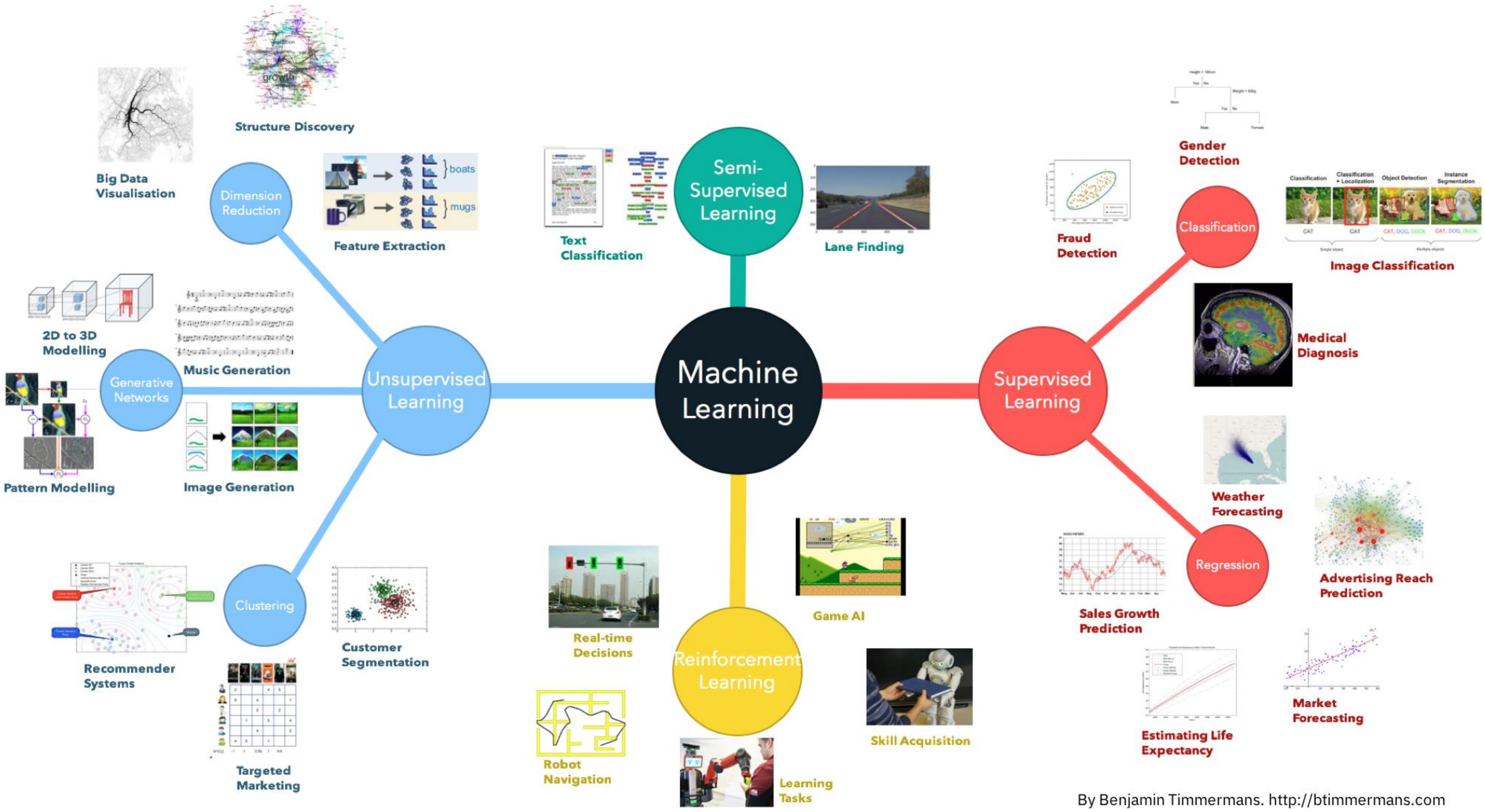
Clustering

DSAIT4020 Elements of Statistical Learning


Machine Learning
Unsupervised Learning
Semi-Supervised Learning
Supervised Learning
Reinforcement Learning
Dimension Reduction
Structure Discovery
Big Data Visualisation
Feature Extraction
boats
mugs
Generative Networks
2D to 3D Modelling
Music Generation
Pattern Modelling
Image Generation
Clustering
Recommender Systems
Targeted Marketing
Customer Segmentation
Text Classification
Lane Finding
Classification
Gender Detection
Fraud Detection
Image Classification
Medical Diagnosis
Regression
Weather Forecasting
Advertising Reach Prediction
Sales Growth Prediction
Estimating Life Expectancy
Market Forecasting
Real-time Decisions
Game AI
Robot Navigation
Learning Tasks
Skill Acquisition
By Benjamin Timmermans. http://btimmermans.com

TUDelft

Supervised learning

Objects

Data collection

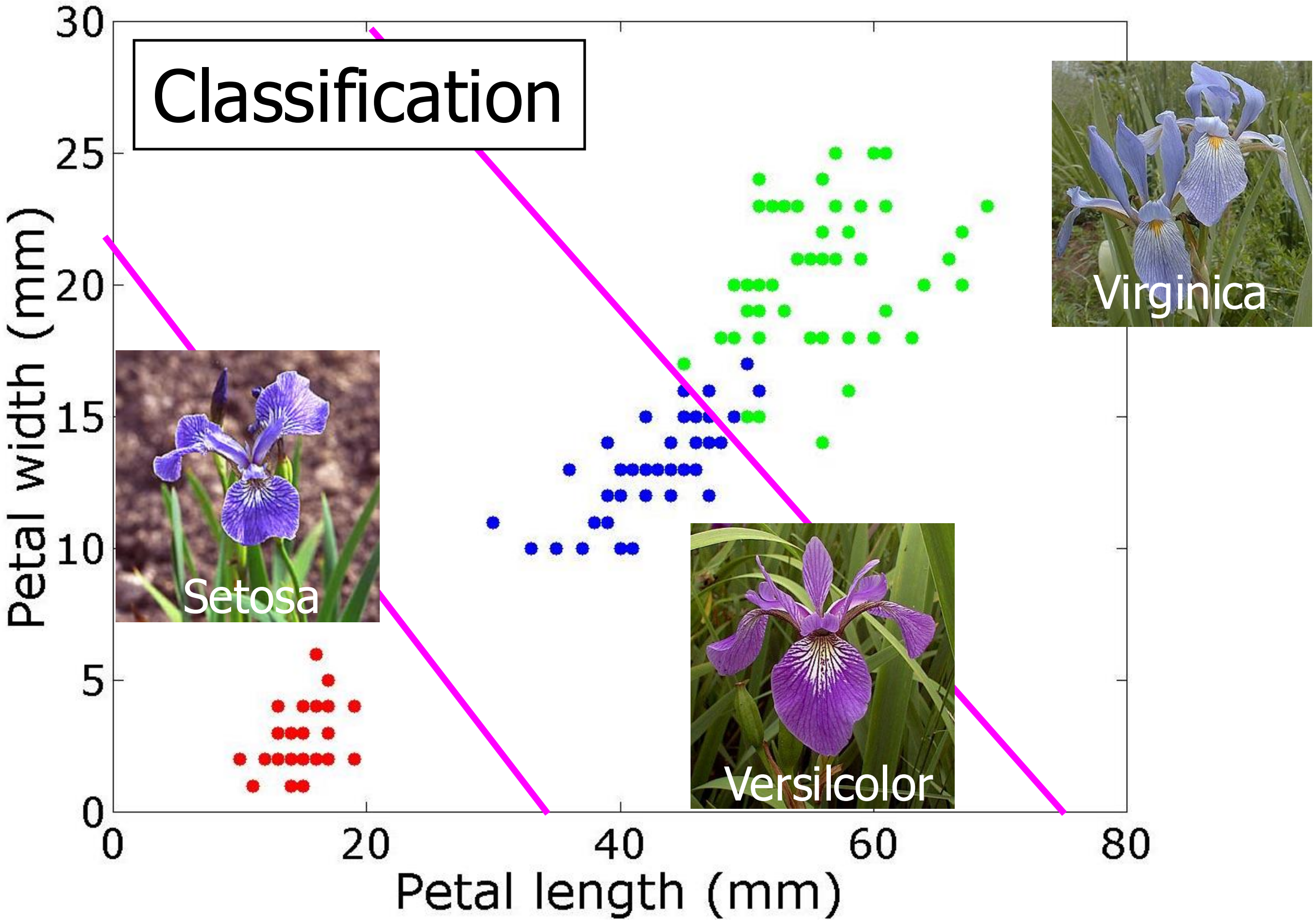
Sepal length | Sepal width | Petal length | Petal width | Class labels

$$\begin{bmatrix} 5.1, 3.5, 1.4, 0.2 \\ \vdots \\ 7.0, 3.2, 4.7, 1.4 \\ \vdots \\ 6.3, 3.3, 6.0, 2.5 \\ \vdots \end{bmatrix} \begin{bmatrix} \text{Iris - Setosa} \\ \vdots \\ \text{Iris - Versicolor} \\ \vdots \\ \text{Iris - Virginica} \\ \vdots \end{bmatrix}$$

Train classifier with **data** and **class labels**

TUDelft

Supervised learning (2)



Unsupervised learning

Objects

Data collection

Sepal length | Sepal width | Petal length | Petal width | ~~Class labels~~

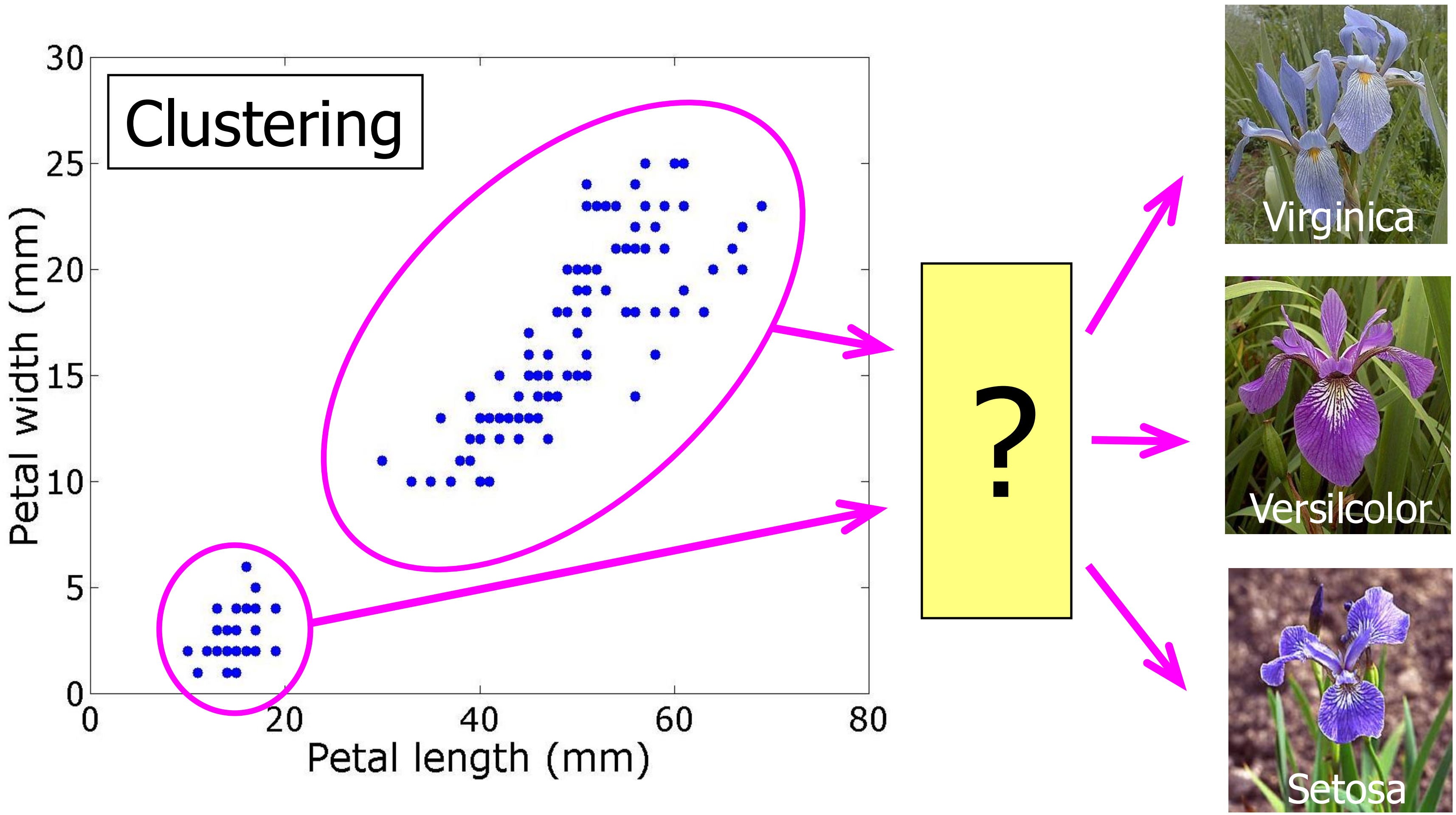
$$\begin{bmatrix} 5.1, 3.5, 1.4, 0.2 \\ \vdots \\ 7.0, 3.2, 4.7, 1.4 \\ \vdots \\ 6.3, 3.3, 6.0, 2.5 \\ \vdots \end{bmatrix} \quad \cancel{\begin{bmatrix} \text{Iris - Setosa} \\ \vdots \\ \text{Iris - Versicolor} \\ \vdots \\ \text{Iris - Virginica} \\ \vdots \end{bmatrix}}$$

Explore data **without class labels:**

PCA, clustering

TU Delft

Unsupervised learning (2)

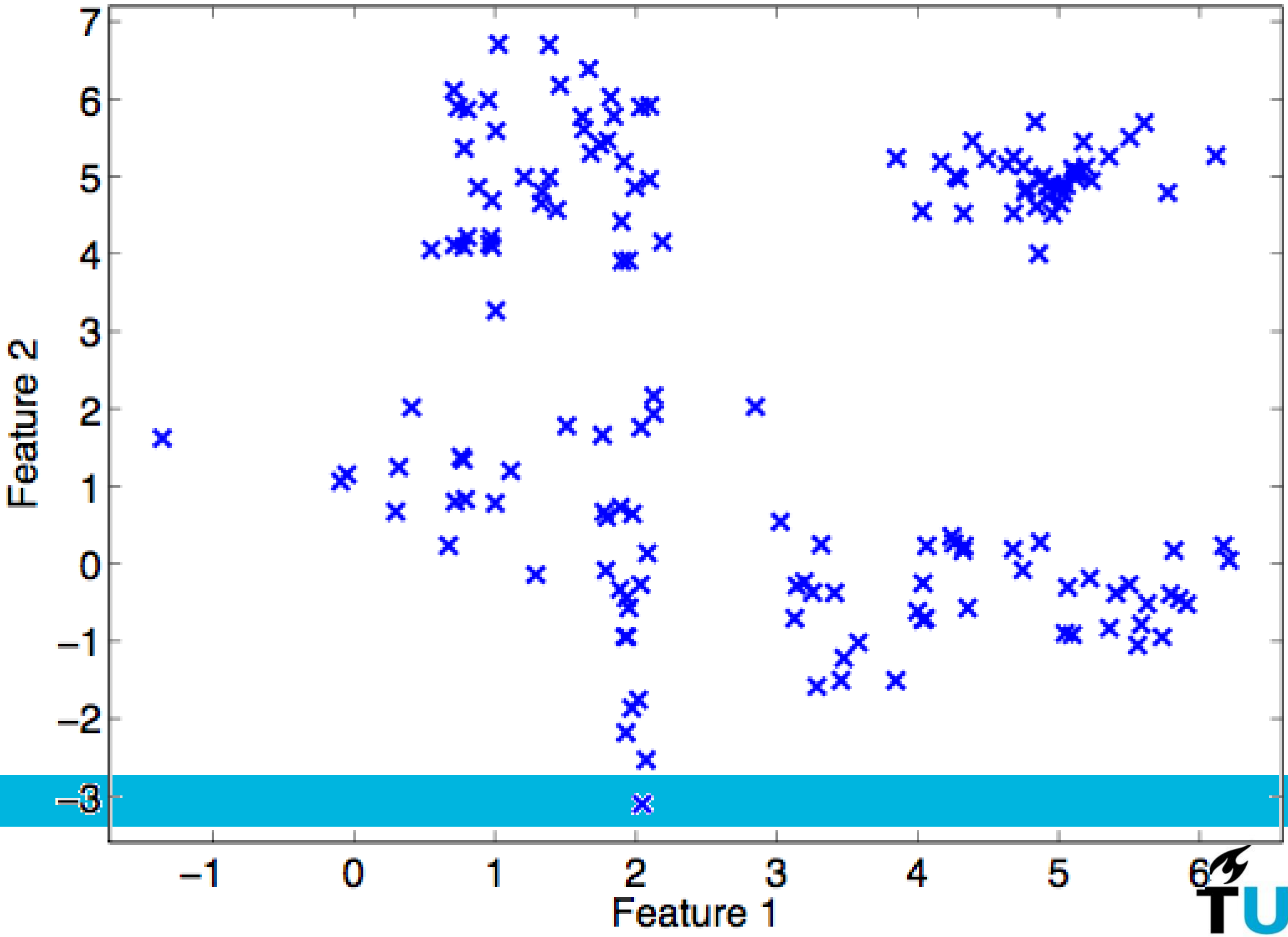


- What salient structures exist in the data?

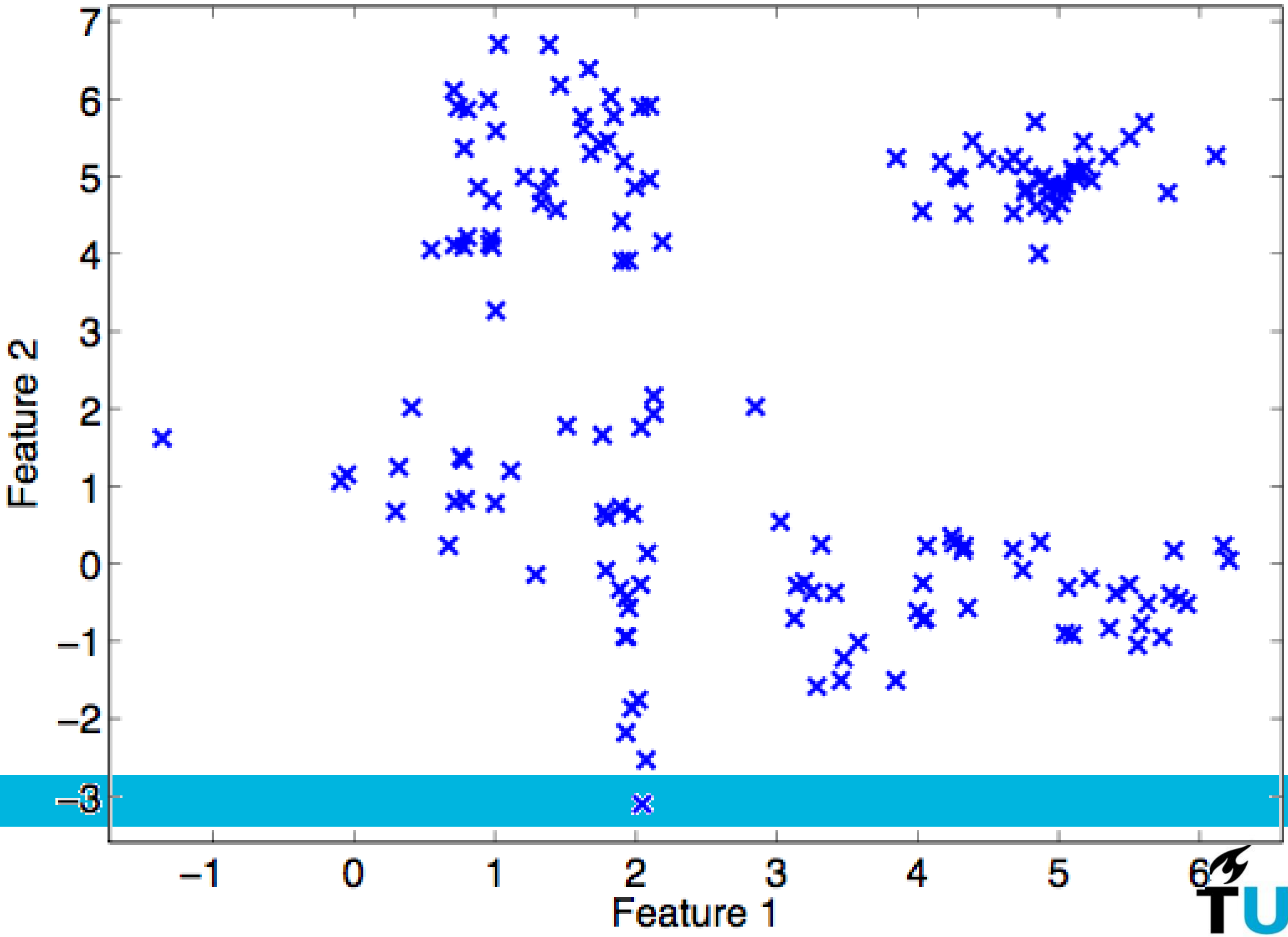
Cluster Analysis

- Grouping observations based on [dis]similarity
- E.g. data mining [exploration, searching for concepts in data]
  - Clustering species based on [genetic] similarity
  - Reducing amount of data to be analysed, helps defining concept / class
- Data reduction: selecting typical class examples
  - Multi-modal classes may be represented using typical examples
- Predicting characteristics for new data

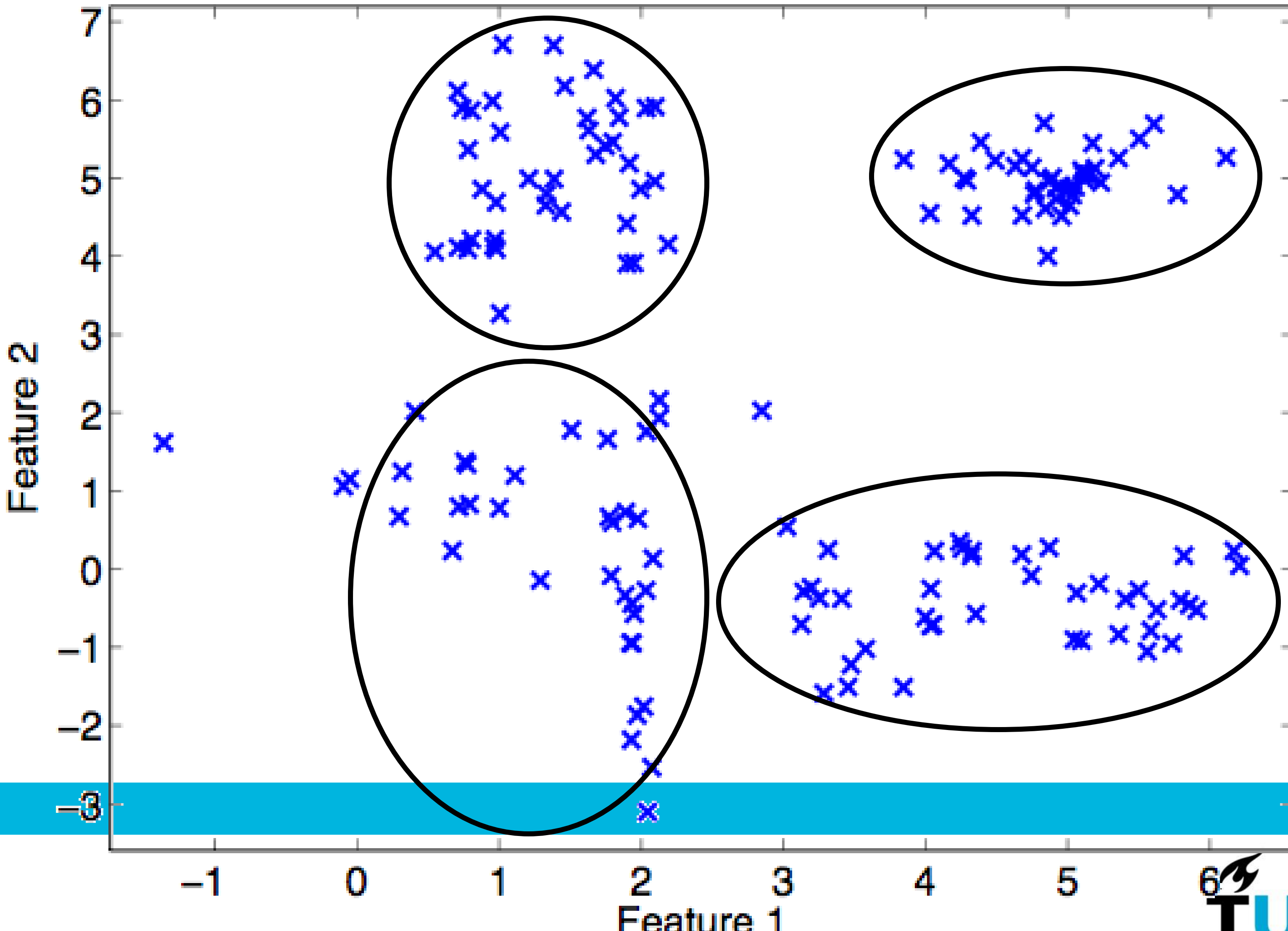
Unlabeled data: what now?



How Many Groups in Data?

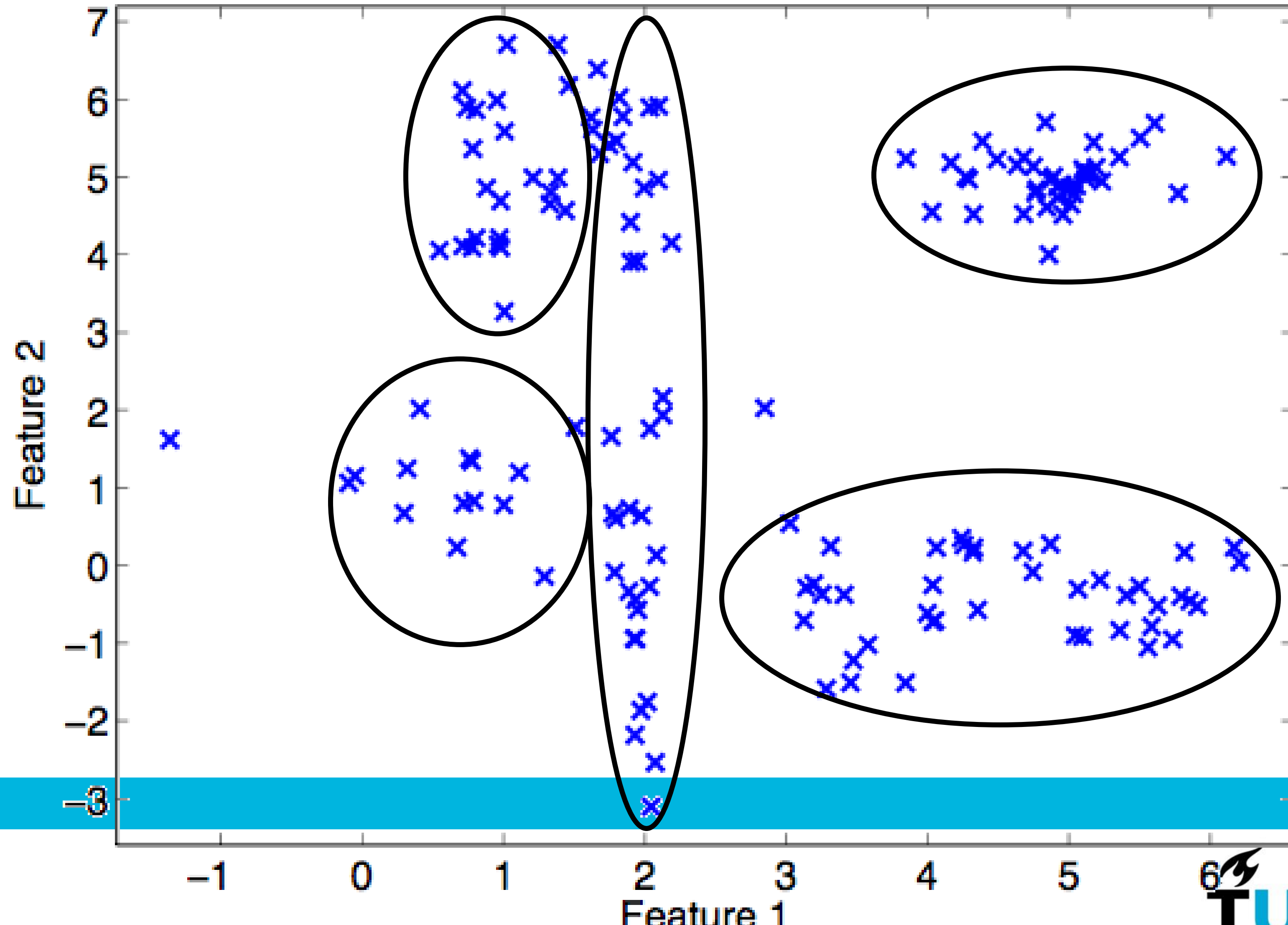


How Many Groups in Data?

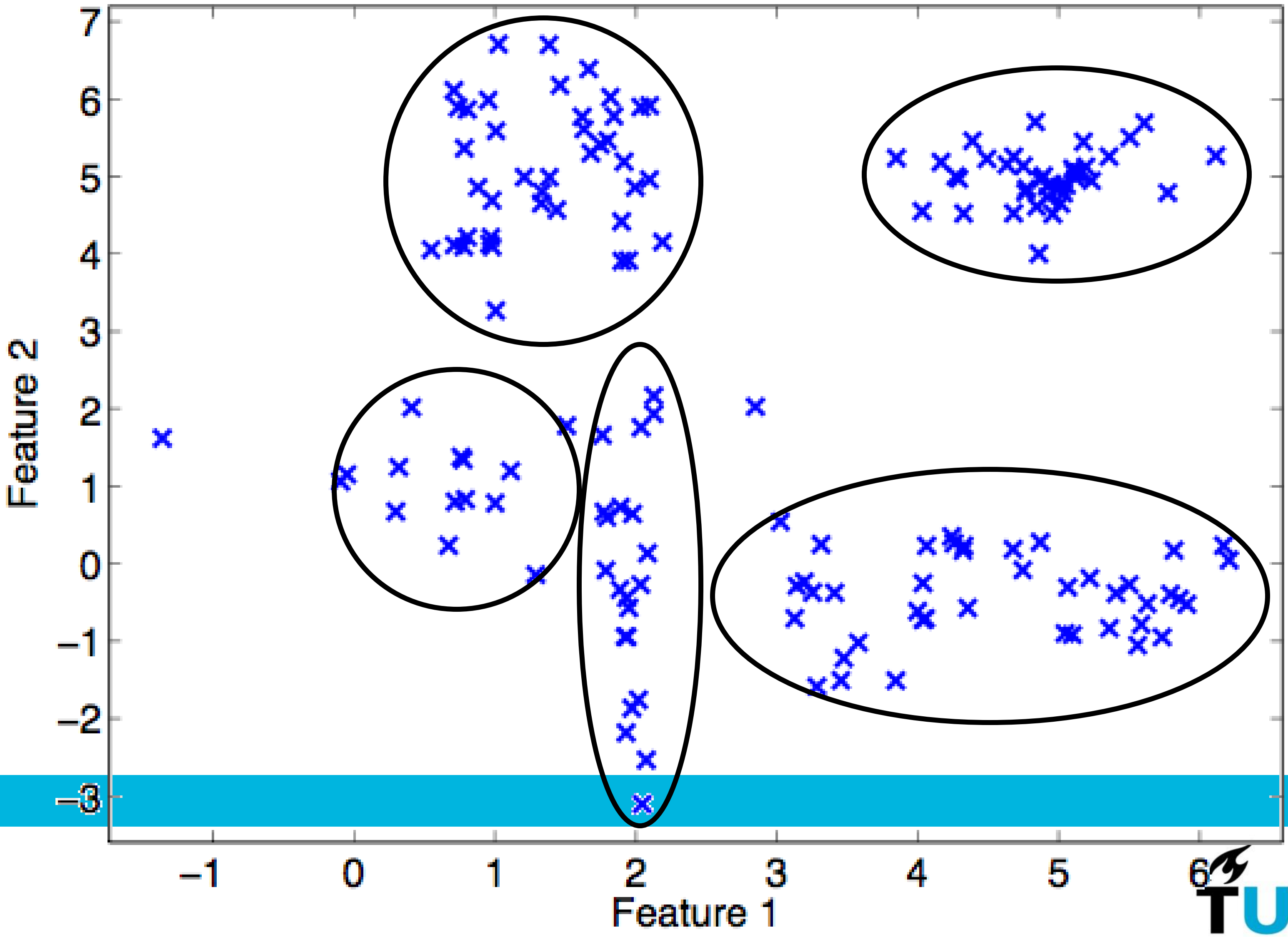


TU Delft

How Many Groups in Data?

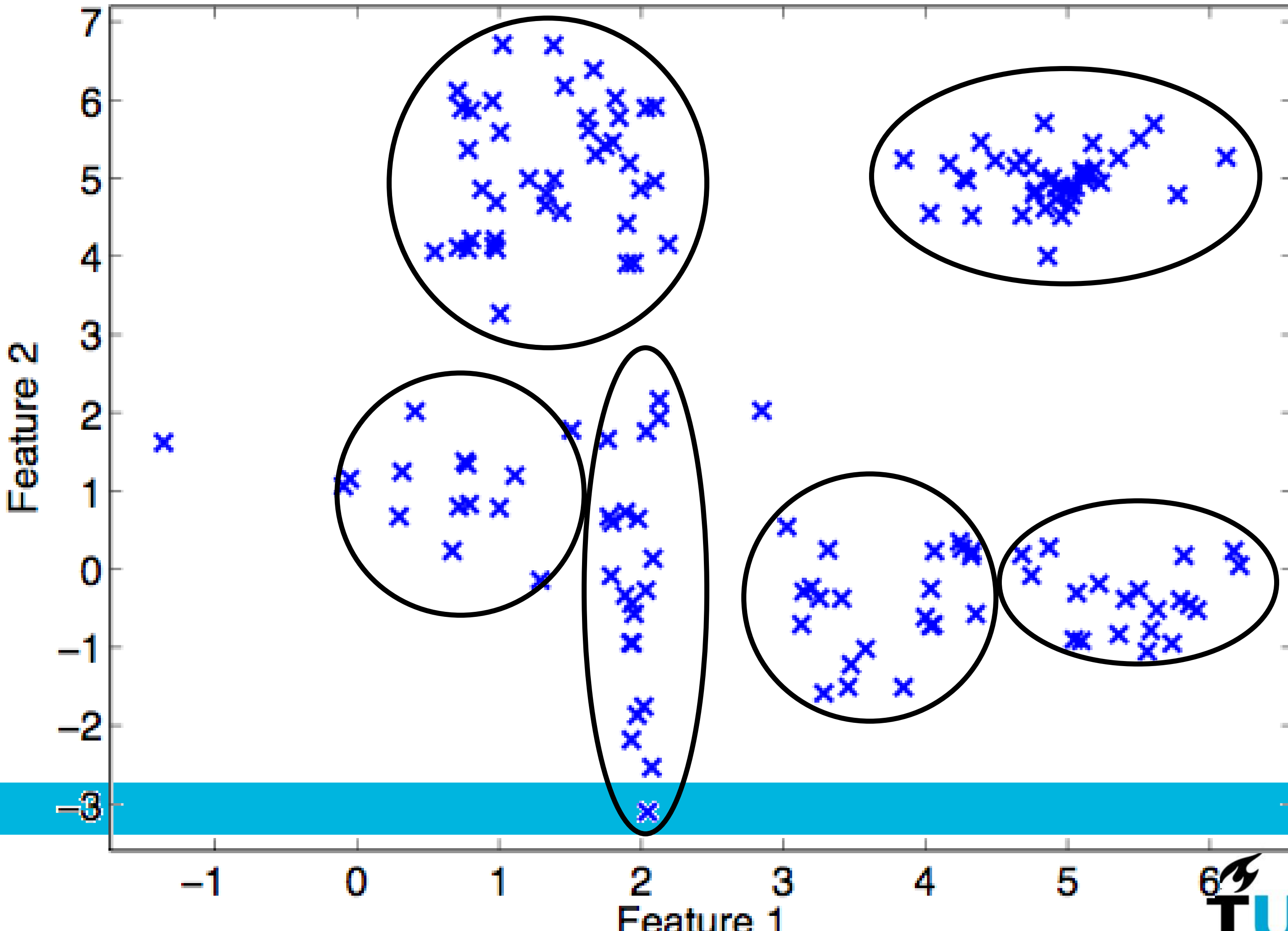


How Many Groups in Data?



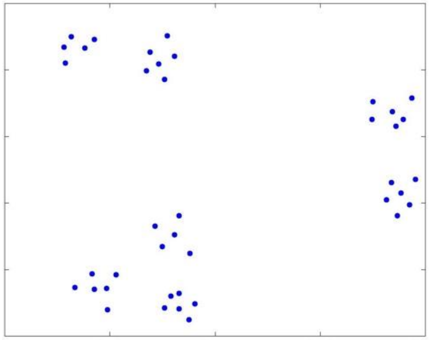
TU Delft

How Many Groups in Data?

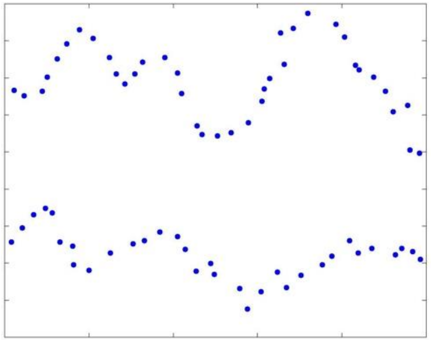


TUDelft

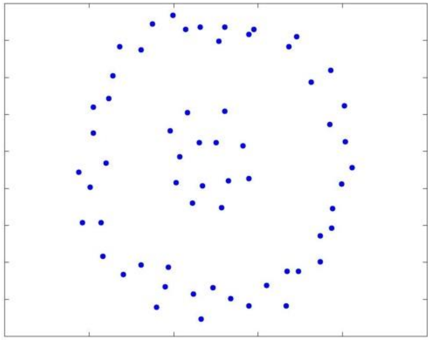
What is a cluster?



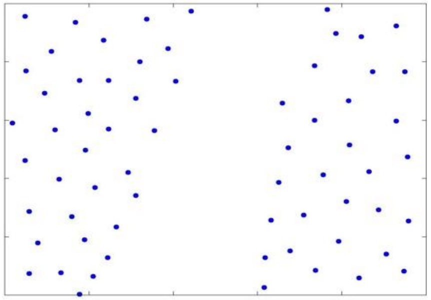
Shape: compact, convex
Separation: large



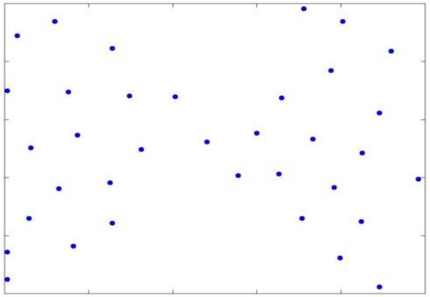
Shape: strings
Separation: large?



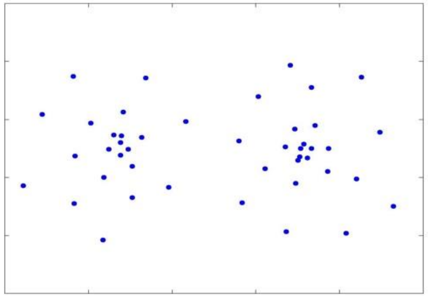
Shape: convex and circular
Separation: large?



Shape: ?
Separation: large?



Shape: loose, convex
Separation: small



Shape: loose, convex
Separation: small

What is a cluster?

- Clustering: finding natural groups in data...
  - which themselves are far apart
  - in which objects are close together

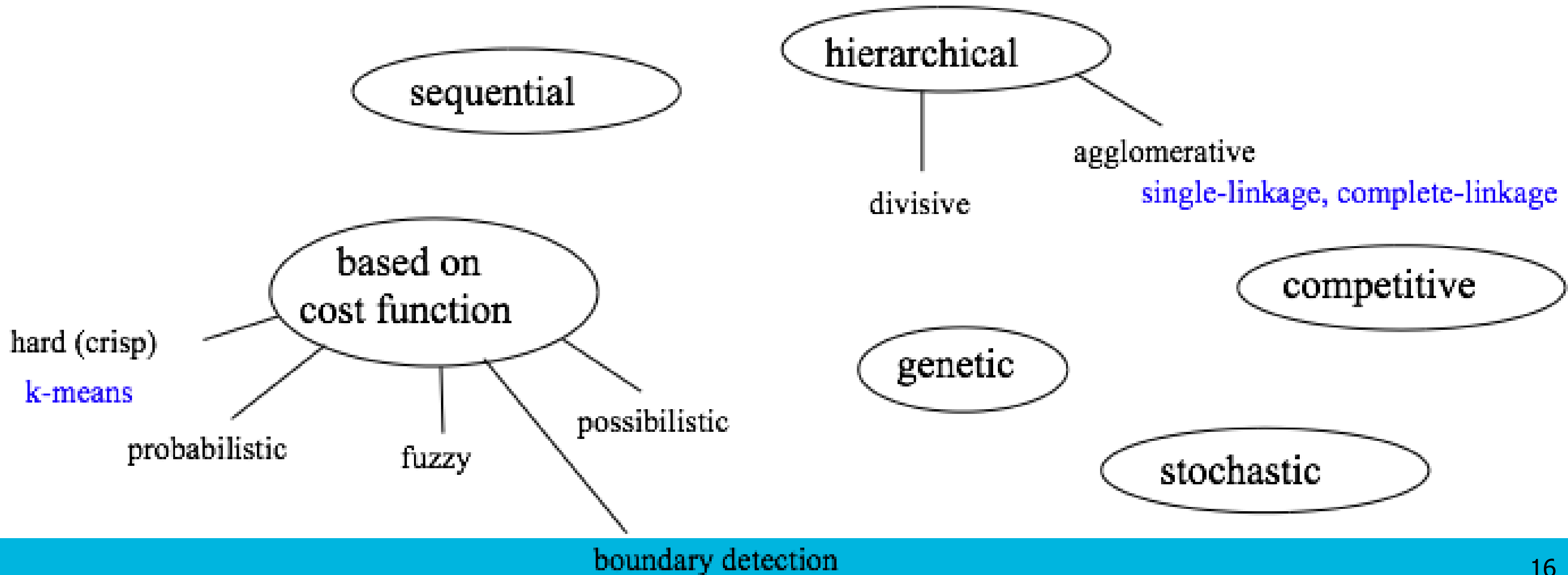
- Define what is “far apart” and “close together”:
  - Need a distance measure or dissimilarity measure
  - This measure should capture what we think is important for the grouping
  - The choice for a certain distance measure is crucial!

- **There is no such thing as *<u>the</u> objective clustering***

TU Delft

Clustering Clustering Methods

- Very large field, huge number of methods
  - See for example **Theodoridis and Koutroumbas, Pattern Recognition, 2003**
    - More than 240 pages overview of cluster analysis



TUDelft

Chapters 11-16 from the book...

- In literature, an almost infinite number of methods is proposed
- The book tries to cover many of them
- We will discuss the most intuitive, and most used, clustering methods
- Ignore sections 12.3, 12.4, 12.5, 12.6, 12.7
- Ignore pages 661-692
- Ignore 14.3, 14.4, 14.6
- Ignore 15.3 till 15.12 (expect maybe 15.8)

TU Delft

Agenda

- Clustering measures

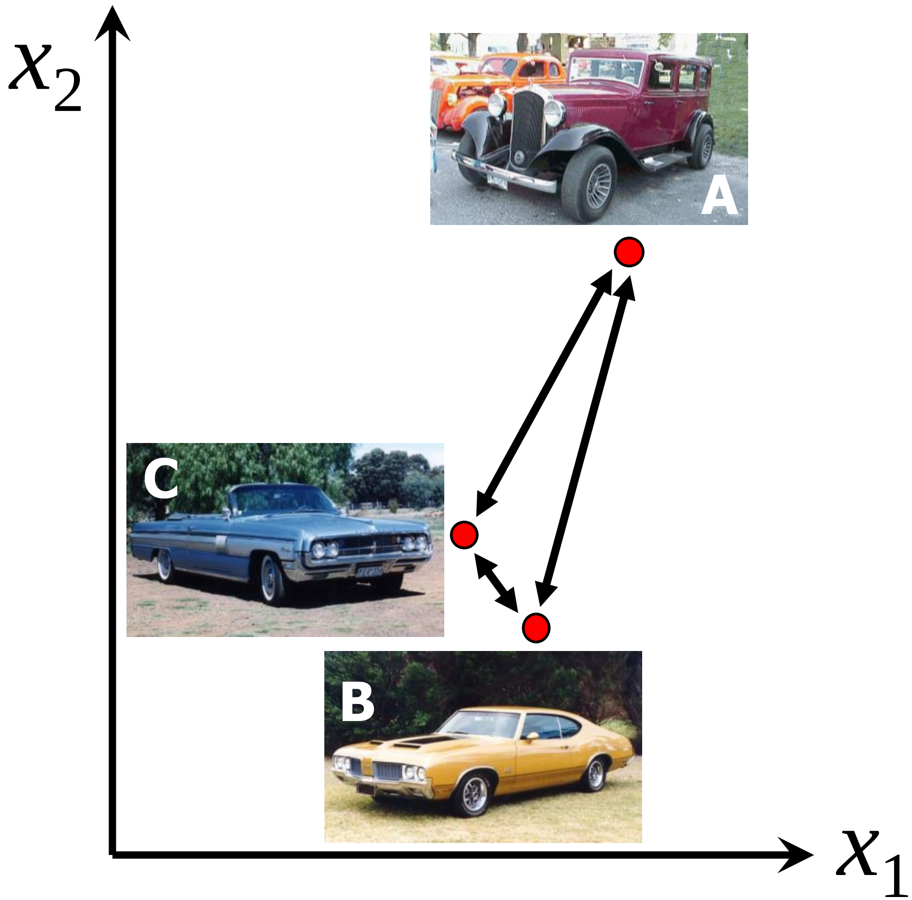
Yes, you also can do soft assignments…

- Clustering methods (hard assignments)
  - Hierarchical clustering
  - *k*-means clustering

- Cluster validation
  - Fusion graphs
  - The Davies-Bouldin Index

TU Delft

Dissimilarity measures



$$
\boldsymbol{D} = \begin{bmatrix} 0 & d(\mathbf{A},\mathbf{B}) & d(\mathbf{A},\mathbf{C}) \\ & 0 & d(\mathbf{B},\mathbf{C}) \\ & & 0 \end{bmatrix}
= \begin{bmatrix} 0 & 10 & 11 \\ & 0 & 2 \\ & & 0 \end{bmatrix}
$$

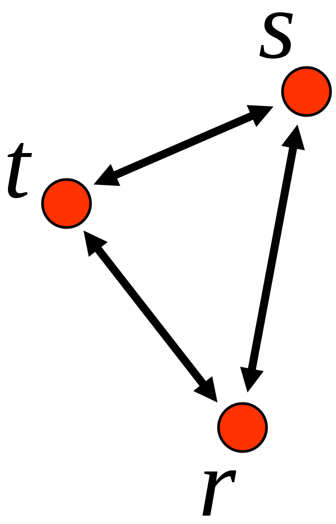
Dissimilarity measures

- Let $d(r,s)$ be the dissimilarity between objects r and s
- Formally, dissimilarity measures should satisfy

$$d(r,s) \geq 0, \forall\, r,s$$

$$d(r,r) = 0, \forall\, r$$

$$d(r,s) = d(s,r), \forall\, r,s$$



- If in addition, the triangle inequality holds, the measure is a metric

$$d(r,t) + d(t,s) \geq d(r,s), \forall\, r,s,t$$

- **Most often used: Euclidean distance (metric)**

Distance measure

- Define a distance between objects:
- Euclidean: $$d(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{l}(x_i - y_i)^2}$$
- City-block $$d(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{l}|x_i - y_i|$$
- ℓ_p-metric $$d_p(\mathbf{x},\mathbf{y}) = \left(\sum_{i=1}^{l}|x_i - y_i|^p\right)^{1/p}$$

More similarity measures

- Cosine similarity

$$s_{cos}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^T \mathbf{y}}{||\mathbf{x}|| \, ||\mathbf{y}||}$$

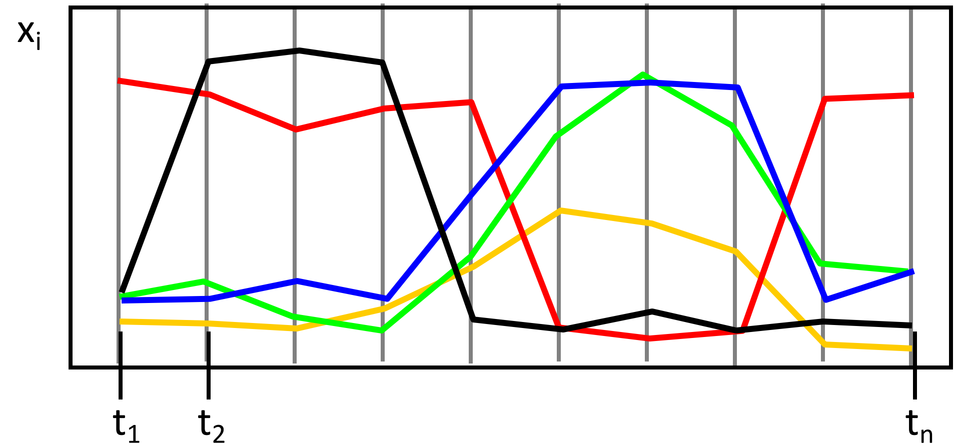
- Pearson's correlation coefficient

$$r_{Pearson}(\mathbf{x}, \mathbf{y}) = \frac{(\mathbf{x} - \mu_x)^T (\mathbf{y} - \mu_y)}{||\mathbf{x} - \mu_x|| \, ||\mathbf{y} - \mu_y||}$$

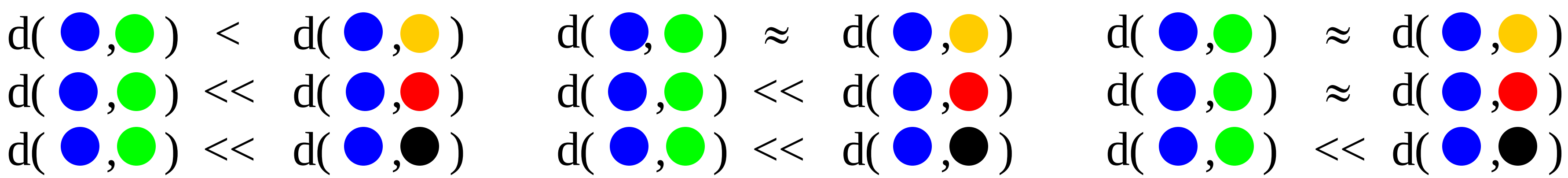
- and more... (for discrete features, mixed features, categorical features, ...)

Using different measures

- Example:
  time series data
  (gene expression)

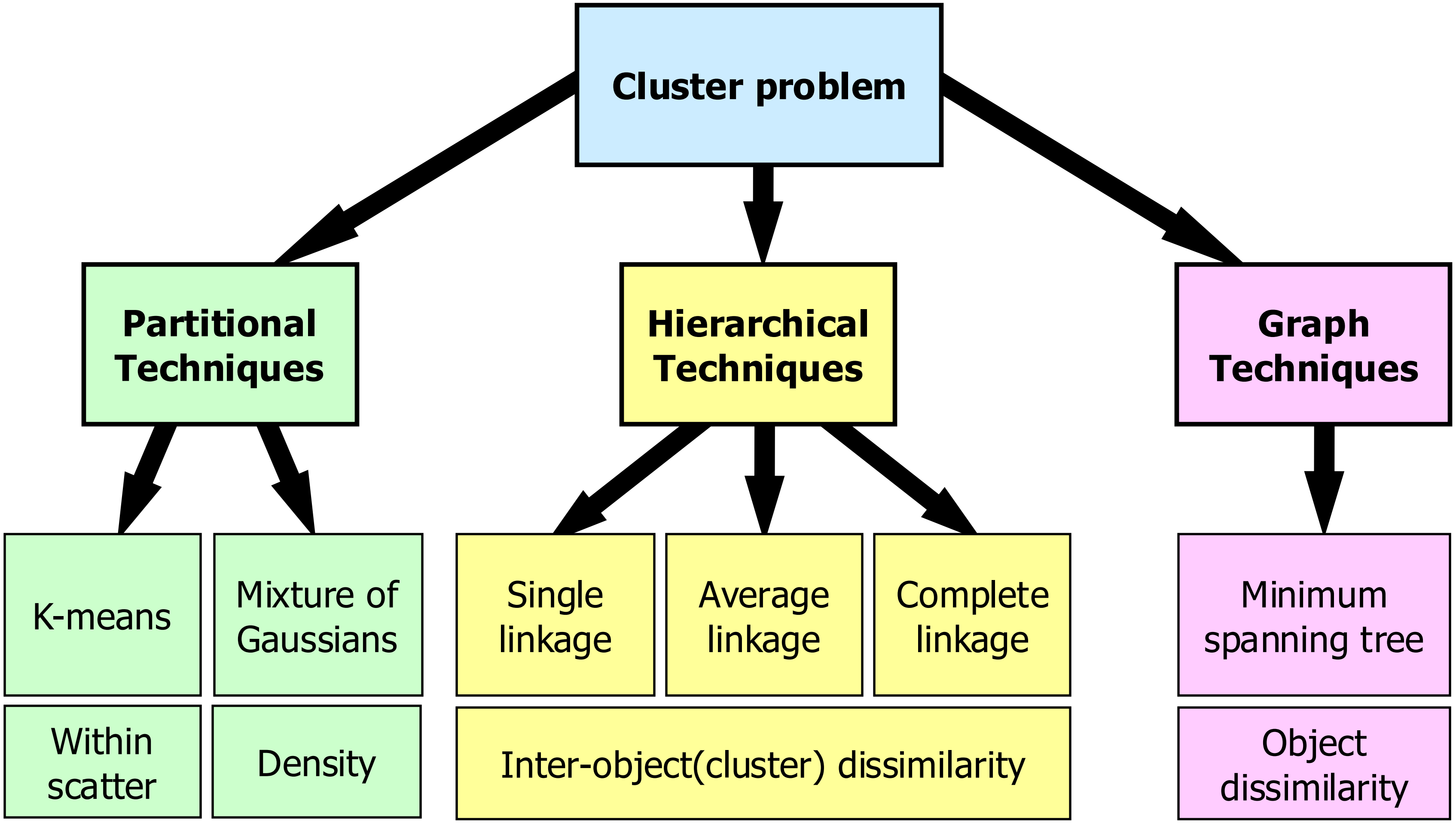


Euclidean distance	Pearson correlation	Absolute correlation
match exact shape	ignore amplitude	ignore amplitude & sign
$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{t=1}^{n} (x_{i,t} - x_{j,t})^2$	$1 - \rho_{ij}$	$1 - \lvert \rho_{ij} \rvert$



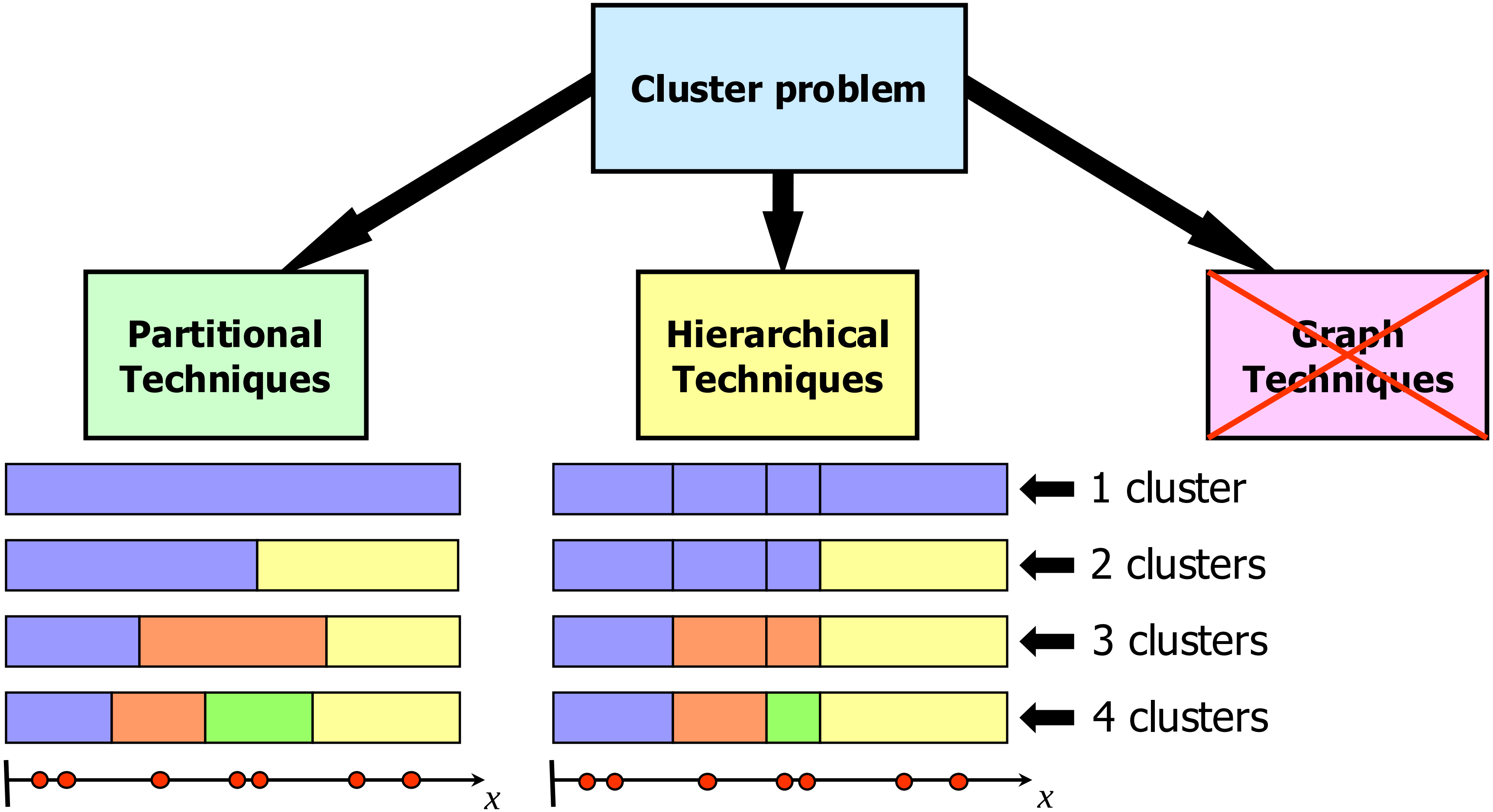
$$\rho_{ij} = \sum_{t=1}^{n} (x_{i,t} - \mu_i)(x_{j,t} - \mu_j) \Big/ \sigma_i \sigma_j$$

Clustering techniques



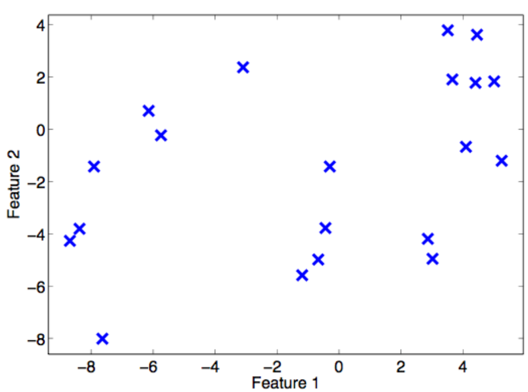
TU Delft

Clustering techniques

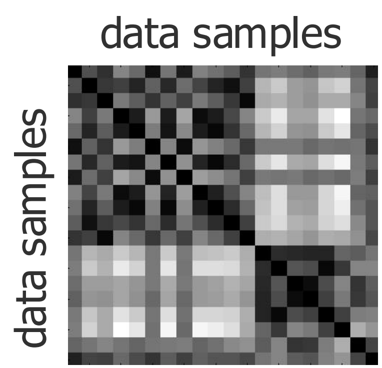


Agglomerative Hierarchical Clustering

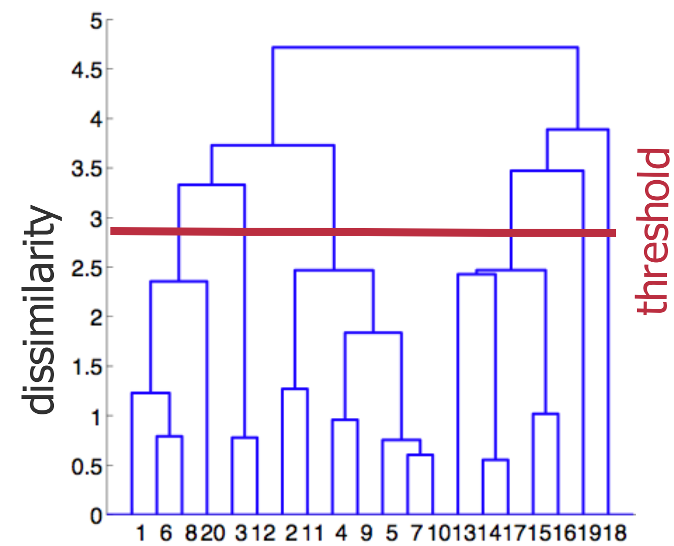
- **Starting from individual observations**, produce sequence of clusterings of increasing size
- At **each level**, **two clusters** chosen by criterion are merged



2D scatter plot of data



dissimilarity matrix



dendrogram

Agglomerative Hierarchical Clustering

1. Determine distances between all clusters
2. Merge clusters that are **closest**
3. IF #clusters>1 THEN GOTO 1

- Which clusters to start with?
- What is the distance between clusters?
- Final number of clusters?

Different Merging Rules

- Two nearest objects in the clusters : single linkage

$$g(R,S) = \min_{ij}\{d(\mathbf{x}_i,\mathbf{x}_j) : \mathbf{x}_i \in R, \mathbf{x}_j \in S\}$$

- Two most remote objects in the clusters : complete linkage

$$g(R,S) = \max_{ij}\{d(\mathbf{x}_i,\mathbf{x}_j) : \mathbf{x}_i \in R, \mathbf{x}_j \in S\}$$

- Cluster centers : average linkage

$$g(R,S) = \frac{1}{|R||S|}\sum_{ij}\{d(\mathbf{x}_i,\mathbf{x}_j) : \mathbf{x}_i \in R, \mathbf{x}_j \in S\}$$

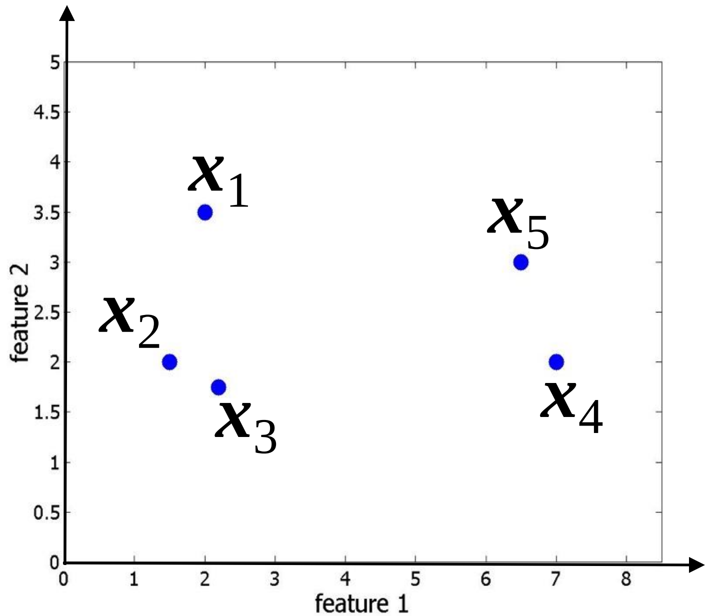
Hierarchical clustering

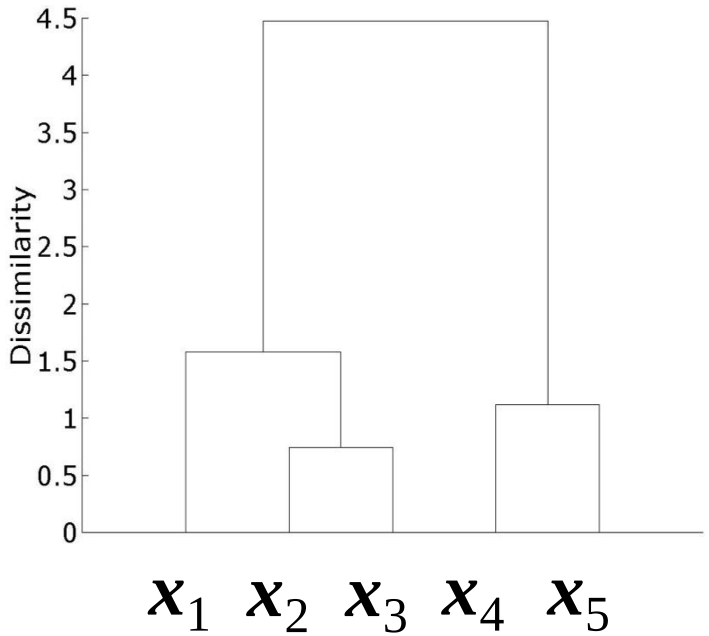
Input:

- dataset, $\boldsymbol{X}$: $[n \times p]$, or directly:
- dissimilarity matrix, $\boldsymbol{D}$: $[n \times n]$
- linkage type

Output:

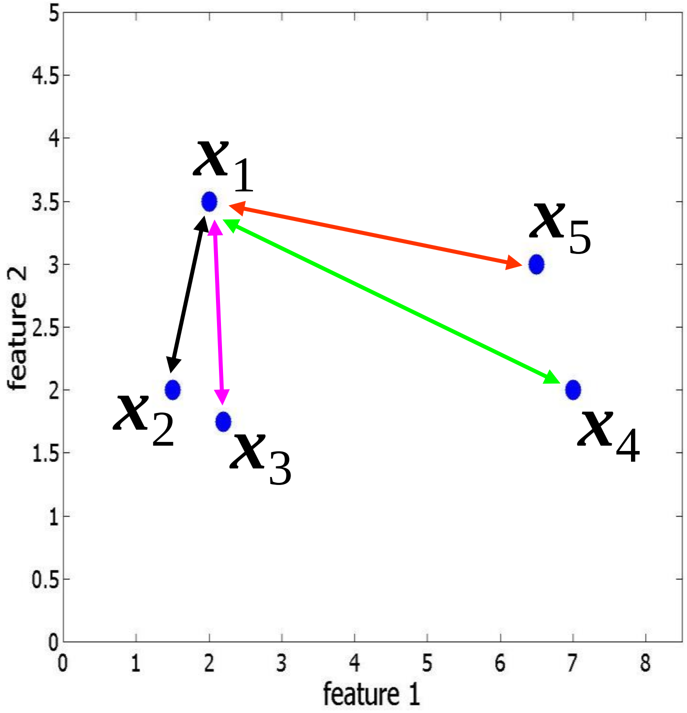
- dendrogram





Hierarchical clustering (2)

Dataset

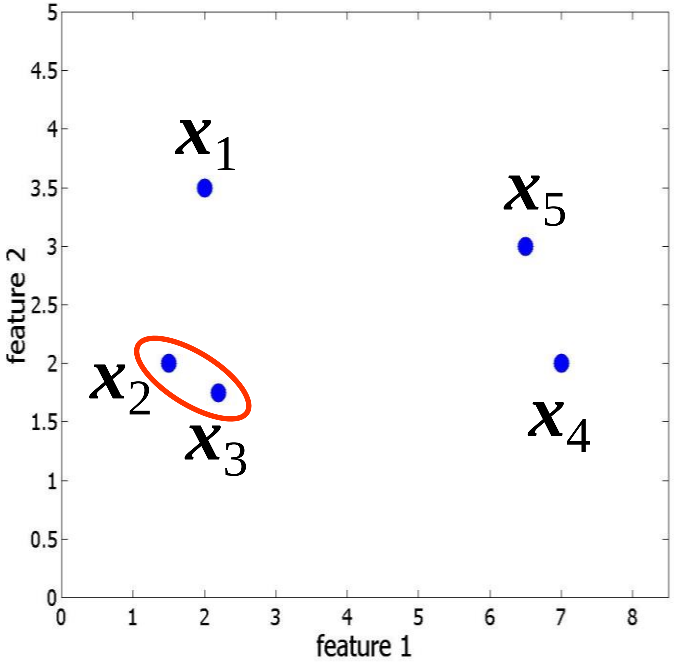


Euclidean distance matrix, $\boldsymbol{D}$

	x_1	x_2	x_3	x_4	x_5
x_1	0.00	$\boxed{1.58}$	$\boxed{1.76}$	$\boxed{5.22}$	$\boxed{4.53}$
x_2		0.00	0.74	5.50	5.10
x_3			0.00	4.81	4.48
x_4				0.00	1.12
x_5					0.00

Hierarchical clustering (3)

- **Step 1:**
  Find the most similar pair of objects: $\min_{(i,j)}\{d(i,j)\} = d(2,3)$

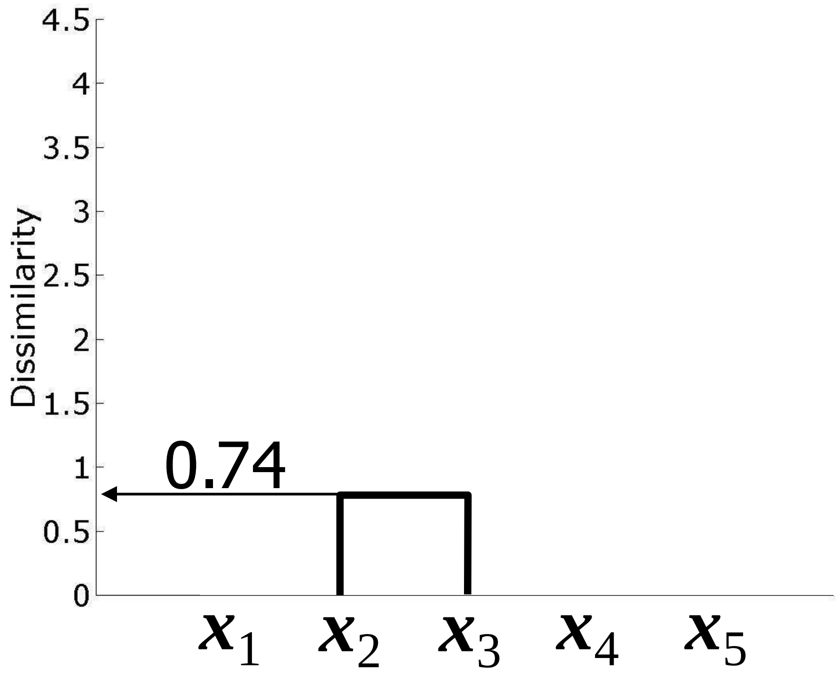


	x_1	x_2	x_3	x_4	x_5
x_1	0.00	1.58	1.76	5.22	4.53
x_2		0.00	$\boxed{0.74}$	5.50	5.10
x_3			0.00	4.81	4.48
x_4				0.00	1.12
x_5					0.00

TU Delft

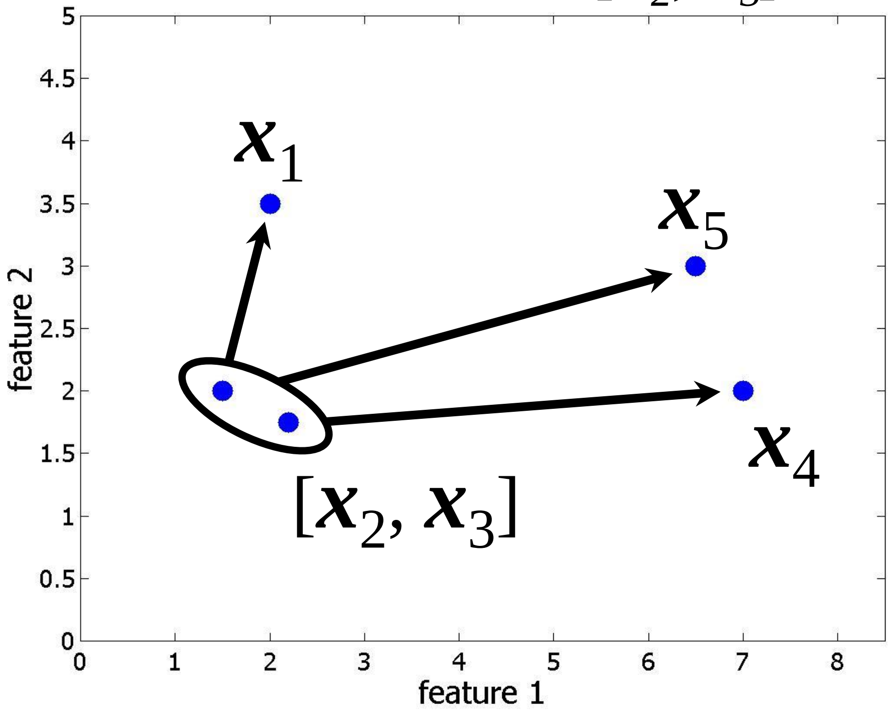
Hierarchical clustering (4)

- **Step 2:**
  Merge x_2 and x_3 into a single object, $[x_2, x_3]$;



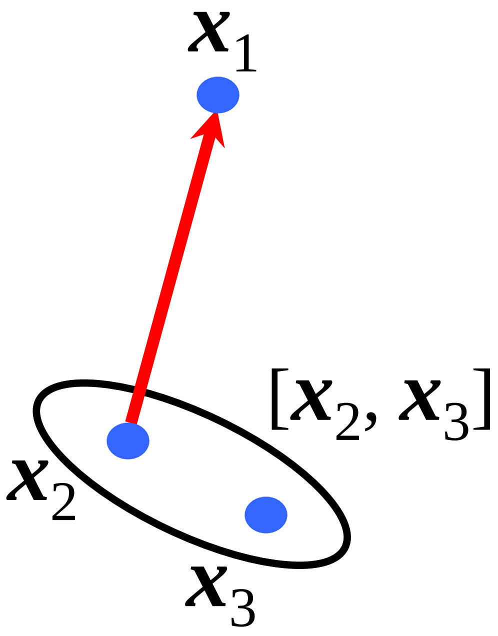
Hierarchical clustering (5)

- **Step 3:**
  Recompute $\boldsymbol{D}$ –
  what is the distance between $[\mathbf{x}_2, \mathbf{x}_3]$ *and the rest?*



Hierarchical clustering (6)

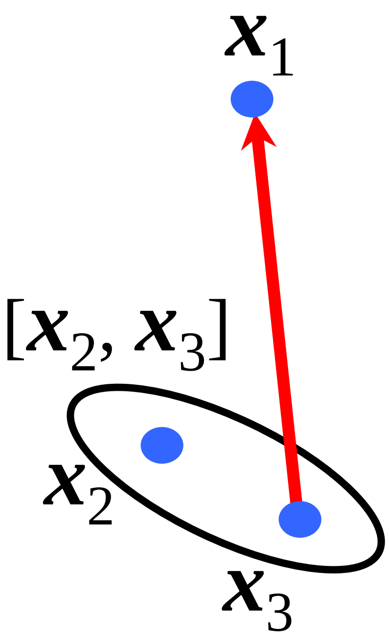
- **Step 3:**
  Recompute $\boldsymbol{D}$ –
  single linkage: $d([\boldsymbol{x}_2,\boldsymbol{x}_3],\boldsymbol{x}_1) = \min(d(\boldsymbol{x}_1,\boldsymbol{x}_2),d(\boldsymbol{x}_1,\boldsymbol{x}_3))$



TU Delft

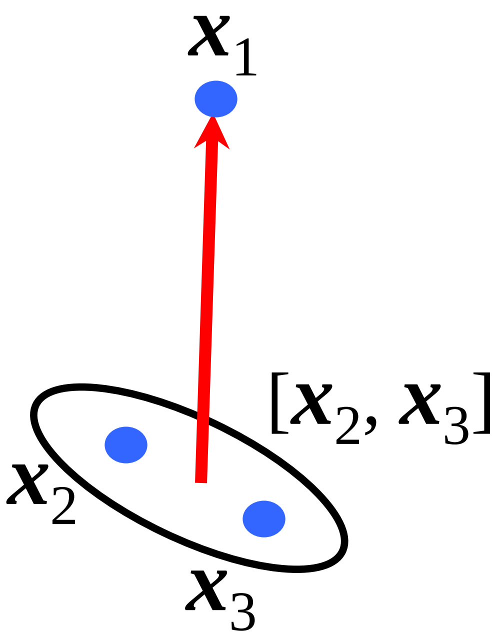
Hierarchical clustering (7)

- **Step 3:**
  Recompute $\boldsymbol{D}$ –
  complete linkage: $d([\boldsymbol{x}_2,\boldsymbol{x}_3],\boldsymbol{x}_1) = \max(d(\boldsymbol{x}_1,\boldsymbol{x}_2),d(\boldsymbol{x}_1,\boldsymbol{x}_3))$



Hierarchical clustering (8)

- **Step 3:**
  Recompute $\boldsymbol{D}$ –
  average linkage: $d([\boldsymbol{x}_2,\boldsymbol{x}_3],\boldsymbol{x}_1) = \mathrm{mean}(d(\boldsymbol{x}_1,\boldsymbol{x}_2),d(\boldsymbol{x}_1,\boldsymbol{x}_3))$



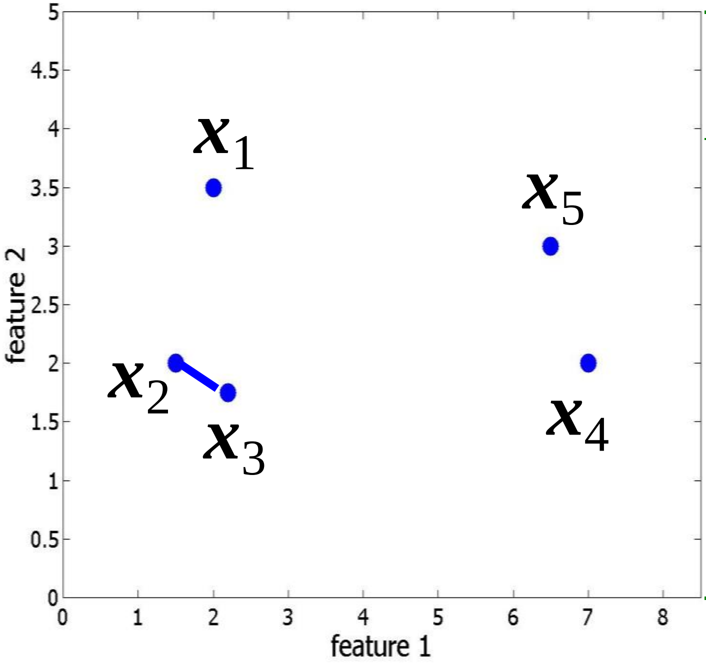
Hierarchical clustering (9)

- **Step 3:**
  Recompute D – **single linkage:**

	x_1	$[x_2,x_3]$	x_4	x_5
x_1	0.00	1.58	5.22	4.53
$[x_2,x_3]$		0.00	4.81	4.48
x_4			0.00	1.12
x_5				0.00

Hierarchical clustering (10)

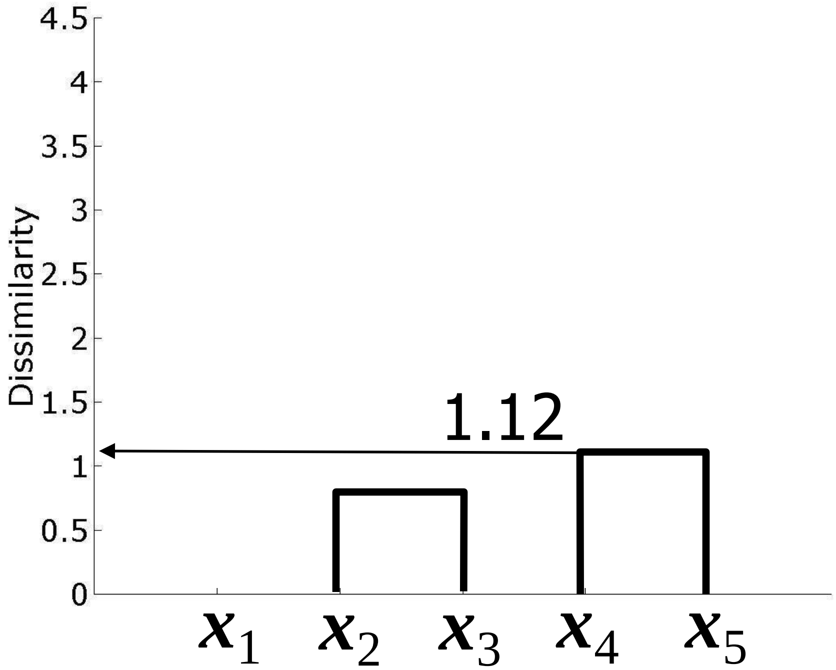
- **Repeat, step 1:**
  Find the most similar pair of objects: $\min_{(i,j)}\{d(i,j)\} = d(4,5)$



	x_1	$[x_2,x_3]$	x_4	x_5
x_1	0.00	1.58	5.22	4.53
$[x_2,x_3]$	1.58	0.00	4.81	4.48
x_4	5.22	4.81	0.00	$\boxed{1.12}$
x_5	4.53	4.48	1.12	0.00

Hierarchical clustering (11)

- **Repeat, step 2:**
  Merge x_4 and x_5 into a single object, $[x_4, x_5]$;



Hierarchical clustering (12)

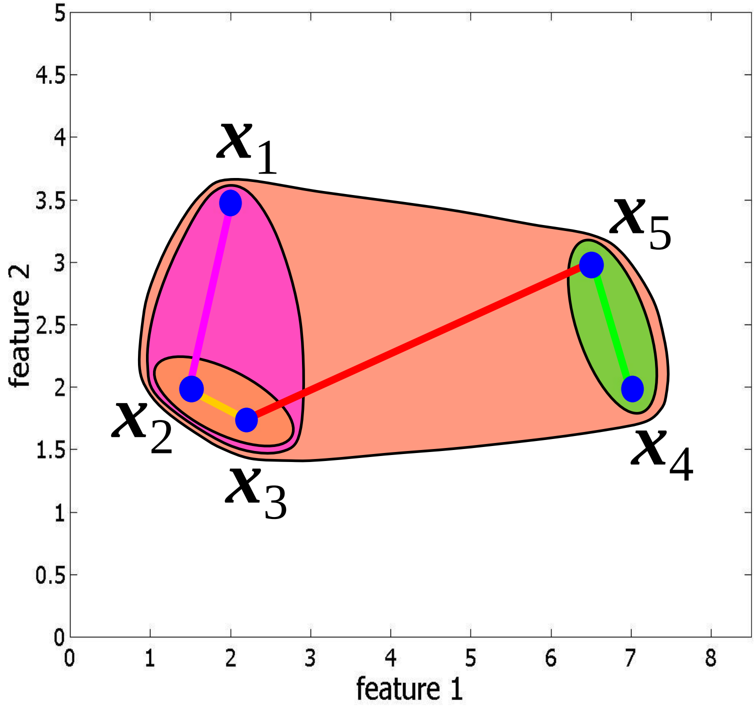
- **Repeat, step 3:**
  Recompute $\boldsymbol{D}$ (single linkage):

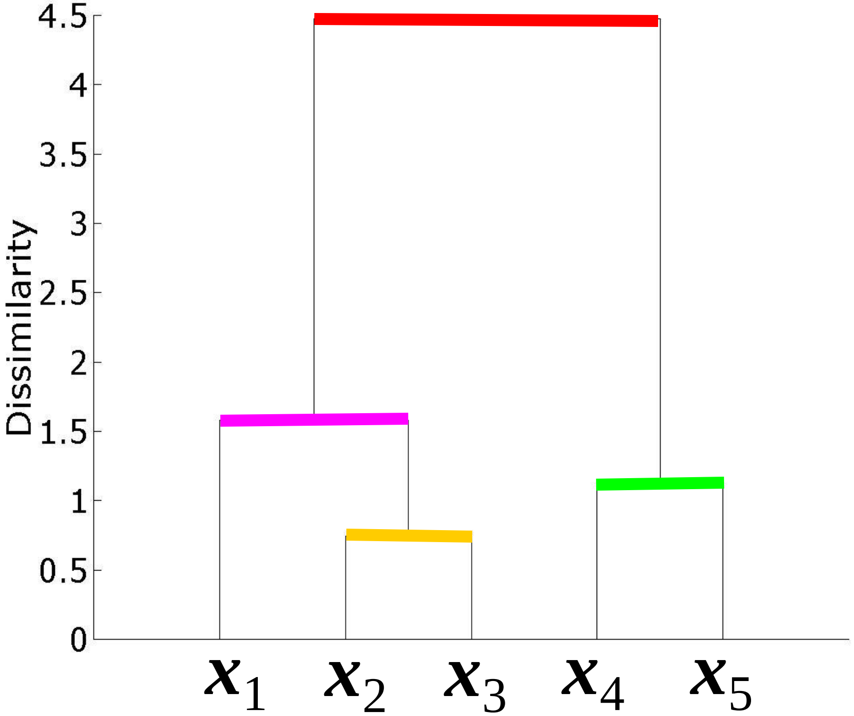
	x_1	$[x_2,x_3]$	$[x_4,x_5]$
x_1	0.00	1.58	4.53
$[x_2,x_3]$		0.00	4.48
$[x_4,x_5]$			0.00

TU Delft

Hierarchical clustering (13)

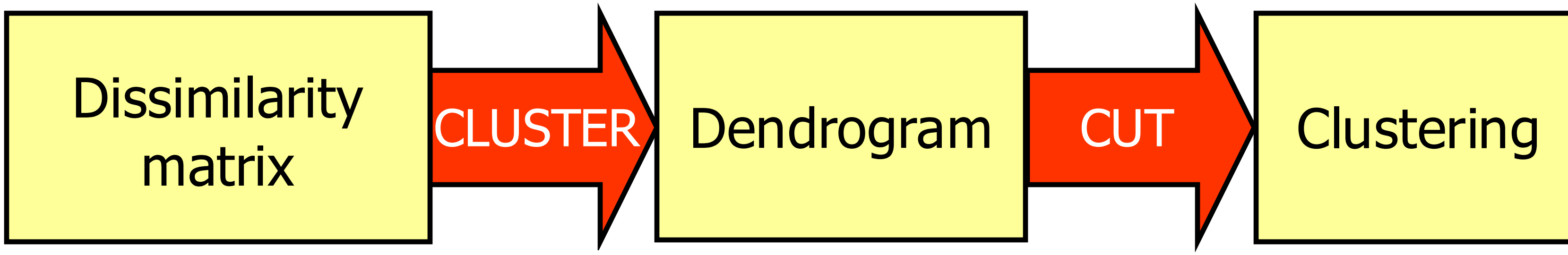
- Repeat steps 1-3 until a single cluster remains...

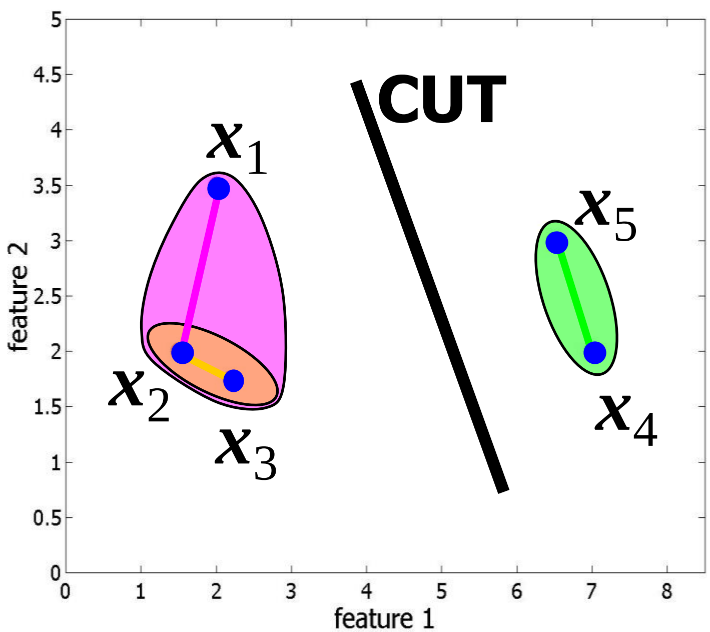


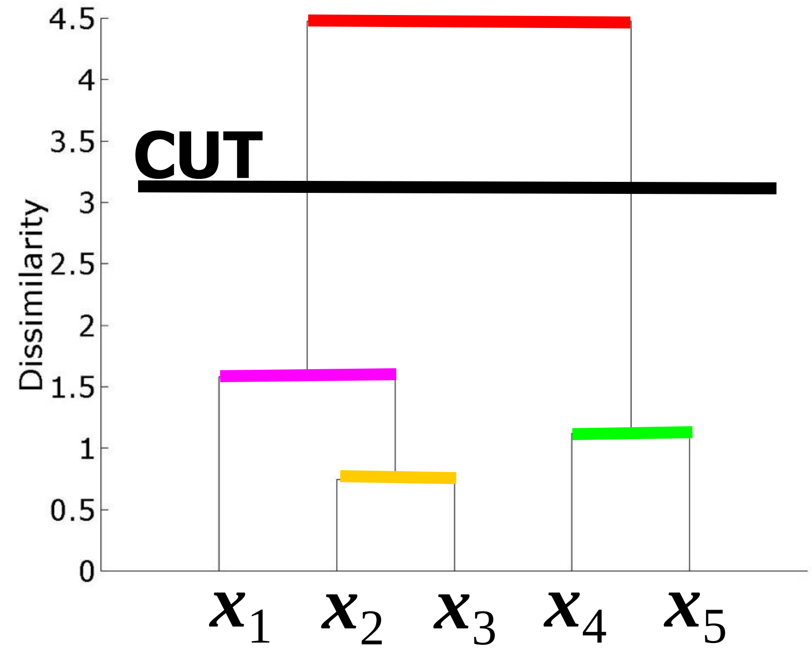


TU Delft

Hierarchical clustering (14)



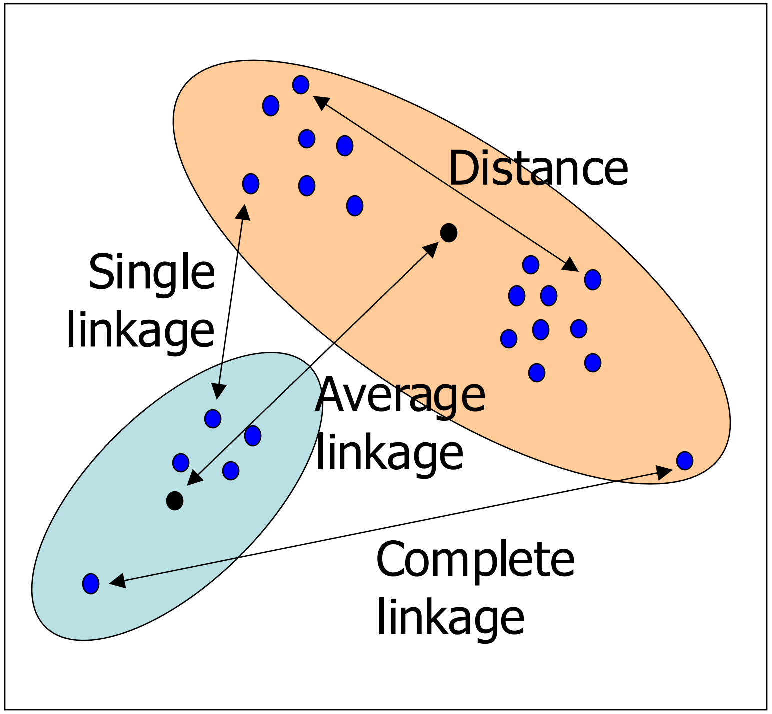




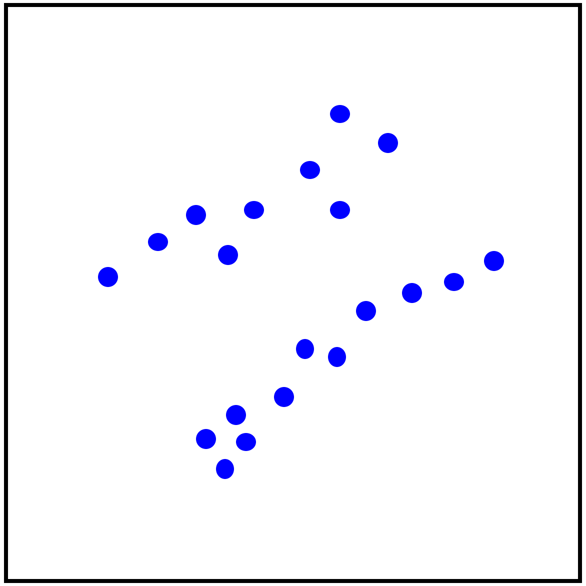
Hierarchical clustering (15)

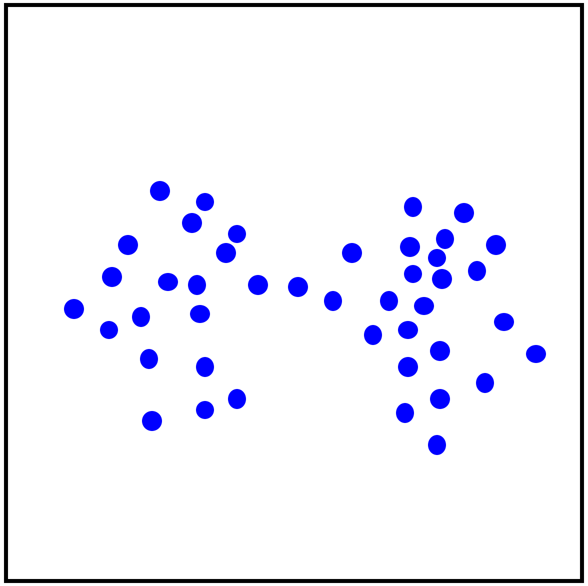
- Hierarchical clustering: repeatedly group closest clusters

- Important choices:
  - *Distance measure* between objects: Euclidean, correlation, ...
  - *Linkage* between clusters: single, average, complete



Linkage and cluster shape

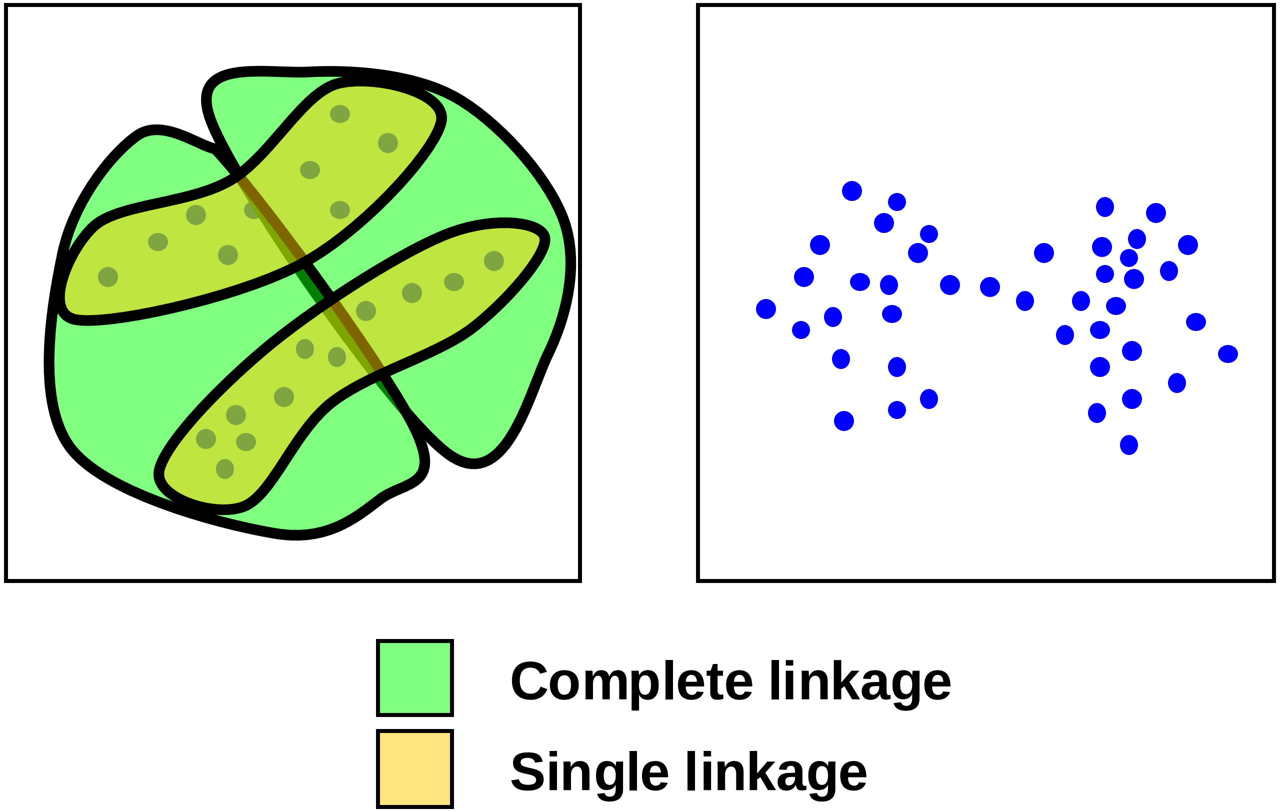




complete linkage

single linkage

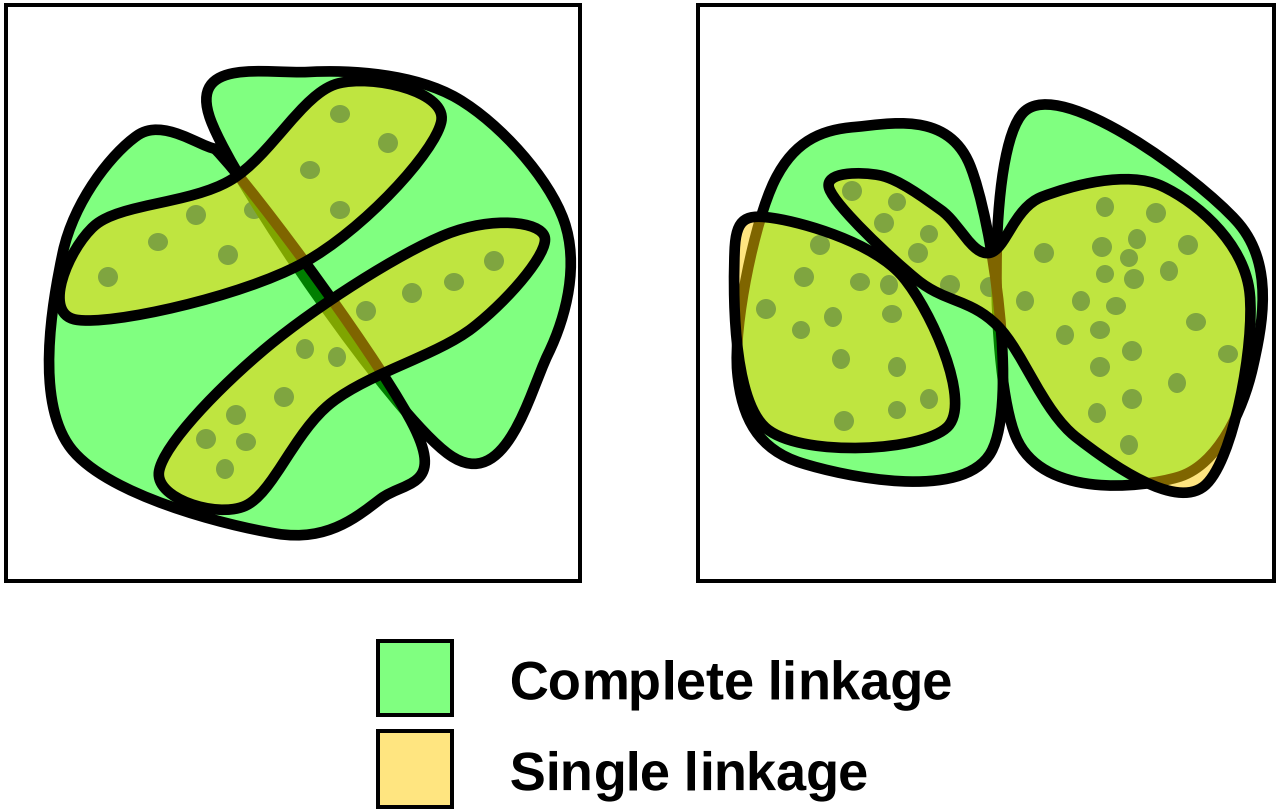
Linkage and cluster shape (2)



Complete linkage

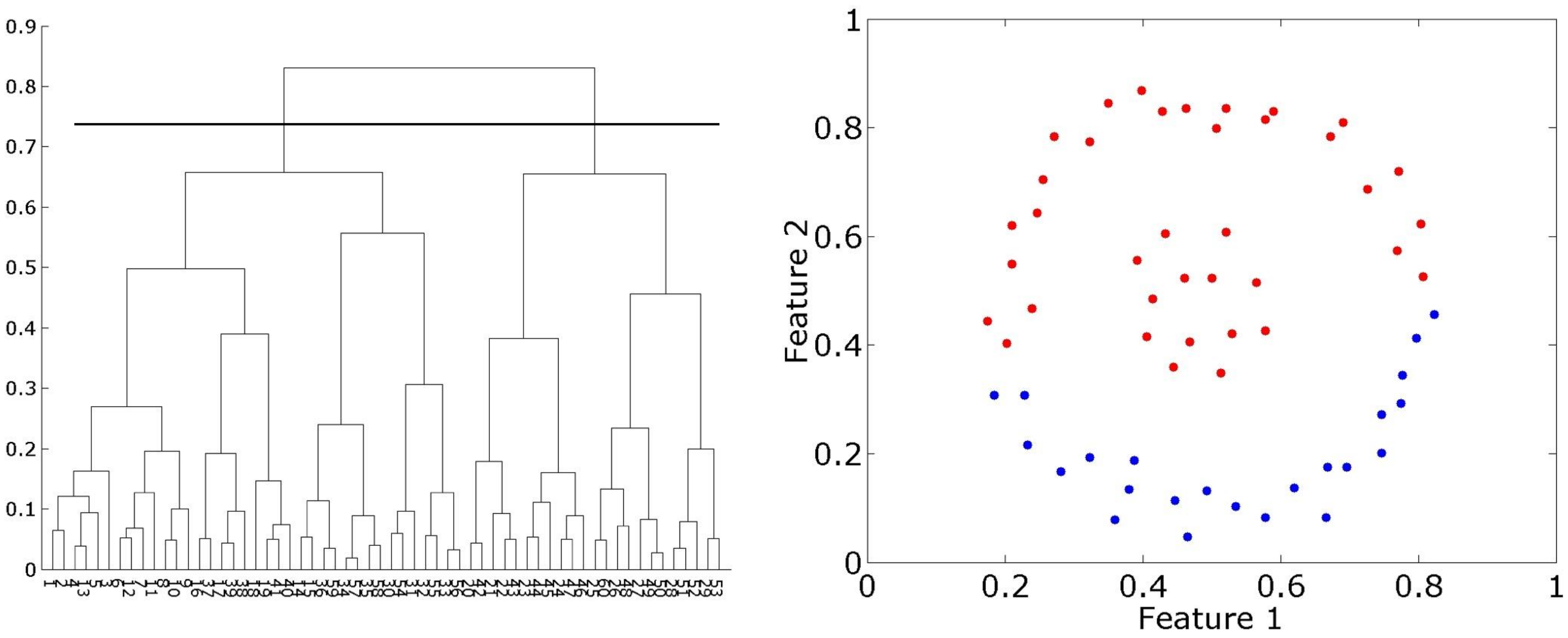
Single linkage

Linkage and cluster shape (3)



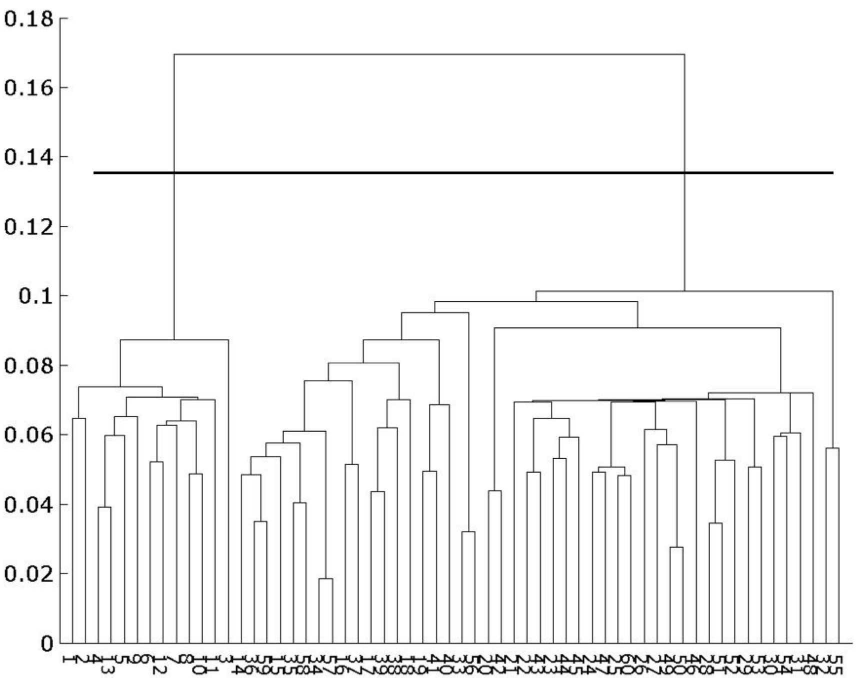
TU Delft

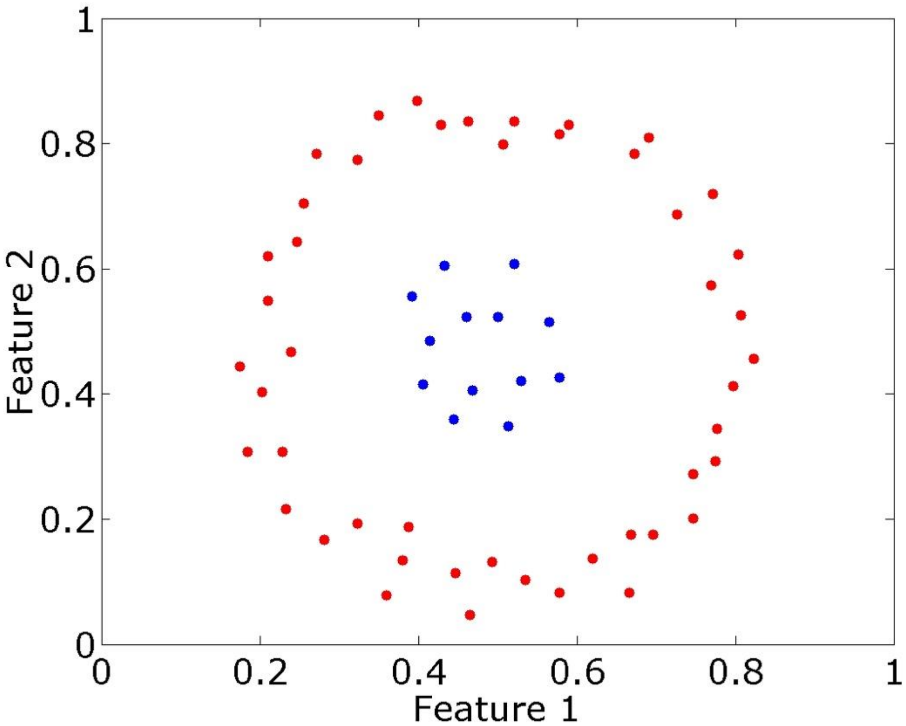
Hierarchical clustering examples
Euclidean, complete linkage



Hierarchical clustering examples

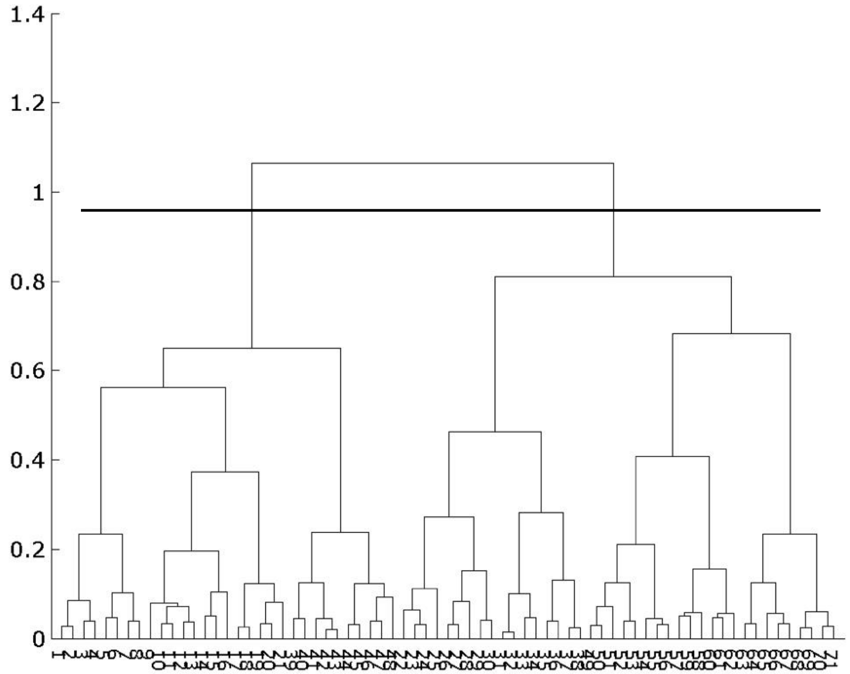
Euclidean, single linkage

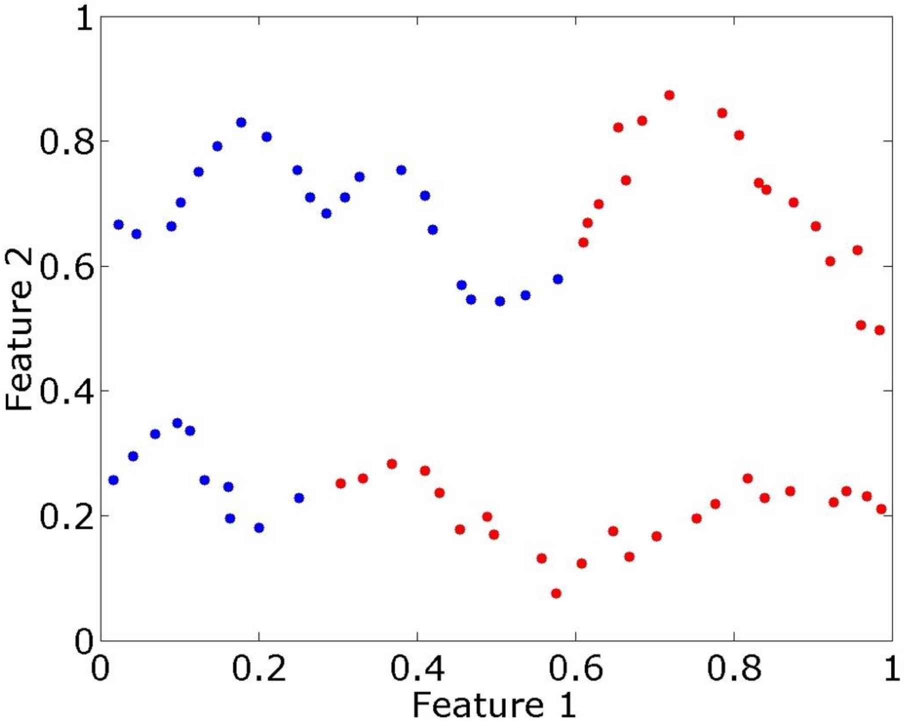




TU Delft

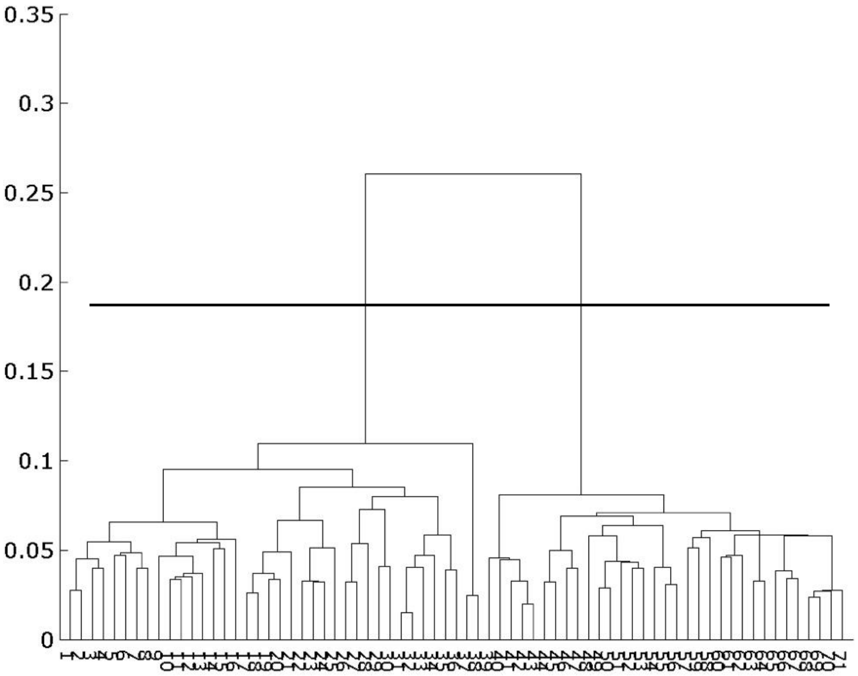
Hierarchical clustering examples
Euclidean, complete linkage

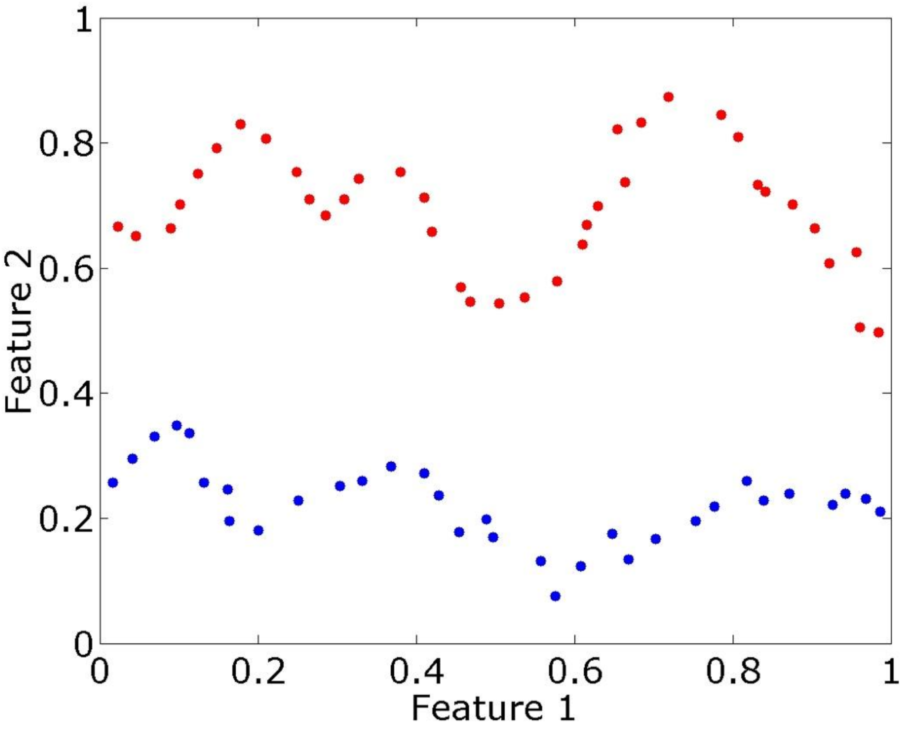




Hierarchical clustering examples

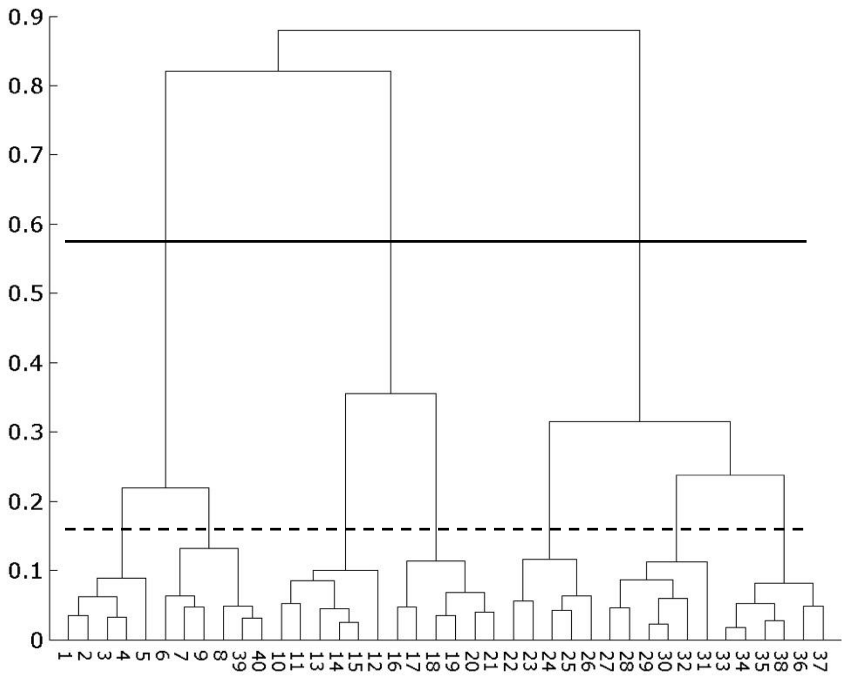
Euclidean, single linkage

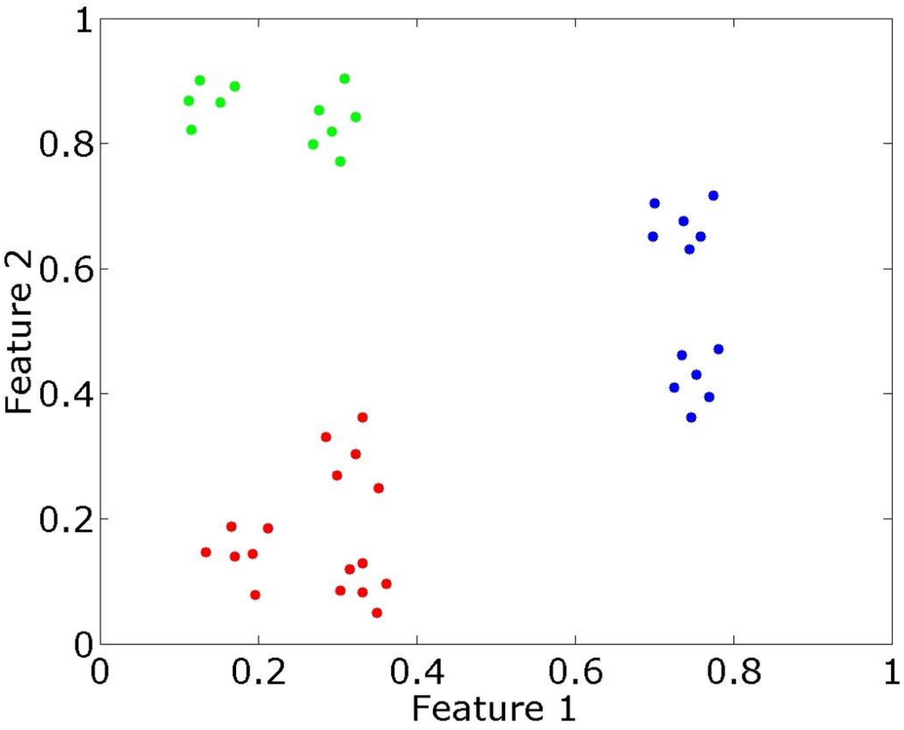




Hierarchical clustering examples

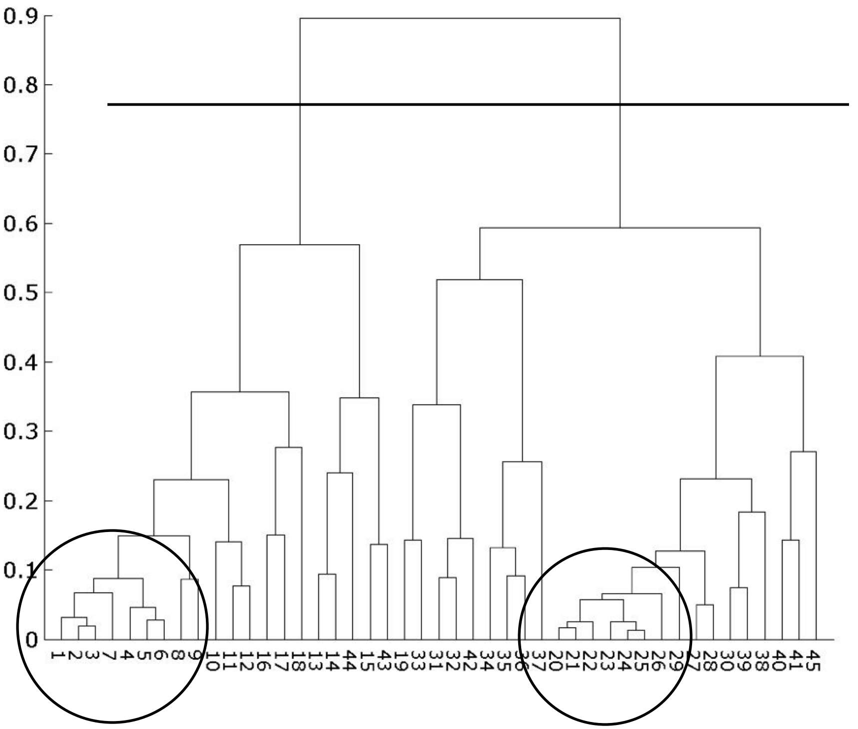
Euclidean, complete linkage

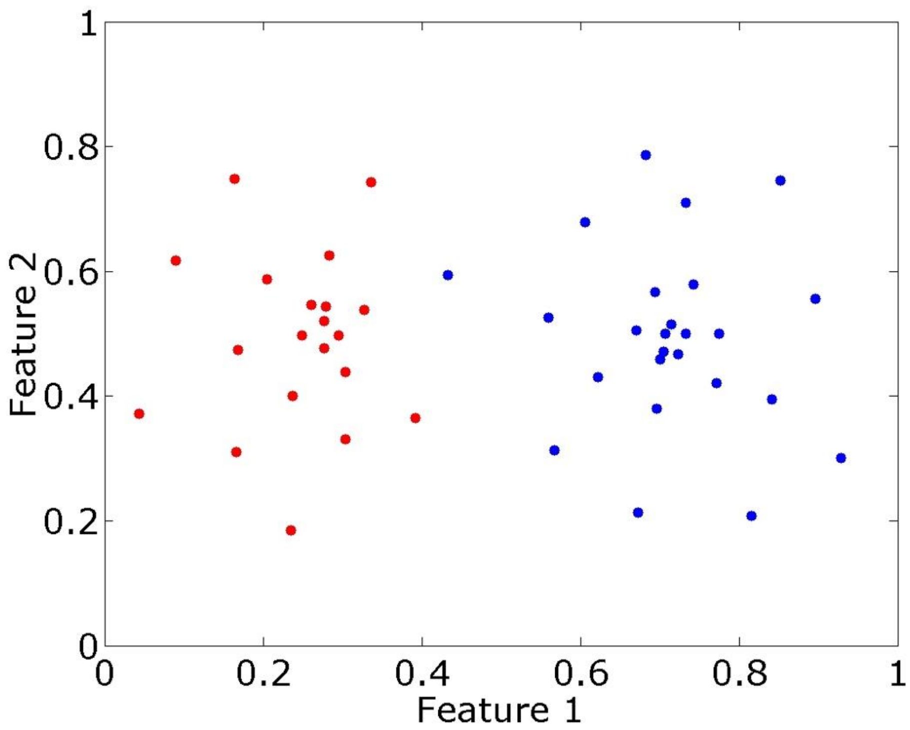




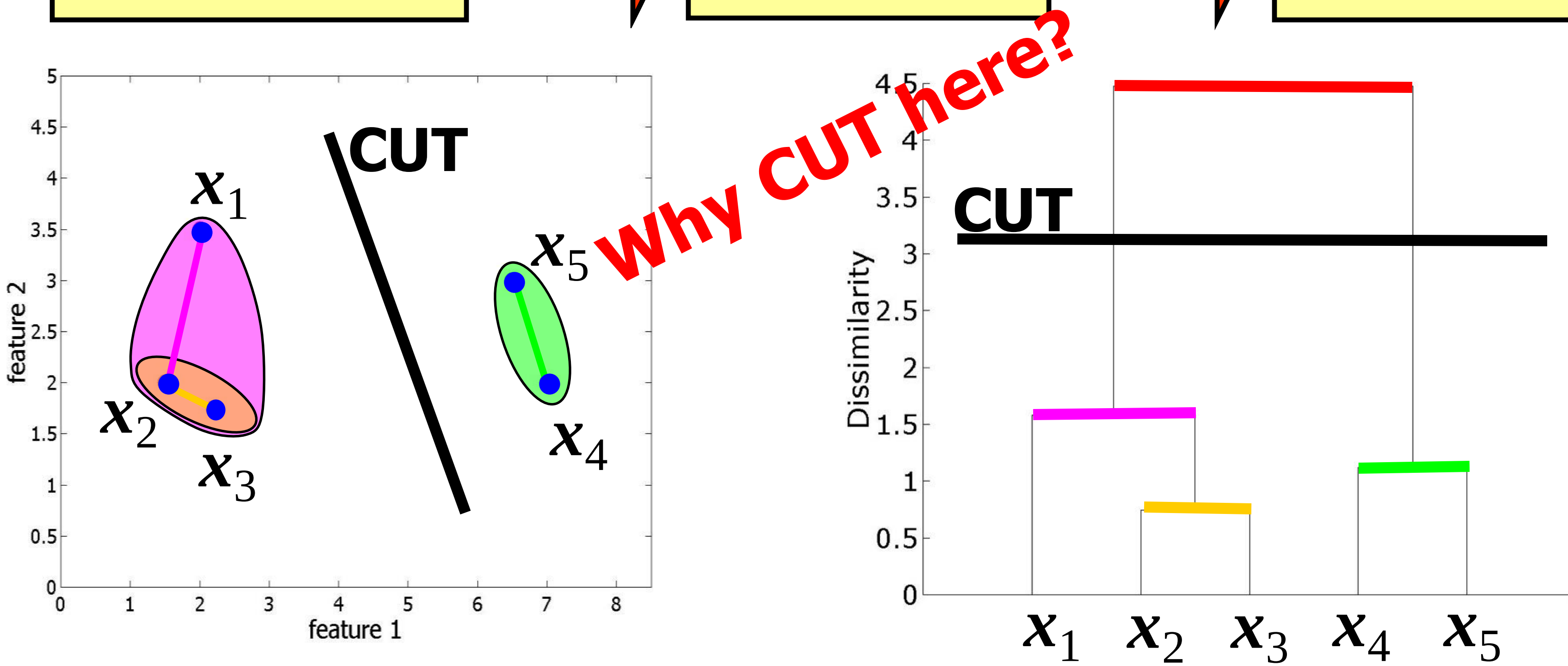
Hierarchical clustering examples

Euclidean, complete linkage



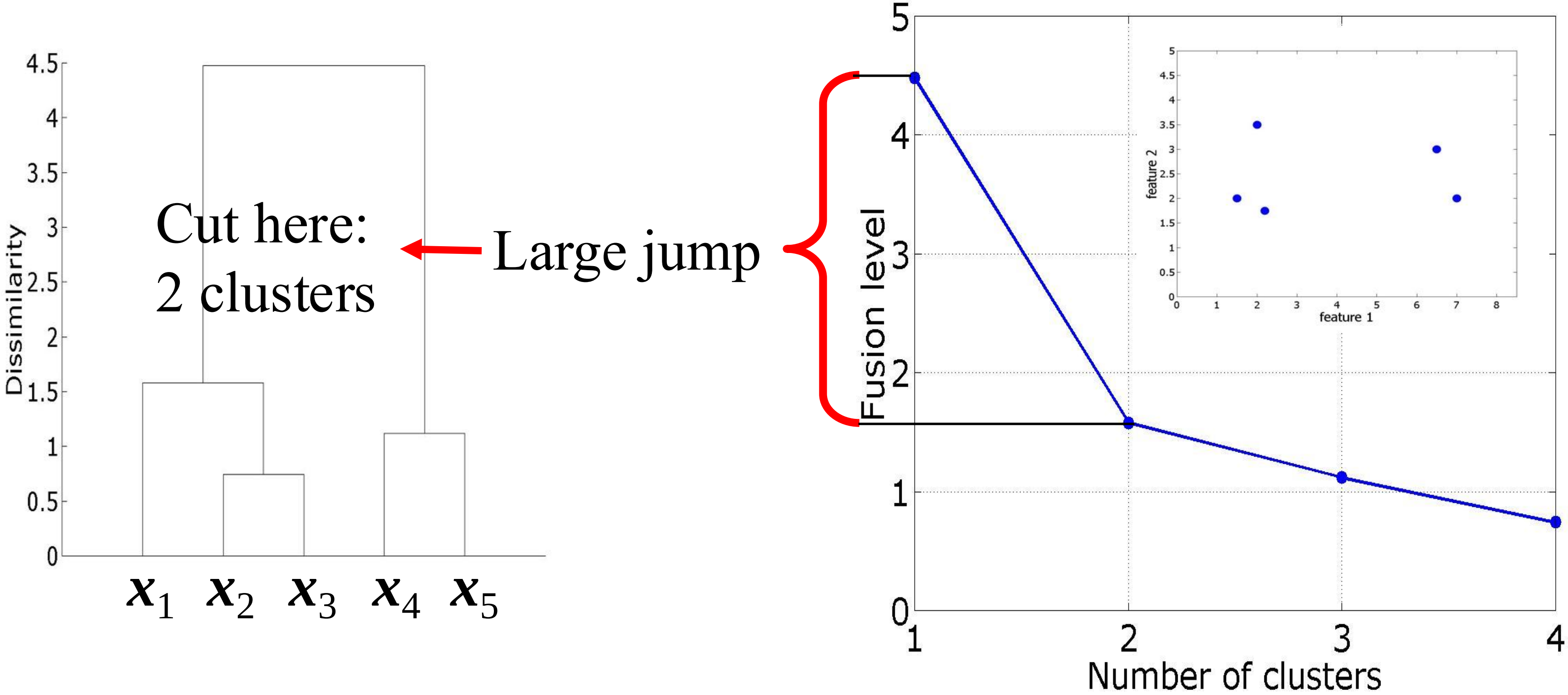


Remember our example...?



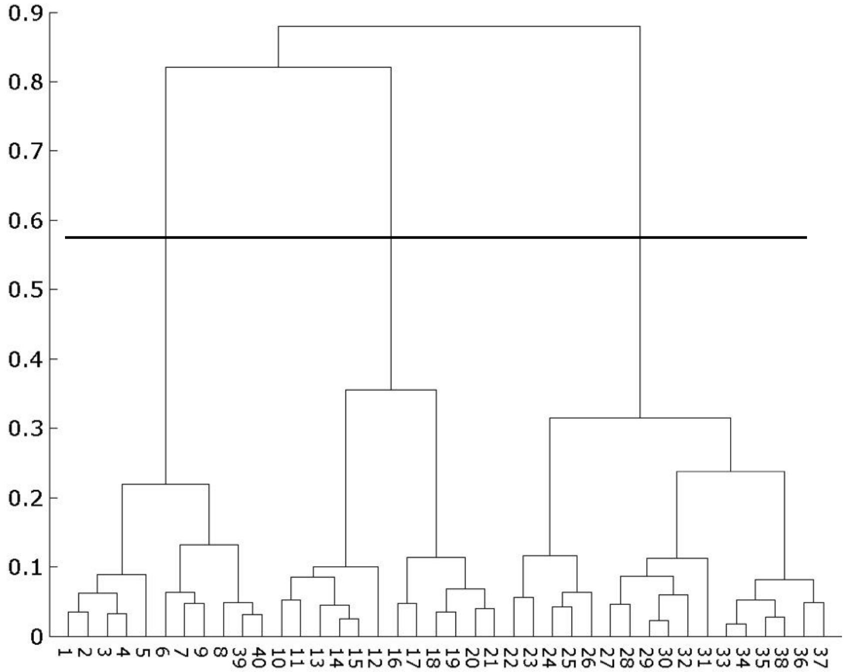
Fusion graph

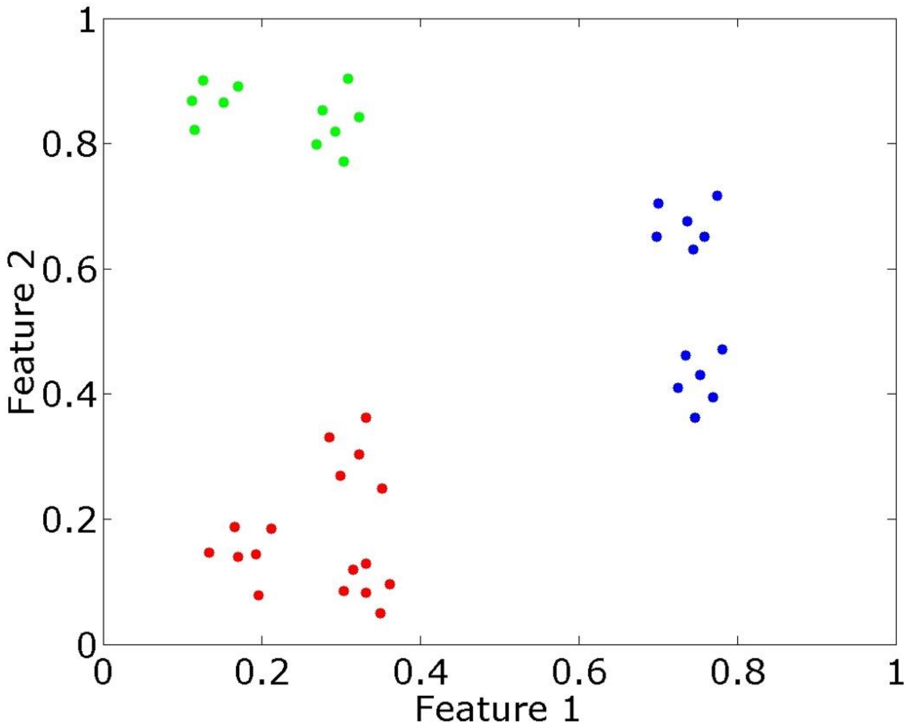
- fusion level



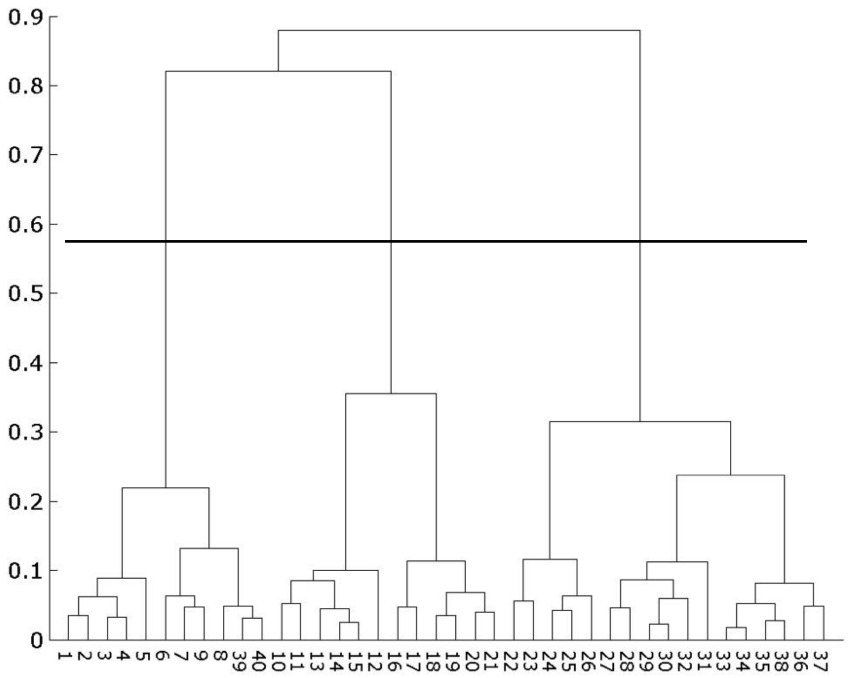
TUDelft

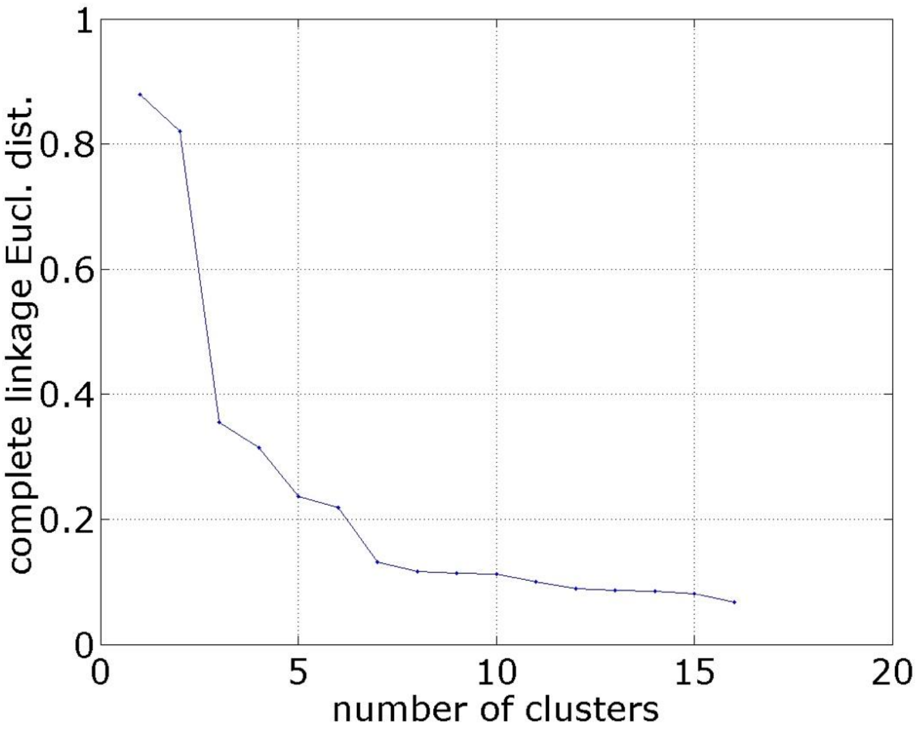
Fusion graph (2)
(Euclidean; complete linkage)



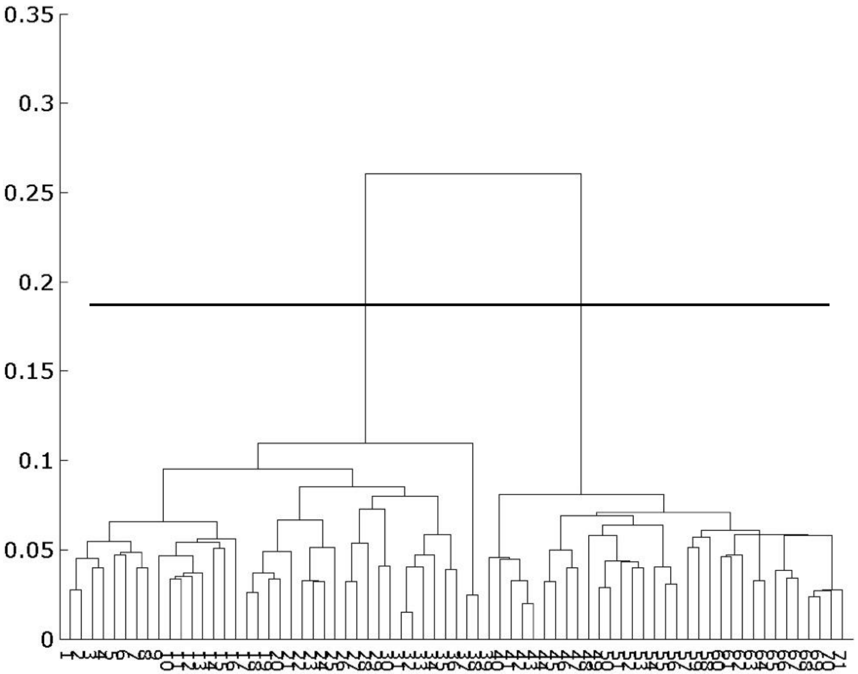


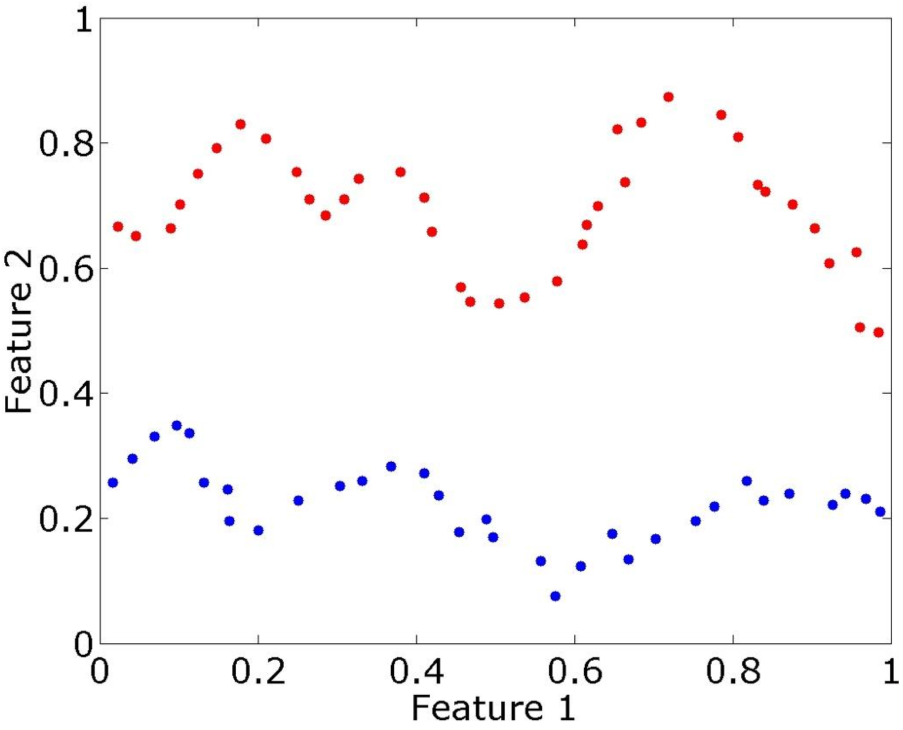
Fusion graph (3)
(Euclidean; complete linkage)



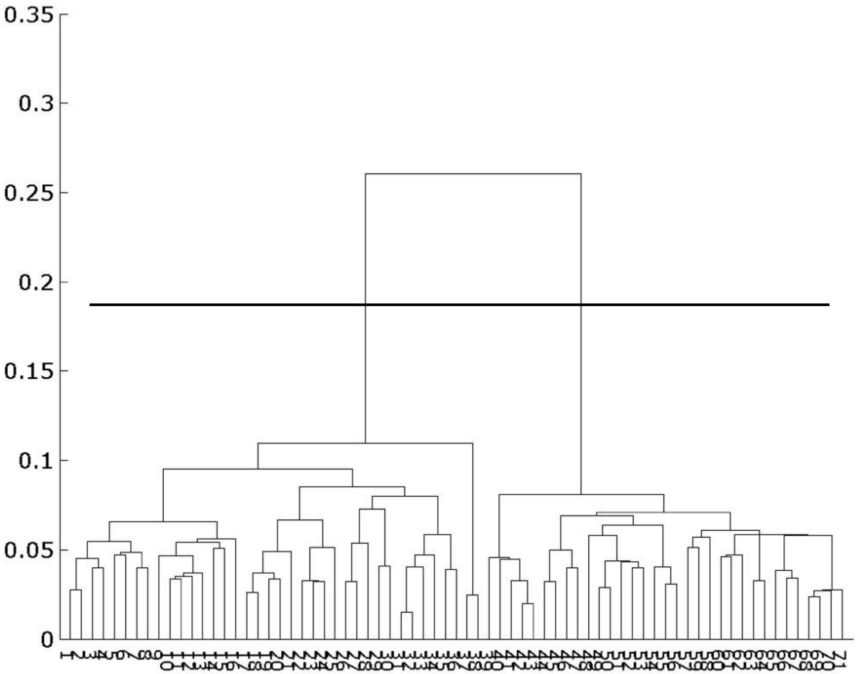


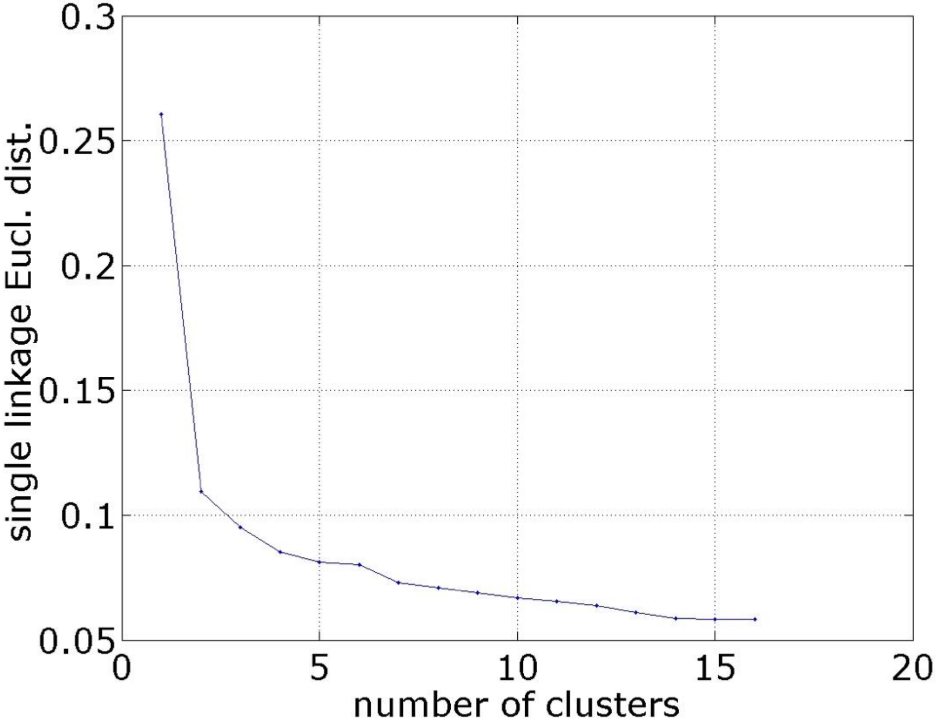
Fusion graph (4)
(Euclidean; single linkage)





Fusion graph (5)
(Euclidean; single linkage)





Hierarchical clustering (16)

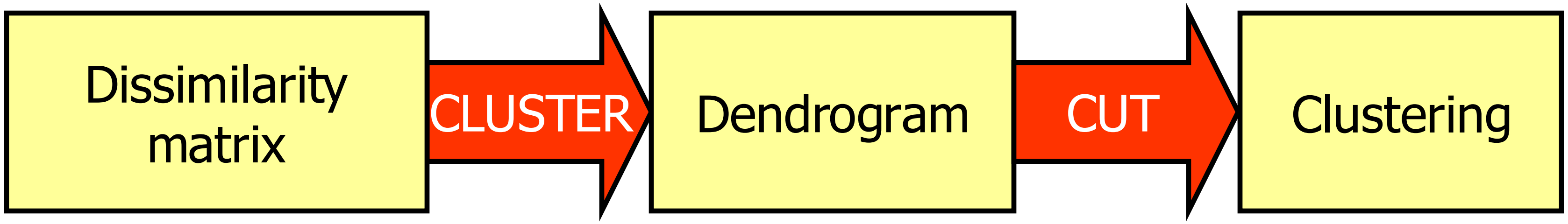
- Advantages:
  - dendrogram gives overview of all possible clusterings
  - linkage type allows to find clusters of varying shapes (convex and non-convex)
  - different dissimilarity measures can be used

- Disadvantages:
  - computationally intensive:
    $O(n^2)$ in complexity and memory
  - clusterings limited to “hierarchical nestings”

TU Delft

Any other way of clustering?

- Hierarchical:



- ?????? E.g., *K*-means (sum-of-squares)

Any other way of clustering?

- Let’s start with what we expect to see at the end after conducting a clustering!

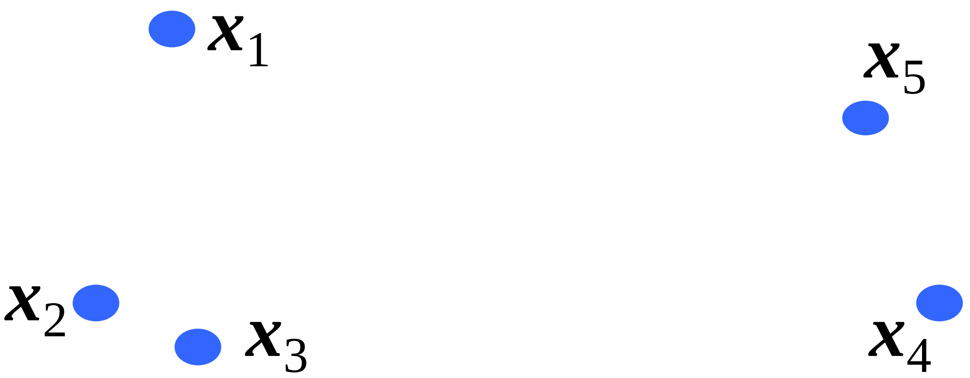
- *… …*

- Basically, our goal is to **partition the data into some number *K* of clusters**.

- K= ???

Intuitively, what should a cluster look like?

- Well…a group of data points…
- whose **inter-point distances** are **small** compared with the **distances to points outside of the cluster**.



A notion to represent the k^{th} cluster

- μ_k , k=1,…,K, a prototype associated with the k^{th} cluster.

- But what is μ_k ? A simple but sensible choice…

A notion to represent the *k* th cluster

- μ_k, $k=1,\ldots,K$, a prototype associated with the k th cluster.

- But what is μ_k ? A simple but sensible choice…

- the centre of the cluster

Our goal is then…

- To find an **assignment of data points** to clusters, as well as a set of vectors μ_k , such that

- the **sum of the squares of the distances** of each data point to its closest μ_k, is a **minimum**.

K-means clustering,
a.k.a., sum-of-squares clustering

Our goal is then…

- To find an **assignment of data points** to clusters, as well as a set of vectors μ_k , such that

- the **sum of the squares of the distances** of each data point to its closest μ_k, is a **minimum**.

- But **where** to put μ_k ...

K-means clustering,

a.k.a., sum-of-squares clustering

TU Delft

K-means clustering

- 1. choose number of clusters (K)
- 2. position prototypes (μ_k, $k=1,\ldots,K$) **randomly**
- 3. assign samples to closest prototype
- 4. compute mean of samples assigned to same prototype ➔ **new prototype position**

Repeat steps 3 and 4 as long as prototypes move

K-means clustering

10.5

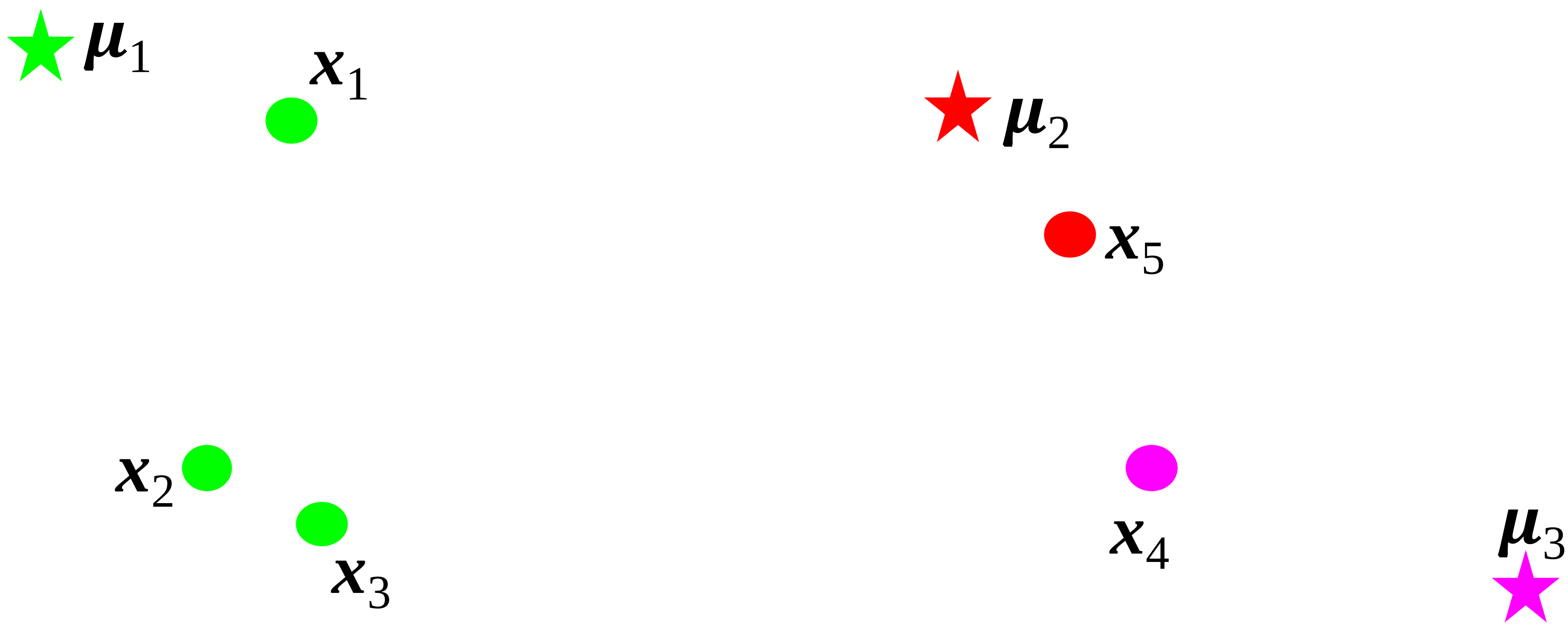
- **Step 1:** Choose number of clusters/prototypes
- **Step 2:** Position prototypes randomly

TU Delft

K-means clustering

10.5

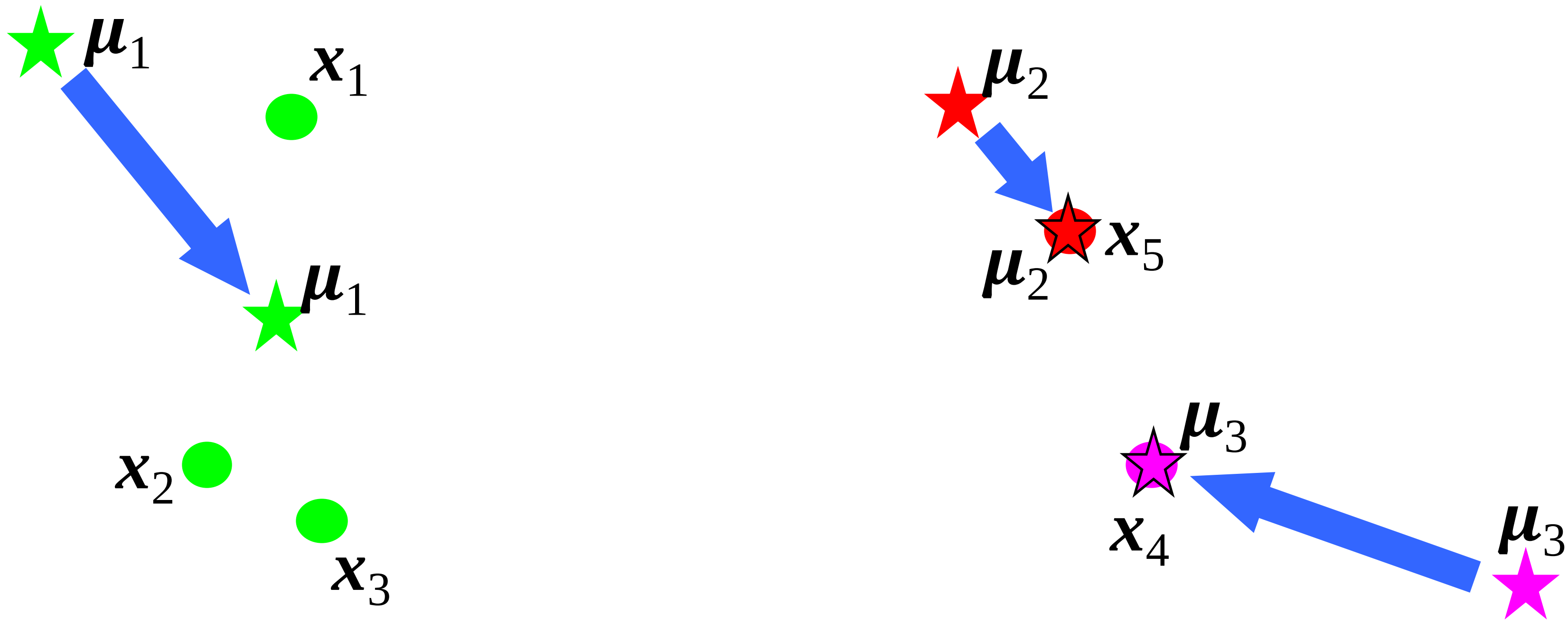
- **Step 3:** Assign samples to closest prototype



TU Delft

K-means clustering

- **Step 4:** Compute mean of samples assigned to same prototype ➔ **new prototype positions**



K-means clustering

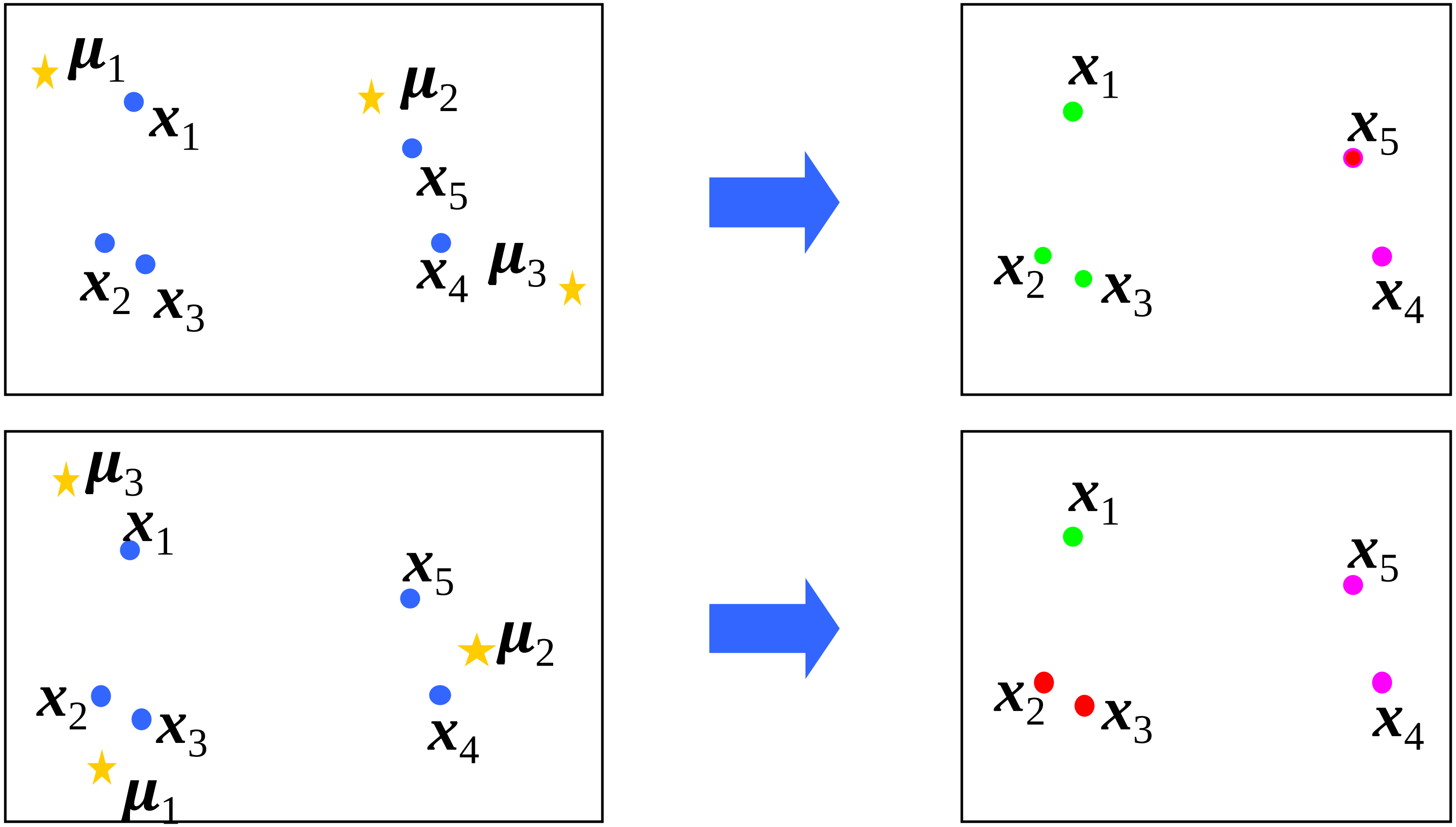
10.5

- **Repeat** as long as prototype positions change**:**
  - **Step 3:** Assign samples
  - **Step 4:** Recompute prototype positions

TUDelft

K-means problems

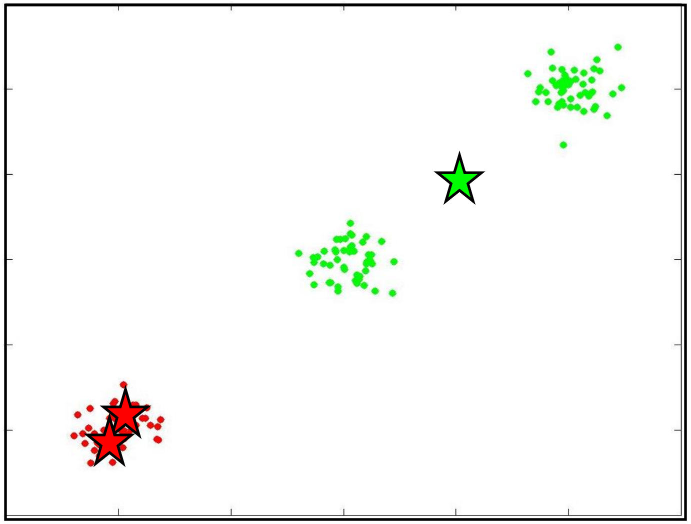
- Clustering depends on initialization



K-means problems

10.5

- Algorithm can get stuck in local minima



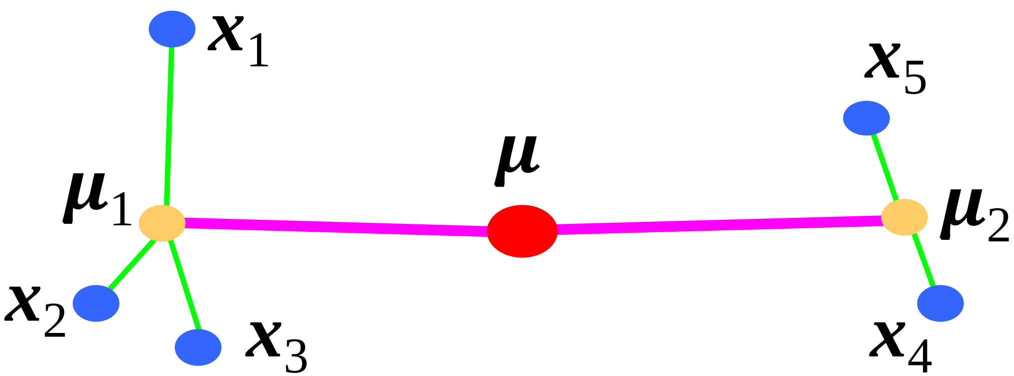
- Solution:
  - start from I **different random initialisations;**
  - **keep the best clustering** (lowest $\mathrm{Tr}(\boldsymbol{S}_{\mathrm{W}})$);
  - For high-dimensional data, many restarts are necessary (e.g. I = 10000)!

K-means clustering

- Recall from Week 4 (within and between group scatter):

$$\boldsymbol{S}_W = \sum_{j=1}^{K} \frac{n_k}{N} \boldsymbol{\Sigma}_k \,,$$

$$\boldsymbol{S}_B = \sum_{k=1}^{K} \frac{n_k}{N} (\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^T, \quad \boldsymbol{\mu} = \sum_{k=1}^{K} \frac{n_k}{N} \boldsymbol{\mu}_k$$



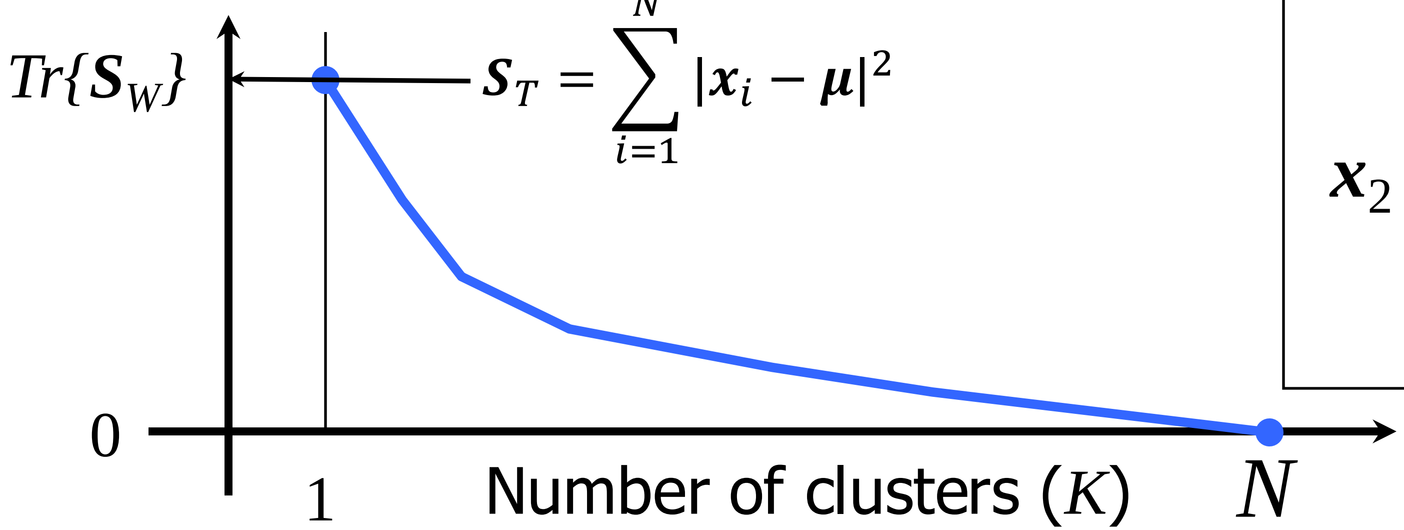
$K = 2, N = 5,$
$n_1 = 3, n_2 = 2$

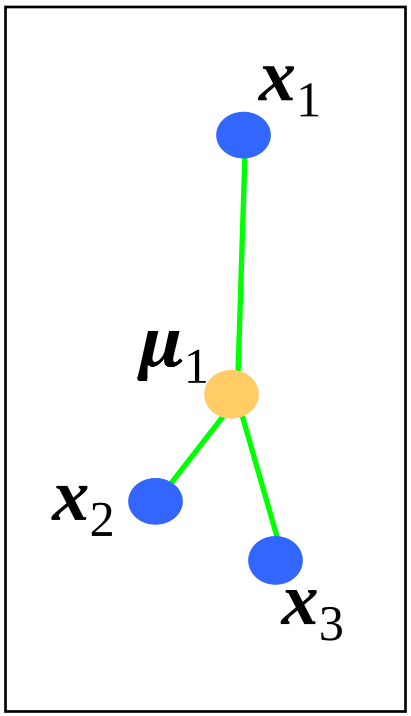
K-means clustering

- Minimize:

$$Tr\{\boldsymbol{S}_W\} = \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n_k}|\boldsymbol{x}_i - \boldsymbol{\mu}_k|^2$$

(sum of cluster within-scatters)





Our goal is then… (Recall)

- To find an **assignment of data points** to clusters, as well as a set of vectors μ_k , such that

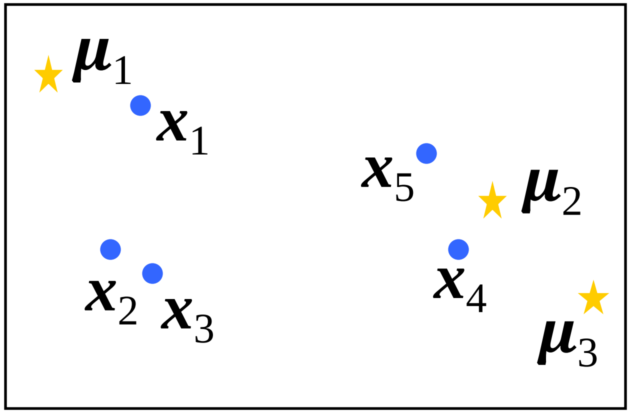
- the **sum of the squares of the distances** of each data point to its closest μ_k, is a **minimum**.

K-means clustering,

a.k.a., sum-of-squares clustering

K-means problems

- Some prototypes do not capture any data points ($\boldsymbol{\mu}_3$)

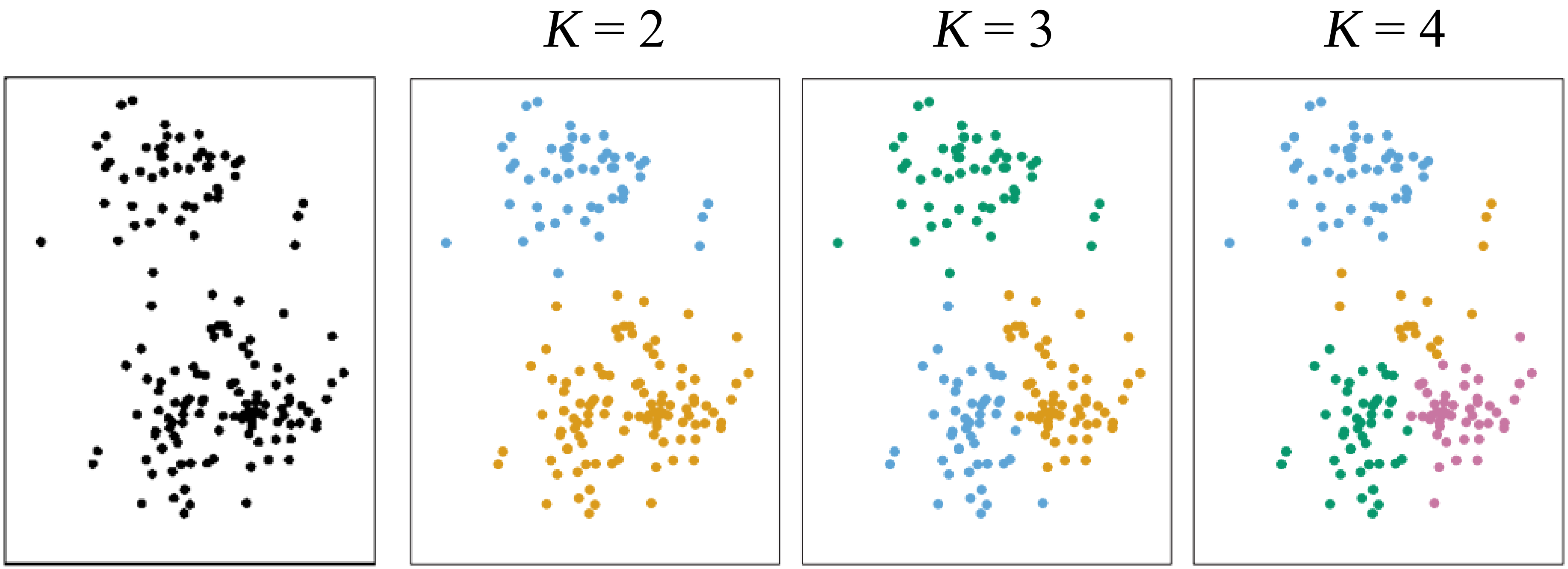


- Possible solution:
  - remove cluster and continue with $K-1$ means
  - alternatively, split largest cluster into two or add a random cluster to continue with K means

TUDelft

K-means clustering

- Choose a different *K*:



TU Delft

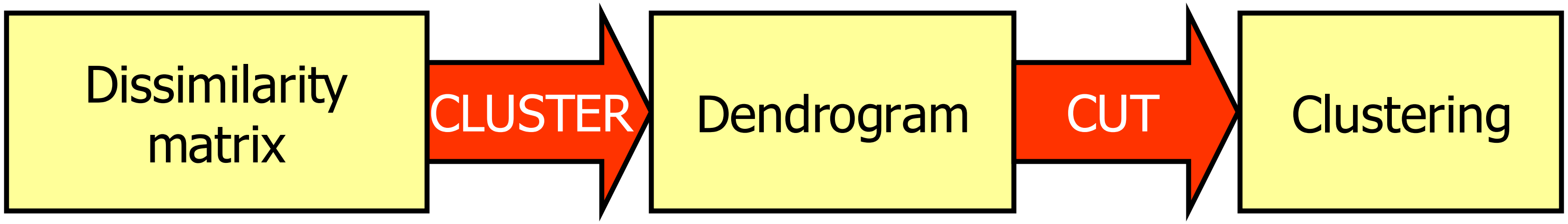
K-means summary

10.5

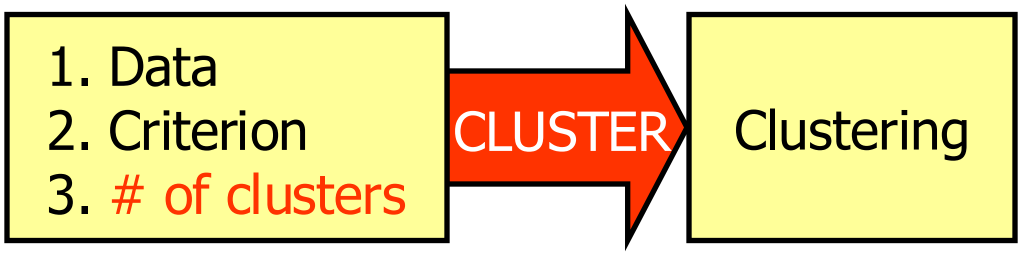
- Disadvantages:
  - Finds only convex clusters (“round shapes”)
  - Sensitive to initialization
  - Can get stuck in local minima
- Advantages:
  - Simple
  - Fast

Clustering

- Hierarchical:

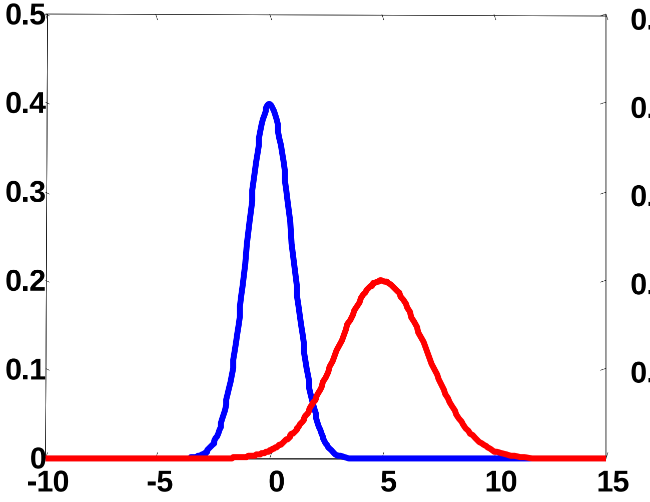


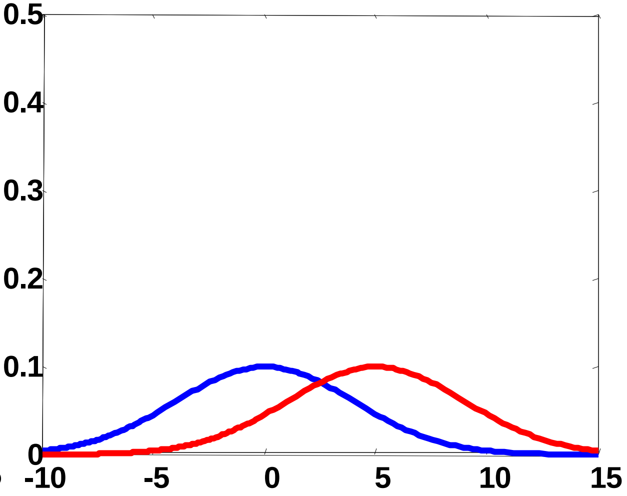
- Sum-of-squares:



Validity measures

- Many are based on within and between group scatter
- The **larger the between** group scatter and **the smaller the within** group scatter, **the better**
- Example: Davies-Bouldin index





TU Delft

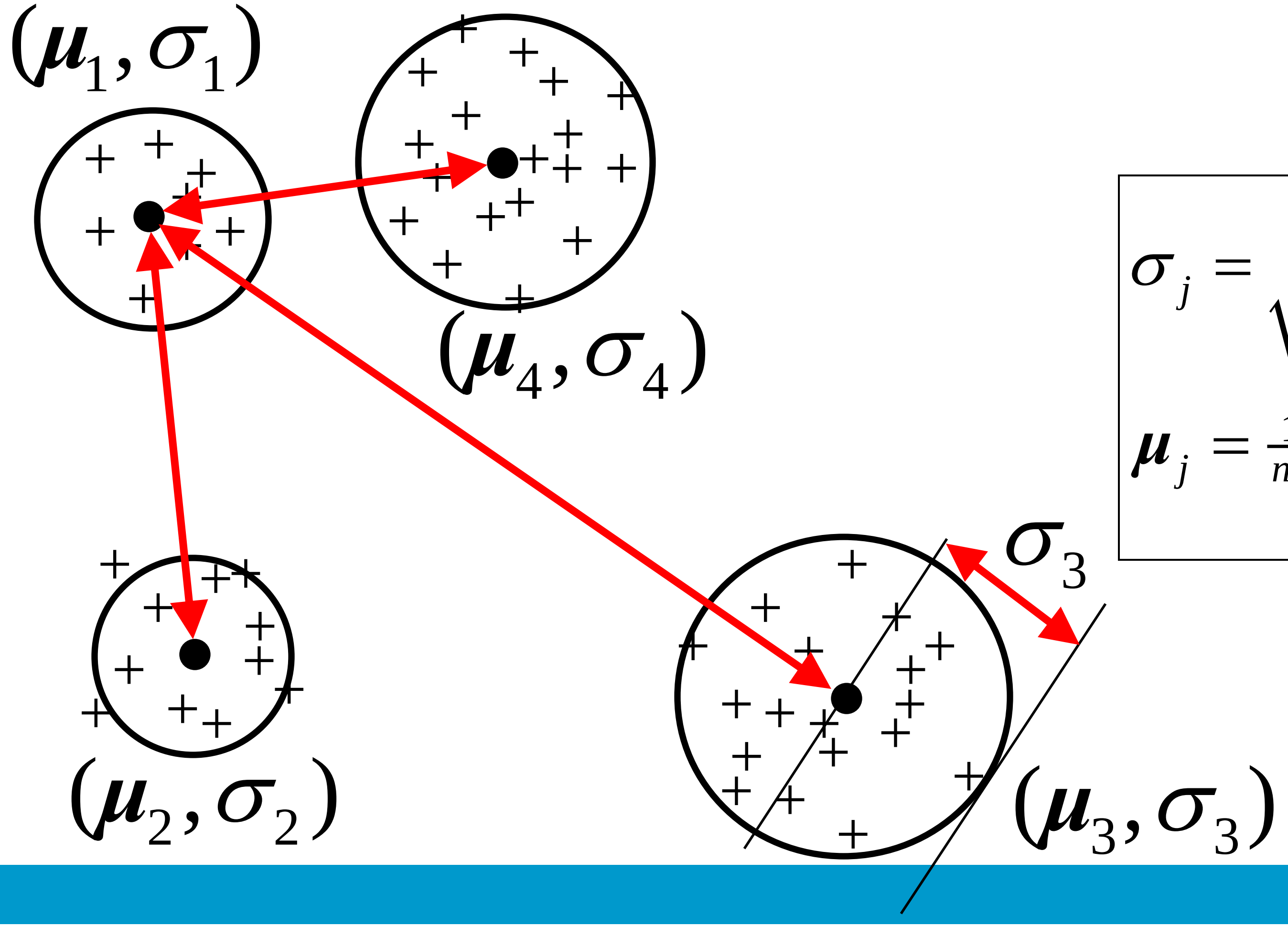
Davies-Bouldin index

- **Assumption: clusters are spherical**
- For a good clustering, it should hold that:
  - objects are compactly organized within a cluster
  - clusters are far apart

- D.L. Davies and D.W. Bouldin, IEEE Transactions on Pattern Analysis and Machine Intelligence 1, pp. 224-227, 1979

TU Delft

Davies-Bouldin index (2)

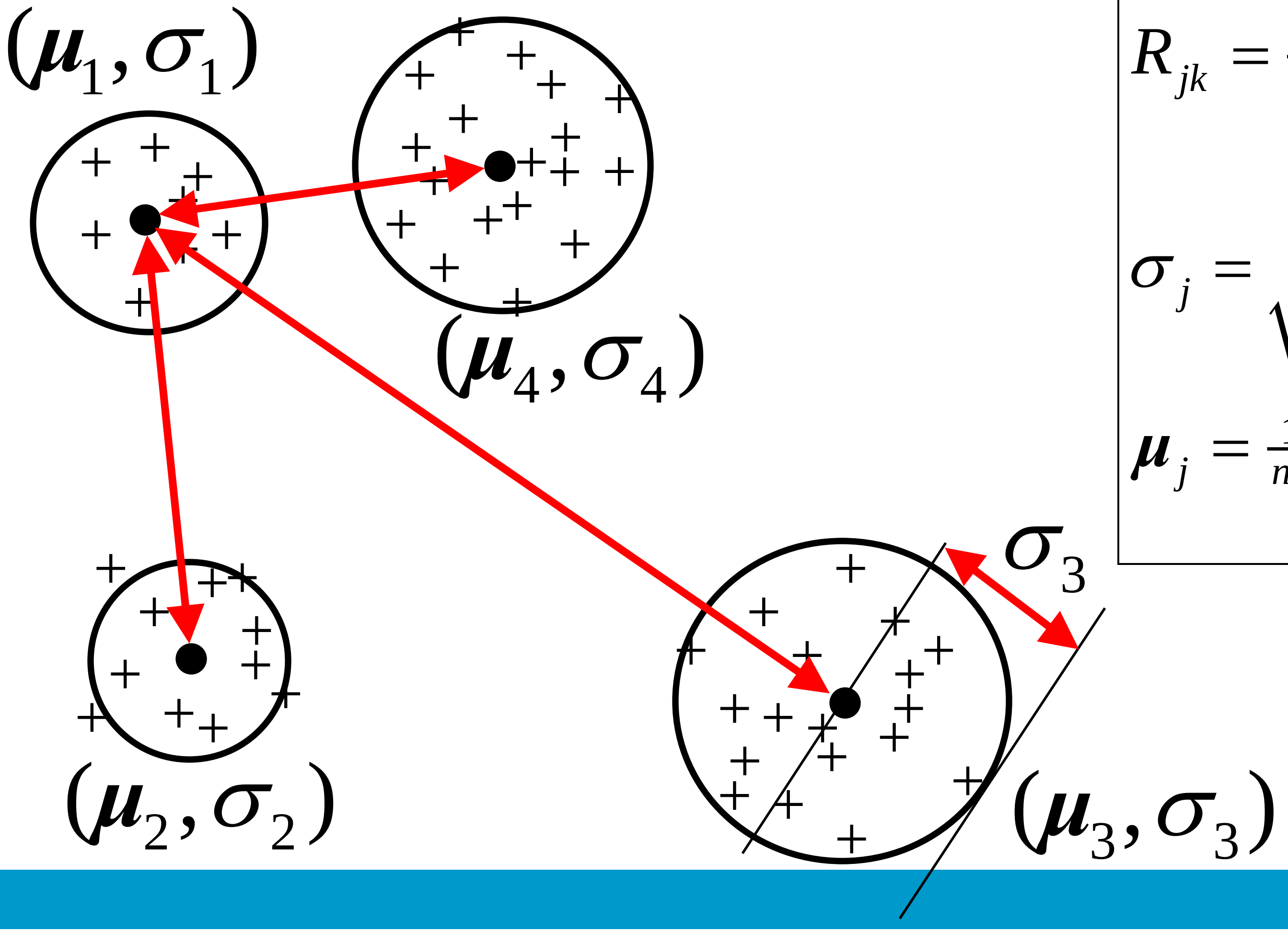


$$\sigma_j = \sqrt{\frac{1}{n_j} \sum_{\boldsymbol{x}_i \in C_j} \left\| \boldsymbol{x}_i - \boldsymbol{\mu}_j \right\|^2}$$

$$\boldsymbol{\mu}_j = \frac{1}{n_j} \sum_{\boldsymbol{x}_i \in C_j} \boldsymbol{x}_i$$

TU Delft

Davies-Bouldin index (3)

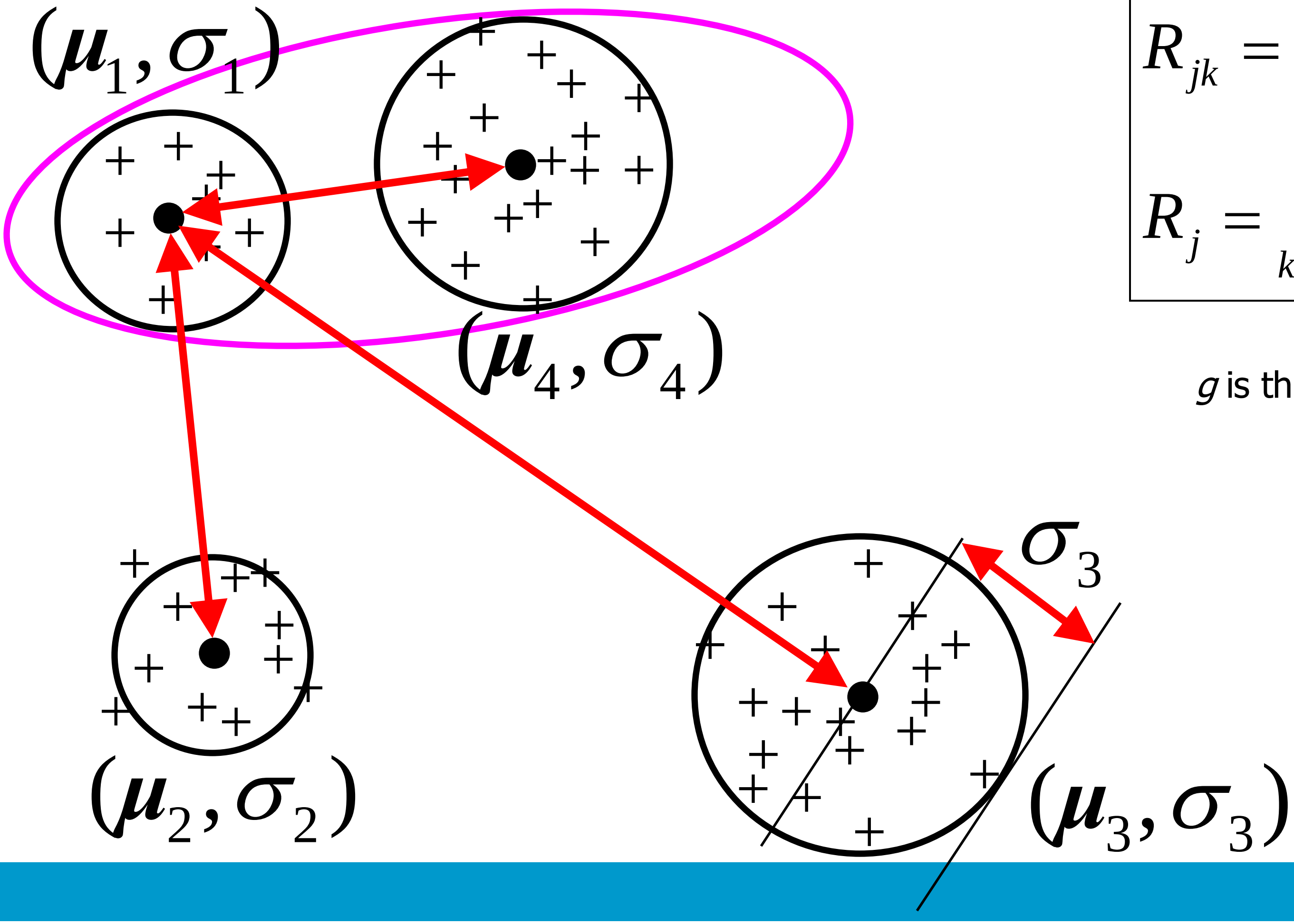


$$R_{jk} = \frac{\sigma_j + \sigma_k}{\left\|\boldsymbol{\mu}_j - \boldsymbol{\mu}_k\right\|}$$

$$\sigma_j = \sqrt{\tfrac{1}{n_j} \sum_{\boldsymbol{x}_i \in C_j} \left\|\boldsymbol{x}_i - \boldsymbol{\mu}_j\right\|^2}$$

$$\boldsymbol{\mu}_j = \tfrac{1}{n_j} \sum_{\boldsymbol{x}_i \in C_j} \boldsymbol{x}_i$$

Davies-Bouldin index (3)



$$R_{jk} = \frac{\sigma_j + \sigma_k}{\left\| \boldsymbol{\mu}_j - \boldsymbol{\mu}_k \right\|}$$

$$R_j = \max_{k=1,..g;k\neq j} R_{jk}$$

g is the number of clusters

Davies-Bouldin index (4)

$$R_{jk} = \frac{\sigma_j + \sigma_k}{\left\| \boldsymbol{\mu}_j - \boldsymbol{\mu}_k \right\|}$$

Paired cluster criterion

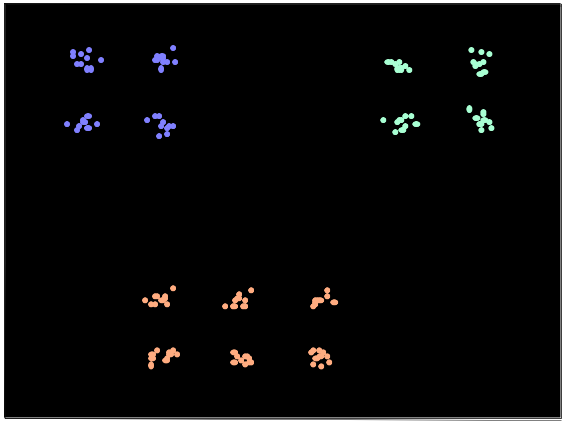
$$R_j = \max_{k=1,..g;k \neq j} R_{jk}$$

Worst-case value per cluster

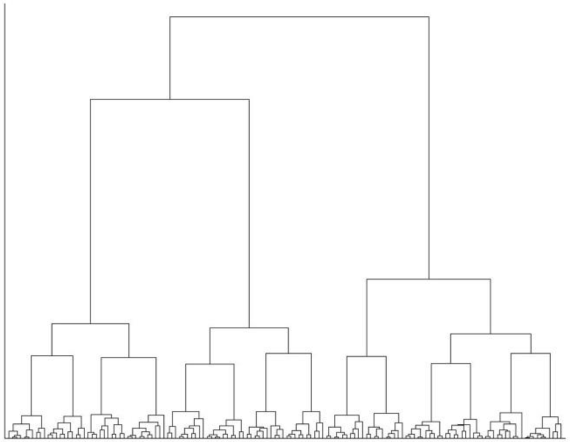
$$I_{DB} = \frac{1}{g} \sum_{j=1}^{g} R_j$$

Average worst-case

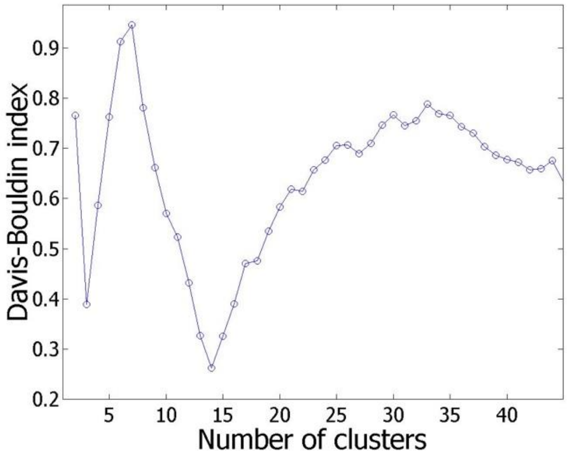
Davies-Bouldin index (5)



Dataset

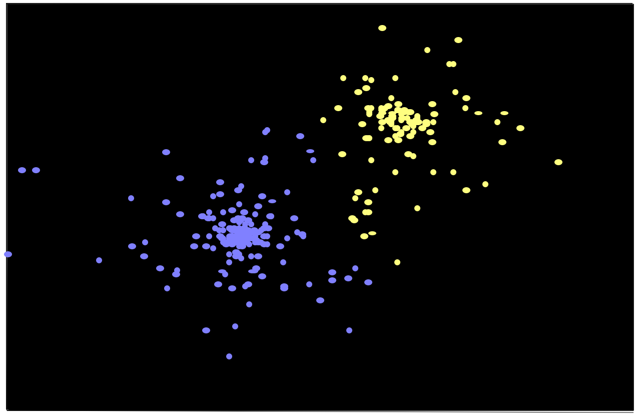


Complete link

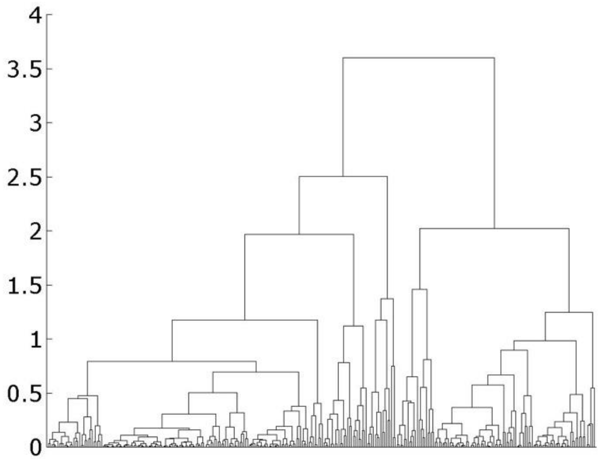


Davis Bouldin:
3 or 14 clusters

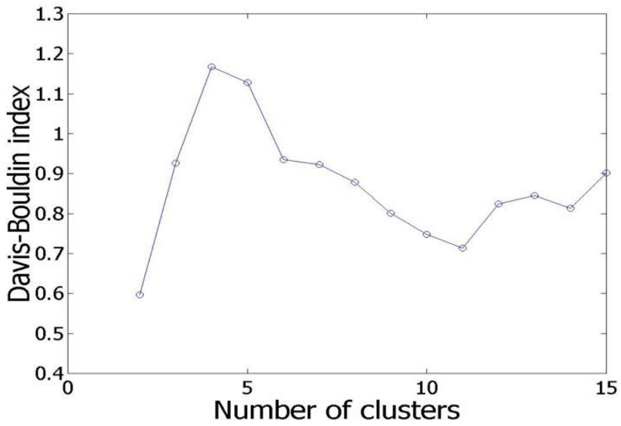
Davies-Bouldin index (5)



Dataset



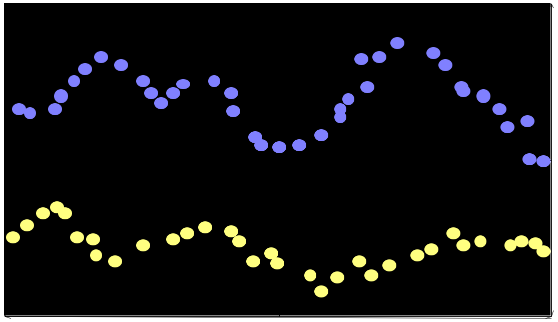
Complete link

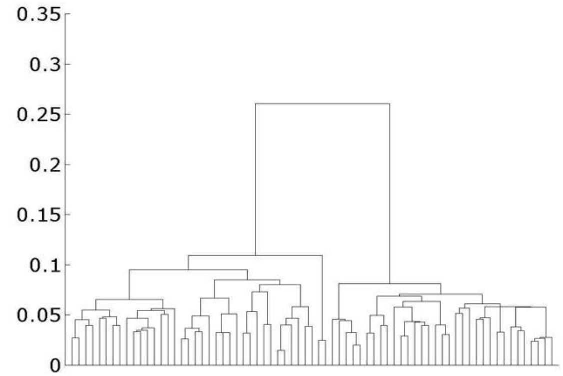


Davis Bouldin:
2 or 11 clusters

TUDelft

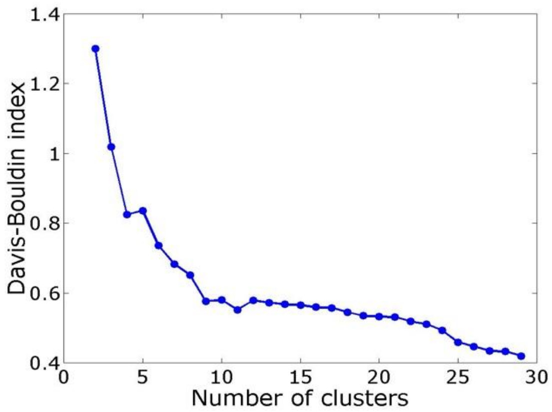
Davies-Bouldin index (6)

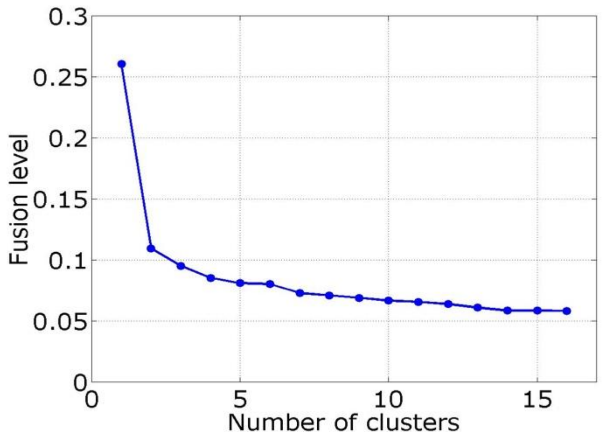




Davis-Bouldin:

Single link





TU Delft