Cluster Analysis

Methods and Open Problems

Frank Klawonn

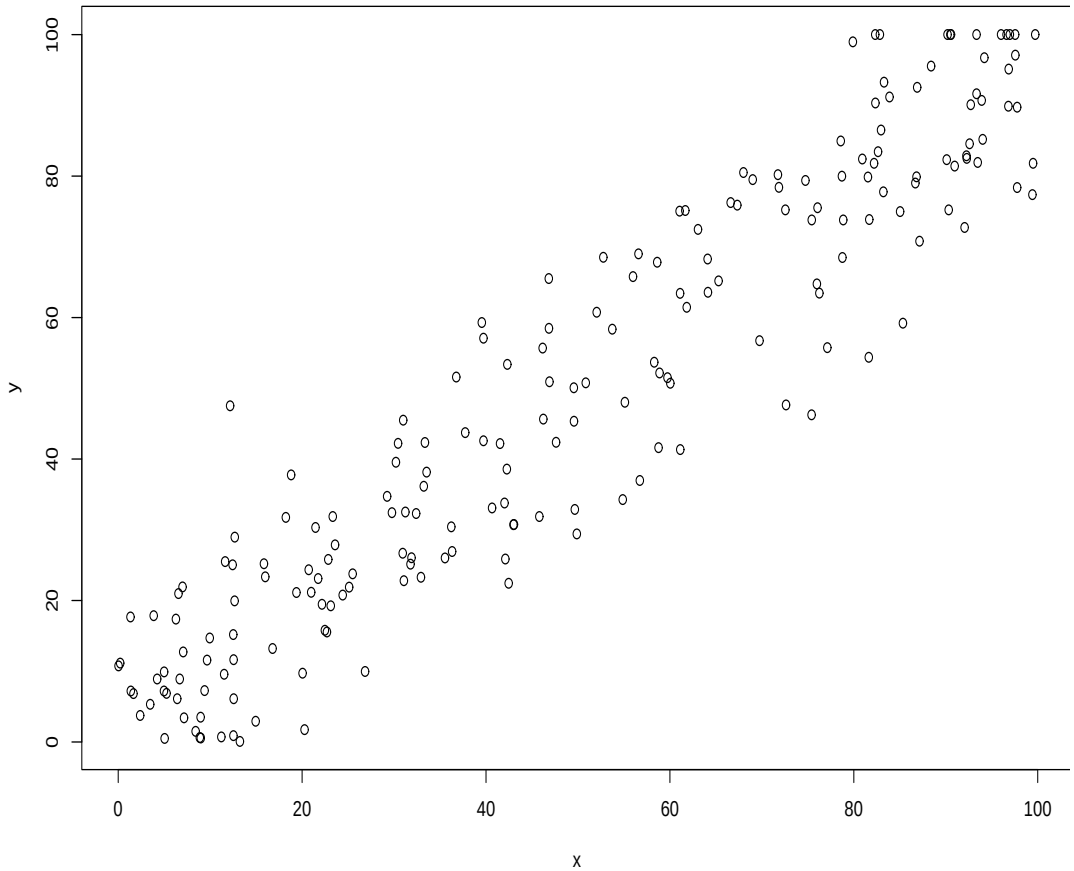
Institute of Applied Informatics, Department of Computer Science
Ostfalia University of Applied Sciences
Wolfenbuettel, Germany
f.klawonn@ostfalia.de

Bioinformatics & Statistics
Helmholtz Centre for Infection Research
Braunschweig, Germany
frank.klawonn@helmholtz-hzi.de

Contents

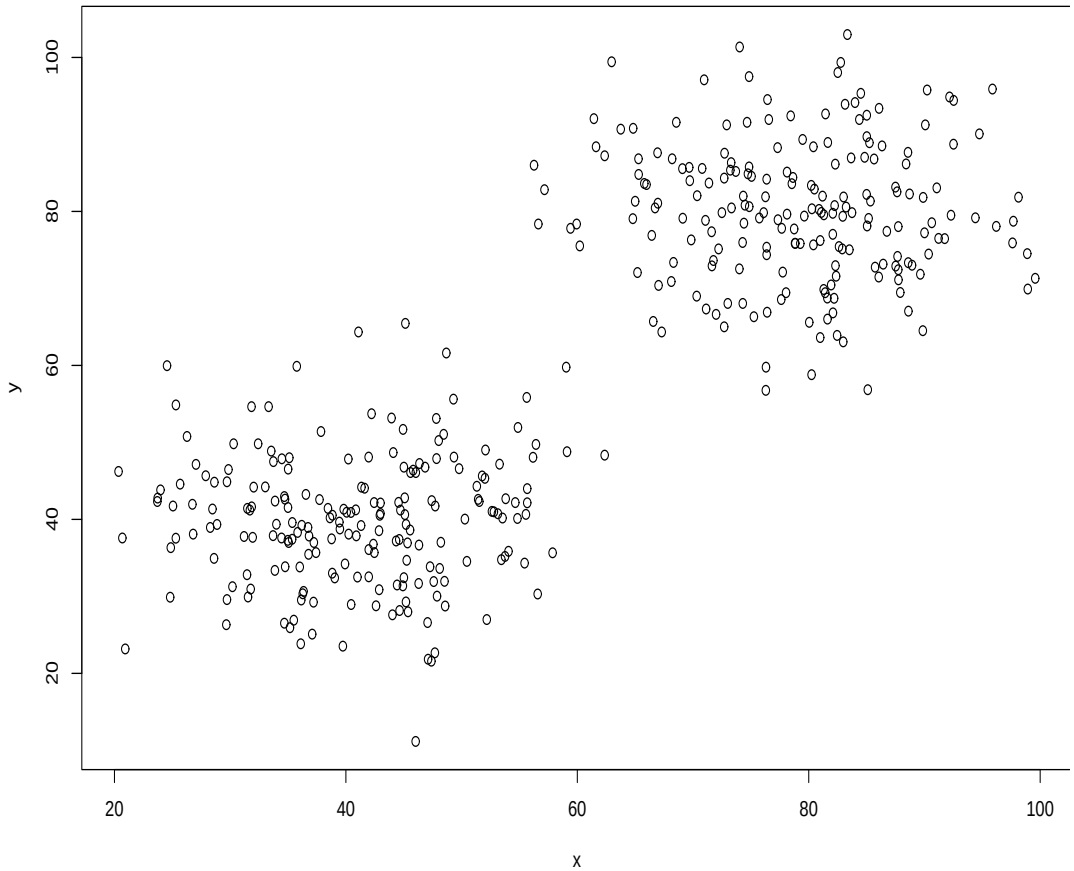
What is cluster analysis?

Percentage of points achieved by students in mathematics (x) and physics (y) exams



What is cluster analysis?

Percentage of points achieved by students in mathematics (x) and physics (y) exams



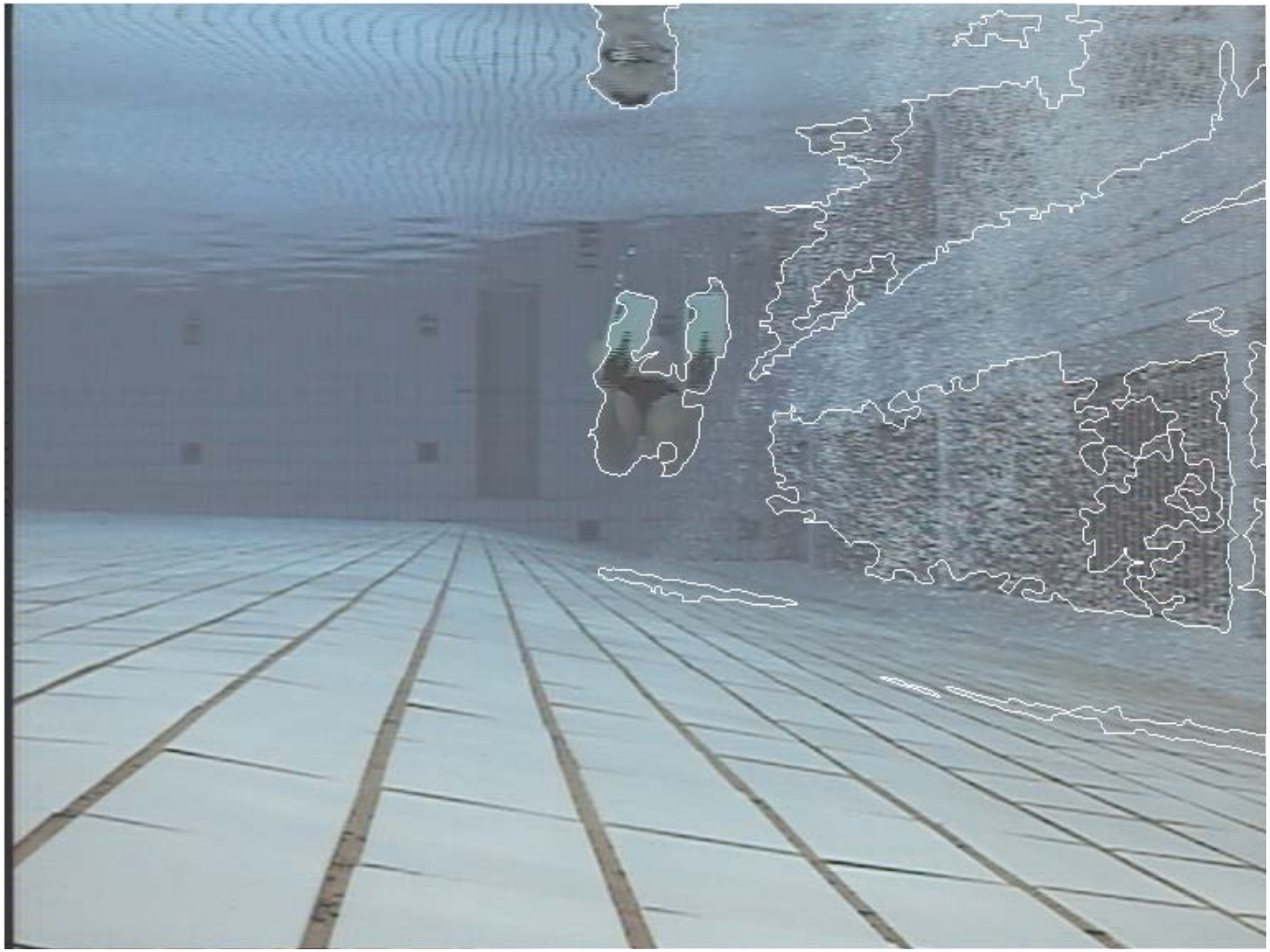
What is cluster analysis?

A cluster (in a given data set) is a subset of data, so that the data

- within the cluster are "similar"
- and differ from the data outside the cluster.

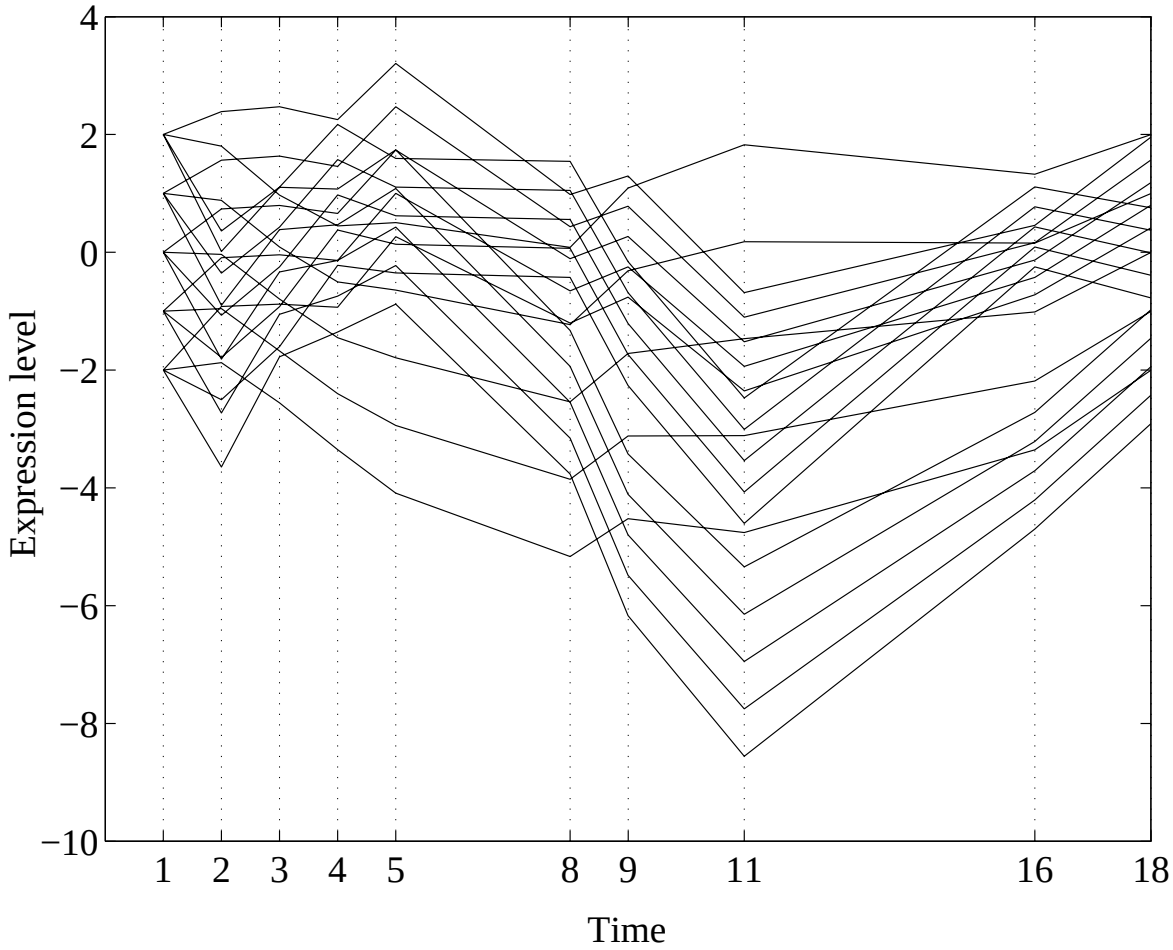
Data inside a cluster should be homogeneous, data from different clusters heterogeneous.

What is cluster analysis?

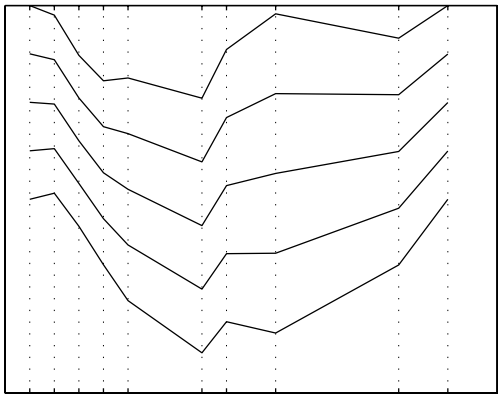


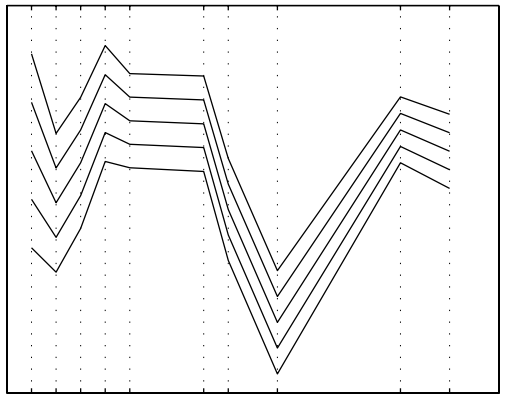
What is cluster analysis?

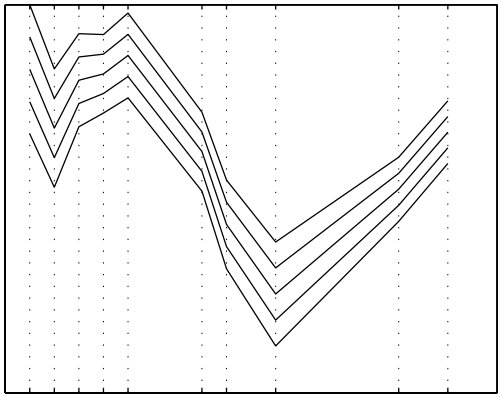
What is cluster analysis?

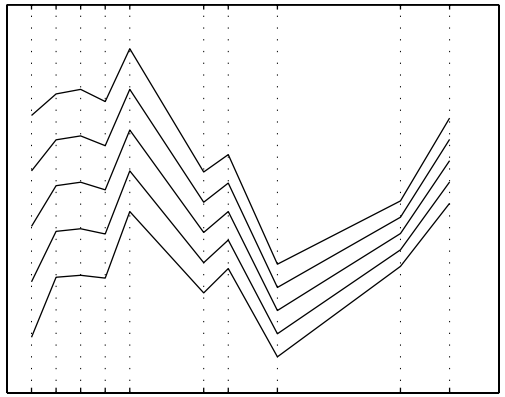


What is cluster analysis?

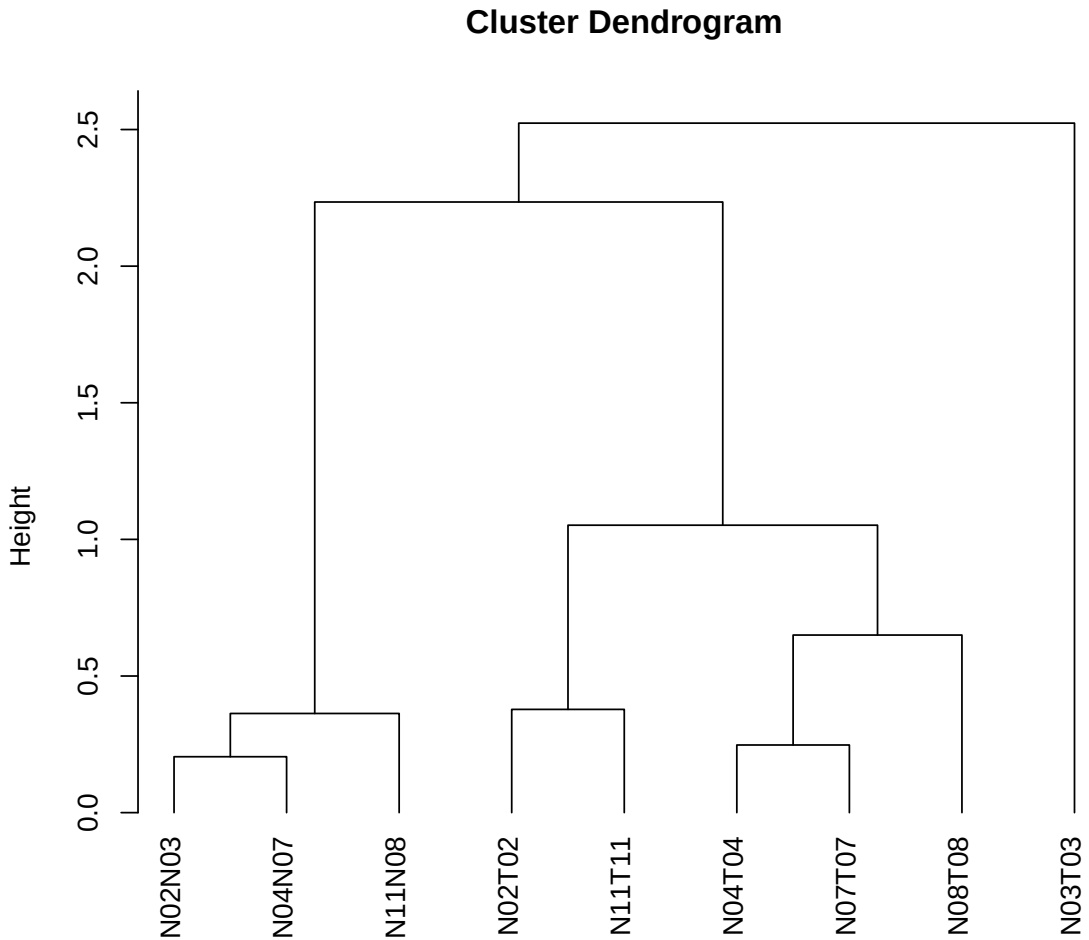




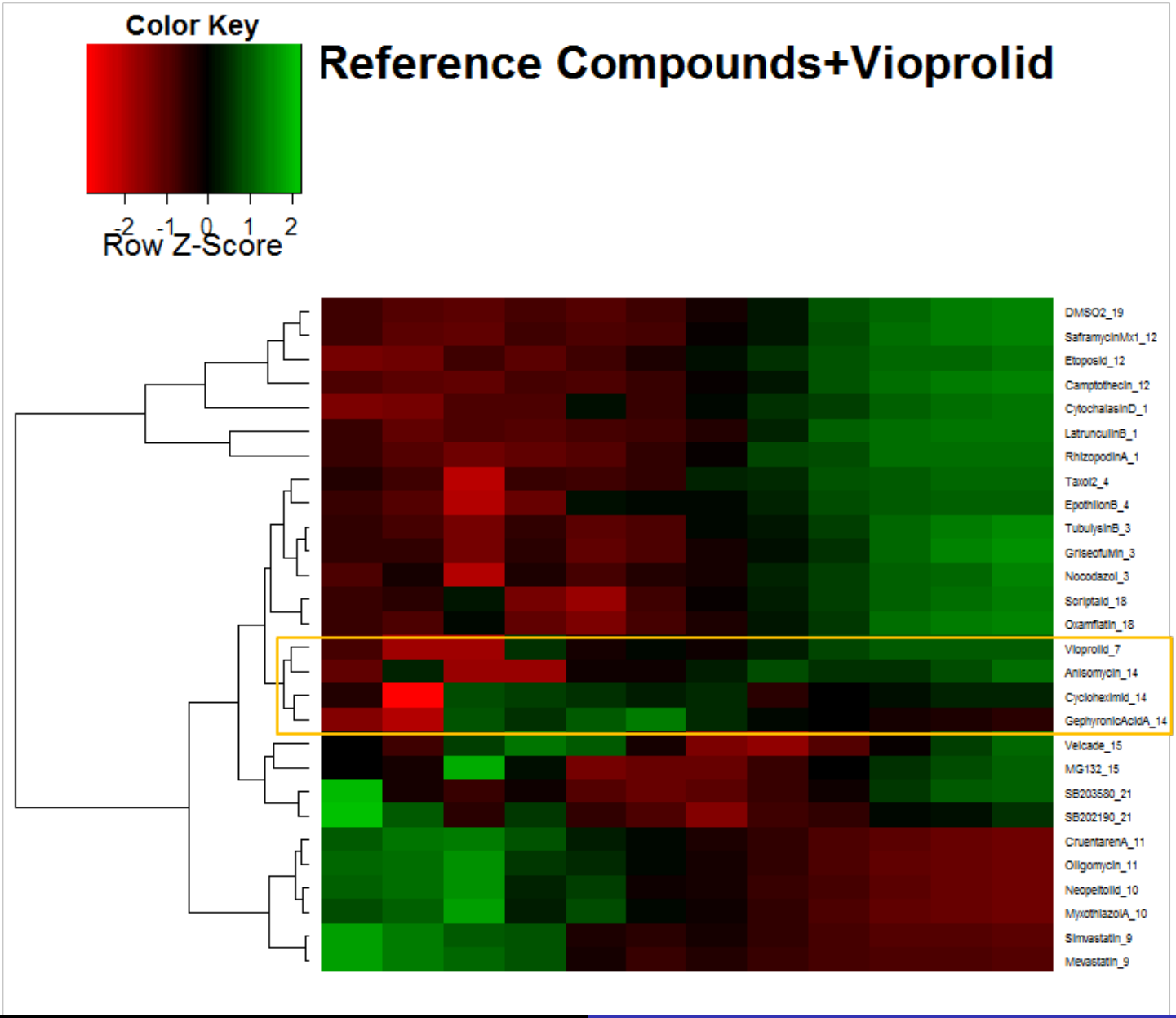




What is cluster analysis?



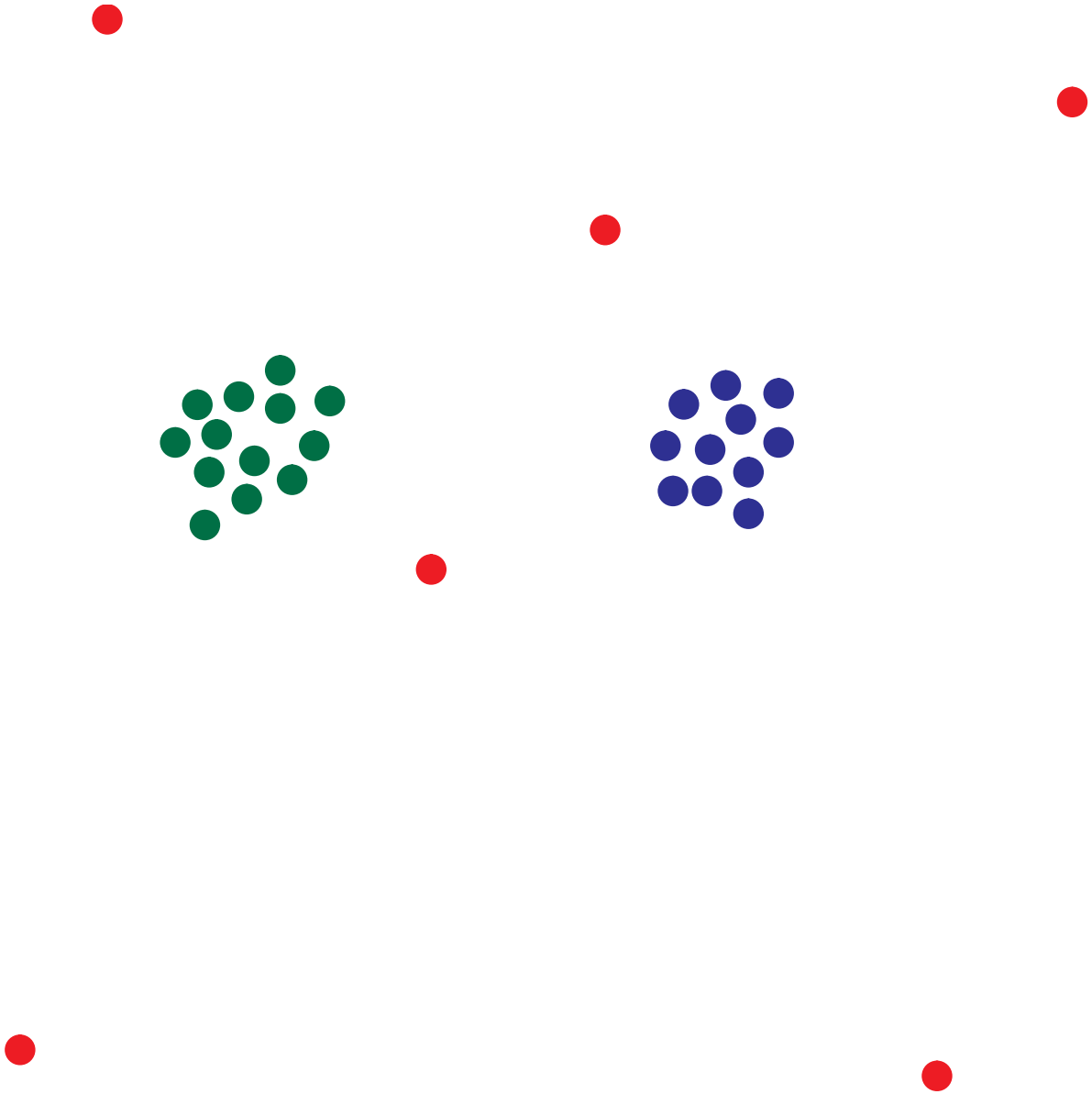
What is cluster analysis?



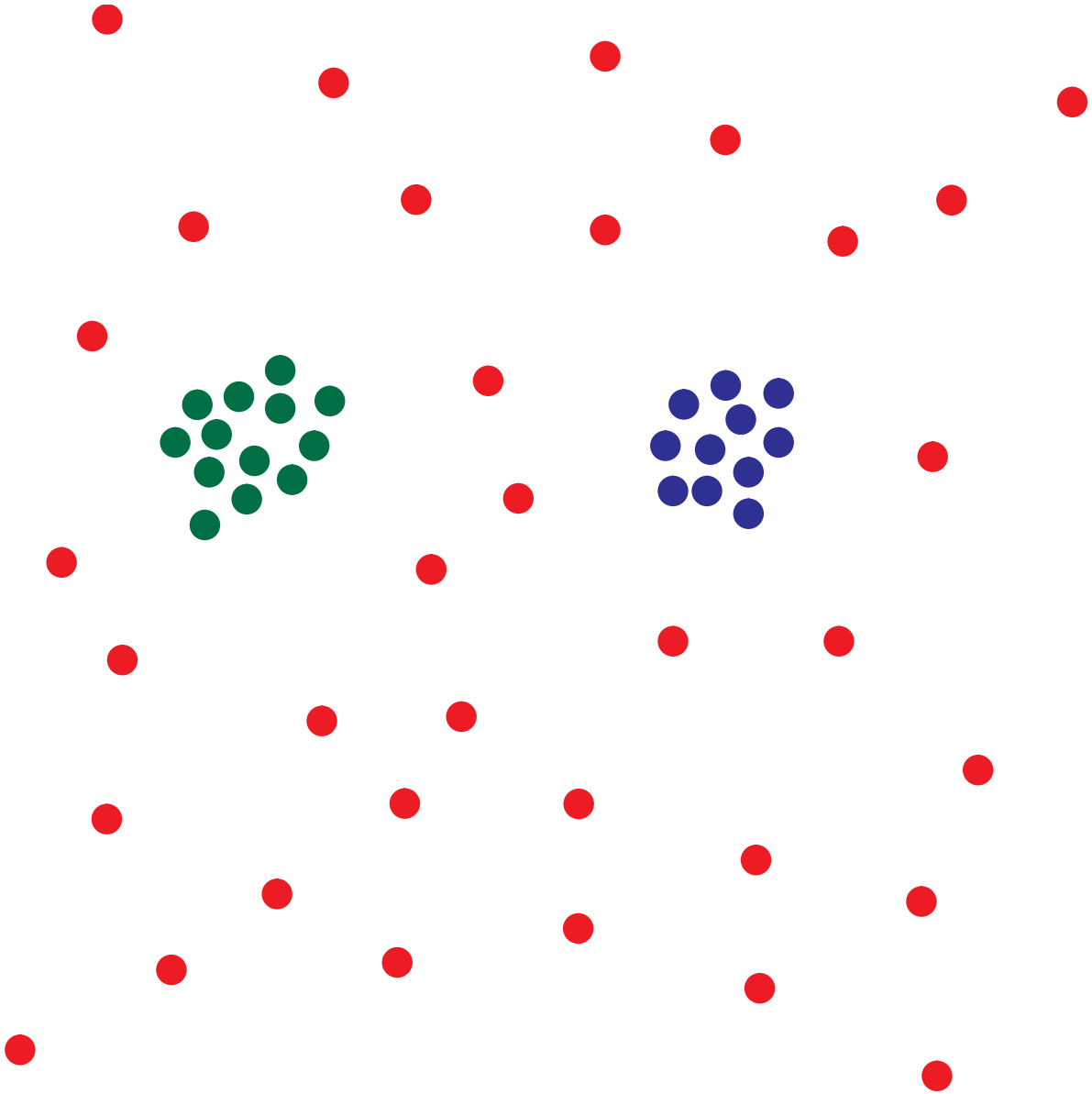
Goals of cluster analysis

- Partition a given data set into clusters
- Check whether related objects cluster together
- Classify unknown objects
- Find single "meaningful" clusters (and do not care about the rest of the data)

Clusters with noise



Noise with clusters



Hierarchical clustering

- Hierarchical clustering builds clusters step by step.
- Usually a bottom up strategy is applied by first considering each data object as a separate cluster and then step by step joining clusters together that are close to each other. This approach is called agglomerative hierarchical clustering.
- In contrast to agglomerative hierarchical clustering, divisive hierarchical clustering starts with the whole data set as a single cluster and then divides clusters step by step into smaller clusters.
- In order to decide which data objects should belong to the same cluster, a (dis-)similarity measure is needed.
- All that is needed for hierarchical clustering is an $n \times n$-matrix $[d_{i,j}]$, where $d_{i,j}$ is the dissimilarity of data objects i and j. (n is the number of data objects.)

Hierarchical clustering: Dissimilarity matrix

The dissimilarity matrix $[d_{i,j}]$ should at least satisfy the following conditions.

- $d_{i,j} \geq 0$, i.e. dissimilarity cannot be negative.
- $d_{i,i} = 0$, i.e. each data object is completely similar to itself.
- $d_{i,j} = d_{j,i}$, i.e. data object i is (dis-)similar to data object j to the same degree as data object j is (dis-)similar to data object i.

It is often useful if the dissimilarity is a (pseudo-)metric, satisfying also the

- triangle inequality $d_{i,k} \leq d_{i,j} + d_{j,k}$.

Agglomerative hierarchical clustering: Algorithm

Input: $n \times n$ dissimilarity matrix $[d_{i,j}]$.

1. Start with n clusters, each data objects forms a single cluster.
2. Reduce the number of clusters by joining those two clusters that are most similar (least dissimilar).
3. Repeat step 3 until there is only one cluster left containing all data objects.

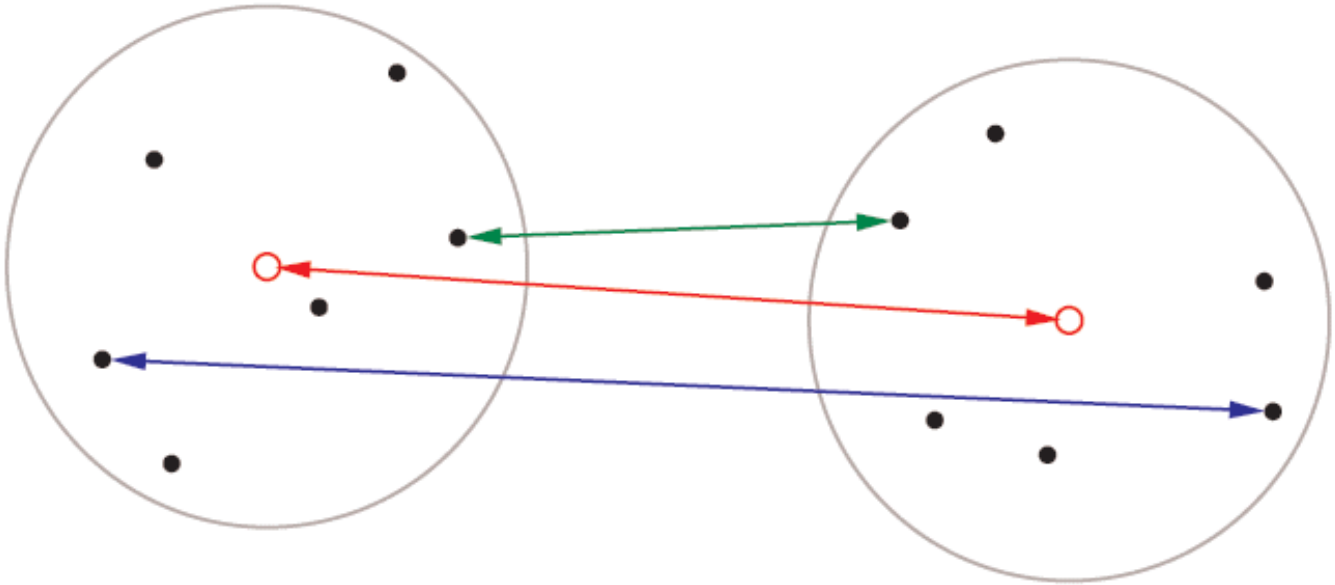
Measuring dissimilarity between clusters

- The dissimilarity between two clusters containing only one data objects each is simply the dissimilarity of the two data objects specified in the dissimilarity matrix $[d_{i,j}]$.
- How do we compute the dissimilarity between clusters that contain more than one data object?

Measuring dissimilarity between clusters

- **Centroid** (red)
  Distance between the centroids (mean value vectors) of the two clusters. (Requires that the data objects are numerical vectors and that the distance matrix is based on the Euclidean distance.)
- **Average Linkage**
  Average dissimilarity between two points of the two clusters.
- **Single Linkage** (green)
  Dissimilarity between the two most similar data objects of the two clusters.
- **Complete Linkage** (blue)
  Dissimilarity between the two most dissimilar data objects of the two clusters.

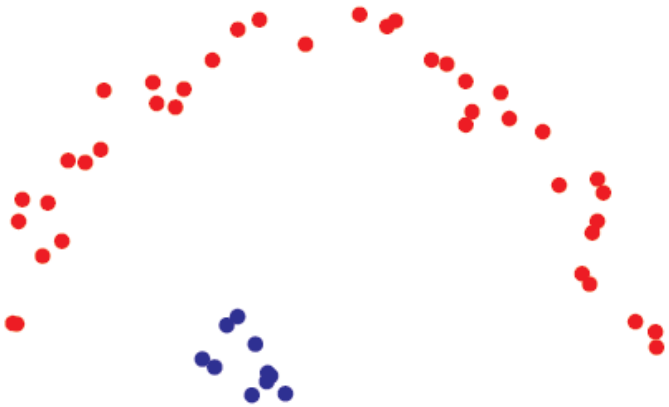
Measuring dissimilarity between clusters



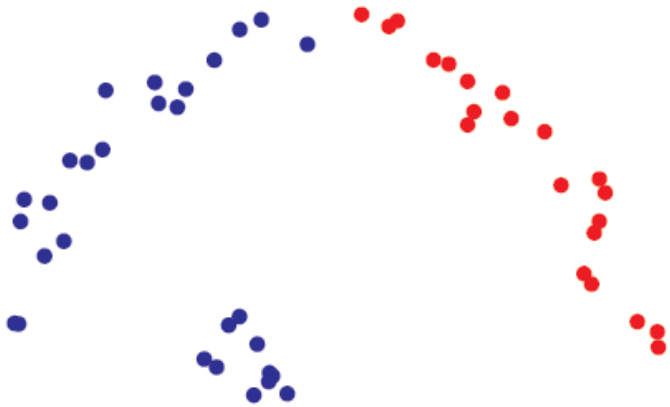
Measuring dissimilarity between clusters

- Single linkage can "follow chains" in the data (may be desirable in certain applications).
- Complete linkage leads to very compact clusters.
- Average linkage also tends clearly towards compact clusters.

Measuring dissimilarity between clusters



Single linkage



Complete linkage

Measuring dissimilarity between clusters

Ward's method is another strategy for merging clusters.

In contrast to single, complete or average linkage, it takes the number of data objects in each cluster into account.

Measuring dissimilarity between clusters

The updated dissimilarity between the newly formed cluster $\{\mathcal{C} \cup \mathcal{C}'\}$ and the cluster $\mathcal{C}''$ is computed in the follwing way.

$$d'(\{\mathcal{C} \cup \mathcal{C}'\}, \mathcal{C}'') = \ldots$$

$$\begin{aligned}
\text{single linkage} &= \min\{d'(\mathcal{C}, \mathcal{C}''), d'(\mathcal{C}', \mathcal{C}'')\} \\
\text{complete linkage} &= \max\{d'(\mathcal{C}, \mathcal{C}''), d'(\mathcal{C}', \mathcal{C}'')\} \\
\text{average linkage} &= \frac{|\mathcal{C}| d'(\mathcal{C}, \mathcal{C}'') + |\mathcal{C}'| d'(\mathcal{C}', \mathcal{C}'')}{|\mathcal{C}| + |\mathcal{C}'|} \\
\text{Ward} &= \frac{(|\mathcal{C}| + |\mathcal{C}''|) d'(\mathcal{C}, \mathcal{C}'') + (|\mathcal{C}'| + |\mathcal{C}''|) d'(\mathcal{C}', \mathcal{C}'') - |\mathcal{C}''| d'(\mathcal{C}, \mathcal{C}')}{|\mathcal{C}| + |\mathcal{C}'| + |\mathcal{C}''|} \\
\text{centroid (metric)} &= \frac{1}{|\mathcal{C} \cup \mathcal{C}'||\mathcal{C}''|} \sum_{\mathbf{x} \in \mathcal{C} \cup \mathcal{C}'} \sum_{\mathbf{y} \in \mathcal{C}''} d(\mathbf{x}, \mathbf{y})
\end{aligned}$$

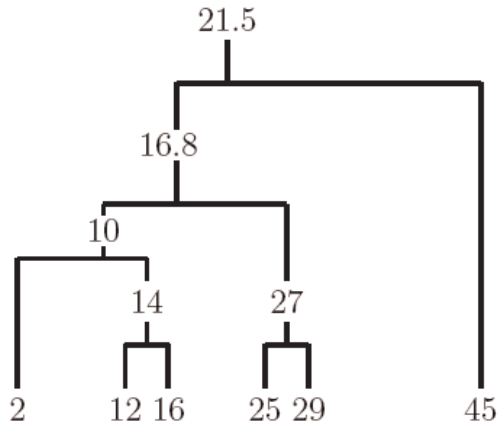
Dendrograms

- The cluster merging process arranges the data points in a binary tree.
- Draw the data tuples at the bottom or on the left (equally spaced if they are multi-dimensional).
- Draw a connection between clusters that are merged, with the distance to the data points representing the distance between the clusters.

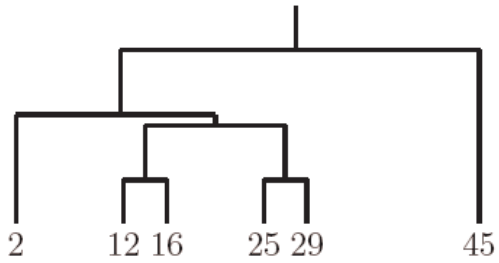
Hierarchical clustering

- Example: Clustering of the 1-dimensional data set $\{2, 12, 16, 25, 29, 45\}$.
- All three approaches to measure the distance between clusters lead to different dendrograms.

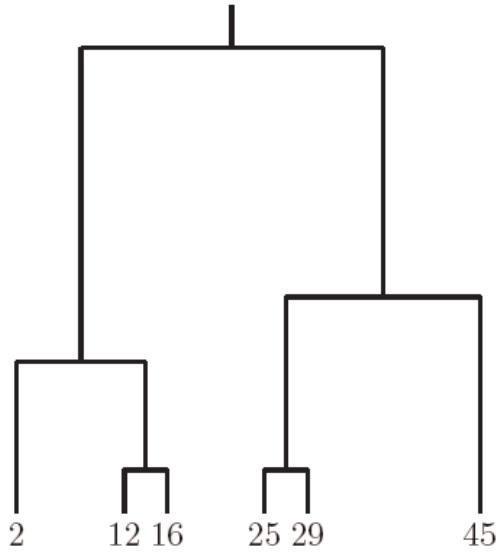
Hierarchical clustering



Centroid

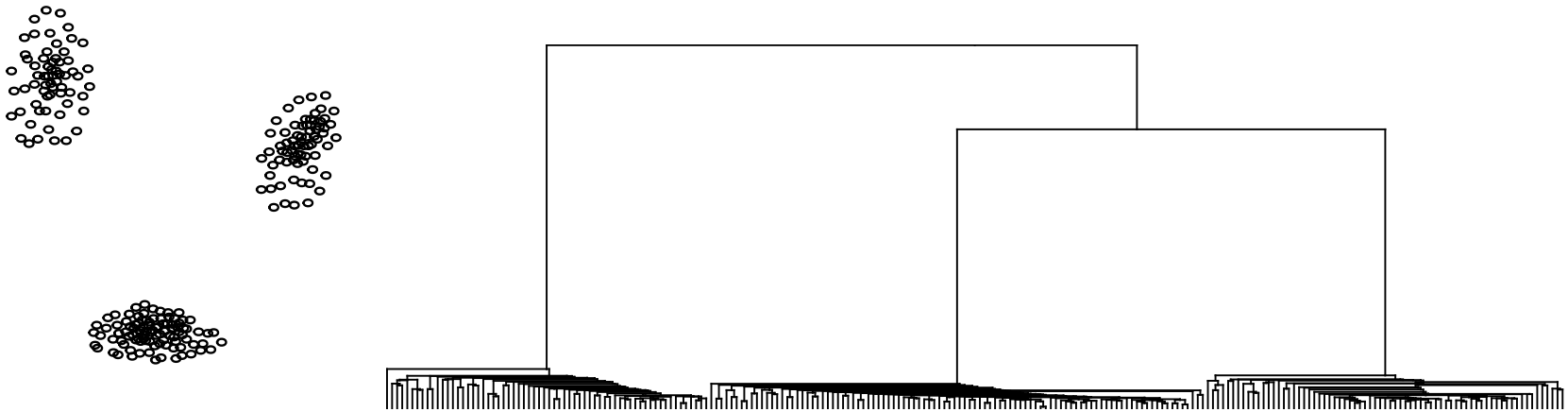


Single linkage

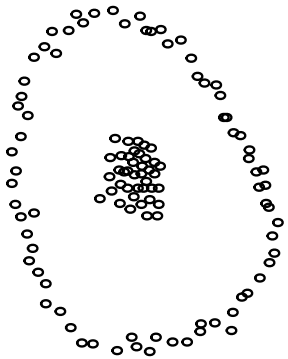


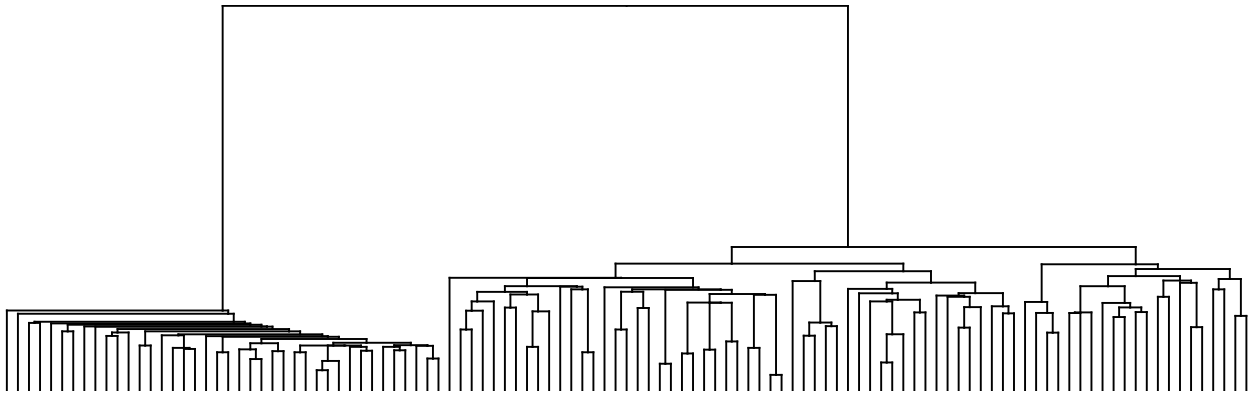
Complete linkage

Dendrograms

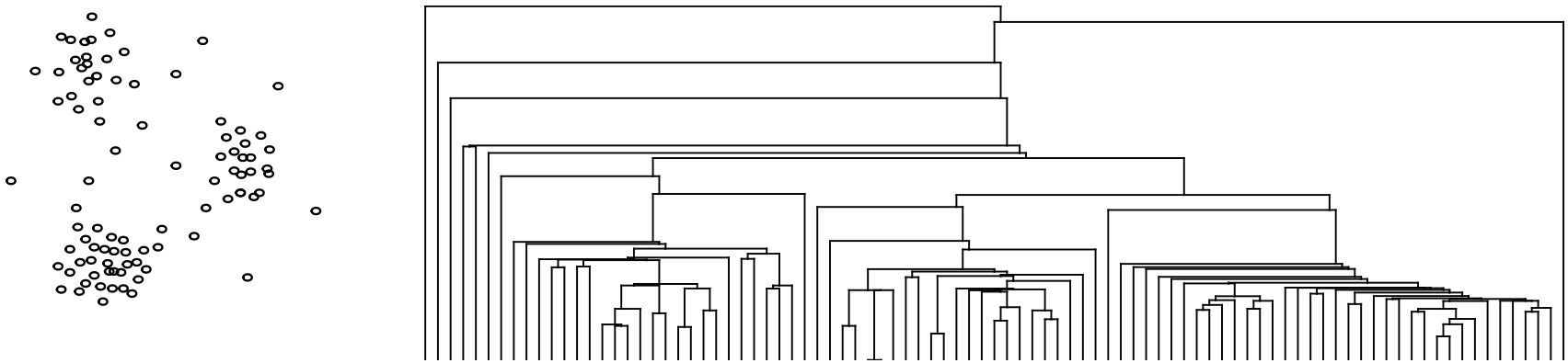


Dendrograms





Dendrograms



Choosing the clusters

- **Simplest Approach:**
  - Specify a minimum desired distance between clusters.
  - Stop merging clusters if the closest two clusters are farther apart than this distance.
- **Visual Approach:**
  - Merge clusters until all data points are combined into one cluster.
  - Draw the dendrogram and find a good cut level.
  - Advantage: Cut need not be strictly horizontal.

Choosing the clusters

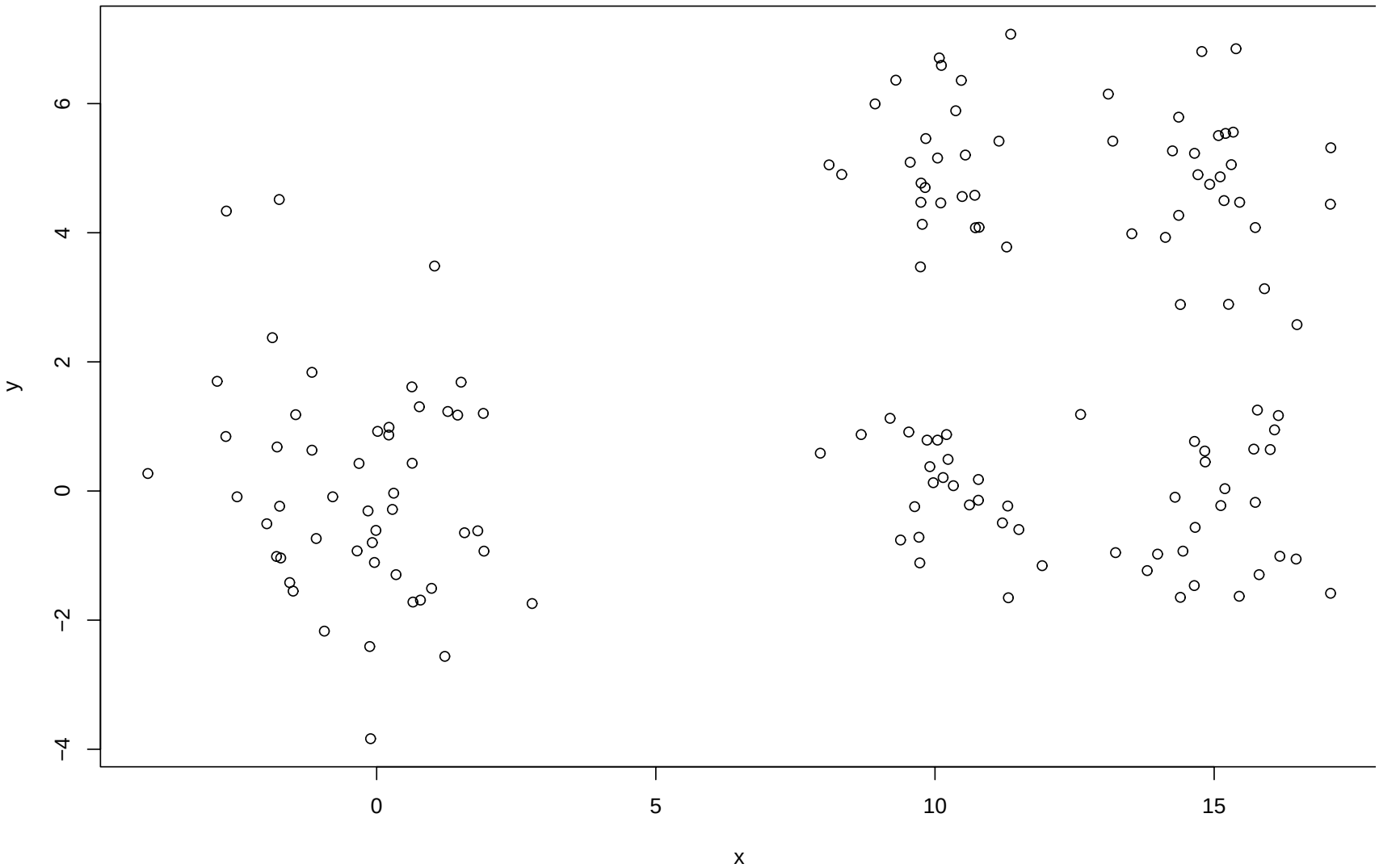
- **More Sophisticated Approaches:**
  - Analyze the sequence of distances in the merging process.
  - Try to find a step in which the distance between the two clusters merged is considerably larger than the distance of the previous step.
  - Several heuristic criteria exist for this step selection.

Heatmaps

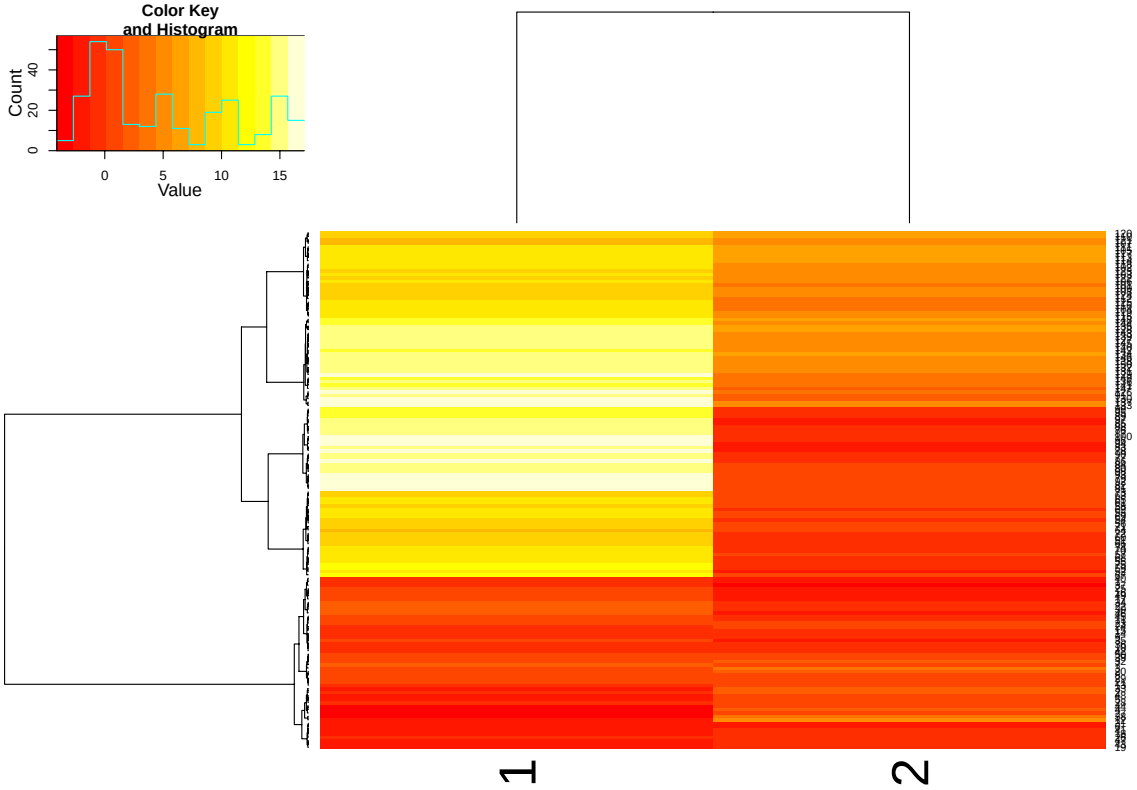
A heatmap combines

- a dendrogram resulting from clustering the data,
- a dendrogram resulting from clustering the attributes and
- colours to indicate the values of the attributes.

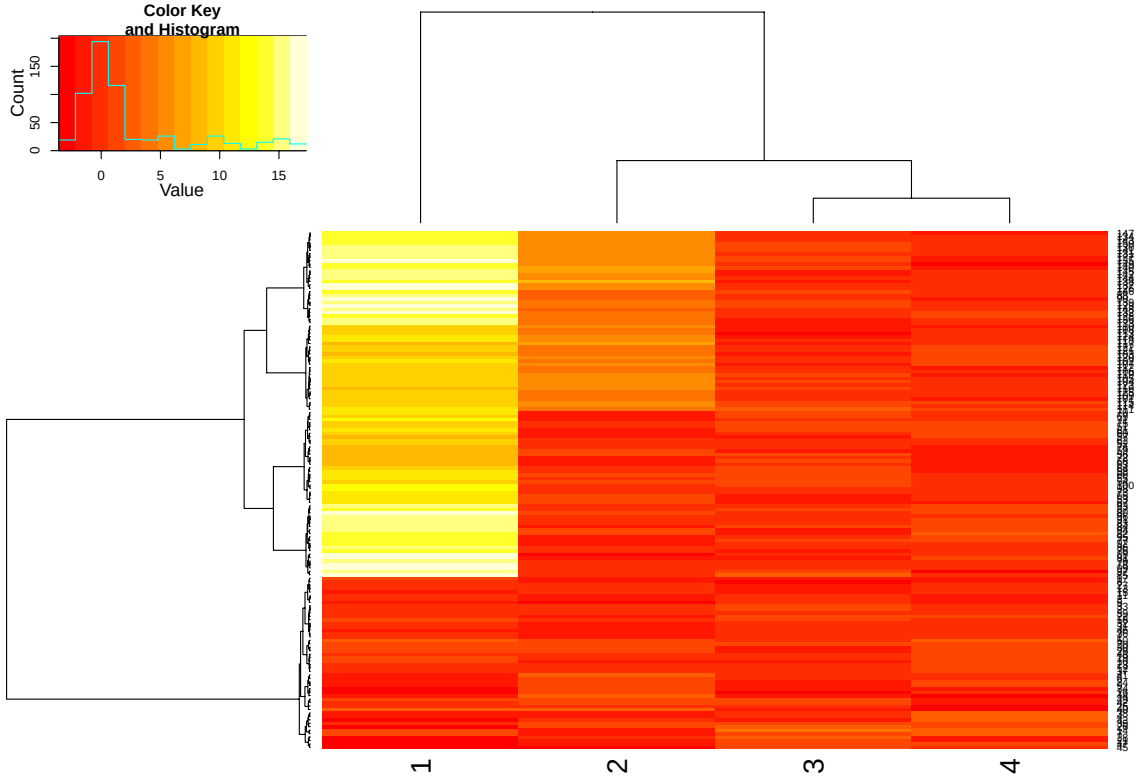
Hierarchical clustering



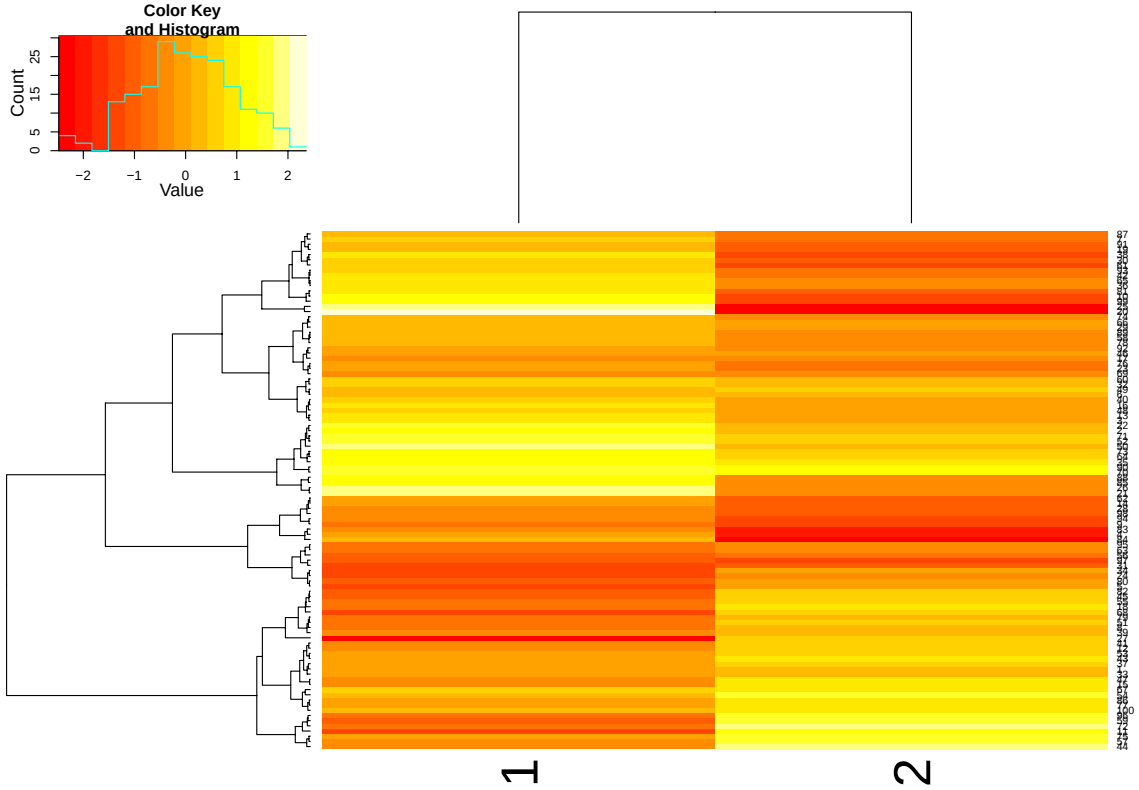
Heatmap and dendrogram



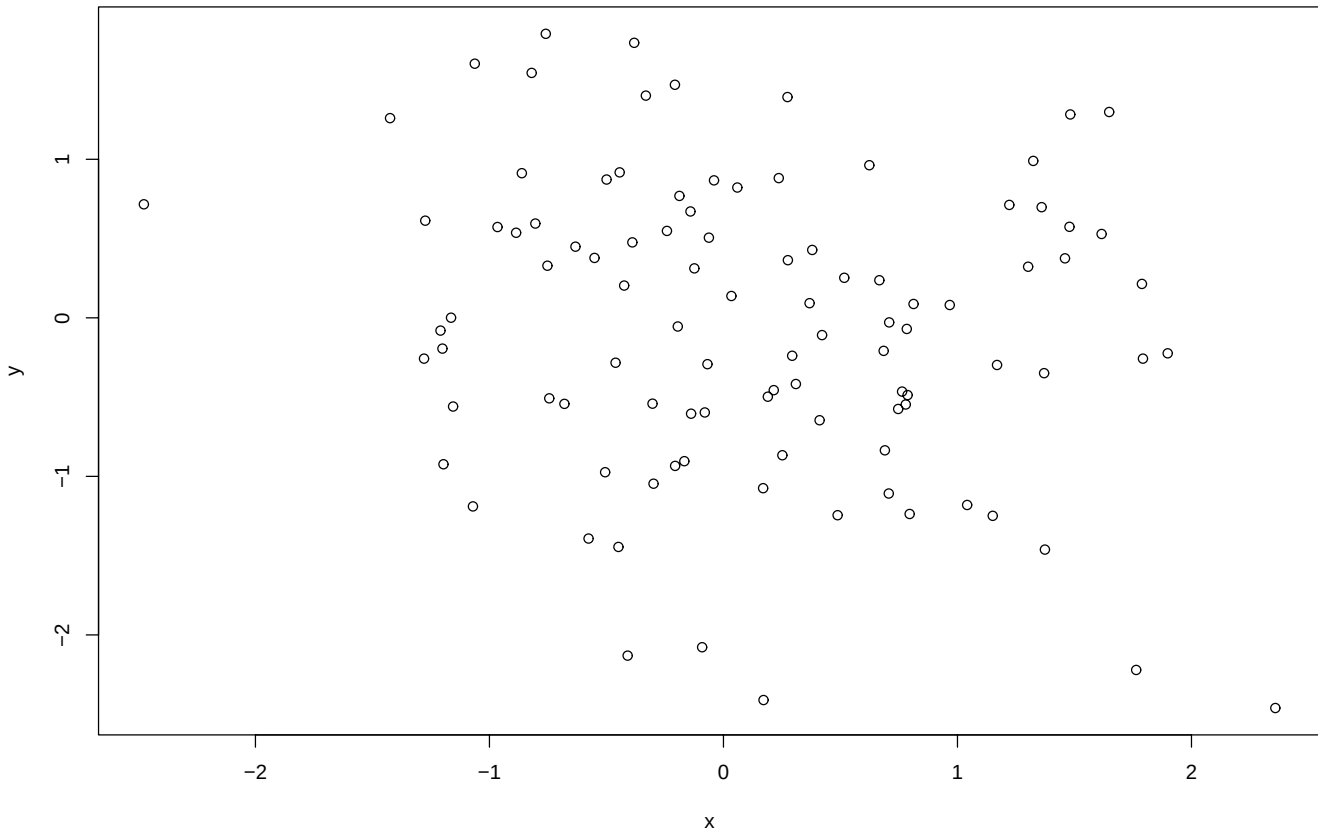
Heatmap and dendrogram



Heatmap and dendrogram



Hierarchical clustering



Divisive hierarchical clustering

The top-down approach of divisive hierarchical clustering is seldom used.

- In agglomerative clustering the minimum of the pairwise dissimilarities has to be determined, leading to a quadratic complexity in each step (quadratic in the number of clusters still present in the corresponding step).
- In divisive clustering for each cluster all possible splits would have to be considered.
- In the first step, there are $2^{n-1} - 1$ possible splits, where n is the number of data objects.

k-Means clustering

- Choose a number k of clusters to be found (user input).
- Initialize the cluster centres randomly
  (for instance, by randomly selecting k data points).
- **Data point assignment**:
  Assign each data point to the cluster centre that is closest to it (i.e. closer than any other cluster centre).
- **Cluster centre update**:
  Compute new cluster centres as the mean vectors of the assigned data points. (Intuitively: centre of gravity if each data point has unit weight.)

k-Means clustering

- Repeat these two steps (data point assignment and cluster centre update) until the clusters centres do not change anymore.
- It can be shown that this scheme must converge, i.e., the update of the cluster centres cannot go on forever.

k-Means clustering

Aim: Minimize the objective function

$$f = \sum_{i=1}^{k} \sum_{j=1}^{n} u_{ij} d_{ij}$$

under the constraints $u_{ij} \in \{0, 1\}$ and

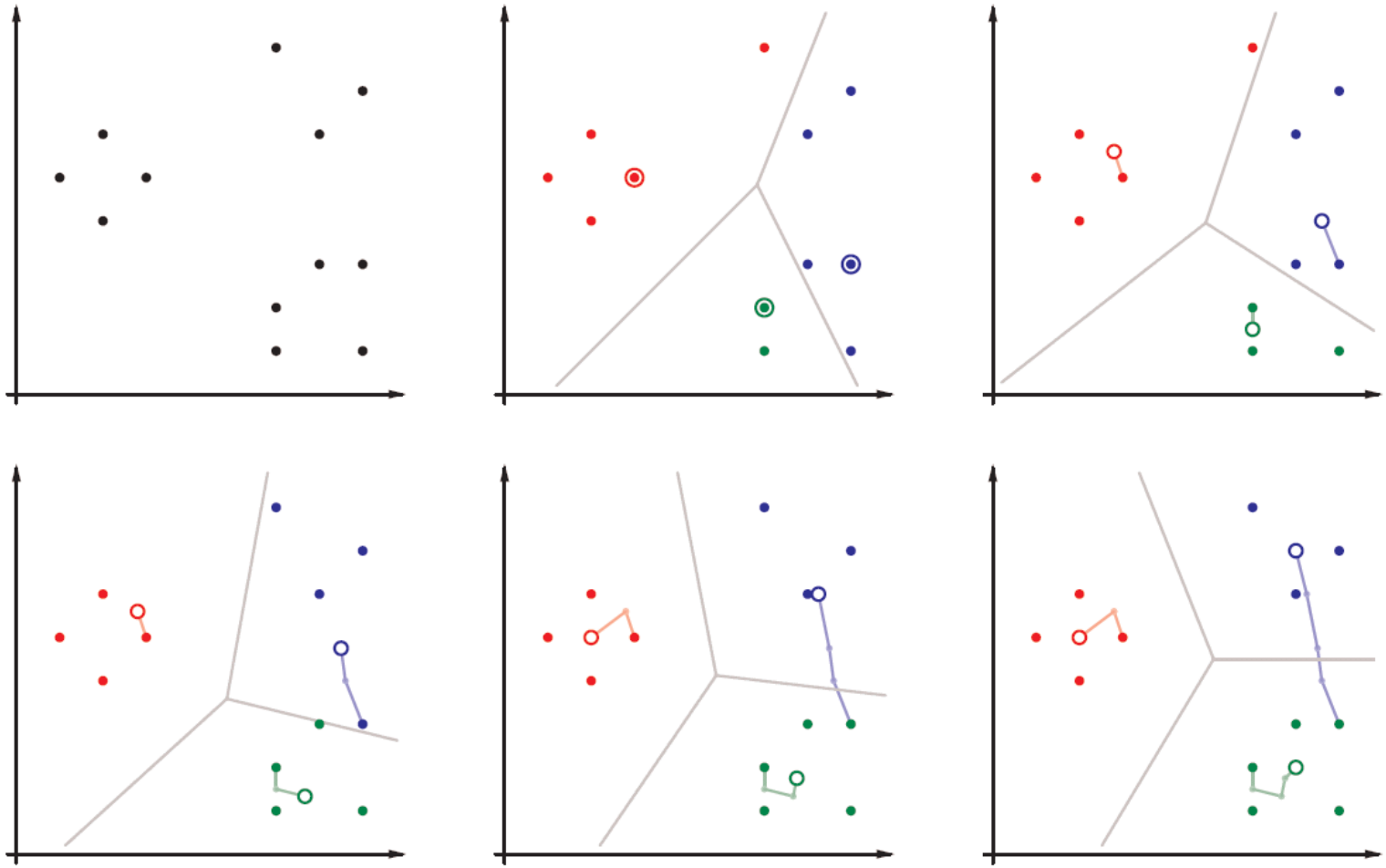
$$\sum_{i=1}^{k} u_{ij} = 1 \qquad \text{for all } j = 1, \ldots, n.$$

Alternating optimization

- Assuming the cluster centres to be fixed, $u_{ij} = 1$ should be chosen for the cluster i to which data object x_j has the smallest distance in order to minimize the objective function.
- Assuming the assignments to the clusters to be fixed, each cluster centre should be chosen as the mean vector of the data objects assigned to the cluster in order to minimize the objective function.

This is a greedy algorithm.

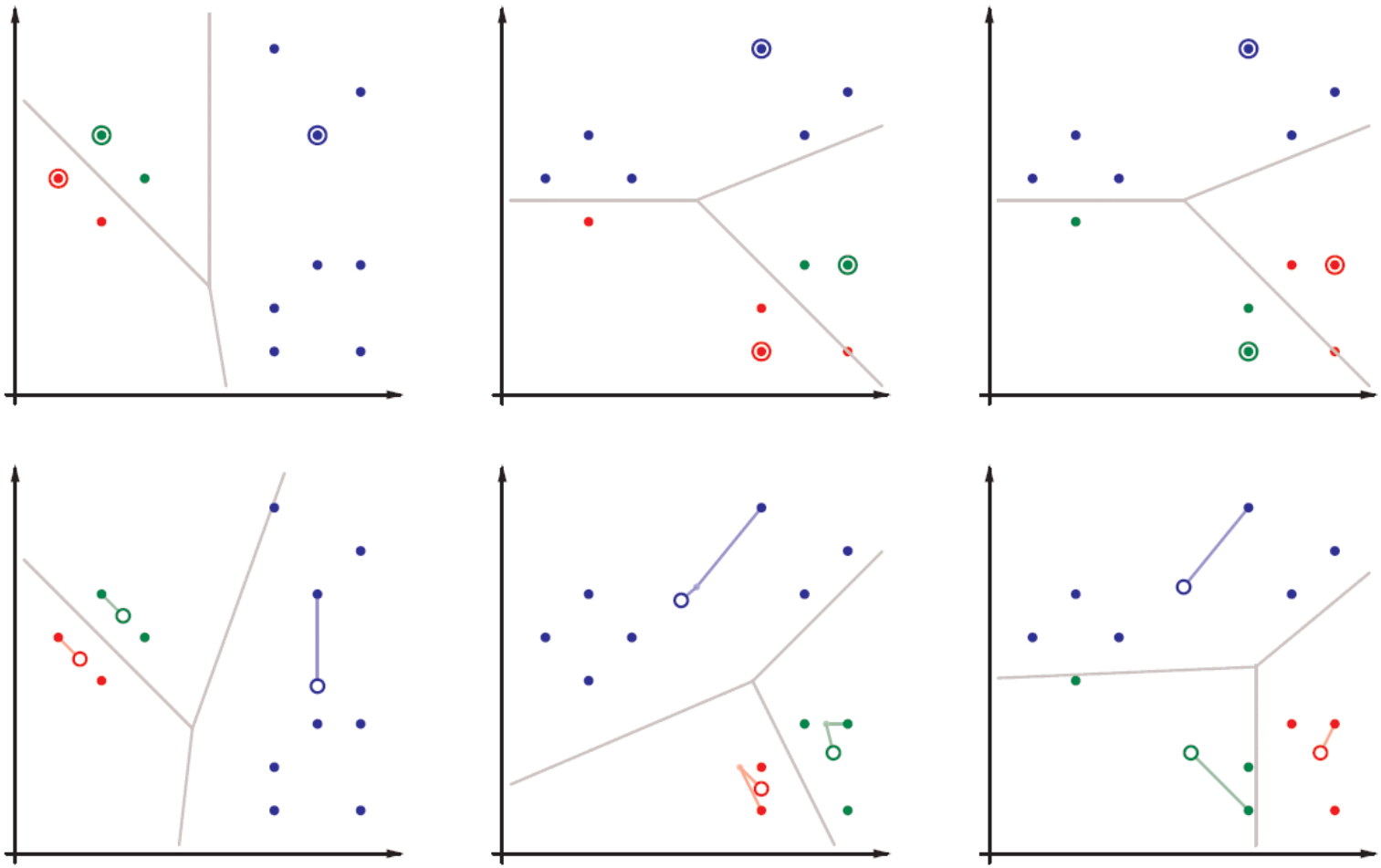
k-Means clustering: Example



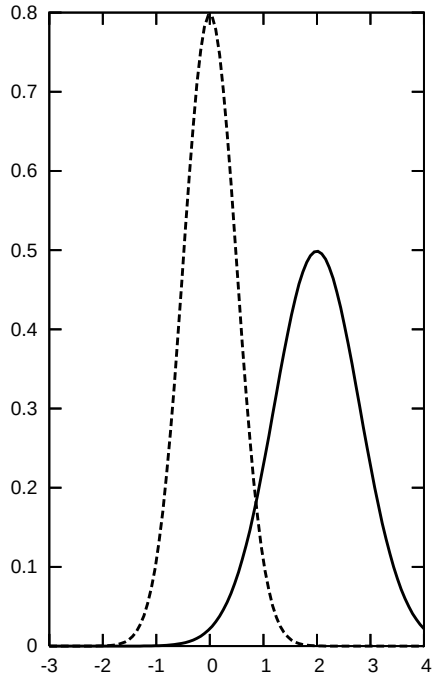
k-Means clustering: Local minima

- Clustering is successful in this example:
  The clusters found are those that would have been formed intuitively.
- Convergence is achieved after only 5 steps.
  (This is typical: convergence is usually very fast.)
- However: The clustering result is fairly **sensitive to the initial positions** of the cluster centres.
- With a bad initialisation clustering may fail
  (the alternating update process gets stuck in a local minimum).

k-Means clustering: Local minima

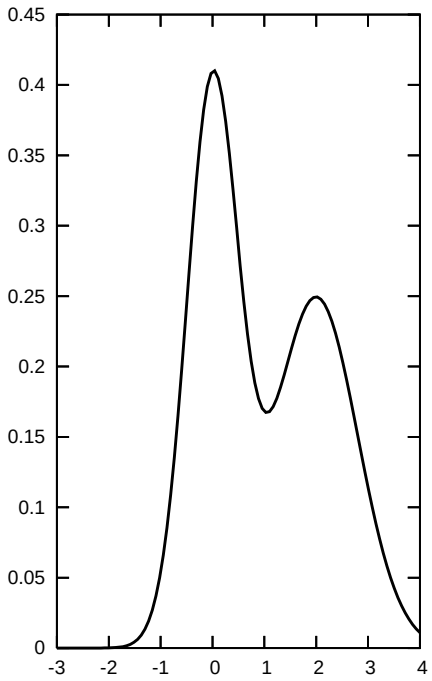


Gaussian mixture models



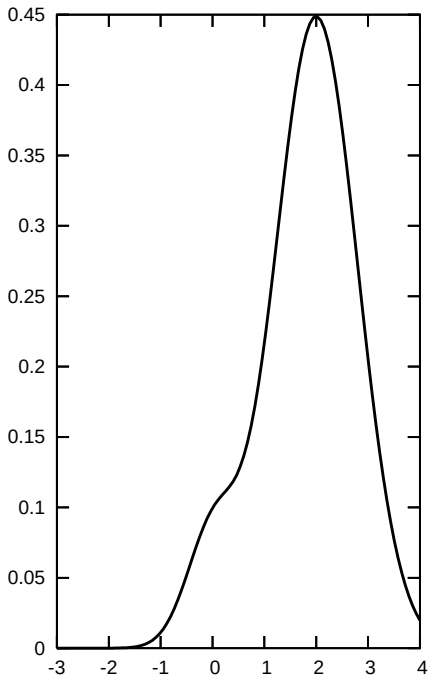
Two normal distributions

Gaussian mixture models



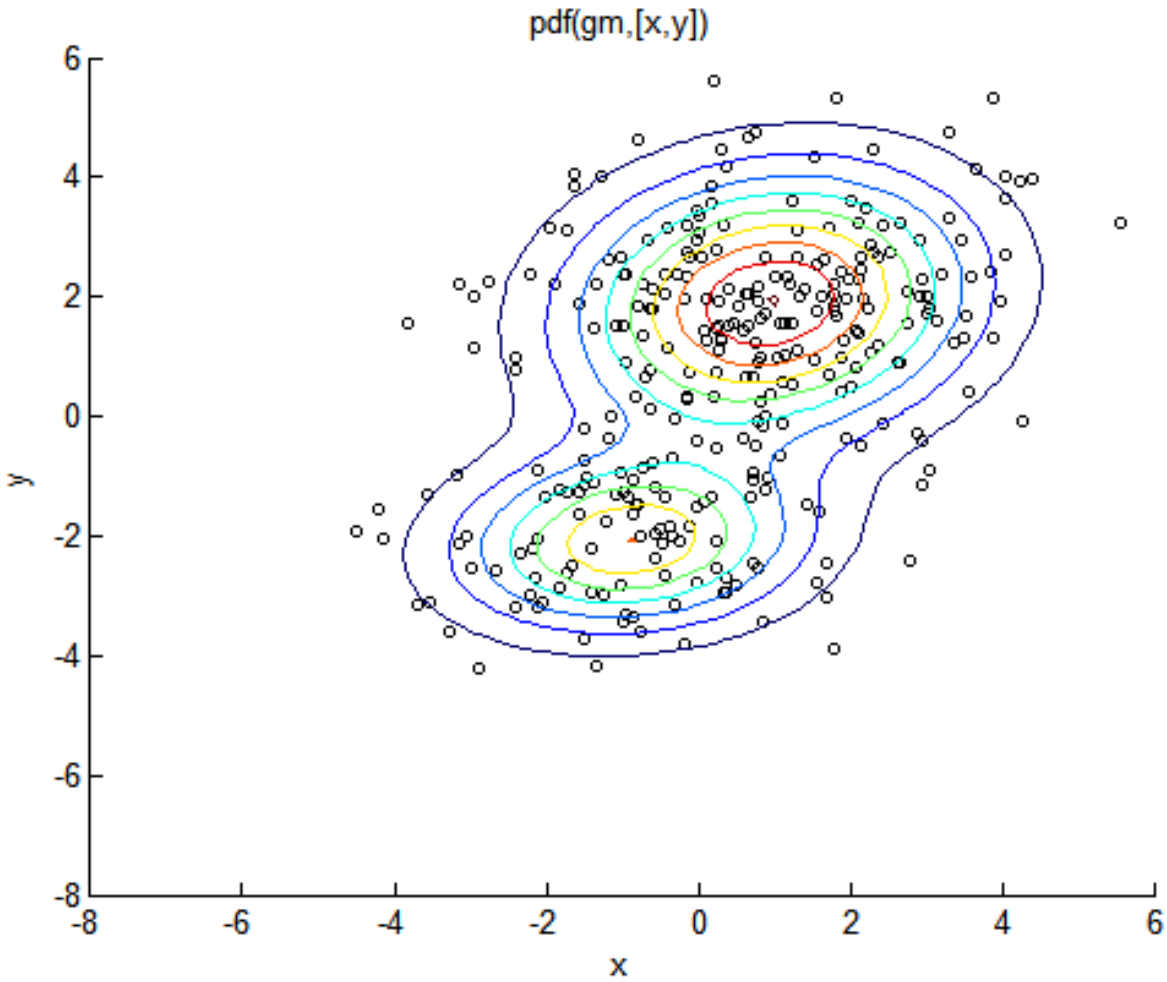
Mixture model (both normal distrubutions contribute 50%)

Gaussian mixture models



Mixture model (one normal distrubutions contributes 10%, the other 90%)

Gaussian mixture models



Gaussian mixture models – EM clustering

- **Assumption:** Data were generated by sampling a set of normal distributions.
  (The probability density is a mixture of normal distributions.)
- **Aim:** Find the parameters for the normal distributions and how much each normal distribution contributes to the data.
- **Algorithm:** EM clustering (expectation maximisation).
  Alternating scheme in which the parameters of the normal distributions and the likelihoods of the data points to be generated by the corresponding normal distributions are estimated.

Density-based clustering

For numerical data, density-based clustering algorithm often yield the best results.

Principle: A connected region with high data density corresponds to one cluster.

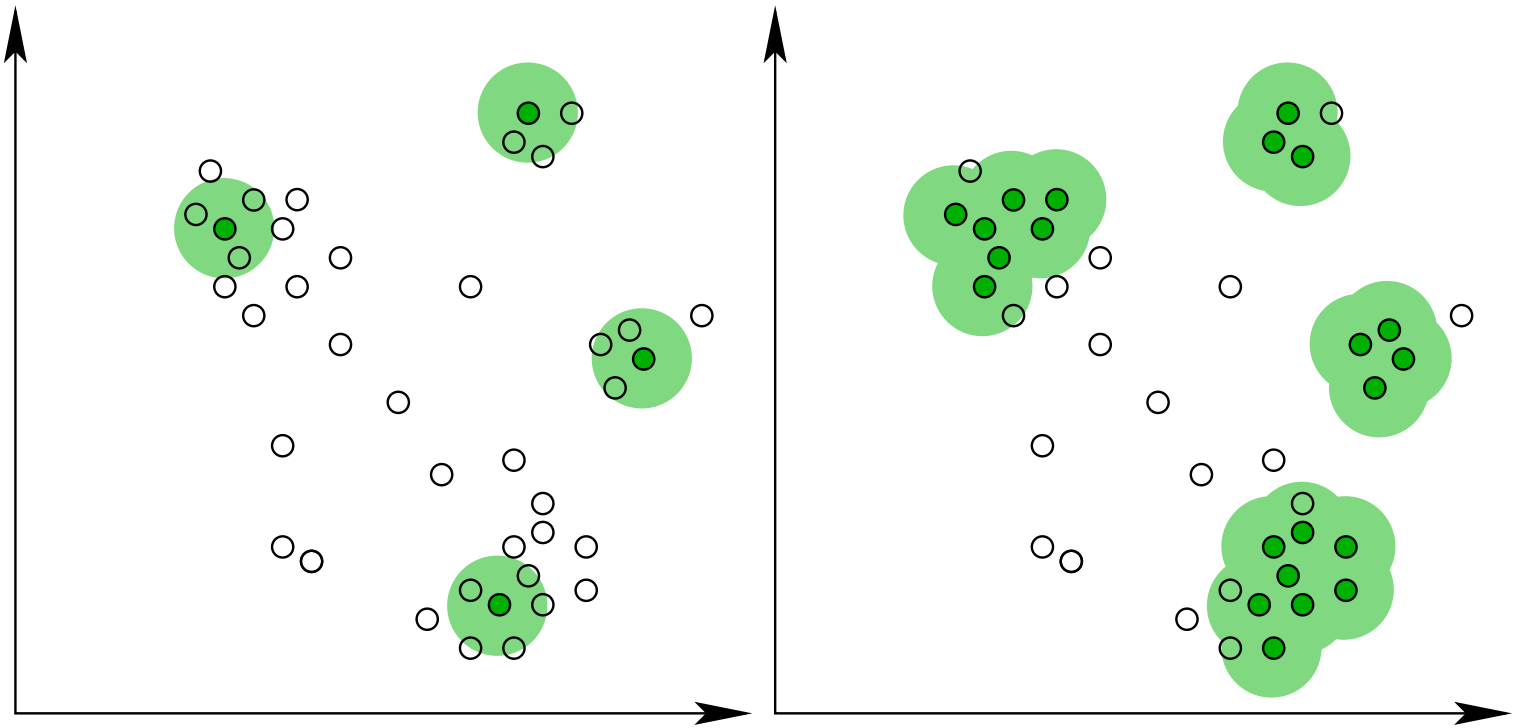
DBScan is one of the density-based clustering algorithms.

Density-based clustering: DBScan

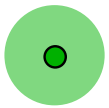
Principle idea of DBScan:

1. Find a data point where the data density is high, i.e. in whose ε-neighbourhood are at least ℓ other points. (ε and ℓ are parameters of the algorithm to be chosen by the user.)
2. All the points in the ε-neighbourhood are considered to belong to one cluster.
3. Expand this ε-neighbourhood (the cluster) as long as the high density criterion is satisfied.
4. Remove the cluster (all data points assigned to the cluster) from the data set and continue with 1. as long as data points with a high data density around them can be found.

Density-based clustering: DBScan

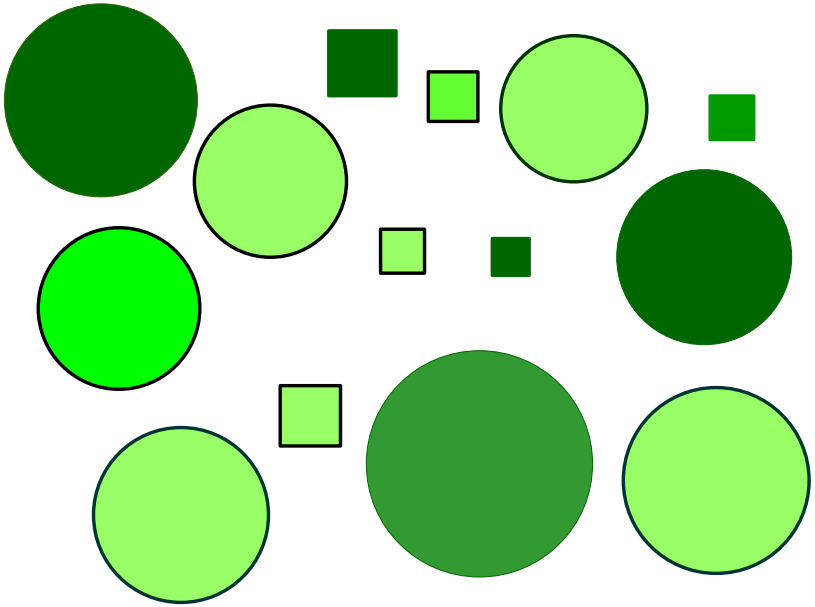


neighbourhood cell with at least 3 hits

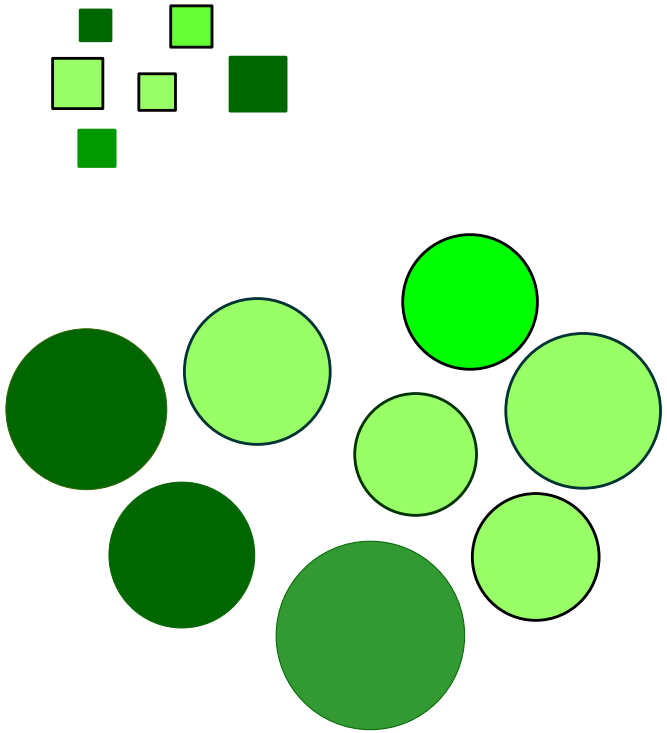


neighbourhood cell with at least 3 hits

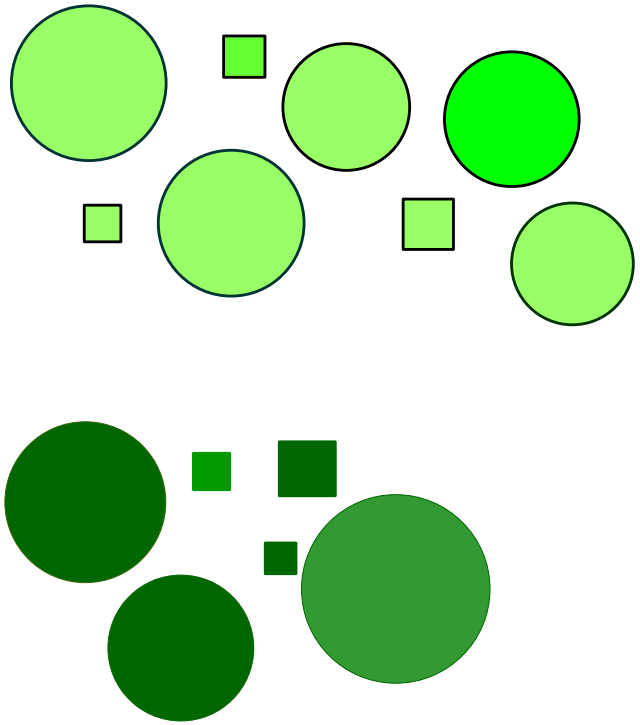
How to cluster these objects?



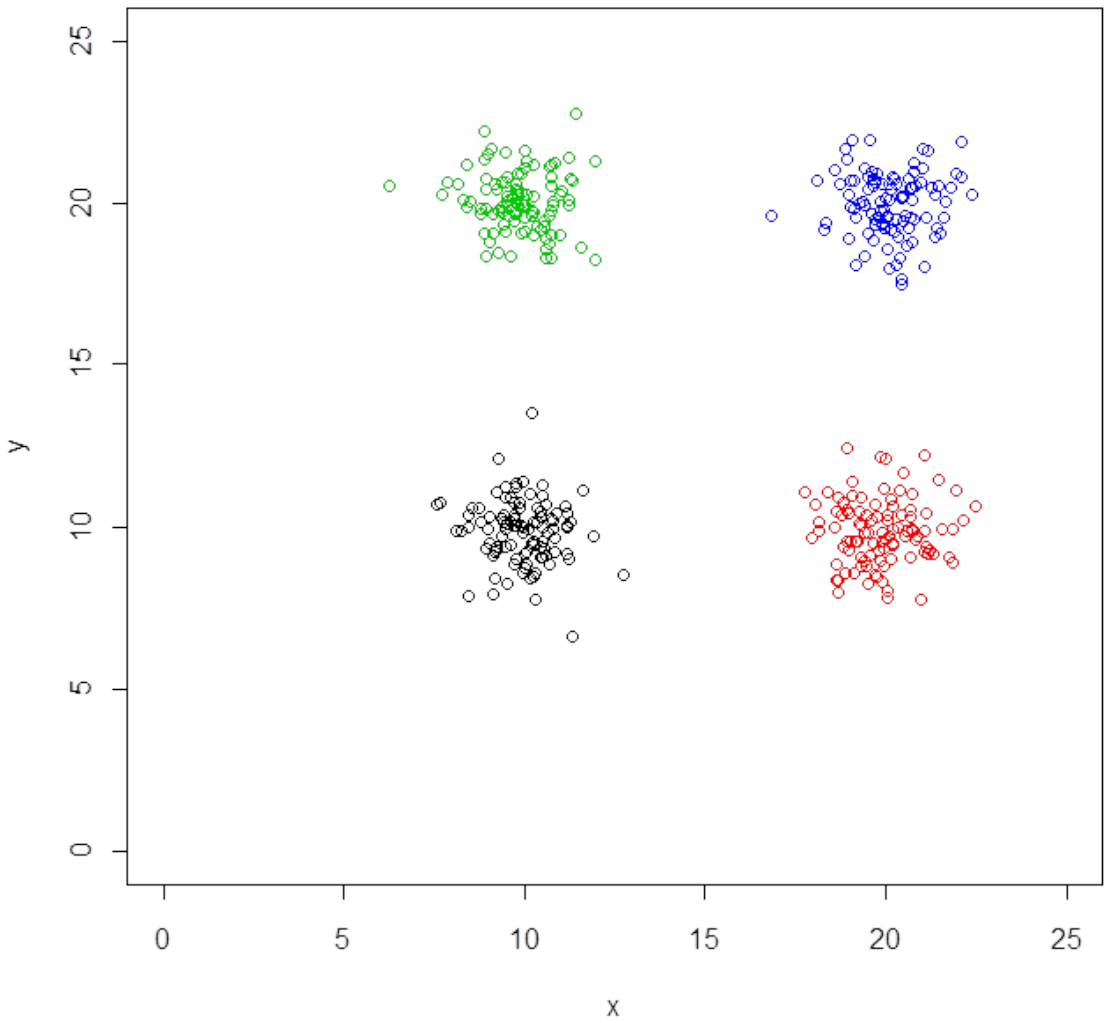
How to cluster these objects?



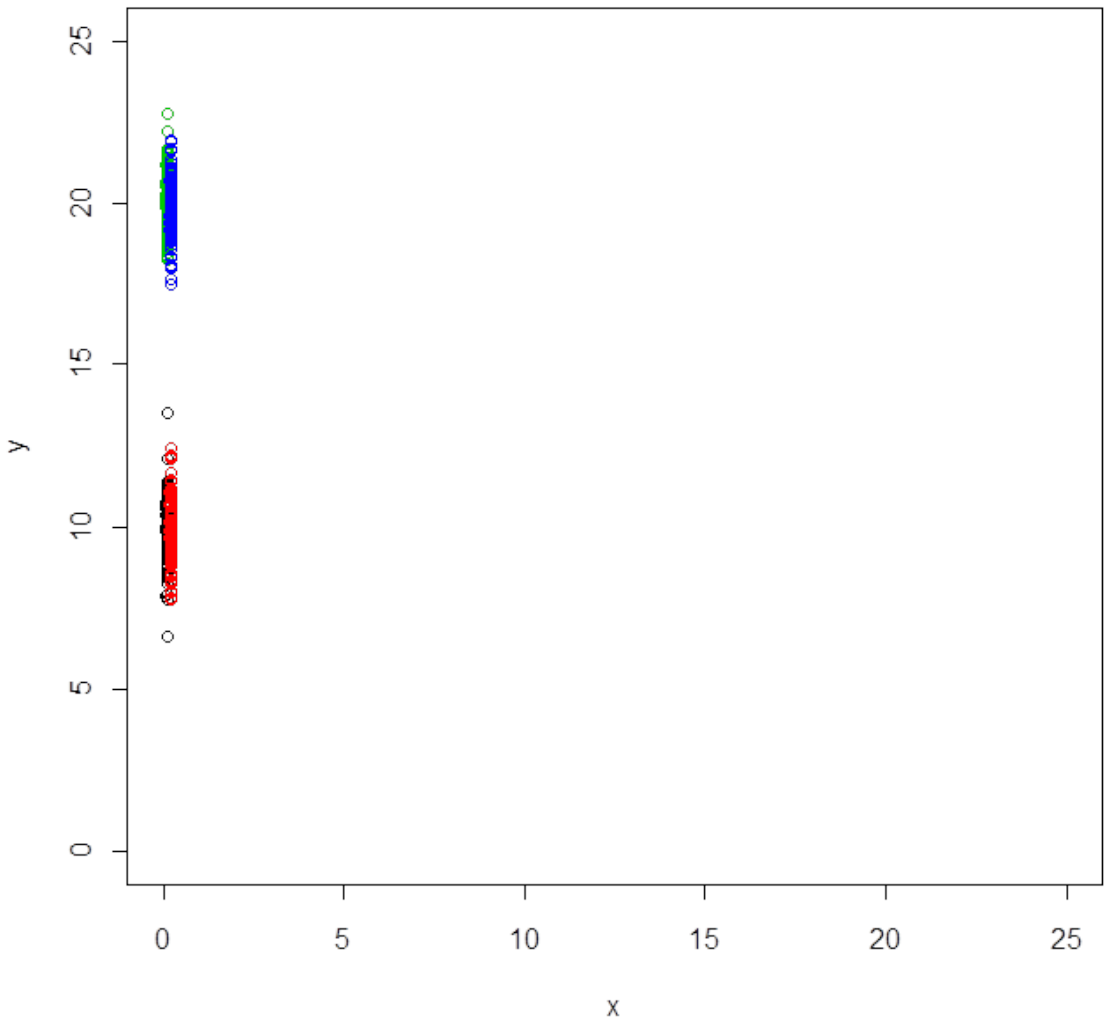
How to cluster these objects?



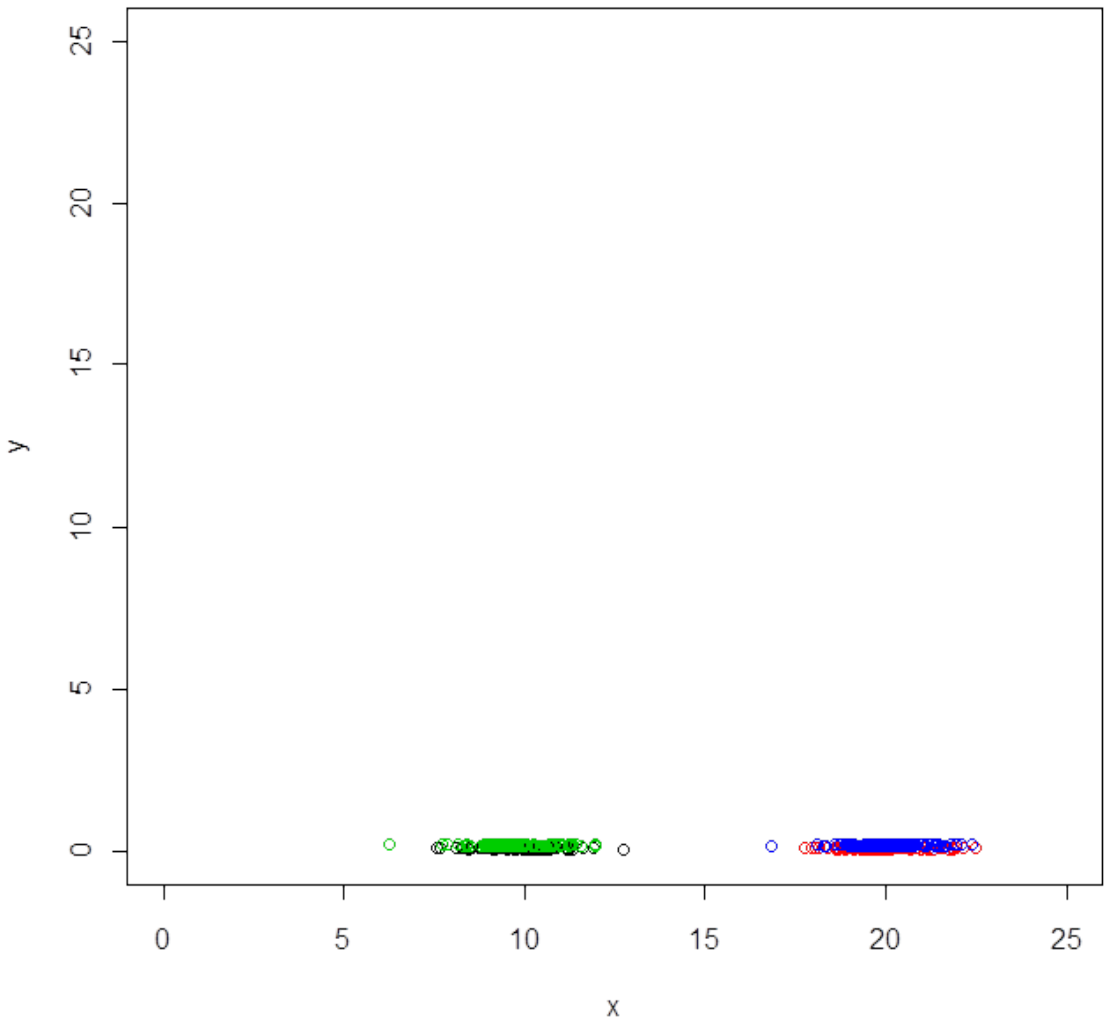
Clustering example



Clustering example



Clustering example



Scaling

The previous three slides show the same data set.

- In the second slide, the unit on the x-axis was changed to milli-units.
- In the third slide, the unit on the y-axis was changed to milli-units.

Clusters should not depend on the measurement unit!

Therefore, some kind of normalisation should be carried out before clustering.

Normalisation/Standardisation

min-max normalization. For a numerical attribute X with $\min_X$ and $\max_X$ being the minimum and maximum value in the sample, the min-max normalization is defined as

$$n : \mathrm{dom}X \to [0,1], \qquad x \mapsto \frac{x - \min_X}{\max_X - \min_X}$$

z-score standardization. For a numerical attribute X with sample mean $\hat{\mu}_X$ and empirical standard deviation $\hat{\sigma}_X$, the z-score standardization is defined as

$$s : \mathrm{dom}X \to \mathbb{R}, \qquad x \mapsto \frac{x - \hat{\mu}_X}{\hat{\sigma}_X}$$

robust z-score standardization. The sample mean and empirical standard deviation are easily affected by outliers. A more robust alternative is (see also boxplots):

$$s : \mathrm{dom}X \to \mathbb{R}, \qquad x \mapsto \frac{x - \tilde{x}}{IQR_X}$$

Normalisation/Standardisation

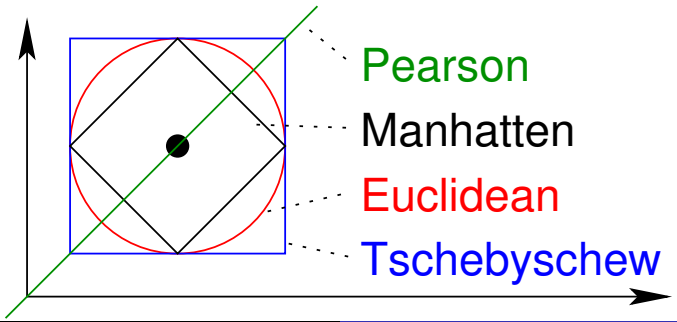
decimal scaling. For a numerical attribute X and the smallest integer value s larger than $\log_{10}(\max_X)$, the decimal scaling is defined as

$$d : \mathrm{dom}X \to [0,1], \qquad x \mapsto \frac{x}{10^s}$$

Notion of (dis-)similarity: Numerical attributes

There are various ways to measure the dissimilarity between two numerical vectors.

Minkowksi	L_p	$d_p(x,y) = \sqrt[p]{\sum_{i=1}^{n} \lvert x_i - y_i \rvert^p}$
Euclidean	L_2	$d_E(x,y) = \sqrt{(x_1 - y_1)^2 + \ldots + (x_n - y_n)^2}$
Manhattan	L_1	$d_M(x,y) = \lvert x_1 - y_1 \rvert + \ldots + \lvert x_n - y_n \rvert$
Tschebyschew	L_∞	$d_\infty(x,y) = \max\{\lvert x_1 - y_1 \rvert, \ldots, \lvert x_n - y_n \rvert\}$
Cosine		$d_C(x,y) = 1 - \frac{x^\top y}{\lVert x \rVert \lVert y \rVert}$
Tanimoto		$d_T(x,y) = \frac{x^\top y}{\lVert x \rVert^2 + \lVert y \rVert^2 - x^\top y}$
Pearson		Euclidean of z-score transformed **x**, **y**



Notion of (dis-)similarity: Binary attributes

The two values (e.g. 0 and 1) of a binary attribute can be interpreted as some property being absent (0) or present (1).

In this sense, a vector of binary attribute can be interpreted as a set of properties that the corresponding object has.

Example.

- The binary vector $(0, 1, 1, 0, 1)$ corresponds to the set of properties $\{a_2, a_3, a_5\}$.
- The binary vector $(0, 0, 0, 0, 0)$ corresponds to the empty set.
- The binary vector $(1, 1, 1, 1, 1)$ corresponds to the set $\{a_1, a_2, a_3, a_4, a_5\}$.

Notion of (dis-)similarity: Binary attributes

Dissimilarity measures for two vectors of binary attributes.
Each data object is represent by the corresponding set of properties that are present.

	binary attributes	sets of properties
simple match	$d_S = 1 - \frac{b+n}{b+n+x}$	
Russel & Rao	$d_R = 1 - \frac{b}{b+n+x}$	$1 - \frac{\lvert X \cap Y \rvert}{\lvert \Omega \rvert}$
Jaccard	$d_J = 1 - \frac{b}{b+x}$	$1 - \frac{\lvert X \cap Y \rvert}{\lvert X \cup Y \rvert}$
Dice	$d_D = 1 - \frac{2b}{2b+x}$	$1 - \frac{2\lvert X \cap Y \rvert}{\lvert X \rvert + \lvert Y \rvert}$

	no. of predicates that...
$b =$	...hold in both records
$n =$	...do not hold in both records
$x =$	...hold in only one of both records

x	**y**	set X	set Y	b	n	x	d_M	d_R	d_J	d_D
101000	111000	$\{a_1, a_3\}$	$\{a_1, a_2, a_3\}$	2	3	1	$0.1\bar{6}$	$0.6\bar{6}$	$0.3\bar{3}$	0.20

Notion of (dis-)similarity: Nominal attributes

- Nominal attributes may be transformed into a set of binary attributes, each of them indicating one particular feature of the attribute. Only one of the introduced binary attributes may be active at a time.
  Example. . Attribute *Manufacturer* with the values *BMW*, *Chrysler*, *Dacia*, *Ford*, *Volkswagen*.

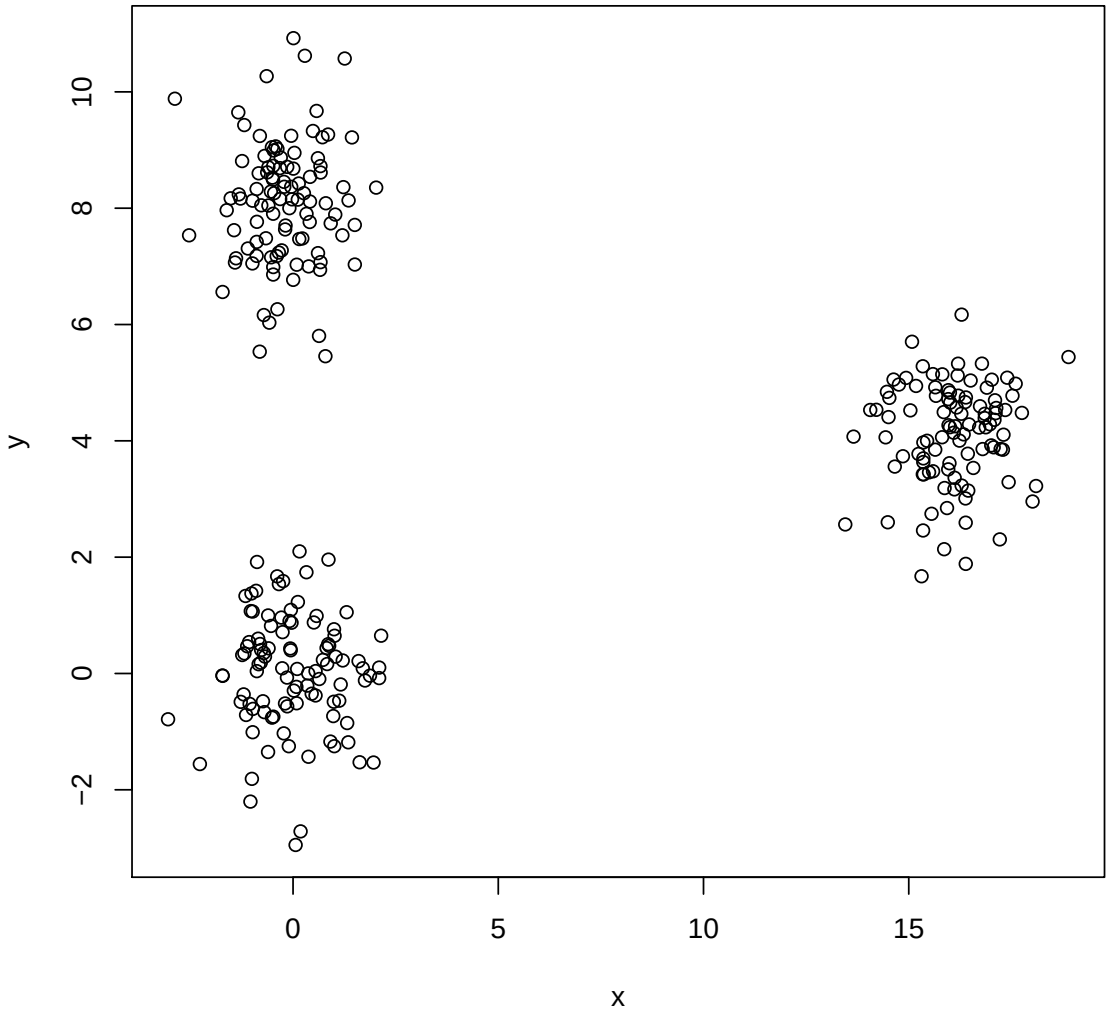
manufacturer	...
Volkswagen	...
Dacia	...
Ford	...

$\rightarrow$

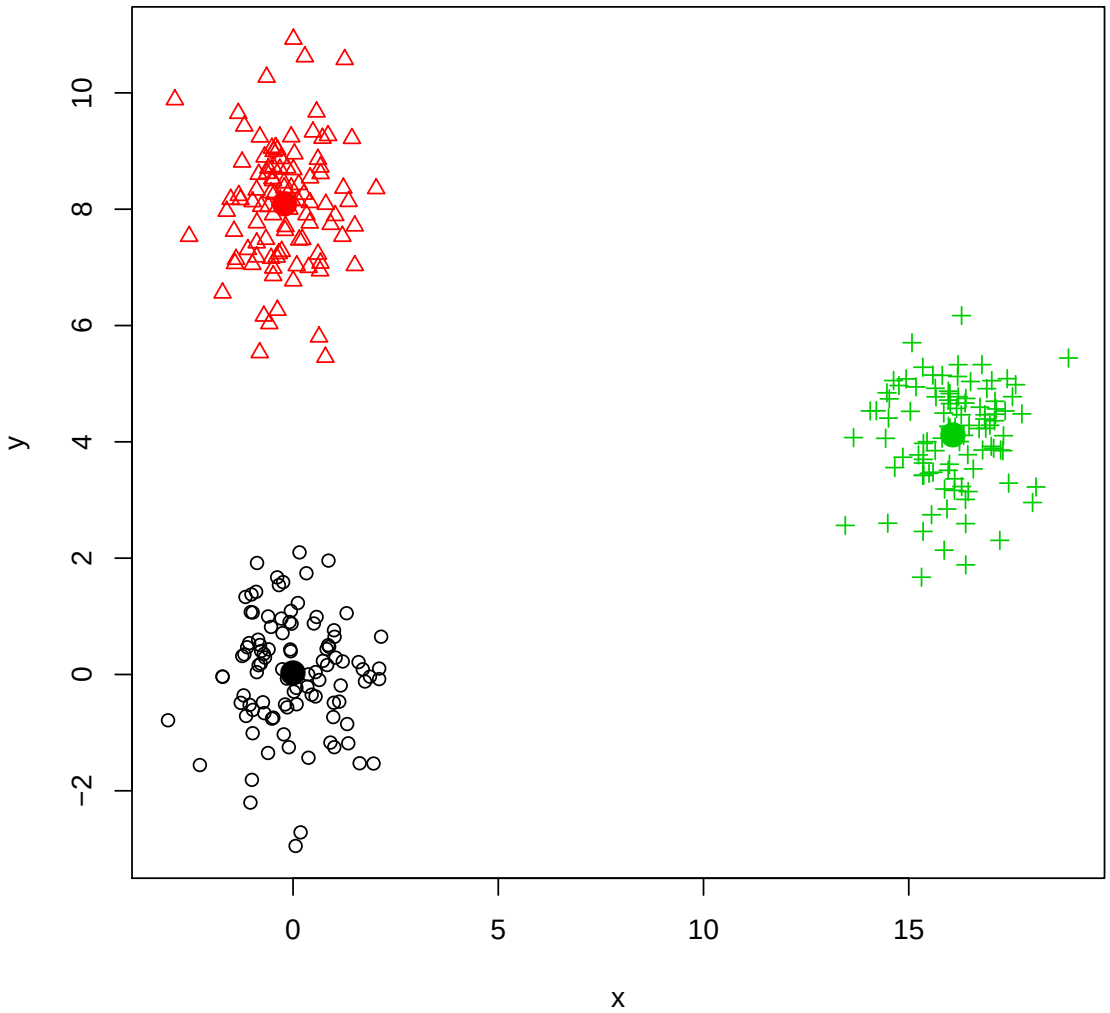
binary vector
00001
00100
00010

  Then one of the dissimilarity measures for binary attribute can be applied.
- Another way to measure similarity between two vectors of nominal attributes is to compute the proportion of attributes where both vectors have the same value, leading to the Russel & Rao dissimilarity measure.

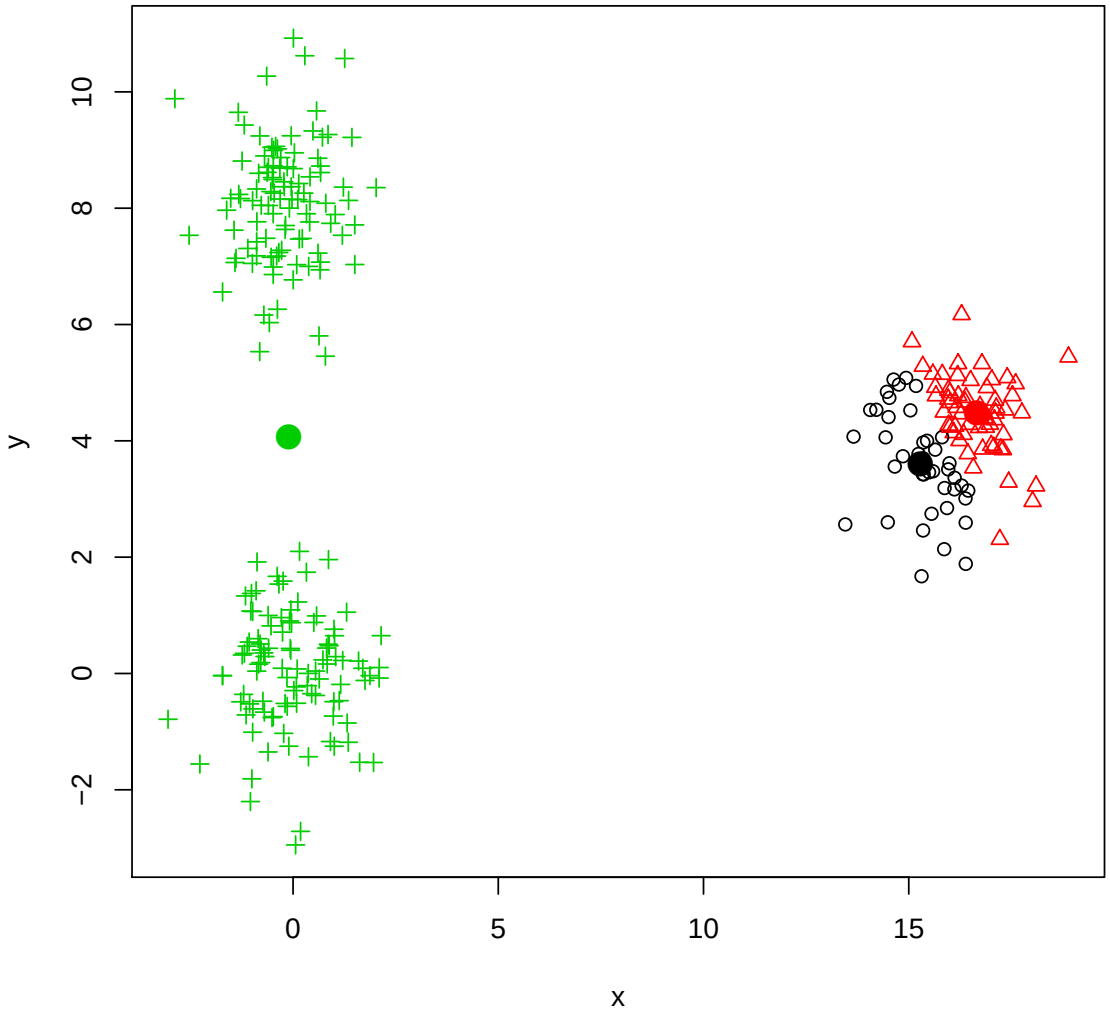
Revisited: k-means clustering



Revisited: k-means clustering



k-means clustering: 25% failure rate



k-Means clustering

Objective function

$$f = \sum_{i=1}^{k} \sum_{j=1}^{n} u_{ij} d_{ij}$$

under the constraints $u_{ij} \in \{0, 1\}$ and

$$\sum_{i=1}^{k} u_{ij} = 1 \qquad \text{for all } j = 1, \ldots, n.$$

k-Means clustering

Objective function

$$f = \sum_{i=1}^{k} \sum_{j=1}^{n} u_{ij} d_{ij}$$

under the constraints $u_{ij} \in [0, 1]$ and

$$\sum_{i=1}^{k} u_{ij} = 1 \qquad \text{for all } j = 1, \ldots, n.$$

Fuzzy c-means clustering (FCM)

Objective function

$$f = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} d_{ij}$$

under the constraints $u_{ij} \in [0, 1]$ and

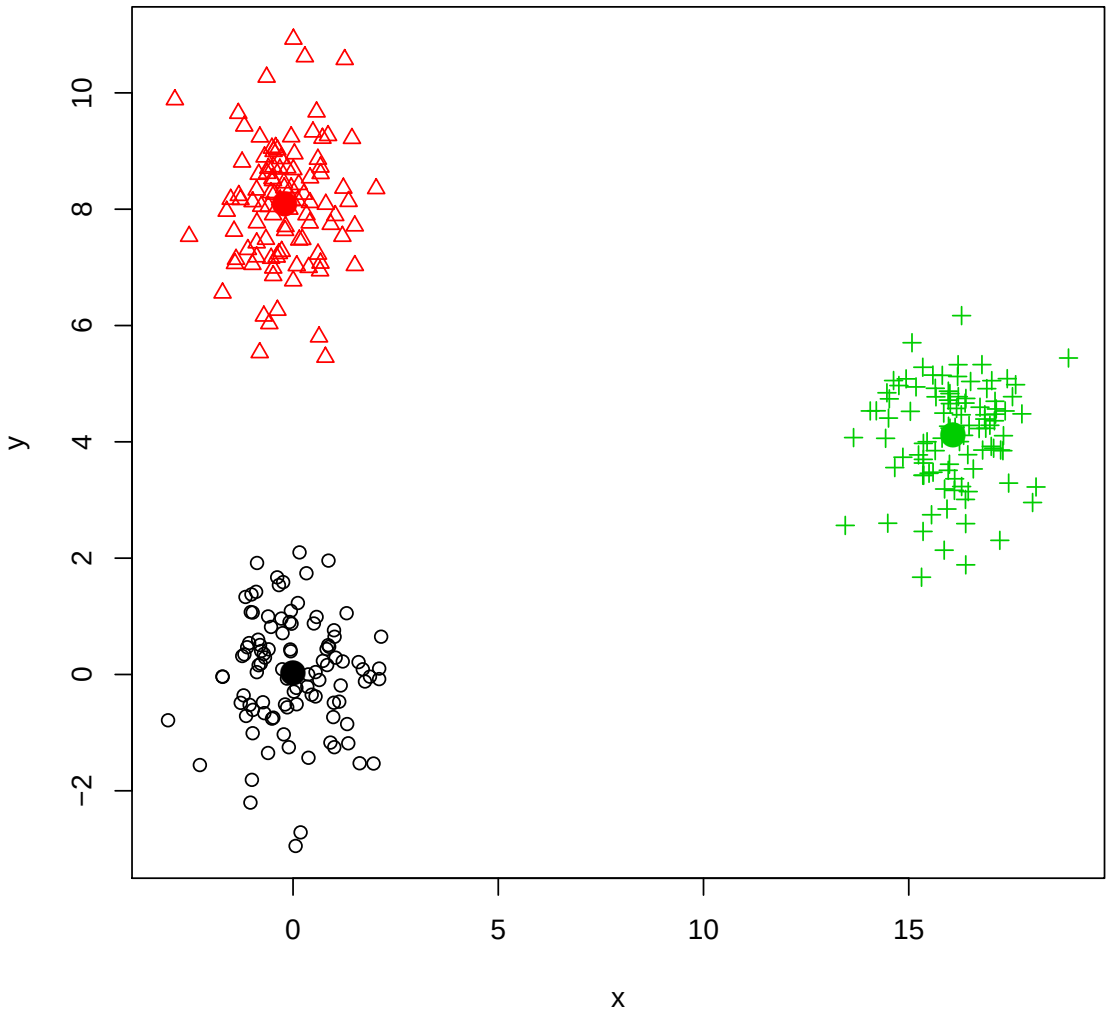
$$\sum_{i=1}^{k} u_{ij} = 1 \qquad \text{for all } j = 1, \ldots, n.$$

FCM update equations

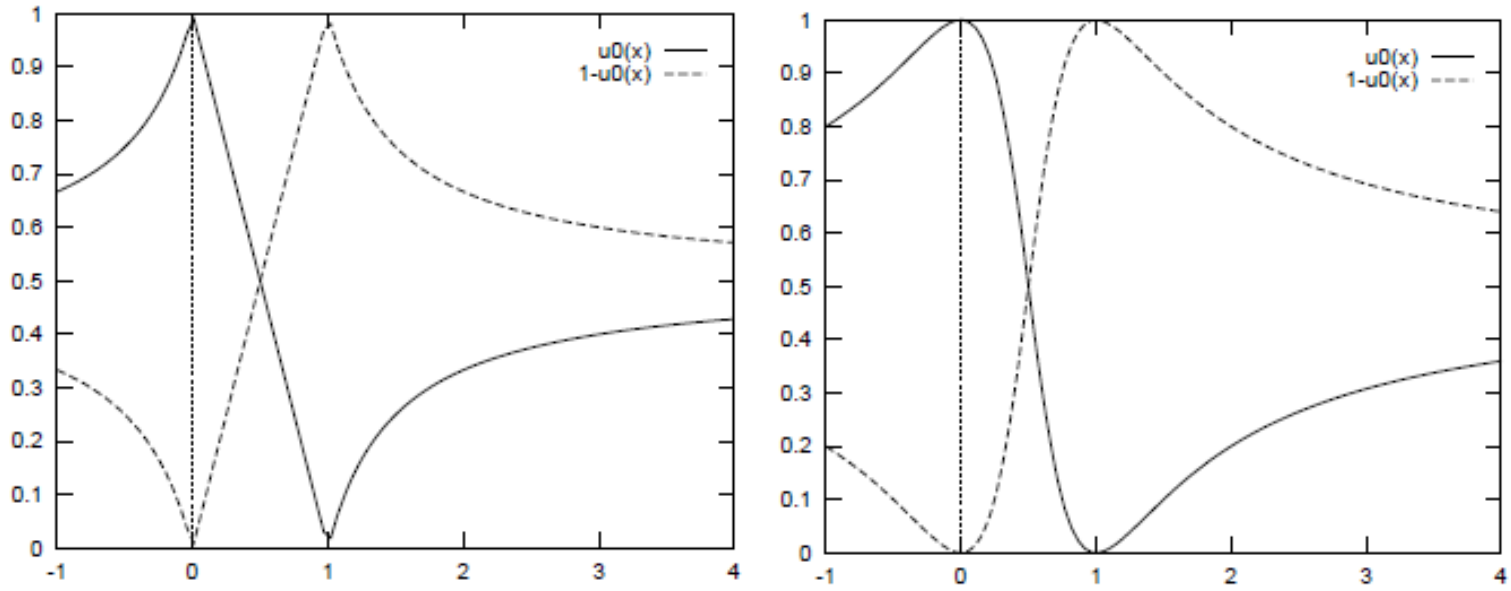
$$\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m} \parallel x_j - v_i \parallel^2 \quad \rightarrow \quad u_{ij} \quad = \quad \frac{1}{\sum_{k=1}^{c}\left(\frac{d_{ij}}{d_{kj}}\right)^{\frac{1}{m-1}}}$$

$$\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m} \parallel x_j - v_i \parallel^2 \quad \rightarrow \quad v_i \quad = \quad \frac{\sum_{j=1}^{n} u_{ij}^{m} x_j}{\sum_{j=1}^{n} u_{ij}^{m}}$$

FCM: 0% failure rate

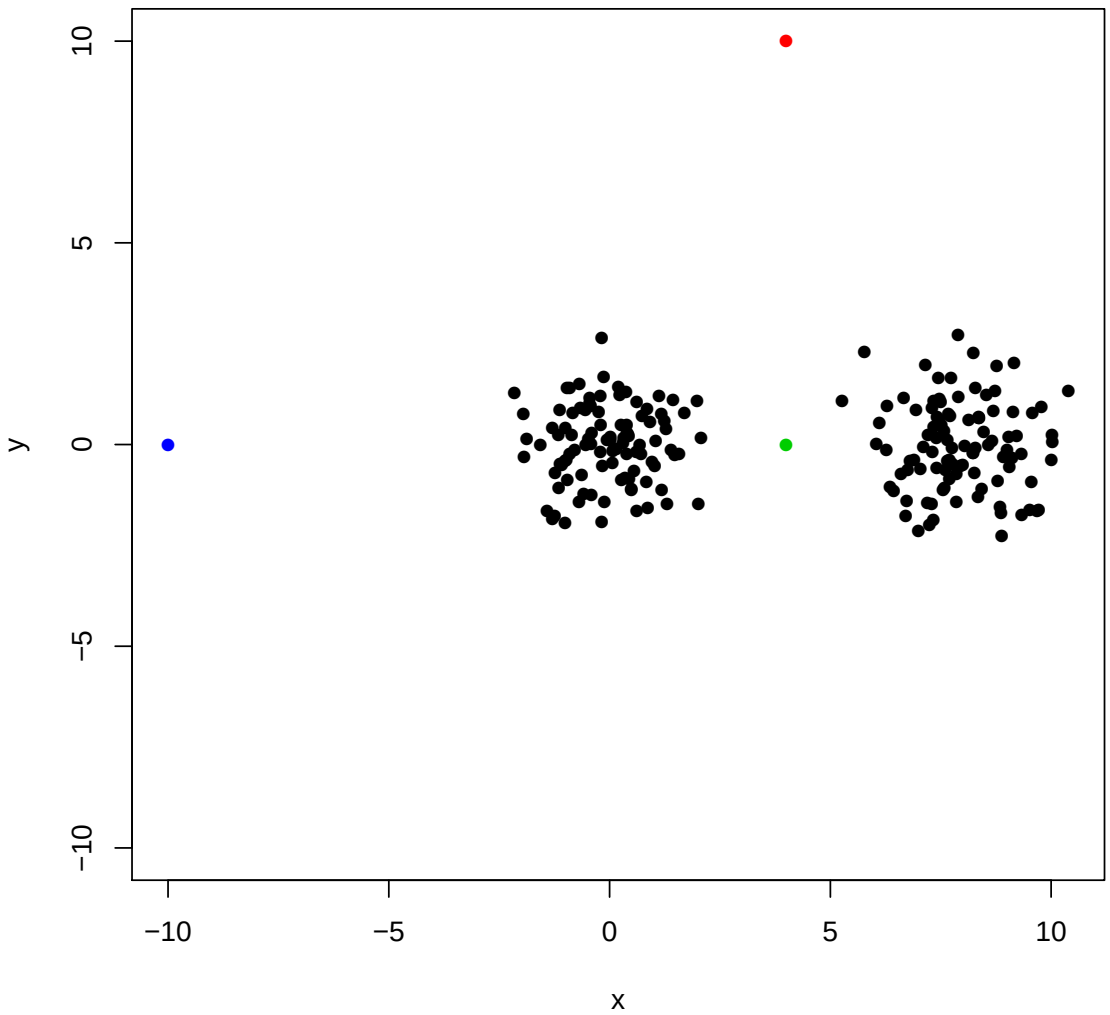


Effects of the fuzzifier



Cluster separation for fuzzifier values $m = 2$ and $m = 1.5$.

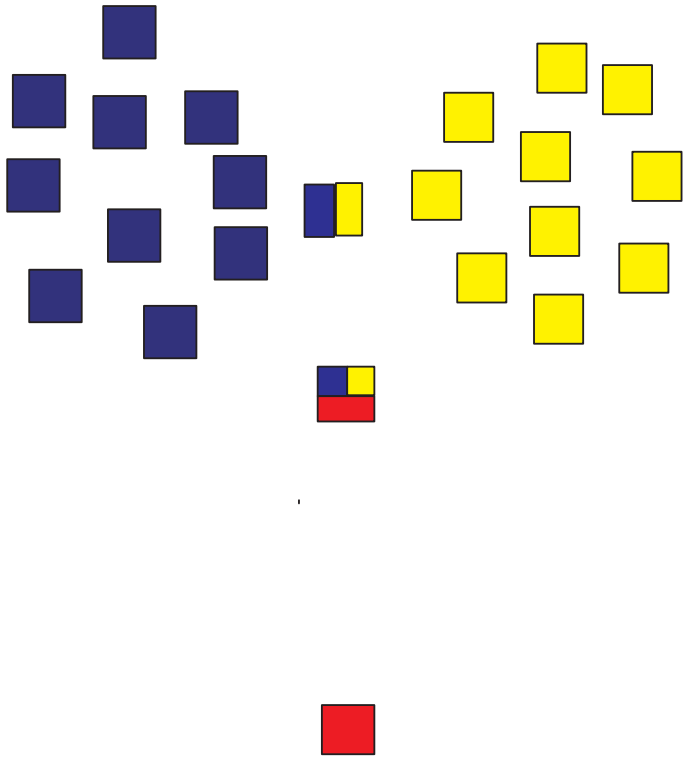
Effects of the fuzzifier



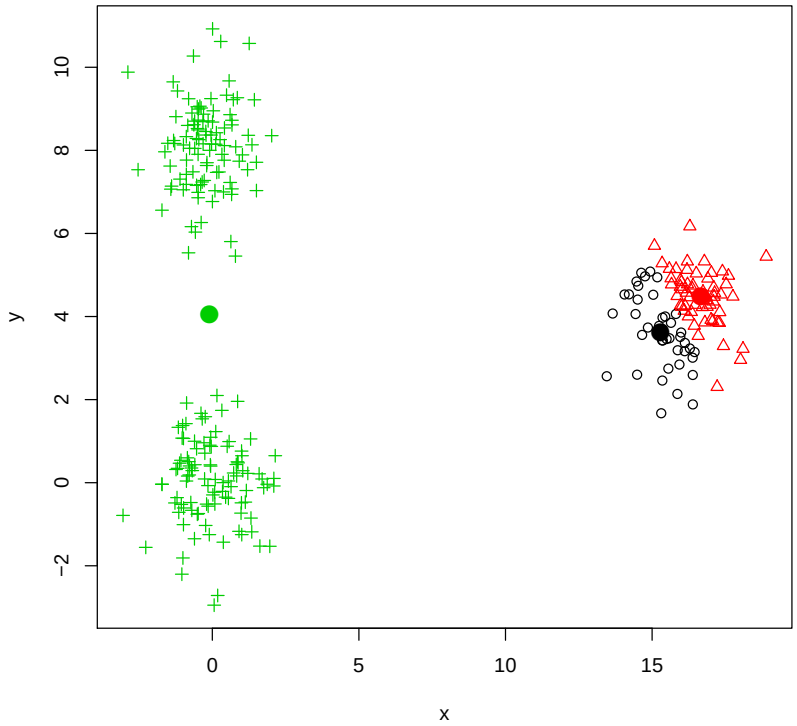
Noise clustering

Introduce an additional noise cluster.

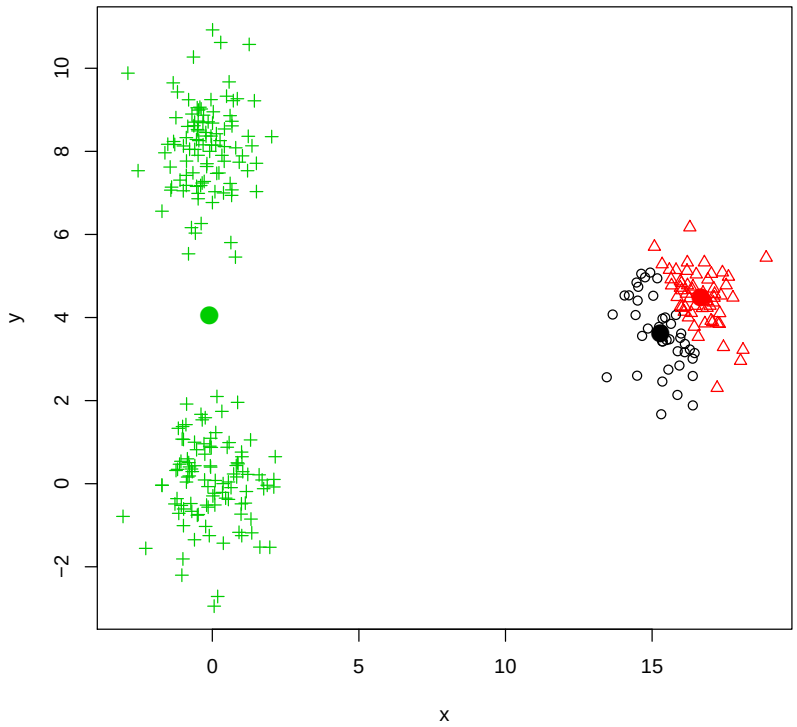
All data objects have the same (large) distance to the noise cluster.



k-means clustering: 25% failure rate, FCM 0%



k-means clustering: 25% failure rate, FCM 0%



Does the "fuzzification" of the objective function let local minima vanish?

FCM objective function

$$f = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} d_{ij}$$

under the constraints $u_{ij} \in [0, 1]$ and

$$\sum_{i=1}^{k} u_{ij} = 1 \qquad \text{for all } j = 1, \ldots, n.$$

FCM objective function

$$f = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} d_{ij}$$

under the constraints $u_{ij} \in [0, 1]$ and

$$\sum_{i=1}^{k} u_{ij} = 1 \qquad \text{for all } j = 1, \ldots, n.$$

Find the (local) minima!

FCM objective function

$$f = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} d_{ij}$$

under the constraints $u_{ij} \in [0, 1]$ and

$$\sum_{i=1}^{k} u_{ij} = 1 \qquad \text{for all } j = 1, \ldots, n.$$

Find the (local) minima!???

FCM objective function

$$f = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m d_{ij}$$

under the constraints $u_{ij} \in [0,1]$ and

$$\sum_{i=1}^{k} u_{ij} = 1 \qquad \text{for all } j = 1, \ldots, n.$$

Find the (local) minima!???

Try to show it at least with some examples.

FCM objective function

How to visualise a function with so many parameters?

$$c \cdot \#\text{dimensions} + c \cdot \#\text{data}$$

FCM objective function

How to visualise a function with so many parameters?

$$c \cdot \#\text{dimensions} + c \cdot \#\text{data}$$

Replace the membership degree by the optinal values:

$$f = \sum_{i=1}^{c} \sum_{j=1}^{n} \left(\frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}} \right)^{\frac{1}{m-1}}} \right)^{m} d_{ij}$$

FCM objective function

How to visualise a function with so many parameters?

$$c \cdot \#\text{dimensions} + c \cdot \#\text{data}$$

Replace the membership degree by the optinal values:

$$f = \sum_{i=1}^{c} \sum_{j=1}^{n} \left(\frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}} \right)^{\frac{1}{m-1}}} \right)^{m} d_{ij}$$

Number of parameters: $c \cdot \#\text{dimensions}$

FCM objective function

How to visualise a function with so many parameters?

$$c \cdot \#\text{dimensions} + c \cdot \#\text{data}$$

Replace the membership degree by the optinal values:

$$f = \sum_{i=1}^{c} \sum_{j=1}^{n} \left(\frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}} \right)^{\frac{1}{m-1}}} \right)^{m} d_{ij}$$

Number of parameters: $c \cdot \#\text{dimensions}$

Visualisation for two clusters in one dimension as a 3D plot possible.

FCM objective function visualisation

Gain an extra cluster by adding a noise cluster.

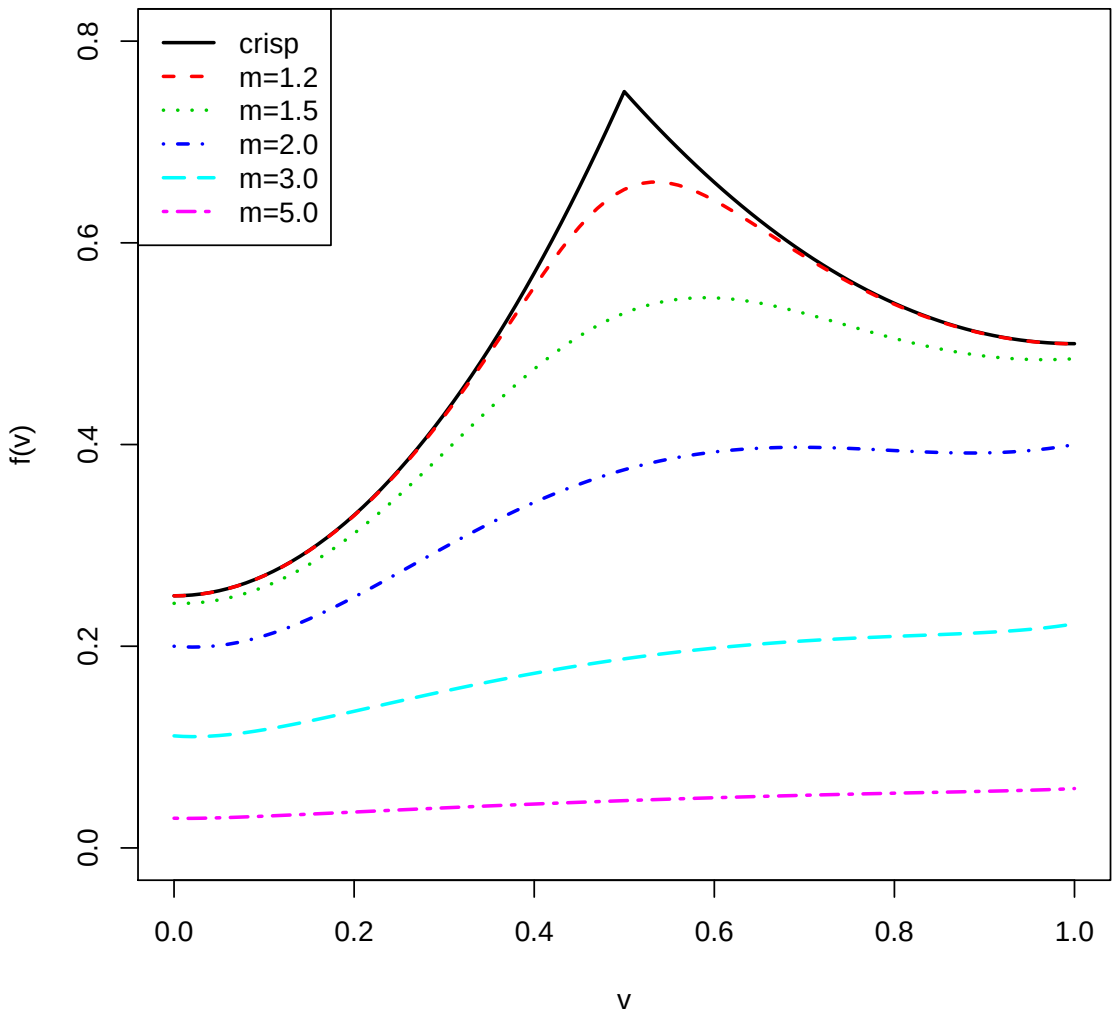
Simple one-dimensional data set:

- k points at 0.
- One outlier at u.
- One cluster (at $v = 0$) and one outlier (at u).

Objective function:

$$f(v) = \frac{k \cdot v^2}{\left(1 + \left(\frac{v^2}{\delta}\right)^{\frac{1}{m-1}}\right)^m} + \frac{(v-u)^2}{\left(1 + \left(\frac{(v-u)^2}{\delta}\right)^{\frac{1}{m-1}}\right)^m} + \frac{k \cdot \delta}{\left(1 + \left(\frac{\delta}{v^2}\right)^{\frac{1}{m-1}}\right)^m} + \frac{\delta}{\left(1 + \left(\frac{\delta}{(v-u)^2}\right)^{\frac{1}{m-1}}\right)^m}$$

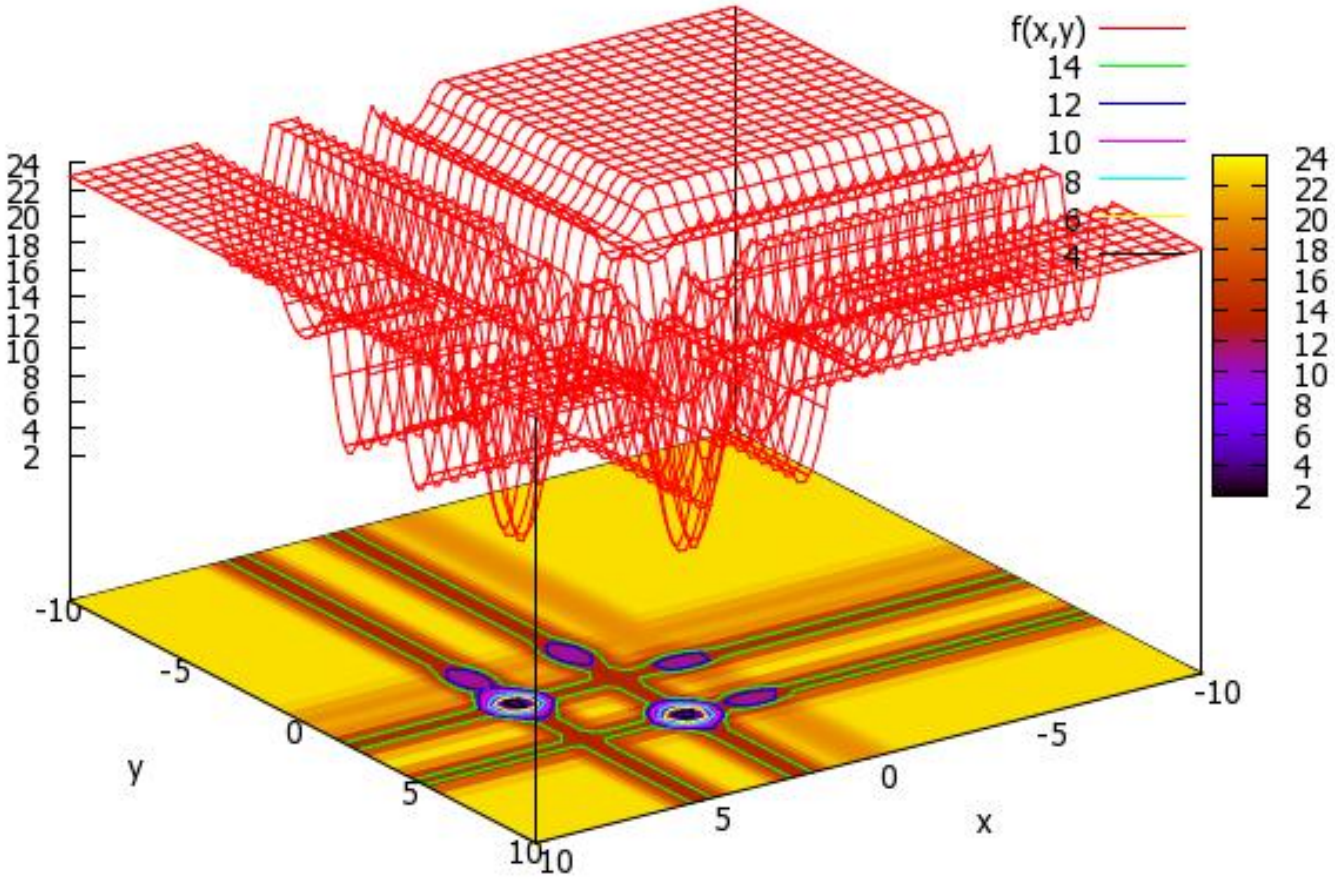
FCM objective function visualisation



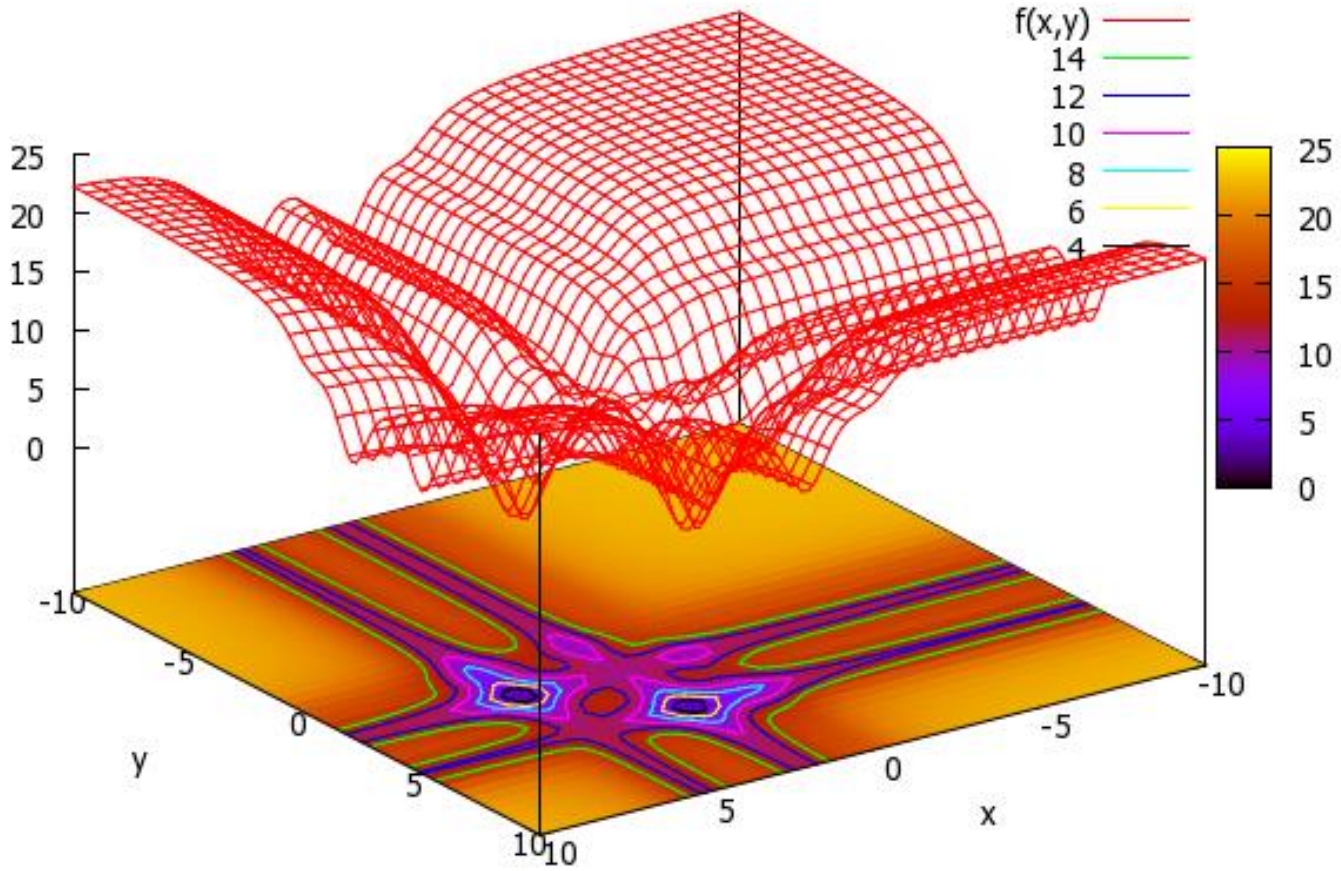
FCM objective function visualisation

- One clusters at $u = 2$ with 10 points.
- One clusters at $w = 10$ with 10 points.
- 3 "noise" points at $x = 0$.

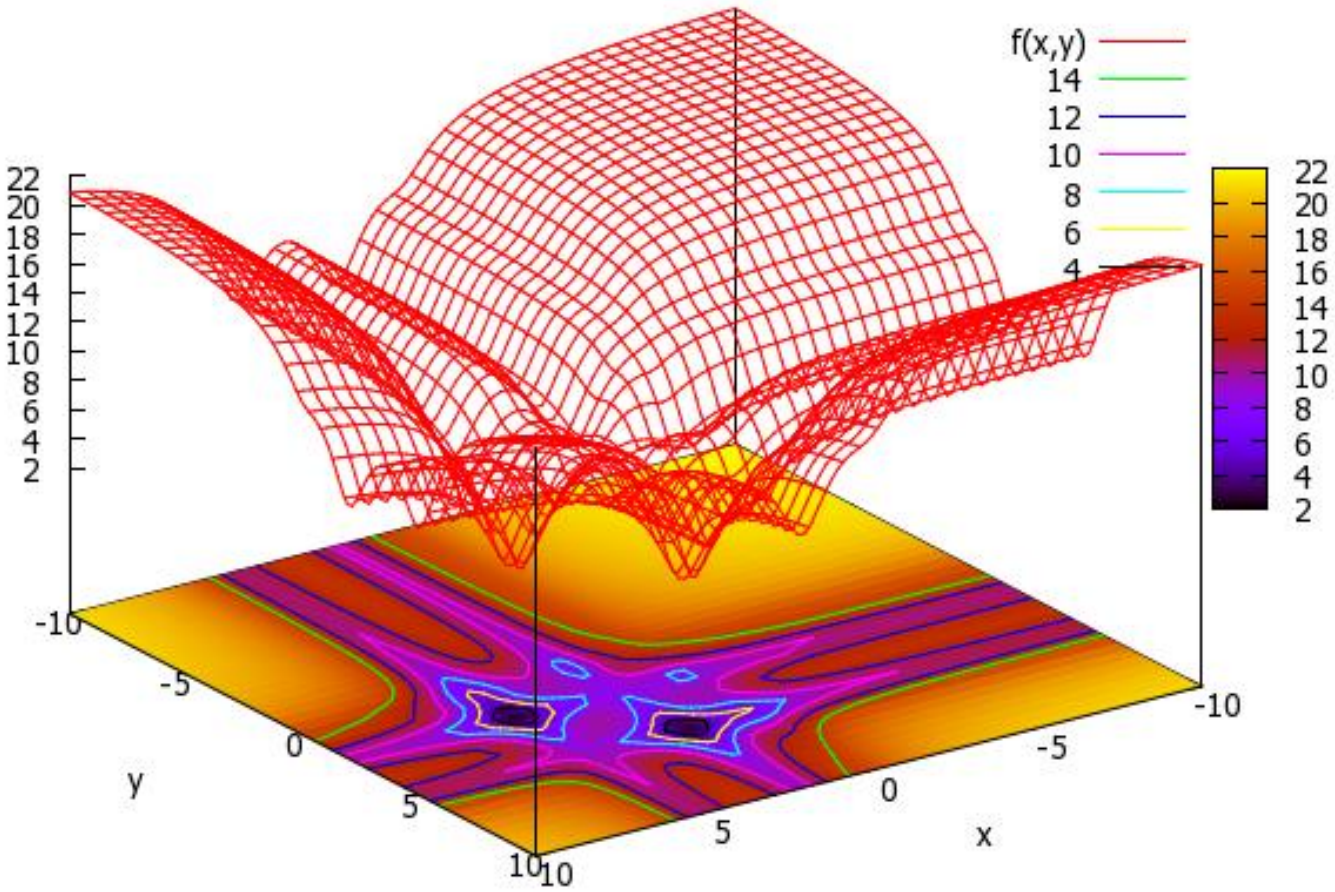
FCM objective function visualisation: $m = 1$



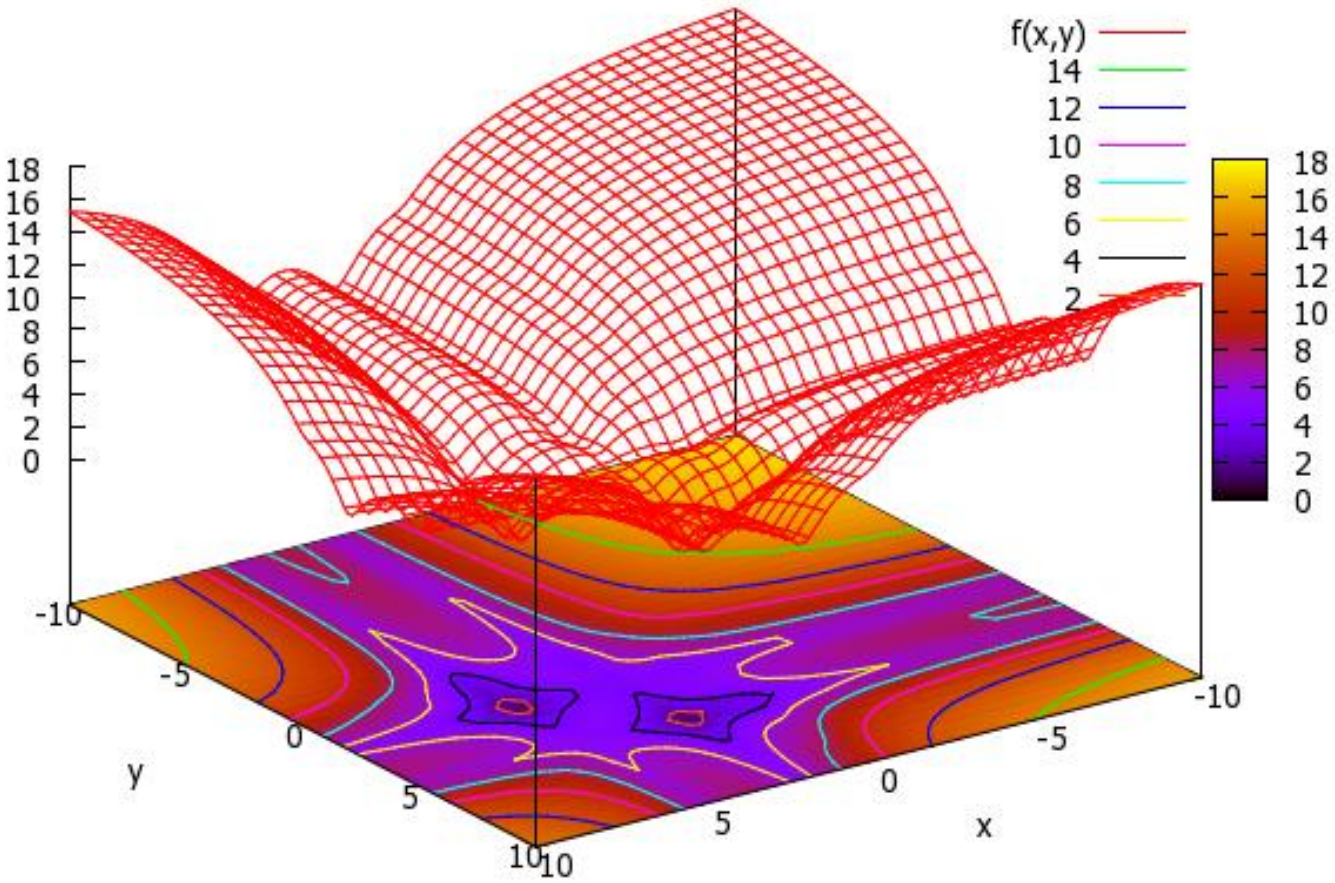
FCM objective function visualisation: $m = 2$



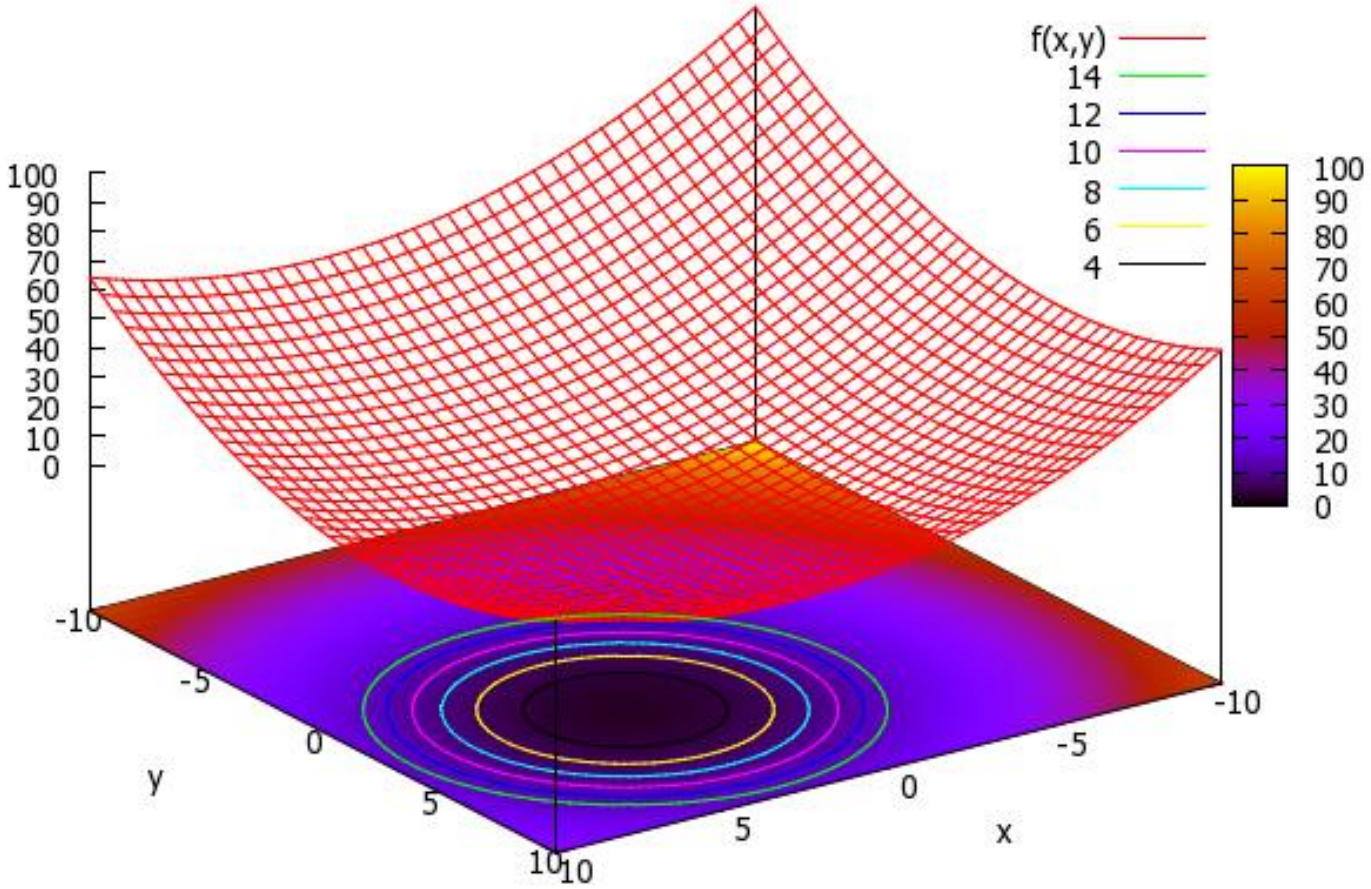
FCM objective function visualisation: $m = 2.3$



FCM objective function visualisation: $m = 3$

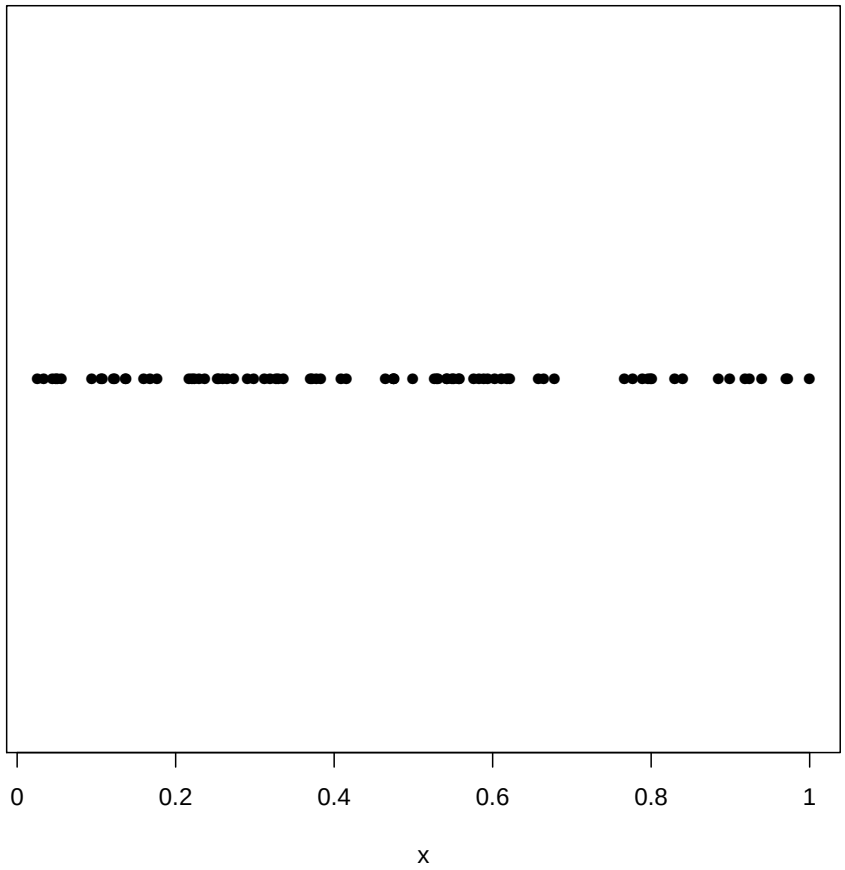


FCM objective function visualisation: $m = 4$



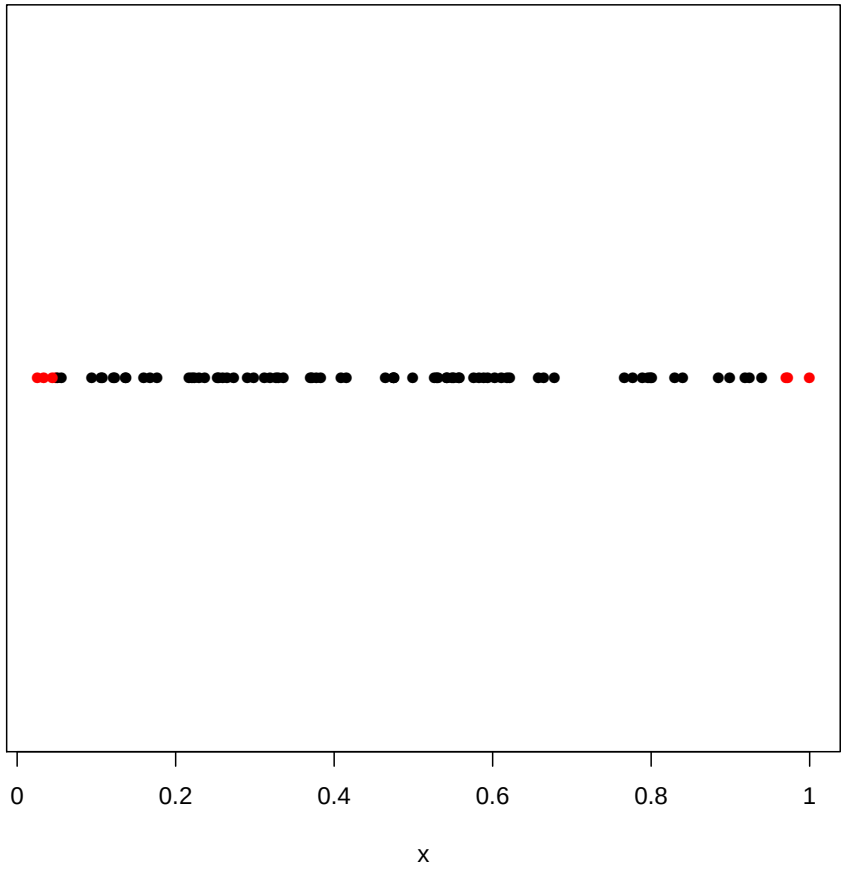
High-dimensional data are different

Attribute with values from a uniform distribution on $[0, 1]$.



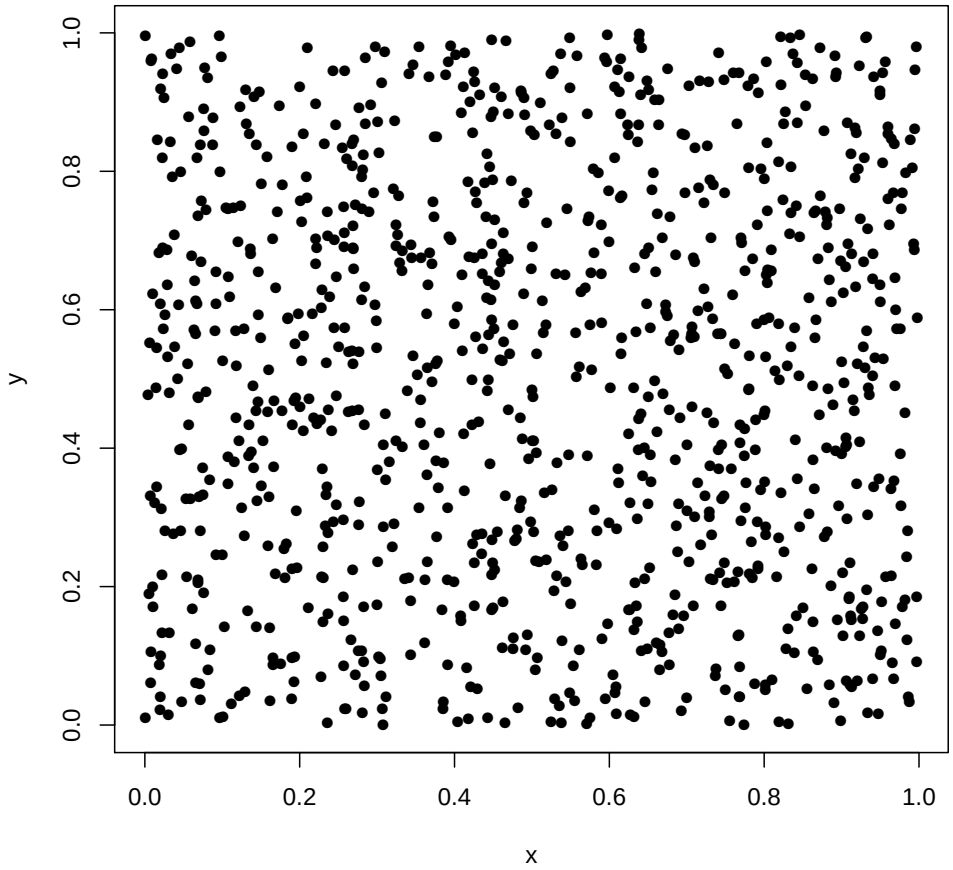
High-dimensional data are different

Data at the edge (distance < 0.05, ca. 10% of the data)



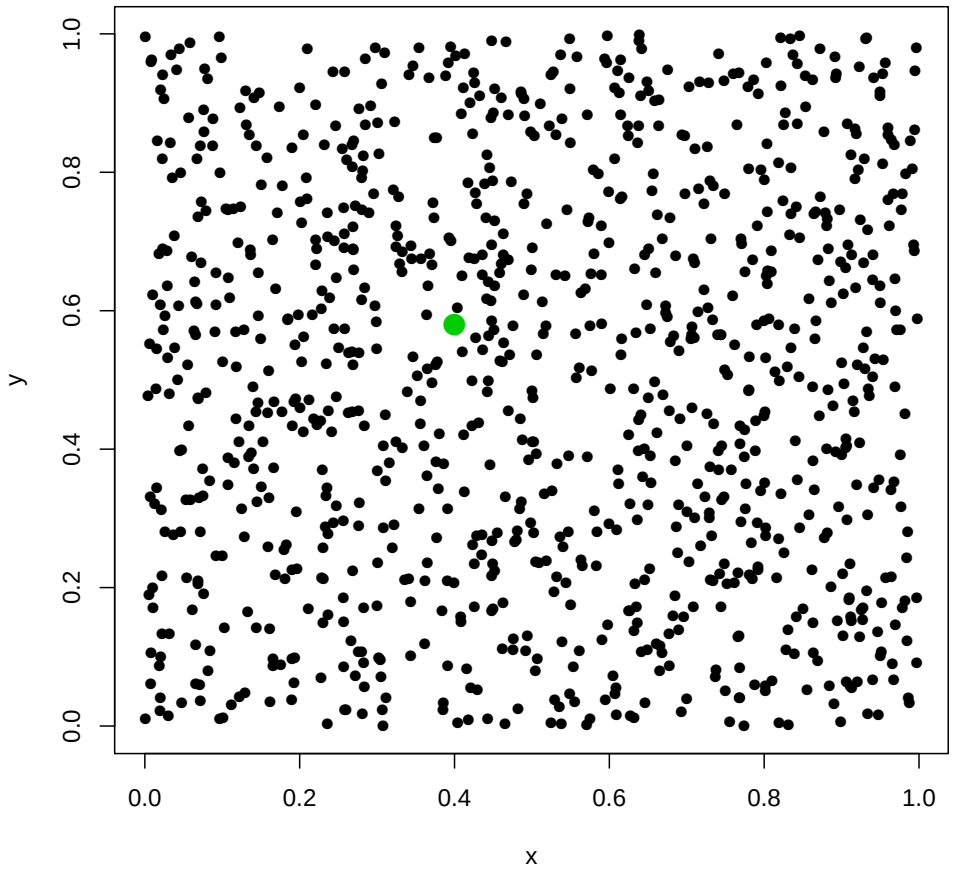
High-dimensional data are different

Two attributes with values in $[0, 1]$.



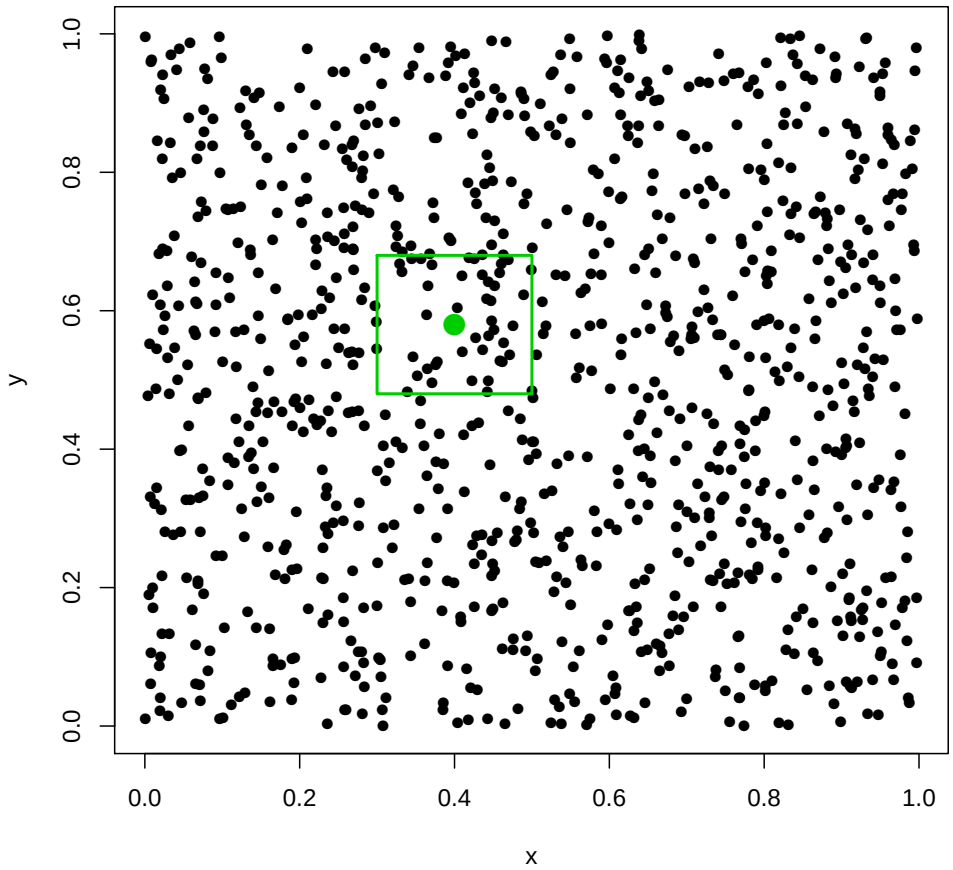
High-dimensional data are different

Typical point



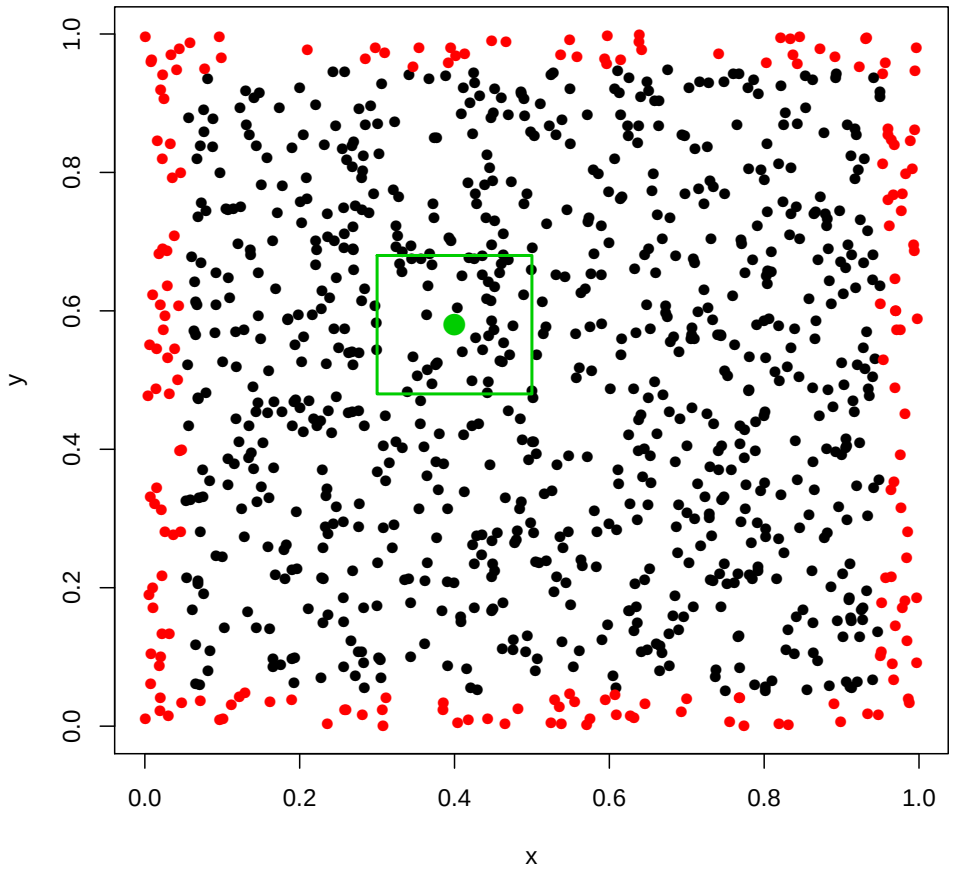
High-dimensional data are different

Typical point and its neighbours



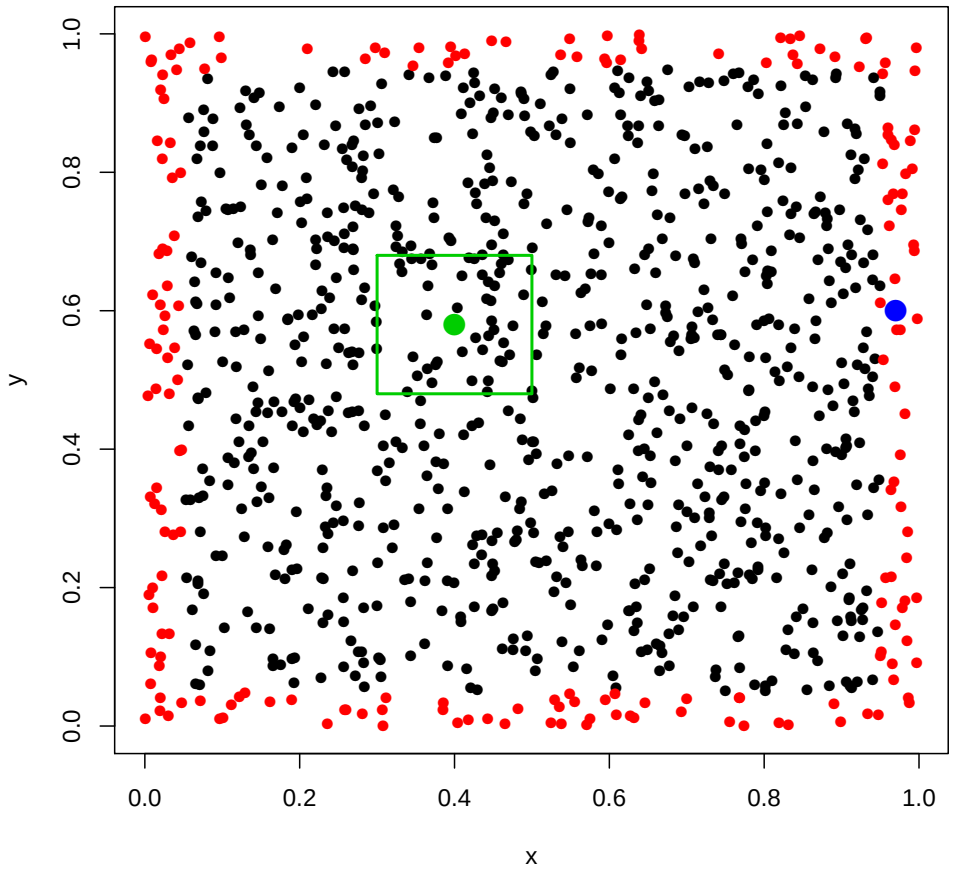
High-dimensional data are different

Points at the edge (distance < 0.05)



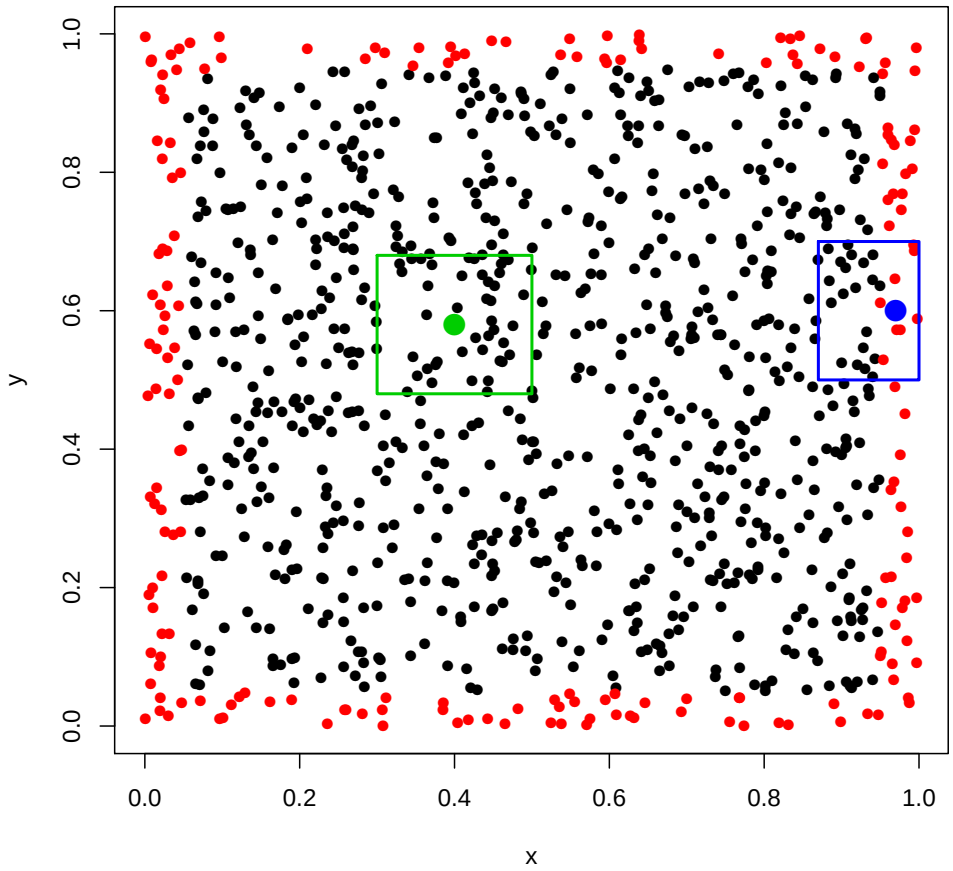
High-dimensional data are different

Typical point at the edge



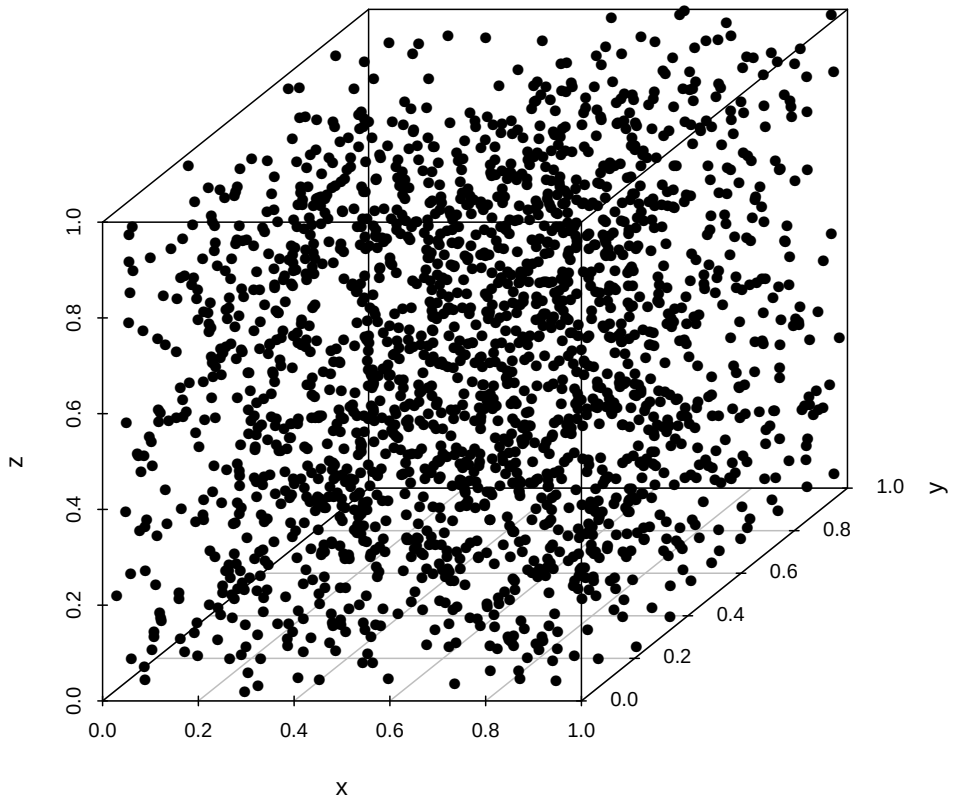
High-dimensional data are different

Typical point at the edge and its neighbours



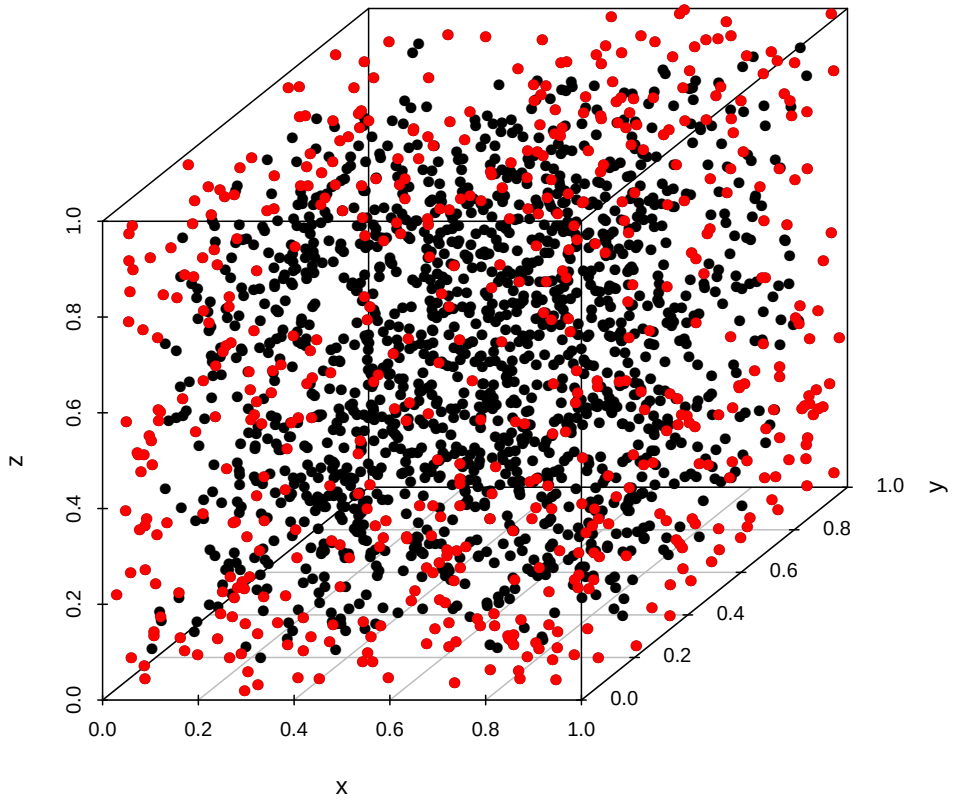
High-dimensional data are different

Three attributes with values in $[0, 1]$.



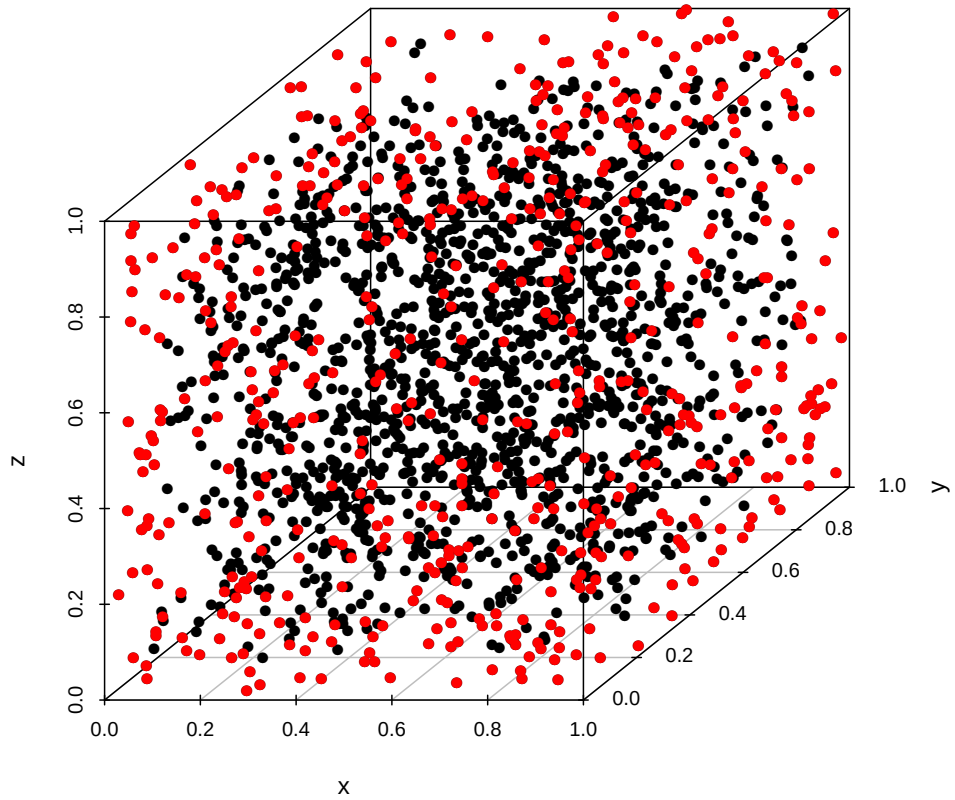
High-dimensional data are different

Points at the edge (distance < 0.05)



High-dimensional data are different

538 of 2000 data points (26,9%) are close to the edge.



High-dimensional data are different

How many data points close to the edge would we expect?

High-dimensional data are different

How many data points close to the edge would we expect?

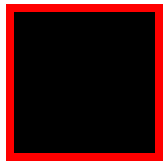
- For one dimension: 10%

High-dimensional data are different

How many data points close to the edge would we expect?

- For one dimension: 10%

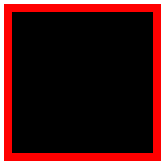
- For two dimensions $1 - (0.9)^2 = 0.19 = 19\%$



High-dimensional data are different

How many data points close to the edge would we expect?

- For one dimension: 10%

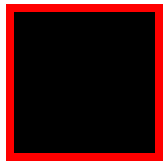


- For two dimensions $1 - (0.9)^2 = 0.19 = 19\%$
- For three dimensions $1 - (0.9)^3 = 0.271 = 27,1\%$

High-dimensional data are different

How many data points close to the edge would we expect?

- For one dimension: 10%

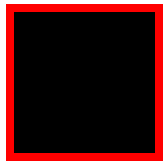


- For two dimensions $1 - (0.9)^2 = 0.19 = 19\%$
- For three dimensions $1 - (0.9)^3 = 0.271 = 27,1\%$
- For 50 dimensions $1 - (0.9)^{50} = 0.995 = 99,5\%$

High-dimensional data are different

How many data points close to the edge would we expect?

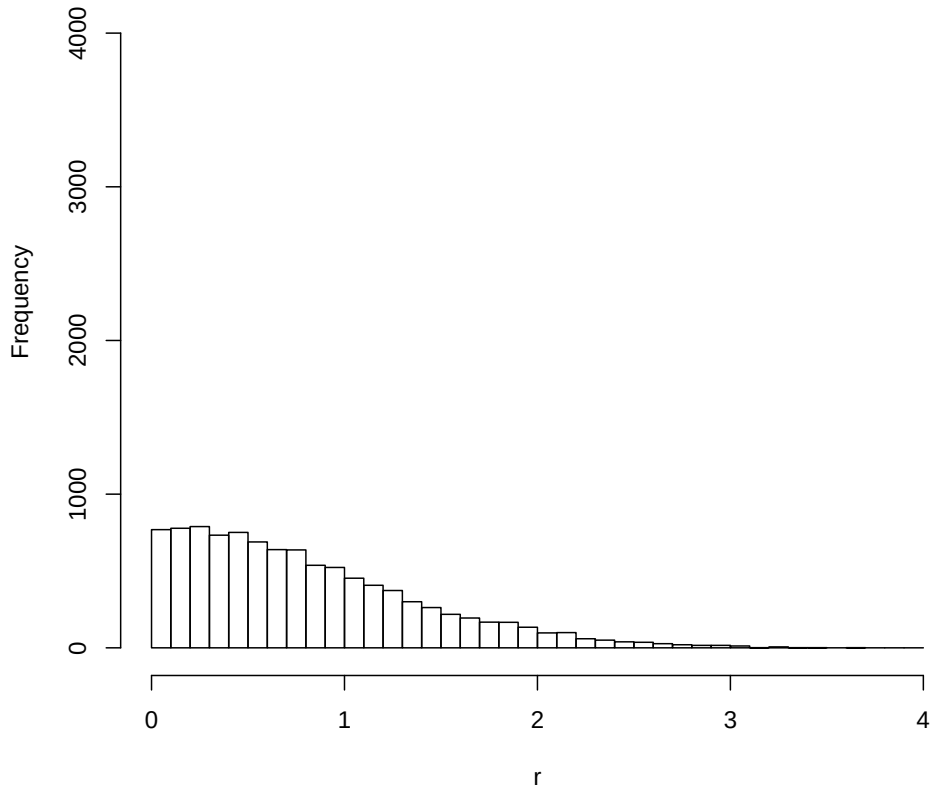
- For one dimension: 10%



- For two dimensions $1-(0.9)^2 = 0.19 = 19\%$
- For three dimensions $1-(0.9)^3 = 0.271 = 27,1\%$
- For 50 dimensions $1-(0.9)^{50} = 0.995 = 99,5\%$
- Defining points close to the edge by a distance less than 0.005 instead of 0.05, in 50 dimensions there are still $1-(0.99)^{50} = 0.395 = 39,5\%$ points close to the edge.

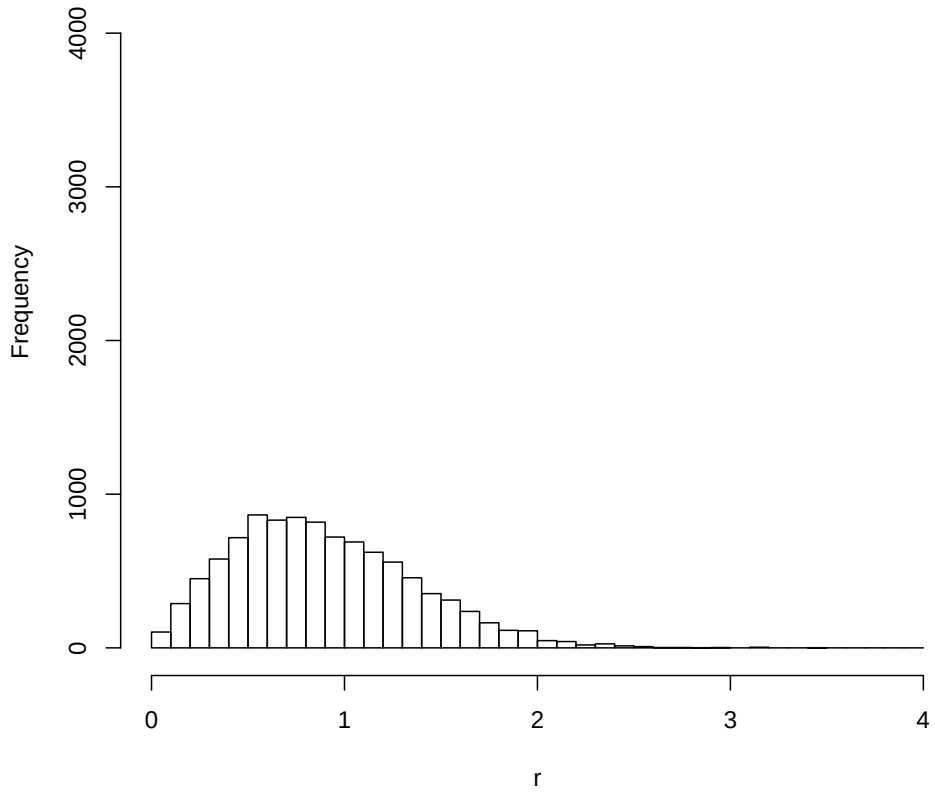
High-dimensional data are different

Distribution of the (absolute) distances to the mean for a standard normal distribution.



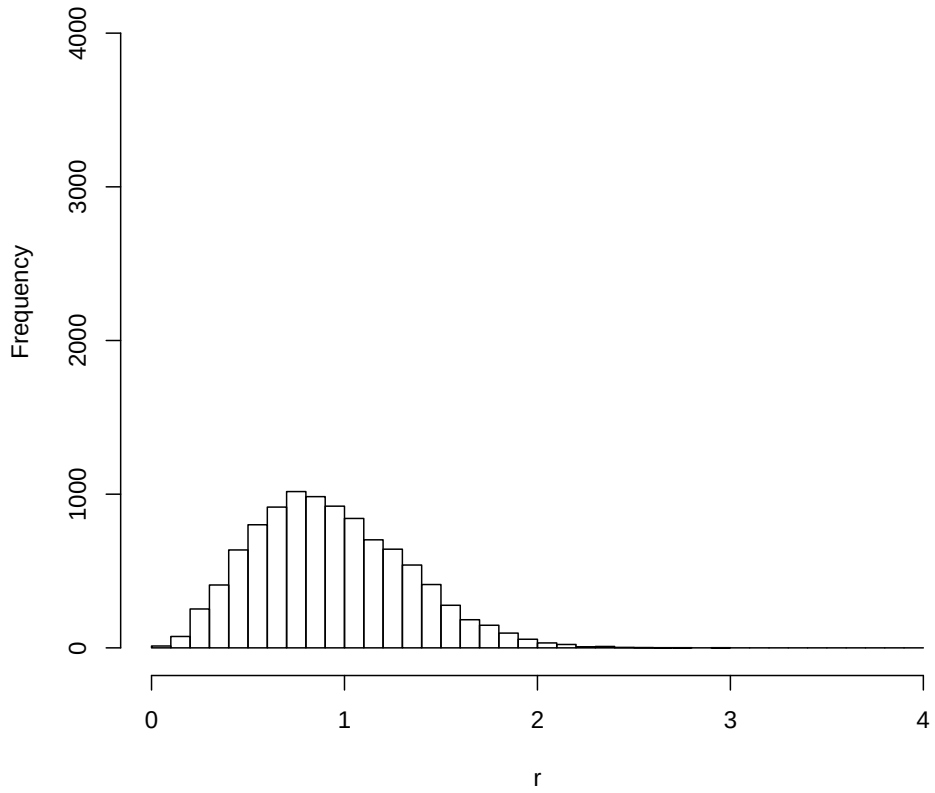
High-dimensional data are different

Distribution of the (absolute) distances to the mean for 2 independent normal distribution (scaled by the factor $\sqrt{2}$).



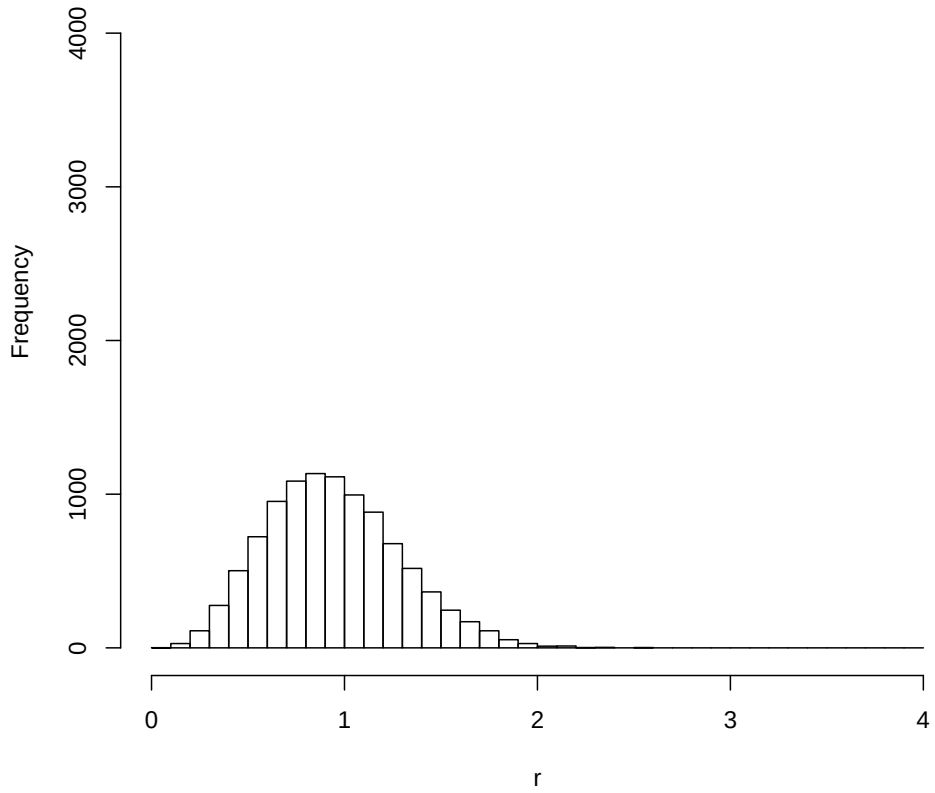
High-dimensional data are different

Distribution of the (absolute) distances to the mean for 3 independent normal distribution (scaled by the factor $\sqrt{3}$).



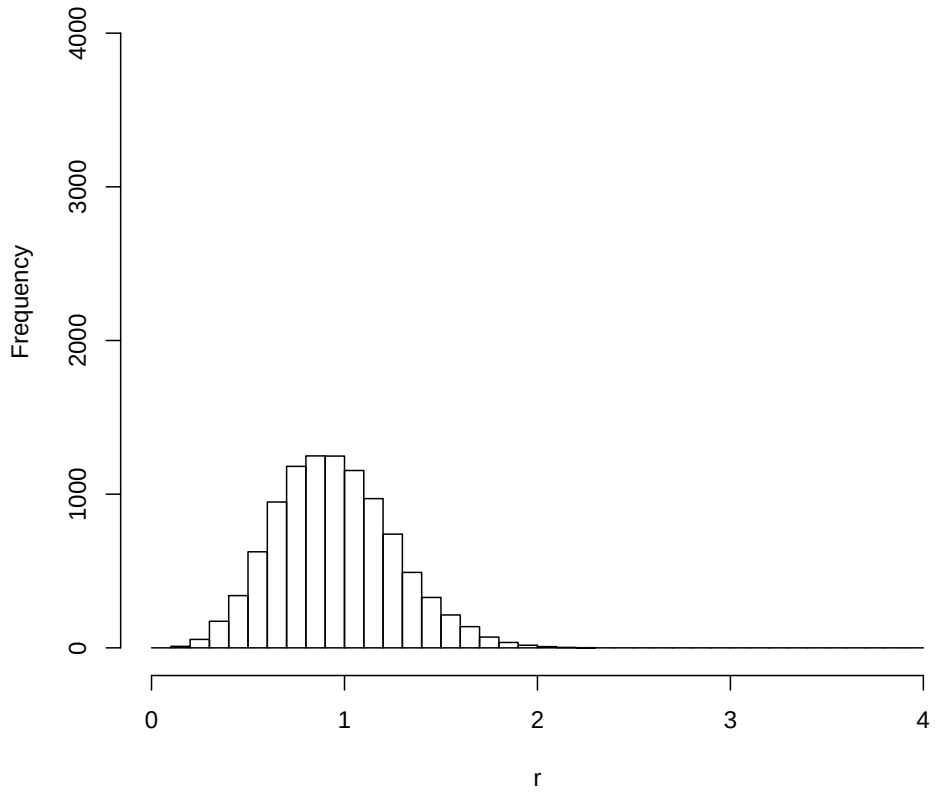
High-dimensional data are different

Distribution of the (absolute) distances to the mean for 4 independent normal distribution (scaled by the factor $\sqrt{4}$).



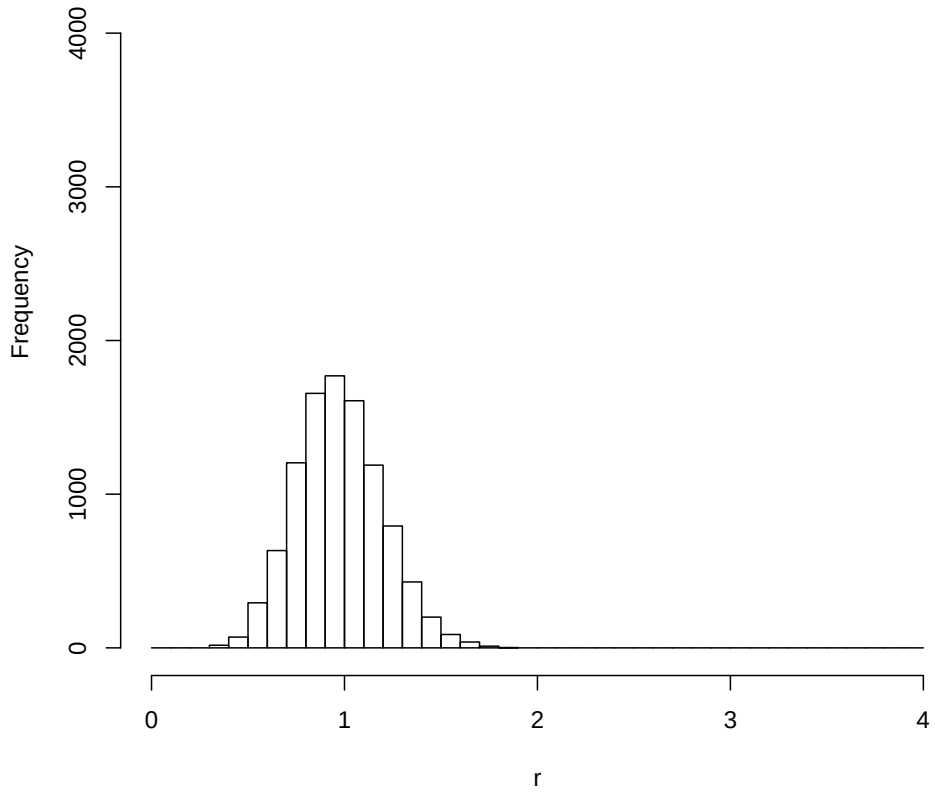
High-dimensional data are different

Distribution of the (absolute) distances to the mean for 5 independent normal distribution (scaled by the factor $\sqrt{5}$).



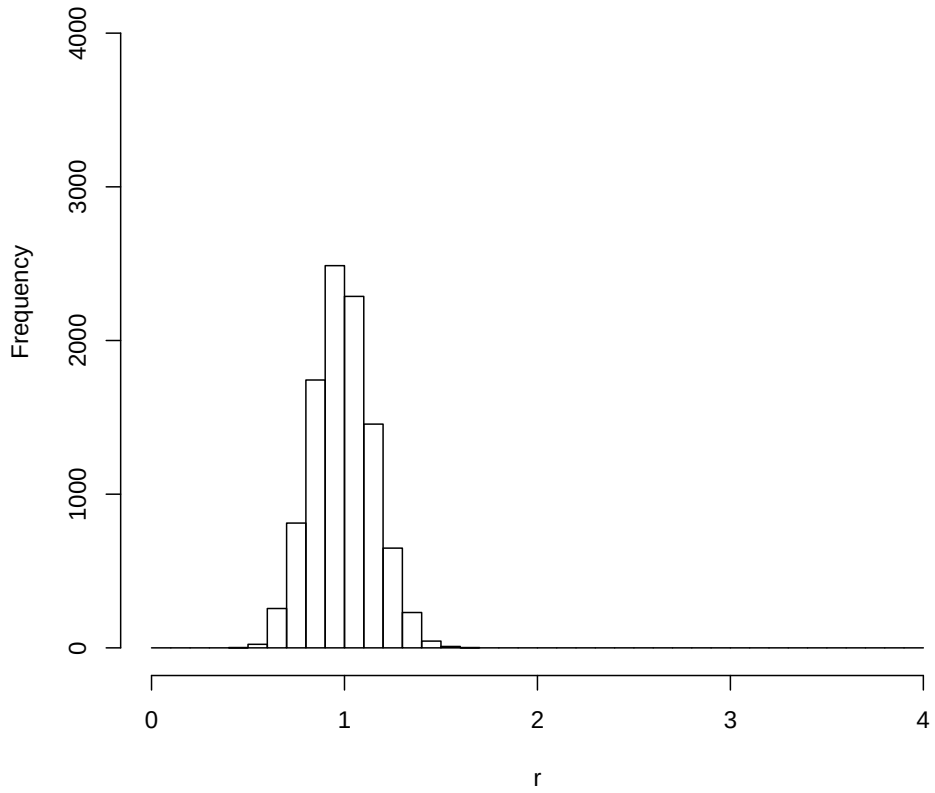
High-dimensional data are different

Distribution of the (absolute) distances to the mean for 10 independent normal distribution (scaled by the factor $\sqrt{10}$).



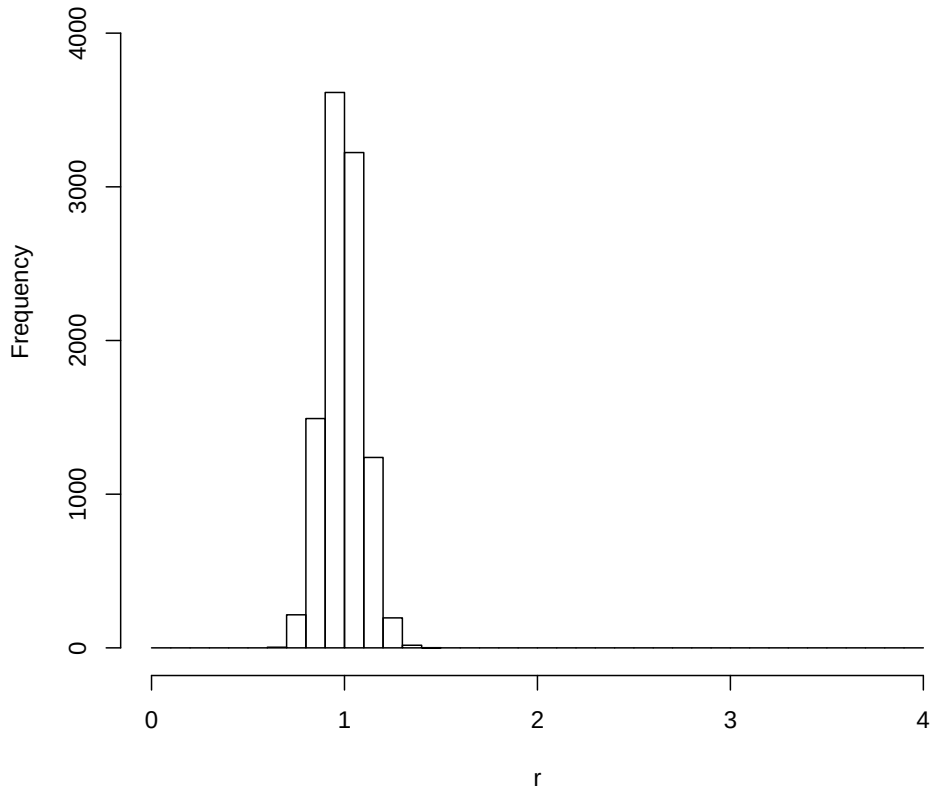
High-dimensional data are different

Distribution of the (absolute) distances to the mean for 20 independent normal distribution (scaled by the factor $\sqrt{20}$).



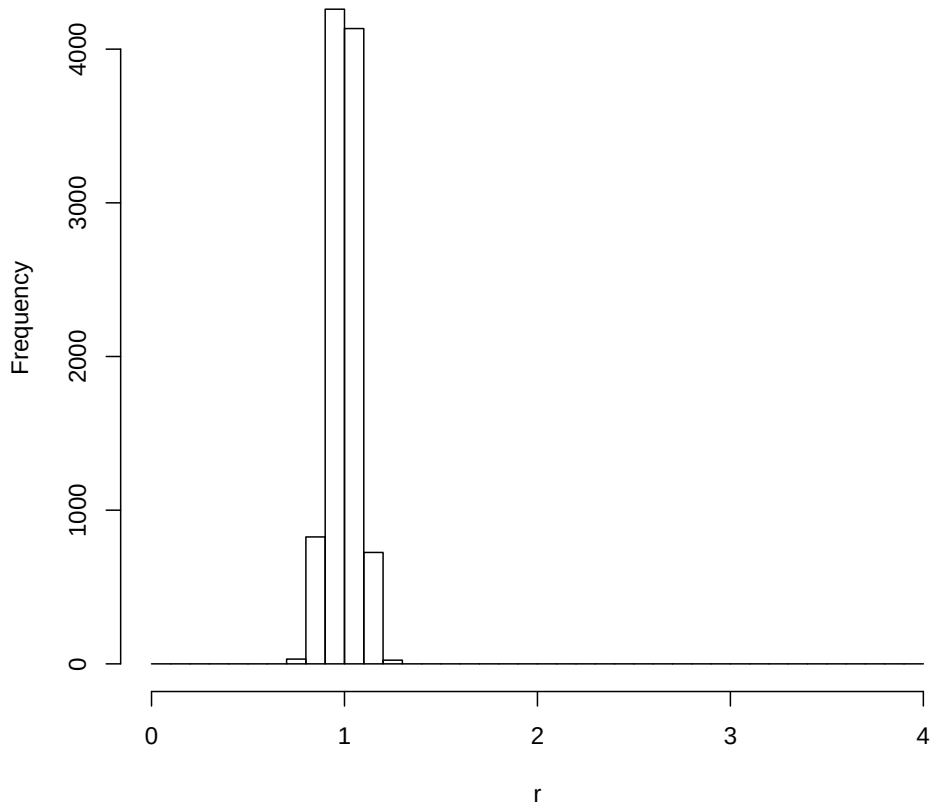
High-dimensional data are different

Distribution of the (absolute) distances to the mean for 50 independent normal distribution (scaled by the factor $\sqrt{50}$).

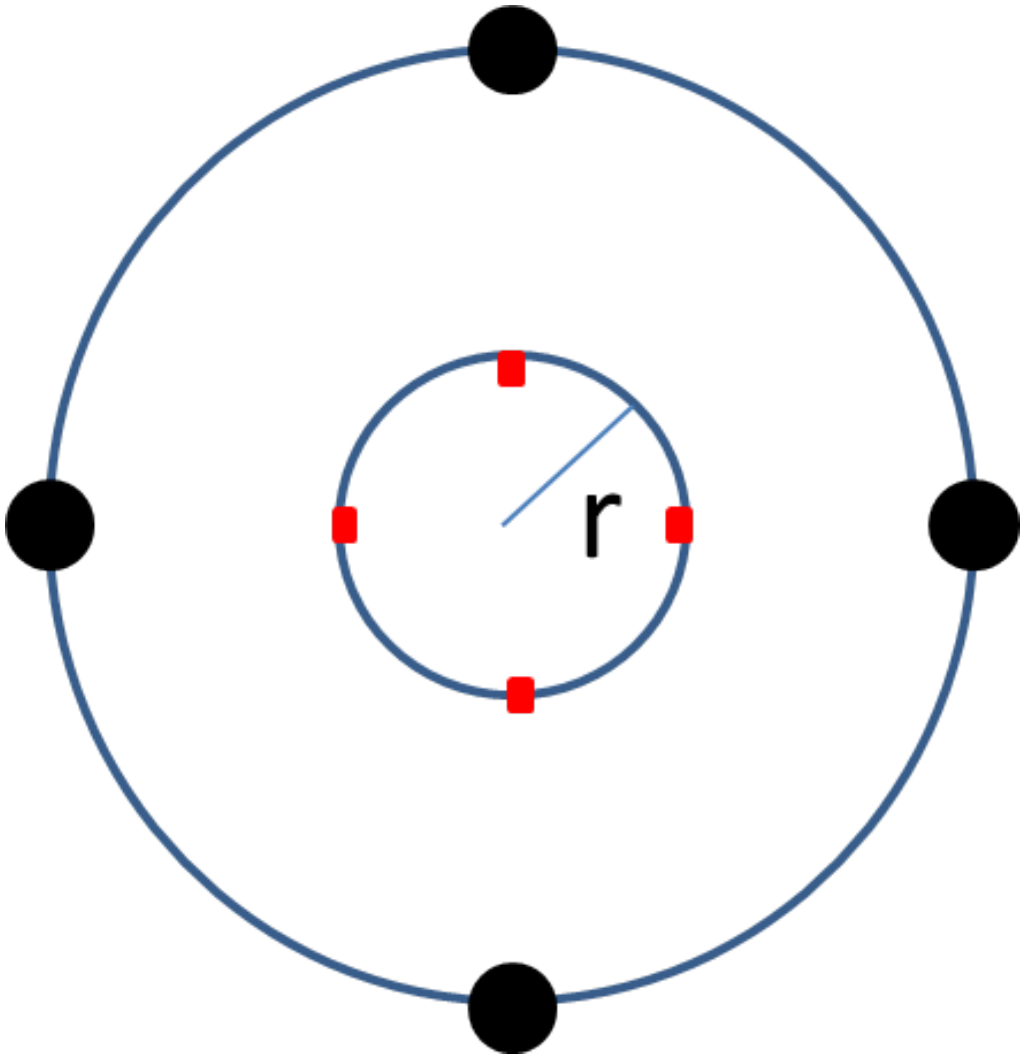


High-dimensional data are different

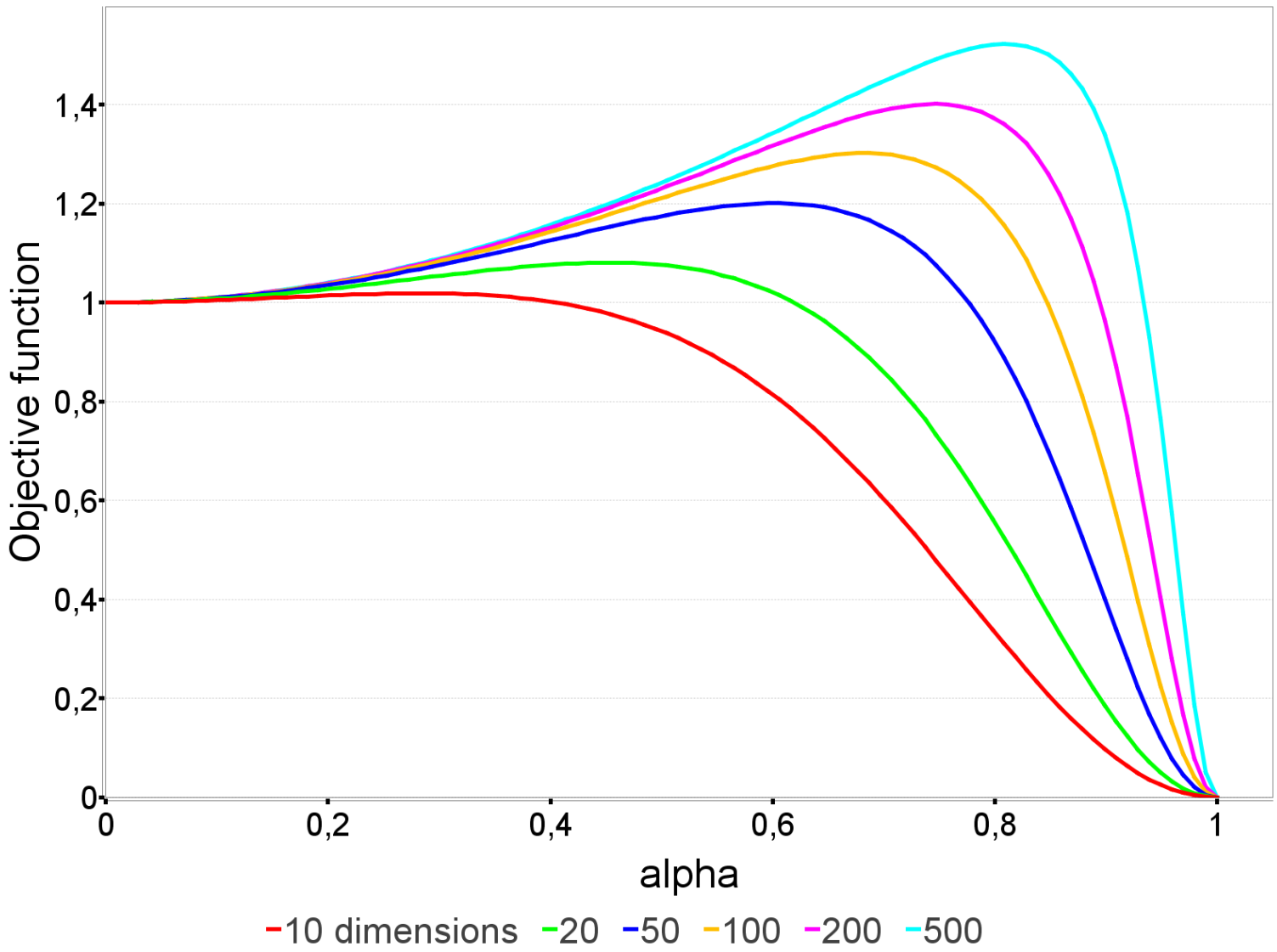
Distribution of the (absolute) distances to the mean for 100 independent normal distribution (scaled by the factor $\sqrt{100}$).



FCM problems with high-dimensional data



FCM problems with high-dimensional data



What I have not talked about . . .

- Solutions for the problems with high-dimensional data,
- Fitting to cluster of different shape and size,
- Determining the number of clusters,
- Clustering of large data sets
- . . . and many other important aspects of cluster analysis.

Thank you for your kind attention!

and thanks to

Balasubramaniam Jayaram

&

Roland Winkler