

# Data Clustering - Part 1

## M2 DMKM

Julien Ah-Pine (julien.ah-pine@univ-lyon2.fr)

Université Lyon 2

2015-2016

## Organization

Organization : 3 lessons (of 3h each)

Outline of today's lesson :

- 1 The clustering problem
- 2 Different types of data and different types of proximity measures
  - Continuous variables
  - Discrete variables and binary data
  - Mixed-typed data

## Data Clustering : what is it about ?

**Data clustering** (or just clustering), also called **cluster analysis**, **segmentation analysis**, **taxonomy analysis**, or **unsupervised classification** : set of methods that aim at creating groups of objects, or clusters, such that objects in a cluster are very similar and objects in different clusters are distinct.

## Data Clustering : what is it about ?

**Data clustering** (or just clustering), also called **cluster analysis**, **segmentation analysis**, **taxonomy analysis**, or **unsupervised classification** : set of methods that aim at creating groups of objects, or clusters, such that objects in a cluster are very similar and objects in different clusters are distinct.

Do not confuse it with *supervised classification* : in a *supervised* context, objects are assigned to predefined classes, or categories or labels and the goal is to learn a decision function from a training (labeled) dataset in order to correctly categorize new objects.

## Data Clustering : what is it about ?

**Data clustering** (or just clustering), also called **cluster analysis**, **segmentation analysis**, **taxonomy analysis**, or **unsupervised classification** : set of methods that aim at creating groups of objects, or clusters, such that objects in a cluster are very similar and objects in different clusters are distinct.

Do not confuse it with *supervised classification* : in a *supervised* context, objects are assigned to predefined classes, or categories or labels and the goal is to learn a decision function from a training (labeled) dataset in order to correctly categorize new objects.

In data clustering the goal is to automatically discover a classification of the objects.

## Data Clustering : what is it about ?

**Data clustering** (or just clustering), also called **cluster analysis**, **segmentation analysis**, **taxonomy analysis**, or **unsupervised classification** : set of methods that aim at creating groups of objects, or clusters, such that objects in a cluster are very similar and objects in different clusters are distinct.

Do not confuse it with *supervised classification* : in a *supervised* context, objects are assigned to predefined classes, or categories or labels and the goal is to learn a decision function from a training (labeled) dataset in order to correctly categorize new objects.

In data clustering the goal is to automatically discover a classification of the objects.

In the following, the term **classification** will refer to the concept of organizing similar objects into groups.

## Classification in human activities and sciences

Classification is a basic human activity :

- Early men must have been able to realize that many individual objects shared certain properties such as eatable, or poisonous. . .
- Classification is needed for the developmet of langage : each noun in a langage, is essentially a label used to describe a class of things which have features in common . . .

## Classification in human activities and sciences

Classification is a basic human activity :

- Early men must have been able to realize that many individual objects shared certain properties such as eatable, or poisonous. . .
- Classification is needed for the developmet of langage : each noun in a langage, is essentially a label used to describe a class of things which have features in common . . .

Classification is also fundamental to most branches of science :

- In biology, the theory and practice of classifying organisms is generally known as taxonomy
- In chemistry, classification of chemical elements regarding to their atomic structure
- In astronomy, classification of stars and galaxies . . .

## Classification in human activities and sciences

Classification is a basic human activity :

- Early men must have been able to realize that many individual objects shared certain properties such as eatable, or poisonous. . .
- Classification is needed for the developmet of langage : each noun in a langage, is essentially a label used to describe a class of things which have features in common . . .

Classification is also fundamental to most branches of science :

- In biology, the theory and practice of classifying organisms is generally known as taxonomy
- In chemistry, classification of chemical elements regarding to their atomic structure
- In astronomy, classification of stars and galaxies . . .

A classification scheme may simply represent a convenient **method for organizing a set of objects** so that it can be **understood more easily** and the **information it conveys retrieved more efficiently**.

## Data clustering : why is it useful ?

There is an ever growing number of large databases available in many areas of science due to the development of IT. For large datasets, designing a classification scheme by hand is unfeasible.

In that context, **data clustering** is the process which aims at automatically discovering a classification scheme in order to organize the objects of a large database.

## Data clustering : why is it useful ?

There is an ever growing number of large databases available in many areas of science due to the development of IT. For large datasets, designing a classification scheme by hand is unfeasible.

## Data clustering : why is it useful ?

There is an ever growing number of large databases available in many areas of science due to the development of IT. For large datasets, designing a classification scheme by hand is unfeasible.

In that context, **data clustering** is the process which aims at automatically discovering a classification scheme in order to organize the objects of a large database.

The exploration of such databases using data clustering and other multivariate analysis techniques is now often called **data mining**.

## Some applications of data clustering

- In market research, cluster analysis is used to segment the market and determine target markets. Another example : group a large number of respondents according to their preferences for particular products and identification of a “niche product” for a particular type of consumers.

## Some applications of data clustering

- In market research, cluster analysis is used to segment the market and determine target markets. Another example : group a large number of respondents according to their preferences for particular products and identification of a “niche product” for a particular type of consumers.
- In information retrieval, cluster analysis is used to cluster the results provided by a search engine in order to organize the web pages according to topics. User can browse the retrieved items in a more efficient way (for example, have you ever tried yippy.com (formerly clusty.com) ? )

## Some applications of data clustering

- In market research, cluster analysis is used to segment the market and determine target markets. Another example : group a large number of respondents according to their preferences for particular products and identification of a “niche product” for a particular type of consumers.
- In information retrieval, cluster analysis is used to cluster the results provided by a search engine in order to organize the web pages according to topics. User can browse the retrieved items in a more efficient way (for example, have you ever tried yippy.com (formerly clusty.com) ? )
- In image processing, cluster analysis is used to segment a gray-scale or a color image in order to detect objects represented in the image and/or to compress the image.

## Some applications of data clustering

- In market research, cluster analysis is used to segment the market and determine target markets. Another example : group a large number of respondents according to their preferences for particular products and identification of a “niche product” for a particular type of consumers.
- In information retrieval, cluster analysis is used to cluster the results provided by a search engine in order to organize the web pages according to topics. User can browse the retrieved items in a more efficient way (for example, have you ever tried yippy.com (formerly clusty.com) ? )
- In image processing, cluster analysis is used to segment a gray-scale or a color image in order to detect objects represented in the image and/or to compress the image.
- In on-line social network analysis (such as Facebook, LinkedIn,...), graph data clustering is used in order to detect communities among people. . .

## Bibliography

Books on data clustering (and data-mining) this course is based on :

- M. Berry, M. Browne. *Lecture Notes in Data Mining*. 2006. World Scientific Publishing.
- B.S. Everitt, S. Landau, M. Leese, D. Stahl. *Cluster Analysis (5th Ed)*. 2011. John Wiley and Sons.
- G. Gan, C. Ma, J. Wu. *Data Clustering - Theory, Algorithms and Applications*. 2007. SIAM.
- J. Han, M. Kamber. *Data Mining - Concepts and Techniques (2nd Ed)*. 2006. Morgan Kaufmann Publishers.
- T. Hastie, R. Tibshirani, J. Friedman. *Elements of Statistical Learning Theory (2nd Ed)*. 2009. Springer.

## Vocabulary and notations

In the literature of data clustering, different words may be used to express the same thing. We are given a database or a dataset to which we associate a data table  $\mathbf{X}$  with  $n$  rows and  $p$  columns.

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \dots & \dots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix}$$

## Outline

- 1 The clustering problem
- 2 Different types of data and different types of proximity measures
  - Continuous variables
  - Discrete variables and binary data
  - Mixed-typed data

## Vocabulary and notations

In the literature of data clustering, different words may be used to express the same thing. We are given a database or a dataset to which we associate a data table  $\mathbf{X}$  with  $n$  rows and  $p$  columns.

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \dots & \dots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix}$$

- A row of  $\mathbf{X}$  is associated to an object, a data point or an item. . .

## Vocabulary and notations

In the literature of data clustering, different words may be used to express the same thing. We are given a database or a dataset to which we associate a data table  $\mathbf{X}$  with  $n$  rows and  $p$  columns.

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \dots & \dots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix}$$

- A row of  $\mathbf{X}$  is associated to an object, a data point or an item. . .
- $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})$  is the (column) vector of size  $(p \times 1)$  associated to the data point  $\mathbf{x}_i$  or  $\mathbf{i}$  of the set of items  $\mathbb{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$

## Vocabulary and notations

In the literature of data clustering, different words may be used to express the same thing. We are given a database or a dataset to which we associate a data table  $\mathbf{X}$  with  $n$  rows and  $p$  columns.

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \dots & \dots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix}$$

- A row of  $\mathbf{X}$  is associated to an object, a data point or an item. . .
- $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})$  is the (column) vector of size  $(p \times 1)$  associated to the data point  $\mathbf{x}_i$  or  $\mathbf{i}$  of the set of items  $\mathbb{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$
- We will also use  $\mathbf{x}$  and  $\mathbf{y}$  to represent two vectors of  $\mathbb{D}$

## Vocabulary and notations

In the literature of data clustering, different words may be used to express the same thing. We are given a database or a dataset to which we associate a data table  $\mathbf{X}$  with  $n$  rows and  $p$  columns.

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \dots & \dots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix}$$

- A row of  $\mathbf{X}$  is associated to an object, a data point or an item. . .
- $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})$  is the (column) vector of size  $(p \times 1)$  associated to the data point  $\mathbf{x}_i$  or  $\mathbf{i}$  of the set of items  $\mathbb{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$
- We will also use  $\mathbf{x}$  and  $\mathbf{y}$  to represent two vectors of  $\mathbb{D}$
- A column of  $\mathbf{X}$  is an attribute, a variable or a feature. . . Objects are represented as vectors of the space generated by the set of features. The representation space is also called the input space.

## Vocabulary and notations

In the literature of data clustering, different words may be used to express the same thing. We are given a database or a dataset to which we associate a data table  $\mathbf{X}$  with  $n$  rows and  $p$  columns.

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \dots & \dots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix}$$

- A row of  $\mathbf{X}$  is associated to an object, a data point or an item. . .
- $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})$  is the (column) vector of size  $(p \times 1)$  associated to the data point  $\mathbf{x}_i$  or  $\mathbf{i}$  of the set of items  $\mathbb{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$
- We will also use  $\mathbf{x}$  and  $\mathbf{y}$  to represent two vectors of  $\mathbb{D}$
- A column of  $\mathbf{X}$  is an attribute, a variable or a feature. . . Objects are represented as vectors of the space generated by the set of features. The representation space is also called the input space.
- $x_{ij}$  is the value of variable  $j$  assigned to the data point  $\mathbf{i}$  (or  $\mathbf{x}_i$ )

## Different types of classification scheme

We can have different types of classification schemes :

- A flat partition (set of clusters or segments)
- A hierarchical tree or taxonomy (a set of nested partitions)
- Hard or soft (or fuzzy) memberships to clusters

## Clustering or segmentation or partition is the same as equivalence relation

### Definition. (Binary relation on $\mathbb{D}$ )

A binary relation  $R$  on a set of objects  $\mathbb{D}$ , is a couple  $(\mathbb{D}, G(R))$ , where  $G(R)$  called the graph of the relation  $R$ , is a subset of the Cartesian product  $\mathbb{D} \times \mathbb{D}$ . If we have  $(\mathbf{x}, \mathbf{y}) \in G(R)$ , then we say that object  $\mathbf{x}$  is in relation with object  $\mathbf{y}$  for the relation  $R$ . This will be denoted by  $\mathbf{xRy}$ .

### Definition. (Equivalence relation on $\mathbb{D}$ )

A binary relation  $(\mathbb{D}, G(R))$  is an equivalence relation is it satisfies the following properties :

- Reflexivity :  $\forall \mathbf{x} (\mathbf{xRx})$
- Symmetry :  $\forall \mathbf{x}, \mathbf{y} (\mathbf{xRy} \Rightarrow \mathbf{yRx})$
- Transitivity :  $\forall \mathbf{x}, \mathbf{y}, \mathbf{z} ((\mathbf{xRy} \wedge \mathbf{yRz}) \Rightarrow \mathbf{xRz})$

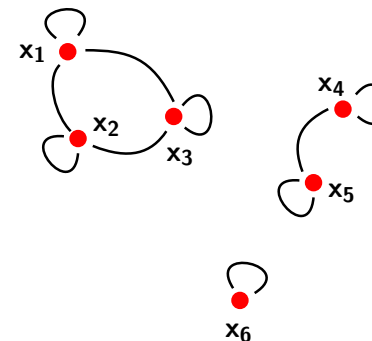
## Clustering or segmentation or partition is the same as equivalence relation

### Definition. (Binary relation on $\mathbb{D}$ )

A binary relation  $R$  on a set of objects  $\mathbb{D}$ , is a couple  $(\mathbb{D}, G(R))$ , where  $G(R)$  called the graph of the relation  $R$ , is a subset of the Cartesian product  $\mathbb{D} \times \mathbb{D}$ . If we have  $(\mathbf{x}, \mathbf{y}) \in G(R)$ , then we say that object  $\mathbf{x}$  is in relation with object  $\mathbf{y}$  for the relation  $R$ . This will be denoted by  $\mathbf{xRy}$ .

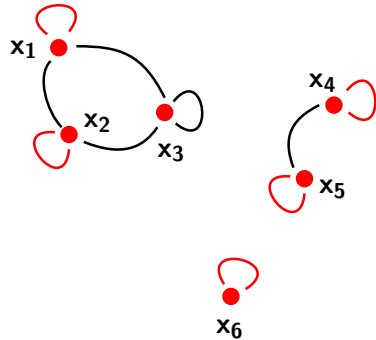
## Illustration

The graph below represents the binary relation  $R$  such that  $\mathbf{x}_i R \mathbf{x}_j \Leftrightarrow$  there's an edge between  $\mathbf{x}_i$  and  $\mathbf{x}_j$



### Illustration

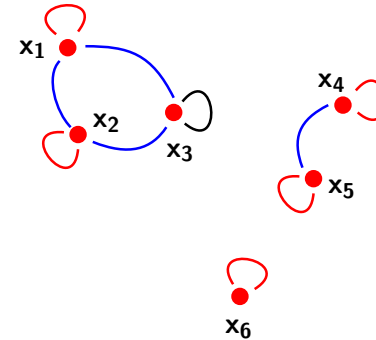
The graph below represents the binary relation  $R$  such that  $x_i R x_j \Leftrightarrow$  there's an edge between  $x_i$  and  $x_j$



- Reflexivity :  
 $x_1 R x_1, \dots, x_6 R x_6$

### Illustration

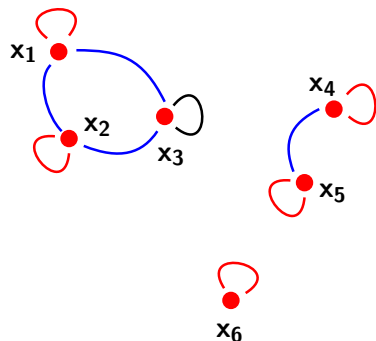
The graph below represents the binary relation  $R$  such that  $x_i R x_j \Leftrightarrow$  there's an edge between  $x_i$  and  $x_j$



- Reflexivity :  
 $x_1 R x_1, \dots, x_6 R x_6$
- Symmetry :  
 $(x_1 R x_2 \wedge x_2 R x_1),$   
...  
 $(x_4 R x_5 \wedge x_5 R x_4)$

### Illustration

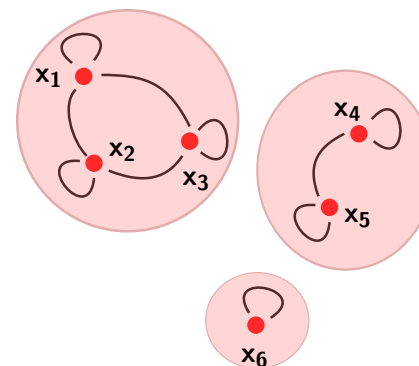
The graph below represents the binary relation  $R$  such that  $x_i R x_j \Leftrightarrow$  there's an edge between  $x_i$  and  $x_j$



- Reflexivity :  
 $x_1 R x_1, \dots, x_6 R x_6$
- Symmetry :  
 $(x_1 R x_2 \wedge x_2 R x_1),$   
...  
 $(x_4 R x_5 \wedge x_5 R x_4)$
- Transitivity :  
 $x_1 R x_2 \wedge x_2 R x_3 \Rightarrow x_1 R x_3$   
 $x_1 R x_3 \wedge x_3 R x_2 \Rightarrow x_1 R x_2$   
...  
 $x_3 R x_2 \wedge x_2 R x_1 \Rightarrow x_3 R x_1$

### Illustration

The graph below represents the binary relation  $R$  such that  $x_i R x_j \Leftrightarrow$  there's an edge between  $x_i$  and  $x_j$



- Reflexivity :  
 $x_1 R x_1, \dots, x_6 R x_6$
- Symmetry :  
 $(x_1 R x_2 \wedge x_2 R x_1),$   
...  
 $(x_4 R x_5 \wedge x_5 R x_4)$
- Transitivity :  
 $x_1 R x_2 \wedge x_2 R x_3 \Rightarrow x_1 R x_3$   
 $x_1 R x_3 \wedge x_3 R x_2 \Rightarrow x_1 R x_2$   
...  
 $x_3 R x_2 \wedge x_2 R x_1 \Rightarrow x_3 R x_1$

Partition :  $\{x_1, x_2, x_3\}, \{x_4, x_5\}, \{x_6\}$



## Hierarchical clustering and dendrograms

### Definition. (Hierarchical clustering and dendrograms)

A hierarchical clustering on a set of objects  $\mathbb{D}$  is a set of **nested partitions of  $\mathbb{D}$** . It is represented by a binary tree such that :

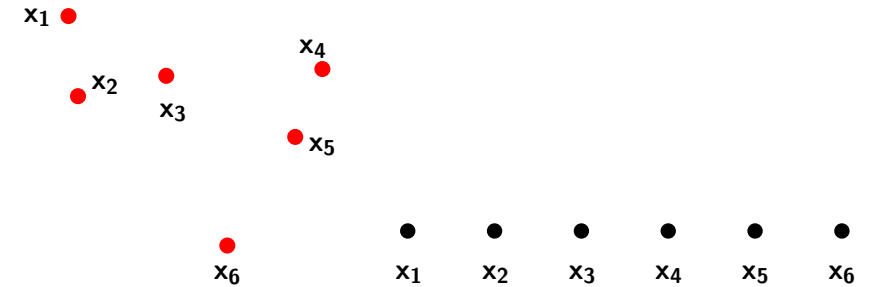
- The root node is a cluster that contains all data points
- Each (parent) node is a cluster made of two subclusters (childs)
- Each leaf node represents one data point (singleton ie cluster with only one item)

More formally, if  $n, n'$  are two nodes of the hierarchical clustering then :

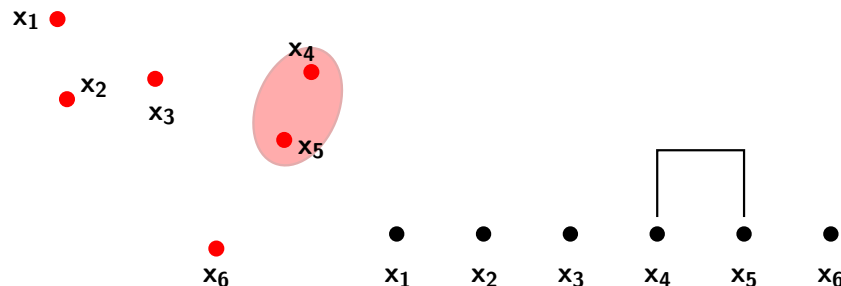
$$(n \cap n' = \emptyset) \vee (n \subset n') \vee (n' \subset n)$$

A hierarchical clustering scheme is also called a taxonomy. In data clustering the binary tree is called a **dendrogram**.

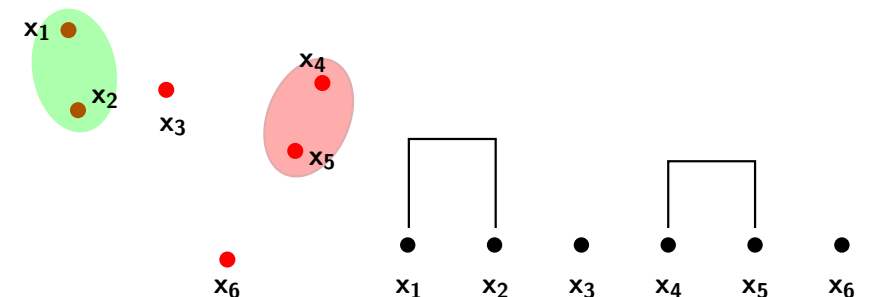
## Illustration



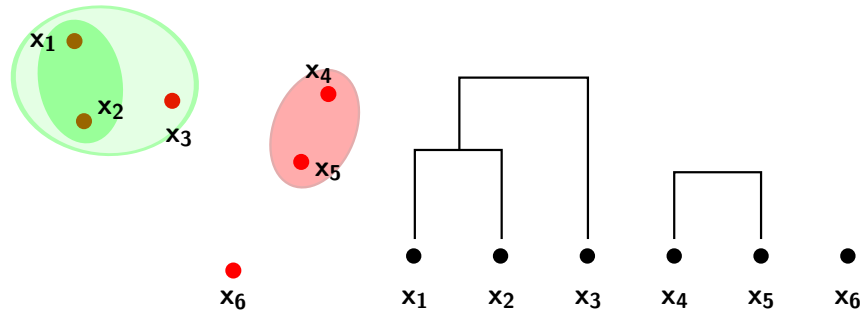
## Illustration



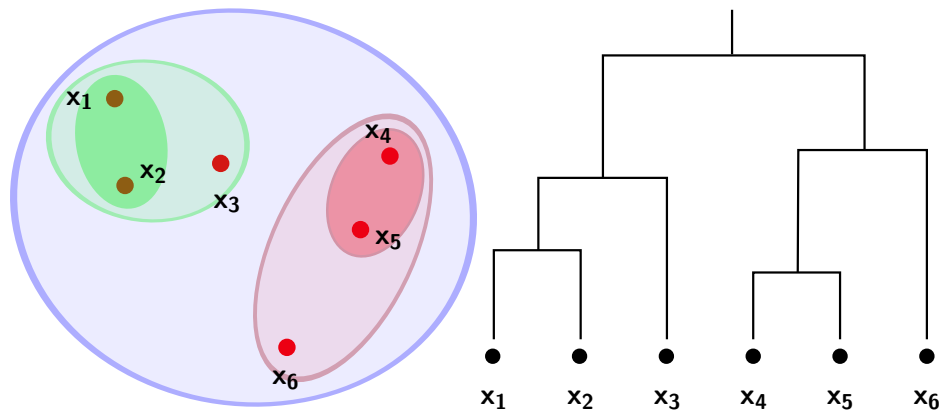
## Illustration



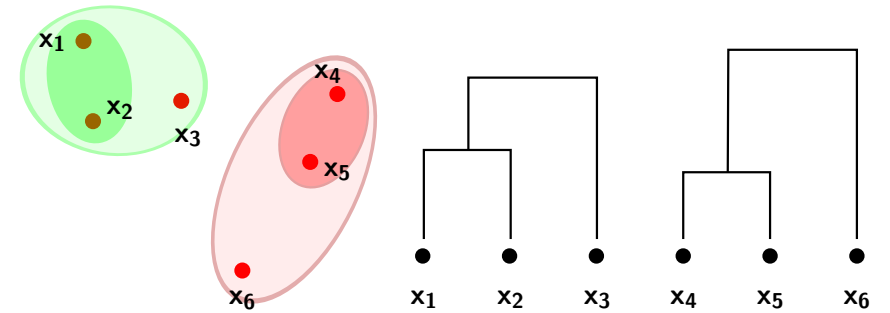
# Illustration



# Illustration



# Illustration



# The partitioning problem

We are given  $F(\mathbb{D}, C)$  which is a function that measures the quality of the clustering  $C$  given the set of data points  $\mathbb{D}$ .

## The partitioning problem

We are given  $F(\mathbb{D}, C)$  which is a function that measures the quality of the clustering  $C$  given the set of data points  $\mathbb{D}$ .

Let us denote  $\mathbb{C}_n$  the set of all partitions of a set of  $n$  data points then the clustering problem can be formally define as follows :

$$\max_{C \in \mathbb{C}_n} F(\mathbb{D}, C)$$

## A combinatorial problem

The number of partitions with  $k$  clusters of a set of  $n$  items is the Stirling number of a second kind  $S(n, k)$  :

$$S(n, k) = \frac{1}{k!} \sum_{j=0}^k (-1)^j \binom{n}{k} j^n$$

## The partitioning problem

We are given  $F(\mathbb{D}, C)$  which is a function that measures the quality of the clustering  $C$  given the set of data points  $\mathbb{D}$ .

Let us denote  $\mathbb{C}_n$  the set of all partitions of a set of  $n$  data points then the clustering problem can be formally define as follows :

$$\max_{C \in \mathbb{C}_n} F(\mathbb{D}, C)$$

A naive approach to solve the clustering problem is the following one :

- 1 Enumerate all possible partitions in  $\mathbb{C}_n$
- 2 For all  $C \in \mathbb{C}_n$  compute the value  $F(\mathbb{D}, C)$
- 3 Keep the partition  $C^*$  such that  $\forall C \in \mathbb{C}_n : F(\mathbb{D}, C^*) \geq F(\mathbb{D}, C)$

## A combinatorial problem

The number of partitions with  $k$  clusters of a set of  $n$  items is the Stirling number of a second kind  $S(n, k)$  :

$$S(n, k) = \frac{1}{k!} \sum_{j=0}^k (-1)^j \binom{n}{k} j^n$$

The total number of partitions of a set of  $n$  items is the Bell number  $B(n)$  :

$$B(n) = \sum_{k=0}^n S(n, k)$$

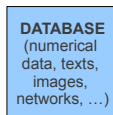
## A combinatorial problem (cont'd)

Some values of  $S(n, k)$  and  $B(n)$  :

$n \backslash k$	0	1	2	3	4	5	6	$B(n)$
0	1	0	0	0	0	0	0	1
1	0	1	0	0	0	0	0	1
2	0	1	1	0	0	0	0	2
3	0	1	3	1	0	0	0	5
4	0	1	7	6	1	0	0	15
5	0	1	15	25	10	1	0	52
6	0	1	31	90	65	15	1	203

Another example :  $B(71) \simeq 4 \times 10^{74}$  !

## The clustering process



Initially there is a database of objects which could be of any kind depending on the type of application.

## A combinatorial problem (cont'd)

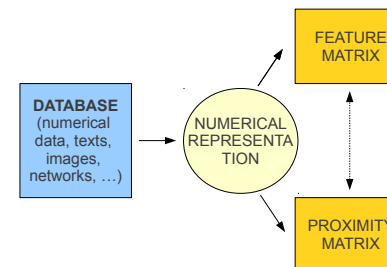
Some values of  $S(n, k)$  and  $B(n)$  :

$n \backslash k$	0	1	2	3	4	5	6	$B(n)$
0	1	0	0	0	0	0	0	1
1	0	1	0	0	0	0	0	1
2	0	1	1	0	0	0	0	2
3	0	1	3	1	0	0	0	5
4	0	1	7	6	1	0	0	15
5	0	1	15	25	10	1	0	52
6	0	1	31	90	65	15	1	203

Another example :  $B(71) \simeq 4 \times 10^{74}$  !

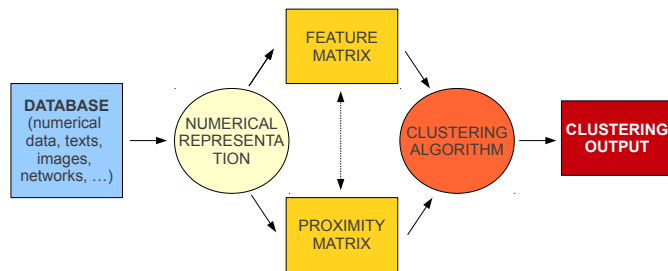
It is thus unfeasible to enumerate all possible partitions of a set whose cardinal is greater than some tens ! In practice we use heuristics ie clustering algorithms.

## The clustering process



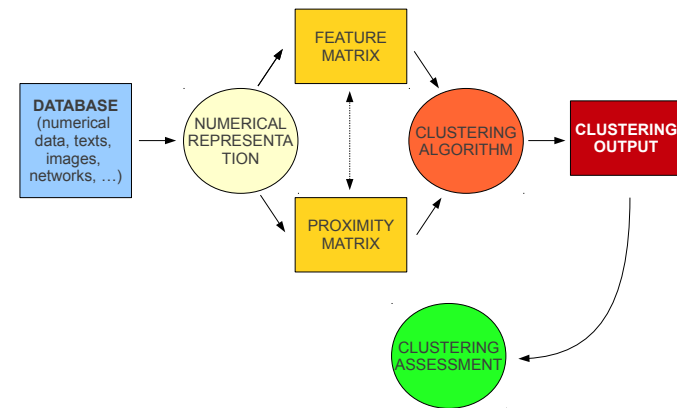
We assume that objects have a (structured) numerical representation. This is our starting point but there are different types of numerical data.

## The clustering process



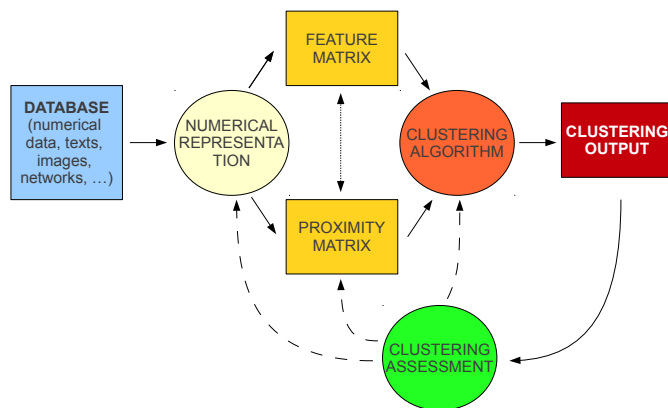
Depending on the type of clustering algorithm we can have as input a feature matrix (data table like  $X$ ) or a proximity matrix.

## The clustering process



Once the clustering algorithm is done, we have to assess the clustering outputs.

## The clustering process



Depending on the quality of the clustering output, either we keep the latter as the result or we start over with another modeling.

## Some comments

- We don't directly deal with unstructured objects such as texts, images, . . . and with how these objects can be numerically represented. This is text processing, image processing, . . .

## Some comments

- We don't directly deal with unstructured objects such as texts, images, . . . and with how these objects can be numerically represented. This is text processing, image processing, . . .
- Our input is a numerical representation of the objects which is either a feature matrix  $\mathbf{X}$  or a proximity matrix between pairs of data points.

## Some comments

- We don't directly deal with unstructured objects such as texts, images, . . . and with how these objects can be numerically represented. This is text processing, image processing, . . .
- Our input is a numerical representation of the objects which is either a feature matrix  $\mathbf{X}$  or a proximity matrix between pairs of data points.
- There are two main critical points : the proximity measure and the (model behind a) clustering algorithm.
- There are many types of numerical data and also many kinds of proximity measures.

## Some comments

- We don't directly deal with unstructured objects such as texts, images, . . . and with how these objects can be numerically represented. This is text processing, image processing, . . .
- Our input is a numerical representation of the objects which is either a feature matrix  $\mathbf{X}$  or a proximity matrix between pairs of data points.
- There are two main critical points : the proximity measure and the (model behind a) clustering algorithm.

## Some comments

- We don't directly deal with unstructured objects such as texts, images, . . . and with how these objects can be numerically represented. This is text processing, image processing, . . .
- Our input is a numerical representation of the objects which is either a feature matrix  $\mathbf{X}$  or a proximity matrix between pairs of data points.
- There are two main critical points : the proximity measure and the (model behind a) clustering algorithm.
- There are many types of numerical data and also many kinds of proximity measures.
- There are many clustering algorithms.

## Some comments

- We don't directly deal with unstructured objects such as texts, images, . . . and with how these objects can be numerically represented. This is text processing, image processing, . . .
- Our input is a numerical representation of the objects which is either a feature matrix  $\mathbf{X}$  or a proximity matrix between pairs of data points.
- There are two main critical points : the proximity measure and the (model behind a) clustering algorithm.
- There are many types of numerical data and also many kinds of proximity measures.
- There are many clustering algorithms.
- There are many ways to assess clustering methods.

## Outline

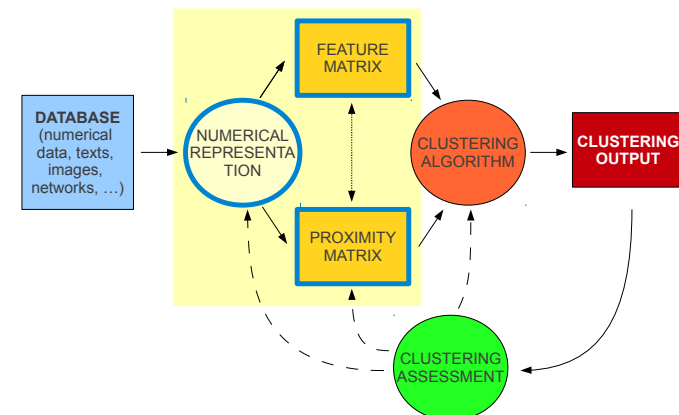
- 1 The clustering problem
- 2 Different types of data and different types of proximity measures
  - Continuous variables
  - Discrete variables and binary data
  - Mixed-typed data

## Some comments

- We don't directly deal with unstructured objects such as texts, images, . . . and with how these objects can be numerically represented. This is text processing, image processing, . . .
- Our input is a numerical representation of the objects which is either a feature matrix  $\mathbf{X}$  or a proximity matrix between pairs of data points.
- There are two main critical points : the proximity measure and the (model behind a) clustering algorithm.
- There are many types of numerical data and also many kinds of proximity measures.
- There are many clustering algorithms.
- There are many ways to assess clustering methods.

In brief, clustering is not a straightforward process !

## Recalling the clustering process



## Link with other courses

Numerical representation of objects :

- Multidimensional Data Analysis and dimension reduction techniques such as Principal Component Analysis (PCA) or Factor Analysis (FA) or Correspondence Analysis (CA), ... and information visualization :
  - ▶ To represent the data points in low dimensional euclidean spaces
  - ▶ To visualize the data points in order to see if there is a "natural" organization of the latter

## Link with other courses

Numerical representation of objects :

- Multidimensional Data Analysis and dimension reduction techniques such as Principal Component Analysis (PCA) or Factor Analysis (FA) or Correspondence Analysis (CA), ... and information visualization :
  - ▶ To represent the data points in low dimensional euclidean spaces
  - ▶ To visualize the data points in order to see if there is a "natural" organization of the latter
- One can do a dimension reduction of the data before the clustering analysis.
- Here given the representation space (reduced or not), we focus on how to measure the proximity between points :
  - ▶ What is the definition of a proximity measure ?
  - ▶ There are different types of numerical data : what are the most used proximity measures for each type ?

## Link with other courses

Numerical representation of objects :

- Multidimensional Data Analysis and dimension reduction techniques such as Principal Component Analysis (PCA) or Factor Analysis (FA) or Correspondence Analysis (CA), ... and information visualization :
  - ▶ To represent the data points in low dimensional euclidean spaces
  - ▶ To visualize the data points in order to see if there is a "natural" organization of the latter
- One can do a dimension reduction of the data before the clustering analysis.

## Definition of a dissimilarity and a distance measure

### Definition. (Dissimilarity and distance measures)

Let  $\mathbb{D}$  be a set of data points and let  $D : \mathbb{D} \times \mathbb{D} \rightarrow \mathbb{R}^+$  be a real function.

$D$  is a **dissimilarity measure** if it satisfies the following properties :

- 1 *Non-negativity* :  $\forall \mathbf{x}, \mathbf{y} : D(\mathbf{x}, \mathbf{y}) \geq 0$
- 2 *Symmetry* :  $\forall \mathbf{x}, \mathbf{y} : D(\mathbf{x}, \mathbf{y}) = D(\mathbf{y}, \mathbf{x})$
- 3 *Identity and indiscernability* :  $\forall \mathbf{x}, \mathbf{y} : \mathbf{x} = \mathbf{y} \Leftrightarrow D(\mathbf{x}, \mathbf{y}) = 0$

If a dissimilarity measure  $D$  also satisfies the following condition, it is a **distance measure**.

- 4 *Triangle inequality* :  $\forall \mathbf{x}, \mathbf{y}, \mathbf{z} : D(\mathbf{x}, \mathbf{y}) \leq D(\mathbf{x}, \mathbf{z}) + D(\mathbf{z}, \mathbf{y})$



## Definition of a dissimilarity and a distance measure

### Definition. (Dissimilarity and distance measures)

Let  $\mathbb{D}$  be a set of data points and let  $D : \mathbb{D} \times \mathbb{D} \rightarrow \mathbb{R}^+$  be a real function.  $D$  is a **dissimilarity measure** if it satisfies the following properties :

- 1 Non-negativity :  $\forall \mathbf{x}, \mathbf{y} : D(\mathbf{x}, \mathbf{y}) \geq 0$
- 2 Symmetry :  $\forall \mathbf{x}, \mathbf{y} : D(\mathbf{x}, \mathbf{y}) = D(\mathbf{y}, \mathbf{x})$
- 3 Identity and indiscernability :  $\forall \mathbf{x}, \mathbf{y} : \mathbf{x} = \mathbf{y} \Leftrightarrow D(\mathbf{x}, \mathbf{y}) = 0$

If a dissimilarity measure  $D$  also satisfies the following condition, it is a **distance measure**.

- 4 Triangle inequality :  $\forall \mathbf{x}, \mathbf{y}, \mathbf{z} : D(\mathbf{x}, \mathbf{y}) \leq D(\mathbf{x}, \mathbf{z}) + D(\mathbf{z}, \mathbf{y})$

A function  $D$  that satisfies conditions 3 ( $\forall \mathbf{x} : D(\mathbf{x}, \mathbf{x}) = 0$ ) and 4 is said to be a **metric**. Thus, a distance measure is a metric.

---

1. a finite vectorial space with a dot product

## Metric vs euclidean

### Theorem.

If  $\mathbf{D}$  is euclidean then  $D$  is a distance measure.

## Definition of a dissimilarity and a distance measure

### Definition. (Dissimilarity and distance measures)

Let  $\mathbb{D}$  be a set of data points and let  $D : \mathbb{D} \times \mathbb{D} \rightarrow \mathbb{R}^+$  be a real function.  $D$  is a **dissimilarity measure** if it satisfies the following properties :

- 1 Non-negativity :  $\forall \mathbf{x}, \mathbf{y} : D(\mathbf{x}, \mathbf{y}) \geq 0$
- 2 Symmetry :  $\forall \mathbf{x}, \mathbf{y} : D(\mathbf{x}, \mathbf{y}) = D(\mathbf{y}, \mathbf{x})$
- 3 Identity and indiscernability :  $\forall \mathbf{x}, \mathbf{y} : \mathbf{x} = \mathbf{y} \Leftrightarrow D(\mathbf{x}, \mathbf{y}) = 0$

If a dissimilarity measure  $D$  also satisfies the following condition, it is a **distance measure**.

- 4 Triangle inequality :  $\forall \mathbf{x}, \mathbf{y}, \mathbf{z} : D(\mathbf{x}, \mathbf{y}) \leq D(\mathbf{x}, \mathbf{z}) + D(\mathbf{z}, \mathbf{y})$

A function  $D$  that satisfies conditions 3 ( $\forall \mathbf{x} : D(\mathbf{x}, \mathbf{x}) = 0$ ) and 4 is said to be a **metric**. Thus, a distance measure is a metric.

If from a distance matrix  $\mathbf{D}$  with  $\mathbf{D}_{ij} = D(\mathbf{x}_i, \mathbf{x}_j)$  we can represent the data points in an euclidean space<sup>1</sup> then  $\mathbf{D}$  is said to be **euclidean**.

---

1. a finite vectorial space with a dot product

## Metric vs euclidean

### Theorem.

If  $\mathbf{D}$  is euclidean then  $D$  is a distance measure.

However, not all distance measures  $D$  give an euclidean distance matrix  $\mathbf{D}$ . Here is a counter example from [Gower and Legendre, 1986] :

## Metric vs euclidean

## Theorem.

If  $\mathbf{D}$  is euclidean then  $D$  is a distance measure.

However, not all distance measures  $D$  give an euclidean distance matrix  $\mathbf{D}$ . Here is a counter example from [Gower and Legendre, 1986] :

$$\mathbf{D} = \begin{matrix} & \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 \\ \mathbf{x}_1 & 0 & 2 & 2 & 1.1 \\ \mathbf{x}_2 & 2 & 0 & 2 & 1.1 \\ \mathbf{x}_3 & 2 & 2 & 0 & 1.1 \\ \mathbf{x}_4 & 1.1 & 1.1 & 1.1 & 0 \end{matrix}$$

- All triples satisfy the triangle inequality

## Metric vs euclidean

## Theorem.

If  $\mathbf{D}$  is euclidean then  $D$  is a distance measure.

However, not all distance measures  $D$  give an euclidean distance matrix  $\mathbf{D}$ . Here is a counter example from [Gower and Legendre, 1986] :

$$\mathbf{D} = \begin{matrix} & \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 \\ \mathbf{x}_1 & 0 & 2 & 2 & 1.1 \\ \mathbf{x}_2 & 2 & 0 & 2 & 1.1 \\ \mathbf{x}_3 & 2 & 2 & 0 & 1.1 \\ \mathbf{x}_4 & 1.1 & 1.1 & 1.1 & 0 \end{matrix}$$

- All triples satisfy the triangle inequality
- $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$  form an equilateral triangle of length 2

## Metric vs euclidean

## Theorem.

If  $\mathbf{D}$  is euclidean then  $D$  is a distance measure.

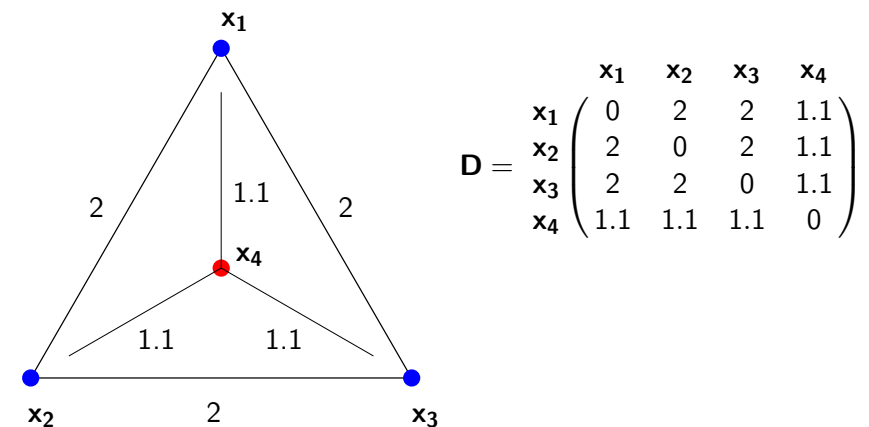
However, not all distance measures  $D$  give an euclidean distance matrix  $\mathbf{D}$ . Here is a counter example from [Gower and Legendre, 1986] :

$$\mathbf{D} = \begin{matrix} & \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 \\ \mathbf{x}_1 & 0 & 2 & 2 & 1.1 \\ \mathbf{x}_2 & 2 & 0 & 2 & 1.1 \\ \mathbf{x}_3 & 2 & 2 & 0 & 1.1 \\ \mathbf{x}_4 & 1.1 & 1.1 & 1.1 & 0 \end{matrix}$$

- All triples satisfy the triangle inequality
- $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$  form an equilateral triangle of length 2
- $\mathbf{x}_4$  is equidistant to all former data points with length 1.1

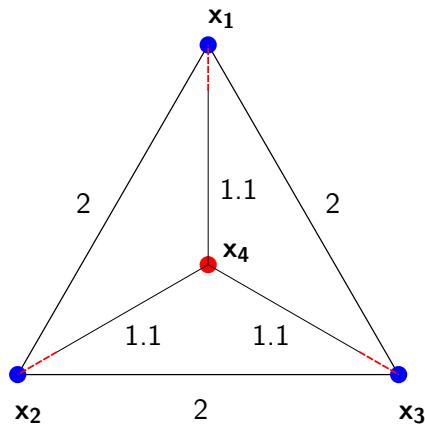
## Metric vs euclidean distance (cont'd)

It is impossible to represent the data points in an euclidean space.



## Metric vs euclidean distance (cont'd)

It is impossible to represent the data points in an euclidean space.



$$D = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{matrix} & \begin{pmatrix} 0 & 2 & 2 & 1.1 \\ 2 & 0 & 2 & 1.1 \\ 2 & 2 & 0 & 1.1 \\ 1.1 & 1.1 & 1.1 & 0 \end{pmatrix} \end{matrix}$$

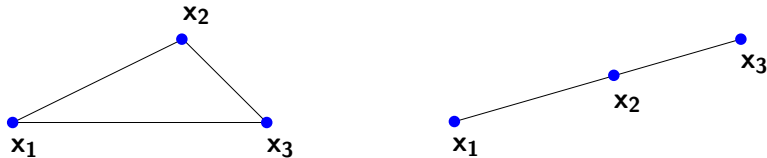
In an euclidean space,  $D(x_4, x_i)$  with  $i = 1, \dots, 3$  should have been **1.15**.

## Metric vs euclidean distance (cont'd)

It is not mandatory for  $D$  to be euclidean but it is often required  $D$  to satisfy the triangle inequality for all triples. Rationale : when comparing three data points we often represent them in an (local) euclidean space.

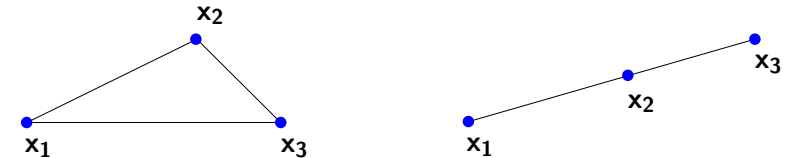
## Metric vs euclidean distance (cont'd)

It is not mandatory for  $D$  to be euclidean but it is often required  $D$  to satisfy the triangle inequality for all triples. Rationale : when comparing three data points we often represent them in an (local) euclidean space.



## Metric vs euclidean distance (cont'd)

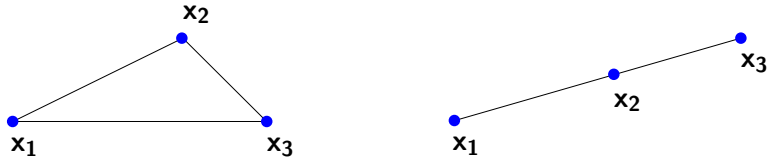
It is not mandatory for  $D$  to be euclidean but it is often required  $D$  to satisfy the triangle inequality for all triples. Rationale : when comparing three data points we often represent them in an (local) euclidean space.



$$D(x_1, x_3) \leq D(x_1, x_2) + D(x_2, x_3)$$

## Metric vs euclidean distance (cont'd)

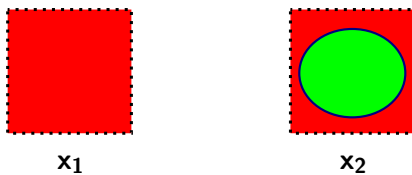
It is not mandatory for  $\mathbf{D}$  to be euclidean but it is often required  $\mathbf{D}$  to satisfy the triangle inequality for all triples. Rationale : when comparing three data points we often represent them in an (local) euclidean space.



$$D(\mathbf{x}_1, \mathbf{x}_3) \leq D(\mathbf{x}_1, \mathbf{x}_2) + D(\mathbf{x}_2, \mathbf{x}_3) \quad D(\mathbf{x}_1, \mathbf{x}_3) = D(\mathbf{x}_1, \mathbf{x}_2) + D(\mathbf{x}_2, \mathbf{x}_3)$$

## On triangle inequality

However there are cases where the triangle inequality is not a good condition [Tversky, 1977, Santini and Jain, 1999].



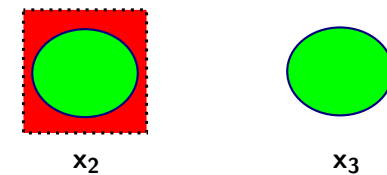
Here, we could have  $D(\mathbf{x}_1, \mathbf{x}_2) = 1$ ,

## On triangle inequality

However there are cases where the triangle inequality is not a good condition [Tversky, 1977, Santini and Jain, 1999].

## On triangle inequality

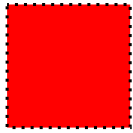
However there are cases where the triangle inequality is not a good condition [Tversky, 1977, Santini and Jain, 1999].



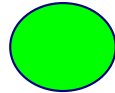
Here, we could have  $D(\mathbf{x}_1, \mathbf{x}_2) = 1$ ,  $D(\mathbf{x}_2, \mathbf{x}_3) = 1$ ,

## On triangle inequality

However there are cases where the triangle inequality is not a good condition [Tversky, 1977, Santini and Jain, 1999].



$x_1$

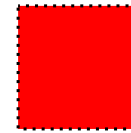


$x_3$

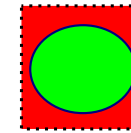
Here, we could have  $D(x_1, x_2) = 1$ ,  $D(x_2, x_3) = 1$ , and  $D(x_1, x_3) = 3$

## On triangle inequality

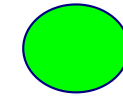
However there are cases where the triangle inequality is not a good condition [Tversky, 1977, Santini and Jain, 1999].



$x_1$



$x_2$



$x_3$

Here, we could have  $D(x_1, x_2) = 1$ ,  $D(x_2, x_3) = 1$ , and  $D(x_1, x_3) = 3$  such that  $D(x_1, x_3) \not\leq D(x_1, x_2) + D(x_2, x_3)$ .

## Definition of a similarity measure

No consensus on the axioms defining a similarity measure.

### Definition. (Similarity measure)

Let  $\mathbb{D}$  be a set of items represented in an euclidean space and let  $S : \mathbb{D} \times \mathbb{D} \rightarrow \mathbb{R}$  be a real function.  $S$  is a **similarity measure** if it satisfies the following properties :

- 1 *Boundary conditions* : there are two fix numbers  $a$  and  $b$  such that  $\forall \mathbf{x}, \mathbf{y} : a \leq S(\mathbf{x}, \mathbf{y}) \leq b$
- 2 *Symmetry* :  $\forall \mathbf{x}, \mathbf{y} : S(\mathbf{x}, \mathbf{y}) = S(\mathbf{y}, \mathbf{x})$
- 3 *Identity and indiscernability* :  $\forall \mathbf{x}, \mathbf{y} : \mathbf{x} = \mathbf{y} \Leftrightarrow S(\mathbf{x}, \mathbf{y}) = b$

The similarity measure  $S$  is said to be metric if the pairwise similarity matrix  $\mathbf{S}$  with  $S_{ij} = S(\mathbf{x}_i, \mathbf{x}_j)$  satisfies the following condition :

- 4 *Metric* :  $\mathbf{S}$  is positive semi-definite PSD (all eigenvalues are non negative)

Note that most of times, we have  $a = 0$  or  $a = -1$  and  $b = 1$ .

## Dissimilarities, Distances, similarities and metrics

### Theorem.

$\mathbf{D}$  is euclidean if the  $(n \times n)$  matrix  $\mathbf{W}$  of general term

$$W_{ij} = -\frac{1}{2} \left( D_{ij}^2 - \frac{D_{i.}^2}{n} - \frac{D_{.j}^2}{n} + \frac{D_{..}^2}{n^2} \right) \text{ is PSD (where } D_{.j}^2 = \sum_{i=1}^n D_{ij}^2 \text{)}$$

## Dissimilarities, Distances, similarities and metrics

## Theorem.

$\mathbf{D}$  is euclidean if the  $(n \times n)$  matrix  $\mathbf{W}$  of general term

$$\mathbf{W}_{ij} = -\frac{1}{2} \left( \mathbf{D}_{ij}^2 - \frac{\mathbf{D}_i^2}{n} - \frac{\mathbf{D}_j^2}{n} + \frac{\mathbf{D}^2}{n^2} \right) \text{ is PSD (where } \mathbf{D}_j^2 = \sum_{i=1}^n \mathbf{D}_{ij}^2 \text{)}.$$

Note that  $\mathbf{W} = (\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{n})\mathbf{\Delta}(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{n})$  with  $\mathbf{\Delta}_{ij} = -\frac{1}{2}\mathbf{D}_{ij}^2$ ,  $\mathbf{I}$  being the unit matrix and  $\mathbf{1}$  the vector full of 1.

## Dissimilarities, Distances, similarities and metrics

## Theorem.

$\mathbf{D}$  is euclidean if the  $(n \times n)$  matrix  $\mathbf{W}$  of general term

$$\mathbf{W}_{ij} = -\frac{1}{2} \left( \mathbf{D}_{ij}^2 - \frac{\mathbf{D}_i^2}{n} - \frac{\mathbf{D}_j^2}{n} + \frac{\mathbf{D}^2}{n^2} \right) \text{ is PSD (where } \mathbf{D}_j^2 = \sum_{i=1}^n \mathbf{D}_{ij}^2 \text{)}.$$

Note that  $\mathbf{W} = (\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{n})\mathbf{\Delta}(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{n})$  with  $\mathbf{\Delta}_{ij} = -\frac{1}{2}\mathbf{D}_{ij}^2$ ,  $\mathbf{I}$  being the unit matrix and  $\mathbf{1}$  the vector full of 1.

## Corollary.

If  $\mathbf{D}$  is euclidean then  $\mathbf{W}$  is metric and it can be interpreted as a pairwise dot product matrix.

## Dissimilarities, Distances, similarities and metrics

## Theorem.

$\mathbf{D}$  is euclidean if the  $(n \times n)$  matrix  $\mathbf{W}$  of general term

$$\mathbf{W}_{ij} = -\frac{1}{2} \left( \mathbf{D}_{ij}^2 - \frac{\mathbf{D}_i^2}{n} - \frac{\mathbf{D}_j^2}{n} + \frac{\mathbf{D}^2}{n^2} \right) \text{ is PSD (where } \mathbf{D}_j^2 = \sum_{i=1}^n \mathbf{D}_{ij}^2 \text{)}.$$

Note that  $\mathbf{W} = (\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{n})\mathbf{\Delta}(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{n})$  with  $\mathbf{\Delta}_{ij} = -\frac{1}{2}\mathbf{D}_{ij}^2$ ,  $\mathbf{I}$  being the unit matrix and  $\mathbf{1}$  the vector full of 1.

## Corollary.

If  $\mathbf{D}$  is euclidean then  $\mathbf{W}$  is metric and it can be interpreted as a pairwise dot product matrix.

**Exercise 1 :** Using R, show that the distance measure  $\mathbf{D}$  of the previous counter example in slide 2 is not euclidean.

## Dissimilarities, Distances, similarities and metrics (cont'd)

## Theorem.

If a similarity matrix  $\mathbf{S}$  is PSD with elements  $0 \leq \mathbf{S}_{ij} \leq 1$  and  $\mathbf{S}_{ii} = 1$ , then the dissimilarity matrix  $\mathbf{D}$  of general term  $\mathbf{D}_{ij} = \sqrt{1 - \mathbf{S}_{ij}}$  is euclidean.

## Dissimilarities, Distances, similarities and metrics (cont'd)

## Theorem.

If a similarity matrix  $\mathbf{S}$  is PSD with elements  $0 \leq \mathbf{S}_{ij} \leq 1$  and  $\mathbf{S}_{ii} = 1$ , then the dissimilarity matrix  $\mathbf{D}$  of general term  $\mathbf{D}_{ij} = \sqrt{1 - \mathbf{S}_{ij}}$  is euclidean.

## Corollary.

If the pairwise matrix of general term  $\sqrt{1 - \mathbf{S}_{ij}}$  is not euclidean then  $\mathbf{S}$  is not PSD.

## Dissimilarities, Distances, similarities and metrics (cont'd)

## Theorem.

If a similarity matrix  $\mathbf{S}$  is PSD with elements  $0 \leq \mathbf{S}_{ij} \leq 1$  and  $\mathbf{S}_{ii} = 1$ , then the dissimilarity matrix  $\mathbf{D}$  of general term  $\mathbf{D}_{ij} = \sqrt{1 - \mathbf{S}_{ij}}$  is euclidean.

## Corollary.

If the pairwise matrix of general term  $\sqrt{1 - \mathbf{S}_{ij}}$  is not euclidean then  $\mathbf{S}$  is not PSD.

## Theorem.

If  $\mathbf{D}$  is a dissimilarity matrix then there exists a constant  $h$  such that the matrix with general term  $\sqrt{\mathbf{D}_{ij}^2 + h}, \forall i \neq j$ , is euclidean.

In that case,  $h \geq -2\lambda_n$  where  $\lambda_n$  is the smallest eigenvalue of  $\mathbf{W} = (\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{n})\mathbf{\Delta}(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{n})$ .

## Dissimilarities, Distances, similarities and metrics (cont'd)

## Theorem.

If a similarity matrix  $\mathbf{S}$  is PSD with elements  $0 \leq \mathbf{S}_{ij} \leq 1$  and  $\mathbf{S}_{ii} = 1$ , then the dissimilarity matrix  $\mathbf{D}$  of general term  $\mathbf{D}_{ij} = \sqrt{1 - \mathbf{S}_{ij}}$  is euclidean.

## Corollary.

If the pairwise matrix of general term  $\sqrt{1 - \mathbf{S}_{ij}}$  is not euclidean then  $\mathbf{S}$  is not PSD.

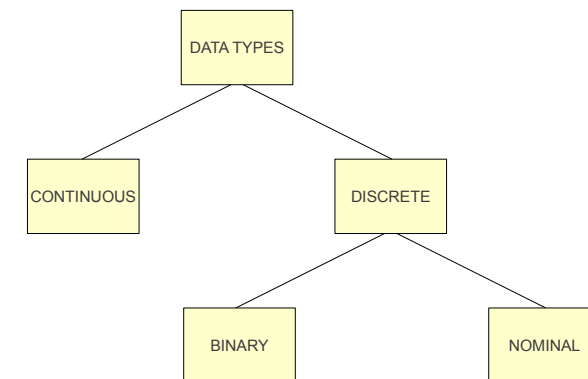
## Theorem.

If  $\mathbf{D}$  is a dissimilarity matrix then there exists a constant  $h$  such that the matrix with general term  $\sqrt{\mathbf{D}_{ij}^2 + h}, \forall i \neq j$ , is euclidean.

In that case,  $h \geq -2\lambda_n$  where  $\lambda_n$  is the smallest eigenvalue of  $\mathbf{W} = (\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{n})\mathbf{\Delta}(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{n})$ .

**Exercise 2 :** Using R, show how to transform the distance measure  $\mathbf{D}$  of the previous counter example in slide 2 so that it becomes euclidean.

## Different types of numerical data



Different types of numerical data hence different kinds of proximity measures.

## Introduction

- The data points are represented in  $\mathbb{R}^p$
- Each dimension of  $\mathbb{R}^p$  is a feature
- The data table  $\mathbf{X}$  is such that  $\mathbf{x}_i$  is a vector of  $\mathbb{R}^p$ .

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \dots & \dots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix}$$

## Data normalization

In general, **raw data** contain features which are different measurements in different scales (eg cm, kg, Euros, ...). In that case, any proximity measure can be biased.

## Example

```
> install.packages("datasets")
> library(datasets)
> data(iris)
> print(iris[,-5])
      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
...

```

## Data normalization

In general, **raw data** contain features which are different measurements in different scales (eg cm, kg, Euros, ...). In that case, any proximity measure can be biased.

Before applying any proximity measure or any clustering algorithm, one should normalize the data. Let denote  $\mathbf{X}^*$  the raw data set :

$$\mathbf{X}^* = \begin{pmatrix} x_{11}^* & x_{12}^* & \dots & x_{1p}^* \\ x_{21}^* & x_{22}^* & \dots & x_{2p}^* \\ \vdots & \dots & \dots & \vdots \\ x_{n1}^* & x_{n2}^* & \dots & x_{np}^* \end{pmatrix}$$



## Data normalization

In general, **raw data** contain features which are different measurements in different scales (eg cm, kg, Euros, ...). In that case, any proximity measure can be biased.

Before applying any proximity measure or any clustering algorithm, one should normalize the data. Let denote  $\mathbf{X}^*$  the raw data set :

$$\mathbf{X}^* = \begin{pmatrix} x_{11}^* & x_{12}^* & \dots & x_{1p}^* \\ x_{21}^* & x_{22}^* & \dots & x_{2p}^* \\ \vdots & \dots & \dots & \vdots \\ x_{n1}^* & x_{n2}^* & \dots & x_{np}^* \end{pmatrix}$$

To **normalize raw data**, we can subtract a location measure and divide a scale measure for each feature  $j$  :

$$x_{ij} = \frac{x_{ij}^* - L_j^*}{M_j^*}$$

## Data normalization (cont'd)

Let us denote for each feature  $j$  its :

- mean average :  $\mu_j^* = \frac{1}{n} \sum_{i=1}^n x_{ij}^*$
- standard deviation :  $\sigma_j^* = \left( \frac{1}{n-1} \sum_{i=1}^n (x_{ij}^* - \mu_j^*)^2 \right)^{1/2}$
- range :  $r_j^* = \max_i \{x_{ij}^*\} - \min_i \{x_{ij}^*\}$

The two most used data normalization using the previous equation are :

## Data normalization (cont'd)

Let us denote for each feature  $j$  its :

- mean average :  $\mu_j^* = \frac{1}{n} \sum_{i=1}^n x_{ij}^*$
- standard deviation :  $\sigma_j^* = \left( \frac{1}{n-1} \sum_{i=1}^n (x_{ij}^* - \mu_j^*)^2 \right)^{1/2}$
- range :  $r_j^* = \max_i \{x_{ij}^*\} - \min_i \{x_{ij}^*\}$

## Data normalization (cont'd)

Let us denote for each feature  $j$  its :

- mean average :  $\mu_j^* = \frac{1}{n} \sum_{i=1}^n x_{ij}^*$
- standard deviation :  $\sigma_j^* = \left( \frac{1}{n-1} \sum_{i=1}^n (x_{ij}^* - \mu_j^*)^2 \right)^{1/2}$
- range :  $r_j^* = \max_i \{x_{ij}^*\} - \min_i \{x_{ij}^*\}$

The two most used data normalization using the previous equation are :

- **z-score** :  $x_{ij} = \frac{x_{ij}^* - \mu_j^*}{\sigma_j^*}$  ;  $\forall j : \mu_j = 0$  and  $\sigma_j = 1$ .

## Data normalization (cont'd)

Let us denote for each feature  $j$  its :

- mean average :  $\mu_j^* = \frac{1}{n} \sum_{i=1}^n x_{ij}^*$
- standard deviation :  $\sigma_j^* = \left( \frac{1}{n-1} \sum_{i=1}^n (x_{ij}^* - \mu_j^*)^2 \right)^{1/2}$
- range :  $r_j^* = \max_i \{x_{ij}^*\} - \min_i \{x_{ij}^*\}$

The two most used data normalization using the previous equation are :

- **z-score** :  $x_{ij} = \frac{x_{ij}^* - \mu_j^*}{\sigma_j^*}$  ;  $\forall j : \mu_j = 0$  and  $\sigma_j = 1$ .
- **range** :  $x_{ij} = \frac{x_{ij}^* - \min_i \{x_{ij}^*\}}{r_j^*}$  ;  $\forall j : \mu_j = \frac{\mu_j^* - \min_i \{x_{ij}^*\}}{r_j^*}$  and  $\sigma_j = \frac{\sigma_j^*}{r_j^*}$ .

This method is sensitive to outliers.

## Data normalization (cont'd)

Another approach to **normalize raw data** is to transform, for each feature  $j$ , the measurements into ranks. Let us denote  $\tau$  the permutation of  $n$  elements such that the sequence  $\{x_{\tau(1)j}^*, x_{\tau(2)j}^*, \dots, x_{\tau(n)j}^*\}$  is the measurements of the attribute  $j$  sorted in increasing order.

## Data normalization (cont'd)

Some authors have defined other types of normalization methods that fall into the following approach :

$$x_{ij} = \frac{x_{ij}^* - L_j^*}{M_j^*}$$

See [Milligan and Cooper, 1988] for eg, for more methods in that context.

## Data normalization (cont'd)

Another approach to **normalize raw data** is to transform, for each feature  $j$ , the measurements into ranks. Let us denote  $\tau$  the permutation of  $n$  elements such that the sequence  $\{x_{\tau(1)j}^*, x_{\tau(2)j}^*, \dots, x_{\tau(n)j}^*\}$  is the measurements of the attribute  $j$  sorted in increasing order.

Then the **rank** normalization is given as follows :

$$x_{ij} = k \text{ if } i = \tau(k)$$

## Data normalization (cont'd)

Another approach to **normalize raw data** is to transform, for each feature  $j$ , the measurements into ranks. Let us denote  $\tau$  the permutation of  $n$  elements such that the sequence  $\{x_{\tau(1)j}^*, x_{\tau(2)j}^*, \dots, x_{\tau(n)j}^*\}$  is the measurements of the attribute  $j$  sorted in increasing order.

Then the **rank** normalization is given as follows :

$$x_{ij} = k \text{ if } i = \tau(k)$$

$$\forall j : \mu_j = \frac{n+1}{2} \text{ and } \sigma_j^2 = (n+1) \left( \frac{2n+1}{6} - \frac{n+1}{4} \right).$$

## Example

```
> install.packages("clusterSim")
> library(clusterSim)
> iris.normalization=data.Normalization(iris[,-5],type="n1")
> print(iris.normalization)
  Sepal.Length Sepal.Width Petal.Length  Petal.Width
1    -0.89767388  1.01560199  -1.33575163 -1.3110521482
2    -1.13920048 -0.13153881  -1.33575163 -1.3110521482
3    -1.38072709  0.32731751  -1.39239929 -1.3110521482
4    -1.50149039  0.09788935  -1.27910398 -1.3110521482
5    -1.01843718  1.24503015  -1.33575163 -1.3110521482
...
> mean(iris.normalization)
  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
-4.484318e-16  2.034094e-16 -2.895326e-17 -2.989362e-17
> sd(iris.normalization)
  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
1              1              1              1              1
```

**Exercise 3 :** Write an R function that takes as an input a raw data table and performs the rank normalization.

## Data normalization (cont'd)

Another approach to **normalize raw data** is to transform, for each feature  $j$ , the measurements into ranks. Let us denote  $\tau$  the permutation of  $n$  elements such that the sequence  $\{x_{\tau(1)j}^*, x_{\tau(2)j}^*, \dots, x_{\tau(n)j}^*\}$  is the measurements of the attribute  $j$  sorted in increasing order.

Then the **rank** normalization is given as follows :

$$x_{ij} = k \text{ if } i = \tau(k)$$

$$\forall j : \mu_j = \frac{n+1}{2} \text{ and } \sigma_j^2 = (n+1) \left( \frac{2n+1}{6} - \frac{n+1}{4} \right).$$

Unlike the range standardization, the rank standardization reduces the impact of outliers.

## Classic distance measures

Below is a non exhaustive list of distance measures between two data points  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$  :

## Classic distance measures

Below is a non exhaustive list of distance measures between two data points  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$  :

- **Euclidean distance** :

$$\begin{aligned} D_{eucl}(\mathbf{x}, \mathbf{y}) &= \sqrt{\sum_{j=1}^p (x_j - y_j)^2} \\ &= \sqrt{(\mathbf{x} - \mathbf{y})^\top (\mathbf{x} - \mathbf{y})} \end{aligned}$$

## Classic distance measures

Below is a non exhaustive list of distance measures between two data points  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$  :

- **Euclidean distance** :

$$\begin{aligned} D_{eucl}(\mathbf{x}, \mathbf{y}) &= \sqrt{\sum_{j=1}^p (x_j - y_j)^2} \\ &= \sqrt{(\mathbf{x} - \mathbf{y})^\top (\mathbf{x} - \mathbf{y})} \end{aligned}$$

- **Manhattan distance or “city block” distance** :

$$D_{manh}(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^p |x_j - y_j|$$

- **Maximal distance** :

$$D_{max}(\mathbf{x}, \mathbf{y}) = \max_{1 \leq j \leq p} \{|x_j - y_j|\}$$

## Classic distance measures

Below is a non exhaustive list of distance measures between two data points  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$  :

- **Euclidean distance** :

$$\begin{aligned} D_{eucl}(\mathbf{x}, \mathbf{y}) &= \sqrt{\sum_{j=1}^p (x_j - y_j)^2} \\ &= \sqrt{(\mathbf{x} - \mathbf{y})^\top (\mathbf{x} - \mathbf{y})} \end{aligned}$$

- **Manhattan distance or “city block” distance** :

$$D_{manh}(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^p |x_j - y_j|$$

## Minkowski distance measures

The Euclidean distance, Manhattan distance, and maximum distance are three particular cases of the **Minkowski distance** defined by :

$$D_{mink}(\mathbf{x}, \mathbf{y}) = \left( \sum_{j=1}^p |x_j - y_j|^r \right)^{1/r}$$

## Minkowski distance measures

The Euclidean distance, Manhattan distance, and maximum distance are three particular cases of the **Minkowski distance** defined by :

$$D_{mink}(\mathbf{x}, \mathbf{y}) = \left( \sum_{j=1}^p |x_j - y_j|^r \right)^{1/r}$$

Previous cases are given by :

- Euclidean distance :  $r = 2$
- Manhattan distance :  $r = 1$
- Max distance :  $r = +\infty$

## Mahalanobis distance measure

Let  $\Sigma$  be the  $(p \times p)$  **covariance matrix** with general term

$$\Sigma_{jj'} = \frac{1}{n} \sum_{i=1}^n (x_{ij} - \mu_j)(x_{ij'} - \mu_{j'}).$$

Note that :

$$\Sigma = \frac{1}{n} (\mathbf{X} - \mathbf{M})^\top (\mathbf{X} - \mathbf{M})$$

with  $\mathbf{M}$  being the  $(n \times p)$  matrix whose column  $j$  is  $\mu_j \mathbf{1}$ .

The **Mahalanobis distance** is then defined as follows :

$$D_{maha}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^\top \Sigma^{-1} (\mathbf{x} - \mathbf{y})}$$

## Mahalanobis distance measure

Let  $\Sigma$  be the  $(p \times p)$  **covariance matrix** with general term

$$\Sigma_{jj'} = \frac{1}{n} \sum_{i=1}^n (x_{ij} - \mu_j)(x_{ij'} - \mu_{j'}).$$

Note that :

$$\Sigma = \frac{1}{n} (\mathbf{X} - \mathbf{M})^\top (\mathbf{X} - \mathbf{M})$$

with  $\mathbf{M}$  being the  $(n \times p)$  matrix whose column  $j$  is  $\mu_j \mathbf{1}$ .

## Mahalanobis distance measure

Let  $\Sigma$  be the  $(p \times p)$  **covariance matrix** with general term

$$\Sigma_{jj'} = \frac{1}{n} \sum_{i=1}^n (x_{ij} - \mu_j)(x_{ij'} - \mu_{j'}).$$

Note that :

$$\Sigma = \frac{1}{n} (\mathbf{X} - \mathbf{M})^\top (\mathbf{X} - \mathbf{M})$$

with  $\mathbf{M}$  being the  $(n \times p)$  matrix whose column  $j$  is  $\mu_j \mathbf{1}$ .

The **Mahalanobis distance** is then defined as follows :

$$D_{maha}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^\top \Sigma^{-1} (\mathbf{x} - \mathbf{y})}$$

Properties :

- $D_{maha}$  applies a weighting scheme to the data. It can alleviate some distortions caused by existing linear dependences between variables.

## Mahalanobis distance measure

Let  $\Sigma$  be the  $(p \times p)$  **covariance matrix** with general term

$$\Sigma_{jj'} = \frac{1}{n} \sum_{i=1}^n (x_{ij} - \mu_j)(x_{ij'} - \mu_{j'}).$$

Note that :

$$\Sigma = \frac{1}{n} (\mathbf{X} - \mathbf{M})^\top (\mathbf{X} - \mathbf{M})$$

with  $\mathbf{M}$  being the  $(n \times p)$  matrix whose column  $j$  is  $\mu_j \mathbf{1}$ .

The **Mahalanobis distance** is then defined as follows :

$$D_{maha}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^\top \Sigma^{-1} (\mathbf{x} - \mathbf{y})}$$

Properties :

- $D_{maha}$  applies a weighting scheme to the data. It can alleviate some distortions caused by existing linear dependences between variables.
- $D_{maha}$  is invariant to any nonsingular transformation of  $\mathbf{X}$ . If  $\mathbf{x}' = \mathbf{C}\mathbf{x}$ ;  $\forall \mathbf{x} \in \mathbb{D}$  with  $\mathbf{C}$  being invertible then  $D_{maha}(\mathbf{x}', \mathbf{y}') = D_{maha}(\mathbf{x}, \mathbf{y})$  for all pairs  $(\mathbf{x}, \mathbf{y})$ . **Exercise 4** : Show it.

## Euclidean distances and dot products

Let  $\langle \cdot, \cdot \rangle$  denote the dot product of the euclidean space  $\mathbb{R}^p$  span by the set of features. We have :

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{j=1}^p x_j y_j = \mathbf{x}^\top \mathbf{y}$$

## Euclidean distances and dot products

Let  $\langle \cdot, \cdot \rangle$  denote the dot product of the euclidean space  $\mathbb{R}^p$  span by the set of features. We have :

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{j=1}^p x_j y_j = \mathbf{x}^\top \mathbf{y}$$

We also recall that :

- The norm of  $\mathbf{x}$  is denoted  $\|\mathbf{x}\|$  and is given by :

$$\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$$

- The euclidean distance  $D_{euc}(\mathbf{x}, \mathbf{y})$  is the same as  $\|\mathbf{x} - \mathbf{y}\|$  and thus :

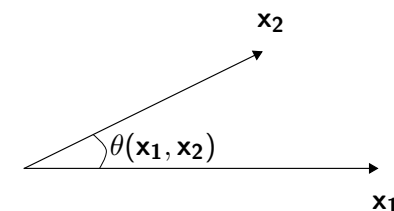
$$\begin{aligned} \|\mathbf{x} - \mathbf{y}\| &= \sqrt{\langle \mathbf{x} - \mathbf{y}, \mathbf{x} - \mathbf{y} \rangle} \\ &= \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle + \langle \mathbf{y}, \mathbf{y} \rangle - 2\langle \mathbf{x}, \mathbf{y} \rangle} \\ &= \sqrt{\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - 2\langle \mathbf{x}, \mathbf{y} \rangle} \end{aligned}$$

## Cosine proximity measures

The cosine or angular proximity measure is one of the most used proximity measure. It is defined as follows :

$$\begin{aligned} S_{\cos}(\mathbf{x}, \mathbf{y}) &= \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\|\mathbf{x}\| \|\mathbf{y}\|} \\ &= \left\langle \frac{\mathbf{x}}{\|\mathbf{x}\|}, \frac{\mathbf{y}}{\|\mathbf{y}\|} \right\rangle \\ &= \cos(\theta(\mathbf{x}, \mathbf{y})) \end{aligned}$$

where  $\theta(\mathbf{x}, \mathbf{y})$  is the angle between the two vectors.



## Cosine proximity measures (cont'd)

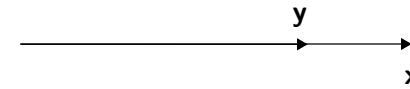
Properties :

- 1 Boundary conditions : there are two fix numbers  $a$  and  $b$  such that  $\forall \mathbf{x}, \mathbf{y} : -1 \leq S_{\cos}(\mathbf{x}, \mathbf{y}) \leq 1$
- 2 Symmetry :  $\forall \mathbf{x}, \mathbf{y} : S_{\cos}(\mathbf{x}, \mathbf{y}) = S_{\cos}(\mathbf{y}, \mathbf{x})$
- 3 Indiscernability of identicals :  $\forall \mathbf{x}, \mathbf{y} : \mathbf{x} = \mathbf{y} \Rightarrow S_{\cos}(\mathbf{x}, \mathbf{y}) = 1$   
**BUT** no identity of "indiscernibles" :  $S_{\cos}(\mathbf{x}, \mathbf{y}) = 1 \not\Rightarrow \mathbf{x} = \mathbf{y}$

## Cosine proximity measures (cont'd)

Properties :

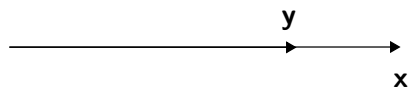
- 1 Boundary conditions : there are two fix numbers  $a$  and  $b$  such that  $\forall \mathbf{x}, \mathbf{y} : -1 \leq S_{\cos}(\mathbf{x}, \mathbf{y}) \leq 1$
- 2 Symmetry :  $\forall \mathbf{x}, \mathbf{y} : S_{\cos}(\mathbf{x}, \mathbf{y}) = S_{\cos}(\mathbf{y}, \mathbf{x})$
- 3 Indiscernability of identicals :  $\forall \mathbf{x}, \mathbf{y} : \mathbf{x} = \mathbf{y} \Rightarrow S_{\cos}(\mathbf{x}, \mathbf{y}) = 1$   
**BUT** no identity of "indiscernibles" :  $S_{\cos}(\mathbf{x}, \mathbf{y}) = 1 \not\Rightarrow \mathbf{x} = \mathbf{y}$

Counter examples : collinear vectors  $\mathbf{y} = \alpha \mathbf{x}$  with  $\alpha > 0$ .

## Cosine proximity measures (cont'd)

Properties :

- 1 Boundary conditions : there are two fix numbers  $a$  and  $b$  such that  $\forall \mathbf{x}, \mathbf{y} : -1 \leq S_{\cos}(\mathbf{x}, \mathbf{y}) \leq 1$
- 2 Symmetry :  $\forall \mathbf{x}, \mathbf{y} : S_{\cos}(\mathbf{x}, \mathbf{y}) = S_{\cos}(\mathbf{y}, \mathbf{x})$
- 3 Indiscernability of identicals :  $\forall \mathbf{x}, \mathbf{y} : \mathbf{x} = \mathbf{y} \Rightarrow S_{\cos}(\mathbf{x}, \mathbf{y}) = 1$   
**BUT** no identity of "indiscernibles" :  $S_{\cos}(\mathbf{x}, \mathbf{y}) = 1 \not\Rightarrow \mathbf{x} = \mathbf{y}$

Counter examples : collinear vectors  $\mathbf{y} = \alpha \mathbf{x}$  with  $\alpha > 0$ .

- 4 Metric :  $\mathbf{S}_{\cos}$  is PSD (Gram matrix)

## Recalling Gram matrices

**Definition.**

Let  $\mathbb{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  a set of vectors belonging to an euclidean space with dot product  $\langle \cdot, \cdot \rangle$ . Then the  $(n \times n)$  matrix  $\mathbf{G}$  of general term  $\mathbf{G}_{ij} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$  is called a Gram matrix.

## Recalling Gram matrices

### Definition.

Let  $\mathbb{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  a set of vectors belonging to an euclidean space with dot product  $\langle \cdot, \cdot \rangle$ . Then the  $(n \times n)$  matrix  $\mathbf{G}$  of general term  $\mathbf{G}_{ij} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$  is called a Gram matrix.

### Theorem.

Any Gram matrix is PSD. Any PSD matrix is the Gram matrix for some set of vectors.

## Introduction

- The data points are described by a set of  $q$  categorical variables each of them having  $p_j$  categories (or states or classes or labels).
- Given a categorical feature, its categories can be :
  - ▶ on an ordinal scale (eg taste : good, medium, bad)
  - ▶ on a nominal scale (eg color : blue, brown, green)

## Introduction

- The data points are described by a set of  $q$  categorical variables each of them having  $p_j$  categories (or states or classes or labels).

## Introduction

- The data points are described by a set of  $q$  categorical variables each of them having  $p_j$  categories (or states or classes or labels).
- Given a categorical feature, its categories can be :
  - ▶ on an ordinal scale (eg taste : good, medium, bad)
  - ▶ on a nominal scale (eg color : blue, brown, green)
- Given a categorical feature, its categories can also be :
  - ▶ symmetric : they have all the same importance
  - ▶ asymmetric : some of them have more importance than others



## Introduction

- The data points are described by a set of  $q$  categorical variables each of them having  $p_j$  categories (or states or classes or labels).
- Given a categorical feature, its categories can be :
  - ▶ on an ordinal scale (eg taste : good, medium, bad)
  - ▶ on a nominal scale (eg color : blue, brown, green)
- Given a categorical feature, its categories can also be :
  - ▶ symmetric : they have all the same importance
  - ▶ asymmetric : some of them have more importance than others
- We assume that the data points are represented in the space spanned by  $\{0, 1\}^p$  with  $p = \sum_{j=1}^q p_j$ .

## Introduction

- The data points are described by a set of  $q$  categorical variables each of them having  $p_j$  categories (or states or classes or labels).
- Given a categorical feature, its categories can be :
  - ▶ on an ordinal scale (eg taste : good, medium, bad)
  - ▶ on a nominal scale (eg color : blue, brown, green)
- Given a categorical feature, its categories can also be :
  - ▶ symmetric : they have all the same importance
  - ▶ asymmetric : some of them have more importance than others
- We assume that the data points are represented in the space spanned by  $\{0, 1\}^p$  with  $p = \sum_{j=1}^q p_j$ .
- Each dimension of  $\{0, 1\}^p$  corresponds to a state of a categorical variable and 0 and 1 respectively means the absence xor the presence of the latter
- Ordinal scales will be treated similarly as nominal scales

## Introduction

- The data points are described by a set of  $q$  categorical variables each of them having  $p_j$  categories (or states or classes or labels).
- Given a categorical feature, its categories can be :
  - ▶ on an ordinal scale (eg taste : good, medium, bad)
  - ▶ on a nominal scale (eg color : blue, brown, green)
- Given a categorical feature, its categories can also be :
  - ▶ symmetric : they have all the same importance
  - ▶ asymmetric : some of them have more importance than others
- We assume that the data points are represented in the space spanned by  $\{0, 1\}^p$  with  $p = \sum_{j=1}^q p_j$ .
- Each dimension of  $\{0, 1\}^p$  corresponds to a state of a categorical variable and 0 and 1 respectively means the absence xor the presence of the latter

## Introduction

- The data points are described by a set of  $q$  categorical variables each of them having  $p_j$  categories (or states or classes or labels).
- Given a categorical feature, its categories can be :
  - ▶ on an ordinal scale (eg taste : good, medium, bad)
  - ▶ on a nominal scale (eg color : blue, brown, green)
- Given a categorical feature, its categories can also be :
  - ▶ symmetric : they have all the same importance
  - ▶ asymmetric : some of them have more importance than others
- We assume that the data points are represented in the space spanned by  $\{0, 1\}^p$  with  $p = \sum_{j=1}^q p_j$ .
- Each dimension of  $\{0, 1\}^p$  corresponds to a state of a categorical variable and 0 and 1 respectively means the absence xor the presence of the latter
- Ordinal scales will be treated similarly as nominal scales
- If a category is not important than its dimension is removed

## Introduction

- The data points are described by a set of  $q$  categorical variables each of them having  $p_j$  categories (or states or classes or labels).
- Given a categorical feature, its categories can be :
  - ▶ on an ordinal scale (eg taste : good, medium, bad)
  - ▶ on a nominal scale (eg color : blue, brown, green)
- Given a categorical feature, its categories can also be :
  - ▶ symmetric : they have all the same importance
  - ▶ asymmetric : some of them have more importance than others
- We assume that the data points are represented in the space spanned by  $\{0, 1\}^p$  with  $p = \sum_{j=1}^q p_j$ .
- Each dimension of  $\{0, 1\}^p$  corresponds to a state of a categorical variable and 0 and 1 respectively means the absence xor the presence of the latter
- Ordinal scales will be treated similarly as nominal scales
- If a category is not important than its dimension is removed
- The data table  $\mathbf{X}$  is such that  $\mathbf{x}_i$  is a binary vector we thus talk about **binary data**

## Another example : text data

```
> poison.temp=data.frame(name=rownames(poison.text),textual=poison.text[,3])
> print(poison.temp)
      name          textual
1 Samantha  Nausea Abdominals Fever Diarrhea Potato Fish Mayo Courgette Cheese Icecream
2 Sarah      Potato Fish Mayo Courgette Icecream
3 Barbara    Vomitting Abdominals Fever Diarrhea Potato Fish Mayo Courgette Cheese Icecream
4 Acha       Potato Fish Courgette Cheese Icecream
5 Zacharias  Vomitting Abdominals Fever Diarrhea Potato Fish Mayo Courgette Cheese Icecream
6 Dalen      Abdominals Fever Diarrhea Potato Mayo Courgette Cheese Icecream
...
> poison.text.bin=textual(tab=poison.temp,num.text=2,contingence.by=c(1))
> print(poison.text.bin)
$cont.table
      abdominals cheese courgette diarrhea fever fish icecream mayo nausea potato vomitting
Acha           0         1         1         0         0         1         1         0         0         1         0
Adela          1         1         1         1         1         1         0         1         0         1         0
Alexandra      0         0         0         0         0         1         1         1         0         1         0
Alison         0         0         1         0         0         1         0         0         0         1         0
Alvis          1         0         1         1         1         1         1         1         0         1         1
Andre          0         1         1         1         1         1         1         1         0         1         1
...
```

## Example

```
> install.packages("FactoMineR")
> library(FactoMineR)
> data(poison.text)
> poison.bin=poison.text[, -3]
> print(poison.bin)
      Sick Sex
Samantha sick F
Sarah     healthy F
Barbara  sick F
Acha     healthy F
Zacharias sick M
Dalen    sick M
...
> poison.temp=poison.text[, -3]
> poison.bin=matrix(0,nrow=nrow(poison.temp),ncol=2*ncol(poison.temp))
> poison.bin[,1]=as.vector(poison.temp[,1]=="sick")
> poison.bin[,2]=as.vector(poison.temp[,1]=="healthy")
> poison.bin[,3]=as.vector(poison.temp[,2]=="M")
> poison.bin[,4]=as.vector(poison.temp[,2]=="F")
> print(poison.bin)
      [,1] [,2] [,3] [,4]
[1,]  1   0   0   1
[2,]  0   1   0   1
[3,]  1   0   0   1
[4,]  0   1   0   1
[5,]  1   0   1   0
[6,]  1   0   1   0
...
```

## Preliminaries

Given two binary vectors  $\mathbf{x}, \mathbf{y}$  in  $\{0, 1\}^p$ , we can introduce the following  $(2 \times 2)$  contingency table :

		$y_j$		Total
		1	0	
$x_j$	1	$a$	$b$	$a + b$
	0	$c$	$d$	$c + d$
Total		$a + c$	$b + d$	$a + b + c + d$

where :

- $a = \sum_{j=1}^p x_j y_j = \text{Nb. of shared labels}$
- $b = \sum_{j=1}^p x_j (1 - y_j) = \text{Nb. of labels } \mathbf{x} \text{ has that } \mathbf{y} \text{ hasn't}$
- $c = \sum_{j=1}^p (1 - x_j) y_j = \text{Nb. of labels } \mathbf{y} \text{ has that } \mathbf{x} \text{ hasn't}$
- $d = \sum_{j=1}^p (1 - x_j) (1 - y_j) = \text{Nb. of labels that neither } \mathbf{x} \text{ nor } \mathbf{y} \text{ has}$

## Some classic similarity measures (cont'd)

- **Jaccard** :  $S_{jaccard}(\mathbf{x}, \mathbf{y}) = \frac{a}{a+b+c} \in [0, 1]$

## Some classic similarity measures (cont'd)

- **Jaccard** :  $S_{jaccard}(\mathbf{x}, \mathbf{y}) = \frac{a}{a+b+c} \in [0, 1]$

- **Dice** :  $S_{dice}(\mathbf{x}, \mathbf{y}) = \frac{2a}{2a+b+c} \in [0, 1]$

## Some classic similarity measures (cont'd)

- **Jaccard** :  $S_{jaccard}(\mathbf{x}, \mathbf{y}) = \frac{a}{a+b+c} \in [0, 1]$

- **Dice** :  $S_{dice}(\mathbf{x}, \mathbf{y}) = \frac{2a}{2a+b+c} \in [0, 1]$

- **Ochiai** :  $S_{ochiai}(\mathbf{x}, \mathbf{y}) = \frac{a}{\sqrt{(a+b)(a+c)}} \in [0, 1]$

## Some classic similarity measures (cont'd)

- **Jaccard** :  $S_{jaccard}(\mathbf{x}, \mathbf{y}) = \frac{a}{a+b+c} \in [0, 1]$

- **Dice** :  $S_{dice}(\mathbf{x}, \mathbf{y}) = \frac{2a}{2a+b+c} \in [0, 1]$

- **Ochiai** :  $S_{ochiai}(\mathbf{x}, \mathbf{y}) = \frac{a}{\sqrt{(a+b)(a+c)}} \in [0, 1]$

- **Kulczynski** :  $S_{kulczynski}(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \left( \frac{a}{a+b} + \frac{a}{a+c} \right) \in [0, 1]$

## Some classic similarity measures (cont'd)

- **Jaccard** :  $S_{jaccard}(\mathbf{x}, \mathbf{y}) = \frac{a}{a+b+c} \in [0, 1]$
- **Dice** :  $S_{dice}(\mathbf{x}, \mathbf{y}) = \frac{2a}{2a+b+c} \in [0, 1]$
- **Ochiai** :  $S_{ochiai}(\mathbf{x}, \mathbf{y}) = \frac{a}{\sqrt{(a+b)(a+c)}} \in [0, 1]$
- **Kulczynski** :  $S_{kulczynski}(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \left( \frac{a}{a+b} + \frac{a}{a+c} \right) \in [0, 1]$
- **Sokal-Michener** :  $S_{soc-mich}(\mathbf{x}, \mathbf{y}) = \frac{a+d}{a+b+c+d} \in [0, 1]$

## Some classic similarity measures (cont'd)

- **Jaccard** :  $S_{jaccard}(\mathbf{x}, \mathbf{y}) = \frac{a}{a+b+c} \in [0, 1]$
- **Dice** :  $S_{dice}(\mathbf{x}, \mathbf{y}) = \frac{2a}{2a+b+c} \in [0, 1]$
- **Ochiai** :  $S_{ochiai}(\mathbf{x}, \mathbf{y}) = \frac{a}{\sqrt{(a+b)(a+c)}} \in [0, 1]$
- **Kulczynski** :  $S_{kulczynski}(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \left( \frac{a}{a+b} + \frac{a}{a+c} \right) \in [0, 1]$
- **Sokal-Michener** :  $S_{soc-mich}(\mathbf{x}, \mathbf{y}) = \frac{a+d}{a+b+c+d} \in [0, 1]$
- **Rogers-Tanimoto** :  $S_{rog-tan}(\mathbf{x}, \mathbf{y}) = \frac{a+d}{a+2(b+c)+d} \in [0, 1]$
- **Phi** :  $S_{phi}(\mathbf{x}, \mathbf{y}) = \frac{ad-bc}{\sqrt{(a+b)(c+d)(a+c)(b+d)}} \in [-1, 1]$

## Some classic similarity measures (cont'd)

- **Jaccard** :  $S_{jaccard}(\mathbf{x}, \mathbf{y}) = \frac{a}{a+b+c} \in [0, 1]$
- **Dice** :  $S_{dice}(\mathbf{x}, \mathbf{y}) = \frac{2a}{2a+b+c} \in [0, 1]$
- **Ochiai** :  $S_{ochiai}(\mathbf{x}, \mathbf{y}) = \frac{a}{\sqrt{(a+b)(a+c)}} \in [0, 1]$
- **Kulczynski** :  $S_{kulczynski}(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \left( \frac{a}{a+b} + \frac{a}{a+c} \right) \in [0, 1]$
- **Sokal-Michener** :  $S_{soc-mich}(\mathbf{x}, \mathbf{y}) = \frac{a+d}{a+b+c+d} \in [0, 1]$
- **Rogers-Tanimoto** :  $S_{rog-tan}(\mathbf{x}, \mathbf{y}) = \frac{a+d}{a+2(b+c)+d} \in [0, 1]$

## Example

$$\mathbf{x} = \begin{matrix} \mathbf{x} \\ \mathbf{y} \end{matrix} \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

## Example

$$\mathbf{x} = \begin{matrix} \mathbf{x} \\ \mathbf{y} \end{matrix} \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

		y <sub>j</sub>		Total
		1	0	
x <sub>j</sub>	1	3	2	5
	0	1	1	2
Total	4	3	7	

## Example

$$\mathbf{x} = \begin{matrix} \mathbf{x} \\ \mathbf{y} \end{matrix} \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

		y <sub>j</sub>		Total
		1	0	
x <sub>j</sub>	1	3	2	5
	0	1	1	2
Total	4	3	7	

$$\bullet S_{jaccard}(\mathbf{x}, \mathbf{y}) = \frac{1}{2}$$

$$\bullet S_{dice}(\mathbf{x}, \mathbf{y}) = \frac{2}{3}$$

$$\bullet S_{ochiai}(\mathbf{x}, \mathbf{y}) = \frac{3}{\sqrt{20}}$$

$$\bullet S_{kulczynski}(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \left( \frac{3}{5} + \frac{3}{4} \right)$$

$$\bullet S_{sok-mich}(\mathbf{x}, \mathbf{y}) = \frac{4}{7}$$

$$\bullet S_{rog-tan}(\mathbf{x}, \mathbf{y}) = \frac{4}{10}$$

$$\bullet S_{phi}(\mathbf{x}, \mathbf{y}) = \frac{1}{\sqrt{120}}$$

## Example

$$\mathbf{x} = \begin{matrix} \mathbf{x} \\ \mathbf{y} \end{matrix} \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

		y <sub>j</sub>		Total
		1	0	
x <sub>j</sub>	1	3	2	5
	0	1	1	2
Total	4	3	7	

$$\bullet S_{jaccard}(\mathbf{x}, \mathbf{y}) = \frac{1}{2}$$

$$\bullet S_{dice}(\mathbf{x}, \mathbf{y}) = \frac{2}{3}$$

$$\bullet S_{ochiai}(\mathbf{x}, \mathbf{y}) = \frac{3}{\sqrt{20}}$$

$$\bullet S_{kulczynski}(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \left( \frac{3}{5} + \frac{3}{4} \right)$$

## Two families of similarity measures for binary data

We can distinguish :

- “symmetric” measure :  $d$  plays a role in the proximity measure
- “asymmetric” measure :  $d$  does not play any role in the proximity measure

## Two families of similarity measures for binary data

We can distinguish :

- “symmetric” measure :  $d$  plays a role in the proximity measure
- “asymmetric” measure :  $d$  does not play any role in the proximity measure

Following [Gower and Legendre, 1986], we have the two following families :

- Symmetric measures :

$$S_{\alpha}^1(\mathbf{x}, \mathbf{y}) = \frac{a + d}{a + d + \alpha(b + c)}$$

- Asymmetric measures :

$$S_{\alpha}^2(\mathbf{x}, \mathbf{y}) = \frac{a}{a + \alpha(b + c)}$$

## Two families of similarity measures (cont'd)

Results on the families  $S^1$  and  $S^2$  (cf [Gower and Legendre, 1986]) :

**Theorem.**

$D_{\alpha}^1 = 1 - S_{\alpha}^1$  is metric for  $\alpha \geq 1$  and  $\sqrt{D_{\alpha}^1} = \sqrt{1 - S_{\alpha}^1}$  is metric for  $\alpha \geq 1/3$ . If  $\alpha < 1$  then  $D_{\alpha}^1$  may be non metric and if  $\alpha < 1/3$  then  $\sqrt{D_{\alpha}^1}$  may be non metric.

## Two families of similarity measures (cont'd)

Results on the families  $S^1$  and  $S^2$  (cf [Gower and Legendre, 1986]) :

**Theorem.**

$D_{\alpha}^1 = 1 - S_{\alpha}^1$  is metric for  $\alpha \geq 1$  and  $\sqrt{D_{\alpha}^1} = \sqrt{1 - S_{\alpha}^1}$  is metric for  $\alpha \geq 1/3$ . If  $\alpha < 1$  then  $D_{\alpha}^1$  may be non metric and if  $\alpha < 1/3$  then  $\sqrt{D_{\alpha}^1}$  may be non metric.

**Theorem.**

$D_{\alpha}^2 = 1 - S_{\alpha}^2$  is metric for  $\alpha \geq 1$  and  $\sqrt{D_{\alpha}^2} = \sqrt{1 - S_{\alpha}^2}$  is metric for  $\alpha \geq 1/3$ . If  $\alpha < 1$  then  $D_{\alpha}^2$  may be non metric and if  $\alpha < 1/3$  then  $\sqrt{D_{\alpha}^2}$  may be non metric.

## Two families of similarity measures (cont'd)

Results on the families  $S^1$  and  $S^2$  (cf [Gower and Legendre, 1986]) :

**Theorem.**

$D_{\alpha}^1 = 1 - S_{\alpha}^1$  is metric for  $\alpha \geq 1$  and  $\sqrt{D_{\alpha}^1} = \sqrt{1 - S_{\alpha}^1}$  is metric for  $\alpha \geq 1/3$ . If  $\alpha < 1$  then  $D_{\alpha}^1$  may be non metric and if  $\alpha < 1/3$  then  $\sqrt{D_{\alpha}^1}$  may be non metric.

**Theorem.**

$D_{\alpha}^2 = 1 - S_{\alpha}^2$  is metric for  $\alpha \geq 1$  and  $\sqrt{D_{\alpha}^2} = \sqrt{1 - S_{\alpha}^2}$  is metric for  $\alpha \geq 1/3$ . If  $\alpha < 1$  then  $D_{\alpha}^2$  may be non metric and if  $\alpha < 1/3$  then  $\sqrt{D_{\alpha}^2}$  may be non metric.

**Theorem.**

The distance matrix of general term  $\sqrt{D_{\alpha}^1} = \sqrt{1 - S_{\alpha}^1}$  is euclidean for  $\alpha \geq 1$ . The distance matrix of general term  $\sqrt{D_{\alpha}^2} = \sqrt{1 - S_{\alpha}^2}$  is euclidean for  $\alpha \geq 1/2$ . Otherwise,  $D_{\alpha}^1$  and  $D_{\alpha}^2$  may be non euclidean.

## Example

```

> install.packages("proxy")
> library(proxy)
> x=c(1,1,1,1,0,1,0)
> y=c(1,0,1,1,1,0,0)
> X=data.frame(rbind(x,y))
> simil(X,method="Jaccard")
  x
y 0.5
> simil(X,method="Dice")
  x
y 0.6666667
> simil(X,method="Ochiai")
  x
y 0.6708204
> simil(X,method="Kulczynski2")
  x
y 0.675
> simil(X,method="Sokal/Michener")
  x
y 0.5714286
> simil(X,method="Rogers")
  x
y 0.4
> simil(X,method="Phi")
  x
y 0.0912871

```

**Exercise 5 :** Does the Jaccard similarity measure belong to  $S_{\alpha}^1$  or  $S_{\alpha}^2$  ?

Using R, consider the data `poison.text.bin`, compute  $S_{jaccard}$  and show that the distance matrix of general term  $\sqrt{1 - S_{jaccard}(\mathbf{x}, \mathbf{y})}$  is euclidean.

## Introduction

- The data points are represented by a mix between continuous and discrete features  $\mathbb{R}^P$

Classical examples are individuals described by socio-economic variables (age, sex, marital status, number of children, incomes, ...)

## Introduction

- The data points are represented by a mix between continuous and discrete features  $\mathbb{R}^P$

Classical examples are individuals described by socio-economic variables (age, sex, marital status, number of children, incomes, ...)

In such a case how do we represent the data points in an homogeneous representation space? and how do we measure proximities between points?

## Introduction

- The data points are represented by a mix between continuous and discrete features  $\mathbb{R}^P$

Classical examples are individuals described by socio-economic variables (age, sex, marital status, number of children, incomes, ...)

In such a case how do we represent the data points in an homogeneous representation space? and how do we measure proximities between points?

Three approaches :

- Transform the data of one type to the other type (continuous to discrete for eg)

## Introduction

- The data points are represented by a mix between continuous and discrete features  $\mathbb{R}^P$

Classical examples are individuals described by socio-economic variables (age, sex, marital status, number of children, incomes, ...)

In such a case how do we represent the data points in an homogeneous representation space? and how do we measure proximities between points?

Three approaches :

- Transform the data of one type to the other type (continuous to discrete for eg)
- Define a similarity measure that can incorporate information from different types in a homogeneous manner

## Data transformation

We suppose that there is a set of continuous features and a set of discrete features. There are two approaches :

## Introduction

- The data points are represented by a mix between continuous and discrete features  $\mathbb{R}^P$

Classical examples are individuals described by socio-economic variables (age, sex, marital status, number of children, incomes, ...)

In such a case how do we represent the data points in an homogeneous representation space? and how do we measure proximities between points?

Three approaches :

- Transform the data of one type to the other type (continuous to discrete for eg)
- Define a similarity measure that can incorporate information from different types in a homogeneous manner
- Define as many proximity measures as types of data, analyze the data involving one type of features independently from the other ones and aggregate the different clustering outputs afterwards

## Data transformation

We suppose that there is a set of continuous features and a set of discrete features. There are two approaches :

- Transform each continuous variable into a discrete one : discretization techniques (see for eg [Dougherty et al., 1995, Zighed et al., 1998])



## Data transformation

We suppose that there is a set of continuous features and a set of discrete features. There are two approaches :

- Transform each continuous variable into a discrete one : discretization techniques (see for eg [Dougherty et al., 1995, Zighed et al., 1998])
- Transform each discrete feature into a continuous one : quantization techniques (CA for eg)

## A general similarity coefficient

Proposed by [Gower, 1971]. Let  $\mathbf{x}$  and  $\mathbf{y}$  be two  $p$ -dimensional data points :

$$S_{gower}(\mathbf{x}, \mathbf{y}) = \frac{1}{\sum_{j=1}^p w(x_j, y_j)} \sum_{j=1}^p w(x_j, y_j) S(x_j, y_j)$$

- If  $j$  is continuous :  $S(x_j, y_j) = 1 - \frac{|x_j - y_j|}{r_j}$  ; and  $w(x_j, y_j) = 0$  if  $x_j$  and  $y_j$  are missing values,  $w(x_j, y_j) = 1$  otherwise

## A general similarity coefficient

Proposed by [Gower, 1971]. Let  $\mathbf{x}$  and  $\mathbf{y}$  be two  $p$ -dimensional data points :

$$S_{gower}(\mathbf{x}, \mathbf{y}) = \frac{1}{\sum_{j=1}^p w(x_j, y_j)} \sum_{j=1}^p w(x_j, y_j) S(x_j, y_j)$$

## A general similarity coefficient

Proposed by [Gower, 1971]. Let  $\mathbf{x}$  and  $\mathbf{y}$  be two  $p$ -dimensional data points :

$$S_{gower}(\mathbf{x}, \mathbf{y}) = \frac{1}{\sum_{j=1}^p w(x_j, y_j)} \sum_{j=1}^p w(x_j, y_j) S(x_j, y_j)$$

- If  $j$  is continuous :  $S(x_j, y_j) = 1 - \frac{|x_j - y_j|}{r_j}$  ; and  $w(x_j, y_j) = 0$  if  $x_j$  and  $y_j$  are missing values,  $w(x_j, y_j) = 1$  otherwise
- If  $j$  is binary :  $S(x_j, y_j) = 1$  if  $x_j = y_j$ ,  $S(x_j, y_j) = 0$  otherwise ; and  $w(x_j, y_j) = 0$  if  $x_j$  and  $y_j$  are missing values,  $w(x_j, y_j) = 1$  otherwise

## Consensus clustering

Consensus clustering is a branch of data clustering that focuses on the following problem :

Given different clusterings of the same data points that correspond to different views (based on different types of features in our case), how do we find a consensus clustering ie how do we aggregate those clusterings ?

## Consensus clustering

Consensus clustering is a branch of data clustering that focuses on the following problem :

Given different clusterings of the same data points that correspond to different views (based on different types of features in our case), how do we find a consensus clustering ie how do we aggregate those clusterings ?

See for eg [Goder and Filkov, 2008].

## Consensus clustering

Consensus clustering is a branch of data clustering that focuses on the following problem :

Given different clusterings of the same data points that correspond to different views (based on different types of features in our case), how do we find a consensus clustering ie how do we aggregate those clusterings ?

-  Dougherty, J., Kohavi, R., and Sahami, M. (1995).  
Supervised and Unsupervised Discretization of Continuous Features.  
In Prieditis, A. and Russell, S., editors, *Proceedings of the Twelfth International Conference on Machine Learning*, pages 194–202.
-  Goder, A. and Filkov, V. (2008).  
Consensus clustering algorithms : Comparison and refinement.  
In *ALENEX*, pages 109–117.
-  Gower, J. and Legendre, P. (1986).  
Metric and euclidean properties of dissimilarity coefficients.  
*Journal of classification*, 3 :5–48.
-  Gower, J. C. (1971).  
A General Coefficient of Similarity and Some of Its Properties.  
*Biometrics*, 27(4) :857–871.
-  Milligan, G. W. and Cooper, M. C. (1988).  
A study of standardization of variables in cluster analysis.  
*Journal of Classification*, 5(2) :181–204.
-  Santini, S. and Jain, R. (1999).  
Similarity measures.  
*IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9) :871–883.
-  Tversky, A. (1977).  
Features of similarity.  
*Psychological review*, 84 :327–352.
-  Zighed, D. A., Rabaséda, S., and Rakotomalala, R. (1998).  
Fusinter : a method for discretization of continuous attributes.  
*Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 6 :307–326.

# Data Clustering - Part 2

M2 DMKM

Julien Ah-Pine (julien.ah-pine@univ-lyon2.fr)

Université Lyon 2

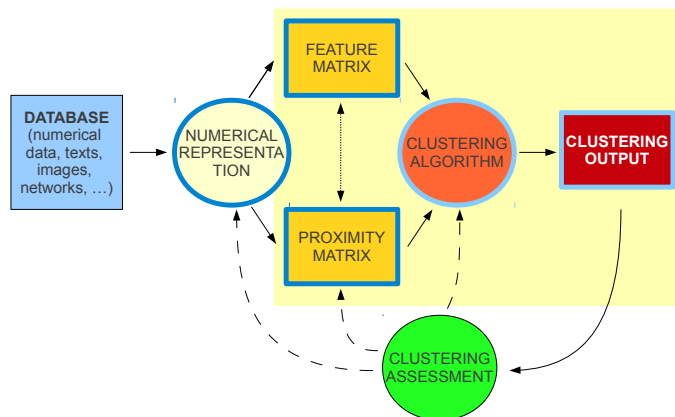
2015-2016

# Organization

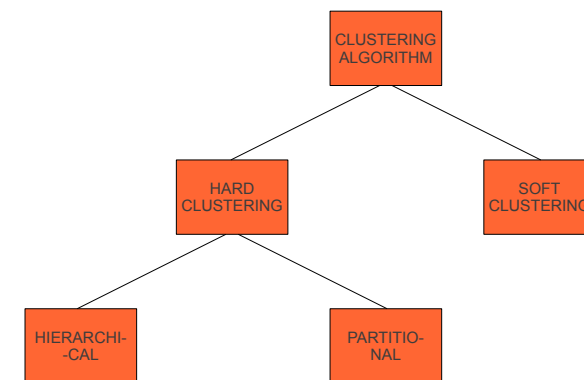
Outline of today's lesson :

- 1 Hierarchical clustering (HC)
  - Agglomerative hierarchical clustering (AHC)
  - Divisive hierarchical clustering (DHC)

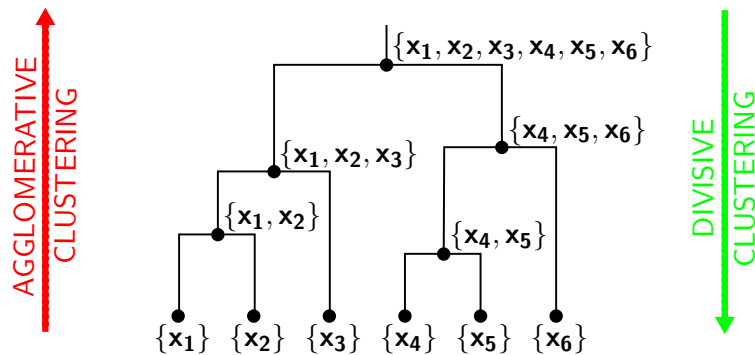
# Recalling the clustering process



# Different types of clustering algorithm



# Hierarchical clustering



Two types of hierarchical clustering algorithms :

- Agglomerative : “bottom-up”
- Divisive : “top-down”

# More about dendrograms

## Definition. (Dendrogram)

A **dendrogram** is a binary tree in which each internal node is associated with a **height**  $H$  satisfying the condition :

$$H(n) \leq H(n') \Leftrightarrow n \subset n'; \text{ where } n \text{ and } n' \text{ are two nodes of the binary tree}$$

$H(n)$  is the distance at which the cluster associated to  $n$  is created.

# More about dendrograms

## Definition. (Dendrogram)

A **dendrogram** is a binary tree in which each internal node is associated with a **height**  $H$  satisfying the condition :

$$H(n) \leq H(n') \Leftrightarrow n \subset n'; \text{ where } n \text{ and } n' \text{ are two nodes of the binary tree}$$

$H(n)$  is the distance at which the cluster associated to  $n$  is created.

## Definition. (Dissimilarity measure associated to a dendrogram)

For each pair of data points  $(x, y)$ , let  $H(x, y)$  be the height of the node in the dendrogram specifying the smallest cluster grouping  $x$  and  $y$  together.

# More about dendrograms

## Definition. (Dendrogram)

A **dendrogram** is a binary tree in which each internal node is associated with a **height**  $H$  satisfying the condition :

$$H(n) \leq H(n') \Leftrightarrow n \subset n'; \text{ where } n \text{ and } n' \text{ are two nodes of the binary tree}$$

$H(n)$  is the distance at which the cluster associated to  $n$  is created.

## Definition. (Dissimilarity measure associated to a dendrogram)

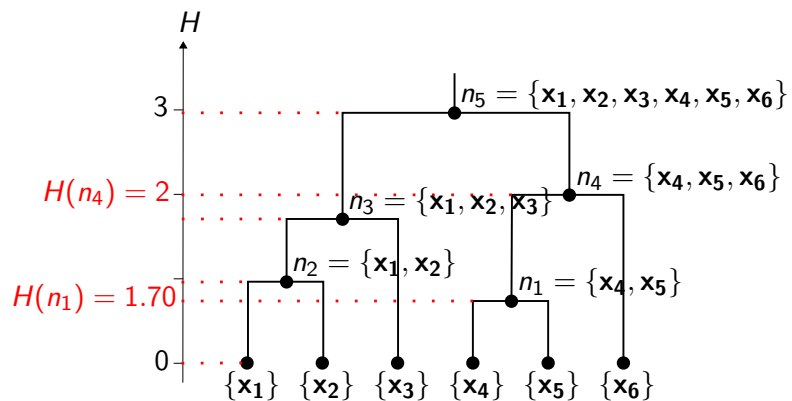
For each pair of data points  $(x, y)$ , let  $H(x, y)$  be the height of the node in the dendrogram specifying the smallest cluster grouping  $x$  and  $y$  together.

## Property. (Ultrametric property)

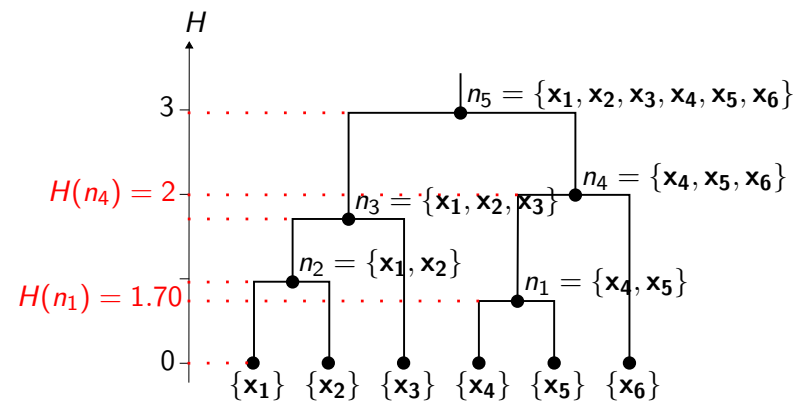
$H$  satisfies the following ultrametric property :

$$\forall x, y, z : H(x, y) \leq \max\{H(x, z), H(z, y)\}$$

### Example

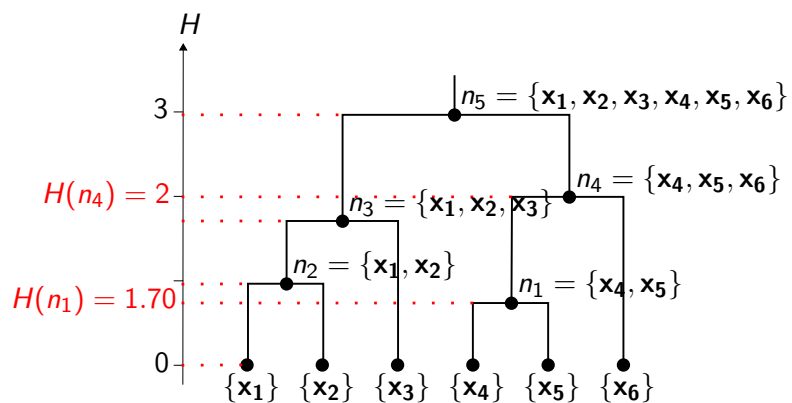


### Example



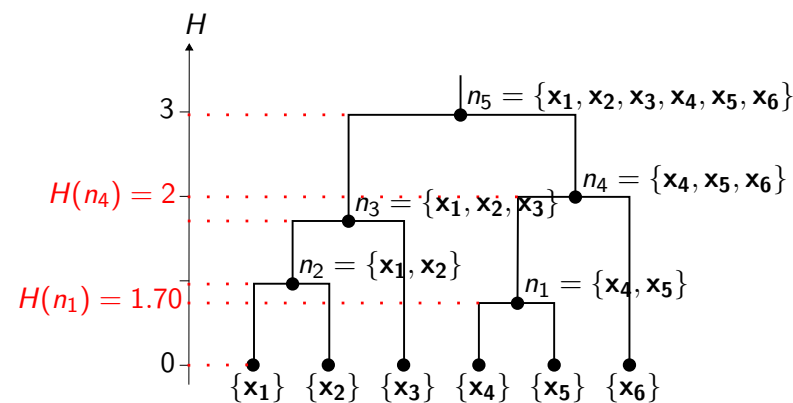
For example :  $H(x_4, x_6) =$

### Example



For example :  $H(x_4, x_6) = H(n_4) = 2$

### Example



For example :  $H(x_4, x_6) = H(n_4) = 2$  and  
 $H(x_4, x_6) \leq \max\{H(x_4, x_5), H(x_5, x_6)\} = \max\{H(n_1), H(n_4)\} = H(n_4)$

## More about dendrograms (cont'd)

## Definition.

More formally, a dendrogram representing a hierarchical clustering of  $n$  data points is represented by a function  $dend : [0, +\infty[ \rightarrow \mathbb{C}_n$  such that :

- $dend(h) \subseteq dend(h')$  if  $h \leq h'$
- $dend(h + \delta) = dend(h)$  for some small  $\delta$

## More about dendrograms (cont'd)

## Definition.

More formally, a dendrogram representing a hierarchical clustering of  $n$  data points is represented by a function  $dend : [0, +\infty[ \rightarrow \mathbb{C}_n$  such that :

- $dend(h) \subseteq dend(h')$  if  $h \leq h'$
- $dend(h + \delta) = dend(h)$  for some small  $\delta$

The dendrogram of the last example is represented by :

$$dend(h) = \begin{cases} \{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\}, \{\mathbf{x}_6\} & \text{if } 0 \leq h < H(n_1) \\ \{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4, \mathbf{x}_5\}, \{\mathbf{x}_6\} & \text{if } H(n_1) \leq h < H(n_2) \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4, \mathbf{x}_5\}, \{\mathbf{x}_6\} & \text{if } H(n_2) \leq h < H(n_3) \\ \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}, \{\mathbf{x}_4, \mathbf{x}_5\}, \{\mathbf{x}_6\} & \text{if } H(n_3) \leq h < H(n_4) \\ \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}, \{\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6\} & \text{if } H(n_4) \leq h < H(n_5) \\ \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} & \text{if } H(n_5) \leq h \end{cases}$$

## More about dendrograms (cont'd)

## Definition.

More formally, a dendrogram representing a hierarchical clustering of  $n$  data points is represented by a function  $dend : [0, +\infty[ \rightarrow \mathbb{C}_n$  such that :

- $dend(h) \subseteq dend(h')$  if  $h \leq h'$
- $dend(h + \delta) = dend(h)$  for some small  $\delta$

The dendrogram of the last example is represented by :

$$dend(h) = \begin{cases} \{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\}, \{\mathbf{x}_6\} & \text{if } 0 \leq h < H(n_1) \\ \{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4, \mathbf{x}_5\}, \{\mathbf{x}_6\} & \text{if } H(n_1) \leq h < H(n_2) \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4, \mathbf{x}_5\}, \{\mathbf{x}_6\} & \text{if } H(n_2) \leq h < H(n_3) \\ \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}, \{\mathbf{x}_4, \mathbf{x}_5\}, \{\mathbf{x}_6\} & \text{if } H(n_3) \leq h < H(n_4) \\ \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}, \{\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6\} & \text{if } H(n_4) \leq h < H(n_5) \\ \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} & \text{if } H(n_5) \leq h \end{cases}$$

**Comment :** Most of HC algorithms are represented by dendrograms but some are not ultrametrics (there can be "reversals"). In such cases we talk about tree diagrams and they are more difficult to interpret.

## Outline

- 1 Hierarchical clustering (HC)
  - Agglomerative hierarchical clustering (AHC)
  - Divisive hierarchical clustering (DHC)

## Pseudo-code of AHC

Pseudo-code of agglomerative hierarchical clusterings (AHC) :

- 1 **Input** :  $D$
- 2 Initialize the tree representation with  $n$  leaves
- 3 **While** not all data points are grouped together **do**
- 4     Merge the two closest clusters according to some distance measure
- 5     Add a parent node in the tree representation accordingly
- 6 **End While**
- 7 **Output** : tree representation

## Classification of AHC

We can distinguish AHC algorithms according to the type of distance measures used. There are two approaches :

- Graph methods :
  - ▶ Single link method
  - ▶ Complete link method
  - ▶ Group average method (UPGMA)
  - ▶ Weighted group average method (WPGMA)

## Pseudo-code of AHC

Pseudo-code of agglomerative hierarchical clusterings (AHC) :

- 1 **Input** :  $D$
- 2 Initialize the tree representation with  $n$  leaves
- 3 **While** not all data points are grouped together **do**
- 4     Merge the two closest clusters according to some distance measure
- 5     Add a parent node in the tree representation accordingly
- 6 **End While**
- 7 **Output** : tree representation

The critical point for AHC algorithms is the distance measure between clusters.

## Classification of AHC

We can distinguish AHC algorithms according to the type of distance measures used. There are two approaches :

- Graph methods :
  - ▶ Single link method
  - ▶ Complete link method
  - ▶ Group average method (UPGMA)
  - ▶ Weighted group average method (WPGMA)
- Geometric :
  - ▶ Ward's method
  - ▶ Centroid method
  - ▶ Median method

## Classification of AHC

We can distinguish AHC algorithms according to the type of distance measures used. There are two approaches :

- Graph methods :
  - ▶ Single link method
  - ▶ Complete link method
  - ▶ Group average method (UPGMA)
  - ▶ Weighted group average method (WPGMA)
- Geometric :
  - ▶ Ward's method
  - ▶ Centroid method
  - ▶ Median method

In graph based methods, distances between clusters rely on distances between the data points in the clusters whereas in geometric based methods, clusters are represented by centroids and the distance between them rely on the distance between the centroids.

## The Lance-Williams formula

### Definition. (Lance-Williams formula)

In AHC algorithms, the **Lance-Williams formula** [Lance and Williams, 1967] is a recurrence equation used to calculate the dissimilarity between a cluster  $C_k$  and a cluster formed by merging two other clusters  $C_l \cup C_{l'}$  ( $C_k, C_l, C_{l'}$  are 3 clusters belonging to the same level of the HC) :

$$D_{LW}(C_k, C_l \cup C_{l'}) = \alpha_l D_{LW}(C_k, C_l) + \alpha_{l'} D_{LW}(C_k, C_{l'}) + \beta D_{LW}(C_l, C_{l'}) + \gamma |D_{LW}(C_k, C_l) - D_{LW}(C_k, C_{l'})|$$

where  $\alpha_l, \alpha_{l'}, \beta, \gamma$  are real numbers.

## The Lance-Williams formula

### Definition. (Lance-Williams formula)

In AHC algorithms, the **Lance-Williams formula** [Lance and Williams, 1967] is a recurrence equation used to calculate the dissimilarity between a cluster  $C_k$  and a cluster formed by merging two other clusters  $C_l \cup C_{l'}$  ( $C_k, C_l, C_{l'}$  are 3 clusters belonging to the same level of the HC) :

$$D_{LW}(C_k, C_l \cup C_{l'}) = \alpha_l D_{LW}(C_k, C_l) + \alpha_{l'} D_{LW}(C_k, C_{l'}) + \beta D_{LW}(C_l, C_{l'}) + \gamma |D_{LW}(C_k, C_l) - D_{LW}(C_k, C_{l'})|$$

where  $\alpha_l, \alpha_{l'}, \beta, \gamma$  are real numbers.

Each aforementioned method is a particular case of the LW formula.

## AHC methods and the Lance-Williams formula

Algorithm	$\alpha_l$	$\alpha_{l'}$	$\beta$	$\gamma$
Single link	1/2	1/2	0	-1/2
Complete link	1/2	1/2	0	1/2
UPGMA	$\frac{ C_l }{ C_l + C_{l'} }$	$\frac{ C_{l'} }{ C_l + C_{l'} }$	0	0
WPGMA	1/2	1/2	0	0
Ward	$\frac{ C_l + C_k }{ C_l + C_{l'} + C_k }$	$\frac{ C_{l'} + C_k }{ C_l + C_{l'} + C_k }$	$-\frac{ C_k }{ C_l + C_{l'} + C_k }$	0
Centroid	$\frac{ C_l }{ C_l + C_{l'} }$	$\frac{ C_{l'} }{ C_l + C_{l'} }$	$-\frac{ C_l  C_{l'} }{( C_l + C_{l'} )^2}$	0
Median	1/2	1/2	-1/4	0



## Ultrametric property and the Lance-Williams formula

### Definition.

The distances  $D_{LW}(C_k, C_l \cup C_{l'})$  are said to increase monotonically if  $D_{LW}(C_l, C_{l'}) \leq D_{LW}(C_k, C_l \cup C_{l'})$  at each level of the hierarchy.

## Ultrametric property and the Lance-Williams formula

### Definition.

The distances  $D_{LW}(C_k, C_l \cup C_{l'})$  are said to increase monotonically if  $D_{LW}(C_l, C_{l'}) \leq D_{LW}(C_k, C_l \cup C_{l'})$  at each level of the hierarchy.

### Property.

If an algorithm produces a monotonic hierarchy then it induces a distance which satisfies the ultrametric property.

## Ultrametric property and the Lance-Williams formula

### Definition.

The distances  $D_{LW}(C_k, C_l \cup C_{l'})$  are said to increase monotonically if  $D_{LW}(C_l, C_{l'}) \leq D_{LW}(C_k, C_l \cup C_{l'})$  at each level of the hierarchy.

### Property.

If an algorithm produces a monotonic hierarchy then it induces a distance which satisfies the ultrametric property.

### Property.

Using AHC with the LW formula, the hierarchical clustering strategy is monotonic iff :

$(\gamma \geq -\min\{\alpha_l, \alpha_{l'}\})$  and  $(\alpha_l + \alpha_{l'} \geq 0)$  and  $(\alpha_l + \alpha_{l'} + \beta \geq 1)$

## Ultrametric property and the Lance-Williams formula

### Definition.

The distances  $D_{LW}(C_k, C_l \cup C_{l'})$  are said to increase monotonically if  $D_{LW}(C_l, C_{l'}) \leq D_{LW}(C_k, C_l \cup C_{l'})$  at each level of the hierarchy.

### Property.

If an algorithm produces a monotonic hierarchy then it induces a distance which satisfies the ultrametric property.

### Property.

Using AHC with the LW formula, the hierarchical clustering strategy is monotonic iff :

$(\gamma \geq -\min\{\alpha_l, \alpha_{l'}\})$  and  $(\alpha_l + \alpha_{l'} \geq 0)$  and  $(\alpha_l + \alpha_{l'} + \beta \geq 1)$

**Exercise 6 :** Which AHC methods are not monotonic and can induce a distance that does not satisfy the ultrametric property ?

## Single link method

- One of the simplest AHC method proposed by Sneath in 1957

## Single link method

- One of the simplest AHC method proposed by Sneath in 1957
- Also known as the nearest neighbor method, since it employs the nearest neighbor to measure the dissimilarity between two clusters
- It is invariant under monotone transformations of the data

## Single link method

- One of the simplest AHC method proposed by Sneath in 1957
- Also known as the nearest neighbor method, since it employs the nearest neighbor to measure the dissimilarity between two clusters

## Single link method

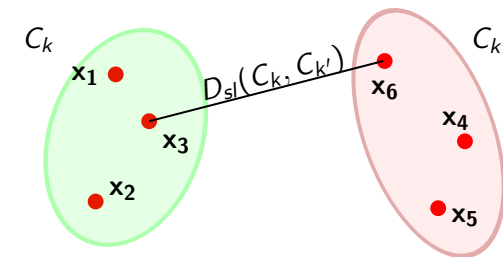
- One of the simplest AHC method proposed by Sneath in 1957
- Also known as the nearest neighbor method, since it employs the nearest neighbor to measure the dissimilarity between two clusters
- It is invariant under monotone transformations of the data
- According to the LW formula we have :

$$\begin{aligned}
 D_{sl}(C_k, C_l \cup C_{l'}) &= \frac{1}{2}D_{sl}(C_k, C_l) + \frac{1}{2}D_{sl}(C_k, C_{l'}) \\
 &\quad - \frac{1}{2}|D_{sl}(C_k, C_l) - D_{sl}(C_k, C_{l'})| \\
 &= \min\{D_{sl}(C_k, C_l), D_{sl}(C_k, C_{l'})\}
 \end{aligned}$$

## Single link method (cont'd)

Suppose that  $C_k$  and  $C_{k'}$  are two nonempty and nonoverlapping clusters and  $D$  is the distance function by which the dissimilarity matrix is computed then the  $D_{sl}$  distance between  $C_k$  and  $C_{k'}$  can be defined as follows :

$$D_{sl}(C_k, C_{k'}) = \min_{x \in C_k, y \in C_{k'}} \{D(x, y)\}$$

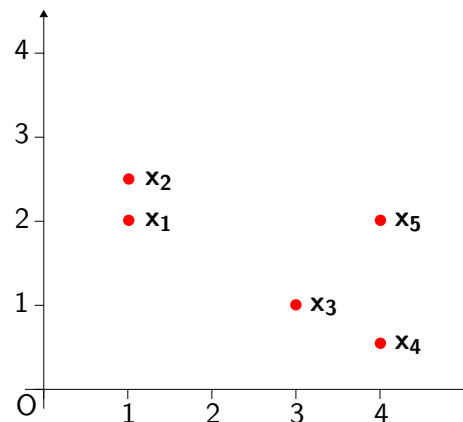


## Example

We consider 5 data points in  $\mathbb{R}^2$  :

- $x_1 = (1, 2)$
- $x_2 = (1, 2.5)$
- $x_3 = (3, 1)$
- $x_4 = (4, 0.5)$
- $x_5 = (4, 2)$

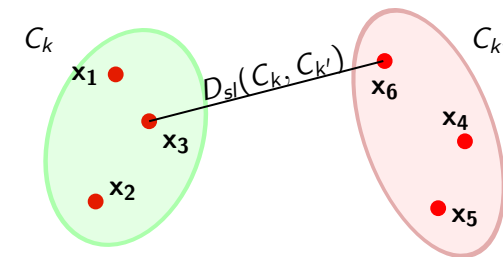
We consider the euclidean distance between data points.



## Single link method (cont'd)

Suppose that  $C_k$  and  $C_{k'}$  are two nonempty and nonoverlapping clusters and  $D$  is the distance function by which the dissimilarity matrix is computed then the  $D_{sl}$  distance between  $C_k$  and  $C_{k'}$  can be defined as follows :

$$D_{sl}(C_k, C_{k'}) = \min_{x \in C_k, y \in C_{k'}} \{D(x, y)\}$$



## Example (cont'd)

The starting distance matrix  $D$  is the euclidean distance matrix between points :

$$D = D_{eucl} = D_{sl} = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix} & \begin{pmatrix} 0 & 0.5 & 2.24 & 3.35 & 3 \\ 0.5 & 0 & 2.5 & 3.61 & 3.04 \\ 2.24 & 2.5 & 0 & 1.12 & 1.41 \\ 3.35 & 3.61 & 1.12 & 0 & 1.5 \\ 3 & 3.04 & 1.41 & 1.5 & 0 \end{pmatrix} \end{matrix}$$

## Example (cont'd)

The starting distance matrix  $\mathbf{D}$  is the euclidean distance matrix between points :

$$\mathbf{D} = \mathbf{D}_{\text{eucl}} = \mathbf{D}_{\text{sl}} = \begin{matrix} & \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{x}_5 \\ \mathbf{x}_1 & 0 & 0.5 & 2.24 & 3.35 & 3 \\ \mathbf{x}_2 & 0.5 & 0 & 2.5 & 3.61 & 3.04 \\ \mathbf{x}_3 & 2.24 & 2.5 & 0 & 1.12 & 1.41 \\ \mathbf{x}_4 & 3.35 & 3.61 & 1.12 & 0 & 1.5 \\ \mathbf{x}_5 & 3 & 3.04 & 1.41 & 1.5 & 0 \end{matrix}$$

$$2 \quad dend(h) = \{ \{ \mathbf{x}_1 \}, \{ \mathbf{x}_2 \}, \{ \mathbf{x}_3 \}, \{ \mathbf{x}_4 \}, \{ \mathbf{x}_5 \} \quad \text{if } 0 \leq h$$

4 Merge  $\mathbf{x}_1$  and  $\mathbf{x}_2$

$$dend(h) = \begin{cases} \{ \mathbf{x}_1 \}, \{ \mathbf{x}_2 \}, \{ \mathbf{x}_3 \}, \{ \mathbf{x}_4 \}, \{ \mathbf{x}_5 \} & \text{if } 0 \leq h < 0.5 \\ \{ \mathbf{x}_1, \mathbf{x}_2 \}, \{ \mathbf{x}_3 \}, \{ \mathbf{x}_4 \}, \{ \mathbf{x}_5 \} & \text{if } 0.5 \leq h \end{cases}$$

## Example (cont'd)

With the single link method, the distance matrix  $\mathbf{D}_{\text{sl}}$  becomes :

- $D_{\text{sl}}(\{ \mathbf{x}_1, \mathbf{x}_2 \}, \mathbf{x}_3) = \min\{D_{\text{sl}}(\mathbf{x}_1, \mathbf{x}_3), D_{\text{sl}}(\mathbf{x}_2, \mathbf{x}_3)\} = 2.24$
- $D_{\text{sl}}(\{ \mathbf{x}_1, \mathbf{x}_2 \}, \mathbf{x}_4) = \min\{D_{\text{sl}}(\mathbf{x}_1, \mathbf{x}_4), D_{\text{sl}}(\mathbf{x}_2, \mathbf{x}_4)\} = 3.35$
- $D_{\text{sl}}(\{ \mathbf{x}_1, \mathbf{x}_2 \}, \mathbf{x}_5) = \min\{D_{\text{sl}}(\mathbf{x}_1, \mathbf{x}_5), D_{\text{sl}}(\mathbf{x}_2, \mathbf{x}_5)\} = 3$

## Example (cont'd)

With the single link method, the distance matrix  $\mathbf{D}_{\text{sl}}$  becomes :

- $D_{\text{sl}}(\{ \mathbf{x}_1, \mathbf{x}_2 \}, \mathbf{x}_3) = \min\{D_{\text{sl}}(\mathbf{x}_1, \mathbf{x}_3), D_{\text{sl}}(\mathbf{x}_2, \mathbf{x}_3)\} = 2.24$
- $D_{\text{sl}}(\{ \mathbf{x}_1, \mathbf{x}_2 \}, \mathbf{x}_4) = \min\{D_{\text{sl}}(\mathbf{x}_1, \mathbf{x}_4), D_{\text{sl}}(\mathbf{x}_2, \mathbf{x}_4)\} = 3.35$
- $D_{\text{sl}}(\{ \mathbf{x}_1, \mathbf{x}_2 \}, \mathbf{x}_5) = \min\{D_{\text{sl}}(\mathbf{x}_1, \mathbf{x}_5), D_{\text{sl}}(\mathbf{x}_2, \mathbf{x}_5)\} = 3$

$$\mathbf{D}_{\text{sl}} = \begin{matrix} & \{ \mathbf{x}_1, \mathbf{x}_2 \} & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{x}_5 \\ \{ \mathbf{x}_1, \mathbf{x}_2 \} & 0 & 2.24 & 3.35 & 3 \\ \mathbf{x}_3 & 2.24 & 0 & 1.12 & 1.41 \\ \mathbf{x}_4 & 3.35 & 1.12 & 0 & 1.5 \\ \mathbf{x}_5 & 3 & 1.41 & 1.5 & 0 \end{matrix}$$

## Example (cont'd)

With the single link method, the distance matrix  $\mathbf{D}_{\text{sl}}$  becomes :

- $D_{\text{sl}}(\{ \mathbf{x}_1, \mathbf{x}_2 \}, \mathbf{x}_3) = \min\{D_{\text{sl}}(\mathbf{x}_1, \mathbf{x}_3), D_{\text{sl}}(\mathbf{x}_2, \mathbf{x}_3)\} = 2.24$
- $D_{\text{sl}}(\{ \mathbf{x}_1, \mathbf{x}_2 \}, \mathbf{x}_4) = \min\{D_{\text{sl}}(\mathbf{x}_1, \mathbf{x}_4), D_{\text{sl}}(\mathbf{x}_2, \mathbf{x}_4)\} = 3.35$
- $D_{\text{sl}}(\{ \mathbf{x}_1, \mathbf{x}_2 \}, \mathbf{x}_5) = \min\{D_{\text{sl}}(\mathbf{x}_1, \mathbf{x}_5), D_{\text{sl}}(\mathbf{x}_2, \mathbf{x}_5)\} = 3$

$$\mathbf{D}_{\text{sl}} = \begin{matrix} & \{ \mathbf{x}_1, \mathbf{x}_2 \} & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{x}_5 \\ \{ \mathbf{x}_1, \mathbf{x}_2 \} & 0 & 2.24 & 3.35 & 3 \\ \mathbf{x}_3 & 2.24 & 0 & 1.12 & 1.41 \\ \mathbf{x}_4 & 3.35 & 1.12 & 0 & 1.5 \\ \mathbf{x}_5 & 3 & 1.41 & 1.5 & 0 \end{matrix}$$

4 Merge  $\mathbf{x}_3$  and  $\mathbf{x}_4$

$$dend(h) = \begin{cases} \{ \mathbf{x}_1 \}, \{ \mathbf{x}_2 \}, \{ \mathbf{x}_3 \}, \{ \mathbf{x}_4 \}, \{ \mathbf{x}_5 \} & \text{if } 0 \leq h < 0.5 \\ \{ \mathbf{x}_1, \mathbf{x}_2 \}, \{ \mathbf{x}_3 \}, \{ \mathbf{x}_4 \}, \{ \mathbf{x}_5 \} & \text{if } 0.5 \leq h < 1.12 \\ \{ \mathbf{x}_1, \mathbf{x}_2 \}, \{ \mathbf{x}_3, \mathbf{x}_4 \}, \{ \mathbf{x}_5 \} & \text{if } 1.12 \leq h \end{cases}$$

## Example (cont'd)

With the single link method, the distance matrix  $\mathbf{D}_{sl}$  becomes :

- $D_{sl}(\{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_1, \mathbf{x}_2\}) = \min\{D_{sl}(\mathbf{x}_3, \mathbf{x}_1), D_{sl}(\mathbf{x}_3, \mathbf{x}_2), D_{sl}(\mathbf{x}_4, \mathbf{x}_1), D_{sl}(\mathbf{x}_4, \mathbf{x}_2)\} = \min\{D_{sl}(\mathbf{x}_3, \{\mathbf{x}_1, \mathbf{x}_2\}), D_{sl}(\mathbf{x}_4, \{\mathbf{x}_1, \mathbf{x}_2\})\} = 2.24$
- $D_{sl}(\{\mathbf{x}_3, \mathbf{x}_4\}, \mathbf{x}_5) = \min\{D_{sl}(\mathbf{x}_3, \mathbf{x}_5), D_{sl}(\mathbf{x}_4, \mathbf{x}_5)\} = 1.41$

## Example (cont'd)

With the single link method, the distance matrix  $\mathbf{D}_{sl}$  becomes :

- $D_{sl}(\{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_1, \mathbf{x}_2\}) = \min\{D_{sl}(\mathbf{x}_3, \mathbf{x}_1), D_{sl}(\mathbf{x}_3, \mathbf{x}_2), D_{sl}(\mathbf{x}_4, \mathbf{x}_1), D_{sl}(\mathbf{x}_4, \mathbf{x}_2)\} = \min\{D_{sl}(\mathbf{x}_3, \{\mathbf{x}_1, \mathbf{x}_2\}), D_{sl}(\mathbf{x}_4, \{\mathbf{x}_1, \mathbf{x}_2\})\} = 2.24$
- $D_{sl}(\{\mathbf{x}_3, \mathbf{x}_4\}, \mathbf{x}_5) = \min\{D_{sl}(\mathbf{x}_3, \mathbf{x}_5), D_{sl}(\mathbf{x}_4, \mathbf{x}_5)\} = 1.41$

$$\mathbf{D}_{sl} = \begin{matrix} & \{\mathbf{x}_1, \mathbf{x}_2\} & \{\mathbf{x}_3, \mathbf{x}_4\} & \mathbf{x}_5 \\ \begin{matrix} \{\mathbf{x}_1, \mathbf{x}_2\} \\ \{\mathbf{x}_3, \mathbf{x}_4\} \\ \mathbf{x}_5 \end{matrix} & \begin{pmatrix} 0 & 2.24 & 3 \\ 2.24 & 0 & \mathbf{1.41} \\ 3 & \mathbf{1.41} & 0 \end{pmatrix} \end{matrix}$$

4 Merge  $\{\mathbf{x}_3, \mathbf{x}_4\}$  and  $\mathbf{x}_5$

$$dend(h) = \begin{cases} \{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 0 \leq h < 0.5 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 0.5 \leq h < 1.12 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 1.12 \leq h < \mathbf{1.41} \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} & \text{if } \mathbf{1.41} \leq h \end{cases}$$

## Example (cont'd)

With the single link method, the distance matrix  $\mathbf{D}_{sl}$  becomes :

- $D_{sl}(\{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_1, \mathbf{x}_2\}) = \min\{D_{sl}(\mathbf{x}_3, \mathbf{x}_1), D_{sl}(\mathbf{x}_3, \mathbf{x}_2), D_{sl}(\mathbf{x}_4, \mathbf{x}_1), D_{sl}(\mathbf{x}_4, \mathbf{x}_2)\} = \min\{D_{sl}(\mathbf{x}_3, \{\mathbf{x}_1, \mathbf{x}_2\}), D_{sl}(\mathbf{x}_4, \{\mathbf{x}_1, \mathbf{x}_2\})\} = 2.24$
- $D_{sl}(\{\mathbf{x}_3, \mathbf{x}_4\}, \mathbf{x}_5) = \min\{D_{sl}(\mathbf{x}_3, \mathbf{x}_5), D_{sl}(\mathbf{x}_4, \mathbf{x}_5)\} = 1.41$

$$\mathbf{D}_{sl} = \begin{matrix} & \{\mathbf{x}_1, \mathbf{x}_2\} & \{\mathbf{x}_3, \mathbf{x}_4\} & \mathbf{x}_5 \\ \begin{matrix} \{\mathbf{x}_1, \mathbf{x}_2\} \\ \{\mathbf{x}_3, \mathbf{x}_4\} \\ \mathbf{x}_5 \end{matrix} & \begin{pmatrix} 0 & 2.24 & 3 \\ 2.24 & 0 & 1.41 \\ 3 & 1.41 & 0 \end{pmatrix} \end{matrix}$$

## Example (cont'd)

With the single link method, the distance matrix  $\mathbf{D}_{sl}$  becomes :

- $D_{sl}(\{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\}, \{\mathbf{x}_1, \mathbf{x}_2\}) = \min\{D_{sl}(\{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_1, \mathbf{x}_2\}), D_{sl}(\mathbf{x}_5, \{\mathbf{x}_1, \mathbf{x}_2\})\} = 2.24$

### Example (cont'd)

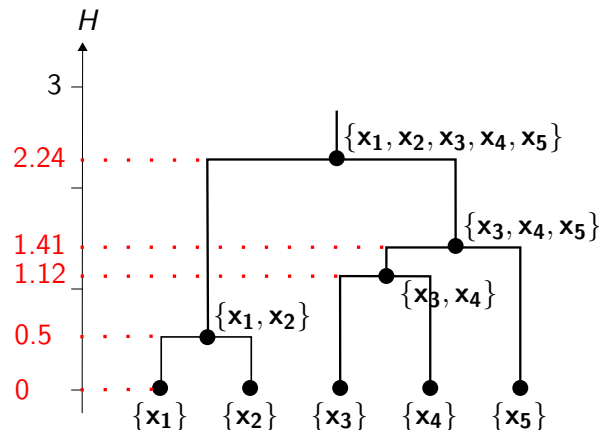
With the single link method, the distance matrix  $D_{sl}$  becomes :

- $D_{sl}(\{x_3, x_4, x_5\}, \{x_1, x_2\}) = \min\{D_{sl}(\{x_3, x_4\}, \{x_1, x_2\}), D_{sl}(x_5, \{x_1, x_2\})\} = 2.24$

$$D_{sl} = \begin{matrix} & \{x_1, x_2\} & \{x_3, x_4, x_5\} \\ \begin{matrix} \{x_1, x_2\} \\ \{x_3, x_4, x_5\} \end{matrix} & \begin{pmatrix} 0 & 2.24 \\ 2.24 & 0 \end{pmatrix} \end{matrix}$$

### Example (cont'd)

The dendrogram :



### Example (cont'd)

With the single link method, the distance matrix  $D_{sl}$  becomes :

- $D_{sl}(\{x_3, x_4, x_5\}, \{x_1, x_2\}) = \min\{D_{sl}(\{x_3, x_4\}, \{x_1, x_2\}), D_{sl}(x_5, \{x_1, x_2\})\} = 2.24$

$$D_{sl} = \begin{matrix} & \{x_1, x_2\} & \{x_3, x_4, x_5\} \\ \begin{matrix} \{x_1, x_2\} \\ \{x_3, x_4, x_5\} \end{matrix} & \begin{pmatrix} 0 & 2.24 \\ 2.24 & 0 \end{pmatrix} \end{matrix}$$

4 Merge  $\{x_3, x_4, x_5\}$  and  $\{x_1, x_2\}$

$$dend(h) = \begin{cases} \{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\} & \text{if } 0 \leq h < 0.5 \\ \{x_1, x_2\}, \{x_3\}, \{x_4\}, \{x_5\} & \text{if } 0.5 \leq h < 1.12 \\ \{x_1, x_2\}, \{x_3, x_4\}, \{x_5\} & \text{if } 1.12 \leq h < 1.41 \\ \{x_1, x_2\}, \{x_3, x_4, x_5\} & \text{if } 1.41 \leq h < 2.24 \\ \{x_1, x_2, x_3, x_4, x_5\} & \text{if } 2.24 \leq h \end{cases}$$

### Complete link method

- Introduced by McQuitty in 1960

## Complete link method

- Introduced by McQuitty in 1960
- Unlike single link methods, it employs the farthest neighbor to measure the dissimilarity between two clusters

## Complete link method

- Introduced by McQuitty in 1960
- Unlike single link methods, it employs the farthest neighbor to measure the dissimilarity between two clusters
- It is also invariant under monotone transformations of the data
- Following the LW formula this method uses the updating rule :

$$\begin{aligned}
 D_{sl}(C_k, C_l \cup C_{l'}) &= \frac{1}{2}D_{sl}(C_k, C_l) + \frac{1}{2}D_{sl}(C_k, C_{l'}) \\
 &\quad + \frac{1}{2}|D_{sl}(C_k, C_l) - D_{sl}(C_k, C_{l'})| \\
 &= \max\{D_{sl}(C_k, C_l), D_{sl}(C_k, C_{l'})\}
 \end{aligned}$$

## Complete link method

- Introduced by McQuitty in 1960
- Unlike single link methods, it employs the farthest neighbor to measure the dissimilarity between two clusters
- It is also invariant under monotone transformations of the data

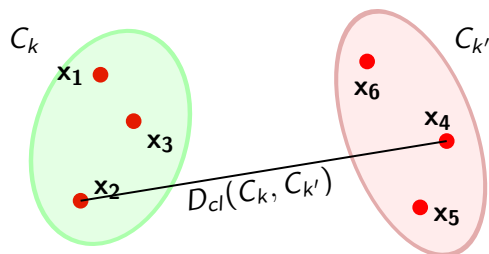
## Complete link method (cont'd)

Suppose that  $C_k$  and  $C_{k'}$  are two nonempty and nonoverlapping clusters and  $D$  the distance function by which the dissimilarity matrix is computed, then the distance between  $C_k$  and  $C_{k'}$  can also be defined as follows :

## Complete link method (cont'd)

Suppose that  $C_k$  and  $C_{k'}$  are two nonempty and nonoverlapping clusters and  $D$  the distance function by which the dissimilarity matrix is computed, then the distance between  $C_k$  and  $C_{k'}$  can also be defined as follows :

$$D_{cl}(C_k, C_{k'}) = \max_{x \in C_k, y \in C_{k'}} \{D(x, y)\}$$



## Example

We consider the same example as previously. The starting distance matrix  $D$  is again the euclidean distance matrix between points :

$$D = D_{eucl} = D_{cl} = \begin{matrix} & \mathbf{x_1} & \mathbf{x_2} & \mathbf{x_3} & \mathbf{x_4} & \mathbf{x_5} \\ \mathbf{x_1} & 0 & 0.5 & 2.24 & 3.35 & 3 \\ \mathbf{x_2} & 0.5 & 0 & 2.5 & 3.61 & 3.04 \\ \mathbf{x_3} & 2.24 & 2.5 & 0 & 1.12 & 1.41 \\ \mathbf{x_4} & 3.35 & 3.61 & 1.12 & 0 & 1.5 \\ \mathbf{x_5} & 3 & 3.04 & 1.41 & 1.5 & 0 \end{matrix}$$

$$2 \quad dend(h) = \{ \{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\} \} \quad \text{if } 0 \leq h$$

4 Merge  $x_1$  and  $x_2$

$$dend(h) = \begin{cases} \{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\} & \text{if } 0 \leq h < 0.5 \\ \{x_1, x_2\}, \{x_3\}, \{x_4\}, \{x_5\} & \text{if } 0.5 \leq h \end{cases}$$

## Example

We consider the same example as previously. The starting distance matrix  $D$  is again the euclidean distance matrix between points :

$$D = D_{eucl} = D_{cl} = \begin{matrix} & \mathbf{x_1} & \mathbf{x_2} & \mathbf{x_3} & \mathbf{x_4} & \mathbf{x_5} \\ \mathbf{x_1} & 0 & 0.5 & 2.24 & 3.35 & 3 \\ \mathbf{x_2} & 0.5 & 0 & 2.5 & 3.61 & 3.04 \\ \mathbf{x_3} & 2.24 & 2.5 & 0 & 1.12 & 1.41 \\ \mathbf{x_4} & 3.35 & 3.61 & 1.12 & 0 & 1.5 \\ \mathbf{x_5} & 3 & 3.04 & 1.41 & 1.5 & 0 \end{matrix}$$

## Example (cont'd)

With the complete link method, the distance matrix  $D_{cl}$  becomes :

- $D_{cl}(\{x_1, x_2\}, x_3) = \max\{D_{cl}(x_1, x_3), D_{cl}(x_2, x_3)\} = 2.5$
- $D_{cl}(\{x_1, x_2\}, x_4) = \max\{D_{cl}(x_1, x_4), D_{cl}(x_2, x_4)\} = 3.61$
- $D_{cl}(\{x_1, x_2\}, x_5) = \max\{D_{cl}(x_1, x_5), D_{cl}(x_2, x_5)\} = 3.04$



## Example (cont'd)

With the complete link method, the distance matrix  $\mathbf{D}_{cl}$  becomes :

- $D_{cl}(\{\mathbf{x}_1, \mathbf{x}_2\}, \mathbf{x}_3) = \max\{D_{cl}(\mathbf{x}_1, \mathbf{x}_3), D_{cl}(\mathbf{x}_2, \mathbf{x}_3)\} = 2.5$
- $D_{cl}(\{\mathbf{x}_1, \mathbf{x}_2\}, \mathbf{x}_4) = \max\{D_{cl}(\mathbf{x}_1, \mathbf{x}_4), D_{cl}(\mathbf{x}_2, \mathbf{x}_4)\} = 3.61$
- $D_{cl}(\{\mathbf{x}_1, \mathbf{x}_2\}, \mathbf{x}_5) = \max\{D_{cl}(\mathbf{x}_1, \mathbf{x}_5), D_{cl}(\mathbf{x}_2, \mathbf{x}_5)\} = 3.04$

$$\mathbf{D}_{cl} = \begin{matrix} & \{\mathbf{x}_1, \mathbf{x}_2\} & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{x}_5 \\ \begin{matrix} \{\mathbf{x}_1, \mathbf{x}_2\} \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \end{matrix} & \begin{pmatrix} 0 & 2.5 & 3.61 & 3.04 \\ 2.5 & 0 & 1.12 & 1.41 \\ 3.61 & 1.12 & 0 & 1.5 \\ 3.04 & 1.41 & 1.5 & 0 \end{pmatrix} \end{matrix}$$

## Example (cont'd)

With the complete link method, the distance matrix  $\mathbf{D}_{cl}$  becomes :

- $D_{cl}(\{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_1, \mathbf{x}_2\}) = \max\{D_{cl}(\mathbf{x}_3, \mathbf{x}_1), D_{cl}(\mathbf{x}_3, \mathbf{x}_2), D_{cl}(\mathbf{x}_4, \mathbf{x}_1), D_{cl}(\mathbf{x}_4, \mathbf{x}_2)\} = \max\{D_{cl}(\mathbf{x}_3, \{\mathbf{x}_1, \mathbf{x}_2\}), D_{cl}(\mathbf{x}_4, \{\mathbf{x}_1, \mathbf{x}_2\})\} = 3.61$
- $D_{cl}(\{\mathbf{x}_3, \mathbf{x}_4\}, \mathbf{x}_5) = \max\{D_{cl}(\mathbf{x}_3, \mathbf{x}_5), D_{cl}(\mathbf{x}_4, \mathbf{x}_5)\} = 1.5$

## Example (cont'd)

With the complete link method, the distance matrix  $\mathbf{D}_{cl}$  becomes :

- $D_{cl}(\{\mathbf{x}_1, \mathbf{x}_2\}, \mathbf{x}_3) = \max\{D_{cl}(\mathbf{x}_1, \mathbf{x}_3), D_{cl}(\mathbf{x}_2, \mathbf{x}_3)\} = 2.5$
- $D_{cl}(\{\mathbf{x}_1, \mathbf{x}_2\}, \mathbf{x}_4) = \max\{D_{cl}(\mathbf{x}_1, \mathbf{x}_4), D_{cl}(\mathbf{x}_2, \mathbf{x}_4)\} = 3.61$
- $D_{cl}(\{\mathbf{x}_1, \mathbf{x}_2\}, \mathbf{x}_5) = \max\{D_{cl}(\mathbf{x}_1, \mathbf{x}_5), D_{cl}(\mathbf{x}_2, \mathbf{x}_5)\} = 3.04$

$$\mathbf{D}_{cl} = \begin{matrix} & \{\mathbf{x}_1, \mathbf{x}_2\} & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{x}_5 \\ \begin{matrix} \{\mathbf{x}_1, \mathbf{x}_2\} \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \end{matrix} & \begin{pmatrix} 0 & 2.5 & 3.61 & 3.04 \\ 2.5 & 0 & \mathbf{1.12} & 1.41 \\ 3.61 & \mathbf{1.12} & 0 & 1.5 \\ 3.04 & 1.41 & 1.5 & 0 \end{pmatrix} \end{matrix}$$

4 Merge  $\mathbf{x}_3$  and  $\mathbf{x}_4$

$$dend(h) = \begin{cases} \{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 0 \leq h < 0.5 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 0.5 \leq h < \mathbf{1.12} \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } \mathbf{1.12} \leq h \end{cases}$$

## Example (cont'd)

With the complete link method, the distance matrix  $\mathbf{D}_{cl}$  becomes :

- $D_{cl}(\{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_1, \mathbf{x}_2\}) = \max\{D_{cl}(\mathbf{x}_3, \mathbf{x}_1), D_{cl}(\mathbf{x}_3, \mathbf{x}_2), D_{cl}(\mathbf{x}_4, \mathbf{x}_1), D_{cl}(\mathbf{x}_4, \mathbf{x}_2)\} = \max\{D_{cl}(\mathbf{x}_3, \{\mathbf{x}_1, \mathbf{x}_2\}), D_{cl}(\mathbf{x}_4, \{\mathbf{x}_1, \mathbf{x}_2\})\} = 3.61$
- $D_{cl}(\{\mathbf{x}_3, \mathbf{x}_4\}, \mathbf{x}_5) = \max\{D_{cl}(\mathbf{x}_3, \mathbf{x}_5), D_{cl}(\mathbf{x}_4, \mathbf{x}_5)\} = 1.5$

$$\mathbf{D}_{cl} = \begin{matrix} & \{\mathbf{x}_1, \mathbf{x}_2\} & \{\mathbf{x}_3, \mathbf{x}_4\} & \mathbf{x}_5 \\ \begin{matrix} \{\mathbf{x}_1, \mathbf{x}_2\} \\ \{\mathbf{x}_3, \mathbf{x}_4\} \\ \mathbf{x}_5 \end{matrix} & \begin{pmatrix} 0 & 3.61 & 3.04 \\ 3.61 & 0 & 1.5 \\ 3.04 & 1.5 & 0 \end{pmatrix} \end{matrix}$$

## Example (cont'd)

With the complete link method, the distance matrix  $\mathbf{D}_{cl}$  becomes :

- $D_{cl}(\{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_1, \mathbf{x}_2\}) = \max\{D_{cl}(\mathbf{x}_3, \mathbf{x}_1), D_{cl}(\mathbf{x}_3, \mathbf{x}_2), D_{cl}(\mathbf{x}_4, \mathbf{x}_1), D_{cl}(\mathbf{x}_4, \mathbf{x}_2)\} = \max\{D_{cl}(\mathbf{x}_3, \{\mathbf{x}_1, \mathbf{x}_2\}), D_{cl}(\mathbf{x}_4, \{\mathbf{x}_1, \mathbf{x}_2\})\} = 3.61$
- $D_{cl}(\{\mathbf{x}_3, \mathbf{x}_4\}, \mathbf{x}_5) = \max\{D_{cl}(\mathbf{x}_3, \mathbf{x}_5), D_{cl}(\mathbf{x}_4, \mathbf{x}_5)\} = 1.5$

$$\mathbf{D}_{cl} = \begin{matrix} & \{\mathbf{x}_1, \mathbf{x}_2\} & \{\mathbf{x}_3, \mathbf{x}_4\} & \mathbf{x}_5 \\ \begin{matrix} \{\mathbf{x}_1, \mathbf{x}_2\} \\ \{\mathbf{x}_3, \mathbf{x}_4\} \\ \mathbf{x}_5 \end{matrix} & \begin{pmatrix} 0 & 3.61 & 3.04 \\ 3.61 & 0 & 1.5 \\ 3.04 & 1.5 & 0 \end{pmatrix} \end{matrix}$$

4 Merge  $\{\mathbf{x}_3, \mathbf{x}_4\}$  and  $\mathbf{x}_5$

$$dend(h) = \begin{cases} \{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 0 \leq h < 0.5 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 0.5 \leq h < 1.12 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 1.12 \leq h < 1.5 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} & \text{if } 1.5 \leq h \end{cases}$$

## Example (cont'd)

With the complete link method, the distance matrix  $\mathbf{D}_{cl}$  becomes :

- $D_{cl}(\{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\}, \{\mathbf{x}_1, \mathbf{x}_2\}) = \max\{D_{cl}(\{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_1, \mathbf{x}_2\}), D_{cl}(\mathbf{x}_5, \{\mathbf{x}_1, \mathbf{x}_2\})\} = 3.61$

$$\mathbf{D}_{cl} = \begin{matrix} & \{\mathbf{x}_1, \mathbf{x}_2\} & \{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} \\ \begin{matrix} \{\mathbf{x}_1, \mathbf{x}_2\} \\ \{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} \end{matrix} & \begin{pmatrix} 0 & 3.61 \\ 3.61 & 0 \end{pmatrix} \end{matrix}$$

4 Merge  $\{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\}$  and  $\{\mathbf{x}_1, \mathbf{x}_2\}$

$$dend(h) = \begin{cases} \{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 0 \leq h < 0.5 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 0.5 \leq h < 1.12 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 1.12 \leq h < 1.5 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} & \text{if } 1.5 \leq h < 3.61 \\ \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} & \text{if } 3.61 \leq h \end{cases}$$

## Example (cont'd)

With the complete link method, the distance matrix  $\mathbf{D}_{cl}$  becomes :

- $D_{cl}(\{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\}, \{\mathbf{x}_1, \mathbf{x}_2\}) = \max\{D_{cl}(\{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_1, \mathbf{x}_2\}), D_{cl}(\mathbf{x}_5, \{\mathbf{x}_1, \mathbf{x}_2\})\} = 3.61$

$$\mathbf{D}_{cl} = \begin{matrix} & \{\mathbf{x}_1, \mathbf{x}_2\} & \{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} \\ \begin{matrix} \{\mathbf{x}_1, \mathbf{x}_2\} \\ \{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} \end{matrix} & \begin{pmatrix} 0 & 3.61 \\ 3.61 & 0 \end{pmatrix} \end{matrix}$$

## Example (cont'd)

With the complete link method, the distance matrix  $\mathbf{D}_{cl}$  becomes :

- $D_{cl}(\{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\}, \{\mathbf{x}_1, \mathbf{x}_2\}) = \max\{D_{cl}(\{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_1, \mathbf{x}_2\}), D_{cl}(\mathbf{x}_5, \{\mathbf{x}_1, \mathbf{x}_2\})\} = 3.61$

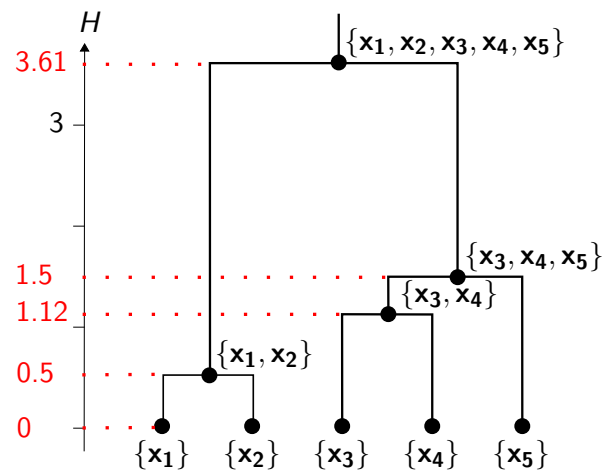
$$\mathbf{D}_{cl} = \begin{matrix} & \{\mathbf{x}_1, \mathbf{x}_2\} & \{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} \\ \begin{matrix} \{\mathbf{x}_1, \mathbf{x}_2\} \\ \{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} \end{matrix} & \begin{pmatrix} 0 & 3.61 \\ 3.61 & 0 \end{pmatrix} \end{matrix}$$

4 Merge  $\{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\}$  and  $\{\mathbf{x}_1, \mathbf{x}_2\}$

$$dend(h) = \begin{cases} \{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 0 \leq h < 0.5 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 0.5 \leq h < 1.12 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 1.12 \leq h < 1.5 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} & \text{if } 1.5 \leq h < 3.61 \\ \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} & \text{if } 3.61 \leq h \end{cases}$$

## Example (cont'd)

The dendrogram :



## Group average method

- Proposed by McQuitty in 1967
- Also referred as UPGMA for "Unweighted Pair Group Method using Arithmetic mean"

## Group average method

- Proposed by McQuitty in 1967

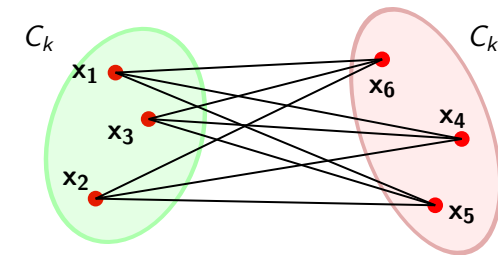
- Proposed by McQuitty in 1967
- Also referred as UPGMA for "Unweighted Pair Group Method using Arithmetic mean"
- According to the LW formula we have :

$$D_{upgma}(C_k, C_l \cup C_{l'}) = \frac{|C_l|}{|C_l| + |C_{l'}|} D_{upgma}(C_k, C_l) + \frac{|C_{l'}|}{|C_l| + |C_{l'}|} D_{upgma}(C_k, C_{l'})$$

## Group average method (cont'd)

Suppose that  $C_k$  and  $C_{k'}$  are two nonempty and nonoverlapping clusters and  $D$  the distance function by which the dissimilarity matrix is computed, then the distance between  $C_k$  and  $C_{k'}$  is defined as follows :

$$D_{upgma}(C_k, C_{k'}) = \frac{1}{|C_k||C_{k'}|} \sum_{x \in C_k, y \in C_{k'}} D(x, y)$$



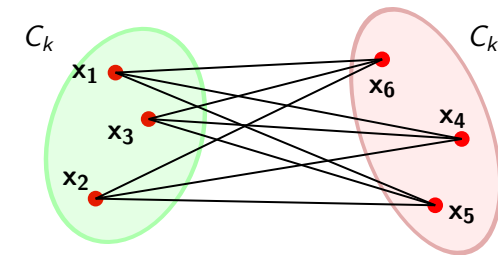
## Weighted group average method

- Proposed by McQuitty in 1966
- Also referred as WPGMA for “Weighted Pair Group Method using Arithmetic mean”

## Group average method (cont'd)

Suppose that  $C_k$  and  $C_{k'}$  are two nonempty and nonoverlapping clusters and  $D$  the distance function by which the dissimilarity matrix is computed, then the distance between  $C_k$  and  $C_{k'}$  is defined as follows :

$$D_{upgma}(C_k, C_{k'}) = \frac{1}{|C_k||C_{k'}|} \sum_{x \in C_k, y \in C_{k'}} D(x, y)$$



## Weighted group average method

- Proposed by McQuitty in 1966
- Also referred as WPGMA for “Weighted Pair Group Method using Arithmetic mean”
- According to the LW formula we have :

$$D_{wpgma}(C_k, C_l \cup C_{l'}) = \frac{1}{2} D_{wpgma}(C_k, C_l) + \frac{1}{2} D_{wpgma}(C_k, C_{l'})$$

## Weighted group average method

- Proposed by McQuitty in 1966
- Also referred as WPGMA for “Weighted Pair Group Method using Arithmetic mean”
- According to the LW formula we have :

$$D_{wpgma}(C_k, C_l \cup C_{l'}) = \frac{1}{2}D_{wpgma}(C_k, C_l) + \frac{1}{2}D_{wpgma}(C_k, C_{l'})$$

- Unlike UPGMA, WPGMA will give more weights to small clusters when updating the distances after merging two clusters.

## Ward's method

- Proposed by Ward in 1963

## Weighted group average method

- Proposed by McQuitty in 1966
- Also referred as WPGMA for “Weighted Pair Group Method using Arithmetic mean”
- According to the LW formula we have :

$$D_{wpgma}(C_k, C_l \cup C_{l'}) = \frac{1}{2}D_{wpgma}(C_k, C_l) + \frac{1}{2}D_{wpgma}(C_k, C_{l'})$$

- Unlike UPGMA, WPGMA will give more weights to small clusters when updating the distances after merging two clusters.

**Exercise 7 :** Apply the WPGMA method to the previous example.

## Ward's method

- Proposed by Ward in 1963
- Based on the loss of information quantified in terms of an error sum of squares criterion (*ESS*). Given a group of data points  $C_k$  the *ESS* associated to this cluster is :

$$\begin{aligned} ESS(C_k) &= \sum_{\mathbf{x} \in C_k} \|\mathbf{x} - \mu(C_k)\|^2 \\ &= \sum_{\mathbf{x} \in C_k} (\mathbf{x} - \mu(C_k))^T (\mathbf{x} - \mu(C_k)) \\ &= \sum_{\mathbf{x} \in C_k} \mathbf{x}^T \mathbf{x} - |C_k| \mu(C_k)^T \mu(C_k) \end{aligned}$$

where  $\mu(C_k) = \frac{1}{|C_k|} \sum_{\mathbf{x} \in C_k} \mathbf{x}$ , is the mean vector of cluster  $C_k$

## Ward's method

- Proposed by Ward in 1963
- Based on the loss of information quantified in terms of an error sum of squares criterion (*ESS*). Given a group of data points  $C_k$  the *ESS* associated to this cluster is :

$$\begin{aligned} ESS(C_k) &= \sum_{\mathbf{x} \in C_k} \|\mathbf{x} - \mu(C_k)\|^2 \\ &= \sum_{\mathbf{x} \in C_k} (\mathbf{x} - \mu(C_k))^\top (\mathbf{x} - \mu(C_k)) \\ &= \sum_{\mathbf{x} \in C_k} \mathbf{x}^\top \mathbf{x} - |C_k| \mu(C_k)^\top \mu(C_k) \end{aligned}$$

where  $\mu(C_k) = \frac{1}{|C_k|} \sum_{\mathbf{x} \in C_k} \mathbf{x}$ , is the mean vector of cluster  $C_k$

- At each step of Ward's method, the union of every possible pair of clusters is considered and we merge the one that leads to the minimum *ESS*

## Ward's method (cont'd)

- If the squared euclidean distance is used ( $D(\mathbf{x}, \mathbf{y}) = D_{eucl}^2(\mathbf{x}, \mathbf{y})$ ) then the dissimilarity matrix can be updated as follows :

$$\begin{aligned} D_{ward}(C_k, C_l \cup C_{l'}) &= \frac{|C_k| + |C_l|}{|C_k| + |C_l| + |C_{l'}|} D_{ward}(C_k, C_l) \\ &\quad + \frac{|C_k| + |C_{l'}|}{|C_k| + |C_l| + |C_{l'}|} D_{ward}(C_k, C_{l'}) \\ &\quad - \frac{|C_k|}{|C_k| + |C_l| + |C_{l'}|} D_{ward}(C_l, C_{l'}) \end{aligned}$$

## Ward's method (cont'd)

- If the squared euclidean distance is used ( $D(\mathbf{x}, \mathbf{y}) = D_{eucl}^2(\mathbf{x}, \mathbf{y})$ ) then the dissimilarity matrix can be updated as follows :

$$\begin{aligned} D_{ward}(C_k, C_l \cup C_{l'}) &= \frac{|C_k| + |C_l|}{|C_k| + |C_l| + |C_{l'}|} D_{ward}(C_k, C_l) \\ &\quad + \frac{|C_k| + |C_{l'}|}{|C_k| + |C_l| + |C_{l'}|} D_{ward}(C_k, C_{l'}) \\ &\quad - \frac{|C_k|}{|C_k| + |C_l| + |C_{l'}|} D_{ward}(C_l, C_{l'}) \end{aligned}$$

- Sketch of proof :** Let denote  $\Delta ESS(C_k, C_l \cup C_{l'})$  the increase of *ESS* when merging  $C_k$  with  $C_{k'} = C_l \cup C_{l'}$ .  
If  $D(\mathbf{x}, \mathbf{y}) = D_{eucl}^2(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^2$  then we can show that :

$$\Delta ESS(C_k, C_l \cup C_{l'}) = \frac{1}{2} D_{ward}(C_k, C_l \cup C_{l'})$$

Thus, picking the smallest  $D_{ward}(C_k, C_l \cup C_{l'})$  at each iteration leads to the merging of clusters that minimizes the loss of information

## Example

We consider the same example as previously. The starting distance matrix  $\mathbf{D}$  is here the **squared** euclidean distance matrix between points :

$$\mathbf{D} = \mathbf{D}_{eucl}^2 = \mathbf{D}_{ward} = \begin{matrix} & \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{x}_5 \\ \mathbf{x}_1 & 0 & 0.25 & 5 & 11.25 & 9 \\ \mathbf{x}_2 & 0.25 & 0 & 6.25 & 13 & 9.35 \\ \mathbf{x}_3 & 5 & 6.25 & 0 & 1.25 & 2 \\ \mathbf{x}_4 & 11.25 & 13 & 1.25 & 0 & 2.25 \\ \mathbf{x}_5 & 9 & 9.25 & 2 & 2.25 & 0 \end{matrix}$$

## Example

We consider the same example as previously. The starting distance matrix  $\mathbf{D}$  is here the **squared** euclidean distance matrix between points :

$$\mathbf{D} = \mathbf{D}_{\text{eucl}}^2 = \mathbf{D}_{\text{ward}} = \begin{matrix} & \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{x}_5 \\ \mathbf{x}_1 & 0 & 0.25 & 5 & 11.25 & 9 \\ \mathbf{x}_2 & 0.25 & 0 & 6.25 & 13 & 9.35 \\ \mathbf{x}_3 & 5 & 6.25 & 0 & 1.25 & 2 \\ \mathbf{x}_4 & 11.25 & 13 & 1.25 & 0 & 2.25 \\ \mathbf{x}_5 & 9 & 9.25 & 2 & 2.25 & 0 \end{matrix}$$

$$2 \quad \text{dend}(h) = \{ \{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} \quad \text{if } 0 \leq h$$

4 Merge  $\mathbf{x}_1$  and  $\mathbf{x}_2$

$$\text{dend}(h) = \begin{cases} \{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 0 \leq h < 0.25 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 0.25 \leq h \end{cases}$$

## Example (cont'd)

- $D_{\text{ward}}(\{\mathbf{x}_1, \mathbf{x}_2\}, \mathbf{x}_3) = \frac{2}{3}(D_{\text{ward}}(\mathbf{x}_1, \mathbf{x}_3) + D_{\text{ward}}(\mathbf{x}_2, \mathbf{x}_3)) - \frac{1}{3}D_{\text{ward}}(\mathbf{x}_1, \mathbf{x}_2) = 7.42$
- $D_{\text{ward}}(\{\mathbf{x}_1, \mathbf{x}_2\}, \mathbf{x}_4) = \frac{2}{3}(D_{\text{ward}}(\mathbf{x}_1, \mathbf{x}_4) + D_{\text{ward}}(\mathbf{x}_2, \mathbf{x}_4)) - \frac{1}{3}D_{\text{ward}}(\mathbf{x}_1, \mathbf{x}_2) = 16.08$
- $D_{\text{ward}}(\{\mathbf{x}_1, \mathbf{x}_2\}, \mathbf{x}_5) = \frac{2}{3}(D_{\text{ward}}(\mathbf{x}_1, \mathbf{x}_5) + D_{\text{ward}}(\mathbf{x}_2, \mathbf{x}_5)) - \frac{1}{3}D_{\text{ward}}(\mathbf{x}_1, \mathbf{x}_2) = 12.08$

## Example (cont'd)

- $D_{\text{ward}}(\{\mathbf{x}_1, \mathbf{x}_2\}, \mathbf{x}_3) = \frac{2}{3}(D_{\text{ward}}(\mathbf{x}_1, \mathbf{x}_3) + D_{\text{ward}}(\mathbf{x}_2, \mathbf{x}_3)) - \frac{1}{3}D_{\text{ward}}(\mathbf{x}_1, \mathbf{x}_2) = 7.42$
- $D_{\text{ward}}(\{\mathbf{x}_1, \mathbf{x}_2\}, \mathbf{x}_4) = \frac{2}{3}(D_{\text{ward}}(\mathbf{x}_1, \mathbf{x}_4) + D_{\text{ward}}(\mathbf{x}_2, \mathbf{x}_4)) - \frac{1}{3}D_{\text{ward}}(\mathbf{x}_1, \mathbf{x}_2) = 16.08$
- $D_{\text{ward}}(\{\mathbf{x}_1, \mathbf{x}_2\}, \mathbf{x}_5) = \frac{2}{3}(D_{\text{ward}}(\mathbf{x}_1, \mathbf{x}_5) + D_{\text{ward}}(\mathbf{x}_2, \mathbf{x}_5)) - \frac{1}{3}D_{\text{ward}}(\mathbf{x}_1, \mathbf{x}_2) = 12.08$

$$\mathbf{D}_{\text{ward}} = \begin{matrix} & \{\mathbf{x}_1, \mathbf{x}_2\} & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{x}_5 \\ \{\mathbf{x}_1, \mathbf{x}_2\} & 0 & 7.42 & 16.08 & 12.08 \\ \mathbf{x}_3 & 7.42 & 0 & 1.25 & 2 \\ \mathbf{x}_4 & 16.08 & 1.25 & 0 & 2.25 \\ \mathbf{x}_5 & 12.08 & 2 & 2.25 & 0 \end{matrix}$$

## Example (cont'd)

- $D_{\text{ward}}(\{\mathbf{x}_1, \mathbf{x}_2\}, \mathbf{x}_3) = \frac{2}{3}(D_{\text{ward}}(\mathbf{x}_1, \mathbf{x}_3) + D_{\text{ward}}(\mathbf{x}_2, \mathbf{x}_3)) - \frac{1}{3}D_{\text{ward}}(\mathbf{x}_1, \mathbf{x}_2) = 7.42$
- $D_{\text{ward}}(\{\mathbf{x}_1, \mathbf{x}_2\}, \mathbf{x}_4) = \frac{2}{3}(D_{\text{ward}}(\mathbf{x}_1, \mathbf{x}_4) + D_{\text{ward}}(\mathbf{x}_2, \mathbf{x}_4)) - \frac{1}{3}D_{\text{ward}}(\mathbf{x}_1, \mathbf{x}_2) = 16.08$
- $D_{\text{ward}}(\{\mathbf{x}_1, \mathbf{x}_2\}, \mathbf{x}_5) = \frac{2}{3}(D_{\text{ward}}(\mathbf{x}_1, \mathbf{x}_5) + D_{\text{ward}}(\mathbf{x}_2, \mathbf{x}_5)) - \frac{1}{3}D_{\text{ward}}(\mathbf{x}_1, \mathbf{x}_2) = 12.08$

$$\mathbf{D}_{\text{ward}} = \begin{matrix} & \{\mathbf{x}_1, \mathbf{x}_2\} & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{x}_5 \\ \{\mathbf{x}_1, \mathbf{x}_2\} & 0 & 7.42 & 16.08 & 12.08 \\ \mathbf{x}_3 & 7.42 & 0 & 1.25 & 2 \\ \mathbf{x}_4 & 16.08 & 1.25 & 0 & 2.25 \\ \mathbf{x}_5 & 12.08 & 2 & 2.25 & 0 \end{matrix}$$

4 Merge  $\mathbf{x}_3$  and  $\mathbf{x}_4$

$$\text{dend}(h) = \begin{cases} \{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 0 \leq h < 0.25 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 0.25 \leq h < 1.25 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 1.25 \leq h \end{cases}$$

### Example (cont'd)

- $D_{ward}(\{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_1, \mathbf{x}_2\}) = \frac{3}{4}(D_{ward}(\mathbf{x}_3, \{\mathbf{x}_1, \mathbf{x}_2\}) + D_{ward}(\mathbf{x}_4, \{\mathbf{x}_1, \mathbf{x}_2\})) - \frac{2}{4}D_{ward}(\mathbf{x}_3, \mathbf{x}_4) = \frac{3}{4}(7.42 + 16.08) - \frac{2}{4}(1.25) = 17$
- $D_{ward}(\{\mathbf{x}_3, \mathbf{x}_4\}, \mathbf{x}_5) = \frac{2}{3}(D_{ward}(\mathbf{x}_3, \mathbf{x}_5) + D_{ward}(\mathbf{x}_4, \mathbf{x}_5)) - \frac{1}{3}D_{ward}(\mathbf{x}_3, \mathbf{x}_4) = 2.42$

$$D_{ward} = \begin{matrix} & \{\mathbf{x}_1, \mathbf{x}_2\} & \{\mathbf{x}_3, \mathbf{x}_4\} & \mathbf{x}_5 \\ \begin{matrix} \{\mathbf{x}_1, \mathbf{x}_2\} \\ \{\mathbf{x}_3, \mathbf{x}_4\} \\ \mathbf{x}_5 \end{matrix} & \begin{pmatrix} 0 & 17 & 12.08 \\ 17 & 0 & 2.42 \\ 12.08 & 2.42 & 0 \end{pmatrix} \end{matrix}$$

### Example (cont'd)

- $D_{ward}(\{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_1, \mathbf{x}_2\}) = \frac{3}{4}(D_{ward}(\mathbf{x}_3, \{\mathbf{x}_1, \mathbf{x}_2\}) + D_{ward}(\mathbf{x}_4, \{\mathbf{x}_1, \mathbf{x}_2\})) - \frac{2}{4}D_{ward}(\mathbf{x}_3, \mathbf{x}_4) = \frac{3}{4}(7.42 + 16.08) - \frac{2}{4}(1.25) = 17$
- $D_{ward}(\{\mathbf{x}_3, \mathbf{x}_4\}, \mathbf{x}_5) = \frac{2}{3}(D_{ward}(\mathbf{x}_3, \mathbf{x}_5) + D_{ward}(\mathbf{x}_4, \mathbf{x}_5)) - \frac{1}{3}D_{ward}(\mathbf{x}_3, \mathbf{x}_4) = 2.42$

$$D_{ward} = \begin{matrix} & \{\mathbf{x}_1, \mathbf{x}_2\} & \{\mathbf{x}_3, \mathbf{x}_4\} & \mathbf{x}_5 \\ \begin{matrix} \{\mathbf{x}_1, \mathbf{x}_2\} \\ \{\mathbf{x}_3, \mathbf{x}_4\} \\ \mathbf{x}_5 \end{matrix} & \begin{pmatrix} 0 & 17 & 12.08 \\ 17 & 0 & 2.42 \\ 12.08 & 2.42 & 0 \end{pmatrix} \end{matrix}$$

4 Merge  $\{\mathbf{x}_3, \mathbf{x}_4\}$  and  $\mathbf{x}_5$

$$dend(h) = \begin{cases} \{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 0 \leq h < 0.25 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 0.25 \leq h < 1.25 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 1.25 \leq h < 2.42 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} & \text{if } 2.42 \leq h \end{cases}$$

### Example (cont'd)

- $D_{ward}(\{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_1, \mathbf{x}_2\}) = \frac{3}{4}(D_{ward}(\mathbf{x}_3, \{\mathbf{x}_1, \mathbf{x}_2\}) + D_{ward}(\mathbf{x}_4, \{\mathbf{x}_1, \mathbf{x}_2\})) - \frac{2}{4}D_{ward}(\mathbf{x}_3, \mathbf{x}_4) = \frac{3}{4}(7.42 + 16.08) - \frac{2}{4}(1.25) = 17$
- $D_{ward}(\{\mathbf{x}_3, \mathbf{x}_4\}, \mathbf{x}_5) = \frac{2}{3}(D_{ward}(\mathbf{x}_3, \mathbf{x}_5) + D_{ward}(\mathbf{x}_4, \mathbf{x}_5)) - \frac{1}{3}D_{ward}(\mathbf{x}_3, \mathbf{x}_4) = 2.42$

$$D_{ward} = \begin{matrix} & \{\mathbf{x}_1, \mathbf{x}_2\} & \{\mathbf{x}_3, \mathbf{x}_4\} & \mathbf{x}_5 \\ \begin{matrix} \{\mathbf{x}_1, \mathbf{x}_2\} \\ \{\mathbf{x}_3, \mathbf{x}_4\} \\ \mathbf{x}_5 \end{matrix} & \begin{pmatrix} 0 & 17 & 12.08 \\ 17 & 0 & 2.42 \\ 12.08 & 2.42 & 0 \end{pmatrix} \end{matrix}$$

### Example (cont'd)

- $D_{ward}(\{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\}, \{\mathbf{x}_1, \mathbf{x}_2\}) = \frac{4}{5}D_{ward}(\{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_1, \mathbf{x}_2\}) + \frac{3}{5}D_{ward}(\mathbf{x}_5, \{\mathbf{x}_1, \mathbf{x}_2\}) - \frac{2}{5}D_{ward}(\{\mathbf{x}_3, \mathbf{x}_4\}, \mathbf{x}_5) = \frac{4}{5}(17) + \frac{3}{5}(12.08) - \frac{2}{5}(2.42) = 19.88$



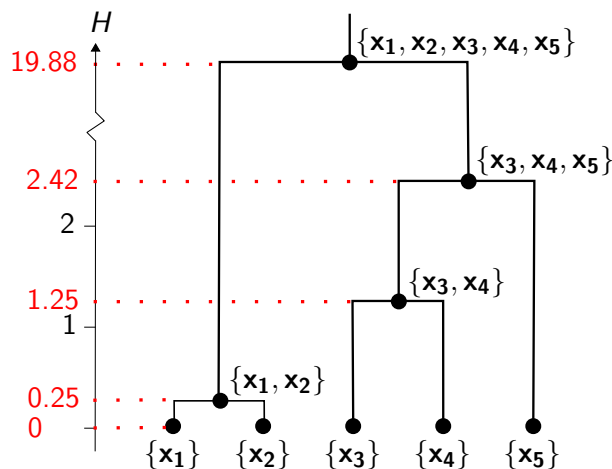
### Example (cont'd)

- $$D_{ward}(\{x_3, x_4, x_5\}, \{x_1, x_2\}) = \frac{4}{5}D_{ward}(\{x_3, x_4\}, \{x_1, x_2\}) + \frac{3}{5}D_{ward}(x_5, \{x_1, x_2\}) - \frac{2}{5}D_{ward}(\{x_3, x_4\}, x_5) = \frac{4}{5}(17) + \frac{3}{5}(12.08) - \frac{2}{5}(2.42) = 19.88$$

$$D_{ward} = \begin{matrix} & \{x_1, x_2\} & \{x_3, x_4, x_5\} \\ \begin{matrix} \{x_1, x_2\} \\ \{x_3, x_4, x_5\} \end{matrix} & \begin{pmatrix} 0 & 19.88 \\ 19.88 & 0 \end{pmatrix} \end{matrix}$$

### Example (cont'd)

The dendrogram :



### Example (cont'd)

- $$D_{ward}(\{x_3, x_4, x_5\}, \{x_1, x_2\}) = \frac{4}{5}D_{ward}(\{x_3, x_4\}, \{x_1, x_2\}) + \frac{3}{5}D_{ward}(x_5, \{x_1, x_2\}) - \frac{2}{5}D_{ward}(\{x_3, x_4\}, x_5) = \frac{4}{5}(17) + \frac{3}{5}(12.08) - \frac{2}{5}(2.42) = 19.88$$

$$D_{ward} = \begin{matrix} & \{x_1, x_2\} & \{x_3, x_4, x_5\} \\ \begin{matrix} \{x_1, x_2\} \\ \{x_3, x_4, x_5\} \end{matrix} & \begin{pmatrix} 0 & 19.88 \\ 19.88 & 0 \end{pmatrix} \end{matrix}$$

4 Merge  $\{x_1, x_2\}$  and  $\{x_3, x_4, x_5\}$

$$dend(h) = \begin{cases} \{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\} & \text{if } 0 \leq h < 0.25 \\ \{x_1, x_2\}, \{x_3\}, \{x_4\}, \{x_5\} & \text{if } 0.25 \leq h < 1.25 \\ \{x_1, x_2\}, \{x_3, x_4\}, \{x_5\} & \text{if } 1.25 \leq h < 2.42 \\ \{x_1, x_2\}, \{x_3, x_4, x_5\} & \text{if } 2.42 \leq h < 19.88 \\ \{x_1, x_2, x_3, x_4, x_5\} & \text{if } 19.88 \leq h \end{cases}$$

### The centroid method

- Proposed by Gower in 1967

## The centroid method

- Proposed by Gower in 1967
- Following the LW formula, this approach is related to the updating rule below :

$$D_{cent}(C_k, C_l \cup C_{l'}) = \frac{|C_l|}{|C_l| + |C_{l'}|} D_{cent}(C_k, C_l) + \frac{|C_{l'}|}{|C_l| + |C_{l'}|} D_{cent}(C_k, C_{l'}) - \frac{|C_l||C_{l'}|}{(|C_l| + |C_{l'}|)^2} D_{cent}(C_l, C_{l'})$$

## The centroid method (cont'd)

- Suppose that  $C_k$  and  $C_{k'}$  are two nonempty and nonoverlapping clusters and  $D$  is the distance function by which the dissimilarity matrix is computed then the  $D_{sl}$  distance between  $C_k$  and  $C_{k'}$  can be defined as follows :

$$D_{cent}(C_k, C_{k'}) = \frac{1}{|C_k||C_{k'}|} \sum_{\mathbf{x} \in C_k, \mathbf{y} \in C_{k'}} D(\mathbf{x}, \mathbf{y}) - \frac{1}{2|C_k|^2} \sum_{\mathbf{x}, \mathbf{y} \in C_k} D(\mathbf{x}, \mathbf{y}) - \frac{1}{2|C_{k'}|^2} \sum_{\mathbf{x}, \mathbf{y} \in C_{k'}} D(\mathbf{x}, \mathbf{y})$$

- If  $D(\mathbf{x}, \mathbf{y}) = D_{eucl}^2(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^2$  then  $D_{cent}(C_k, C_{k'})$  is the squared distance between the centroids of  $C_k$  and  $C_{k'}$  and we have :

$$D_{cent}(C_k, C_{k'}) = D_{eucl}^2(\mu(C_k), \mu(C_{k'}))$$

## The centroid method (cont'd)

- Suppose that  $C_k$  and  $C_{k'}$  are two nonempty and nonoverlapping clusters and  $D$  is the distance function by which the dissimilarity matrix is computed then the  $D_{sl}$  distance between  $C_k$  and  $C_{k'}$  can be defined as follows :

$$D_{cent}(C_k, C_{k'}) = \frac{1}{|C_k||C_{k'}|} \sum_{\mathbf{x} \in C_k, \mathbf{y} \in C_{k'}} D(\mathbf{x}, \mathbf{y}) - \frac{1}{2|C_k|^2} \sum_{\mathbf{x}, \mathbf{y} \in C_k} D(\mathbf{x}, \mathbf{y}) - \frac{1}{2|C_{k'}|^2} \sum_{\mathbf{x}, \mathbf{y} \in C_{k'}} D(\mathbf{x}, \mathbf{y})$$

## The centroid method (cont'd)

- Suppose that  $C_k$  and  $C_{k'}$  are two nonempty and nonoverlapping clusters and  $D$  is the distance function by which the dissimilarity matrix is computed then the  $D_{sl}$  distance between  $C_k$  and  $C_{k'}$  can be defined as follows :

$$D_{cent}(C_k, C_{k'}) = \frac{1}{|C_k||C_{k'}|} \sum_{\mathbf{x} \in C_k, \mathbf{y} \in C_{k'}} D(\mathbf{x}, \mathbf{y}) - \frac{1}{2|C_k|^2} \sum_{\mathbf{x}, \mathbf{y} \in C_k} D(\mathbf{x}, \mathbf{y}) - \frac{1}{2|C_{k'}|^2} \sum_{\mathbf{x}, \mathbf{y} \in C_{k'}} D(\mathbf{x}, \mathbf{y})$$

- If  $D(\mathbf{x}, \mathbf{y}) = D_{eucl}^2(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^2$  then  $D_{cent}(C_k, C_{k'})$  is the squared distance between the centroids of  $C_k$  and  $C_{k'}$  and we have :

$$D_{cent}(C_k, C_{k'}) = D_{eucl}^2(\mu(C_k), \mu(C_{k'}))$$

**Exercise 8 :** Show the last relationship.

## Example

We consider the same example as previously. We chose as the starting distance matrix  $\mathbf{D}$  the euclidean distance matrix between points :

$$\mathbf{D} = \mathbf{D}_{\text{eucl}} = \mathbf{D}_{\text{cent}} = \begin{matrix} & \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{x}_5 \\ \mathbf{x}_1 & \left( \begin{array}{ccccc} 0 & 0.5 & 2.24 & 3.35 & 3 \\ 0.5 & 0 & 2.5 & 3.61 & 3.04 \\ 2.24 & 2.5 & 0 & 1.12 & 1.41 \\ 3.35 & 3.61 & 1.12 & 0 & 1.5 \\ 3 & 3.04 & 1.41 & 1.5 & 0 \end{array} \right) \end{matrix}$$

## Example (cont'd)

- $D_{\text{cent}}(\{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_1, \mathbf{x}_2\}) = \frac{1}{2}(D_{\text{cent}}(\mathbf{x}_3, \{\mathbf{x}_1, \mathbf{x}_2\}) + D_{\text{cent}}(\mathbf{x}_4, \{\mathbf{x}_1, \mathbf{x}_2\})) - \frac{1}{4}D_{\text{cent}}(\mathbf{x}_3, \mathbf{x}_4) = 2.52$
- $D_{\text{cent}}(\{\mathbf{x}_3, \mathbf{x}_4\}, \mathbf{x}_5) = \frac{1}{2}(D_{\text{cent}}(\mathbf{x}_3, \mathbf{x}_5) + D_{\text{cent}}(\mathbf{x}_4, \mathbf{x}_5)) - \frac{1}{4}D_{\text{cent}}(\mathbf{x}_3, \mathbf{x}_4) = 1.175$

## Example

We consider the same example as previously. We chose as the starting distance matrix  $\mathbf{D}$  the euclidean distance matrix between points :

$$\mathbf{D} = \mathbf{D}_{\text{eucl}} = \mathbf{D}_{\text{cent}} = \begin{matrix} & \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{x}_5 \\ \mathbf{x}_1 & \left( \begin{array}{ccccc} 0 & \mathbf{0.5} & 2.24 & 3.35 & 3 \\ \mathbf{0.5} & 0 & 2.5 & 3.61 & 3.04 \\ 2.24 & 2.5 & 0 & 1.12 & 1.41 \\ 3.35 & 3.61 & 1.12 & 0 & 1.5 \\ 3 & 3.04 & 1.41 & 1.5 & 0 \end{array} \right) \end{matrix}$$

$$2 \quad dend(h) = \{ \{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} \quad \text{if } 0 \leq h$$

$$4 \quad \text{Merge } \mathbf{x}_1 \text{ and } \mathbf{x}_2$$

$$dend(h) = \begin{cases} \{ \{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 0 \leq h < \mathbf{0.5} \\ \{ \mathbf{x}_1, \mathbf{x}_2 \}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } \mathbf{0.5} \leq h \end{cases}$$

## Example (cont'd)

- $D_{\text{cent}}(\{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_1, \mathbf{x}_2\}) = \frac{1}{2}(D_{\text{cent}}(\mathbf{x}_3, \{\mathbf{x}_1, \mathbf{x}_2\}) + D_{\text{cent}}(\mathbf{x}_4, \{\mathbf{x}_1, \mathbf{x}_2\})) - \frac{1}{4}D_{\text{cent}}(\mathbf{x}_3, \mathbf{x}_4) = 2.52$
- $D_{\text{cent}}(\{\mathbf{x}_3, \mathbf{x}_4\}, \mathbf{x}_5) = \frac{1}{2}(D_{\text{cent}}(\mathbf{x}_3, \mathbf{x}_5) + D_{\text{cent}}(\mathbf{x}_4, \mathbf{x}_5)) - \frac{1}{4}D_{\text{cent}}(\mathbf{x}_3, \mathbf{x}_4) = 1.175$

$$\mathbf{D}_{\text{cent}} = \begin{matrix} & \{\mathbf{x}_1, \mathbf{x}_2\} & \{\mathbf{x}_3, \mathbf{x}_4\} & \mathbf{x}_5 \\ \{\mathbf{x}_1, \mathbf{x}_2\} & \left( \begin{array}{ccc} 0 & 2.52 & 2.895 \\ 2.52 & 0 & 1.175 \\ 2.895 & 1.175 & 0 \end{array} \right) \\ \{\mathbf{x}_3, \mathbf{x}_4\} & & & \\ \mathbf{x}_5 & & & \end{matrix}$$

## Example (cont'd)

- $D_{cent}(\{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_1, \mathbf{x}_2\}) = \frac{1}{2}(D_{cent}(\mathbf{x}_3, \{\mathbf{x}_1, \mathbf{x}_2\}) + D_{cent}(\mathbf{x}_4, \{\mathbf{x}_1, \mathbf{x}_2\})) - \frac{1}{4}D_{cent}(\mathbf{x}_3, \mathbf{x}_4) = 2.52$
- $D_{cent}(\{\mathbf{x}_3, \mathbf{x}_4\}, \mathbf{x}_5) = \frac{1}{2}(D_{cent}(\mathbf{x}_3, \mathbf{x}_5) + D_{cent}(\mathbf{x}_4, \mathbf{x}_5)) - \frac{1}{4}D_{cent}(\mathbf{x}_3, \mathbf{x}_4) = 1.175$

$$\mathbf{D}_{cent} = \begin{matrix} & \{\mathbf{x}_1, \mathbf{x}_2\} & \{\mathbf{x}_3, \mathbf{x}_4\} & \mathbf{x}_5 \\ \begin{matrix} \{\mathbf{x}_1, \mathbf{x}_2\} \\ \{\mathbf{x}_3, \mathbf{x}_4\} \\ \mathbf{x}_5 \end{matrix} & \begin{pmatrix} 0 & 2.52 & 2.895 \\ 2.52 & 0 & \mathbf{1.175} \\ 2.895 & \mathbf{1.175} & 0 \end{pmatrix} \end{matrix}$$

4 Merge  $\{\mathbf{x}_3, \mathbf{x}_4\}$  with  $\mathbf{x}_5$

$$dend(h) = \begin{cases} \{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 0 \leq h < 0.5 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 0.5 \leq h < 1.12 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 1.12 \leq h < \mathbf{1.175} \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} & \text{if } \mathbf{1.175} \leq h \end{cases}$$

## Example (cont'd)

- $D_{cent}(\{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\}, \{\mathbf{x}_1, \mathbf{x}_2\}) = \frac{2}{3}D_{cent}(\{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_1, \mathbf{x}_2\}) + \frac{1}{3}D_{cent}(\mathbf{x}_5, \{\mathbf{x}_1, \mathbf{x}_2\}) - \frac{2}{9}D_{cent}(\{\mathbf{x}_3, \mathbf{x}_4\}, \mathbf{x}_5) = 2.38$

## Example (cont'd)

- $D_{cent}(\{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\}, \{\mathbf{x}_1, \mathbf{x}_2\}) = \frac{2}{3}D_{cent}(\{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_1, \mathbf{x}_2\}) + \frac{1}{3}D_{cent}(\mathbf{x}_5, \{\mathbf{x}_1, \mathbf{x}_2\}) - \frac{2}{9}D_{cent}(\{\mathbf{x}_3, \mathbf{x}_4\}, \mathbf{x}_5) = 2.38$

$$\mathbf{D}_{cent} = \begin{matrix} & \{\mathbf{x}_1, \mathbf{x}_2\} & \{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} \\ \begin{matrix} \{\mathbf{x}_1, \mathbf{x}_2\} \\ \{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} \end{matrix} & \begin{pmatrix} 0 & 2.38 \\ 2.38 & 0 \end{pmatrix} \end{matrix}$$

## Example (cont'd)

- $D_{cent}(\{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\}, \{\mathbf{x}_1, \mathbf{x}_2\}) = \frac{2}{3}D_{cent}(\{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_1, \mathbf{x}_2\}) + \frac{1}{3}D_{cent}(\mathbf{x}_5, \{\mathbf{x}_1, \mathbf{x}_2\}) - \frac{2}{9}D_{cent}(\{\mathbf{x}_3, \mathbf{x}_4\}, \mathbf{x}_5) = 2.38$

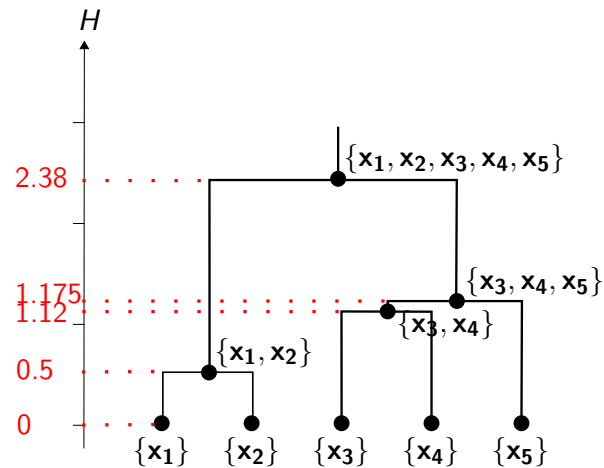
$$\mathbf{D}_{cent} = \begin{matrix} & \{\mathbf{x}_1, \mathbf{x}_2\} & \{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} \\ \begin{matrix} \{\mathbf{x}_1, \mathbf{x}_2\} \\ \{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} \end{matrix} & \begin{pmatrix} 0 & \mathbf{2.38} \\ \mathbf{2.38} & 0 \end{pmatrix} \end{matrix}$$

4 Merge  $\{\mathbf{x}_1, \mathbf{x}_2\}$  and  $\{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\}$

$$dend(h) = \begin{cases} \{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 0 \leq h < 0.5 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 0.5 \leq h < 1.12 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 1.12 \leq h < 1.175 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} & \text{if } 1.175 \leq h < \mathbf{2.38} \\ \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} & \text{if } \mathbf{2.38} \leq h \end{cases}$$

## Example (cont'd)

The dendrogram :



## The median method

- Proposed by Gower in 1967
- Following the LW formula, this approach is related to the updating rule below :

$$D_{med}(C_k, C_l \cup C_{l'}) = \frac{1}{2}D_{med}(C_k, C_l) + \frac{1}{2}D_{med}(C_k, C_{l'}) - \frac{1}{4}D_{med}(C_l, C_{l'})$$

## The median method

- Proposed by Gower in 1967

## The median method

- Proposed by Gower in 1967
- Following the LW formula, this approach is related to the updating rule below :

$$D_{med}(C_k, C_l \cup C_{l'}) = \frac{1}{2}D_{med}(C_k, C_l) + \frac{1}{2}D_{med}(C_k, C_{l'}) - \frac{1}{4}D_{med}(C_l, C_{l'})$$

- In the centroid method, if the size of the clusters to be merged are different, then the centroid of the new group will be close to that of the larger cluster. In the median method, the centroid of the new group is independent of the size of the groups.

## Some comments on the different approaches

- Single link : Tends to produce unbalanced clusters and clusters with a “chaining” phenomenon, especially in large data sets. Does not take account of cluster structure.

## Some comments on the different approaches

- Single link : Tends to produce unbalanced clusters and clusters with a “chaining” phenomenon, especially in large data sets. Does not take account of cluster structure.
- Complete link : Tends to find compact clusters with equal diameters (maximum distance between objects). Does not take account of cluster structure.
- UPGMA : Tends to join clusters with small variances. Intermediate between single and complete link methods. Takes account of cluster structure. Relatively robust.

## Some comments on the different approaches

- Single link : Tends to produce unbalanced clusters and clusters with a “chaining” phenomenon, especially in large data sets. Does not take account of cluster structure.
- Complete link : Tends to find compact clusters with equal diameters (maximum distance between objects). Does not take account of cluster structure.

## Some comments on the different approaches

- Single link : Tends to produce unbalanced clusters and clusters with a “chaining” phenomenon, especially in large data sets. Does not take account of cluster structure.
- Complete link : Tends to find compact clusters with equal diameters (maximum distance between objects). Does not take account of cluster structure.
- UPGMA : Tends to join clusters with small variances. Intermediate between single and complete link methods. Takes account of cluster structure. Relatively robust.
- WPGMA : As for UPGMA, but points in small clusters weighted more highly than points in large clusters (useful if cluster sizes are likely to be uneven).

## Some comments on the different approaches (cont'd)

- Ward method : Assumes points can be represented in an Euclidean space (for geometrical interpretation). Tends to find same-size, spherical clusters. Sensitive to outliers.

## Some comments on the different approaches (cont'd)

- Ward method : Assumes points can be represented in an Euclidean space (for geometrical interpretation). Tends to find same-size, spherical clusters. Sensitive to outliers.
- Centroid : Assumes points can be represented in an Euclidean space (for geometrical interpretation). The most numerous of the two groups clustered dominates the merged cluster. May not satisfy the ultrametric property.
- Median : Assumes points can be represented in Euclidean space for geometrical interpretation. New group is intermediate in position between merged groups. May not satisfy the ultrametric property.

## Some comments on the different approaches (cont'd)

- Ward method : Assumes points can be represented in an Euclidean space (for geometrical interpretation). Tends to find same-size, spherical clusters. Sensitive to outliers.
- Centroid : Assumes points can be represented in an Euclidean space (for geometrical interpretation). The most numerous of the two groups clustered dominates the merged cluster. May not satisfy the ultrametric property.

## Pros and cons of AHC

Pros :

- AHC are simple and versatile since they can be applied to any kinds of objects providing that we have a distance matrix

## Pros and cons of AHC

Pros :

- AHC are simple and versatile since they can be applied to any kinds of objects providing that we have a distance matrix
- There are many approaches and AHC can cope with clusters having non spherical shapes (using the single link method for example)

## Pros and cons of AHC (cont'd)

Cons :

- In AHC once two objects have been grouped together, we cannot ungroup them later on during the course of the algorithm

## Pros and cons of AHC

Pros :

- AHC are simple and versatile since they can be applied to any kinds of objects providing that we have a distance matrix
- There are many approaches and AHC can cope with clusters having non spherical shapes (using the single link method for example)
- The classification scheme is a dendrogram ie a set of nested partitions, which could be more informative than a flat partition. Furthermore, depending on the number of clusters we want, we can cut the tree diagram accordingly and have a flat partition

## Pros and cons of AHC (cont'd)

Cons :

- In AHC once two objects have been grouped together, we cannot ungroup them later on during the course of the algorithm
- Time complexity : since at each iteration, we need to find the lowest distance among  $\frac{n(n-1)}{2}$  pairs of data points and since there are  $n$  iterations, the time complexity is  $O(n^3)$



## Pros and cons of AHC (cont'd)

Cons :

- In AHC once two objects have been grouped together, we cannot ungroup them later on during the course of the algorithm
- Time complexity : since at each iteration, we need to find the lowest distance among  $\frac{n(n-1)}{2}$  pairs of data points and since there are  $n$  iterations, the time complexity is  $O(n^3)$
- Storage complexity : since we need to store the distance matrix, it has an  $O(n^2)$  complexity

## Pros and cons of AHC (cont'd)

Cons :

- In AHC once two objects have been grouped together, we cannot ungroup them later on during the course of the algorithm
- Time complexity : since at each iteration, we need to find the lowest distance among  $\frac{n(n-1)}{2}$  pairs of data points and since there are  $n$  iterations, the time complexity is  $O(n^3)$
- Storage complexity : since we need to store the distance matrix, it has an  $O(n^2)$  complexity
- Because of the complexities, most of AHC algorithms cannot be used on large datasets

⇒ To overcome those limits some recent HC approaches have been proposed see for eg [Han et al., 2006, Murtagh and Contreras, 2012, Xu and Wunsch, 2005]

## Pros and cons of AHC (cont'd)

Cons :

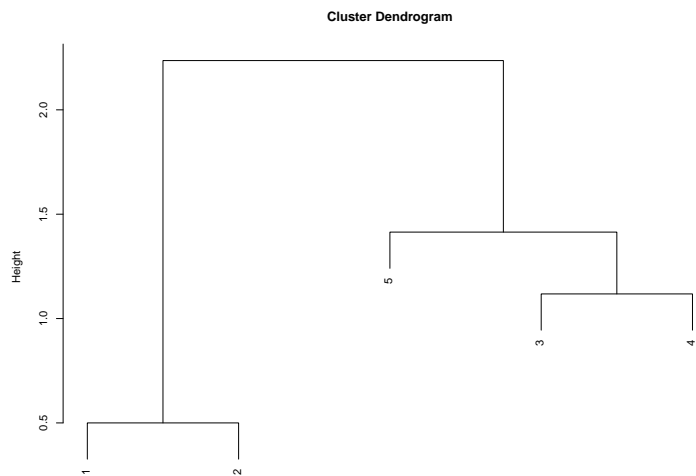
- In AHC once two objects have been grouped together, we cannot ungroup them later on during the course of the algorithm
- Time complexity : since at each iteration, we need to find the lowest distance among  $\frac{n(n-1)}{2}$  pairs of data points and since there are  $n$  iterations, the time complexity is  $O(n^3)$
- Storage complexity : since we need to store the distance matrix, it has an  $O(n^2)$  complexity
- Because of the complexities, most of AHC algorithms cannot be used on large datasets

## Example using R

```
> X=matrix(c(1,1,3,4,4,2,2.5,1,0.5,2),nrow=5,ncol=2)
> D=dist(X,method="euclidean")
> hc_single=hclust(D,method="single")
> str(as.dendrogram(hc_single))
--[dendrogram w/ 2 branches and 5 members at h = 2.24]
  |--[dendrogram w/ 2 branches and 2 members at h = 0.5]
  | |--leaf 1
  | '--leaf 2
  '--[dendrogram w/ 2 branches and 3 members at h = 1.41]
    |--leaf 5
    '--[dendrogram w/ 2 branches and 2 members at h = 1.12]
      |--leaf 3
      '--leaf 4
```

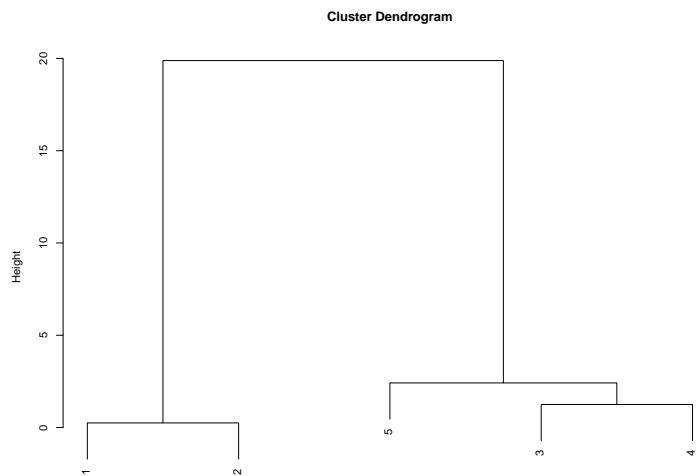
## Example using R (cont'd)

```
> plot(hc_single)
```



## Example using R (cont'd)

```
> plot(hc_ward)
```



## Example using R (cont'd)

```
> hc_ward=hclust(D^2,method="ward")
> str(as.dendrogram(hc_ward))
--[dendrogram w/ 2 branches and 5 members at h = 19.9]
 |--[dendrogram w/ 2 branches and 2 members at h = 0.25]
 | |--leaf 1
 | |--leaf 2
 |--[dendrogram w/ 2 branches and 3 members at h = 2.42]
 |--leaf 5
 |--[dendrogram w/ 2 branches and 2 members at h = 1.25]
 |--leaf 3
 |--leaf 4
```

**Exercise 9 :** Use R and compute the AHC of the previous example with the median method.

## Outline

- 1 Hierarchical clustering (HC)
  - Agglomerative hierarchical clustering (AHC)
  - Divisive hierarchical clustering (DHC)

## Pseudo-code of DHC

Pseudo-code of divisive hierarchical clusterings (DHC) :

- 1 **Input** :  $D$
- 2 Initialize the tree representation with 1 root (all points in 1 cluster)
- 3 **While** there are not  $n$  leaves **do**
- 4     Split a cluster into two according to some criterion
- 5     Add two child nodes in the tree representation accordingly
- 6 **End While**
- 7 **Output** : tree representation

## Monothetic vs Polythetic methods

DHC could be computationnaly demanding : there are  $2^{n-1} - 1$  possible subdivisions into two clusters when splitting a cluster of size  $n$ . For data consisting of  $p$  binary variables, there are relatively efficient approaches known as :

## Pseudo-code of DHC

Pseudo-code of divisive hierarchical clusterings (DHC) :

- 1 **Input** :  $D$
- 2 Initialize the tree representation with 1 root (all points in 1 cluster)
- 3 **While** there are not  $n$  leaves **do**
- 4     Split a cluster into two according to some criterion
- 5     Add two child nodes in the tree representation accordingly
- 6 **End While**
- 7 **Output** : tree representation

The critical point for DHC algorithms is the splitting criterion.

## Monothetic vs Polythetic methods

DHC could be computationnaly demanding : there are  $2^{n-1} - 1$  possible subdivisions into two clusters when splitting a cluster of size  $n$ . For data consisting of  $p$  binary variables, there are relatively efficient approaches known as :

- monothetic methods : they generally divide a cluster according to the presence or absence of one of the  $p$  variables

## Monothetic vs Polythetic methods

DHC could be computationnaly demanding : there are  $2^{n-1} - 1$  possible subdivisions into two clusters when splitting a cluster of size  $n$ . For data consisting of  $p$  binary variables, there are relatively efficient approaches known as :

- monothetic methods : they generally divide a cluster according to the presence or absence of one of the  $p$  variables

Otherwise, there are other techniques known as :

- polythetic methods : they divide the data based on the values taken by all  $p$  variables

## Diana

- Proposed by MacNaughton-Smith et al. in 1964 and further described by Kaufman and Rousseeuw in 1990 (the latter researchers developed the R function of Diana)

## Monothetic vs Polythetic methods

DHC could be computationnaly demanding : there are  $2^{n-1} - 1$  possible subdivisions into two clusters when splitting a cluster of size  $n$ . For data consisting of  $p$  binary variables, there are relatively efficient approaches known as :

- monothetic methods : they generally divide a cluster according to the presence or absence of one of the  $p$  variables

Otherwise, there are other techniques known as :

- polythetic methods : they divide the data based on the values taken by all  $p$  variables

DHC are less used than AHC. Globally pros and cons about AHC are valids for DHC. In the sequel, we present Diana a polythetic method.

## Diana

- Proposed by MacNaughton-Smith et al. in 1964 and further described by Kaufman and Rousseeuw in 1990 (the latter researchers developed the R function of Diana)
- Diana stands for Divisive ANAlysis clustering

## Diana

- Proposed by MacNaughton-Smith et al. in 1964 and further described by Kaufman and Rousseeuw in 1990 (the latter researchers developed the R function of Diana)
- Diana stands for Divisive ANALysis clustering
- At each step, the biggest cluster according to the diameter criterion is split. Let  $C_k$  be a cluster its diameter is defined as follows :

$$Diam(C_k) = \max_{\mathbf{x}, \mathbf{y} \in C_k} D(\mathbf{x}, \mathbf{y})$$

where  $D$  is the starting distance matrix between items

## Diana

- Proposed by MacNaughton-Smith et al. in 1964 and further described by Kaufman and Rousseeuw in 1990 (the latter researchers developed the R function of Diana)
- Diana stands for Divisive ANALysis clustering
- At each step, the biggest cluster according to the diameter criterion is split. Let  $C_k$  be a cluster its diameter is defined as follows :

$$Diam(C_k) = \max_{\mathbf{x}, \mathbf{y} \in C_k} D(\mathbf{x}, \mathbf{y})$$

where  $D$  is the starting distance matrix between items

- The values of the diameter are also used as heights to represent the hierarchical clustering as a tree diagram
- At each step, let  $C_I$  and  $C_{I'}$  be the clusters divided from  $C_k$  ie  $C_I \cap C_{I'} = \emptyset$  and  $C_I \cup C_{I'} = C_k$

## Diana

- Proposed by MacNaughton-Smith et al. in 1964 and further described by Kaufman and Rousseeuw in 1990 (the latter researchers developed the R function of Diana)
- Diana stands for Divisive ANALysis clustering
- At each step, the biggest cluster according to the diameter criterion is split. Let  $C_k$  be a cluster its diameter is defined as follows :

$$Diam(C_k) = \max_{\mathbf{x}, \mathbf{y} \in C_k} D(\mathbf{x}, \mathbf{y})$$

where  $D$  is the starting distance matrix between items

- The values of the diameter are also used as heights to represent the hierarchical clustering as a tree diagram

## Diana

- Proposed by MacNaughton-Smith et al. in 1964 and further described by Kaufman and Rousseeuw in 1990 (the latter researchers developed the R function of Diana)
- Diana stands for Divisive ANALysis clustering
- At each step, the biggest cluster according to the diameter criterion is split. Let  $C_k$  be a cluster its diameter is defined as follows :

$$Diam(C_k) = \max_{\mathbf{x}, \mathbf{y} \in C_k} D(\mathbf{x}, \mathbf{y})$$

where  $D$  is the starting distance matrix between items

- The values of the diameter are also used as heights to represent the hierarchical clustering as a tree diagram
- At each step, let  $C_I$  and  $C_{I'}$  be the clusters divided from  $C_k$  ie  $C_I \cap C_{I'} = \emptyset$  and  $C_I \cup C_{I'} = C_k$
- Diana finds  $C_I$  and  $C_{I'}$  by moving points from  $C_I$  to  $C_{I'}$  iteratively

## Diana (cont'd)

- At the first stage  $C_l = C_k$  and  $C_{l'} = \emptyset$  and the data point  $\mathbf{x}^*$  that maximizes the following criterion is moved from  $C_l$  to  $C_{l'}$  :

$$D_{diana}(\mathbf{x}, C_l \setminus \{\mathbf{x}\}) = \frac{1}{|C_l| - 1} \sum_{\mathbf{y} \in C_l, \mathbf{y} \neq \mathbf{x}} D(\mathbf{x}, \mathbf{y})$$

Then we update the two clusters as follows :  $C_l \leftarrow C_l \setminus \{\mathbf{x}^*\}$  and  $C_{l'} \leftarrow C_{l'} \cup \{\mathbf{x}^*\}$

## Example

Suppose we are given the distance matrix below as an input :

$$\begin{array}{c} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \\ \mathbf{x}_6 \\ \mathbf{x}_7 \end{array} \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{x}_5 & \mathbf{x}_6 & \mathbf{x}_7 \\ 0 & 10 & 7 & 30 & 29 & 38 & 42 \\ 10 & 0 & 7 & 23 & 25 & 34 & 36 \\ 7 & 7 & 0 & 21 & 22 & 31 & 36 \\ 30 & 23 & 21 & 0 & 7 & 10 & 13 \\ 29 & 25 & 22 & 7 & 0 & 11 & 17 \\ 38 & 34 & 31 & 10 & 11 & 0 & 9 \\ 42 & 36 & 36 & 13 & 17 & 9 & 0 \end{pmatrix}$$

## Diana (cont'd)

- At the first stage  $C_l = C_k$  and  $C_{l'} = \emptyset$  and the data point  $\mathbf{x}^*$  that maximizes the following criterion is moved from  $C_l$  to  $C_{l'}$  :

$$D_{diana}(\mathbf{x}, C_l \setminus \{\mathbf{x}\}) = \frac{1}{|C_l| - 1} \sum_{\mathbf{y} \in C_l, \mathbf{y} \neq \mathbf{x}} D(\mathbf{x}, \mathbf{y})$$

Then we update the two clusters as follows :  $C_l \leftarrow C_l \setminus \{\mathbf{x}^*\}$  and  $C_{l'} \leftarrow C_{l'} \cup \{\mathbf{x}^*\}$

- At the following steps, we look at other items to move from  $C_l$  to  $C_{l'}$  according to the following measure :

$$D_{diana}(\mathbf{x}, C_l \setminus \{\mathbf{x}\}) - D_{diana}(\mathbf{x}, C_{l'}) = \frac{1}{|C_l| - 1} \sum_{\mathbf{y} \in C_l, \mathbf{y} \neq \mathbf{x}} D(\mathbf{x}, \mathbf{y}) - \frac{1}{|C_{l'}|} \sum_{\mathbf{z} \in C_{l'}} D(\mathbf{x}, \mathbf{z})$$

If  $\mathbf{x}^*$  maximizes the previous criterion and if the optimum value is positive then :  $C_l \leftarrow C_l \setminus \{\mathbf{x}^*\}$  and  $C_{l'} \leftarrow C_{l'} \cup \{\mathbf{x}^*\}$ . If the optimum value is negative we stop moving points.

## Example

Suppose we are given the distance matrix below as an input :

$$\begin{array}{c} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \\ \mathbf{x}_6 \\ \mathbf{x}_7 \end{array} \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{x}_5 & \mathbf{x}_6 & \mathbf{x}_7 \\ 0 & 10 & 7 & 30 & 29 & 38 & 42 \\ 10 & 0 & 7 & 23 & 25 & 34 & 36 \\ 7 & 7 & 0 & 21 & 22 & 31 & 36 \\ 30 & 23 & 21 & 0 & 7 & 10 & 13 \\ 29 & 25 & 22 & 7 & 0 & 11 & 17 \\ 38 & 34 & 31 & 10 & 11 & 0 & 9 \\ 42 & 36 & 36 & 13 & 17 & 9 & 0 \end{pmatrix}$$

- $dend(h) = \{ \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7 \}$  if  $42 \leq h$
- Split  $C_k = \{ \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7 \}$

## Example (cont'd)

For the first split and its first iteration, we have

$$C_I = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\} :$$

- Compute  $D_{diana}(\mathbf{x}, C_I \setminus \{\mathbf{x}\})$  for all  $\mathbf{x} \in C_I$  :

$\mathbf{x} \in C_I$	$D_{diana}(\mathbf{x}, C_I \setminus \{\mathbf{x}\})$
$\mathbf{x}_1$	26
$\mathbf{x}_2$	22.5
$\mathbf{x}_3$	20.7
$\mathbf{x}_4$	17.3
$\mathbf{x}_5$	18.5
$\mathbf{x}_6$	22.17
$\mathbf{x}_7$	25.5

## Example (cont'd)

For the first split and its first iteration, we have

$$C_I = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\} :$$

- Compute  $D_{diana}(\mathbf{x}, C_I \setminus \{\mathbf{x}\})$  for all  $\mathbf{x} \in C_I$  :

$\mathbf{x} \in C_I$	$D_{diana}(\mathbf{x}, C_I \setminus \{\mathbf{x}\})$
$\mathbf{x}_1$	26
$\mathbf{x}_2$	22.5
$\mathbf{x}_3$	20.7
$\mathbf{x}_4$	17.3
$\mathbf{x}_5$	18.5
$\mathbf{x}_6$	22.17
$\mathbf{x}_7$	25.5

- We find  $\mathbf{x}^* = \mathbf{x}_1$  and thus  $C_I \leftarrow \{\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\}$  and  $C_{I'} \leftarrow \{\mathbf{x}_1\}$

## Example (cont'd)

For the first split and its second iteration, we have now

$$C_I = \{\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\} :$$

- Compute  $D_{diana}(\mathbf{x}, C_I \setminus \{\mathbf{x}\}) - D_{diana}(\mathbf{x}, C_{I'})$  for all  $\mathbf{x} \in C_I$  :

$\mathbf{x} \in C_I$	$D_{diana}(\mathbf{x}, C_I \setminus \{\mathbf{x}\}) - D_{diana}(\mathbf{x}, C_{I'})$
$\mathbf{x}_2$	15
$\mathbf{x}_3$	16.4
$\mathbf{x}_4$	-15.2
$\mathbf{x}_5$	-12.6
$\mathbf{x}_6$	-19
$\mathbf{x}_7$	-19.8

## Example (cont'd)

For the first split and its second iteration, we have now

$$C_I = \{\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\} :$$

- Compute  $D_{diana}(\mathbf{x}, C_I \setminus \{\mathbf{x}\}) - D_{diana}(\mathbf{x}, C_{I'})$  for all  $\mathbf{x} \in C_I$  :

$\mathbf{x} \in C_I$	$D_{diana}(\mathbf{x}, C_I \setminus \{\mathbf{x}\}) - D_{diana}(\mathbf{x}, C_{I'})$
$\mathbf{x}_2$	15
$\mathbf{x}_3$	16.4
$\mathbf{x}_4$	-15.2
$\mathbf{x}_5$	-12.6
$\mathbf{x}_6$	-19
$\mathbf{x}_7$	-19.8

- We find  $\mathbf{x}^* = \mathbf{x}_3$  and thus  $C_I \leftarrow \{\mathbf{x}_2, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\}$  and  $C_{I'} \leftarrow \{\mathbf{x}_1, \mathbf{x}_3\}$

## Example (cont'd)

For the first split and its third iteration, we have now

$$C_I = \{\mathbf{x}_2, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\} :$$

- Compute  $D_{diana}(\mathbf{x}, C_I \setminus \{\mathbf{x}\}) - D_{diana}(\mathbf{x}, C_{I'})$  for all  $\mathbf{x} \in C_I$  :

$\mathbf{x} \in C_I$	$D_{diana}(\mathbf{x}, C_I \setminus \{\mathbf{x}\}) - D_{diana}(\mathbf{x}, C_{I'})$
$\mathbf{x}_2$	21
$\mathbf{x}_4$	-12.3
$\mathbf{x}_5$	-10.5
$\mathbf{x}_6$	-18.5
$\mathbf{x}_7$	-20.3

## Example (cont'd)

For the first split and its fourth iteration, we have now

$$C_I = \{\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\} :$$

- Compute  $D_{diana}(\mathbf{x}, C_I \setminus \{\mathbf{x}\}) - D_{diana}(\mathbf{x}, C_{I'})$  for all  $\mathbf{x} \in C_I$  :

$\mathbf{x} \in C_I$	$D_{diana}(\mathbf{x}, C_I \setminus \{\mathbf{x}\}) - D_{diana}(\mathbf{x}, C_{I'})$
$\mathbf{x}_4$	-14.3
$\mathbf{x}_5$	-13.6
$\mathbf{x}_6$	-24.3
$\mathbf{x}_7$	-25

## Example (cont'd)

For the first split and its third iteration, we have now

$$C_I = \{\mathbf{x}_2, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\} :$$

- Compute  $D_{diana}(\mathbf{x}, C_I \setminus \{\mathbf{x}\}) - D_{diana}(\mathbf{x}, C_{I'})$  for all  $\mathbf{x} \in C_I$  :

$\mathbf{x} \in C_I$	$D_{diana}(\mathbf{x}, C_I \setminus \{\mathbf{x}\}) - D_{diana}(\mathbf{x}, C_{I'})$
$\mathbf{x}_2$	21
$\mathbf{x}_4$	-12.3
$\mathbf{x}_5$	-10.5
$\mathbf{x}_6$	-18.5
$\mathbf{x}_7$	-20.3

- We find  $\mathbf{x}^* = \mathbf{x}_2$  and thus  $C_I \leftarrow \{\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\}$  and  $C_{I'} \leftarrow \{\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_2\}$

## Example (cont'd)

For the first split and its fourth iteration, we have now

$$C_I = \{\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\} :$$

- Compute  $D_{diana}(\mathbf{x}, C_I \setminus \{\mathbf{x}\}) - D_{diana}(\mathbf{x}, C_{I'})$  for all  $\mathbf{x} \in C_I$  :

$\mathbf{x} \in C_I$	$D_{diana}(\mathbf{x}, C_I \setminus \{\mathbf{x}\}) - D_{diana}(\mathbf{x}, C_{I'})$
$\mathbf{x}_4$	-14.3
$\mathbf{x}_5$	-13.6
$\mathbf{x}_6$	-24.3
$\mathbf{x}_7$	-25

- We find  $\mathbf{x}^* = \mathbf{x}_5$  BUT the optimal value is negative so we stop moving points



## Example (cont'd)

For the first split and its fourth iteration, we have now

$$C_I = \{\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\} :$$

- Compute  $D_{diana}(\mathbf{x}, C_I \setminus \{\mathbf{x}\}) - D_{diana}(\mathbf{x}, C_I')$  for all  $\mathbf{x} \in C_I$  :

$\mathbf{x} \in C_I$	$D_{diana}(\mathbf{x}, C_I \setminus \{\mathbf{x}\}) - D_{diana}(\mathbf{x}, C_I')$
$\mathbf{x}_4$	-14.3
$\mathbf{x}_5$	-13.6
$\mathbf{x}_6$	-24.3
$\mathbf{x}_7$	-25

- We find  $\mathbf{x}^* = \mathbf{x}_5$  BUT the optimal value is negative so we stop moving points
- 5  $dend(h) = \begin{cases} \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\} & \text{if } 42 \leq h \\ \{\{\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\}, \{\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_2\}\} & \text{if } 17 \leq h < 42 \end{cases}$   
where 17 is the diameter of the next cluster to be split.

## Example with R

```
> D=matrix(c(0,10,7,30,29,38,42,0,0,7,23,25,34,36,0,0,0,21,22,...
> D=D+t(D)
> install.packages("cluster")
> library(cluster)
> hc_diana=diana(D,diss=T)
```

**Exercise 11 :** Use R and compute the DHC of the example used for AHC methods with the Diana algorithm.

## Example (cont'd)

For the first split and its fourth iteration, we have now

$$C_I = \{\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\} :$$

- Compute  $D_{diana}(\mathbf{x}, C_I \setminus \{\mathbf{x}\}) - D_{diana}(\mathbf{x}, C_I')$  for all  $\mathbf{x} \in C_I$  :

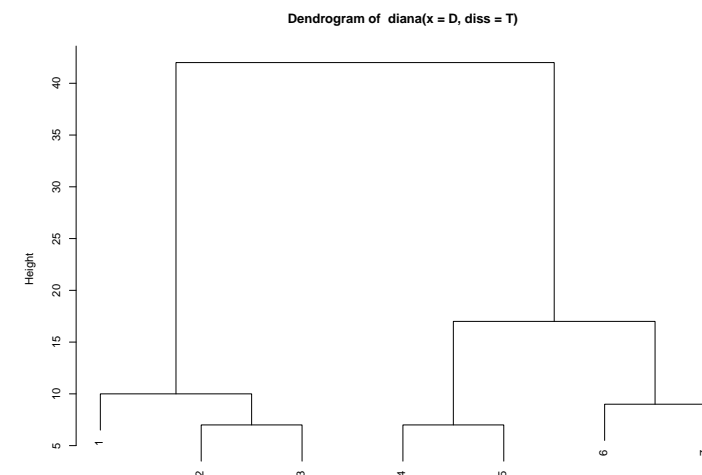
$\mathbf{x} \in C_I$	$D_{diana}(\mathbf{x}, C_I \setminus \{\mathbf{x}\}) - D_{diana}(\mathbf{x}, C_I')$
$\mathbf{x}_4$	-14.3
$\mathbf{x}_5$	-13.6
$\mathbf{x}_6$	-24.3
$\mathbf{x}_7$	-25

- We find  $\mathbf{x}^* = \mathbf{x}_5$  BUT the optimal value is negative so we stop moving points
- 5  $dend(h) = \begin{cases} \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\} & \text{if } 42 \leq h \\ \{\{\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\}, \{\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_2\}\} & \text{if } 17 \leq h < 42 \end{cases}$   
where 17 is the diameter of the next cluster to be split.

**Exercise 10 :** Finish the Diana algorithm applied to this example.


## Example using R (cont'd)

```
> pltree(hc_diana)
```




 Han, J., Kamber, M., and Pei, J. (2006).  
Data Mining : Concepts and Techniques, Second Edition (The Morgan Kaufmann Series in Data Management Systems).

Morgan Kaufmann, 2 edition.

 Lance, G. N. and Williams, W. T. (1967).  
A General Theory of Classificatory Sorting Strategies : 1. Hierarchical Systems.  
[The Computer Journal](#), 9(4) :373–380.

 Murtagh, F. and Contreras, P. (2012).  
Algorithms for hierarchical clustering : an overview.  
[Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery](#), 2(1) :86–97.

 Xu, R. and Wunsch, D. I. I. (2005).  
Survey of clustering algorithms.  
[IEEE Transactions on Neural Networks](#), 16(3) :645–678.

## Data Clustering - Part 3

### M2 DMKM

Julien Ah-Pine (julien.ah-pine@univ-lyon2.fr)

Université Lyon 2

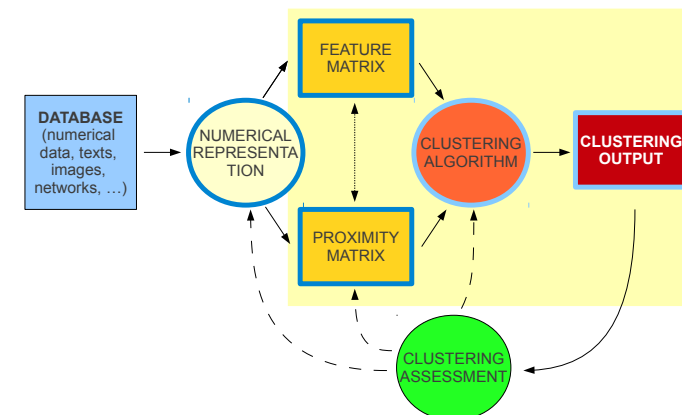
2015-2016

## Organization

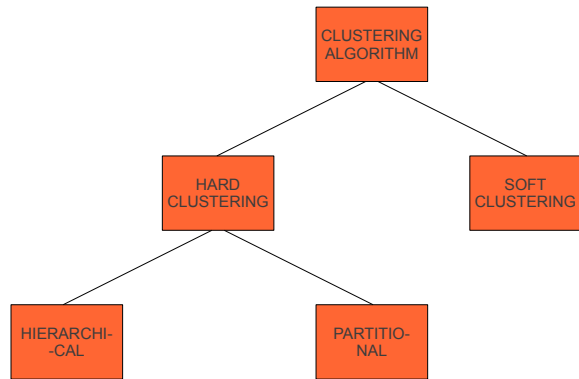
Outline of today's lesson :

- 1 Hard partitional clustering
  - $k$ -means
  - Some extensions of the  $k$ -means algorithm
- 2 Soft partitional clustering
  - Fuzzy  $k$ -means and fuzzy  $k$ -modes
  - Density mixtures and EM algorithm
- 3 Some (external) validity indices for assessing clustering outputs

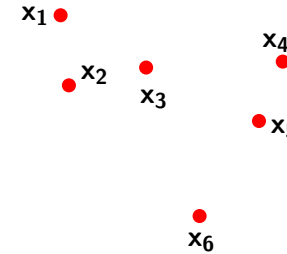
## Recalling the clustering process



## Different types of clustering algorithm

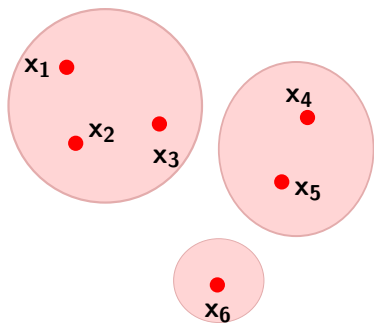


## Partitional clustering



We seek for a flat partition  $C$  such that objects belonging to a cluster are similar and objects belonging to different clusters are dissimilar.

## Partitional clustering



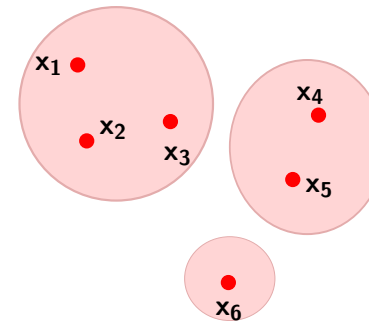
We seek for a flat partition  $C$  such that objects belonging to a cluster are similar and objects belonging to different clusters are dissimilar.

Recall that a partition is the same as clustering or as an equivalence relation.

## Hard vs Soft partitional clustering

We can make the distinction between hard (or crisp) and soft (or fuzzy) partitional clustering.

**Hard clustering** : an object belongs to only one cluster.



$$U = \begin{matrix} & C_1 & C_2 & C_3 \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{matrix} & \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{matrix}$$

$$C = \left\{ \underbrace{\{x_1, x_2, x_3\}}_{C_1}, \underbrace{\{x_4, x_5\}}_{C_2}, \underbrace{\{x_6\}}_{C_3} \right\}$$

where  $U$  is the **assignment matrix** of size  $(n \times |C|)$

## Hard vs Soft partitional clustering (cont'd)

In hard clustering, the assignment matrix  $\mathbf{U}$  is such that :

- $\forall i = 1, \dots, n; \forall l = 1, \dots, |C| : u_{il} \in \{0, 1\}$
- $\forall i = 1, \dots, n : \sum_{l=1}^{|C|} u_{il} = 1$

## Hard vs Soft partitional clustering (cont'd)

In hard clustering, the assignment matrix  $\mathbf{U}$  is such that :

- $\forall i = 1, \dots, n; \forall l = 1, \dots, |C| : u_{il} \in \{0, 1\}$
- $\forall i = 1, \dots, n : \sum_{l=1}^{|C|} u_{il} = 1$

In **soft clustering**, an object can belong to several clusters and in that case, it has a non null membership value with all clusters it belongs to :

- $\forall i = 1, \dots, n; \forall l = 1, \dots, |C| : u_{il} \in [0, 1]$
- $\forall i = 1, \dots, n : \sum_{l=1}^{|C|} u_{il} = 1$

$u_{il}$  indicates the strength of the membership of  $\mathbf{x}_i$  to  $C_l$ . Illustration :

$$\mathbf{U} = \begin{matrix} & C_1 & C_2 & C_3 \\ \mathbf{x}_1 & \left( \begin{array}{ccc} 0.9 & 0.05 & 0.05 \\ 0.7 & 0.2 & 0.1 \\ 0.6 & 0.25 & 0.15 \\ 0.2 & 0.7 & 0.1 \\ 0.25 & 0.5 & 0.25 \\ 0.25 & 0.3 & 0.45 \end{array} \right) \\ \mathbf{x}_2 & \\ \mathbf{x}_3 & \\ \mathbf{x}_4 & \\ \mathbf{x}_5 & \\ \mathbf{x}_6 & \end{matrix}$$

## Outline

- 1 Hard partitional clustering
  - $k$ -means
  - Some extensions of the  $k$ -means algorithm
- 2 Soft partitional clustering
  - Fuzzy  $k$ -means and fuzzy  $k$ -modes
  - Density mixtures and EM algorithm
- 3 Some (external) validity indices for assessing clustering outputs

## $k$ -means

- First proposed by Forgy in 1965 and MacQueen in 1967

## k-means

- First proposed by Forgy in 1965 and MacQueen in 1967
- The conventional  $k$ -means algorithm is the one described by Hartigan in 1975

## k-means

- First proposed by Forgy in 1965 and MacQueen in 1967
- The conventional  $k$ -means algorithm is the one described by Hartigan in 1975
- The  $k$  stands for the number of clusters which has to be fixed as a parameter
- The conventional  $k$ -means algorithm is applied to a continuous data table  $\mathbf{X}$  and attempts to minimize the SSE (Sum of Square) error function :

$$SSE(C) = \sum_{l=1}^{|C|} \sum_{\mathbf{x} \in C_l} \underbrace{\|\mathbf{x} - \mu(C_l)\|^2}_{D_{eucl}^2(\mathbf{x}, \mu(C_l))}$$

where  $\mu(C_l) = \frac{1}{|C_l|} \sum_{\mathbf{x} \in C_l} \mathbf{x}$  is the mean vector of cluster  $C_l$

## k-means

- First proposed by Forgy in 1965 and MacQueen in 1967
- The conventional  $k$ -means algorithm is the one described by Hartigan in 1975
- The  $k$  stands for the number of clusters which has to be fixed as a parameter

## k-means (cont'd)

- Recall that the partitioning problem is an NP-Hard problem

## k-means (cont'd)

- Recall that the partitioning problem is an NP-Hard problem
- The  $k$ -means algorithm is an **hill-climbing** optimization heuristics which finds a local minimum of  $SSE$

## k-means (cont'd)

The conventional  $k$ -means algorithm can be divided into 2 phases :

- 1 The initialization phase : the algorithm randomly assigned objects of  $\mathbf{D}$  to  $k$  clusters
- 2 The iteration phase : the algorithm computes the distances between each object and each cluster and assigns the object to the nearest cluster according to the euclidean distance

## k-means (cont'd)

- Recall that the partitioning problem is an NP-Hard problem
- The  $k$ -means algorithm is an **hill-climbing** optimization heuristics which finds a local minimum of  $SSE$

```

1  Input :  $\mathbf{X}$  and  $k$ 
2  Initialize  $C$  with  $k$  different clusters
3  While a stopping criterion is not reached do
4      For all  $\mathbf{x} \in \mathbf{D}$  do
5          For all  $C_l \in C$  do
6              Compute  $D_{eucl}^2(\mathbf{x}, \mu(C_l))$ 
7          End For
8          Find  $C_{l^*} = \operatorname{argmin}_{C_l \in C} D_{eucl}^2(\mathbf{x}, \mu(C_l))$ 
9          Move  $\mathbf{x}$  from its current cluster to  $C_{l^*}$ 
10         Update the mean vectors accordingly
11     End For
12 End While
13 Output :  $C$ 

```

## k-means (cont'd)

The conventional  $k$ -means algorithm can be divided into 2 phases :

- 1 The initialization phase : the algorithm randomly assigned objects of  $\mathbf{D}$  to  $k$  clusters
- 2 The iteration phase : the algorithm computes the distances between each object and each cluster and assigns the object to the nearest cluster according to the euclidean distance

The algorithm stops when :

- A maximal number of iterations is reached
- The  $SSE$  value does not change significantly
- The clusters do not change any longer

## Another view of $k$ -means

We can also introduce the partitioning problem related to the  $k$ -means algorithm as follows :

$$SSE(\mathbf{U}, \mathbf{Q}) = \sum_{\mathbf{q}_l \in \mathbf{Q}} \sum_{\mathbf{x}_i \in \mathbf{D}} u_{ij} D_{eucl}^2(\mathbf{x}_i, \mathbf{q}_l)$$

where  $\mathbf{Q} = \{\mathbf{q}_1, \dots, \mathbf{q}_k\}$  is the set of cluster prototypes which are points of the input space ( $\mathbb{R}^p$ ),  $\mathbf{Q}$  is the  $(k \times p)$  matrix whose rows are the cluster prototypes coordinates and  $\mathbf{U}$  is an assignment matrix such that :

- (1)  $\forall i = 1, \dots, n; \forall l = 1, \dots, k : u_{il} \in \{0, 1\}$
- (2)  $\forall i = 1, \dots, n : \sum_{l=1}^k u_{il} = 1$ 
  - $u_{il} = 1$  if  $\mathbf{x}_i$  is assigned to cluster  $C_l$  which is represented by  $\mathbf{q}_l$

## Another view of $k$ -means (cont'd)

One can solve the optimization problem approximatively by iteratively considering the two following subproblems (alternating minimization approach) :

- 1 : Fix  $\mathbf{Q} = \hat{\mathbf{Q}}$  and solve the reduced problem  $SSE(\mathbf{U}, \hat{\mathbf{Q}})$
- 2 : Fix  $\mathbf{U} = \hat{\mathbf{U}}$  and solve the reduced problem  $SSE(\hat{\mathbf{U}}, \mathbf{Q})$

## Another view of $k$ -means

We can also introduce the partitioning problem related to the  $k$ -means algorithm as follows :

$$SSE(\mathbf{U}, \mathbf{Q}) = \sum_{\mathbf{q}_l \in \mathbf{Q}} \sum_{\mathbf{x}_i \in \mathbf{D}} u_{ij} D_{eucl}^2(\mathbf{x}_i, \mathbf{q}_l)$$

where  $\mathbf{Q} = \{\mathbf{q}_1, \dots, \mathbf{q}_k\}$  is the set of cluster prototypes which are points of the input space ( $\mathbb{R}^p$ ),  $\mathbf{Q}$  is the  $(k \times p)$  matrix whose rows are the cluster prototypes coordinates and  $\mathbf{U}$  is an assignment matrix such that :

- (1)  $\forall i = 1, \dots, n; \forall l = 1, \dots, k : u_{il} \in \{0, 1\}$
- (2)  $\forall i = 1, \dots, n : \sum_{l=1}^k u_{il} = 1$ 
  - $u_{il} = 1$  if  $\mathbf{x}_i$  is assigned to cluster  $C_l$  which is represented by  $\mathbf{q}_l$

We have to minimize  $SSE(\mathbf{U}, \mathbf{Q})$  with respect to  $\mathbf{U}$  and  $\mathbf{Q}$  under the constraints (1) and (2).

## Another view of $k$ -means (cont'd)

One can solve the optimization problem approximatively by iteratively considering the two following subproblems (alternating minimization approach) :

- 1 : Fix  $\mathbf{Q} = \hat{\mathbf{Q}}$  and solve the reduced problem  $SSE(\mathbf{U}, \hat{\mathbf{Q}})$
- 2 : Fix  $\mathbf{U} = \hat{\mathbf{U}}$  and solve the reduced problem  $SSE(\hat{\mathbf{U}}, \mathbf{Q})$

We can solve these two subproblems efficiently (see theorems in next slide).

Another view of  $k$ -means (cont'd)

One can solve the optimization problem approximatively by iteratively considering the two following subproblems (alternating minimization approach) :

- 1 : Fix  $\mathbf{Q} = \hat{\mathbf{Q}}$  and solve the reduced problem  $SSE(\mathbf{U}, \hat{\mathbf{Q}})$
- 2 : Fix  $\mathbf{U} = \hat{\mathbf{U}}$  and solve the reduced problem  $SSE(\hat{\mathbf{U}}, \mathbf{Q})$

We can solve these two subproblems efficiently (see theorems in next slide).

Since the sequence of  $SSE$  is strictly decreasing the previous algorithm will converge to a local minimum after a finite number of iterations.

Another view of  $k$ -means (cont'd)

Theorem.

In the subproblem 1, if  $\hat{\mathbf{Q}}$  is fixed then  $SSE(\mathbf{U}, \hat{\mathbf{Q}})$  is minimized iff :

$$\forall i : 1, \dots, n : u_{ij} = \begin{cases} 1 & \text{if } D_{eucl}^2(\mathbf{x}_i, \hat{\mathbf{q}}_l) = \min_{\hat{\mathbf{q}}_{l'} \in \hat{\mathbf{Q}}} \{D_{eucl}^2(\mathbf{x}_i, \hat{\mathbf{q}}_{l'})\} \\ 0 & \text{otherwise} \end{cases}$$

Theorem.

In the subproblem 2, if  $\hat{\mathbf{U}}$  is fixed then  $SSE(\hat{\mathbf{U}}, \mathbf{Q})$  is minimized iff :

$$\forall l : 1, \dots, k : \hat{\mathbf{q}}_l = \frac{1}{\sum_{i=1}^n \hat{u}_{il}} \sum_{i=1}^n \hat{u}_{il} \mathbf{x}_i$$

Another view of  $k$ -means (cont'd)

Theorem.

In the subproblem 1, if  $\hat{\mathbf{Q}}$  is fixed then  $SSE(\mathbf{U}, \hat{\mathbf{Q}})$  is minimized iff :

$$\forall i : 1, \dots, n : u_{ij} = \begin{cases} 1 & \text{if } D_{eucl}^2(\mathbf{x}_i, \hat{\mathbf{q}}_l) = \min_{\hat{\mathbf{q}}_{l'} \in \hat{\mathbf{Q}}} \{D_{eucl}^2(\mathbf{x}_i, \hat{\mathbf{q}}_{l'})\} \\ 0 & \text{otherwise} \end{cases}$$

Another view of  $k$ -means (cont'd)

Theorem.

In the subproblem 1, if  $\hat{\mathbf{Q}}$  is fixed then  $SSE(\mathbf{U}, \hat{\mathbf{Q}})$  is minimized iff :

$$\forall i : 1, \dots, n : u_{ij} = \begin{cases} 1 & \text{if } D_{eucl}^2(\mathbf{x}_i, \hat{\mathbf{q}}_l) = \min_{\hat{\mathbf{q}}_{l'} \in \hat{\mathbf{Q}}} \{D_{eucl}^2(\mathbf{x}_i, \hat{\mathbf{q}}_{l'})\} \\ 0 & \text{otherwise} \end{cases}$$

Theorem.

In the subproblem 2, if  $\hat{\mathbf{U}}$  is fixed then  $SSE(\hat{\mathbf{U}}, \mathbf{Q})$  is minimized iff :

$$\forall l : 1, \dots, k : \hat{\mathbf{q}}_l = \frac{1}{\sum_{i=1}^n \hat{u}_{il}} \sum_{i=1}^n \hat{u}_{il} \mathbf{x}_i$$

Both approaches minimize  $SSE$  approximatively. The difference is that the conventional  $k$ -means algorithm updates both  $\mathbf{U}$  and  $\mathbf{Q}$  progressively and after each data point reallocation.



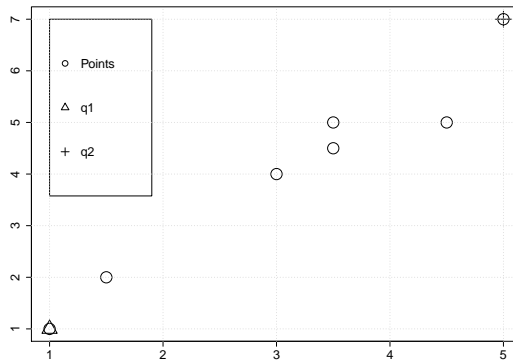
### Example

Data table :

$$X = \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{matrix} \begin{pmatrix} 1 & 1 \\ 1.5 & 2 \\ 3 & 4 \\ 5 & 7 \\ 3.5 & 5 \\ 4.5 & 5 \\ 3.5 & 4.5 \end{pmatrix}$$

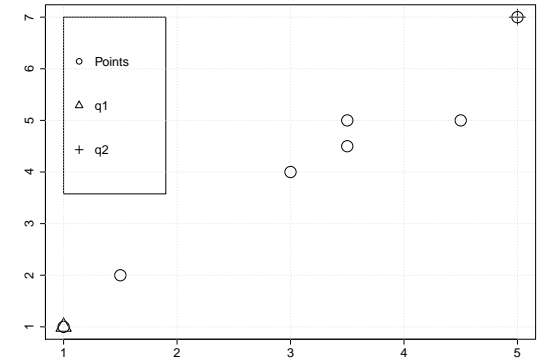
Initialization of Q :

$$q_1 = \begin{pmatrix} 1 & 1 \\ 5 & 7 \end{pmatrix}$$



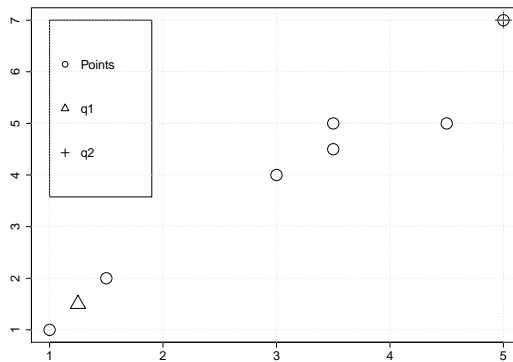
### Example (cont'd)

$$\begin{aligned} 5-7 \quad D_{eucl}^2(x_1, q_1) &= 0 \\ D_{eucl}^2(x_1, q_2) &= 52 \\ 8-10 \quad C_1 &= \{x_1\} \\ q_1 &= x_1 \\ &= (1, 1) \end{aligned}$$



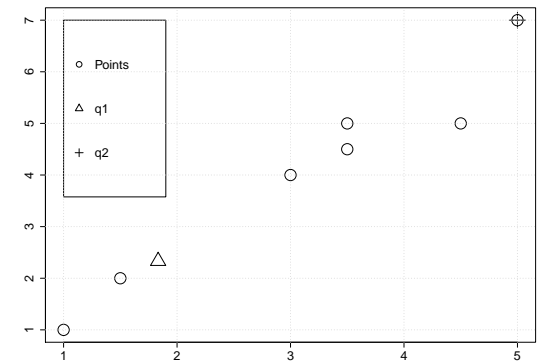
### Example (cont'd)

$$\begin{aligned} 5-7 \quad D_{eucl}^2(x_1, q_1) &= 0 \\ D_{eucl}^2(x_1, q_2) &= 52 \\ 8-10 \quad C_1 &= \{x_1\} \\ q_1 &= x_1 \\ &= (1, 1) \\ 5-7 \quad D_{eucl}^2(x_2, q_1) &= 1.25 \\ D_{eucl}^2(x_2, q_2) &= 37.25 \\ 8-10 \quad C_1 &= \{x_1, x_2\} \\ q_1 &= \frac{x_1 + x_2}{2} \\ &= (1.25, 1.5) \end{aligned}$$



### Example (cont'd)

$$\begin{aligned} 5-7 \quad D_{eucl}^2(x_1, q_1) &= 0 \\ D_{eucl}^2(x_1, q_2) &= 52 \\ 8-10 \quad C_1 &= \{x_1\} \\ q_1 &= x_1 \\ &= (1, 1) \\ 5-7 \quad D_{eucl}^2(x_2, q_1) &= 1.25 \\ D_{eucl}^2(x_2, q_2) &= 37.25 \\ 8-10 \quad C_1 &= \{x_1, x_2\} \\ q_1 &= \frac{x_1 + x_2}{2} \\ &= (1.25, 1.5) \\ 5-7 \quad D_{eucl}^2(x_3, q_1) &= 9.31 \\ D_{eucl}^2(x_3, q_2) &= 13 \\ 8-10 \quad C_1 &= \{x_1, x_2, x_3\} \\ q_1 &= \frac{x_1 + x_2 + x_3}{3} \\ &= (1.83, 2.33) \end{aligned}$$



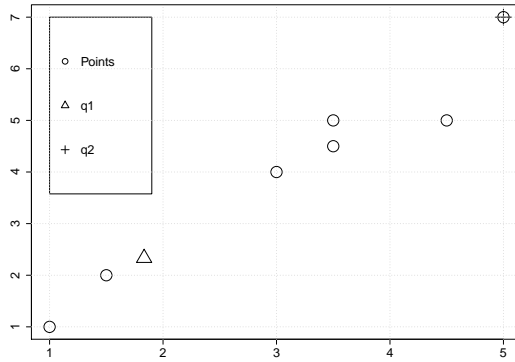
### Example (cont'd)

5-7  $D_{eucl}^2(\mathbf{x}_4, \mathbf{q}_1) = 31.80$

$D_{eucl}^2(\mathbf{x}_4, \mathbf{q}_2) = 0$

8-10  $C_2 = \{\mathbf{x}_4\}$

$\mathbf{q}_2 = \mathbf{x}_4$   
 $= (5, 7)$



### Example (cont'd)

5-7  $D_{eucl}^2(\mathbf{x}_4, \mathbf{q}_1) = 31.80$

$D_{eucl}^2(\mathbf{x}_4, \mathbf{q}_2) = 0$

8-10  $C_2 = \{\mathbf{x}_4\}$

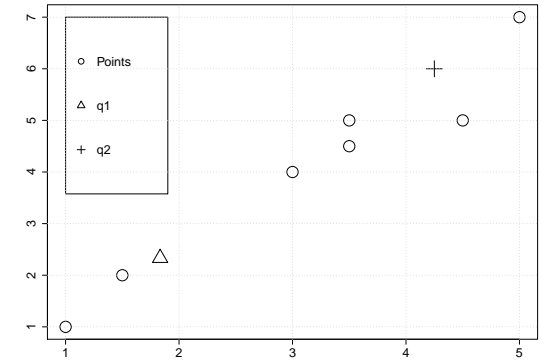
$\mathbf{q}_2 = \mathbf{x}_4$   
 $= (5, 7)$

5-7  $D_{eucl}^2(\mathbf{x}_5, \mathbf{q}_1) = 9.89$

$D_{eucl}^2(\mathbf{x}_5, \mathbf{q}_2) = 6.25$

8-10  $C_2 = \{\mathbf{x}_4, \mathbf{x}_5\}$

$\mathbf{q}_2 = \frac{\mathbf{x}_4 + \mathbf{x}_5}{2}$   
 $= (4.25, 6)$



### Example (cont'd)

5-7  $D_{eucl}^2(\mathbf{x}_4, \mathbf{q}_1) = 31.80$

$D_{eucl}^2(\mathbf{x}_4, \mathbf{q}_2) = 0$

8-10  $C_2 = \{\mathbf{x}_4\}$

$\mathbf{q}_2 = \mathbf{x}_4$   
 $= (5, 7)$

5-7  $D_{eucl}^2(\mathbf{x}_5, \mathbf{q}_1) = 9.89$

$D_{eucl}^2(\mathbf{x}_5, \mathbf{q}_2) = 6.25$

8-10  $C_2 = \{\mathbf{x}_4, \mathbf{x}_5\}$

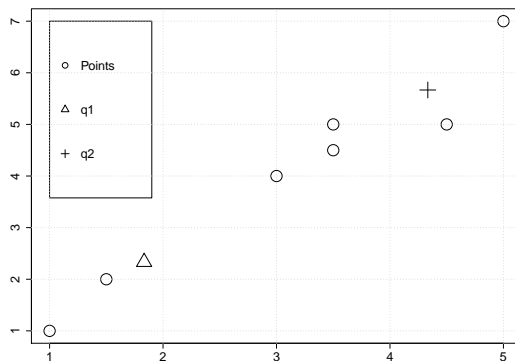
$\mathbf{q}_2 = \frac{\mathbf{x}_4 + \mathbf{x}_5}{2}$   
 $= (4.25, 6)$

5-7  $D_{eucl}^2(\mathbf{x}_6, \mathbf{q}_1) = 14.22$

$D_{eucl}^2(\mathbf{x}_6, \mathbf{q}_2) = 1.06$

8-10  $C_2 = \{\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6\}$

$\mathbf{q}_2 = \frac{\mathbf{x}_4 + \mathbf{x}_5 + \mathbf{x}_6}{3}$   
 $= (4.33, 5.67)$



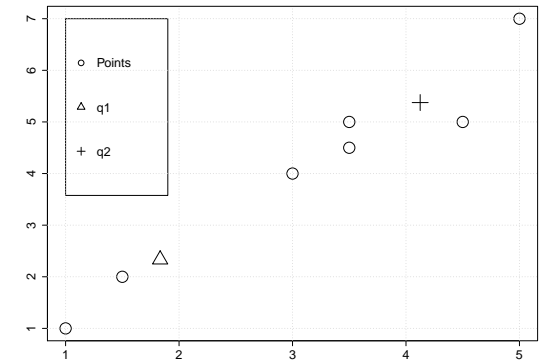
### Example (cont'd)

5-7  $D_{eucl}^2(\mathbf{x}_7, \mathbf{q}_1) = 7.47$

$D_{eucl}^2(\mathbf{x}_7, \mathbf{q}_2) = 2.05$

8-10  $C_2 = \{\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\}$

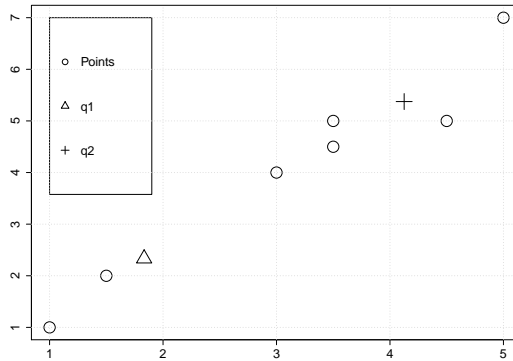
$\mathbf{q}_2 = \frac{\sum_{i=4}^7 \mathbf{x}_i}{4}$   
 $= (4.12, 5.37)$



### Example (cont'd)

$$\begin{aligned}
 5-7 \quad D_{eucl}^2(\mathbf{x}_7, \mathbf{q}_1) &= 7.47 \\
 D_{eucl}^2(\mathbf{x}_7, \mathbf{q}_2) &= 2.05 \\
 8-10 \quad C_2 &= \{\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\} \\
 \mathbf{q}_2 &= \frac{\sum_{i=4}^7 \mathbf{x}_i}{4} \\
 &= (4.12, 5.37)
 \end{aligned}$$

At the end of the first scan, we have the two clusters :  $\{\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}, \{\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\}\}$ .

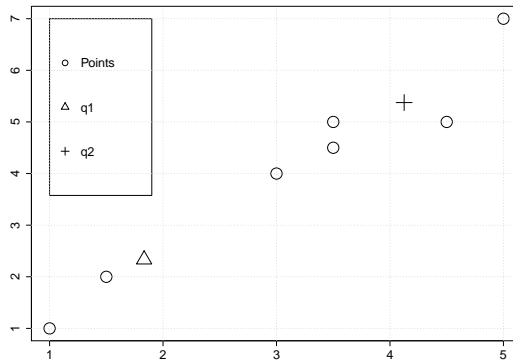


### Example (cont'd)

$$\begin{aligned}
 5-7 \quad D_{eucl}^2(\mathbf{x}_7, \mathbf{q}_1) &= 7.47 \\
 D_{eucl}^2(\mathbf{x}_7, \mathbf{q}_2) &= 2.05 \\
 8-10 \quad C_2 &= \{\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\} \\
 \mathbf{q}_2 &= \frac{\sum_{i=4}^7 \mathbf{x}_i}{4} \\
 &= (4.12, 5.37)
 \end{aligned}$$

At the end of the first scan, we have the two clusters :  $\{\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}, \{\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\}\}$ .

We start over from line 3 in the *k*-means algorithm ...



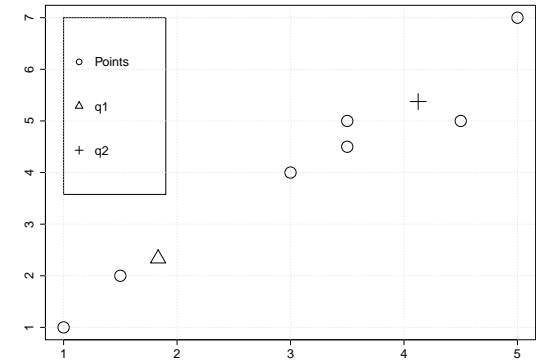
**Exercise 12 :** Continue the *k*-means algorithm on this example for one more iteration.

### Example (cont'd)

$$\begin{aligned}
 5-7 \quad D_{eucl}^2(\mathbf{x}_7, \mathbf{q}_1) &= 7.47 \\
 D_{eucl}^2(\mathbf{x}_7, \mathbf{q}_2) &= 2.05 \\
 8-10 \quad C_2 &= \{\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\} \\
 \mathbf{q}_2 &= \frac{\sum_{i=4}^7 \mathbf{x}_i}{4} \\
 &= (4.12, 5.37)
 \end{aligned}$$

At the end of the first scan, we have the two clusters :  $\{\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}, \{\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\}\}$ .

We start over from line 3 in the *k*-means algorithm ...



### Pros and cons of the *k*-means algorithm

Pros :

- One of the most used clustering algorithm since its a quite strong baseline. See for eg [Bock, 2007, Steinley, 2006] as general references for this method

## Pros and cons of the *k*-means algorithm

Pros :

- One of the most used clustering algorithm since its a quite strong baseline. See for eg [Bock, 2007, Steinley, 2006] as general references for this method
- It is efficient in clustering large data sets, since its computational complexity is linearly proportional to the size of the data sets  $O(n)$  (providing that  $k \ll n$  and  $p \ll n$ ), see for eg [Hamerly, 2010]

## Pros and cons of the *k*-means algorithm (cont'd)

Cons :

- The performance is dependent on the initialization of the centers. There have been many papers studying and proposing initialization techniques for the *k*-means algorithm, see for eg [Khan, 2004, Bradley and Fayyad, 1998]

## Pros and cons of the *k*-means algorithm

Pros :

- One of the most used clustering algorithm since its a quite strong baseline. See for eg [Bock, 2007, Steinley, 2006] as general references for this method
- It is efficient in clustering large data sets, since its computational complexity is linearly proportional to the size of the data sets  $O(n)$  (providing that  $k \ll n$  and  $p \ll n$ ), see for eg [Hamerly, 2010]
- It often terminates at a local optimum

## Pros and cons of the *k*-means algorithm (cont'd)

Cons :

- The performance is dependent on the initialization of the centers. There have been many papers studying and proposing initialization techniques for the *k*-means algorithm, see for eg [Khan, 2004, Bradley and Fayyad, 1998]
- The found clusters have convex shapes, such as a ball in three-dimensional space. Thus more complex shapes (like for high-dimensional data) are not well treated by the *k*-means approach. But new approaches have been developed to deal with this aspect like using kernels for eg [Dhillon et al., 2004]

Pros and cons of the *k*-means algorithm (cont'd)

Cons :

- The number of clusters  $k$  needs to be fixed beforehand which is a drawback when one has no clue on an adequate value<sup>1</sup>. There have also been many proposals in that context see for eg [Pelleg and Moore, 2000, Milligan and Cooper, 1985]

1. Note that we have the same issue with HC but after the clustering process.

## Example using R

```
> X=matrix(c(1,1.5,3,5,3.5,4.5,3.5,1,2,4,7,5,5,4.5),nrow=7)
> kmeans_eucl=kmeans(X,centers=2)
> print(kmeans_eucl)
K-means clustering with 2 clusters of sizes 2, 5
```

Cluster means:

```
 [,1] [,2]
 1 1.25 1.5
 2 3.90 5.1
```

Clustering vector:

```
[1] 1 1 2 2 2 2 2
```

Within cluster sum of squares by cluster:

```
[1] 0.625 7.900
```

Available components:

```
[1] "cluster" "centers" "withinss" "size"
```

Pros and cons of the *k*-means algorithm (cont'd)

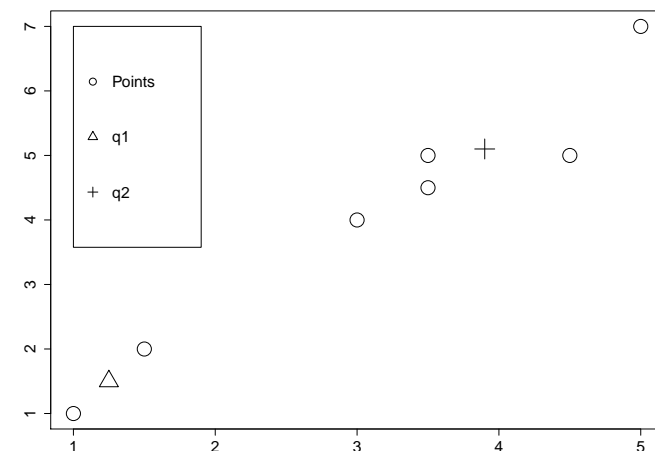
Cons :

- The number of clusters  $k$  needs to be fixed beforehand which is a drawback when one has no clue on an adequate value<sup>1</sup>. There have also been many proposals in that context see for eg [Pelleg and Moore, 2000, Milligan and Cooper, 1985]
- It deals with numerical data and this restricts the use of the *k*-means algorithm but some related techniques have been proposed (see *k*-modes in the sequel)

1. Note that we have the same issue with HC but after the clustering process.

## Example using R (cont'd)

```
> plot(rbind(X,kmeans_eucl$centers),pch=as.integer(c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)),
> legend(1, 7, c("Points","q1","q2"), pch=1:3,cex=1.5)
```



## Outline

- 1 Hard partitional clustering
  - $k$ -means
  - Some extensions of the  $k$ -means algorithm
- 2 Soft partitional clustering
  - Fuzzy  $k$ -means and fuzzy  $k$ -modes
  - Density mixtures and EM algorithm
- 3 Some (external) validity indices for assessing clustering outputs

## The generalized $k$ -means approach

Generalized approach of  $k$ -means (see [Bock, 2007]) : any kinds of error function, distance measure, and representative points or cluster prototype.

$$E(\mathbf{U}, \mathbf{Q}) = \sum_{\mathbf{q}_i \in \mathbf{Q}} \sum_{\mathbf{x}_i \in \mathbf{D}} u_{ij} D(\mathbf{x}_i, \mathbf{q}_i)$$

where  $\mathbf{U}$  is an assignment matrix,  $\mathbf{q}_i$  is a prototype vector<sup>2</sup> representing the cluster  $C_i$  and  $D$  is a dissimilarity measure.

More flexibility :

- No constraint on the type of underlying data
- Many ways to specify a family  $\mathbf{Q}$  of appropriate cluster prototypes to represent specific aspects of the clusters
- However, not all models are computationally attractive :  $SSE$  has particular properties that makes the conventional  $k$ -means algorithm fast (see previous theorems)

2. Some methods even propose several prototypes vectors to represent a single cluster.

## The generalized $k$ -means approach

Generalized approach of  $k$ -means (see [Bock, 2007]) : any kinds of error function, distance measure, and representative points or cluster prototype.

$$E(\mathbf{U}, \mathbf{Q}) = \sum_{\mathbf{q}_i \in \mathbf{Q}} \sum_{\mathbf{x}_i \in \mathbf{D}} u_{ij} D(\mathbf{x}_i, \mathbf{q}_i)$$

where  $\mathbf{U}$  is an assignment matrix,  $\mathbf{q}_i$  is a prototype vector<sup>2</sup> representing the cluster  $C_i$  and  $D$  is a dissimilarity measure.

---

2. Some methods even propose several prototypes vectors to represent a single cluster.

## The generalized $k$ -means approach (cont'd)

A general approach to approximatively solve any generalized  $k$ -means model : the **alternating minimization** algorithm.

To solve  $\min_{\mathbf{U} \in \mathbf{U}, \mathbf{Q} \in \mathbf{Q}} E(\mathbf{U}, \mathbf{Q})$  :

## The generalized $k$ -means approach (cont'd)

A general approach to approximatively solve any generalized  $k$ -means model : the **alternating minimization** algorithm.

To solve  $\min_{\mathbf{U} \in \mathcal{U}, \mathbf{Q} \in \mathcal{Q}} E(\mathbf{U}, \mathbf{Q})$  :

0 ( $t = 0$ ) Start with an arbitrary prototype system  $\mathbf{Q}^0 = (\mathbf{q}_1^0, \dots, \mathbf{q}_k^0)$

1 Fix  $\mathbf{Q} = \mathbf{Q}^t$  and minimize  $E(\mathbf{U}, \mathbf{Q}^t)$  with respect to  $\mathbf{U}$  and find  $\mathbf{U}^t$

2 Fix  $\mathbf{U} = \mathbf{U}^t$  and minimize  $E(\mathbf{U}^t, \mathbf{Q})$  with respect to  $\mathbf{Q}$  and find  $\mathbf{Q}^{t+1}$

3 Repeat 1-2 until a stopping criterion is reached

## The generalized $k$ -means approach (cont'd)

A general approach to approximatively solve any generalized  $k$ -means model : the **alternating minimization** algorithm.

To solve  $\min_{\mathbf{U} \in \mathcal{U}, \mathbf{Q} \in \mathcal{Q}} E(\mathbf{U}, \mathbf{Q})$  :

0 ( $t = 0$ ) Start with an arbitrary prototype system  $\mathbf{Q}^0 = (\mathbf{q}_1^0, \dots, \mathbf{q}_k^0)$

1 Fix  $\mathbf{Q} = \mathbf{Q}^t$  and minimize  $E(\mathbf{U}, \mathbf{Q}^t)$  with respect to  $\mathbf{U}$  and find  $\mathbf{U}^t$

2 Fix  $\mathbf{U} = \mathbf{U}^t$  and minimize  $E(\mathbf{U}^t, \mathbf{Q})$  with respect to  $\mathbf{Q}$  and find  $\mathbf{Q}^{t+1}$

3 Repeat 1-2 until a stopping criterion is reached

## The generalized $k$ -means approach (cont'd)

A general approach to approximatively solve any generalized  $k$ -means model : the **alternating minimization** algorithm.

To solve  $\min_{\mathbf{U} \in \mathcal{U}, \mathbf{Q} \in \mathcal{Q}} E(\mathbf{U}, \mathbf{Q})$  :

0 ( $t = 0$ ) Start with an arbitrary prototype system  $\mathbf{Q}^0 = (\mathbf{q}_1^0, \dots, \mathbf{q}_k^0)$

1 Fix  $\mathbf{Q} = \mathbf{Q}^t$  and minimize  $E(\mathbf{U}, \mathbf{Q}^t)$  with respect to  $\mathbf{U}$  and find  $\mathbf{U}^t$

2 Fix  $\mathbf{U} = \mathbf{U}^t$  and minimize  $E(\mathbf{U}^t, \mathbf{Q})$  with respect to  $\mathbf{Q}$  and find  $\mathbf{Q}^{t+1}$

3 Repeat 1-2 until a stopping criterion is reached

## The generalized $k$ -means approach (cont'd)

A general approach to approximatively solve any generalized  $k$ -means model : the **alternating minimization** algorithm.

To solve  $\min_{\mathbf{U} \in \mathcal{U}, \mathbf{Q} \in \mathcal{Q}} E(\mathbf{U}, \mathbf{Q})$  :

0 ( $t = 0$ ) Start with an arbitrary prototype system  $\mathbf{Q}^0 = (\mathbf{q}_1^0, \dots, \mathbf{q}_k^0)$

1 Fix  $\mathbf{Q} = \mathbf{Q}^t$  and minimize  $E(\mathbf{U}, \mathbf{Q}^t)$  with respect to  $\mathbf{U}$  and find  $\mathbf{U}^t$

2 Fix  $\mathbf{U} = \mathbf{U}^t$  and minimize  $E(\mathbf{U}^t, \mathbf{Q})$  with respect to  $\mathbf{Q}$  and find  $\mathbf{Q}^{t+1}$

3 Repeat 1-2 until a stopping criterion is reached

## $k$ -modes

- Proposed by Huang in 1997 [Huang, 1998]

## $k$ -modes

- Proposed by Huang in 1997 [Huang, 1998]
- Extension of the  $k$ -means algorithm to deal with categorical data
- $k$ -modes algorithm is applied to a discrete data table  $\mathbf{X}$  and attempts to minimize the error function :

$$E(\mathbf{U}, \mathbf{Q}) = \sum_{\mathbf{q}_l \in \mathbf{Q}} \sum_{\mathbf{x}_i \in \mathbf{D}} u_{il} D_{sm}(\mathbf{x}_i, \mathbf{q}_l)$$

where :

- ▶  $\mathbf{U}$  is the regular assignment matrix
- ▶  $D_{sm}(\mathbf{x}_i, \mathbf{q}_l)$  is the simple matching distance
- ▶  $\mathbf{q}_l$  is the prototype of cluster  $C_l$  defined as the mode vector of a set of points

## $k$ -modes

- Proposed by Huang in 1997 [Huang, 1998]
- Extension of the  $k$ -means algorithm to deal with categorical data

## $k$ -modes (cont'd)

- Let  $\mathbf{x}$  and  $\mathbf{y}$  be two data points described in a discrete input space  $\mathbf{V}$  made of  $p$  categorical variables  $\mathbf{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_p\}$



$k$ -modes (cont'd)

- Let  $\mathbf{x}$  and  $\mathbf{y}$  be two data points described in a discrete input space  $\mathbf{V}$  made of  $p$  categorical variables  $\mathbf{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_p\}$
- Each  $\mathbf{v}_j$  has  $p_j$  categories and domain  $\text{dom}(\mathbf{v}_j) = \{v_j^1, \dots, v_j^r, \dots, v_j^{p_j}\}$

 $k$ -modes (cont'd)

- Let  $\mathbf{x}$  and  $\mathbf{y}$  be two data points described in a discrete input space  $\mathbf{V}$  made of  $p$  categorical variables  $\mathbf{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_p\}$
- Each  $\mathbf{v}_j$  has  $p_j$  categories and domain  $\text{dom}(\mathbf{v}_j) = \{v_j^1, \dots, v_j^r, \dots, v_j^{p_j}\}$
- The general term of  $\mathbf{X}$  denoted  $x_{ij} \in \text{dom}(\mathbf{v}_j)$  is the category assigned to data point  $\mathbf{x}_i$  according to  $\mathbf{v}_j$
- For the categorical feature  $\mathbf{v}_j$ , let define :  $\delta(x_j, y_j) = \begin{cases} 0 & \text{if } x_j = y_j \\ 1 & \text{otherwise} \end{cases}$

 $k$ -modes (cont'd)

- Let  $\mathbf{x}$  and  $\mathbf{y}$  be two data points described in a discrete input space  $\mathbf{V}$  made of  $p$  categorical variables  $\mathbf{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_p\}$
- Each  $\mathbf{v}_j$  has  $p_j$  categories and domain  $\text{dom}(\mathbf{v}_j) = \{v_j^1, \dots, v_j^r, \dots, v_j^{p_j}\}$
- The general term of  $\mathbf{X}$  denoted  $x_{ij} \in \text{dom}(\mathbf{v}_j)$  is the category assigned to data point  $\mathbf{x}_i$  according to  $\mathbf{v}_j$

 $k$ -modes (cont'd)

- Let  $\mathbf{x}$  and  $\mathbf{y}$  be two data points described in a discrete input space  $\mathbf{V}$  made of  $p$  categorical variables  $\mathbf{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_p\}$
- Each  $\mathbf{v}_j$  has  $p_j$  categories and domain  $\text{dom}(\mathbf{v}_j) = \{v_j^1, \dots, v_j^r, \dots, v_j^{p_j}\}$
- The general term of  $\mathbf{X}$  denoted  $x_{ij} \in \text{dom}(\mathbf{v}_j)$  is the category assigned to data point  $\mathbf{x}_i$  according to  $\mathbf{v}_j$
- For the categorical feature  $\mathbf{v}_j$ , let define :  $\delta(x_j, y_j) = \begin{cases} 0 & \text{if } x_j = y_j \\ 1 & \text{otherwise} \end{cases}$
- The simple matching distance between  $\mathbf{x}$  and  $\mathbf{y}$  is defined as :

$$D_{sm}(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^p \delta(x_j, y_j)$$

## $k$ -modes (cont'd)

- Let  $\mathbf{x}$  and  $\mathbf{y}$  be two data points described in a discrete input space  $\mathbf{V}$  made of  $p$  categorical variables  $\mathbf{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_p\}$
- Each  $\mathbf{v}_j$  has  $p_j$  categories and domain  $\text{dom}(\mathbf{v}_j) = \{v_j^1, \dots, v_j^r, \dots, v_j^{p_j}\}$
- The general term of  $\mathbf{X}$  denoted  $x_{ij} \in \text{dom}(\mathbf{v}_j)$  is the category assigned to data point  $\mathbf{x}_i$  according to  $\mathbf{v}_j$
- For the categorical feature  $\mathbf{v}_j$ , let define :  $\delta(x_j, y_j) = \begin{cases} 0 & \text{if } x_j = y_j \\ 1 & \text{otherwise} \end{cases}$
- The simple matching distance between  $\mathbf{x}$  and  $\mathbf{y}$  is defined as :

$$D_{sm}(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^p \delta(x_j, y_j)$$

- Let  $C_l$  be a set of data points then the mode vector  $\mathbf{q}_l$  representing  $C_l$  is defined as :

$$\mathbf{q}_l = \underset{\mathbf{q} \in \mathbf{V}}{\text{argmin}} \sum_{\mathbf{x} \in C_l} D_{sm}(\mathbf{q}, \mathbf{x})$$

## $k$ -modes (cont'd)

Is there any efficient way to determine  $\mathbf{q}_l$  given  $C_l$  ?

- Let  $n_{jl}^r$  be the number of objects in  $C_l$  having the category  $v_j^r$  of  $\mathbf{v}_j$
- Let  $\frac{n_{jl}^r}{|C_l|}$  be the frequency of category  $v_j^r$  of  $\mathbf{v}_j$  in  $C_l$

## $k$ -modes (cont'd)

Is there any efficient way to determine  $\mathbf{q}_l$  given  $C_l$  ?

## $k$ -modes (cont'd)

Is there any efficient way to determine  $\mathbf{q}_l$  given  $C_l$  ?

- Let  $n_{jl}^r$  be the number of objects in  $C_l$  having the category  $v_j^r$  of  $\mathbf{v}_j$
- Let  $\frac{n_{jl}^r}{|C_l|}$  be the frequency of category  $v_j^r$  of  $\mathbf{v}_j$  in  $C_l$

### Theorem.

Given a set of objects  $C_l$ , the quantity  $\sum_{\mathbf{x} \in C_l} D_{sm}(\mathbf{q}, \mathbf{x})$  is minimized iff,  $\forall j = 1, \dots, p$  :

$$\forall r \neq q_j : \frac{n_{jl}^{q_j}}{|C_l|} \geq \frac{n_{jl}^r}{|C_l|}$$

where  $q_j \in \text{dom}(\mathbf{v}_j)$  is the category assigned to  $\mathbf{q}$  wrt  $\mathbf{v}_j$

## $k$ -modes (cont'd)

Is there any efficient way to determine  $\mathbf{q}_l$  given  $C_l$ ?

- Let  $n_{jl}^r$  be the number of objects in  $C_l$  having the category  $v_j^r$  of  $\mathbf{v}_j$
- Let  $\frac{n_{jl}^r}{|C_l|}$  be the frequency of category  $v_j^r$  of  $\mathbf{v}_j$  in  $C_l$

### Theorem.

Given a set of objects  $C_l$ , the quantity  $\sum_{\mathbf{x} \in C_l} D_{sm}(\mathbf{q}, \mathbf{x})$  is minimized iff,  $\forall j = 1, \dots, p$ :

$$\forall r \neq q_j : \frac{n_{jl}^{q_j}}{|C_l|} \geq \frac{n_{jl}^r}{|C_l|}$$

where  $q_j \in \text{dom}(\mathbf{v}_j)$  is the category assigned to  $\mathbf{q}$  wrt  $\mathbf{v}_j$

Given this result, the  $k$ -modes algorithm can be similar to the conventional  $k$ -means algorithm except that we use the simple matching distance and we apply the previous rule to update the cluster prototypes. Thus the time complexity of  $k$ -modes is the same as  $k$ -means.

## $k$ -medoids or $k$ -medians or Partitioning Around Medoids

- First proposed by Vinod in 1964 then developed by Kaufman and Rousseeuw in 1987 see [Kaufman and Rousseeuw, 2005]

## Example using R

```
> library(FactoMineR)
> data(poison.text)
> poison.temp=poison.text[, -3]
> install.packages("klaR")
> library("klaR")
> kmodes_res=kmodes(data=poison.temp,modes=2,weighted=F)
> print(kmodes_res)
K-modes clustering with 2 clusters of sizes 28, 27

Cluster modes:
 Sick Sex
1 sick  F
2 sick  M

Clustering vector:
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
 1  1  1  1  2  2  1  1  2  2  1  1  1  1  2  1  2  2  1  1  2  1  1  2  1  2
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
 1  2  1  1  1  2  1  2  1  2  2  1  2  2  2  2  1  1  2  2  2  2  1  1  2  2
53 54 55
 1  2  2

Within cluster simple-matching distance by cluster:
[1] 9 8

Available components:
[1] "cluster" "size" "modes" "withindiff" "iterations"
[6] "weighted"
```

## $k$ -medoids or $k$ -medians or Partitioning Around Medoids

- First proposed by Vinod in 1964 then developed by Kaufman and Rousseeuw in 1987 see [Kaufman and Rousseeuw, 2005]
- Extension of the  $k$ -means algorithm that better deals with outliers

$k$ -medoids or  $k$ -medians or Partitioning Around Medoids

- First proposed by Vinod in 1964 then developed by Kaufman and Rousseeuw in 1987 see [Kaufman and Rousseeuw, 2005]
- Extension of the  $k$ -means algorithm that better deals with outliers

$$E(\mathbf{U}, \mathbf{Q}) = \sum_{\mathbf{q}_i \in \mathbf{Q}} \sum_{\mathbf{x}_i \in \mathbf{D}} u_{ij} \underbrace{\sum_{j=1}^p |x_{ij} - q_{ij}|}_{D_{manh}(\mathbf{x}_i, \mathbf{q}_i)}$$

where :

- ▶  $\mathbf{U}$  is the regular assignment matrix
- ▶  $\sum_{j=1}^p |x_{ij} - q_{ij}|$  is the Manhattan distance between  $\mathbf{x}_i$  and  $\mathbf{q}_i$
- ▶  $\mathbf{q}_i$  is the prototype of cluster  $C_i$  and it should be an item of  $\mathbf{D}$

 $k$ -medoids or  $k$ -medians or PAM (cont'd)

Since clusters should be represented by data points in  $\mathbf{D}$ , one can solve the  $k$ -medoids problem exactly by using integer linear programming :

$$\min_{\mathbf{U}} E(\mathbf{U}) = \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathbf{D}} u_{ij} D_{manh}(\mathbf{x}_i, \mathbf{x}_j)$$

subject to :

$$\begin{cases} \forall i, j : u_{ij} \in \{0, 1\} \\ \forall i : \sum_{j=1}^n u_{ij} = 1 \\ \forall i, j : u_{ij} \leq u_{jj} \\ \sum_{j=1}^n u_{jj} = k \end{cases}$$

 $k$ -medoids or  $k$ -medians or PAM (cont'd)

Since clusters should be represented by data points in  $\mathbf{D}$ , one can solve the  $k$ -medoids problem exactly by using integer linear programming :

 $k$ -medoids or  $k$ -medians or PAM (cont'd)

Since clusters should be represented by data points in  $\mathbf{D}$ , one can solve the  $k$ -medoids problem exactly by using integer linear programming :

$$\min_{\mathbf{U}} E(\mathbf{U}) = \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathbf{D}} u_{ij} D_{manh}(\mathbf{x}_i, \mathbf{x}_j)$$

subject to :

$$\begin{cases} \forall i, j : u_{ij} \in \{0, 1\} \\ \forall i : \sum_{j=1}^n u_{ij} = 1 \\ \forall i, j : u_{ij} \leq u_{jj} \\ \sum_{j=1}^n u_{jj} = k \end{cases}$$

However, this is an NP-hard problem and in practice we use the following heuristic.

$k$ -medoids or  $k$ -medians or PAM (cont'd)

- 1 **Input** :  $\mathbf{X}$  and  $k$
- 2 Initialize  $\mathbf{Q}$  and  $C$  with  $k$  different data points in  $\mathbf{D}$

 $k$ -medoids or  $k$ -medians or PAM (cont'd)

- 1 **Input** :  $\mathbf{X}$  and  $k$
- 2 Initialize  $\mathbf{Q}$  and  $C$  with  $k$  different data points in  $\mathbf{D}$
- 3 **While** a stopping criterion is not reached **do**
- 4     **For all**  $\mathbf{x} \in \mathbf{D}$  **do**
- 5         Find  $C_{l^*} = \operatorname{argmin}_{C_l \in C} D_{\text{manh}}(\mathbf{x}, \mathbf{q}_l)$
- 6         Move  $\mathbf{x}$  to  $C_{l^*}$
- 7     **End For**

 $k$ -medoids or  $k$ -medians or PAM (cont'd)

- 1 **Input** :  $\mathbf{X}$  and  $k$
- 2 Initialize  $\mathbf{Q}$  and  $C$  with  $k$  different data points in  $\mathbf{D}$
- 3 **While** a stopping criterion is not reached **do**
- 4     **For all**  $\mathbf{x} \in \mathbf{D}$  **do**
- 5         Find  $C_{l^*} = \operatorname{argmin}_{C_l \in C} D_{\text{manh}}(\mathbf{x}, \mathbf{q}_l)$
- 6         Move  $\mathbf{x}$  to  $C_{l^*}$
- 7     **End For**
- 8     Compute  $E(\mathbf{U}, \mathbf{Q})$
- 9     Randomly select an object  $\mathbf{q}_{\text{rand}}$  in  $\mathbf{D} \setminus \mathbf{Q}$
- 10     **For all**  $\mathbf{q}_l \in \mathbf{Q}$  **do**

 $k$ -medoids or  $k$ -medians or PAM (cont'd)

- 1 **Input** :  $\mathbf{X}$  and  $k$
- 2 Initialize  $\mathbf{Q}$  and  $C$  with  $k$  different data points in  $\mathbf{D}$
- 3 **While** a stopping criterion is not reached **do**
- 4     **For all**  $\mathbf{x} \in \mathbf{D}$  **do**
- 5         Find  $C_{l^*} = \operatorname{argmin}_{C_l \in C} D_{\text{manh}}(\mathbf{x}, \mathbf{q}_l)$
- 6         Move  $\mathbf{x}$  to  $C_{l^*}$
- 7     **End For**
- 8     Compute  $E(\mathbf{U}, \mathbf{Q})$
- 9     Randomly select an object  $\mathbf{q}_{\text{rand}}$  in  $\mathbf{D} \setminus \mathbf{Q}$
- 10     **For all**  $\mathbf{q}_l \in \mathbf{Q}$  **do**
- 11          $\mathbf{Q}_{\text{rand}} \leftarrow \mathbf{Q} \setminus \mathbf{q}_l \cup \mathbf{q}_{\text{rand}}$
- 12         Compute  $S(\mathbf{q}_l, \mathbf{q}_{\text{rand}}) = E(\mathbf{U}, \mathbf{Q}) - E(\mathbf{U}, \mathbf{Q}_{\text{rand}})$
- 13     **End For**

$k$ -medoids or  $k$ -medians or PAM (cont'd)

```

1  Input :  $\mathbf{X}$  and  $k$ 
2  Initialize  $\mathbf{Q}$  and  $\mathbf{C}$  with  $k$  different data points in  $\mathbf{D}$ 
3  While a stopping criterion is not reached do
4      For all  $\mathbf{x} \in \mathbf{D}$  do
5          Find  $C_{j^*} = \operatorname{argmin}_{C_j \in \mathbf{C}} D_{\text{manh}}(\mathbf{x}, \mathbf{q}_j)$ 
6          Move  $\mathbf{x}$  to  $C_{j^*}$ 
7      End For
8      Compute  $E(\mathbf{U}, \mathbf{Q})$ 
9      Randomly select an object  $\mathbf{q}_{\text{rand}}$  in  $\mathbf{D} \setminus \mathbf{Q}$ 
10     For all  $\mathbf{q}_l \in \mathbf{Q}$  do
11          $\mathbf{Q}_{\text{rand}} \leftarrow \mathbf{Q} \setminus \mathbf{q}_l \cup \mathbf{q}_{\text{rand}}$ 
12         Compute  $S(\mathbf{q}_l, \mathbf{q}_{\text{rand}}) = E(\mathbf{U}, \mathbf{Q}) - E(\mathbf{U}, \mathbf{Q}_{\text{rand}})$ 
13     End For
14     Select  $\mathbf{q}_{l^*} = \operatorname{argmax}_{\mathbf{q}_l \in \mathbf{Q}} \{S(\mathbf{q}_l, \mathbf{q}_{\text{rand}})\}$ 
15     If  $S(\mathbf{q}_{l^*}, \mathbf{q}_{\text{rand}}) > 0$  do  $\mathbf{Q} \leftarrow \mathbf{Q} \setminus \mathbf{q}_{l^*} \cup \mathbf{q}_{\text{rand}}$  End If
16 End While
17 Output :  $\mathbf{C}$ 

```

## Example using R

```

> library(cluster)
> X=matrix(c(1,1.5,3,5,3.5,4.5,3.5,1,2,4,7,5,5,4.5),nrow=7)
> kmedoids_res=pam(X,k=2,metric='manhattan')
> print(kmedoids_res)
Medoids:
      ID
[1,]  2 1.5 2
[2,]  5 3.5 5
Clustering vector:
[1] 1 1 2 2 2 2 2
Objective function:
      build      swap
1.214286 1.142857

Available components:
[1] "medoids"      "id.med"      "clustering"  "objective"  "isolation"
[6] "clusinfo"     "silinfo"     "diss"        "call"       "data"

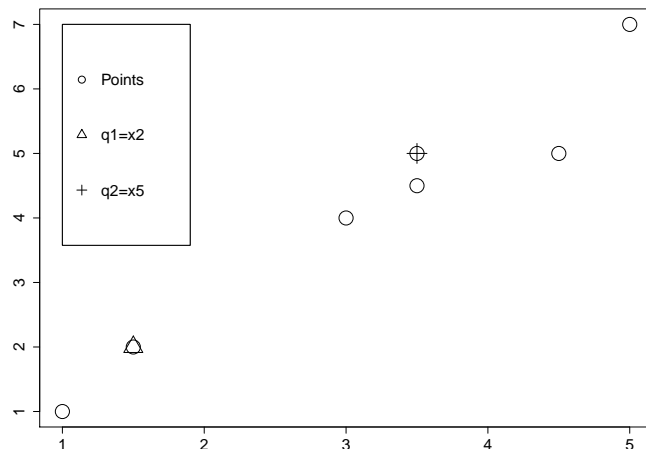
```

## Example using R (cont'd)

```

> plot(rbind(X,kmedoids_res$medoids),pch=as.integer(c(1,1,1,1,1,1,1),
> legend(1, 7, c("Points","q1=x2","q2=x5"), pch=1:3,cex=1.5)

```

Complexity and extensions of  $k$ -medoids

- $k$ -medoids is claimed to be more robust than  $k$ -means however, the time complexity is worst since it is  $O(k(n-k)^2)$  (quadratic in  $n$ )

## Complexity and extensions of $k$ -medoids

- $k$ -medoids is claimed to be more robust than  $k$ -means however, the time complexity is worst since it is  $O(k(n - k)^2)$  (quadratic in  $n$ )
- Some techniques that embed PAM have been proposed in order to address large datasets :
  - ▶ CLARA (Clustering LARge Applications) [Kaufman and Rousseeuw, 2005] : instead of taking the whole dataset into consideration, a small portion of the latter is chosen as a representative of the data. Medoids are then chosen from this sample using PAM. Multiple samples are drawn from the dataset and CLARA keeps the best set of medoids (wrt  $E$ ).

## Bisecting $k$ -means

- Described in [Steinbach et al., 2000]
- $k$ -means is iteratively applied to clusters in order to split them into two
- It is a kind of divisive hierarchical clustering technique (DHC)

## Complexity and extensions of $k$ -medoids

- $k$ -medoids is claimed to be more robust than  $k$ -means however, the time complexity is worst since it is  $O(k(n - k)^2)$  (quadratic in  $n$ )
- Some techniques that embed PAM have been proposed in order to address large datasets :
  - ▶ CLARA (Clustering LARge Applications) [Kaufman and Rousseeuw, 2005] : instead of taking the whole dataset into consideration, a small portion of the latter is chosen as a representative of the data. Medoids are then chosen from this sample using PAM. Multiple samples are drawn from the dataset and CLARA keeps the best set of medoids (wrt  $E$ ).
  - ▶ CLARANS (Clustering Large Applications based upon RANdomized) [Ng and Han, 1994] : while CLARA has a fixed sample at each stage of the search, CLARANS draws a sample with some randomness in each step of the search. The approach is like searching in a graph where nodes are set of  $k$ -medoids and two nodes are neighbors if they differ by only one medoid (a data point that is selected randomly in  $\mathbf{D}$ ). PAM is then used to go from one node to another : from one node we move to the neighbor that leads to the best decrease in terms of  $E$ .

## Bisecting $k$ -means

- Described in [Steinbach et al., 2000]
- $k$ -means is iteratively applied to clusters in order to split them into two
- It is a kind of divisive hierarchical clustering technique (DHC)

- 1 **Input** :  $\mathbf{X}$  and  $k$
- 2 Initialize  $C = \mathbf{D}$
- 3 **While**  $|C| < k$  **do**
- 4     Pick a cluster of  $C$  according to a criterion
- 4     Split the selected cluster using the conventional  $k$ -means
- 5 **End While**
- 6 **Ouput** :  $C$

## Bisecting $k$ -means

- Described in [Steinbach et al., 2000]
- $k$ -means is iteratively applied to clusters in order to split them into two
- It is a kind of divisive hierarchical clustering technique (DHC)

```

1 Input :  $\mathbf{X}$  and  $k$ 
2 Initialize  $C = \mathbf{D}$ 
3 While  $|C| < k$  do
4     Pick a cluster of  $C$  according to a criterion
4     Split the selected cluster using the conventional  $k$ -means
5 End While
6 Ouput :  $C$ 

```

**Exercise 13** : What is the time complexity of the bisecting  $k$ -means ?

## Bisecting $k$ -means

Different criteria might be used to select which cluster to split :

- Pick the largest cluster
- Pick the cluster with the largest  $SSE$  defined as :

$$SSE(C_I) = \sum_{\mathbf{x} \in C_I} \|\mathbf{x} - \mu(C_I)\|^2$$

- A criterion that mixes both the size and the  $SSE$  ...

However, it is reported in [Steinbach et al., 2000] that the results do not significantly change from one strategy to another (experiments were conducted on documents clustering).

## Bisecting $k$ -means

Different criteria might be used to select which cluster to split :

- Pick the largest cluster
- Pick the cluster with the largest  $SSE$  defined as :

$$SSE(C_I) = \sum_{\mathbf{x} \in C_I} \|\mathbf{x} - \mu(C_I)\|^2$$

- A criterion that mixes both the size and the  $SSE$  ...

## Bisecting $k$ -means

Different criteria might be used to select which cluster to split :

- Pick the largest cluster
- Pick the cluster with the largest  $SSE$  defined as :

$$SSE(C_I) = \sum_{\mathbf{x} \in C_I} \|\mathbf{x} - \mu(C_I)\|^2$$

- A criterion that mixes both the size and the  $SSE$  ...

However, it is reported in [Steinbach et al., 2000] that the results do not significantly change from one strategy to another (experiments were conducted on documents clustering).

**Exercise 14** : Using the `kmeans` function, write an R function that implements the bisecting  $k$ -means.



## Outline

- 1 Hard partitional clustering
  - $k$ -means
  - Some extensions of the  $k$ -means algorithm
- 2 Soft partitional clustering
  - Fuzzy  $k$ -means and fuzzy  $k$ -modes
  - Density mixtures and EM algorithm
- 3 Some (external) validity indices for assessing clustering outputs

## Fuzzy $k$ -means (also know as fuzzy $c$ -means)

- Proposed by Bezdek in 1973 [Bezdek, 1973]
- Extension of the  $k$ -means algorithm that allows data points to belong to several clusters

## Fuzzy $k$ -means (also know as fuzzy $c$ -means)

- Proposed by Bezdek in 1973 [Bezdek, 1973]

## Fuzzy $k$ -means (also know as fuzzy $c$ -means)

- Proposed by Bezdek in 1973 [Bezdek, 1973]
- Extension of the  $k$ -means algorithm that allows data points to belong to several clusters
- The objective function is the following one :

$$E(\mathbf{U}, \mathbf{Q}) = \sum_{\mathbf{q}_l \in \mathbf{Q}} \sum_{\mathbf{x}_i \in \mathbf{D}} u_{il}^\alpha D_{eucl}^2(\mathbf{x}_i, \mathbf{q}_l)$$

where  $\mathbf{Q}$  is again the set of cluster representative points belonging to the input space but here  $\mathbf{U}$  is a fuzzy assignment matrix with  $\alpha > 1$  such that :

- (1)  $\forall i = 1, \dots, n; \forall l = 1, \dots, k : u_{il} \in [0, 1]$
- (2)  $\forall i = 1, \dots, n : \sum_{l=1}^k u_{il} = 1$
- (3)  $\forall l = 1, \dots, k : \sum_{i=1}^n u_{il} > 0$ 
  - ▶  $u_{il}$  is the membership value of object  $\mathbf{x}_i$  to cluster  $C_l$
  - ▶  $\alpha$  is called the fuzzifier and affects the final membership distribution. Typically  $\alpha = 2$  (setting  $\alpha = 1$  leads to the crisp solution)

Fuzzy  $k$ -means (cont'd)

## Theorem.

For  $\alpha > 1$ , Bezdek gave the two following necessary conditions for a minimum  $(\mathbf{U}^*, \mathbf{Q}^*)$  of  $E(\mathbf{U}, \mathbf{Q})$ .

1 Regarding  $\mathbf{Q}^*$  :

$$\forall l = 1, \dots, k : \mathbf{q}_l^* = \frac{\sum_{\mathbf{x}_i \in C_l} (u_{il}^*)^\alpha \mathbf{x}_i}{\sum_{\mathbf{x}_i \in C_l} (u_{il}^*)^\alpha} \quad (1)$$

2 Regarding  $\mathbf{U}^*$  :

▶ If  $\forall l = 1, \dots, k : D_{eucl}^2(\mathbf{x}_i, \mathbf{q}_l^*) > 0$  then we have :

$$\forall l : u_{il}^* = \frac{(D_{eucl}^2(\mathbf{x}_i, \mathbf{q}_l^*))^{-\frac{1}{\alpha-1}}}{\sum_{\mathbf{q}_l^* \in \mathbf{Q}^*} (D_{eucl}^2(\mathbf{x}_i, \mathbf{q}_l^*))^{-\frac{1}{\alpha-1}}} \quad (2)$$

▶ If  $\exists l : D_{eucl}^2(\mathbf{x}_i, \mathbf{q}_l^*) = 0$  then  $u_{il}^*$  are any non negative numbers such that :  $\sum_{l=1}^k u_{il}^* = 1$  and  $u_{il}^* = 0$  if  $D_{eucl}^2(\mathbf{x}_i, \mathbf{q}_l^*) > 0$ .

Fuzzy  $k$ -means (cont'd)

- 1 **Input** :  $\mathbf{X}, k, \alpha$
- 2 Initialize  $k$  different clusters

Fuzzy  $k$ -means (cont'd)

- 1 **Input** :  $\mathbf{X}, k, \alpha$
- 2 Initialize  $k$  different clusters
- 3 **While** a stopping criterion is not reached **do**
- 4     **For all**  $l = 1, \dots, k$  **do**
- 5         Compute  $\mathbf{q}_l$  using Eq. (1)
- 6     **End For**

Fuzzy  $k$ -means (cont'd)

- 1 **Input** :  $\mathbf{X}, k, \alpha$
- 2 Initialize  $k$  different clusters
- 3 **While** a stopping criterion is not reached **do**
- 4     **For all**  $l = 1, \dots, k$  **do**
- 5         Compute  $\mathbf{q}_l$  using Eq. (1)
- 6     **End For**
- 7     **For all**  $\mathbf{x}_i \in \mathbf{D}$  **do**
- 8         **For all**  $l = 1, \dots, k$  **do**
- 9             Compute  $u_{il}$  using Eq. (2)
- 10         **End For**
- 11     **End For**
- 12 **End While**
- 13 **Output** :  $\mathbf{U}$  and  $\mathbf{Q}$

Fuzzy  $k$ -means (cont'd)

```

1  Input :  $\mathbf{X}$ ,  $k$ ,  $\alpha$ 
2  Initialize  $k$  different clusters
3  While a stopping criterion is not reached do
4      For all  $l = 1, \dots, k$  do
5          Compute  $\mathbf{q}_l$  using Eq. (1)
6      End For
7      For all  $\mathbf{x}_i \in \mathbf{D}$  do
8          For all  $l = 1, \dots, k$  do
9              Compute  $u_{il}$  using Eq. (2)
10         End For
11     End For
12 End While
13 Output :  $\mathbf{U}$  and  $\mathbf{Q}$ 

```

The stopping criterion is generally based on the maximum change regarding  $\mathbf{U}$  obtained after two consecutive iterations : if  $\max\{|u_{ij}^{t-1} - u_{ij}^t|\} < \epsilon$  then we stop.

## Example using R

```

> library(e1071)
> X=matrix(c(1,1.5,3,5,3.5,4.5,3.5,1,2,4,7,5,5,4.5),nrow=7)
> fuzz_kmeans_res=cmeans(x=X,centers=2,dist='euclidean',m=2)
> print(fuzz_kmeans_res)
Fuzzy c-means clustering with 2 clusters

```

```

Cluster centers:
      [,1] [,2]
1 3.931070 5.119722
2 1.304819 1.579120

```

```

Memberships:
      1      2
[1,] 0.01647822 0.98352178
[2,] 0.01357266 0.98642734
[3,] 0.80463607 0.19536393
[4,] 0.90196526 0.09803474
[5,] 0.98803001 0.01196999
[6,] 0.98480808 0.01519192
[7,] 0.95906127 0.04093873

```

```

Closest hard clustering:
[1] 2 2 1 1 1 1 1

```

```

Available components:
[1] "centers" "size" "cluster" "membership" "iter"
[6] "withinerror" "call"

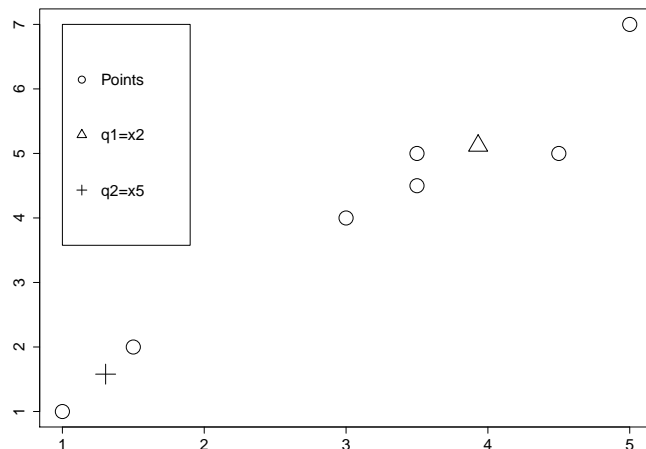
```

## Example using R (cont'd)

```

> plot(rbind(X,fuzz_kmeans_res$centers),pch=as.integer(c(1,1,1,1,1,1,1,2,3)),
> legend(1, 7, c("Points","q1=x2","q2=x5"), pch=1:3,cex=1.5)

```

Fuzzy  $k$ -modes

- Proposed by Huang and Ng in 1999 [Huang and Ng, 1999]

## Fuzzy $k$ -modes

- Proposed by Huang and Ng in 1999 [Huang and Ng, 1999]
- Extension of the fuzzy  $k$ -means algorithm to deal with categorical data

## Fuzzy $k$ -modes (cont'd)

We use an alternating minimization algorithm. The following results allow one to easily find the minimization of the consecutive subproblems.

## Fuzzy $k$ -modes

- Proposed by Huang and Ng in 1999 [Huang and Ng, 1999]
- Extension of the fuzzy  $k$ -means algorithm to deal with categorical data
- The objective function is as follows :

$$E(\mathbf{U}, \mathbf{Q}) = \sum_{\mathbf{q}_l \in \mathbf{Q}} \sum_{\mathbf{x}_i \in \mathbf{D}} u_{il}^\alpha D_{sm}(\mathbf{x}_i, \mathbf{q}_l)$$

where :

- ▶  $\mathbf{U}$  is the fuzzy assignment matrix
- ▶  $D_{sm}(\mathbf{x}_i, \mathbf{q}_l)$  is the simple matching distance
- ▶  $\mathbf{q}_l$  is the prototype of cluster  $C_l$
- ▶  $\alpha > 1$  is the fuzzifier ( $\alpha = 1$  leads to the crisp  $k$ -modes)

## Fuzzy $k$ -modes (cont'd)

We use an alternating minimization algorithm. The following results allow one to easily find the minimization of the consecutive subproblems.

### Theorem.

If  $\hat{\mathbf{U}}$  is fixed, then  $E(\hat{\mathbf{U}}, \mathbf{Q})$  is minimized iff,  $\forall l = 1, \dots, k; \forall j = 1, \dots, p :$

$$\forall r \neq q_{lj} : \sum_{\mathbf{x}_i: \mathbf{x}_{ij}=q_{lj}} u_{il}^\alpha \geq \sum_{\mathbf{x}_i: \mathbf{x}_{ij}=r} u_{il}^\alpha$$

where  $q_{lj} \in \text{dom}(\mathbf{v}_j)$  is the category assigned to  $\mathbf{q}_l$  wrt  $\mathbf{v}_j$

Fuzzy  $k$ -modes (cont'd)

## Theorem.

If  $\hat{\mathbf{Q}}$  is fixed, then  $E(\mathbf{U}, \hat{\mathbf{Q}})$  is minimized iff,  $\forall i = 1, \dots, n; \forall l = 1, \dots, k$  :

$$u_{il}^\alpha = \begin{cases} 1 & \text{if } \mathbf{x}_i = \mathbf{q}_l \\ 0 & \text{if } \mathbf{x}_i = \mathbf{q}_h \text{ with } h \neq l \\ \frac{(D_{sm}(\mathbf{x}_i, \mathbf{q}_l))^{\frac{-1}{\alpha-1}}}{\sum_{\mathbf{q}_l \in \hat{\mathbf{Q}}} (D_{sm}(\mathbf{x}_i, \mathbf{q}_l))^{\frac{-1}{\alpha-1}}} & \text{otherwise} \end{cases}$$

Fuzzy  $k$ -modes (cont'd)

- 1 **Input** :  $\mathbf{X}, k, \alpha$
- 2 Initialize  $\mathbf{Q}^0$  with  $k$  different prototypes in  $\mathbf{V}$
- 3 Determine  $\mathbf{U}^0$  minimizing  $E(\mathbf{U}, \mathbf{Q}^0)$  using the 2nd theorem

Fuzzy  $k$ -modes (cont'd)

## Theorem.

If  $\hat{\mathbf{Q}}$  is fixed, then  $E(\mathbf{U}, \hat{\mathbf{Q}})$  is minimized iff,  $\forall i = 1, \dots, n; \forall l = 1, \dots, k$  :

$$u_{il}^\alpha = \begin{cases} 1 & \text{if } \mathbf{x}_i = \mathbf{q}_l \\ 0 & \text{if } \mathbf{x}_i = \mathbf{q}_h \text{ with } h \neq l \\ \frac{(D_{sm}(\mathbf{x}_i, \mathbf{q}_l))^{\frac{-1}{\alpha-1}}}{\sum_{\mathbf{q}_l \in \hat{\mathbf{Q}}} (D_{sm}(\mathbf{x}_i, \mathbf{q}_l))^{\frac{-1}{\alpha-1}}} & \text{otherwise} \end{cases}$$

Based on these two theorems the fuzzy  $k$ -modes algorithm can be implemented as follows.

Fuzzy  $k$ -modes (cont'd)

- 1 **Input** :  $\mathbf{X}, k, \alpha$
- 2 Initialize  $\mathbf{Q}^0$  with  $k$  different prototypes in  $\mathbf{V}$
- 3 Determine  $\mathbf{U}^0$  minimizing  $E(\mathbf{U}, \mathbf{Q}^0)$  using the 2nd theorem
- 4 **While** a stopping criterion is not reached **do**
- 5     Determine  $\mathbf{Q}^1$  minimizing  $E(\mathbf{U}^0, \mathbf{Q})$  using the 1st theorem
- 6     **If**  $E(\mathbf{U}^0, \mathbf{Q}^0) = E(\mathbf{U}^0, \mathbf{Q}^1)$  **do break**

Fuzzy  $k$ -modes (cont'd)

```

1  Input :  $\mathbf{X}$ ,  $k$ ,  $\alpha$ 
2  Initialize  $\mathbf{Q}^0$  with  $k$  different prototypes in  $\mathbf{V}$ 
3  Determine  $\mathbf{U}^0$  minimizing  $E(\mathbf{U}, \mathbf{Q}^0)$  using the 2nd theorem
4  While a stopping criterion is not reached do
5      Determine  $\mathbf{Q}^1$  minimizing  $E(\mathbf{U}^0, \mathbf{Q})$  using the 1st theorem
6      If  $E(\mathbf{U}^0, \mathbf{Q}^0) = E(\mathbf{U}^0, \mathbf{Q}^1)$  do break
7      Else do Determine  $\mathbf{U}^1$  minimizing  $E(\mathbf{U}, \mathbf{Q}^1)$  using the 2nd theorem
8          If  $E(\mathbf{U}^0, \mathbf{Q}^1) = E(\mathbf{U}^1, \mathbf{Q}^1)$  do break
9          Else do  $\mathbf{U}^0 \leftarrow \mathbf{U}^1$ 
10         End If
11     End If
12 End While
13 Ouput : current  $\mathbf{U}$  and  $\mathbf{Q}$ 

```

Fuzzy  $k$ -modes in R?

**Exercise 15** : Is there any freely available R code for the fuzzy  $k$ -modes algorithm ?

Fuzzy  $k$ -modes (cont'd)

```

1  Input :  $\mathbf{X}$ ,  $k$ ,  $\alpha$ 
2  Initialize  $\mathbf{Q}^0$  with  $k$  different prototypes in  $\mathbf{V}$ 
3  Determine  $\mathbf{U}^0$  minimizing  $E(\mathbf{U}, \mathbf{Q}^0)$  using the 2nd theorem
4  While a stopping criterion is not reached do
5      Determine  $\mathbf{Q}^1$  minimizing  $E(\mathbf{U}^0, \mathbf{Q})$  using the 1st theorem
6      If  $E(\mathbf{U}^0, \mathbf{Q}^0) = E(\mathbf{U}^0, \mathbf{Q}^1)$  do break
7      Else do Determine  $\mathbf{U}^1$  minimizing  $E(\mathbf{U}, \mathbf{Q}^1)$  using the 2nd theorem
8          If  $E(\mathbf{U}^0, \mathbf{Q}^1) = E(\mathbf{U}^1, \mathbf{Q}^1)$  do break
9          Else do  $\mathbf{U}^0 \leftarrow \mathbf{U}^1$ 
10         End If
11     End If
12 End While
13 Ouput : current  $\mathbf{U}$  and  $\mathbf{Q}$ 

```

The stopping criterion is usually a maximum number of iterations.

## Outline

- 1 Hard partitional clustering
  - $k$ -means
  - Some extensions of the  $k$ -means algorithm
- 2 Soft partitional clustering
  - Fuzzy  $k$ -means and fuzzy  $k$ -modes
  - Density mixtures and EM algorithm
- 3 Some (external) validity indices for assessing clustering outputs

## Model based clustering or density mixture models

- Clustering algorithms based upon probability models
- Data are viewed as coming from a finite mixture of probability distributions
- Each distribution represents a cluster

## Model based clustering or density mixture models

- Clustering algorithms based upon probability models
- Data are viewed as coming from a finite mixture of probability distributions
- Each distribution represents a cluster
- The clustering problem becomes that of estimating the parameters of the assumed mixture
- Once the parameters of the model are estimated, we can compute the posterior probabilities of cluster membership of the objects
- A general reference on finite mixture models :  
[Mclachlan and Peel, 2000]

## Model based clustering or density mixture models

- Clustering algorithms based upon probability models
- Data are viewed as coming from a finite mixture of probability distributions
- Each distribution represents a cluster
- The clustering problem becomes that of estimating the parameters of the assumed mixture
- Once the parameters of the model are estimated, we can compute the posterior probabilities of cluster membership of the objects

## Model based clustering or density mixture models (cont'd)

- Finite mixture models are a family of probability density functions of the form :

$$f(\mathbf{x}|\mathbf{p}, \theta) = \sum_{l=1}^k p_l g_l(\mathbf{x}|\theta_l)$$

where :

- ▶  $\mathbf{x}$  is a  $p$ -dimensional random variable
- ▶  $\mathbf{p} = (p_1, \dots, p_k)$  is the vector of mixing proportions such that  $\sum_{l=1}^k p_l = 1$
- ▶  $g_l; l = 1, \dots, k$  are the different component densities of the mixture
- ▶ Each  $g_l$  is parametrized by a vector of parameters  $\theta_l$

## Model based clustering or density mixture models (cont'd)

- Finite mixture models are a family of probability density functions of the form :

$$f(\mathbf{x}|\mathbf{p}, \theta) = \sum_{l=1}^k p_l g_l(\mathbf{x}|\theta_l)$$

where :

- ▶  $\mathbf{x}$  is a  $p$ -dimensional random variable
- ▶  $\mathbf{p} = (p_1, \dots, p_k)$  is the vector of mixing proportions such that  $\sum_{l=1}^k p_l = 1$
- ▶  $g_l; l = 1, \dots, k$  are the different component densities of the mixture
- ▶ Each  $g_l$  is parametrized by a vector of parameters  $\theta_l$

Note that all  $g_l$  can be of the same density family but they differ from their parameters.

## Model based clustering or density mixture models (cont'd)

- If we have the estimation of all the parameters of the models (ie  $\hat{\mathbf{p}}$  and the  $\hat{\theta}_l$ ) then we can deduce :

$$\Pr(C_l|\mathbf{x}_i) = \frac{\hat{p}_l g_l(\mathbf{x}_i|\hat{\theta}_l)}{f(\mathbf{x}_i|\hat{\mathbf{p}}, \hat{\theta})}$$

- $\Pr(C_l|\mathbf{x}_i)$  is the posterior probability of having  $C_l$  given  $\mathbf{x}_i$ . In other words, it represents the “membership” value of  $\mathbf{x}_i$  to cluster  $C_l$ .

Suppose now that we are given  $\mathbf{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ . From the mixture density given previously, we have the following log-likelihood function  $l$  :

$$l(\mathbf{p}, \theta) = \sum_{\mathbf{x}_i \in \mathbf{D}} \ln f(\mathbf{x}_i|\mathbf{p}, \theta)$$

## Model based clustering or density mixture models (cont'd)

- If we have the estimation of all the parameters of the models (ie  $\hat{\mathbf{p}}$  and the  $\hat{\theta}_l$ ) then we can deduce :

$$\Pr(C_l|\mathbf{x}_i) = \frac{\hat{p}_l g_l(\mathbf{x}_i|\hat{\theta}_l)}{f(\mathbf{x}_i|\hat{\mathbf{p}}, \hat{\theta})}$$

- $\Pr(C_l|\mathbf{x}_i)$  is the posterior probability of having  $C_l$  given  $\mathbf{x}_i$ . In other words, it represents the “membership” value of  $\mathbf{x}_i$  to cluster  $C_l$ .

## Model based clustering or density mixture models (cont'd)

- If we have the estimation of all the parameters of the models (ie  $\hat{\mathbf{p}}$  and the  $\hat{\theta}_l$ ) then we can deduce :

$$\Pr(C_l|\mathbf{x}_i) = \frac{\hat{p}_l g_l(\mathbf{x}_i|\hat{\theta}_l)}{f(\mathbf{x}_i|\hat{\mathbf{p}}, \hat{\theta})}$$

- $\Pr(C_l|\mathbf{x}_i)$  is the posterior probability of having  $C_l$  given  $\mathbf{x}_i$ . In other words, it represents the “membership” value of  $\mathbf{x}_i$  to cluster  $C_l$ .

Suppose now that we are given  $\mathbf{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ . From the mixture density given previously, we have the following log-likelihood function  $l$  :

$$l(\mathbf{p}, \theta) = \sum_{\mathbf{x}_i \in \mathbf{D}} \ln f(\mathbf{x}_i|\mathbf{p}, \theta)$$

Estimates of the parameters would usually be obtained as a solution of the likelihood equations :  $\frac{\partial l(\phi)}{\partial \phi} = 0$  with  $\phi = (\mathbf{p}, \theta)$ . But the likelihood function is too complicated to employ the usual methods for its maximization.



## Model based clustering or density mixture models (cont'd)

- If we have the estimation of all the parameters of the models (ie  $\hat{\mathbf{p}}$  and the  $\hat{\theta}_l$ ) then we can deduce :

$$\Pr(C_l|\mathbf{x}_i) = \frac{\hat{p}_l g_l(\mathbf{x}_i|\hat{\theta}_l)}{f(\mathbf{x}_i|\hat{\mathbf{p}}, \hat{\theta})}$$

- $\Pr(C_l|\mathbf{x}_i)$  is the posterior probability of having  $C_l$  given  $\mathbf{x}_i$ . In other words, it represents the “membership” value of  $\mathbf{x}_i$  to cluster  $C_l$ .

Suppose now that we are given  $\mathbf{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ . From the mixture density given previously, we have the following log-likelihood function  $l$  :

$$l(\mathbf{p}, \theta) = \sum_{\mathbf{x}_i \in \mathbf{D}} \ln f(\mathbf{x}_i|\mathbf{p}, \theta)$$

Estimates of the parameters would usually be obtained as a solution of the likelihood equations :  $\frac{\partial l(\phi)}{\partial \phi} = 0$  with  $\phi = (\mathbf{p}, \theta)$ . But the likelihood function is too complicated to employ the usual methods for its maximization.

To estimate the parameters, the most widely used approach is the iterative **expectation maximization (EM)** algorithm [Dempster et al., 1977].

## Gaussians mixtures and EM algorithm

- Gaussians mixture is one of the most used mixture model :

$$f(\mathbf{x}|\mathbf{p}, \mu, \Sigma) = \sum_{l=1}^k p_l \underbrace{\Phi(\mathbf{x}|\mu_l, \Sigma_l)}_{g_l(\mathbf{x}|\theta_l)}$$

in that case :

- ▶  $\forall l : g_l(\mathbf{x}|\theta_l) = \Phi(\mathbf{x}|\underbrace{\mu_l, \Sigma_l}_{\theta_l}) = \frac{\exp[-\frac{1}{2}(\mathbf{x}-\mu_l)^t \Sigma_l^{-1}(\mathbf{x}-\mu_l)]}{\sqrt{(2\pi)^p |\Sigma_l|}}$
- ▶  $\forall l : \mu_l$  is the mean vector related to  $C_l$
- ▶  $\forall l : \Sigma_l$  is the covariance matrix related to  $C_l$

## Gaussians mixtures and EM algorithm (cont'd)

In the case of Gaussians mixtures, the EM algorithm iteratively updates the following quantities :

- Expectation step :

$$\Pr(C_l|\mathbf{x}_i) = \frac{\hat{p}_l \Phi(\mathbf{x}_i|\hat{\mu}_l, \hat{\Sigma}_l)}{\sum_{l=1}^k p_l \Phi(\mathbf{x}_i|\hat{\mu}_l, \hat{\Sigma}_l)} \quad (3)$$

- Maximization step :

$$\hat{p}_l = \frac{1}{n} \sum_{\mathbf{x}_i \in \mathbf{D}} \Pr(C_l|\mathbf{x}_i) \quad (4)$$

$$\hat{\mu}_l = \frac{1}{n \hat{p}_l} \sum_{\mathbf{x}_i \in \mathbf{D}} \mathbf{x}_i \Pr(C_l|\mathbf{x}_i) \quad (5)$$

$$\hat{\Sigma}_l = \frac{1}{n} \sum_{\mathbf{x}_i \in \mathbf{D}} (\mathbf{x}_i - \mu_l)(\mathbf{x}_i - \mu_l)^t \Pr(C_l|\mathbf{x}_i) \quad (6)$$

## Gaussians mixtures and EM algorithm (cont'd)

- 1 **Input** :  $\mathbf{X}, k$
- 2 Initialize the means  $\mu_l$ , covariances  $\Sigma_l$  and mixing coefficients  $p_l$

## Gaussians mixtures and EM algorithm (cont'd)

```

1  Input :  $\mathbf{X}$ ,  $k$ 
2  Initialize the means  $\mu_l$ , covariances  $\Sigma_l$  and mixing coefficients  $p_l$ 
3  While a stopping criterion is not reached do
4      For all  $i = 1, \dots, n$  do (E step)
8          For all  $l = 1, \dots, k$  do
5              Compute  $\Pr(C_l | \mathbf{x}_i)$  using Eq. (3)
6          End For
6      End For

```

## Gaussians mixtures and EM algorithm (cont'd)

```

1  Input :  $\mathbf{X}$ ,  $k$ 
2  Initialize the means  $\mu_l$ , covariances  $\Sigma_l$  and mixing coefficients  $p_l$ 
3  While a stopping criterion is not reached do
4      For all  $i = 1, \dots, n$  do (E step)
8          For all  $l = 1, \dots, k$  do
5              Compute  $\Pr(C_l | \mathbf{x}_i)$  using Eq. (3)
6          End For
6      End For
4      For all  $l = 1, \dots, k$  do (M step)
8          Compute  $\hat{p}_l$  using Eq. (4)
8          Compute  $\hat{\mu}_l$  using Eq. (5)
8          Compute  $\Sigma_l$  using Eq. (6)
11     End For
12 End While
13 Ouput :  $\mathbf{U}$  and  $\mathbf{Q}$ 

```

## Gaussians mixtures and EM algorithm (cont'd)

```

1  Input :  $\mathbf{X}$ ,  $k$ 
2  Initialize the means  $\mu_l$ , covariances  $\Sigma_l$  and mixing coefficients  $p_l$ 
3  While a stopping criterion is not reached do
4      For all  $i = 1, \dots, n$  do (E step)
8          For all  $l = 1, \dots, k$  do
5              Compute  $\Pr(C_l | \mathbf{x}_i)$  using Eq. (3)
6          End For
6      End For
4      For all  $l = 1, \dots, k$  do (M step)
8          Compute  $\hat{p}_l$  using Eq. (4)
8          Compute  $\hat{\mu}_l$  using Eq. (5)
8          Compute  $\Sigma_l$  using Eq. (6)
11     End For
12 End While
13 Ouput :  $\mathbf{U}$  and  $\mathbf{Q}$ 

```

The stopping criterion is either the convergence of the parameters or the log likelihood.

Gaussians mixtures - EM algorithm and  $k$ -means

- EM algorithm for Gaussians mixtures and  $k$ -means are related
- The former can be viewed as a soft version of the latter
- $k$ -means is a particular limit of EM for Gaussians mixtures

Gaussians mixtures - EM algorithm and  $k$ -means

- EM algorithm for Gaussians mixtures and  $k$ -means are related
- The former can be viewed as a soft version of the latter
- $k$ -means is a particular limit of EM for Gaussians mixtures
- If we assume that  $\Sigma_l = \epsilon \mathbf{I}$  for all  $l = 1, \dots, k$ , we obtain :

$$\Phi(\mathbf{x}|\mu_l, \Sigma_l) = \frac{\exp[-\frac{1}{2\epsilon}\|\mathbf{x} - \mu_l\|^2]}{\sqrt{(2\pi)^p \epsilon}}$$

$$\Pr(C_l|\mathbf{x}_i) = \frac{\hat{p}_l \exp[-\frac{1}{2\epsilon}\|\mathbf{x} - \mu_l\|^2]}{\sum_{l=1}^k \hat{p}_l \exp[-\frac{1}{2\epsilon}\|\mathbf{x} - \mu_l\|^2]}$$

## Example using R

```
> library(mclust)
> X=matrix(c(1,1.5,3,5,3.5,4.5,3.5,1,2,4,7,5,5,4.5),nrow=7)
> gauss_em_res=Mclust(data=X,G=2)
> gauss_em_res$z
      [,1]      [,2]
[1,] 9.969483e-01 0.003051682
[2,] 9.848557e-01 0.015144322
[3,] 3.216620e-24 1.000000000
[4,] 3.218174e-107 1.000000000
[5,] 2.747011e-35 1.000000000
[6,] 1.793497e-134 1.000000000
[7,] 1.989165e-45 1.000000000
> gauss_em_res$parameters$mean
      [,1]      [,2]
[1,] 1.248495 3.891120
[2,] 1.496989 5.088318
> print(gauss_em_res)
```

best model: ellipsoidal, equal shape with 2 components

Comment : the Mclust function first applies an AHC in order to initialize the parameters. Then it tests several models that differ regarding the assumptions about  $\Sigma$  (eg all  $\Sigma_l$  are the same, ...). Those different assumptions reflect the shape, volume and orientation of the multivariate Gaussians. It then uses the BIC (Bayesian Information Criterion) to select the best model (see [Fraley and Raftery, 2009] for details).

Gaussians mixtures - EM algorithm and  $k$ -means

- EM algorithm for Gaussians mixtures and  $k$ -means are related
- The former can be viewed as a soft version of the latter
- $k$ -means is a particular limit of EM for Gaussians mixtures
- If we assume that  $\Sigma_l = \epsilon \mathbf{I}$  for all  $l = 1, \dots, k$ , we obtain :

$$\Phi(\mathbf{x}|\mu_l, \Sigma_l) = \frac{\exp[-\frac{1}{2\epsilon}\|\mathbf{x} - \mu_l\|^2]}{\sqrt{(2\pi)^p \epsilon}}$$

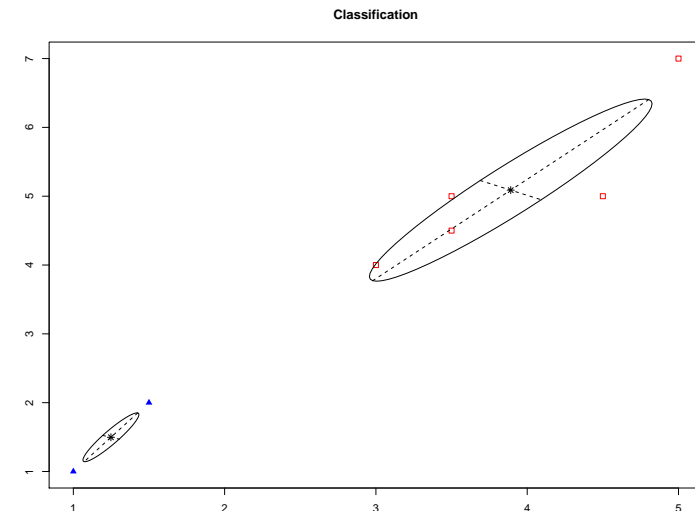
$$\Pr(C_l|\mathbf{x}_i) = \frac{\hat{p}_l \exp[-\frac{1}{2\epsilon}\|\mathbf{x} - \mu_l\|^2]}{\sum_{l=1}^k \hat{p}_l \exp[-\frac{1}{2\epsilon}\|\mathbf{x} - \mu_l\|^2]}$$

- If  $\epsilon \rightarrow 0$  then :
  - ▶  $\Pr(C_l|\mathbf{x}_i)$  converges to 0 except for  $C_{l^*} = \operatorname{argmin}_l \|\mathbf{x} - \mu_l\|^2$  (posterior probabilities tend to an hard assignment matrix  $\mathbf{U}$ )
  - ▶ The expected log likelihood tends to (see [Bishop, 2006] for details) :

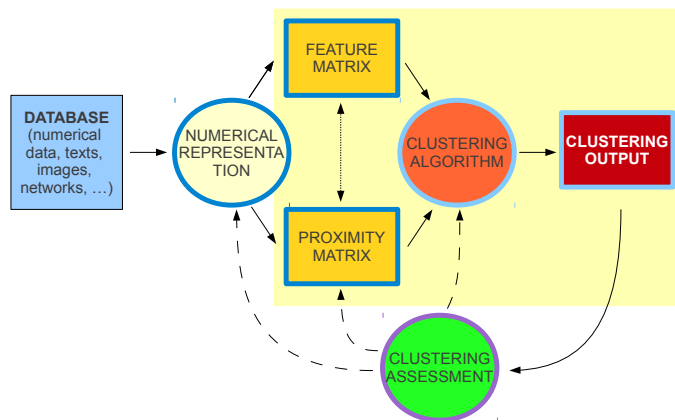
$$-\frac{1}{2} \sum_{i=1}^n \sum_{l=1}^k \mathbf{U}_{il} \|\mathbf{x}_i - \mu_l\|^2 + \text{const}$$

## Example using R (cont'd)

```
> plot(gauss_em_res,X,what="classification")
Tapez <Entrée> pour voir le graphique suivant :
```



## Recalling the clustering process



## Different types of assessment measures

How to assess clustering outputs? We can have different approaches :

- **External criteria** : we evaluate the results of a clustering algorithm based on a pre-specified structure (a ground-truth typically). The closer the clustering output to the pre-specified structure according to an assessment measure, the better the output.
- **Internal criteria** : in this case, the clustering results are evaluated in terms of quantities that involve the vectors of the dataset themselves (the proximity matrix for eg)

## Different types of assessment measures

How to assess clustering outputs? We can have different approaches :

- **External criteria** : we evaluate the results of a clustering algorithm based on a pre-specified structure (a ground-truth typically). The closer the clustering output to the pre-specified structure according to an assessment measure, the better the output.

## Different types of assessment measures

How to assess clustering outputs? We can have different approaches :

- **External criteria** : we evaluate the results of a clustering algorithm based on a pre-specified structure (a ground-truth typically). The closer the clustering output to the pre-specified structure according to an assessment measure, the better the output.
- **Internal criteria** : in this case, the clustering results are evaluated in terms of quantities that involve the vectors of the dataset themselves (the proximity matrix for eg)

We are going to present some classical external criteria. Note that in that case we assume hard flat clustering. However from either HC or soft clustering, we can extract an hard partition with  $k$  clusters. For more types of validity indices see for eg [Gan et al., 2007].

## Some external measures

Before introducing the assessment measures we need to introduce some notations :

- Let  $C = \{C_1, \dots, C_l, \dots, C_k\}$  be the clustering output
- Let  $C' = \{C'_1, \dots, C'_{l'}, \dots, C'_{k'}\}$  be the pre-specified partition

## Some external measures

Before introducing the assessment measures we need to introduce some notations :

- Let  $C = \{C_1, \dots, C_l, \dots, C_k\}$  be the clustering output
- Let  $C' = \{C'_1, \dots, C'_{l'}, \dots, C'_{k'}\}$  be the pre-specified partition
- Let  $A = \text{Nb of pairs where both objects belong to the same cluster both for } C \text{ and } C'$
- Let  $B = \text{Nb of pairs where both objects belong to the same cluster for } C \text{ but not for } C'$
- Let  $C = \text{Nb of pairs where both objects belong to the same cluster for } C' \text{ but not for } C$
- Let  $D = \text{Nb of pairs where both objects do not belong to the same cluster neither for } C' \text{ nor } C$
- Let  $M = A + B + C + D$

Note that  $M = \binom{n}{2} = \frac{n(n-1)}{2}$ .

## Some external measures

Before introducing the assessment measures we need to introduce some notations :

- Let  $C = \{C_1, \dots, C_l, \dots, C_k\}$  be the clustering output
- Let  $C' = \{C'_1, \dots, C'_{l'}, \dots, C'_{k'}\}$  be the pre-specified partition
- Let  $A = \text{Nb of pairs where both objects belong to the same cluster both for } C \text{ and } C'$
- Let  $B = \text{Nb of pairs where both objects belong to the same cluster for } C \text{ but not for } C'$
- Let  $C = \text{Nb of pairs where both objects belong to the same cluster for } C' \text{ but not for } C$
- Let  $D = \text{Nb of pairs where both objects do not belong to the same cluster neither for } C' \text{ nor } C$
- Let  $M = A + B + C + D$

## Example

- Let  $C = \{\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}, \{\mathbf{x}_4, \mathbf{x}_5\}\}$
- Let  $C' = \{\{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_5\}\}$

## Example

- Let  $C = \{\{x_1, x_2, x_3\}, \{x_4, x_5\}\}$
- Let  $C' = \{\{x_1, x_2\}, \{x_3, x_4\}, \{x_5\}\}$
- These equivalence relations can be represented by their adjacency matrices :

$$C = \begin{matrix} & x_1 & x_2 & x_3 & x_4 & x_5 \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix} & \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{matrix}; \quad C' = \begin{matrix} & x_1 & x_2 & x_3 & x_4 & x_5 \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix} & \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{matrix}$$

## Example

- Let  $C = \{\{x_1, x_2, x_3\}, \{x_4, x_5\}\}$
- Let  $C' = \{\{x_1, x_2\}, \{x_3, x_4\}, \{x_5\}\}$
- These equivalence relations can be represented by their adjacency matrices :

$$C = \begin{matrix} & x_1 & x_2 & x_3 & x_4 & x_5 \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix} & \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{matrix}; \quad C' = \begin{matrix} & x_1 & x_2 & x_3 & x_4 & x_5 \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix} & \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{matrix}$$

In this example we have  $A = 1; B = 3; C = 1$  and  $D = 5$ .

## Comparing partitions

Many external evaluation measures in clustering are statistics used to compare two partitions.

- Rand :  $\text{Rand}(C, C') = \frac{A+D}{M}$

## Comparing partitions

Many external evaluation measures in clustering are statistics used to compare two partitions.

- Rand :  $\text{Rand}(C, C') = \frac{A+D}{M}$
- Jaccard :  $\text{Jaccard}(C, C') = \frac{A}{A+B+C}$

## Comparing partitions

Many external evaluation measures in clustering are statistics used to compare two partitions.

- Rand :  $\text{Rand}(C, C') = \frac{A+D}{M}$
- Jaccard :  $\text{Jaccard}(C, C') = \frac{A}{A+B+C}$
- Folkes and Mallows :  $\text{FM}(C, C') = \frac{A}{\sqrt{(A+B)(A+C)}}$

## Comparing partitions

Many external evaluation measures in clustering are statistics used to compare two partitions.

- Rand :  $\text{Rand}(C, C') = \frac{A+D}{M}$
- Jaccard :  $\text{Jaccard}(C, C') = \frac{A}{A+B+C}$
- Folkes and Mallows :  $\text{FM}(C, C') = \frac{A}{\sqrt{(A+B)(A+C)}}$
- Russel and Rao :  $\text{RR}(C, C') = \frac{A}{M}$
- Phi :  $\text{Phi}(C, C') = \frac{AD-BC}{(A+B)(A+C)(D+B)(D+C)}$

## Comparing partitions

Many external evaluation measures in clustering are statistics used to compare two partitions.

- Rand :  $\text{Rand}(C, C') = \frac{A+D}{M}$
- Jaccard :  $\text{Jaccard}(C, C') = \frac{A}{A+B+C}$
- Folkes and Mallows :  $\text{FM}(C, C') = \frac{A}{\sqrt{(A+B)(A+C)}}$
- Russel and Rao :  $\text{RR}(C, C') = \frac{A}{M}$

## Comparing partitions





Many external evaluation measures in clustering are statistics used to compare two partitions.

- Rand :  $\text{Rand}(C, C') = \frac{A+D}{M}$
- Jaccard :  $\text{Jaccard}(C, C') = \frac{A}{A+B+C}$
- Folkes and Mallows :  $\text{FM}(C, C') = \frac{A}{\sqrt{(A+B)(A+C)}}$
- Russel and Rao :  $\text{RR}(C, C') = \frac{A}{M}$
- Phi :  $\text{Phi}(C, C') = \frac{AD-BC}{(A+B)(A+C)(D+B)(D+C)}$

Note that these measures are similar to similarity measures between binary vectors.

## Example using R

```
> data("iris", package="datasets")
> library(clusterSim)
> iris.normalization=data.Normalization(iris[,-5], type="n1")
> iris_kmeans_res=kmeans(x=iris.normalization, centers=3)
> iris_labels=as.integer(iris[,5])
> install.packages("clv")
> library(clv)
> iris_kmeans_res_ext_eval=std.ext(iris_kmeans_res$cluster, iris_labels)
> clv.Rand(iris_kmeans_res_ext_eval)
[1] 0.8322148
> clv.Jaccard(iris_kmeans_res_ext_eval)
[1] 0.5938921
> clv.Folkes.Mallows(iris_kmeans_res_ext_eval)
[1] 0.7452105
> clv.Russel.Rao(iris_kmeans_res_ext_eval)
[1] 0.2453691
```

-  [Bezdek, J. C. \(1973\). Fuzzy Mathematics in Pattern Classification.](#)  
PhD thesis, Applied Math. Center, Cornell University, Ithaca.
-  [Bishop, C. M. \(2006\). Pattern recognition and machine learning.](#)  
Springer, 1st ed. 2006. corr. 2nd printing edition.
-  [Bock, H.-H. \(2007\). Clustering Methods : A History of k-Means Algorithms.](#)  
In Brito, P., Cucumel, G., Bertrand, P., and Carvalho, F., editors, *Selected Contributions in Data Analysis and Classification*, Studies in Classification, Data Analysis, and Knowledge Organization, chapter 15, pages 161–172. Springer Berlin Heidelberg, Berlin, Heidelberg.
-  [Bradley, P. S. and Fayyad, U. M. \(1998\). Refining initial points for k-means clustering.](#)  
In *ICML*, pages 91–99.
-  [Dempster, A. P., Laird, N. M., and Rubin, D. B. \(1977\). Maximum Likelihood from Incomplete Data via the EM Algorithm.](#)  
*Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1) :1–38.
-  [Dhillon, I. S., Guan, Y., and Kulis, B. \(2004\). Kernel k-means : spectral clustering and normalized cuts.](#)  
In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '04*, pages 551–556, New York, NY, USA. ACM.
-  [Fraleigh, C. and Raftery, A. E. \(\(2006, revised in 2009\)\)\). MCLUST version 3 for R : Normal mixture modeling and model-based clustering.](#)  
Technical Report 504, University of Washington, Department of Statistics.

## Assignment using R









**Exercise 16 :** Take a small datasets (no more than a few undreds of objects) designed for classification tasks from UCI ML

<http://archive.ics.uci.edu/ml/>.

With R, apply several clustering techniques (at least one HC, one hard partitioning and one fuzzy clustering) and compare the clustering outputs using some external validity indices (via the comparisons of the different outputs to the ground-truth). The code should be commented with some concise explanations regarding the used libraries, the applied clustering techniques corresponding to the employed functions and the specified parameters of the latter.

In order to help you do this exercise, you can also consult the following webpage :

<http://cran.at.r-project.org/web/views/Cluster.html>.

-  [Gan, G., Ma, C., and Wu, J. \(2007\). Data Clustering : Theory, Algorithms, and Applications \(ASA-SIAM Series on Statistics and Applied Probability\).](#)  
SIAM, Society for Industrial and Applied Mathematics, illustrated edition edition.
-  [Hamerly, G. \(2010\). Making k-means even faster.](#)  
In *SDM*, pages 130–140.
-  [Huang, Z. \(1998\). Extensions to the k-means algorithm for clustering large data sets with categorical values.](#)  
*Data Min. Knowl. Discov.*, 2(3) :283–304.
-  [Huang, Z. and Ng, M. \(1999\). IEEE Transactions on Fuzzy Systems, 7\(4\) :446–452.](#)
-  [Kaufman, L. and Rousseeuw, P. J. \(2005\). Finding Groups in Data : An Introduction to Cluster Analysis \(Wiley Series in Probability and Statistics\).](#)  
Wiley-Interscience.
-  [Khan, S. \(2004\). Cluster center initialization algorithm for K-means clustering.](#)  
*Pattern Recognition Letters*, 25(11) :1293–1302.
-  [Mclachlan, G. and Peel, D. \(2000\). Finite Mixture Models.](#)  
Wiley Series in Probability and Statistics. Wiley-Interscience, 1 edition.
-  [Milligan, G. and Cooper, M. \(1985\). An examination of procedures for determining the number of clusters in a data set.](#)  
*Psychometrika*, 50 :159–179.  
10.1007/BF02294245.





Ng, R. T. and Han, J. (1994).

Efficient and effective clustering methods for spatial data mining.

In [Proceedings of the 20th International Conference on Very Large Data Bases, VLDB '94](#), pages 144–155, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.



Pelleg, D. and Moore, A. (2000).

X-means : Extending k-means with efficient estimation of the number of clusters.

In [In Proceedings of the 17th International Conf. on Machine Learning](#), pages 727–734. Morgan Kaufmann.



Steinbach, M., Karypis, G., and Kumar, V. (2000).

A comparison of document clustering techniques.

In [Gobelnik, M., Mladenic, D., and Milic-Frayling, N., editors, KDD-2000 Workshop on Text Mining, August 20](#), pages 109–111, Boston, MA.



Steinley, D. (2006).

K-means clustering : A half-century synthesis.

[British Journal of Mathematical and Statistical Psychology](#), 59(1) :1–34.