# Cluster analysis

David J. Hessen

Utrecht University

Academic year 2023-2024

# Overview

- Introduction
- Dissimilarity
- Clustering algorithms ($K$-means, Gaussian mixtures, $K$-medoids)
- Selection of the number of clusters
- Hierarchical clustering
- Practical issues

# Research question

Are there a few different unknown subgroups of cases or clusters?

## Examples

- Can different types of cancer be distinguished based on tumor stage, tumor grade, and gene expressions?
- Are there clusters of consumers who might be more receptive to a particular form of advertising?

# Goals

- Grouping or segmenting a collection of cases into subsets or ‘clusters,’ such that those within each cluster are more similar than cases assigned to different clusters
- Arranging clusters into a natural hierarchy
- To form descriptive statistics to ascertain whether or not the data consist of a number of distinct subgroups

# Goals

Based on the features $x_1, \ldots, x_p$, the cases are partitioned into a (pre-specified) number of clusters → the number $K$

The feature data matrix

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1p} \\ x_{21} & x_{22} & \ldots & x_{2p} \\ \vdots & \vdots & & \vdots \\ x_{N1} & x_{N2} & \ldots & x_{Np} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \ldots & \mathbf{x}_N \end{bmatrix}^T$$

Based on $\mathbf{x}_i = \begin{bmatrix} x_{i1} & \ldots & x_{ip} \end{bmatrix}^T$ each case $i \in \{1, \ldots, N\}$ is assigned to only one cluster, that is,

$$C(i) = k, \quad \text{where } k \in \{1, \ldots, K\}$$

Usually, $K$ is much smaller than $N$

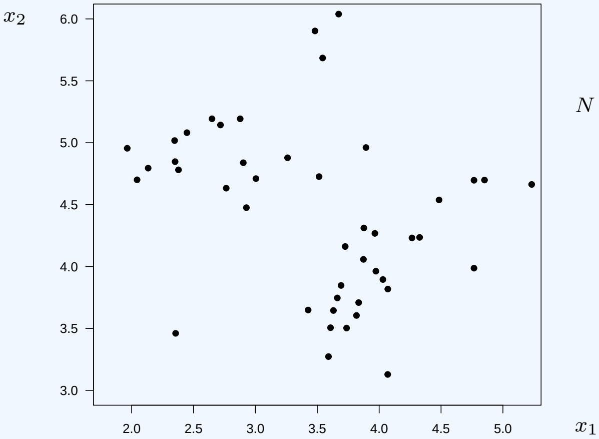
# Goals

## Example

| $i$ | $x_{i1}$ | $x_{i2}$ |
|---|---|---|
| 1 | 2.38 | 4.78 |
| 2 | 3.48 | 5.90 |
| 3 | 2.04 | 4.70 |
| 4 | 2.88 | 5.19 |
| 5 | 3.67 | 6.04 |
| 6 | 2.76 | 4.63 |
| 7 | 2.45 | 5.08 |
| 8 | 2.35 | 5.02 |
| 9 | 3.00 | 4.71 |
| 10 | 3.26 | 4.88 |
| 11 | 3.54 | 5.68 |
| 12 | 2.35 | 4.85 |
| 13 | 1.96 | 4.96 |
| 14 | 2.65 | 5.19 |
| 15 | 2.13 | 4.79 |
| 16 | 2.72 | 5.14 |
| 17 | 2.93 | 4.48 |
| 18 | 2.36 | 3.46 |
| 19 | 3.51 | 4.73 |
| 20 | 2.90 | 4.84 |
| 21 | 4.85 | 4.70 |
| 22 | 4.07 | 3.13 |
| 23 | 4.03 | 3.90 |
| 24 | 5.23 | 4.66 |
| 25 | 3.83 | 3.71 |
| 26 | 4.33 | 4.24 |
| 27 | 3.61 | 3.51 |
| 28 | 3.69 | 3.85 |
| 29 | 3.63 | 3.65 |
| 30 | 3.87 | 4.06 |
| 31 | 4.77 | 4.70 |
| 32 | 3.88 | 4.31 |
| 33 | 4.48 | 4.54 |
| 34 | 3.66 | 3.75 |
| 35 | 4.07 | 3.82 |
| 36 | 3.43 | 3.65 |
| 37 | 4.77 | 3.99 |
| 38 | 3.73 | 4.16 |
| 39 | 3.97 | 3.96 |
| 40 | 3.97 | 4.27 |
| 41 | 3.59 | 3.27 |
| 42 | 3.89 | 4.96 |
| 43 | 3.82 | 3.60 |
| 44 | 4.27 | 4.23 |
| 45 | 3.74 | 3.50 |

# Goals

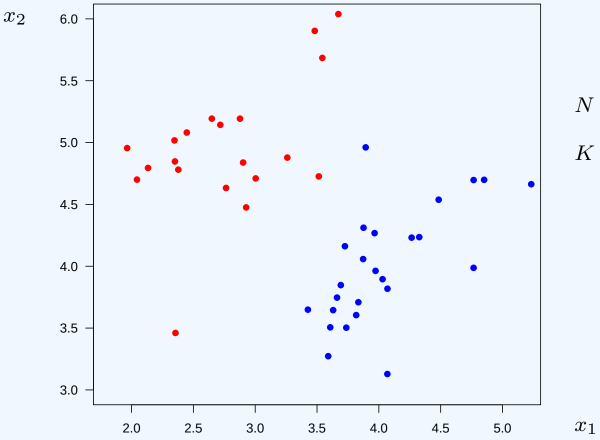
## Example (continued)



$N = 45$

# Goals

## Example (continued)

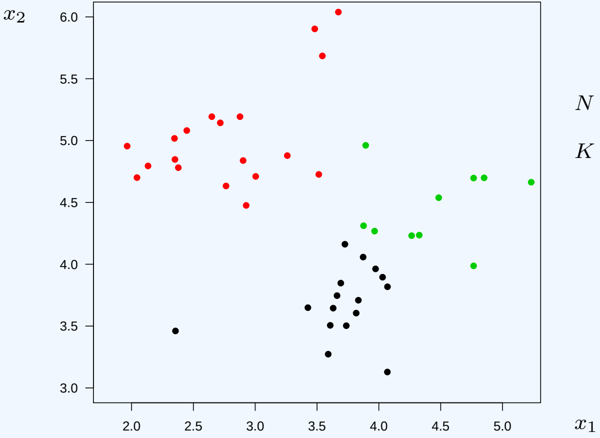


$N = 45$

$K = 2$

# Goals

## Example (continued)



$N = 45$

$K = 3$

# Proximity matrices

Data → directly in terms of the proximity between any two cases

The $N \times N$ proximity (dissimilarity) matrix

$$\mathbf{D} = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1N} \\ d_{21} & d_{22} & \dots & d_{2N} \\ \vdots & \vdots & & \\ d_{N1} & d_{N2} & \dots & d_{NN} \end{bmatrix}$$

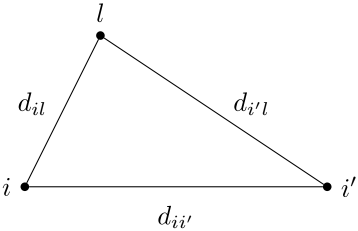
where $d_{ii'}$ is the proximity between cases $i$ and $i'$

Usually $d_{ii} = 0$, for all $i$

If the (dis)similarities between cases are subjectively judged, then $\mathbf{D}$ might not be symmetric and is replaced by $(\mathbf{D} + \mathbf{D}^T)/2$

# Proximity matrices

If $d_{ii'} \leq d_{il} + d_{i'l}$ does not hold for all $i$, $i'$, and $l$, then algorithms that assume distance cannot be used



# Dissimilarities based on attributes

Most often a *dissimilarity* between cases $i$ and $i'$ is constructed on the basis of

$$\mathbf{x}_i = [\, x_{i1} \;\; \ldots \;\; x_{ip} \,]^T \quad \text{and} \quad \mathbf{x}_{i'} = [\, x_{i'1} \;\; \ldots \;\; x_{i'p} \,]^T$$

The dissimilarity $d_{ii'}$ between cases $i$ and $i'$ is then defined as

$$D(\mathbf{x}_i, \mathbf{x}_{i'}) = \sum_{j=1}^{p} d_j(x_{ij}, x_{i'j})$$

where $d_j(x_{ij}, x_{i'j})$ is the dissimilarity between cases $i$ and $i'$ on the $j$th feature

# Dissimilarities based on attributes

In the case of quantitative features, the most common choice is the squared distance

$$d_j(x_{ij}, x_{i'j}) = (x_{ij} - x_{i'j})^2$$

Another choice is the absolute difference

$$d_j(x_{ij}, x_{i'j}) = |x_{ij} - x_{i'j}|$$

Alternatively, clustering can be based on the correlation

$$r(\mathbf{x}_i, \mathbf{x}_{i'}) = \frac{\sum_j (x_{ij} - \bar{x}_i)(x_{i'j} - \bar{x}_{i'})}{\sqrt{\sum_j (x_{ij} - \bar{x}_i)^2 \sum_j (x_{i'j} - \bar{x}_{i'})^2}}$$

where $\bar{x}_i = \sum_j x_{ij}/p$ and $\bar{x}_{i'} = \sum_j x_{i'j}/p$

# Dissimilarities based on attributes

In the case of ordinal features, the assigned values $x_{ij} \in \{1, \ldots, M_j\}$ can be transformed using

$$\frac{x_{ij} - 1/2}{M_j}$$

and then treated as quantitative

In the case of nominal features, a popular choice is

$$d_j(x_{ij}, x_{i'j}) = \begin{cases} 0, & \text{if } x_{ij} = x_{i'j} \\ 1, & \text{if } x_{ij} \neq x_{i'j} \end{cases}$$

However, latent class analysis is probably more suitable for categorical variables

# Case (object) dissimilarity

A more general measure of case dissimilarity is the weighted average

$$D(\mathbf{x}_i, \mathbf{x}_{i'}) = \sum_{j=1}^{p} w_j d_j(x_{ij}, x_{i'j})$$

where $\sum_{j=1}^{p} w_j = 1$ and the choice of the weight $w_j$ should be based on subject matter considerations

Setting $w_j = 1$ or $w_j = 1/p$, for all $j$, does not necessarily give all features equal influence in characterizing overall dissimilarity between cases

# Case (object) dissimilarity

The average case dissimilarity over all pairs of cases is

$$
\begin{aligned}
\bar{D} &= \frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'=1}^{N} D(\mathbf{x}_i, \mathbf{x}_{i'}) \\
&= \frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'=1}^{N} \sum_{j=1}^{p} w_j d_j(x_{ij}, x_{i'j}) \\
&= \sum_{j=1}^{p} w_j \frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'=1}^{N} d_j(x_{ij}, x_{i'j}) \\
&= \sum_{j=1}^{p} w_j \bar{d}_j
\end{aligned}
$$

where $\bar{d}_j$ is the average case dissimilarity on the $j$th feature

# Case (object) dissimilarity

The relative influence of the $j$th feature is $w_j \bar{d}_j$

Setting $w_j = 1/\bar{d}_j$, where

$$\bar{d}_j = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'=1}^{N} d_j(x_{ij}, x_{i'j})$$

gives all attributes equal influence in characterizing overall dissimilarity between cases (often recommended but can be highly counterproductive)

# Object dissimilarity

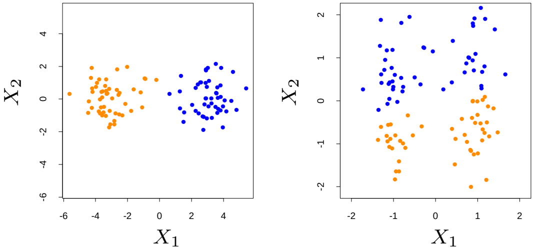


**FIGURE 14.5.** *Simulated data: on the left, $K$-means clustering (with $K=2$) has been applied to the raw data. The two colors indicate the cluster memberships. On the right, the features were first standardized before clustering. This is equivalent to using feature weights* $1/[2 \cdot \mathrm{var}(X_j)]$. *The standardization has obscured the two well-separated groups. Note that each plot uses the same units in the horizontal and vertical axes.*

# Combinatorial algorithms

The *total* point scatter does not depend on $K$ and is given by

$$T = \frac{1}{2}\sum_{i=1}^{N}\sum_{i'=1}^{N} D(\mathbf{x}_i, \mathbf{x}_{i'}) = W(C) + B(C)$$

where

$$W(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(i')=k} D(\mathbf{x}_i, \mathbf{x}_{i'})$$

is the *within-cluster* point scatter, and

$$B(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(i')\neq k} D(\mathbf{x}_i, \mathbf{x}_{i'})$$

is the *between-cluster* point scatter

$W(C)$ is minimized (or equivalently $B(C)$ is maximized) over all possible assignments of the $N$ data points to $K$ clusters

# Combinatorial algorithms

Unfortunately, combinatorial optimization by complete enumeration is feasible only for very small data sets

The total number of distinct assignments of $N$ cases to $K$ clusters is

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

Note that $S(10, 4) = 34,105$ and $S(19, 4) \approx 10^{10}$

Practically feasible strategies are based on iterative greedy descent

- An initial partition is specified
- At each iteration, the assignment of data points to clusters is improved

# $K$-means clustering

The squared Euclidean distance

$$D(\mathbf{x}_i, \mathbf{x}_{i'}) = \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

is chosen as the case dissimilarity measure

The *within-cluster* point scatter

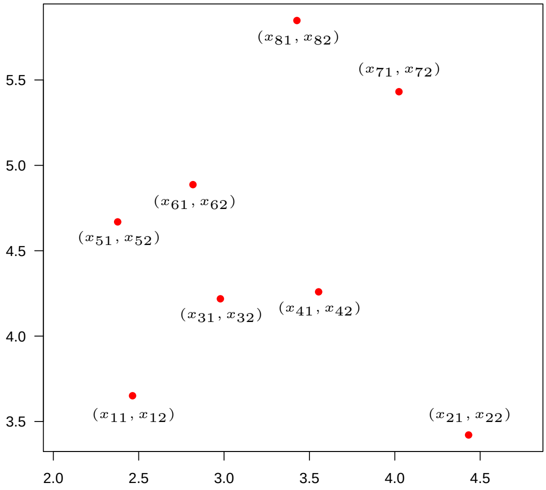
$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(i')=k} D(\mathbf{x}_i, \mathbf{x}_{i'}) = \sum_{k=1}^{K} N_k \sum_{C(i)=k} \overbrace{\sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2}^{\text{squared Euclidean distance to centroid } \bar{\mathbf{x}}_k}$$

where $N_k$ is the number of observations assigned to cluster $k$, is minimized

The centroid of cluster $k$ is $\bar{\mathbf{x}}_k = [\, \bar{x}_{k1} \ \bar{x}_{k2} \ \ldots \ \bar{x}_{kp} \,]^T$

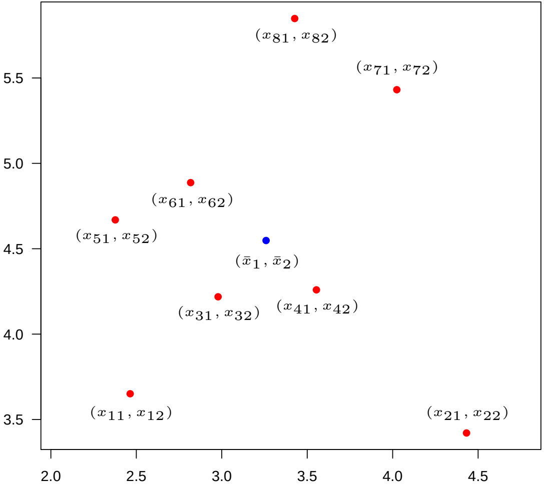
# $K$-means clustering

Scatter plot



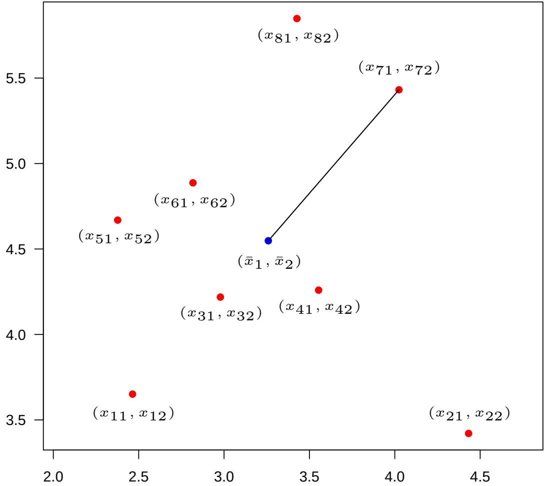
# $K$-means clustering

$(\bar{x}_1, \bar{x}_2)$ is the centroid



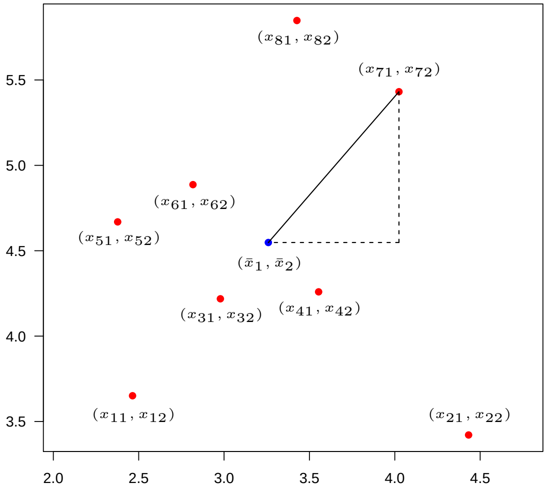
# $K$-means clustering

Euclidean distance between $(x_{71}, x_{72})$ and $(\bar{x}_1, \bar{x}_2)$



# $K$-means clustering

Euclidean distance between $(x_{71}, x_{72})$ and $(\bar{x}_1, \bar{x}_2)$ is $\sqrt{(x_{71} - \bar{x}_1)^2 + (x_{72} - \bar{x}_2)^2}$



# $K$-means clustering

The most popular iterative descent algorithm

Assignment procedure

1. A random number, from 1 to $K$, is assigned to each of the observations
2. Iteratively

   (a) cluster centroids are computed: $\bar{\mathbf{x}}_k = (\bar{x}_{k1}, \ldots, \bar{x}_{kp})$, for $k = 1, \ldots, K$

   (b) each observation is assigned to the cluster whose centroid is closest

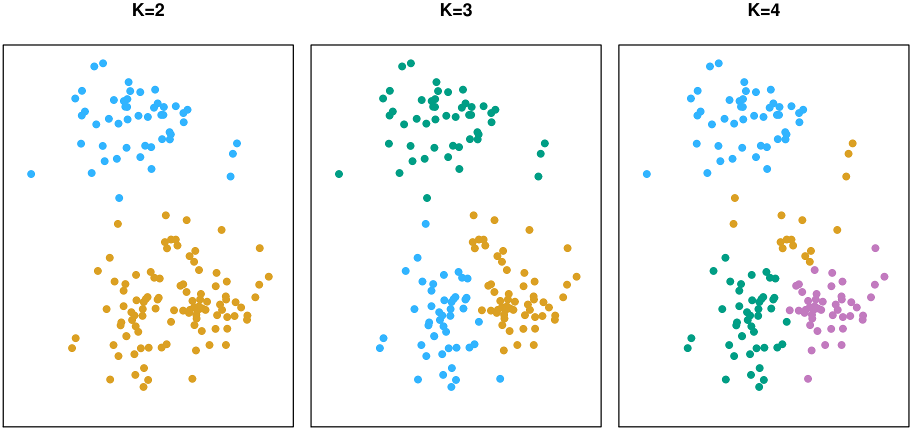
   until the cluster assignments stop changing

Since the result might be a suboptimal local minimum

→ the algorithm should be started with many random cluster assignments

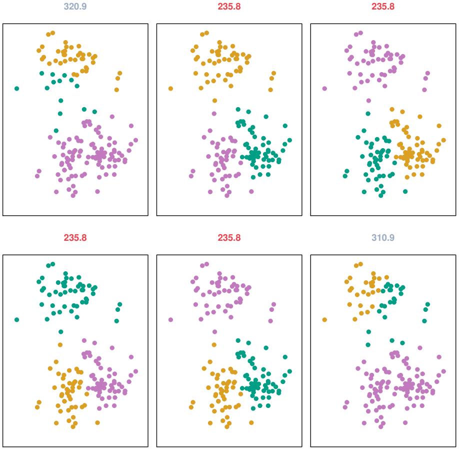
# $K$-means clustering

Simulated data with $N = 150$, two features, and different values of $K$



# $K$-means clustering

Because the algorithm finds a local rather than a global minimum, the results depend on the initial random cluster assignment



# Gaussian mixtures as soft $K$-means clustering

Model based clustering

## The Gaussian mixture model

The multivariate density of $\mathbf{x} = (x_1, \ldots, x_p)$ is

$$g(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \cdot n(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

where $\pi_k$ is a mixing probability, $n(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is a multivariate normal density, $\boldsymbol{\mu}_k$ is the mean vector, and $\boldsymbol{\Sigma}_k$ is the covariance matrix, for cluster $k$

The maximum likelihood estimates of $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ and $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K$ are the values that maximize the log-likelihood function

$$l(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K) = \sum_{i=1}^{N} \ln\left\{\sum_{k=1}^{K} \pi_k \cdot n(\mathbf{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right\}$$

# Gaussian mixtures as soft $K$-means clustering

EM algorithm for obtaining the maximum likelihood estimates

1. Take initial guesses for $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ and $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K$
2. *Expectation step*: compute a provisional estimate of the responsibility

$$\gamma_{ik} = \Pr(C_i = k \,|\, \mathbf{x}_i) = \frac{\pi_k n(\mathbf{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum\limits_{k=1}^{K} \pi_k n(\mathbf{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}, \quad \text{for all } i \text{ and } k$$

3. *Maximization step*: compute provisional estimates of

$$\boldsymbol{\mu}_k = \frac{\sum\limits_{i=1}^{N} \gamma_{ik} \mathbf{x}_i}{\sum\limits_{i=1}^{N} \gamma_{ik}}, \; \boldsymbol{\Sigma}_k = \frac{\sum\limits_{i=1}^{N} \gamma_{ik} (\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T}{\sum\limits_{i=1}^{N} \gamma_{ik}}, \text{ and } \pi_k = \sum_{i=1}^{N} \gamma_{ik}/N$$

for all $k$

4. Iterate steps 2 and 3 until convergence

# Gaussian mixtures as soft $K$-means clustering

Suppose $K$ mixture components, each with a multivariate Gaussian density having (scalar) covariance matrix

$$\boldsymbol{\Sigma}_k = \sigma^2 \mathbf{I} = \begin{bmatrix} \sigma^2 & & & \\ 0 & \sigma^2 & & \\ \vdots & & \ddots & \\ 0 & 0 & \dots & \sigma^2 \end{bmatrix}$$

In this setup, the EM algorithm is a ‘soft’ version of the $K$-means algorithm, making probabilistic assignments of observations to clusters

As $\sigma^2 \to 0$, each responsibility becomes either 0 or 1, and the two methods coincide

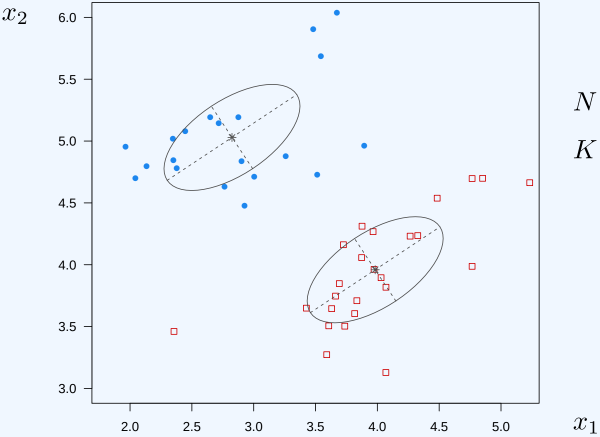
# Gaussian mixtures as soft $K$-means clustering

## Example (continued)

| $i$ | $\hat{\gamma}_{i1}$ | $\hat{\gamma}_{i2}$ | $k$ |
|---|---|---|---|
| 1 | 1.000 | 0.000 | 1 |
| 2 | 1.000 | 0.000 | 1 |
| 3 | 1.000 | 0.000 | 1 |
| 4 | 1.000 | 0.000 | 1 |
| 5 | 1.000 | 0.000 | 1 |
| 6 | 1.000 | 0.000 | 1 |
| 7 | 1.000 | 0.000 | 1 |
| 8 | 1.000 | 0.000 | 1 |
| 9 | 0.999 | 0.001 | 1 |
| 10 | 0.998 | 0.002 | 1 |
| 11 | 1.000 | 0.000 | 1 |
| 12 | 1.000 | 0.000 | 1 |
| 13 | 1.000 | 0.000 | 1 |
| 14 | 1.000 | 0.000 | 1 |
| 15 | 1.000 | 0.000 | 1 |
| 16 | 1.000 | 0.000 | 1 |
| 17 | 0.986 | 0.014 | 1 |
| 18 | 0.026 | 0.974 | 2 |
| 19 | 0.852 | 0.148 | 1 |
| 20 | 1.000 | 0.000 | 1 |
| 21 | 0.000 | 1.000 | 2 |
| 22 | 0.000 | 1.000 | 2 |
| 23 | 0.000 | 1.000 | 2 |
| 24 | 0.000 | 1.000 | 2 |
| 25 | 0.000 | 1.000 | 2 |
| 26 | 0.000 | 1.000 | 2 |
| 27 | 0.000 | 1.000 | 2 |
| 28 | 0.000 | 1.000 | 2 |
| 29 | 0.000 | 1.000 | 2 |
| 30 | 0.000 | 1.000 | 2 |
| 31 | 0.000 | 1.000 | 2 |
| 32 | 0.001 | 0.999 | 2 |
| 33 | 0.000 | 1.000 | 2 |
| 34 | 0.000 | 1.000 | 2 |
| 35 | 0.000 | 1.000 | 2 |
| 36 | 0.000 | 1.000 | 2 |
| 37 | 0.000 | 1.000 | 2 |
| 38 | 0.000 | 1.000 | 2 |
| 39 | 0.000 | 1.000 | 2 |
| 40 | 0.000 | 1.000 | 2 |
| 41 | 0.000 | 1.000 | 2 |
| 42 | 0.751 | 0.249 | 1 |
| 43 | 0.000 | 1.000 | 2 |
| 44 | 0.000 | 1.000 | 2 |
| 45 | 0.000 | 1.000 | 2 |

# Gaussian mixtures as soft $K$-means clustering

## Example (continued)

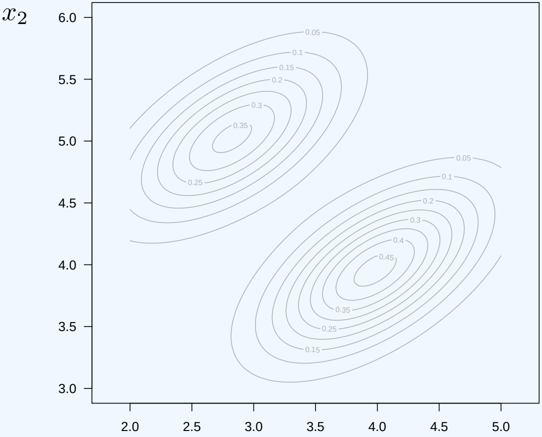


$N = 45$

$K = 2$

# Gaussian mixtures as soft $K$-means clustering

## Example (continued)

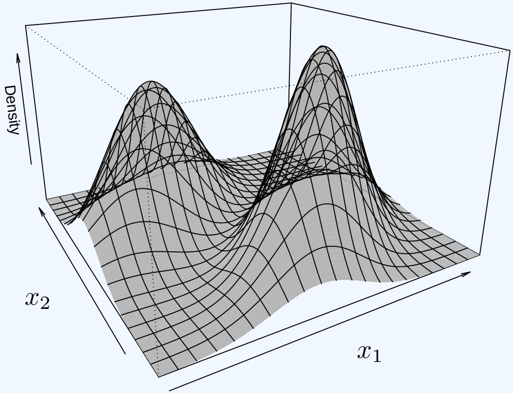


$N = 45$

$K = 2$

# Gaussian mixtures as soft $K$-means clustering

## Example (continued)



$N = 45$

$K = 2$

# $K$-medoids

Disadvantages of the $K$-means algorithm

- the features are required to be of the quantitative type
- squared Euclidean distance places the highest influence on the largest distances → not robust against outliers

Instead of the centroids, the $K$-medoids algorithm chooses cases as centers and minimizes the dissimilarity between cases labeled to be in a cluster and the case designated as the center of that cluster

It is more robust to noise and outliers as compared to $K$-means because it minimizes a sum of pairwise dissimilarities

A medoid can be defined as the case of a cluster whose average dissimilarity to all the cases in the cluster is minimal

# Selection of the number of clusters

- The within-cluster point scatter or dissimilarity $W_K$ can be plotted as a function of the number of clusters $K$

  There will be a sharp decrease in $W_K - W_{K+1}$ at $K = K^*$, where $K^*$ is the true number of clusters

  An estimate for $K^*$ is obtained by identifying a 'kink' in the plot of $W_K$ as a function of $K$

- The optimal number of cluster is the place where the gap between the curve $\log W_K$ and the curve obtained from data uniformly distributed over a rectangle containing the data, is largest

# Selection of the number of clusters

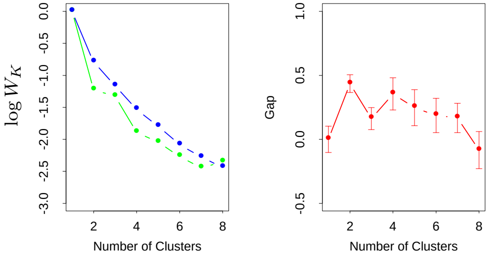


**FIGURE 14.11.** *(Left panel): observed (green) and expected (blue) values of* $\log W_K$ *for the simulated data of Figure 14.4. Both curves have been translated to equal zero at one cluster. (Right panel): Gap curve, equal to the difference between the observed and expected values of* $\log W_K$. *The Gap estimate* $K^*$ *is the smallest* $K$ *producing a gap within one standard deviation of the gap at* $K + 1$*; here* $K^* = 2$.

# Hierarchical clustering

Bottom-up or agglomerative clustering

Iteratively

1. dissimilarities are measured between each two clusters (the initial observations are also seen as clusters)
2. the two clusters that are most similar to each other are fused to form a new cluster

until all the observations belong to one single cluster

Advantages

- Does not require the pre-specification of the number of clusters
- It provides a dendogram (a tree-based representation of the observations)

# Hierarchical clustering

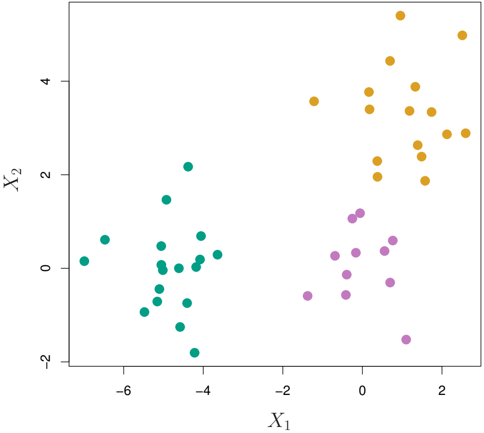
A usual measure of dissimilarity is Euclidean distance

How is dissimilarity defined between clusters?

| **Linkage** | **Description** |
|---|---|
| *Complete* | Maximal pairwise intercluster dissimilarity<br>*Furthest-neighbor*<br>→ violates the ‘closeness’ property |
| *Single* | Minimal pairwise intercluster dissimilarity<br>*Nearest-neighbor*<br>→ violates the ‘compactness’ property (chaining) |
| *Average* | Mean pairwise intercluster dissimilarity |
| *Centroid* | Centroid dissimilarity |

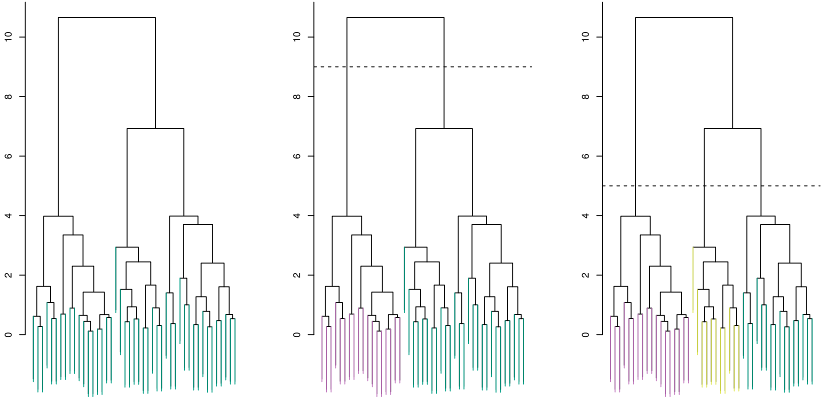
# Hierarchical clustering

Simulated data with $N = 45$, two features, and three classes

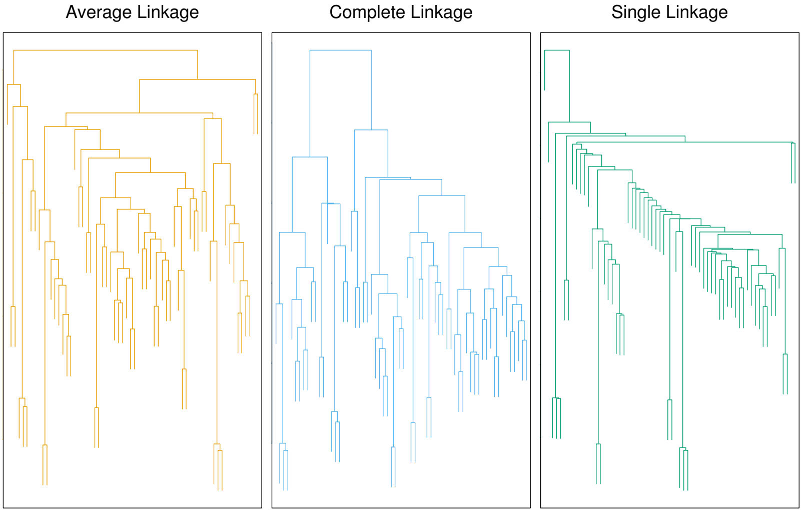


# Hierarchical clustering

Dendograms for the 45 simulated observations, 2 features, and 3 classes
Complete linkage and Euclidean distance



# Hierarchical clustering



# Practical issues

- Should the features be standardized?
- In the case of hierarchical clustering,
  - What dissimilarity measure should be used?
  - What type of linkage should be used?
  - Where should we cut the dendogram in order to obtain clusters?
- In the case of $K$-means clustering, how many clusters should we look for in the data?

In practice, several choices should be tried, and the one with the most useful or interpretable solution should be selected