 ...	1
Obj j	0	1	1	0 ...	0

table : obj j

obj i	1	0	
1	a	b	a+b
0	c	d	c+d
	a+c	b+d	a+b+c+d = p

(1,1) = a # (0,1) = c

(1,0) = b # (0,0) = d

1. Euclidean distance

$$d_{ij} = \sqrt{\sum_{k=1}^{p} (x_{ik} - x_{jk})^2} = (b+c)^{1/2}$$

2. Coefficient of concordance or (similarities):
Simple matching coefficient

$$s_{ij} = \frac{a+d}{a+b+c+d}$$ proportion of objects with matching

$0 < s_{ij} < 1$

3. Jacard coefficient

$$s_{ij} = \frac{a}{a+b+c}$$, the number of (0,0) is irrelevant

4. Gower and Legendre coefficient

$$s_{ij} = \frac{(a+d)-(b+c)}{a+b+c+d}$$

, proportion of difference between matching and not matching

$-1 < s_{ij} < 1$

B_2 - variables with more than 2 categories

X_1 hair colour = { black – B_1, Brown – B_2, Blond – B_3, Red – R }

X_2 eye colour = { blue – B_1, Brown – B_2, Green – G }

	Hair colour				eye colour		
	B_1	B_2	B_3	R	B_1	B_2	G
Obj i (O_i)	0	1	0	0	0	1	0
Obj j (O_j)	0	0	1	0	0	1	0

		o_i		
		1	0	
o_j	1	1	1	2
	0	1	4	5
		2	5	7

2. Simple matching $= \frac{a+d}{a+b+c+d} = \frac{5}{7}$

3. Jaccard $= \frac{a}{a+b+c} = \frac{1}{3}$

4. Gower Legendre $= \frac{(a+d)-(b+c)}{a+b+c+d} = \frac{5-2}{7} =$

$= 3/7$

Proximity measures: Similarity/Dissimilarity between objects

If they used similarities measures sometimes is possible to convert these similarities in dissimilarities, for example:

- $d_{ij} = 1 - sij$
- $d_{ij} = 1 - s^2 ij$
- $d_{ij} = \sqrt{1 - sij}$
- $d_{ij} = \sqrt{1 - s^2 ij}$

Ex. List

3. Consider the following species:

T D W H M

Tiger, Dog, Whale, Hare, Man,

and the following attributes:

$x_i \in \{0, 1\}$

- eats other animals, = X_1
- eat vegetables, = X_2
- moves on four legs, = X_3
- is a domestic animal, = X_4
- is a wild animal. = X_5

Obtain the similarity matrix based on the Jaccard coefficient.

Home work : Simple maching coefficient

	X_1	X_2	X_3	X_4	X_5
T	1	0	1	0	1
D	1	0	1	1	0
W	1	0	0	0	1
H	0	1	1	0	1
M	1	1	0	0	0

$\frac{a}{a+b+c} = \frac{2}{4}$

T \ D	1	0
1	2	1
0	1	1

T \ W	1	0
1	2	1
0	0	2

	1	0
1	a	b
0	c	d

T \ H	1	0
1	2	1
0	1	1

T \ M	1	0
1	1	2
0	1	1

D \ W	1	0
1	1	2
0	1	1

D \ H	1	0
1	1	2
0	2	0

D \ M	1	0
1	1	2
0	1	1

W \ H	1	0
1	1	1
0	2	1

W \ M	1	0
1	1	1
0	1	2

H \ M	1	0
1	1	2
0	1	1

$$Jaccard = \frac{a}{a+b+c} =$$

	T	D	W	H	M
T	1	2/4	2/3	2/4	1/4
D		1	1/4	1/5	1/4
W			1	1/4	1/3
H				1	1/4
M					1

6.3- Clustering Methods

- visual Methods

(1) • hierarchical { Agglomeratives (A) / divisives (B) }

(2) • non-hierarchical

(1) Hierarchical clustering methods

1. Use the dissimilarity matrices or Data matrices
2. Can be used to cluster objects or variables
3. After an object / variables enters in a cluster it will not be removed from the cluster.
4. The number of clusters does not have to be known in advance

A. Agglomerative methods (most common) start with n clusters and joint clusters until there is only one cluster with all the objects

B. Divisive methods: start with one cluster (with all obj.) and split the cluster to obtain n clusters

the results are display in the form of a two-dimensional diagram called Dendrogram

(A) Agglomerative methods:

Algorithm:

step 1: Given n objects, obtain the dissimilarity matrix $\underset{\sim}{D}$

Step 2 : choose the smallest value of $\tilde{D}$. Let (A, B) be such that $d_{AB} = \min d_{ij}$ $i \neq j$

threshold distance

Step 3 : Merge A and B at distance d_{AB}. Update $\tilde{D}$ obtaining the $d_{(A \cup B) i}$ for all the remaining clusters i

Step 4 : Repeat 2 and 3 $(n-1)$ times until we have only one cluster.

the problem is to define what is a distance between clusters

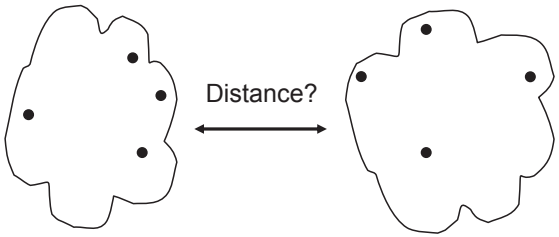
Similarity/Dissimilarity between cluster/groups

Defining Closeness of Clusters

- The key in a hierarchical clustering algorithm is specifying how to determine the two "closest" clusters at any given step
- For the first step, it's easy: Join the two objects whose distance is smallest
- After that, we have a choice: Do we join two individual objects together, or merge an object into a cluster that already has multiple objects?

How to Merge Clusters?

How to measure the distance between clusters?

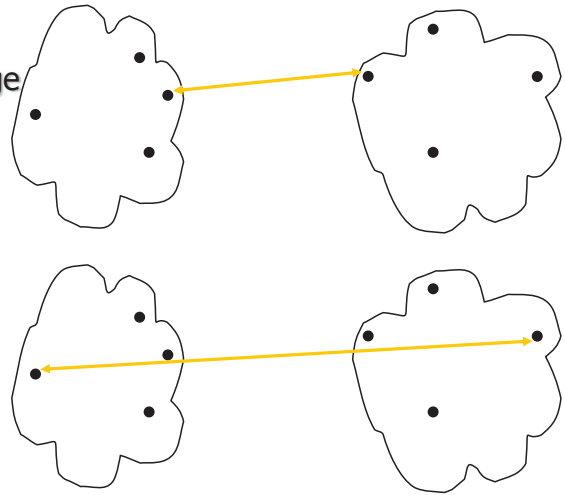


Hint: *Distance between clusters* is usually defined on the basis of *distance between objects.*

How to Merge Clusters?

How to measure the distance between clusters?

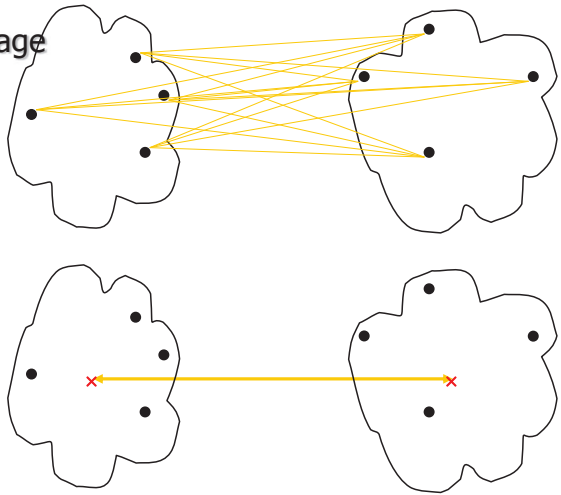
- MIN
  single-linkage
- MAX
  complete
  -linkage



How to Merge Clusters?

How to measure the distance between clusters?

- average-linkage
- Centroides



Similarity/Dissimilarity between cluster/groups

Let A and B represent two such groups

- The single linkage (also called nearest neighbour) , at each step, joins the clusters whose minimum distance between objects is smallest, i.e., joins the clusters A and B with the smallest

$$D_{AB} = \min \{d_{ij} : i \in A, j \in B\}$$

Similarity/Dissimilarity between cluster/groups

Let A and B represent two such groups

- Complete linkage (also called farthest neighbour), at each step, joins the clusters whose maximum distance between objects is smallest, i.e., joins the clusters A and B with the smallest

$$D_{AB} = \max \{d_{ij} : i \in A, j \in B\}$$

Similarity/Dissimilarity between cluster/groups

- Average linkage (also called Group average) Here the distance between two clusters is defined as the average distance between all possible pairs of objects with one object in each pair belonging to a distinct cluster, i.e.

$$D_{AB} = \frac{1}{n_A \times n_B} \sum_{i=l}^{n_A} \sum_{j=1}^{n_B} d_{ij}$$

- Centroid method. The distance between two groups A and B is the distance between group centres or other points considered groups "representatives" (centroid), i.e.:

$$D_{AB} = d(\bar{\mathbf{x}}_A, \bar{\mathbf{x}}_B),$$

where $\bar{\mathbf{x}}_A = \frac{\sum_{i \in A} \mathbf{x}_i}{n_A}$ e $\bar{\mathbf{x}}_B = \frac{\sum_{i \in B} \mathbf{x}_i}{n_B}$, where $\mathbf{x}_i$ is the vector of de p observations for object i.

Similarity/Dissimilarity between cluster/groups

- Ward method (also called minimum variance method). Ward's minimum variance criterion minimizes the total within-cluster variance. At each step the pair of clusters with minimum between-cluster distance are merged. This method uses as a criterion for merger two groups A and B the increased sum of squares that occurs when the groups A and B are merged together in a group $C = A \cup B$

$$SSW_C - (SSWA + SSW_B)$$

where

$$SSW_A = \sum_{i \in A} \sum_{j=1}^{p} (x_{ijA} - \bar{x}_{jA})^2$$

is the A group sum of squares. Similar expressions for the sum of squares of B and C.

Consider the following distance matrix

$\tilde{D}^{(1)}$

	a	b	c	d	e
a	0				
b	2	0			
c	6	5	0		
d	10	9	4	0	
e	9	8	5	3	0

Each individual is in it's own cluster!

$D_{AB} = \min \{d_{ij} : i \in A, j \in B\}$ Single-Linkage

smallest distance = threshold distance (h_1) = 2

$c_1 = \{a, b\}$

$D^{(2)}$

	(ab)	c	d	e
(ab)	0			
c	5	0		
d	9	4	0	
e	8	5	3	0

$d_{(ab),c} = \min \{d_{ac}, d_{bc}\} = 5$

$d_{(ab),d} = 9$

$d_{(ab),e}$

threshold distance = $h_2 = 3$

$D^{(3)}$

	(ab)	c	(de)
(ab)	0		
c	5	0	
(de)	8	4	0

$d_{(ab),(de)}$

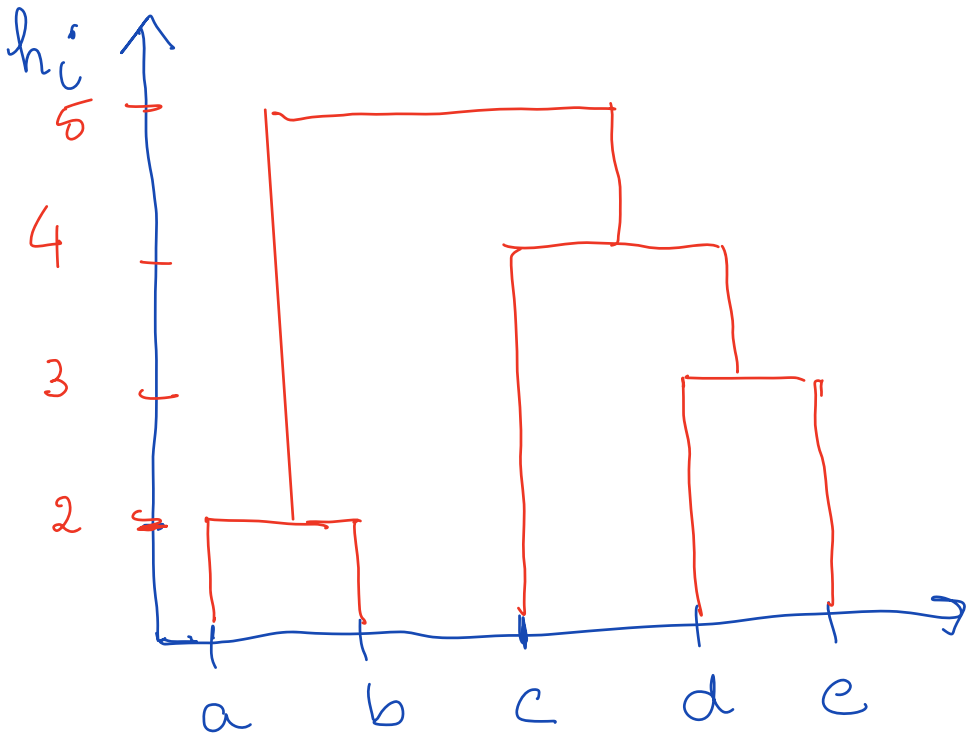
threshold distance = $h_3 = 4$

(c, de)

$D^{(4)}$

	(ab)	(cde)
(ab)	0	
(cde)	5	0

Dendrogram



Average Linkage

i.e.

$$D_{AB} = \frac{1}{n_A \times n_B} \sum_{i=1}^{n_A} \sum_{j=1}^{n_B} d_{ij}$$

$D^{(1)}$

	a	b	c	d	e
a	0				
b	2	0			
c	6	5	0		
d	10	9	4	0	
e	9	8	5	3	0

Each individual is in it's own cluster!

$h_1 = 2$

$d_{(ab),c} = \frac{d_{ac} + d_{bc}}{2 \times 1} = \frac{6+5}{2} = 5.5$

$D^{(2)}$

	(ab)	c	d	e
(ab)	0			
c	5.5	0		
d	9.5	4	0	
e	8.5	5	3	0

$D^{(4)}$

	(ab)	(cde)
(ab)	0	
(cde)	7.8(3)	0

Nearest neighbour clustering

Decision to merge groups is based on the distance of the nearest member of the group to the nearest other object.

In our example, with a distance of 2, individuals a and b are the most similar.

	a	b	c	d	e
a	0				
b	2	0			
c	6	5	0		
d	10	9	4	0	
e	9	8	5	3	0

We therefore merge these into a cluster at level 2:

Distance	Groups
0	a b c d e
2	(ab) c d e

Next step:

and we now need to re-write our distance matrix, whereby:

$$d_{(ab)c)} = min(d_{ac}, d_{bc}) = d_{bc} = 5$$
$$d_{(ab)d)} = min(d_{ad}, d_{bd}) = d_{bc} = 9$$
$$d_{(ab)e)} = min(d_{ae}, d_{be}) = d_{bc} = 8$$

This gives us a new distance matrix

	(ab)	c	d	e
(ab)	0			
c	5	0		
d	9	4	0	
e	8	5	3	0

What do we merge next?

Next step:

Distance	Groups
0	$a\ b\ c\ d\ e$
2	$(ab)\ c\ d\ e$
3	$(ab)\ c\ (de)$

So, find the minimum distance from d and e to the other objects and reform the distance matrix:

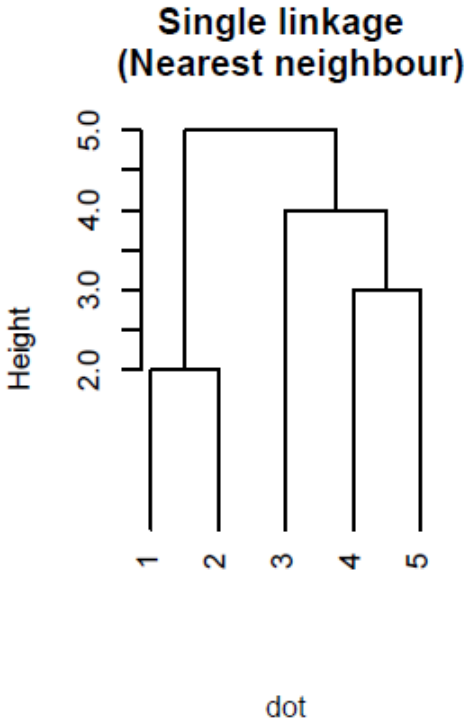
	(ab)	c	(de)
(ab)	0		
c	5	0	
(de)	8	4	0

And so on . . .

Clearly, the next merger is between (de) and c, at a height of 4, the final merger will take place at a height of 5.

Distance	Groups
0	$a\ b\ c\ d\ e$
2	$(ab)\ c\ d\ e$
3	$(ab)\ c\ (de)$
4	$(ab)\ (cde)$
5	$(abcde)$

We can plot this information



Furthest neighbour / Complete linkage

Objects are merged when the furthest member of the group is close enough to the new object

	a	b	c	d	e
a	0				
b	2	0			
c	6	5	0		
d	10	9	4	0	
e	9	8	5	3	0

Starts as before, merge *a* and *b* as these are the nearest:

Distance	Groups
0	a b c d e
2	(ab) c d e

Furthest neighbour / Complete linkage

Life changes now when we calculate the new distance matrix:

$$d_{(ab)c)} = max(d_{ac}, d_{bc}) = d_{bc} = 6$$
$$d_{(ab)d)} = max(d_{ad}, d_{bd}) = d_{bc} = 10$$
$$d_{(ab)e)} = max(d_{ae}, d_{be}) = d_{bc} = 9$$

	(ab)	c	d	e
(ab)	0			
c	6	0		
d	10	4	0	
e	9	5	3	0

So what do we merge next?

Furthest neighbour / Complete linkage

Actually we still merge d and e, but note the height!

Distance	Groups
0	a b c d e
2	(ab) c d e
3	(ab) c (de)

And reforming the new distance matrix:

	(ab)	c	(de)
(ab)	0		
c	6	0	
(de)	10	5	0

Compare the next merge with the same step before, but compare the heights (noting this is a very artificial example)

Furthest neighbour / Complete linkage

Completing the clustering

Distance	Groups
0	a b c d e
2	(ab) c d e
3	(ab) c (de)
5	(ab) (cde)

and the final distance matrix:

	(ab)	(cde)
(ab)	0	
(cde)	10	0

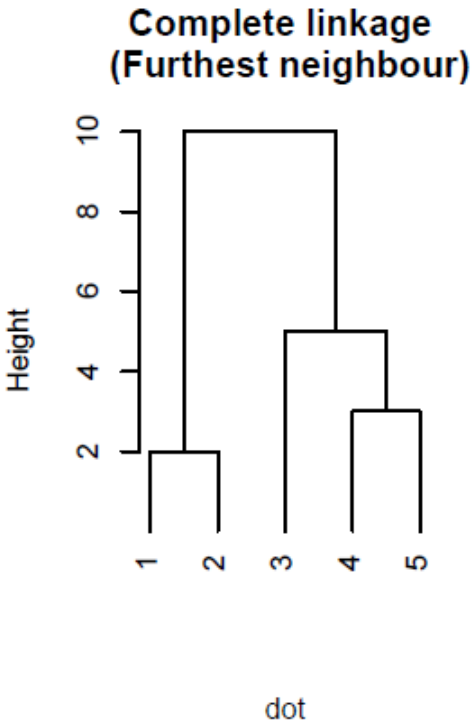
Furthest neighbour / Complete linkage

Final merge at height 10

Distance	Groups
0	a b c d e
2	(ab) c d e
3	(ab) c (de)
5	(ab) (cde)
10	(abcde)

This is a very artificial example. Merges happen in the same order, but at different heights. In more realistic examples you would expect to see some different mergers taking place

We can plot this information



Group average link

Merge two groups is the average distance between them is small enough

Again, we start by merging a and b, but again the reduced distance matrix will be different:

$$d_{(ab)c)} = (d_{ac} + d_{bc})/2 = d_{bc} = 5.5$$
$$d_{(ab)d)} = (d_{ad} + d_{bd})/2 = d_{bc} = 9.5$$
$$d_{(ab)e)} = (d_{ae} + d_{be})/2 = d_{bc} = 8.5$$

	(ab)	c	d	e
(ab)	0			
c	5.5	0		
d	9.5	4	0	
e	8.5	5	3	0

$$d_{(ab),d} = \frac{d_{ad} + d_{bd}}{2} = \frac{10+9}{2} = 9.5$$

$$d_{(ab),e} = \frac{d_{ae} + d_{be}}{2} = \frac{9+8}{2} = 8.5$$

$h_2 = 3$

$D^{(3)}$

	(ab)	c	(de)
(ab)	0		
c	5.5	0	
(de)	9	**4.5**	0

$$d_{(\underbrace{ab}_{2}),(\underbrace{de}_{2})} = \frac{d_{ad} + d_{ae} + d_{bd} + d_{be}}{4}$$

$$= \frac{10+9+9+8}{4} = 9$$

Group average link

Next merge (same order, different height)

Merge d and e, at height 3:

Distance	Groups
0	a b c d e
2	(ab) c d e
3	(ab) c (de)

Again, need to recalculate distances:

	(ab)	c	(de)
(ab)	0		
c	5.5	0	
(de)	9	4.5	0

Group average link

and leaping on a bit

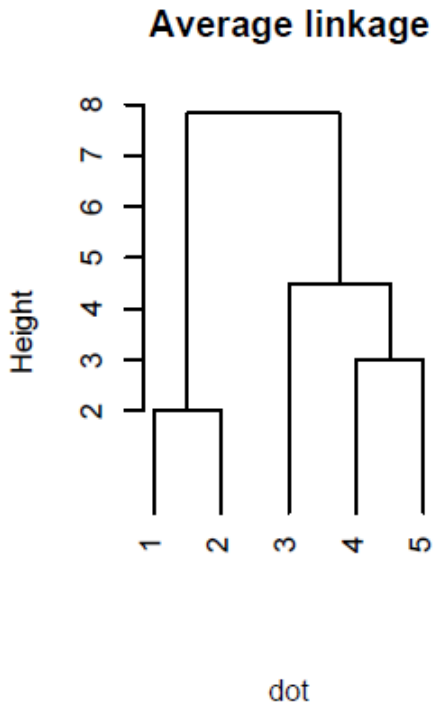
after merging (de) and c:

	(ab)	(cde)
(ab)	0	
(cde)	7.8	0

our final merge will take place at height 7.8.

Distance	Groups
0	a b c d e
2	(ab) c d e
3	(ab) c (de)
4.5	(ab) (cde)
7.8	(abcde)

We can plot this information



We can plot this information

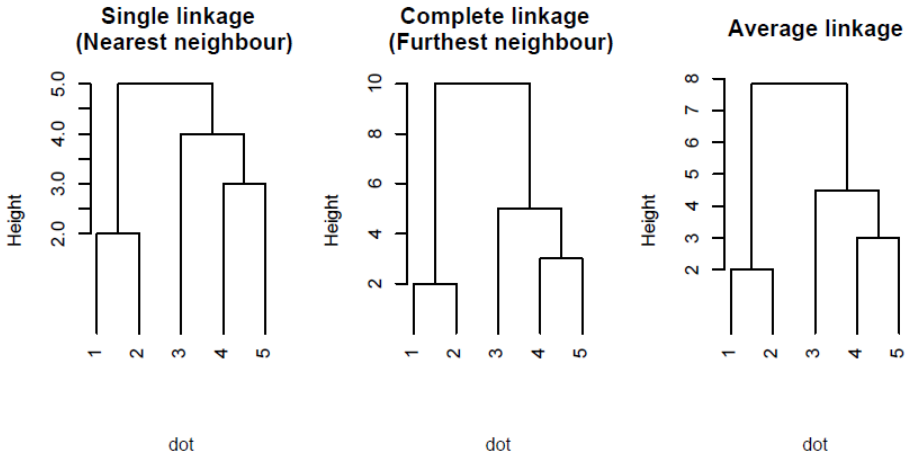


Figure: Dendrograms from three basic cluster methods

Ward Method : Method of Incremental sum of squares

Let $AB = A \cup B$ then :

$$SSW(A) = \sum_{i=1}^{n_A} \sum_{j=1}^{p} (x_{ijA} - \bar{x}_{jA})^2$$ sum of squares

deviation of every item in cluster A to the cluster mean (centroid)

$$SSW(B) = \sum_{i=1}^{n_B} \sum_{j=1}^{p} (x_{ijB} - \bar{x}_{jA})^2$$

$$SSW(AB) = \sum_{i=1}^{n_A+n_B} \sum_{j=1}^{p} (x_{ijAB} - \bar{x}_{jAB})^2$$

thus, $d_{AB} = SSW(AB) - SSW(A) - SSW(B)$

Ex: Exercise 5

$$\mathbf{A} = \begin{bmatrix} 1.0 & 7.0 \\ 1.5 & 7.5 \\ 2.0 & 8.5 \\ 3.0 & 7.0 \\ 4.0 & 4.0 \\ 2.0 & 1.0 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix}$$

x_1 x_2

$$\underset{\sim}{D} = \begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & & & & & \\ 2 & 0.71 & 0 & & & & \\ 3 & 1.8 & 1.12 & 0 & & & \\ 4 & 2.0 & 1.58 & 1.8 & 0 & & \\ 5 & 4.24 & 4.3 & 4.92 & 3.16 & 0 & \\ 6 & 6.08 & 6.52 & 7.5 & 6.08 & 3.61 & 0 \end{array}$$

$h_1 = 0.71 \Rightarrow$

$C_1 = \{1, 2\}$

$d_{e_{1,3}} = ?$

$A = C_1$ centroid: $\bar{x}_A = \left(\frac{1+1.5}{2}, \frac{7.0+7.5}{2}\right) = (1.25; 7.25)$

$B = \{3\}$ centroid $B = (2.0; 8.5) = \bar{x}_B$ ($\bar{x}_1$ $\bar{x}_2$)

$$SSW(A) = \sum_{i \in A} \sum_{j=1}^{2} (x_{ij\,A} - \bar{x}_{j\,A})^2 = (1.0-1.25)^2 + (1.5-1.25)^2 + (7.0-7.25)^2 + (7.5-7.25)^2 = 0.25$$

$SSW(B) = 0$

$AB = (A \cup B) = C_1 \cup \{3\} = \{1, 2, 3\}$

Centroid $AB = \bar{x}_{AB} = \left(\frac{1+1.5+2.0}{3}; \frac{7.0+7.5+8.5}{3}\right) = (1.5; 7.(6))$

$$SSW(AB) = \sum_{i \in AB} \sum_{j=1}^{2} (x_{ij\,AB} - \bar{x}_{j\,AB})^2 =$$

$$= (1-1.5)^2 + (1.5-1.5)^2 + (2.0-1.5)^2 + (7.0-7.(6))^2 + (7.5-7.(6))^2 + (8.5-7.(6))^2$$

$$= 1.(6)$$

$d_{AB} = SSW(AB) - SSW(A) - SSW(B) = 1.(6) - 0.25 - 0 = 1.417$ ✓

$d_{C_1,4} = ?$ $\quad SSW(A) = 0.25$

$C_1 = A$ $\quad SSW(B) = 0$

$\{4\} = B$

Centroid $AUB = \bar{x}_{AB} = \left(\frac{1.0+1.5+3.0}{3}, \frac{7.0+7.5+7.0}{3}\right) =$

$= (1.8(3); 7.1(6))$

$$SSW(AB) = (1-1.8(3))^2 + (1.5-1.8(3))^2 + (3.0-1.8(3))^2 + (7.0-7.1(6))^2 \times 2 + (7.5-7.1(6))^2 = 2.333$$

$d_{AB} = 2.333 - 0.25 = 2.083$

6.3.3 - Non-hierarchical cluster

1. Designed to group only objects
2. use data matrices only
3. Based on the optimization of an objective function. this is a measure of the internal cohesion and external isolation.
4. Number of clusters need to be known in advanced.
5. During the analysis, the same object can belong to different clusters.
6. IT is impossible to analysed all possible partitions.

General procedure :

1. choose an initial partitions of the data in k-clusters.
2. consider every movement from each object to another cluster Save the change in the objective function.
3. Do the changes leading to the highest improvement in the objective function.
4. Repeat 2. and 3. until you cannot change any object without decreasing the objective function.

these methods demands an initial partition

Initial Partitions:

1. the result of another clustering method
2. Based on previous knowledge about the data
3. Randomly

After knowing this initial partition, K points have to be chosen as representants of each cluster

K- means method :

the idea is to Assign an object to the cluster with the nearest centroid.

Algorithm :

1. Partition the objects into K initial clusters.
2. Proceed through the objects list, assigning an object to the cluster whose centroid (mean) is nearest. (Distance is usually computed using the Euclidean distance or the square of the Euclidean distance).
   Recalculate the centroid for the clusters receiving the new objects and losing objects.
3. Repeat step 2. until no more reassignments take place

the basic idea behind K-means clustering consists of defining clusters so that the total intra-cluster variation (know as total within-cluster variation) is minimized. the standard algorithm is the Hartigan-Wong algorithm (1979), which defines the total within-cluster variation as the sum of squared Euclidean distances between items and corresponding centroid:

$$SSW(C_K) = \sum_{\underset{\sim}{x}_i \in C_K} (\underset{\sim}{x}_i - \bar{\underset{\sim}{x}}_K)^2$$

where $\underset{\sim}{x}_i$ is a data point belonging to C_K

$\bar{\underset{\sim}{x}}_K$ is the mean of C_K

Each observation $\underset{\sim}{x}_i$ is assigned to a given cluster such that the sum of squares (SS) distance of the observation to their assigned cluster center ($\bar{\underset{\sim}{x}}_K$) is minimized.

the total within-cluster variation is defined as:

$$\text{tot. withiness} = \sum_{j=1}^{K} SSW(C_j) = \sum_{j=1}^{K} \sum_{\underset{\sim}{x}_i \in C_j} (\underset{\sim}{x}_i - \bar{\underset{\sim}{x}}_j)^2$$

SSW

the total within-cluster sum of square measure the compactness (i.e. goodness) of the clustering and we want it to be as small as possible

Example:

Let us consider that 6 wines were classified by its fragancy (x_1) and flavour (x_2) leading to: (scale 0-10)

	x_1 Fragancy	x_2 Flavour	
DS 94	4	6	C_1
DS 92	5	7	C_2
DS 90	5	8	C_2
DS 70	2	4	C_1
Bairrada 87	3	4	C_1
Bairrada 80	6	8	C_2

Considering the initial partition $K=2$ be:

$C_1 = \{94, 92, 90, 70\}$ DS

$C_2 = \{87, 80\}$ Bairrada

and the squared Euclidean distance between the objects, obtain the final clustering with the K-means method.

$C_1^{(1)} = \{94, 92, 90, 70\}$

$C_2^{(1)} = \{87, 80\}$

centroids:

$\underset{\sim}{\bar{x}}_1 = \left(\frac{4+5+5+2}{4}; \frac{6+7+8+4}{4}\right) = (4; 6.25)$ ($\bar{x}_{11}$, $\bar{x}_{12}$)

$\underset{\sim}{\bar{x}}_2 = \left(\frac{3+6}{2}; \frac{4+8}{2}\right) = (4.5; 6)$ ($\bar{x}_{21}$, $\bar{x}_{22}$)

$d^2_{94, C_1} = (4-4)^2 + (6-6.25)^2 = 0.0625$

$d^2_{94, C_2} = (4-4.5)^2 + (6-6)^2 = 0.25$

⋮

	$\bar{x}_i$	94	92	90	87	80	70
$c_1^{(1)}$	(4; 6.25)	0.0625	1.5625	4.0625	6.0625	7.0625	9.0625
$c_2^{(1)}$	(4.5; 6)	0.25	1.25	4.25	6.25	6.25	10.25

— smallest distance

thus,

$c_1^{(2)} = \{94, 90, 87, 70\}$

$c_2^{(2)} = \{92, 80\}$

centroids:

$\bar{x}_1 = \left(\frac{4+5+3+2}{4}; \frac{6+8+4+4}{4}\right) =$

$(3.5; 5.5)$

$\bar{x}_2 = \left(\frac{5+6}{2}; \frac{7+8}{2}\right) = (5.5; 7.5)$

$d_{94, c_1} = (4-3.5)^2 + (6-5.5)^2 = 0.5$

$d_{94, c_2} = (4-5.5)^2 + (6-7.5)^2 = 4.5$

⋮

	$\bar{x}_i$	94	92	90	87	80	70
$c_1^{(2)}$	(3.5; 5.5)	0.5	4.5	8.5	2.5	12.5	4.5
$c_2^{(2)}$	(5.5; 7.5)	4.5	0.5	0.5	18.5	0.5	24.5

thus,

$c_1^{(3)} = \{94, 87, 70\}$

$c_2^{(3)} = \{92, 90, 80\}$

centroids

$\bar{x}_1 = (3; 4.67)$

$\bar{x}_2 = (5.33; 7.67)$

recalculated $d^2_{(x, y)}$

	$\bar{x}_i$	94	92	90	87	80	70
$c_1^{(3)}$	(3; 4.67)	2.77	9.43	15.09	0.45	20.09	1.45
$c_2^{(3)}$	(5.33; 7.67)	4.56	0.56	0.22	18.9	0.56	24.56

$c_1^{(4)} = c_1^{(3)}$

$c_2^{(4)} = c_2^{(3)}$

No changes, so the algorithm stops

Final clusters:

$$C_1 = \{94, 87, 70\} \text{ and } C_2 = \{92, 90, 80\}$$

6.4- Choosing the number of clusters

In hierarchical clustering choose the number of clusters based on the largest change in the merging distances.

- Mojena (1977): choose the number of clusters by the first stage in the dendrogram at which

$h_j > \bar{h} + K s_h$, where $h_1, \dots, h_n$ are the threshold distances

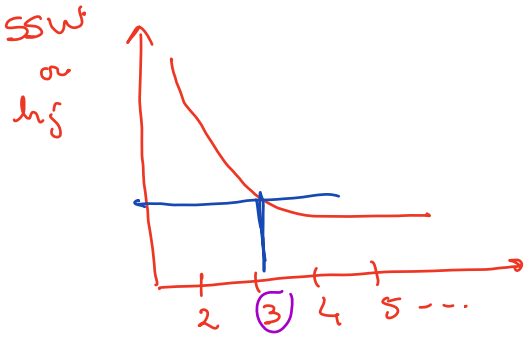
$$\bar{h} = \frac{\sum_i h_i}{n} \quad \text{and} \quad s_h = \sqrt{\frac{\sum_i (h_i - \bar{h})^2}{n-1}}$$

and $K = 1.25$ (suggested by Milligan and Cooper (1985))

- cut the dendrogram at $\overline{h}$

- Elbow Method

1. compute clustering algorithm (e.g., K-means, hierarchical) for different values of K
2. For each K, calculate the WSS (total within-cluster sum of squares) or h_K
3. Plot the curve of WSS according to the number of clusters
4. the location of the elbow in the plot is generally considered as an indicator of the number of clusters



- Silhouette index

the silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation)

the silhouette plot displays a measure of how close each point in a cluster is to points in the neighboring clusters and thus provides a way to assess the "best" number of clusters visually.

the silhouette coefficient is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample.

Silhouette coefficient (S(i))

$$S(i) = (b(i) - a(i)) / \max\{a(i), b(i)\},$$

where $a(i)$ = average dissimilarity of the ith object to all objects in the same cluster

$b(i)$ = average dissimilarity of the ith object to all objects in the closest cluster

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

$S(i)$ near 1 $\Rightarrow$ object is well clustered

$S(i)$ near 0 $\Rightarrow$ object could be assign to another cluster closest to it

$S(i)$ near -1 $\Rightarrow$ object are probably placed in the wrong cluster

How Good is the clustering?

Cophenetic Correlation

The cophenetic correlation can be used as some kind of measure of the goodness of fit of a particular dendrogram.

$$\rho_{Cophenetic} = \frac{\sum_{i=1,j=1,i<j}^{n}(d_{ij} - \bar{d})(h_{ij} - \bar{h})}{\left(\sum_{i=1,j=1,i<j}^{n}(d_{ij} - \bar{d})^2(h_{ij} - \bar{h})^2\right)^{0.5}} \quad (2)$$

Easily extracted in R, but less clear what it means. A value below 0.6 implies some distortion in the dendrogram.

Obs: d_{ij} is the ordinary Euclidean distance between the ith and jth observations and h_{ij} is the dendrogrammatic distance between the model points i and j. This distance is the height of the node at which these two points are first joined together.

Agglomerative coefficient

- Agglomerative coefficient (AC):(cluster library in R) is a measure of the clustering structure of the dataset.
- For each observation i, denote by $m(i)$ its dissimilarity to the first cluster it is merged with, divided by the dissimilarity of the merger in the final step of the algorithm. The AC is the average of all $1 - m(i)$.

$0 \leq AC \leq 1$ $AC \to 1$ good clustering

Reminder: linkages

Our setup: given $X_1, \dots X_n$ and pairwise dissimilarities d_{ij}. (E.g., think of $X_i \in \mathbb{R}^p$ and $d_{ij} = \|X_i - X_j\|_2$)

Single linkage: measures the closest pair of points

$$d_{\text{single}}(G, H) = \min_{i \in G,\, j \in H} d_{ij}$$

Complete linkage: measures the farthest pair of points

$$d_{\text{complete}}(G, H) = \max_{i \in G,\, j \in H} d_{ij}$$

Average linkage: measures the average dissimilarity over all pairs

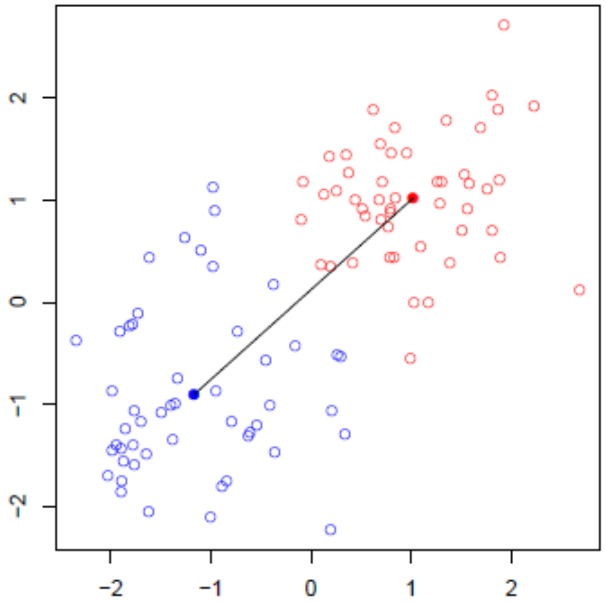
$$d_{\text{average}}(G, H) = \frac{1}{n_G \cdot n_H} \sum_{i \in G,\, j \in H} d_{ij}$$

Centroid linkage

Centroid linkage[1] is commonly used. Assume that $X_i \in \mathbb{R}^p$, and $d_{ij} = \|X_i - X_j\|_2$. Let $\bar{X}_G, \bar{X}_H$ denote group averages for G, H. Then:

$$d_{\text{centroid}}(G, H) = \|\bar{X}_G - \bar{X}_H\|_2$$

Example (dissimilarities d_{ij} are distances, groups are marked by colors): centroid linkage score $d_{\text{centroid}}(G, H)$ is the distance between the group centroids (i.e., group averages)



[1] Eisen et al. (1998), "Cluster Analysis and Display of Genome-Wide Expression Patterns"

Pros and Cons of Hierarchical Clustering

- An advantage of hierarchical clustering methods is their computational speed for small data sets
- Another advantage is that the dendrogram gives a picture of the clustering solution for a variety of choices of k
- On the other hand, a major disadvantage is that once two clusters have been joined, they can never be split apart later in the algorithm, even if such a move would improve the clustering
- The so-called partitioning methods of cluster analysis do not have this restriction
- In addition, hierarchical methods can be less efficient than partitioning methods for large data sets, when n is much greater than k

Centroid-based clustering (Partitioning method)

In centroid-based clustering, clusters are represented by a central vector, which may not necessarily be a member of the data set.

When the number of clusters is fixed to k, k-means clustering gives a formal definition as an optimization problem: find the k cluster center and assign the objects to the nearest cluster center, such that the squared distances from the cluster are minimized.

The optimization problem itself is known to be NP-hard, and thus the common approach is to search only for approximate solutions. A particularly well known approximative method is "k-means algorithm" (multivariate statistics) often actually referred to as Lloyd's algorithm (computer science).

How Do Partitioning Methods Work?

- Given n objects and k clusters, find a partition of k clusters that minimizes a given score
- Each of the k clusters is usually identified by its centroid C_m with m is the cluster identifier
- Sum of squares is a rather typical score for partitioning methods
- Global optimal is possible exhaustively enumerate all partitions
- Heuristic methods are always used (k-means and k-medoids)

k-means

The K-means algorithm is one of the most popular iterative descent clustering methods. It is intended for situations in which all variables are of the quantitative type, and squared Euclidean distance is chosen as the dissimilarity measure

k-means

- Given n objects with measures $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$, we want to split in k clusters/groups $\mathbf{C} = C_1, C_2, \ldots, C_k$, $k \leq n$, such that minimize the sum of squared distances in each cluster:

$$\arg\min_{\mathbf{C}} \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in C_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2,$$

where $\boldsymbol{\mu}_i$ is the mean in group C_i.

k-means algorithm

Given a current set of k means $m_1^{(1)}, m_2^{(1)}, \ldots, m_k^{(1)}$:

Assignment step: Assigning each object to the closest (current) cluster mean:

$$C_i^{(t)} = \left\{ \mathbf{x}_p : \left\| \mathbf{x}_p - \mathbf{m}_i^{(t)} \right\| \leq \left\| \mathbf{x}_p - \mathbf{m}_j^{(t)} \right\| \ \forall \ 1 \leq j \leq k \right\},$$

where each object with measure $\mathbf{x}_p$ is assign exactly to one group $C^{(t)}$.

Update step: Calculate the new means to be the centroids of the observations in the new clusters:

$$\mathbf{m}_i^{(t+1)} = \frac{1}{|C_i^{(t)}|} \sum_{\mathbf{x}_j \in C_i^{(t)}} \mathbf{x}_j$$

k-means algorithm

This is done iteratively by repeating the two steps until a stopping criterion is met. We can apply one of the following termination conditions:

- A fixed number of iterations has been completed. This condition limits the runtime of the clustering algorithm, but in some cases the quality of the clustering will be poor because of an insufficient number of iterations
- Assignment of objects to clusters (the partitioning function) does not change between iterations
- Centroids $\mathbf{m}_i$ do not change between iterations. This is equivalent to partitioning function not changing

Let's take a look: k-means

Pros and Cons of k-means

Drawbacks

Sensitive to initial seed points

Converge to a local optimum that may be unwanted solution

Need to specify k, the number of clusters, in advance

Unable to handle noisy data and outliers

Not suitable for discovering clusters with non-convex shapes

Applicable only when mean is defined, then what about categorical data?

Advantages

Efficient in computation $O(tkn)$, where n is number of objects, k is number of clusters, and t is number of iterations. Normally, $k, t << n$

k-means

The final clustering depend on the initial cluster center. Sometimes, different initial center lead to very different final outputs.

So, we typically run k-means multiple times (e.g., 10 times), randomly initializing clusters center for each run, then choose among from collection of center based on which on gives the smallest within-clusters variation

k-medoids*

As discussed above, the k-means algorithm is appropriate when the dissimilarity measure is taken to be squared Euclidean distance

This requires all of the variables to be of the quantitative type. In addition, using squared Euclidean distance places the highest influence on the largest distances. This causes the procedure to lack robustness against outliers that produce very large distances.

These restrictions can be removed at the expense of computation

k-medoids

The only part of the k-means algorithm that assumes squared Euclidean distance is the minimization step; the cluster representatives $\{\mathbf{m}_1, ..., \mathbf{m}_k\}$ are taken to be the means of the currently assigned clusters.

The algorithm can be generalized for use with arbitrarily defined dissimilarities $d(\mathbf{x}_i, \mathbf{x}_j)$ by replacing this step by an explicit optimization with respect to $\{\mathbf{m}_1, ..., \mathbf{m}_k\}$ –> k-medoids algorithm

k-medoids

The k-medoids algorithm is a clustering algorithm related to the k-means algorithm and the medoidshift algorithm.

Both the k-means and k-medoids algorithms are partitional (breaking the dataset up into groups) and both attempt to minimize the distance between points labelled to be in a cluster and a point designated as the centre of that cluster.

In contrast to the k-means algorithm, k-medoids chooses data points as centres (medoids or representatives) and works with an arbitrary matrix of distances between data points

k-medoids

The most common realisation of k-medoid clustering is the Partitioning Around Medoids (PAM) algorithm:

1. Initialize: randomly select k of the n data points as the medoids
2. Associate each data point to the closest medoid. ("closest" here is defined using any valid similarity measure)
3. For each medoid *m*
    1. For each non-medoid data point *l*
        1. Swap *m* and *l* and compute the total cost of the configuration
4. Select the configuration with the lowest cost
5. Repeat steps 2 to 4 until there is no change in the medoid

k-medoids

It is more robust to noise and outliers as compared to k-means because it minimizes a sum of pairwise dissimilarities instead of a sum of squared Euclidean distances.

A medoid can be defined as the object of a cluster, whose average dissimilarity to all the objects in the cluster is minimal i.e. it is a most centrally located point in the cluster.

A useful tool for determining k is the silhouette

Silhouette

Silhouette refers to a method of interpretation and validation of clusters of data. The technique provides a succinct graphical representation of how well each object lies within its cluster

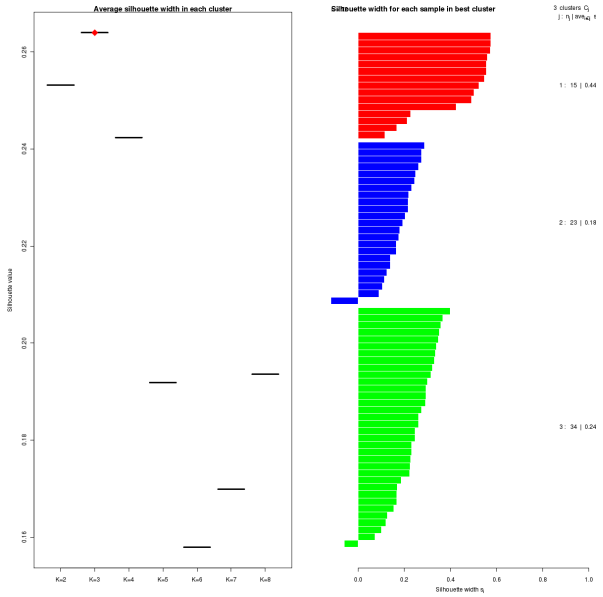


Figure 1. Silhouette width was calculated and the average silhouette width for all samples within one cluster was shown below according to different clusters (left panel). The robust cluster was pointed out by blue symbol (left panel) and the silhouette width of each sample in robust cluster was shown on right panel

Silhouette

From Peter J. Rousseeuw (1986):

"Is a graphical display proposed for partitioning techniques.

Each cluster is represented by a so-called silhouette, which is based on the comparison of its tightness and separation

This silhouette shows which objects lie well within their cluster, and which ones are merely somewhere in between clusters

The entire clustering is displayed by combining the silhouettes into a single plot, allowing an appreciation of the relative quality of the clusters and an overview of the data configuration

The average silhouette width provides an evaluation of clustering validity, and might be used to select an "appropriate" number of clusters"

Choosing the number of clusters

Sometimes, using K-means, K-medoids, or hierarchical clustering, we might have no problem specifying the number of clusters K ahead of time, e.g.,

- Segmenting a client database into K clusters for K salesman
- Compressing an image using vector quantization, where K controls the compression rate

Other times, K is implicitly defined by cutting a hierarchical clustering tree at a given height, e.g., designing a clever radio system or placing cell phone towers

But in most exploratory applications, the number of clusters K is unknown. So we are left asking the question: what is the "right" value of K?

This is a hard problem

Determining the number of clusters is a hard problem!

Why is it hard?

- Determining the number of clusters is a hard task for humans to perform (unless the data are low-dimensional). Not only that, it's just as hard to explain what it is we're looking for. Usually, statistical learning is successful when at least one of these is possible

Why is it important?

- E.g., it might mean a big difference scientifically if we were convinced that there were $K = 2$ subtypes of breast cancer vs. $K = 3$ subtypes
- One of the (larger) goals of data mining/statistical learning is automatic inference; choosing K is certainly part of this

Within-cluster variation

We're going to focus on K-means, but most ideas will carry over to other settings

Recall: given the number of clusters K, the K-means algorithm approximately minimizes the within-cluster variation:

$$W = \sum_{k=1}^{K} \sum_{C(i)=k} \|X_i - \bar{X}_k\|_2^2$$

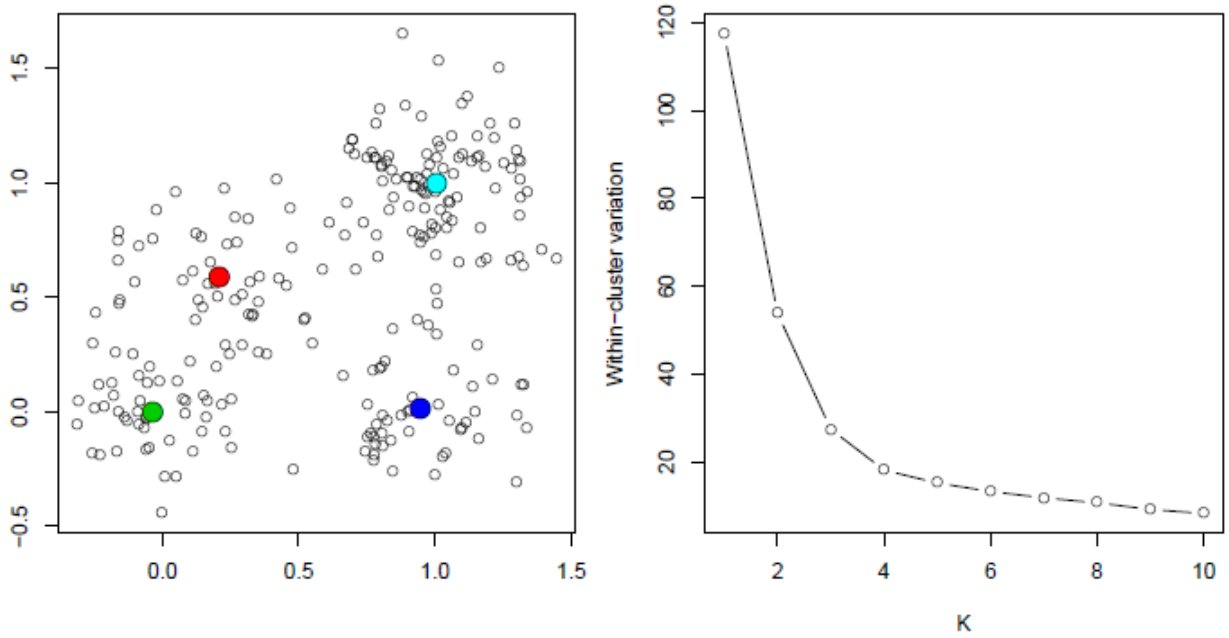
over clustering assignments C, where $\bar{X}_k$ is the average of points in group k, $\bar{X}_k = \frac{1}{n_k} \sum_{C(i)=k} X_i$

Clearly a lower value of W is better. So why not just run K-means for a bunch of different values of K, and choose the value of K that gives the smallest $W(K)$?

That is not going to work

Problem: within-cluster variation just keeps decreasing

Example: $n = 250$, $p = 2$, $K = 1, \ldots 10$



Between-cluster variation

Within-cluster variation measures how tightly grouped the clusters are. As we increase the number of clusters K, this just keeps going down. What are we missing?

Between-cluster variation measures how spread apart the groups are from each other:

$$B = \sum_{k=1}^{K} n_K \|\bar{X}_k - \bar{X}\|_2^2$$

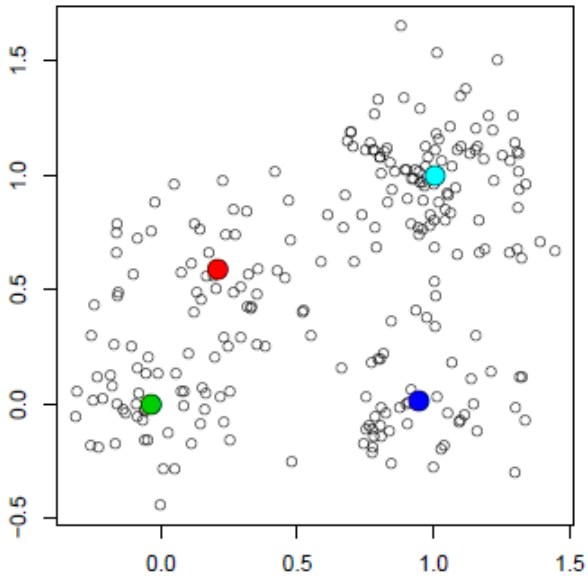
where as before $\bar{X}_k$ is the average of points in group k, and $\bar{X}$ is the overall average, i.e.

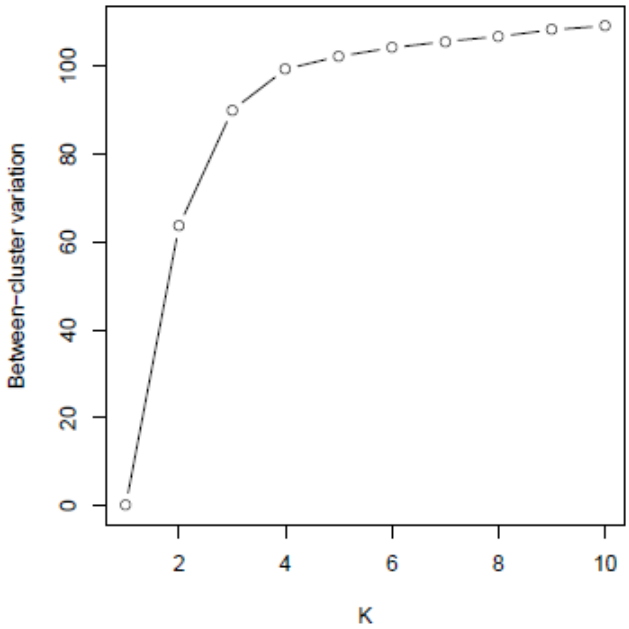
$$\bar{X}_k = \frac{1}{n_k} \sum_{C(i)=k} X_i \quad \text{and} \quad \bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i$$

Still not going to work

Bigger B is better, can we use it to choose K? Problem: between-cluster variation just keeps increasing

Running example: $n = 250$, $p = 2$, $K = 1, \ldots 10$





CH index

Ideally we'd like our clustering assignments C to simultaneously have a small W and a large B

This is the idea behind the CH index.[3] For clustering assignments coming from K clusters, we record CH score:

$$\text{CH}(K) = \frac{B(K)/(K-1)}{W(K)/(n-K)}$$

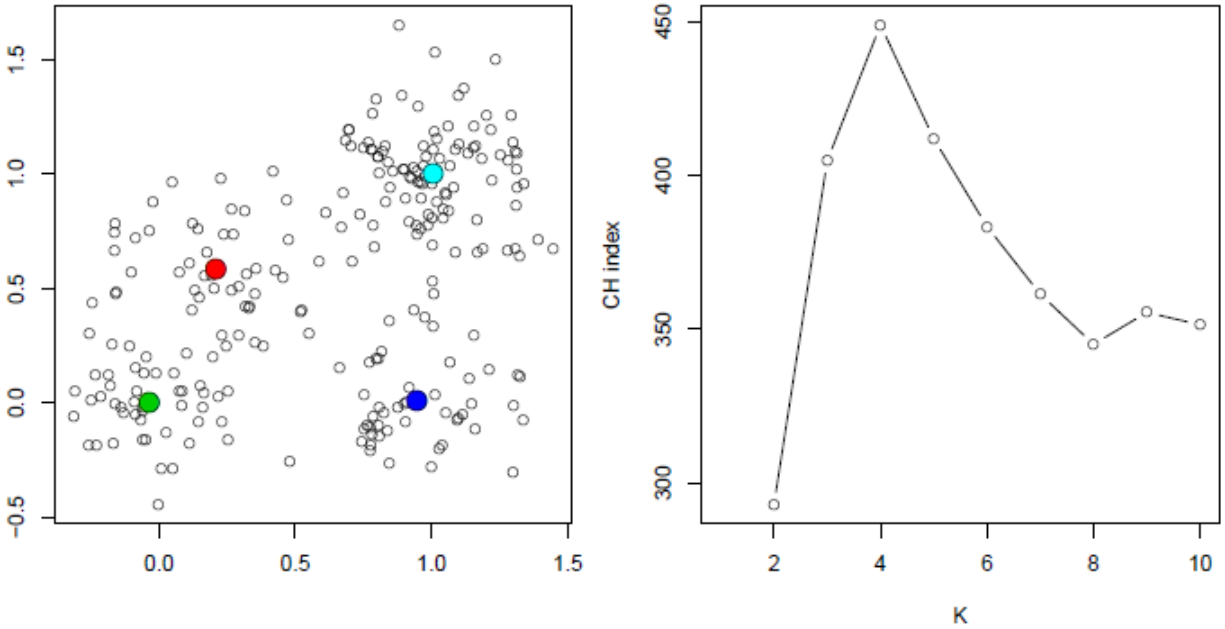
To choose K, just pick some maximum number of clusters to be considered K_{max} (e.g., $K = 20$), and choose the value of K with the largest score $\text{CH}(K)$, i.e.,

$$\hat{K} = \underset{K \in \{2,\ldots K_{\text{max}}\}}{\text{argmax}} \ \text{CH}(K)$$

[3] Calinski and Harabasz (1974), "A dendrite method for cluster analysis"

Example: CH index

Running example: $n = 250$, $p = 2$, $K = 2, \ldots 10$.



We would choose $K = 4$ clusters, which seems reasonable

General problem: the CH index is not defined for $K = 1$. We could never choose just one cluster (the null model)!

Observations

Same final notes

Standardization of Observations

If the variables in our data set are of different types or are measured on very different scales, then some variables may play an inappropriately dominant role in the clustering process

In this case, it is recommended to standardize the variables in some way before clustering the objects. Possible standardization approaches:

1. Divide each column by its sample standard deviation, so that all variables have standard deviation 1

2. Divide each variable by its sample range (max-min); Milligan and Cooper (1988) found that this approach best preserved the clustering structure

3. Convert data to z-scores by (for each variable) subtracting the sample mean and then dividing by the sample standard deviation - a common option in clustering software package

Cluster Analysis- Interpreting the clusters

The cluster centroid (a mean profile of the cluster on each cluster variable) is particularly useful in the interpretation stage

Interpretation involves:

Examining and distinguishing characteristics of each cluster's profile and identifying substantial differences between clusters

Cluster solution failing to reveal significant differences indicate that other solutions should be examined

The cluster centroid should also be assessed for correspondence to researcher's prior expectation based on theory or practical experience

Cluster Analysis - Validation

"*The validation of clustering structures is the most difficult and frustrating part of cluster analysis. Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage*."

1. Determining the clustering tendency of a set of data, i.e., distinguishing whether nonrandom structure actually exists in the data

2. Comparing the results of a cluster analysis to externally known results, e.g., to externally given class labels

3. Evaluating how well the results of a cluster analysis fit the data without reference to external information

4. Comparing the results of two different sets of cluster analyses to determine the stability of the solution

5. Determining the "correct" number of clusters

Some numbers...

- $S_{k,n}$, the number of ways of partitioning n objects into k groups is given by:

$$S_{k,n} = \frac{1}{k!} \sum_{j=1}^{k} \binom{k}{j} (-1)^{k-j} j^n \approx_{n \to \infty} \frac{k^n}{k!}$$

  a second type Stirling number.
- Where k is not specified we have $\sum_{k=1}^{K} S_{k,n}$ partitions.
- For $n = 50$ and $k = 2$ this is in the order of 6×10^{29}