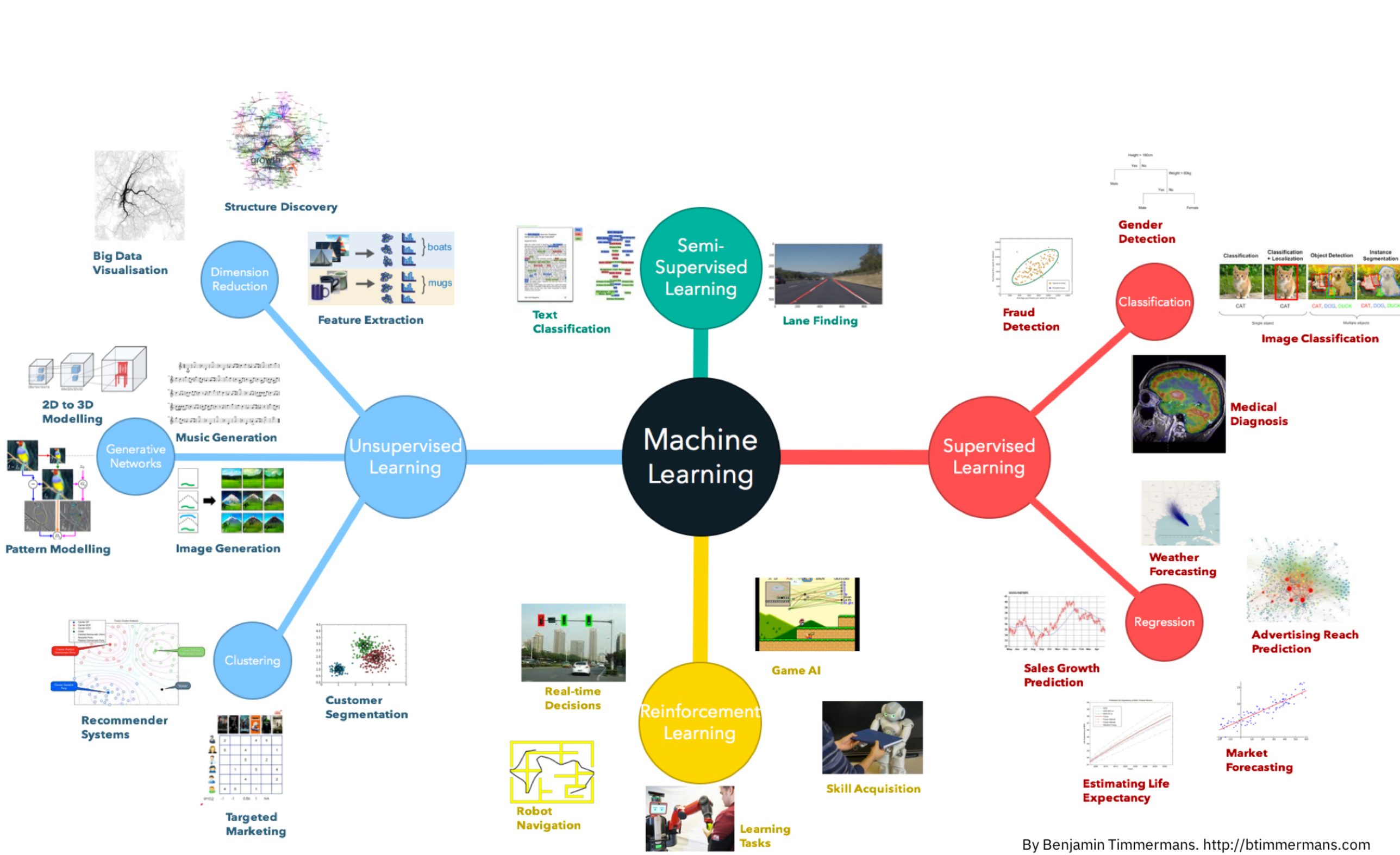
Clustering

- Unsupervised learning: no labels/targets present

TU Delft

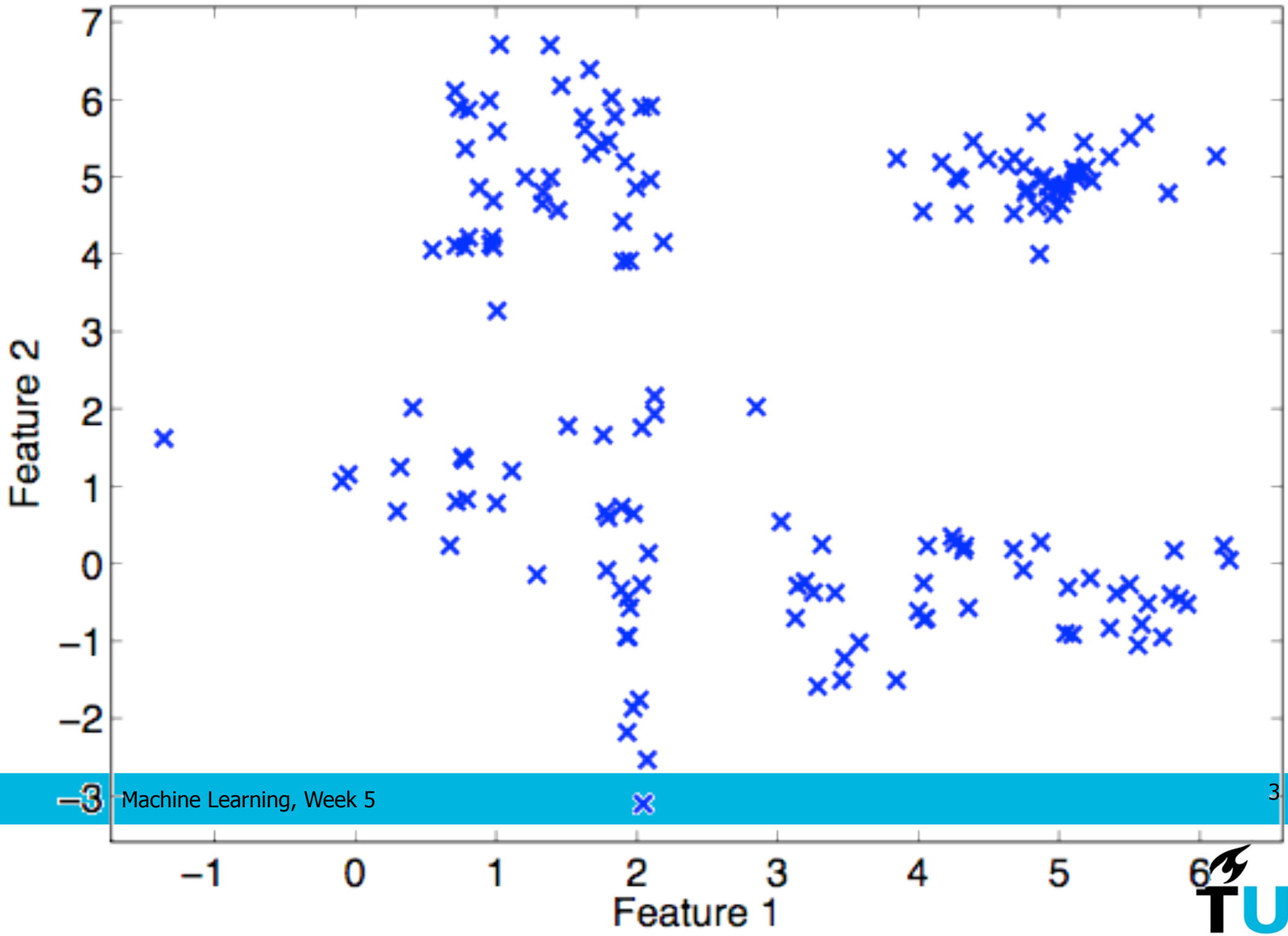


By Benjamin Timmermans. http://btimmermans.com

TU Delft

Unlabeled data: what now?

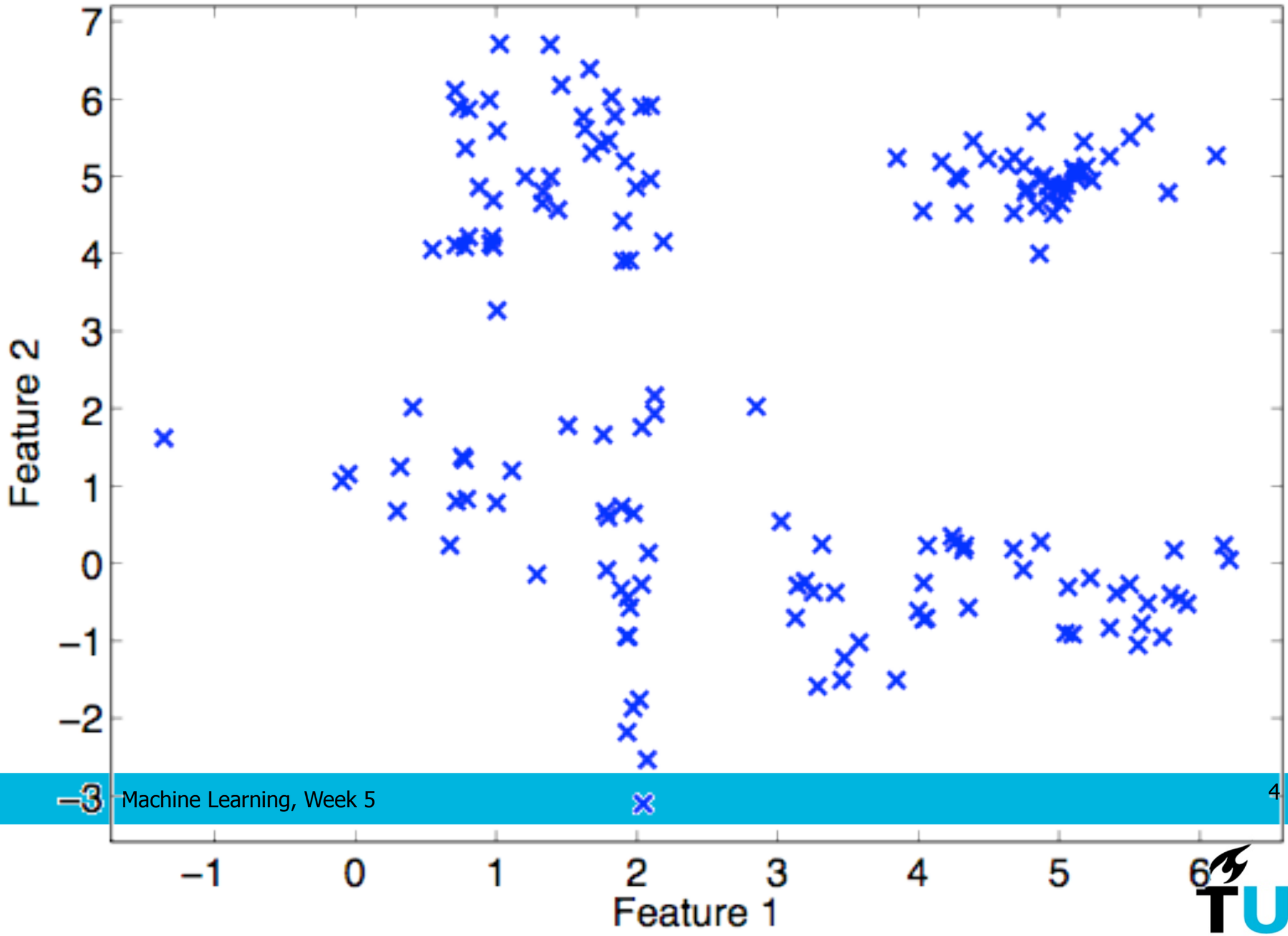
Ch 11



TU Delft

How Many Groups in Data?

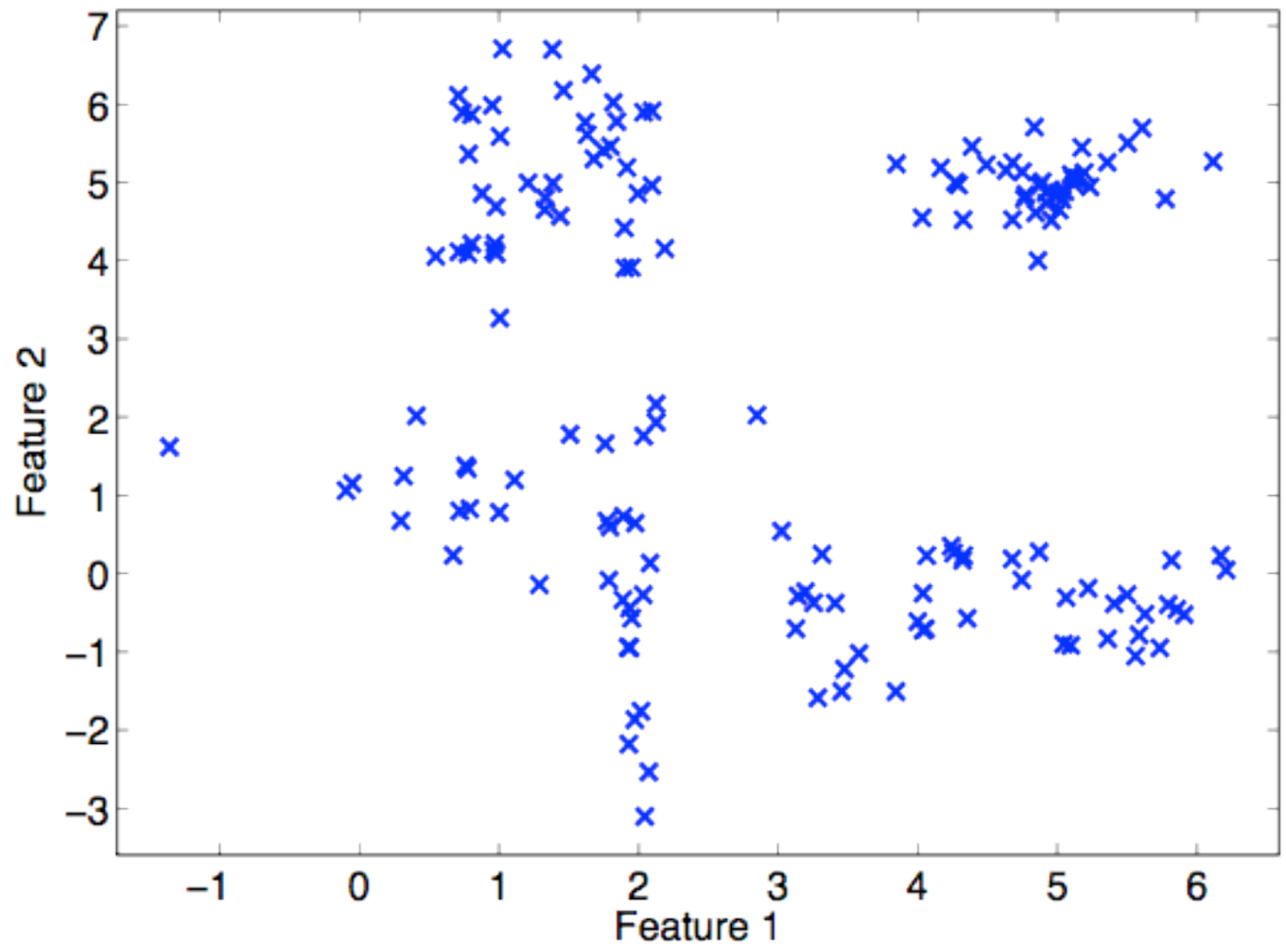
Ch 11



TU Delft

Clustering

Ch 11



- Unsupervised methods [no labels]
- What salient structures exist in the data?

TU Delft

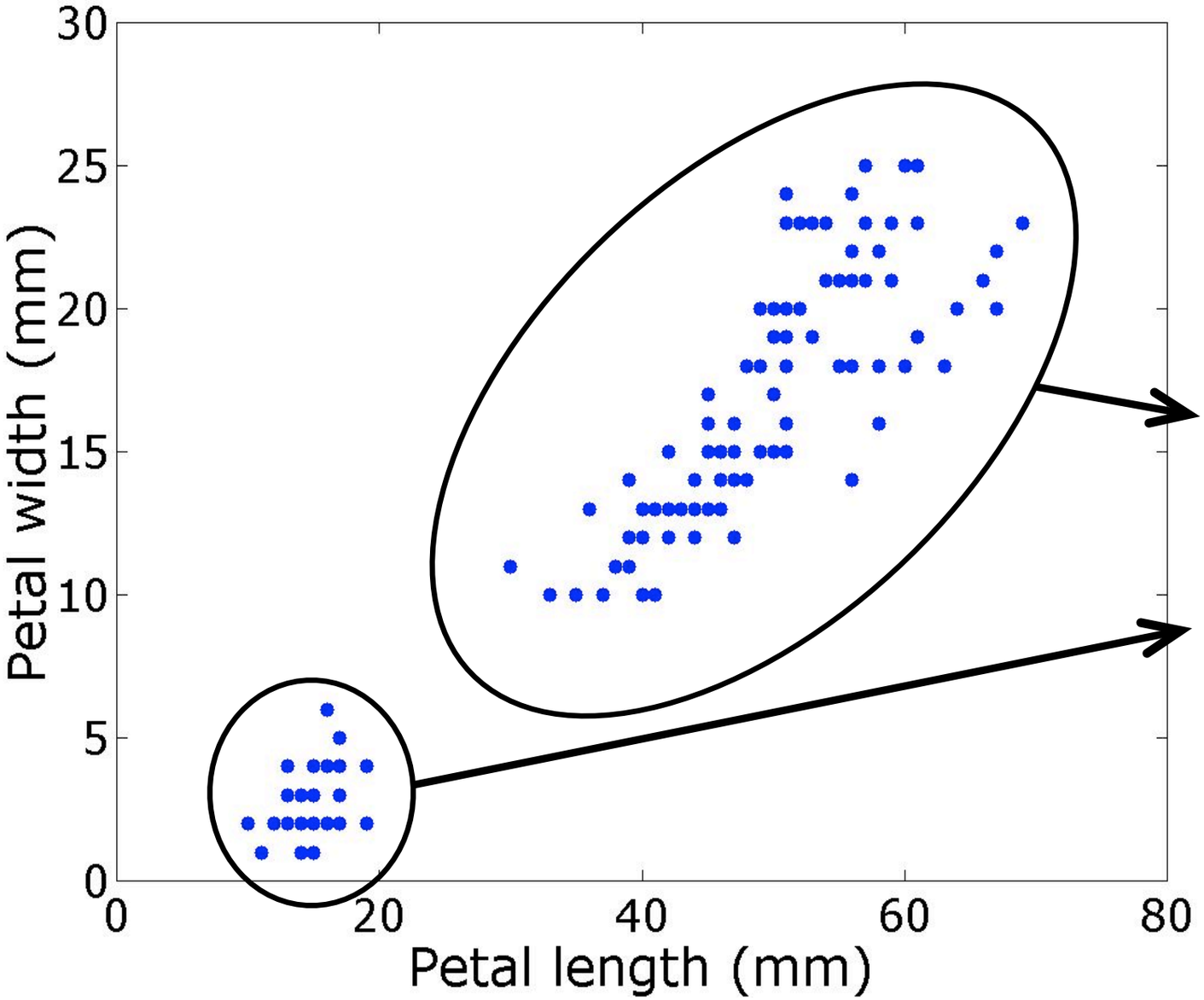
Cluster Analysis

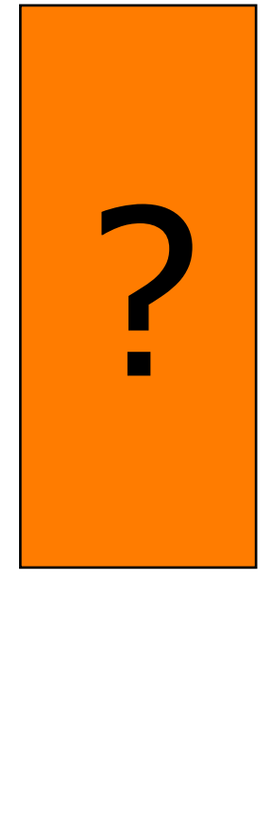
11.1

- 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

E.g. The Iris Data

11.1

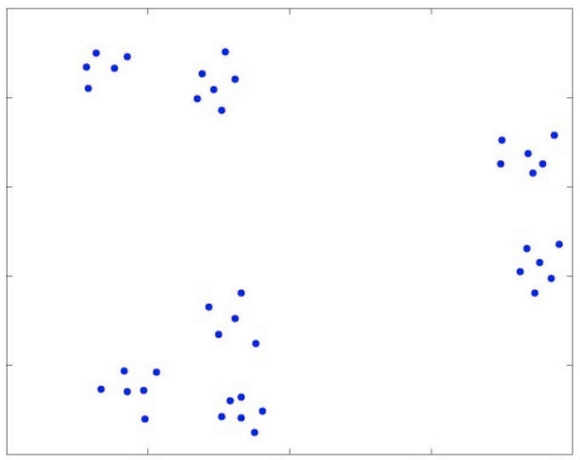




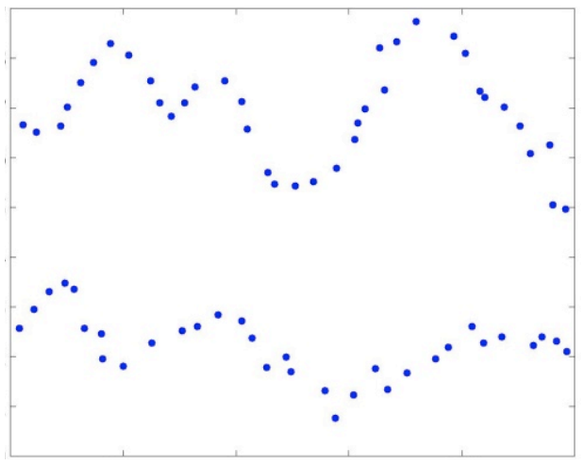
TU Delft

What Makes a Clustering?

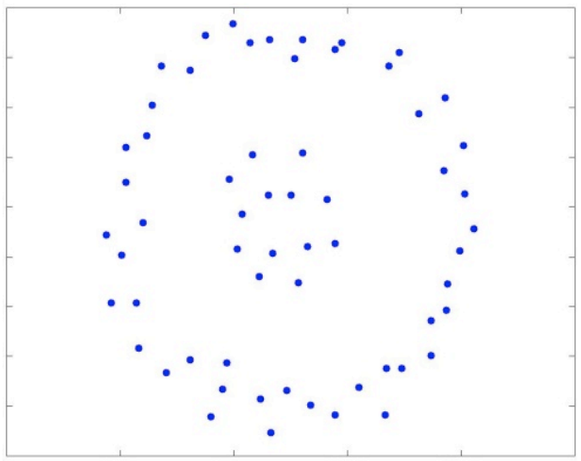
11.1



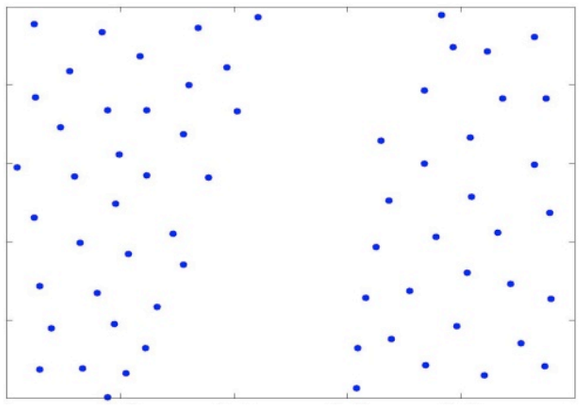
Shape : compact, convex
Separation: large



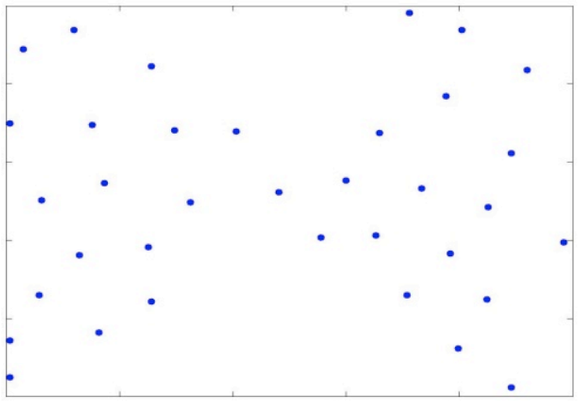
Shape : strings
Separation: large?



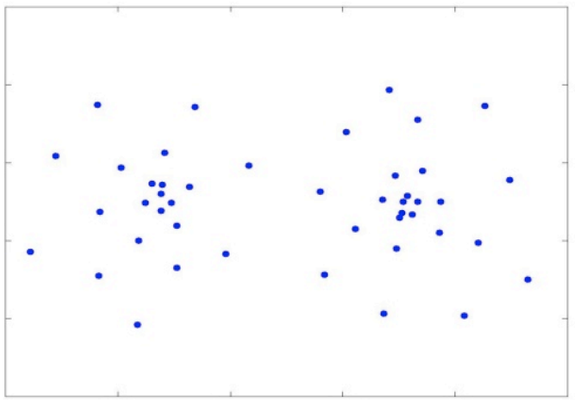
Shape : convex, circular
Separation: large?



Shape : ?
Separation: large?



Shape : loose, convex
Separation: small



Shape : loose, convex
Separation: small

What Makes a Clustering?

11.1

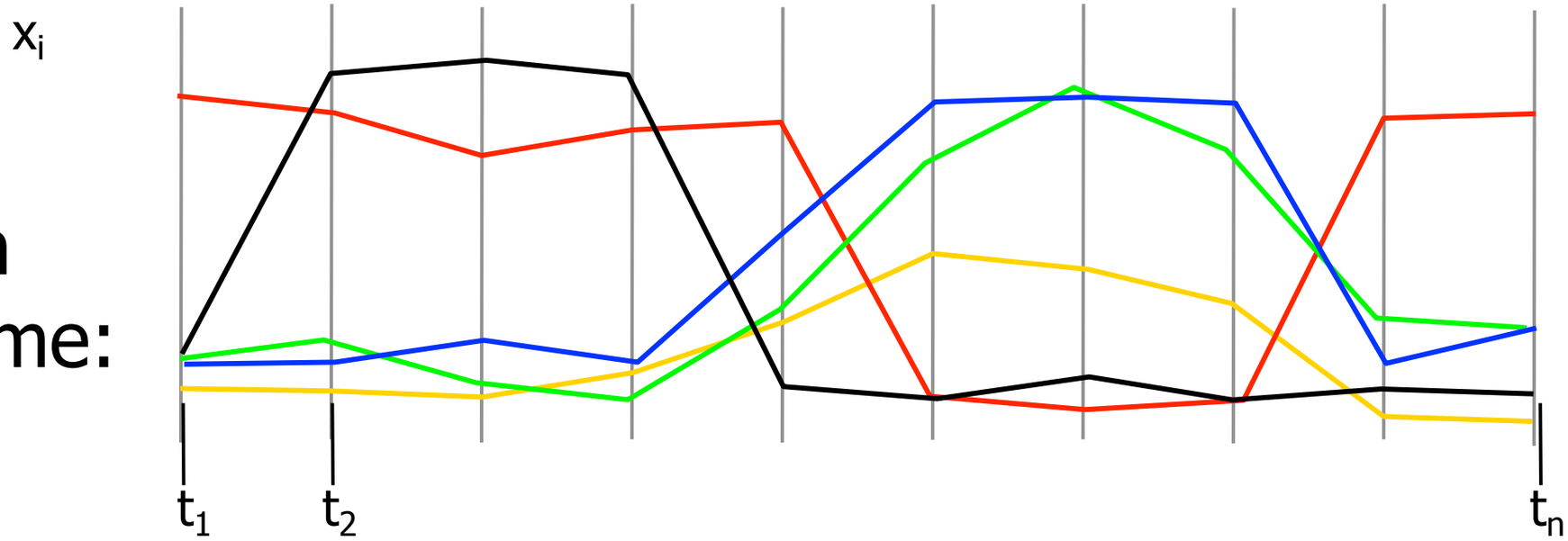
- Clustering : Finding natural groups in data...
  - Which themselves are far apart
  - In which objects are close together

- Define “far apart” and “close together”
  - Need distance or dissimilarity measure
  - Particular choice of measure is crucial

E.g.

11.1

- Gene expression through time:



Euclidean distance	Pearson correlation	Absolute correlation
$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{t=1}^{n} (x_{i,t} - x_{j,t})^2$	$1 - \rho_{ij}$	$1 - \lvert\rho_{ij}\rvert$
d(●,●) < d(●,●)	d(●,●) ≈ d(●,●)	d(●,●) ≈ d(●,●)
d(●,●) << d(●,●)	d(●,●) << d(●,●)	d(●,●) ≈ d(●,●)
d(●,●) << d(●,●)	d(●,●) << d(●,●)	d(●,●) << d(●,●)

TU Delft

Distance measure

- Typically, we need to define a distance between objects first:
- 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

11.2

- 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, ...)

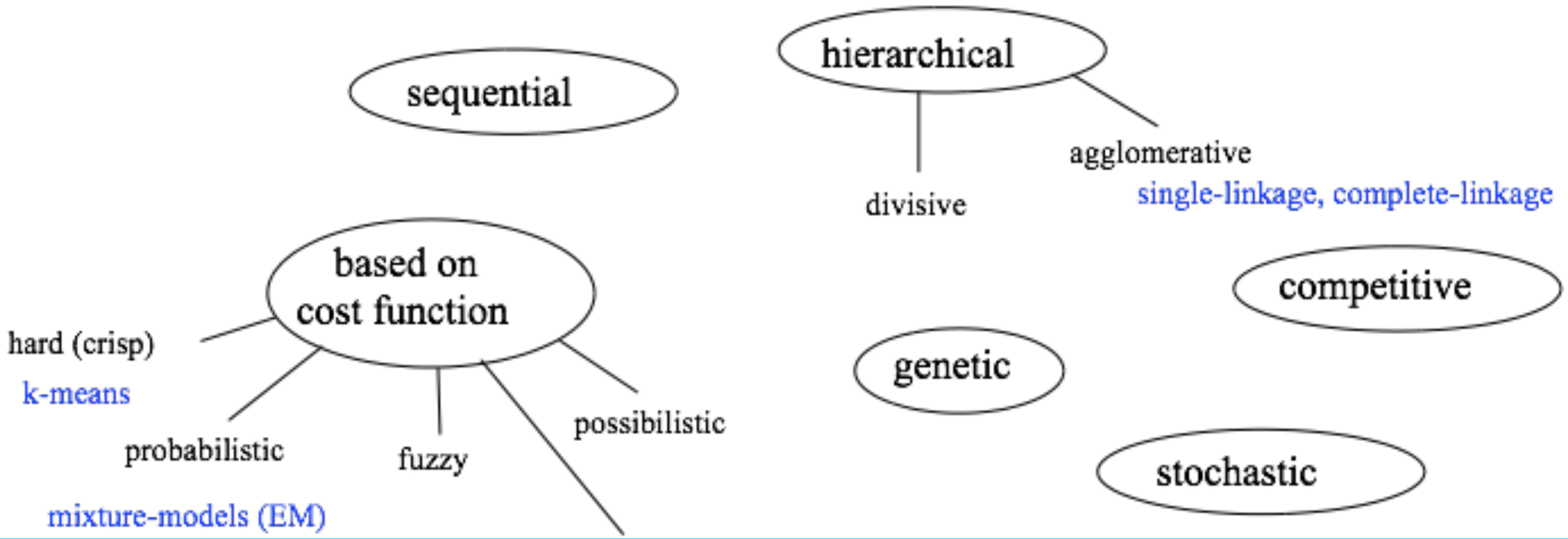
Hard vs. Soft

- Hard assignments
  - K-Means
  - Hierarchical clustering

- Soft assignments
  - Fuzzy C-means
  - Probabilistic mixture models

Clustering Clustering Algorithms

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



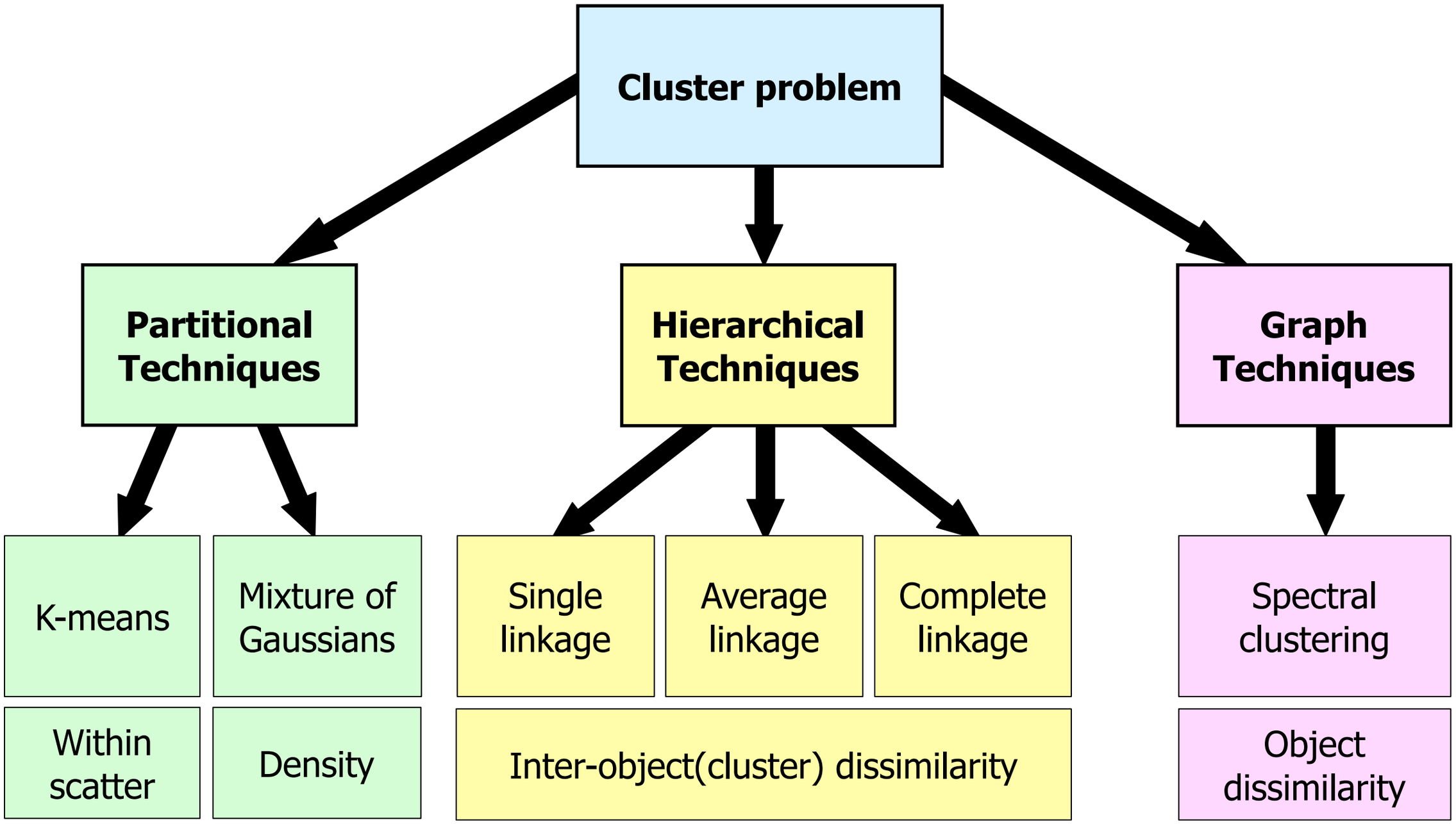
Chapters 11-15 from the book…

- 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)

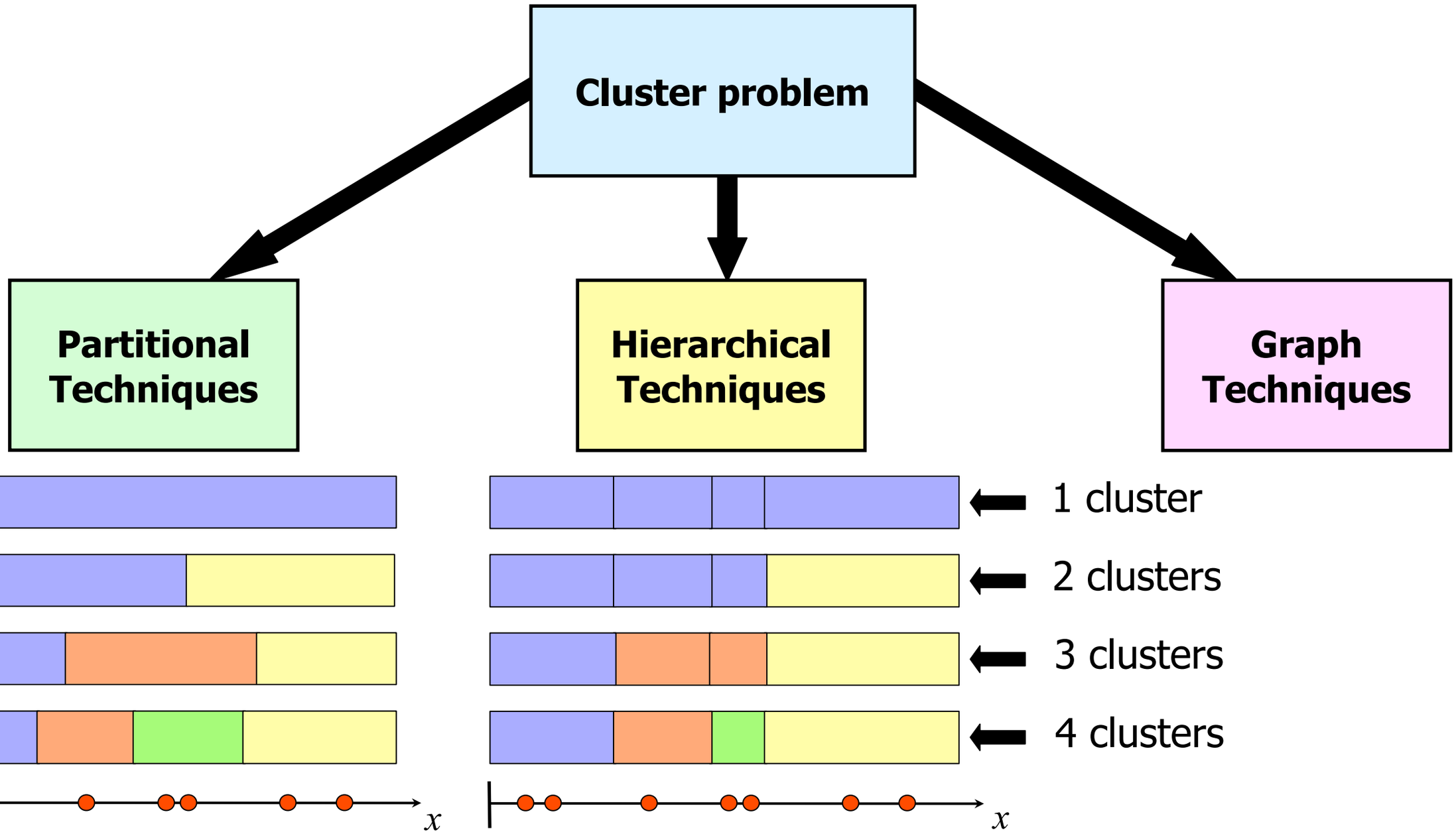
One Way or The Other…

- There is no such thing as an objective clustering

Clustering techniques



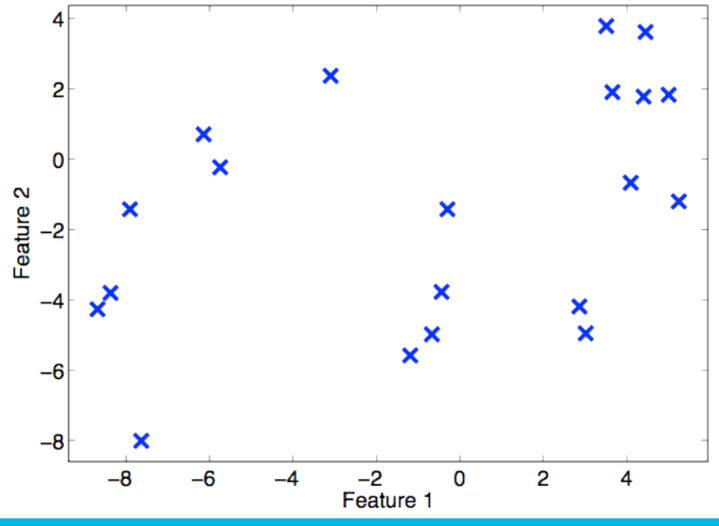
Clustering techniques (2)



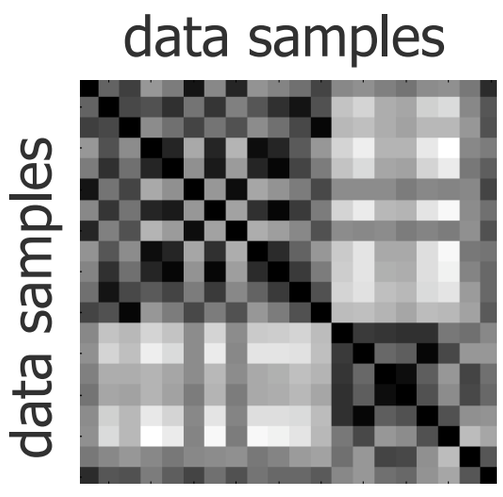
Agglomerative Hierarchical Clustering

Ch 13

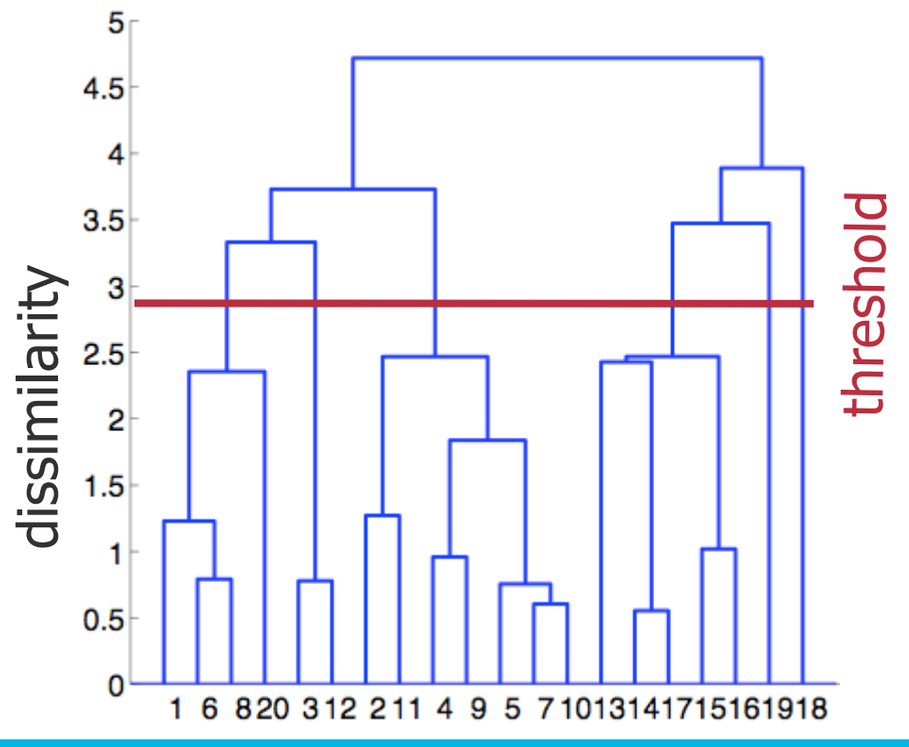
- Starting from individual observations, produce sequence of clusterings of increasing size
- At each level, two clusters chosen by criterion are merged



2D scatter plot of data



dissimilarity matrix



dendrogram

TU Delft

Agglomerative Hierarchical Clustering

13.2

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

11.2.4

- 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

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

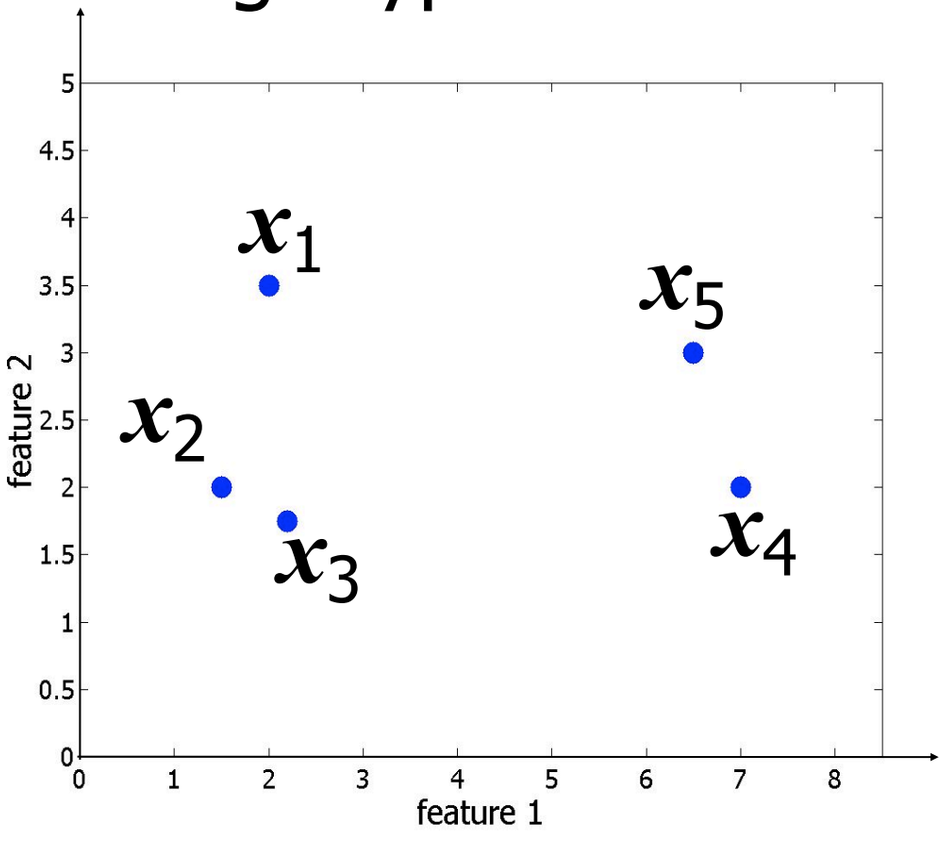
Hierarchical clustering

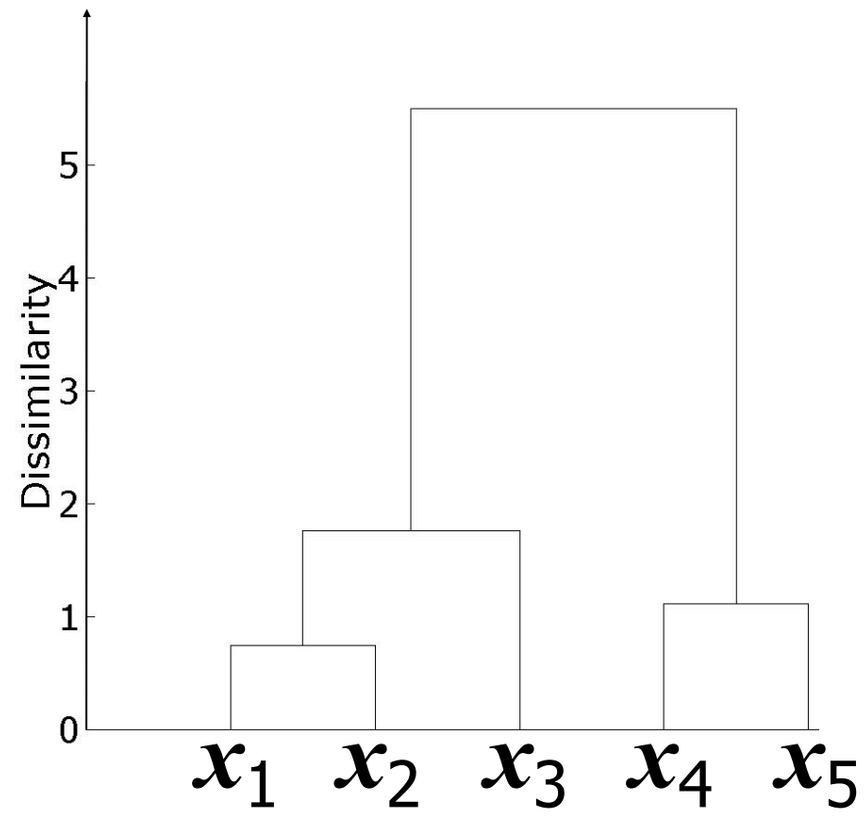
Input:

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

Output:

- dendrogram



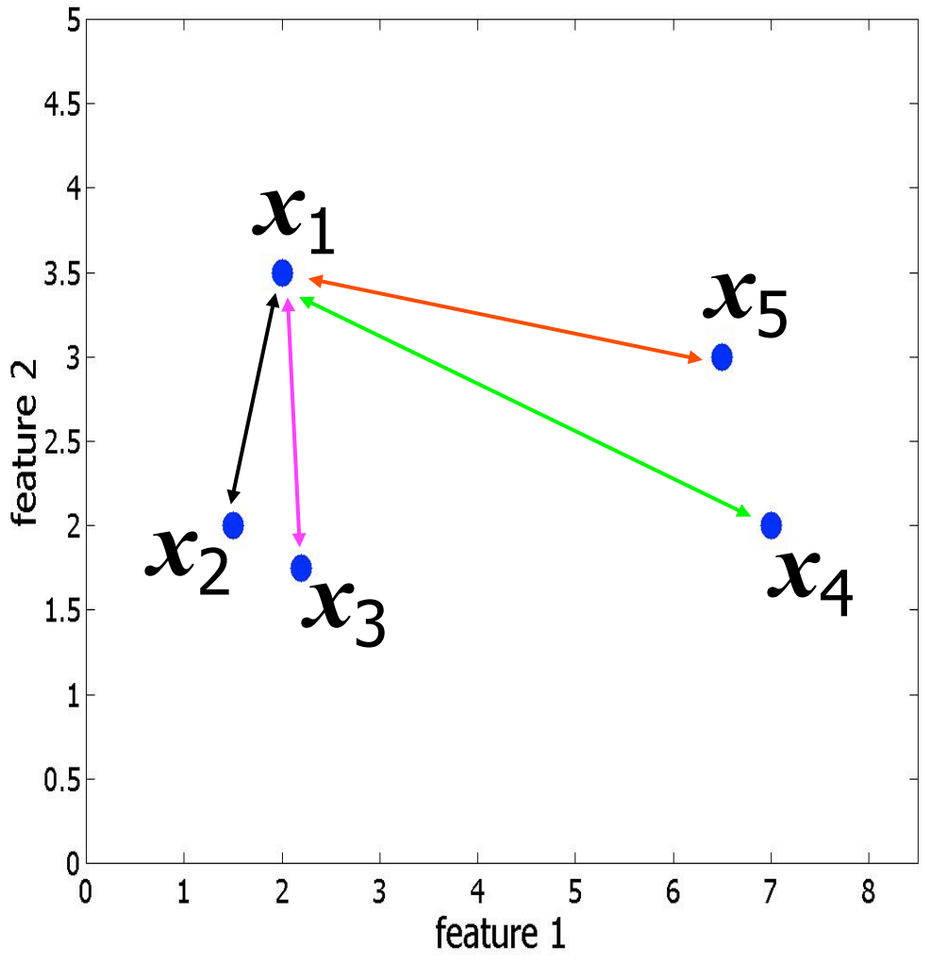


TU Delft

Hierarchical clustering (2)

- **Step 0:** all objects are a cluster:

Dataset



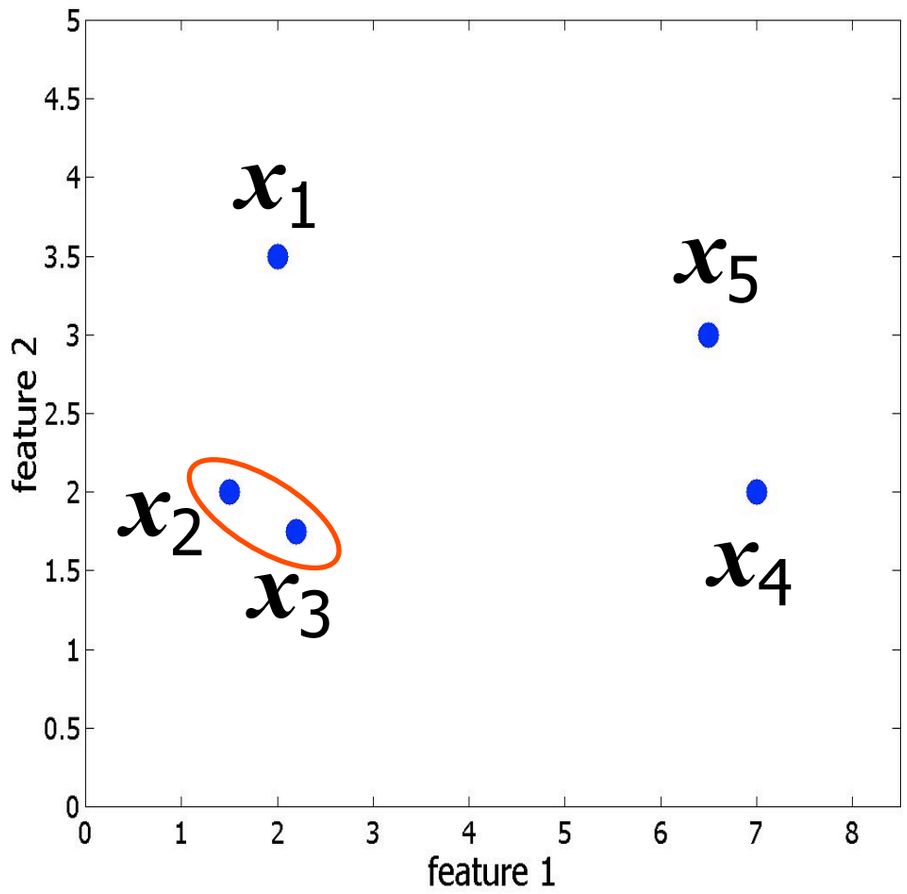
(Euclidean) distance matrix, $\boldsymbol{D}$

	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	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 (3)

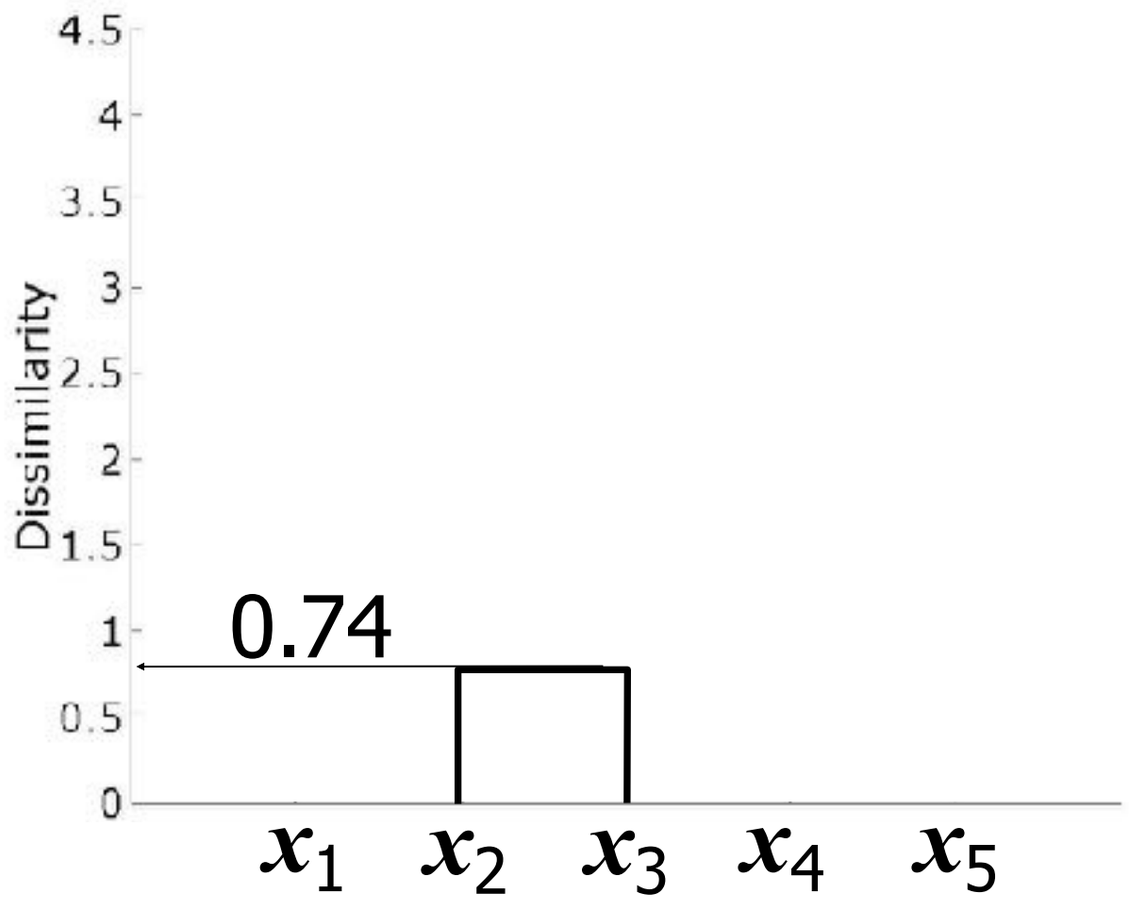
- **Step 1:**
  Find the most similar pair: $\min_{(i,j)}\{d(i,j)\} = d(2,3)$



	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	0.74	5.50	5.10
x_3			0.00	4.81	4.48
x_4				0.00	1.12
x_5					0.00

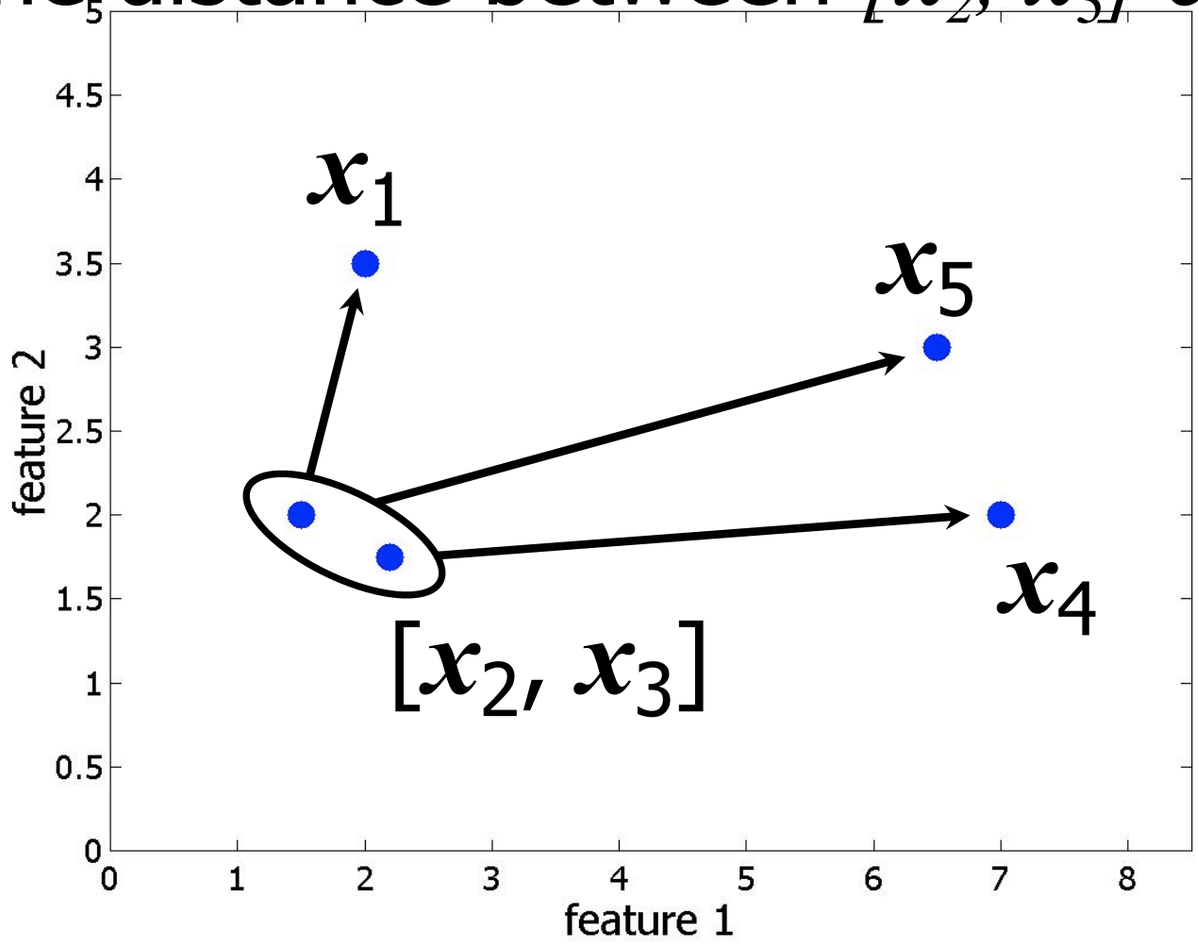
Hierarchical clustering (4)

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



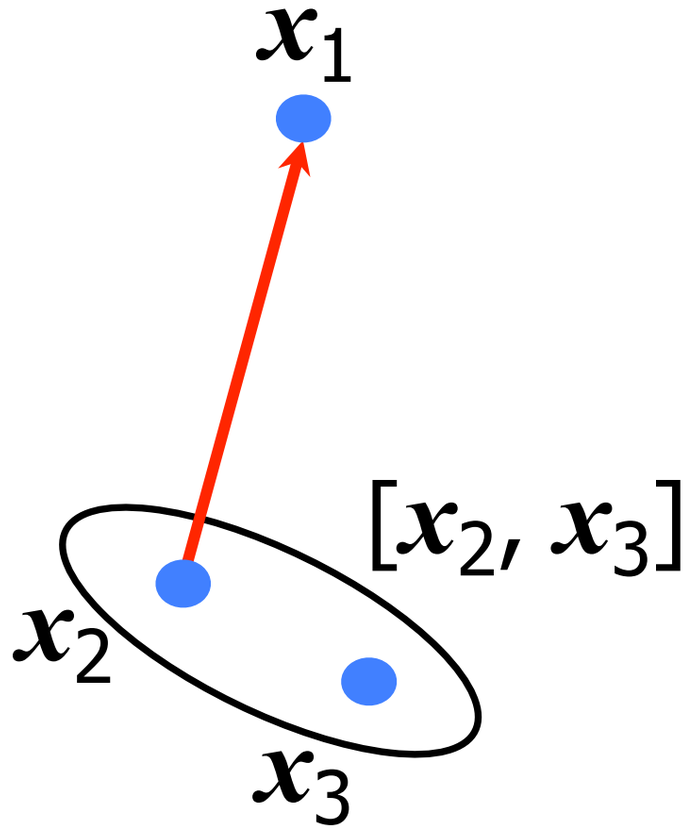
Hierarchical clustering (5)

- **Step 3:**
  Recompute $\boldsymbol{D}$ –
  what is the distance between $[x_2, x_3]$ and the rest?



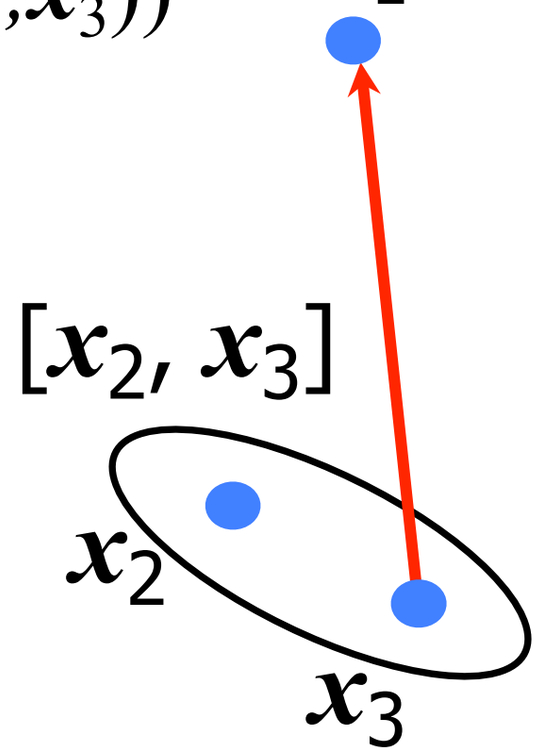
Hierarchical clustering (6)

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



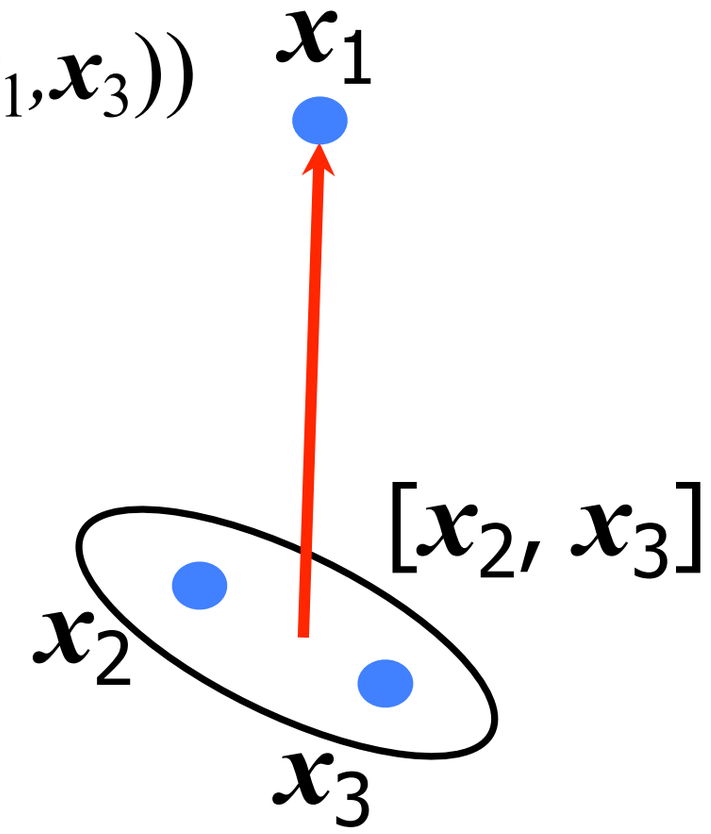
Hierarchical clustering (7)

- **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: $d([\boldsymbol{x}_2,\boldsymbol{x}_3],\boldsymbol{x}_1) =$
  $\text{mean}(d(\boldsymbol{x}_1,\boldsymbol{x}_2),d(\boldsymbol{x}_1,\boldsymbol{x}_3))$



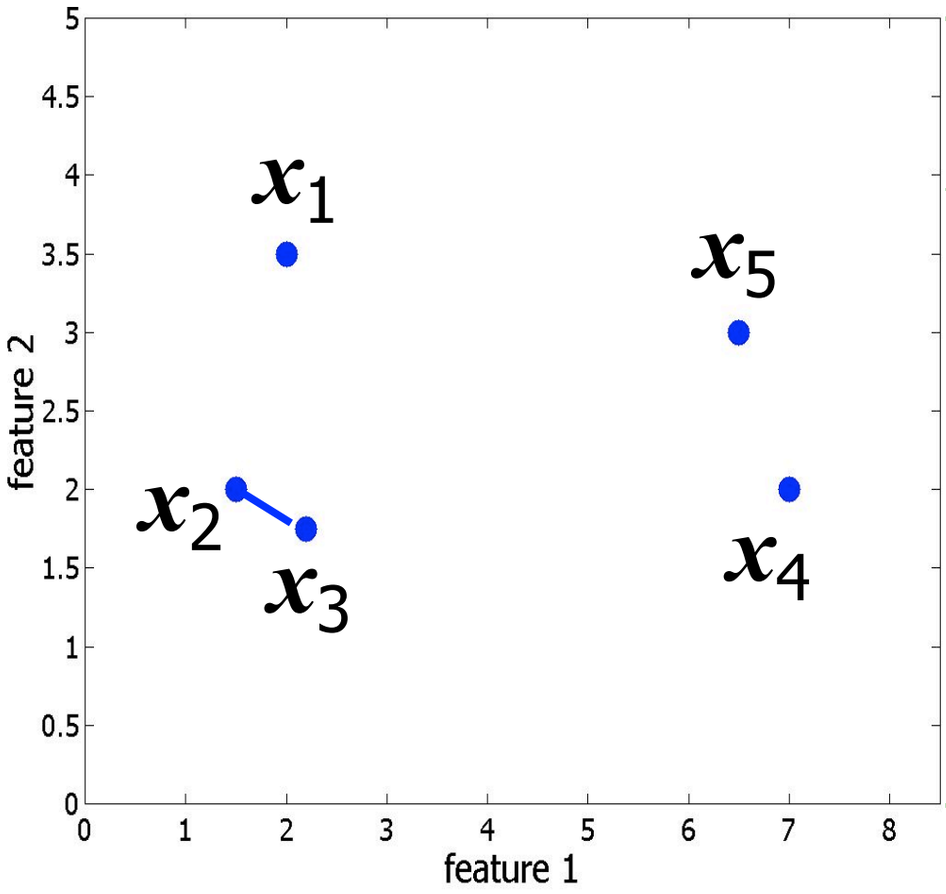
Hierarchical clustering (9)

- **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)

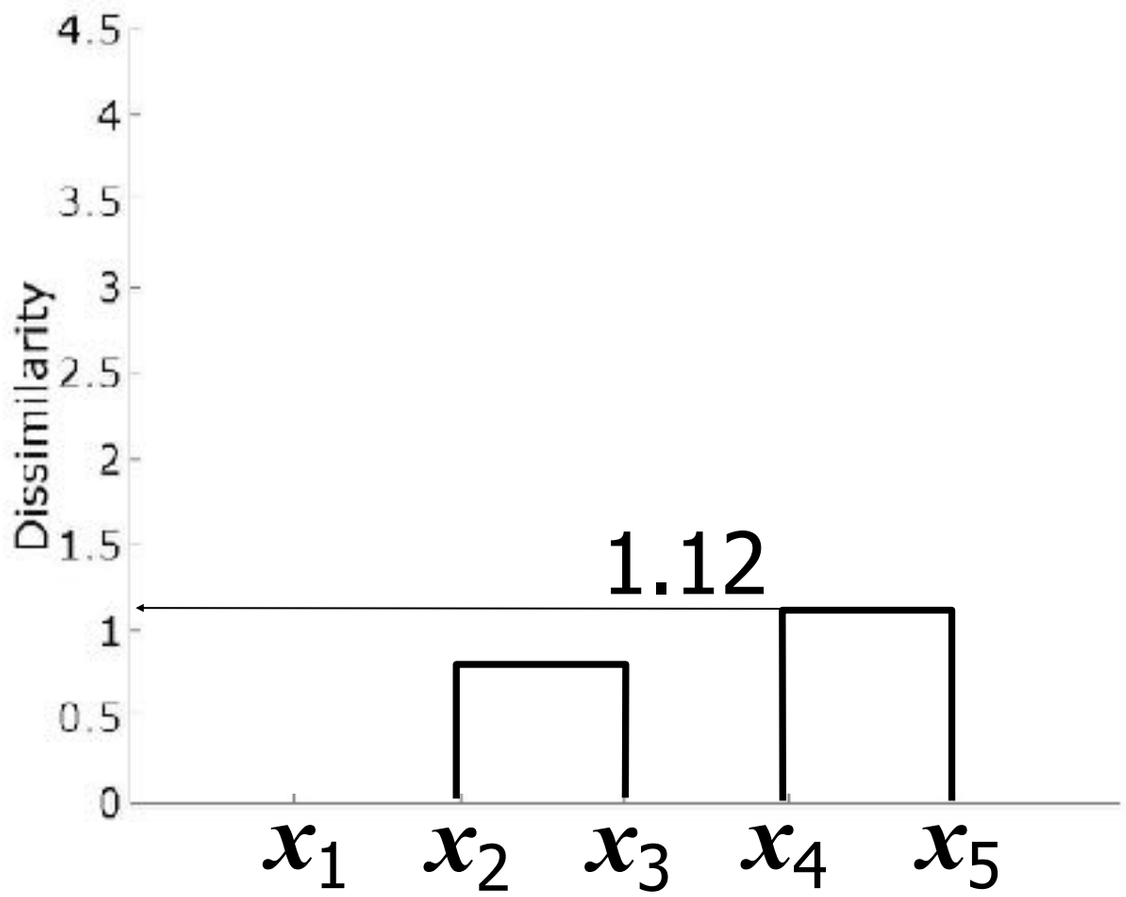
- **Repeat, step 1:**
  Find the most similar pair of objects: $\min_{(i,j)}\{d(i,j)\} = d(4,5)$



	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	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, $[x_4, x_5]$;



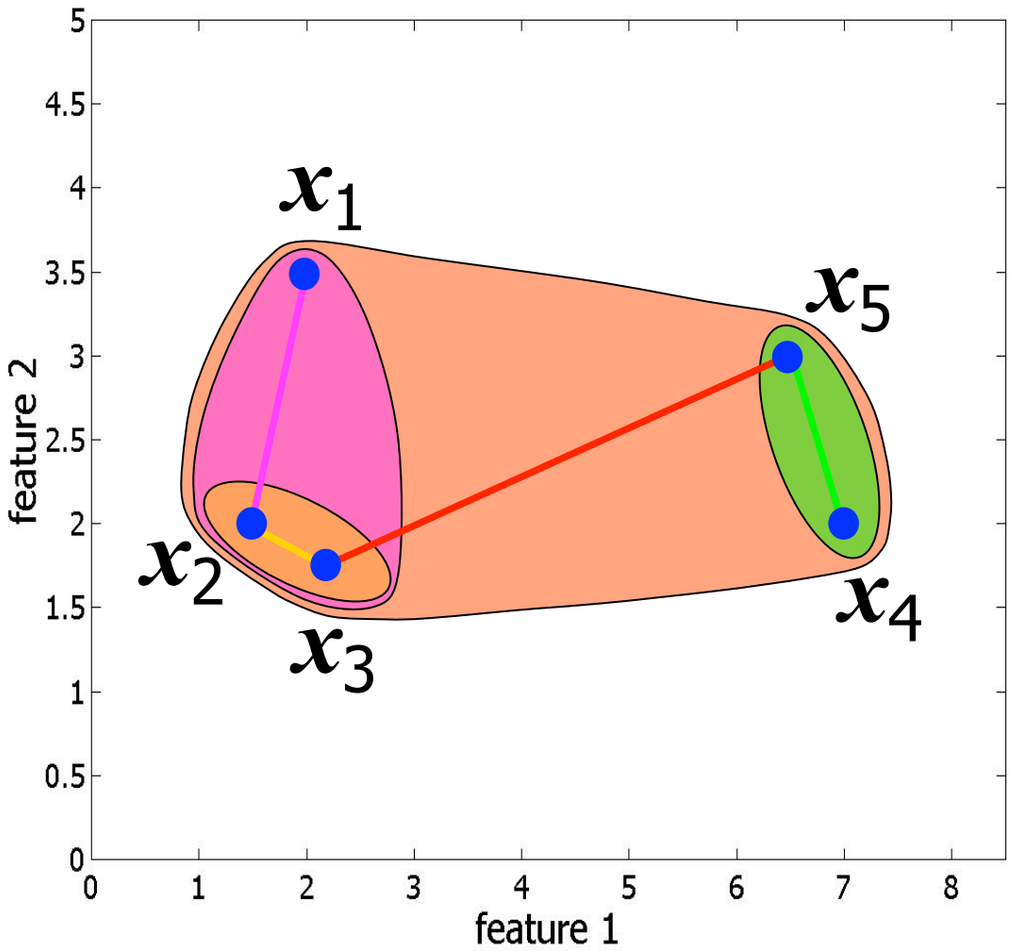
Hierarchical clustering (12)

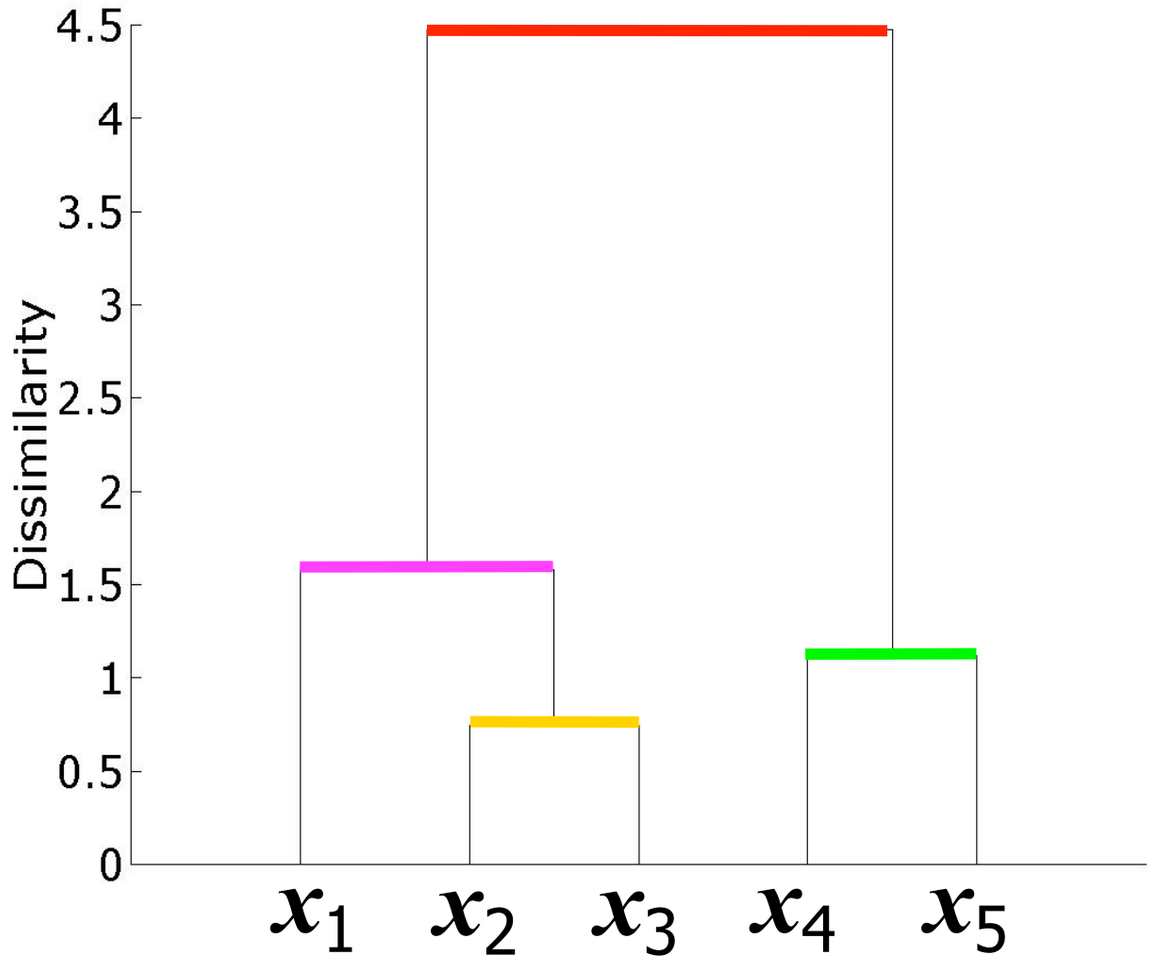
- **Repeat, step 3:**
  Recompute D (single linkage):

	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

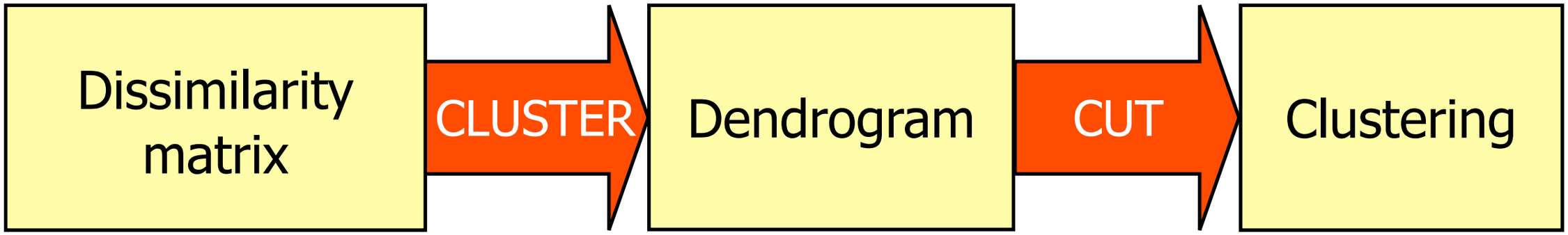
Hierarchical clustering (13)

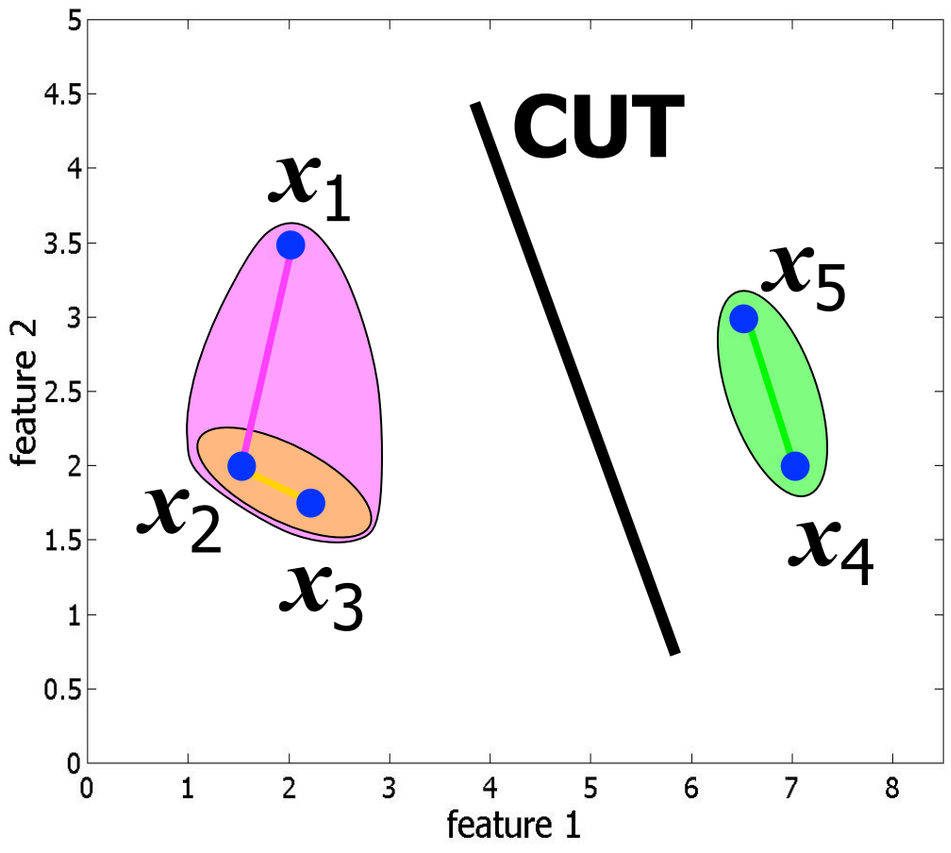
- Repeat steps 1-3 until a single cluster remains...

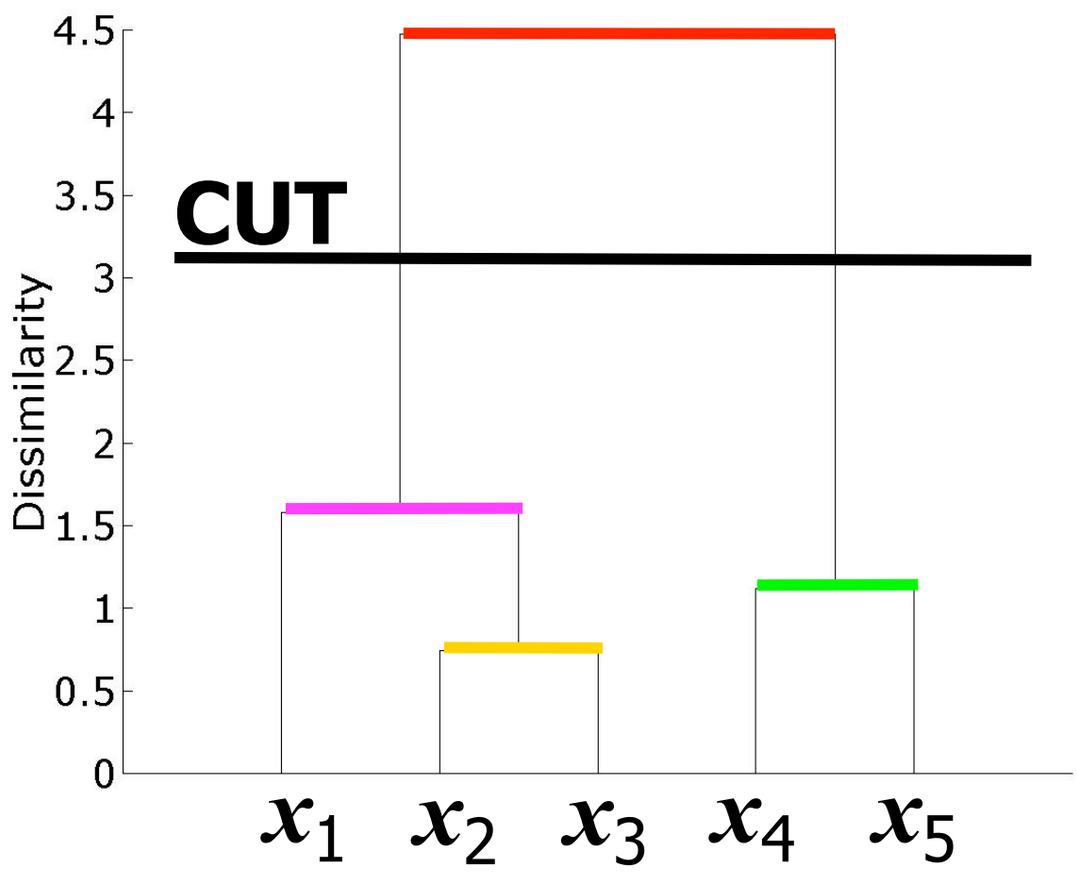




Hierarchical clustering (14)



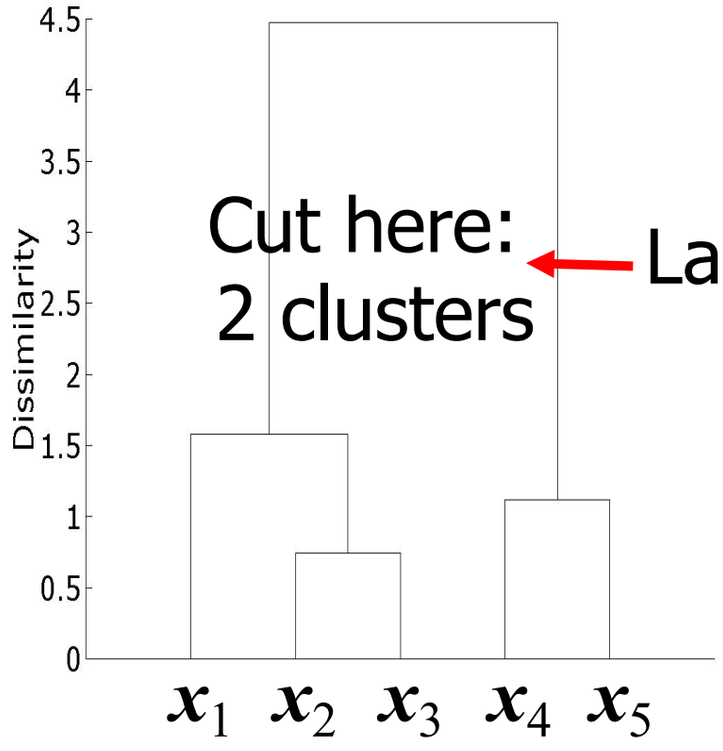


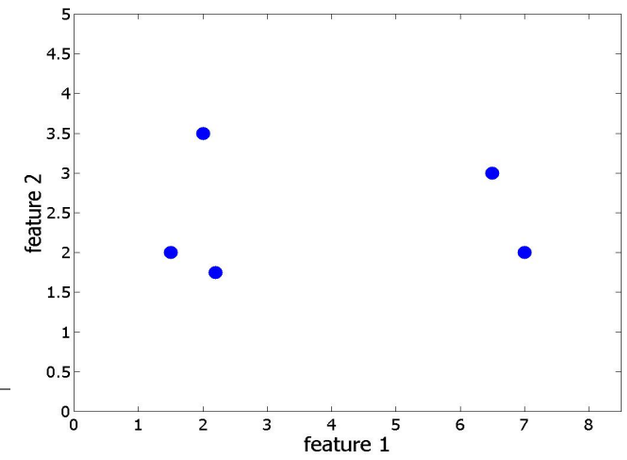


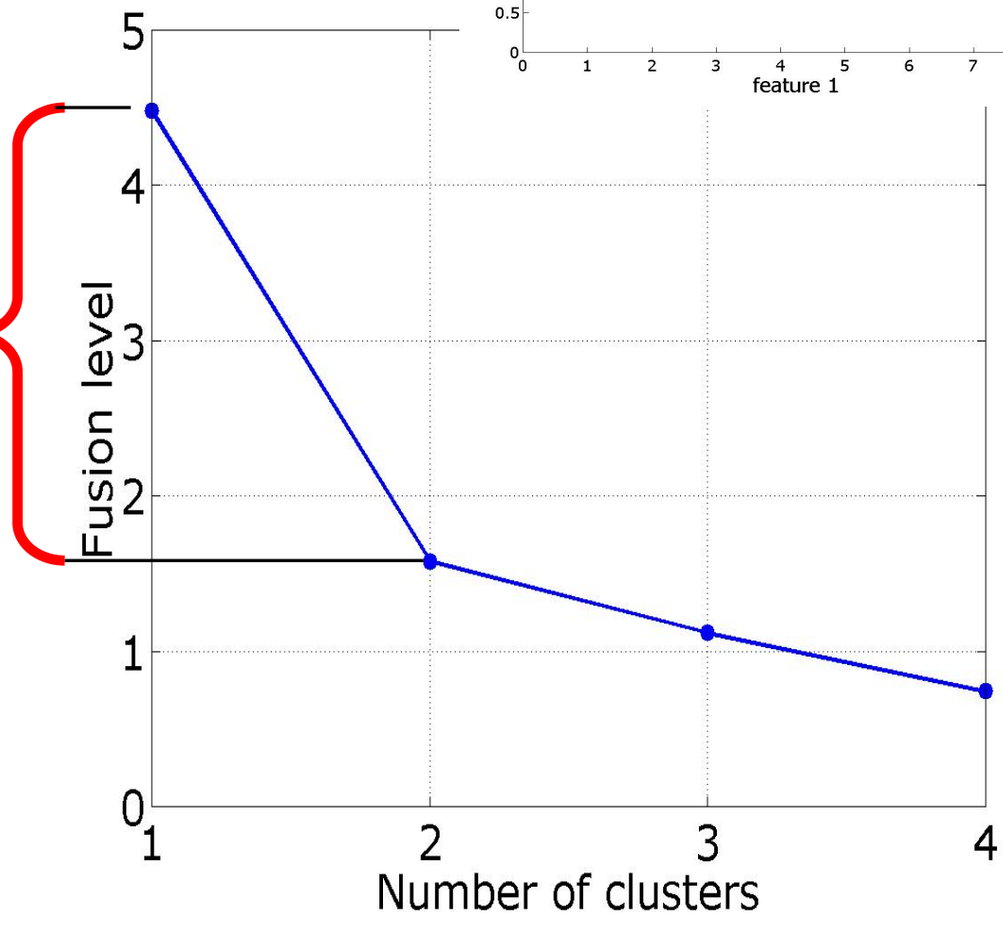
TU Delft

Fusion graph

- Heuristic approach: fusion level



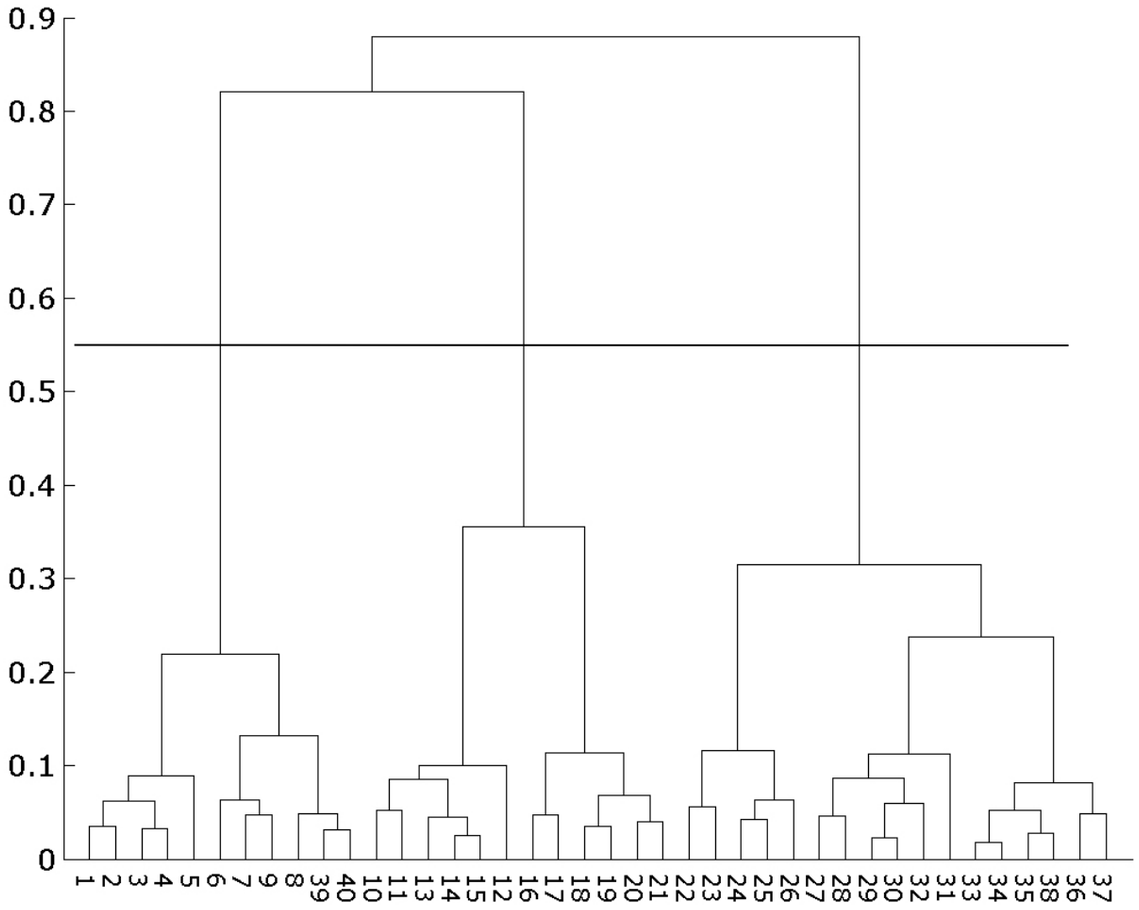


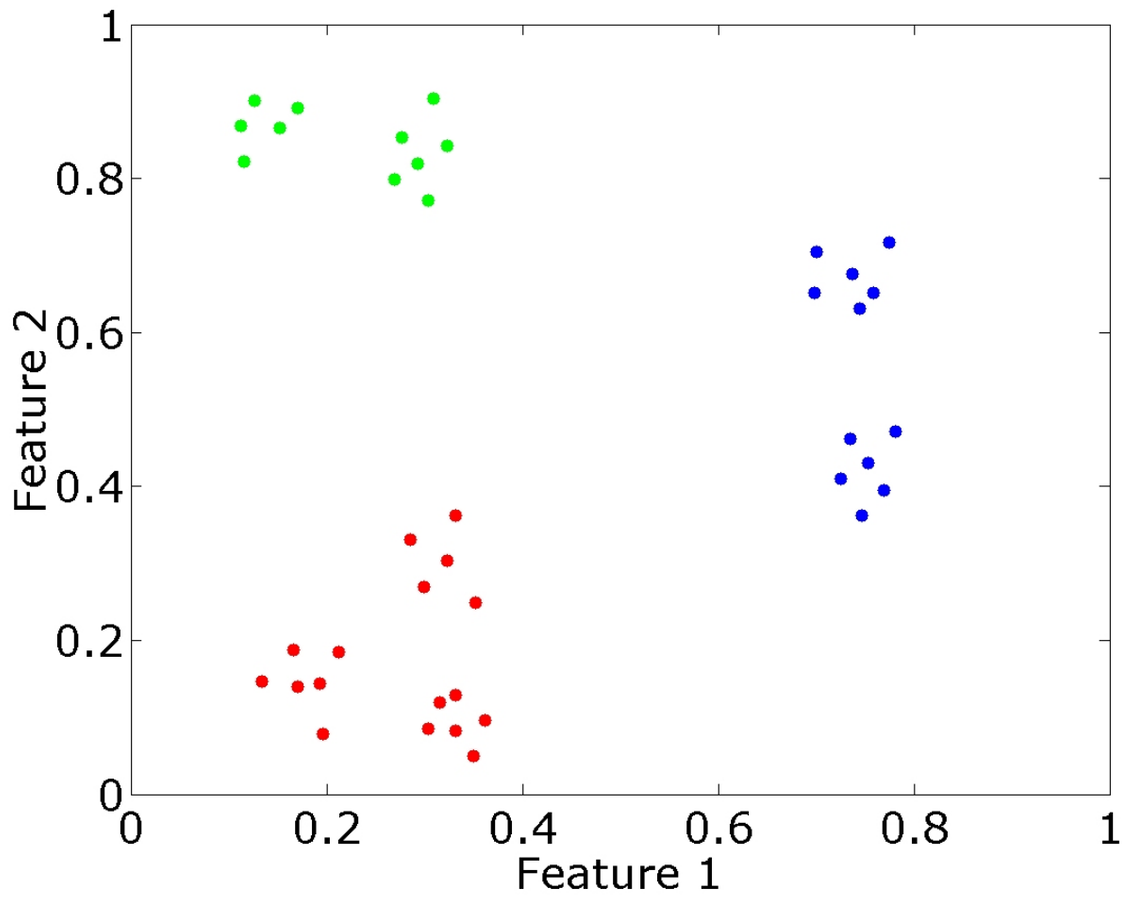


TU Delft

Hierarchical clustering examples (1)

Euclidean, complete linkage

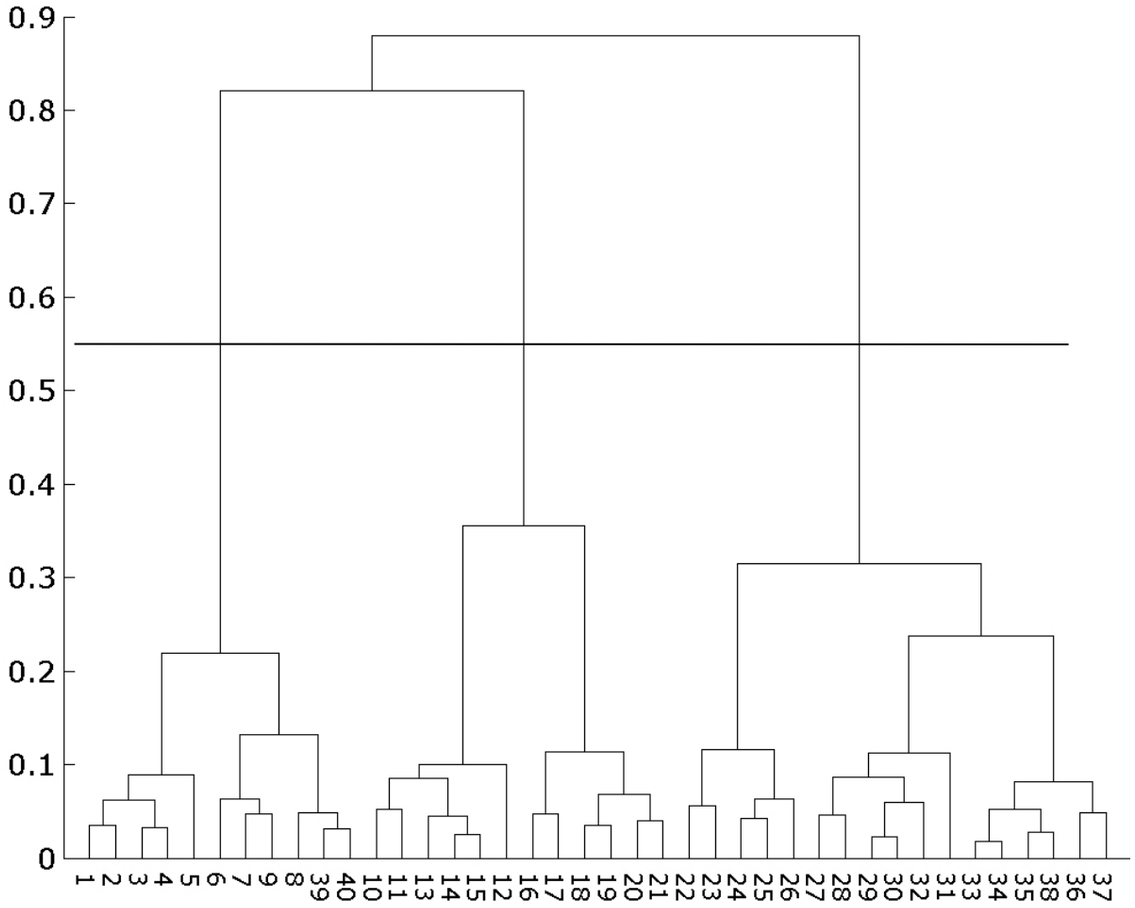


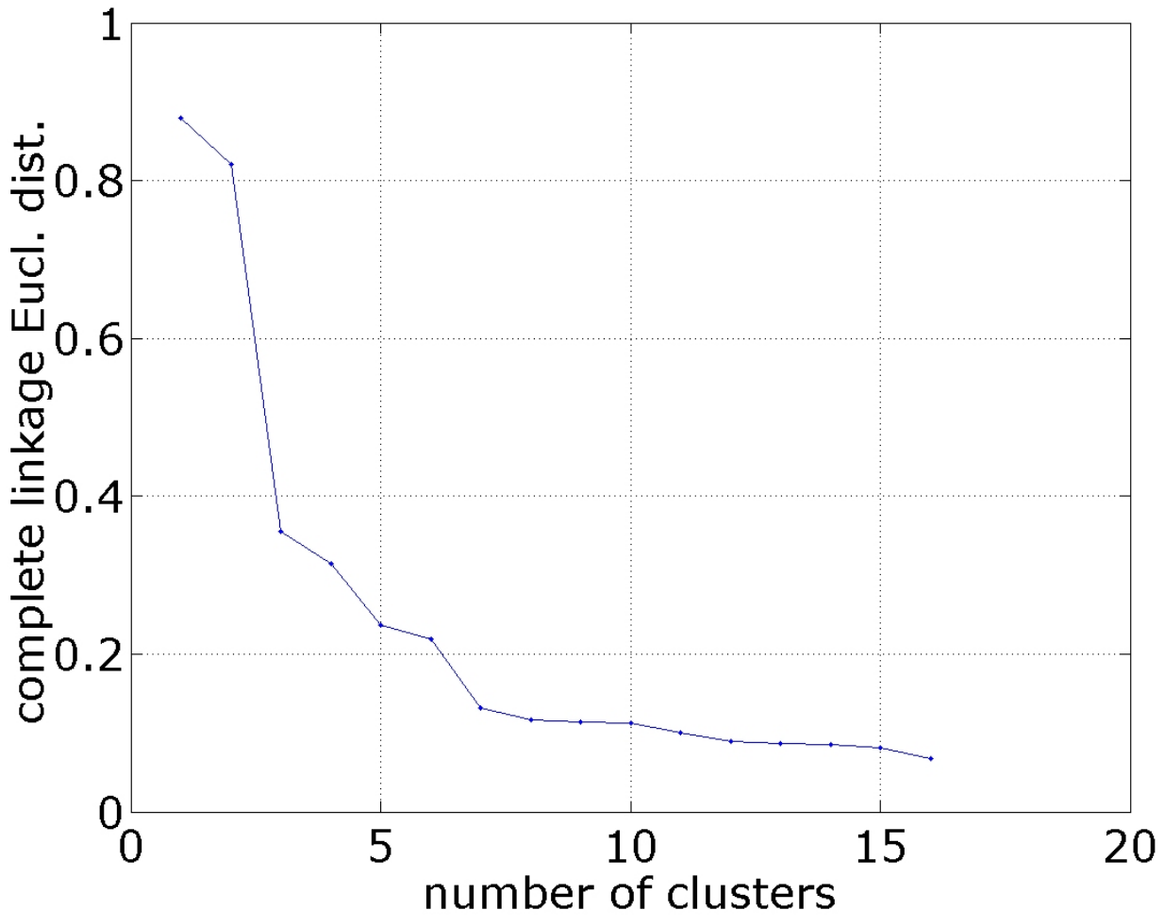


TU Delft

Hierarchical clustering examples (1)

Euclidean, complete linkage

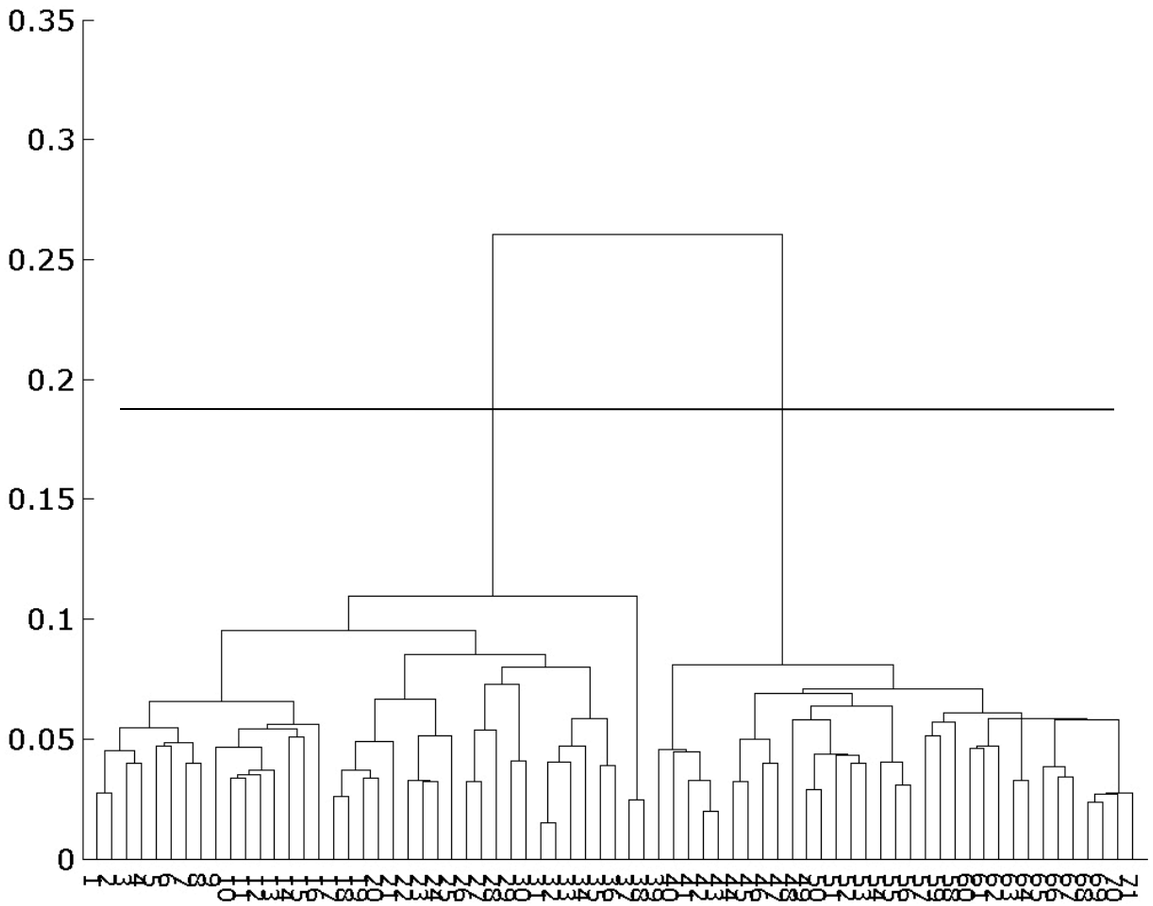


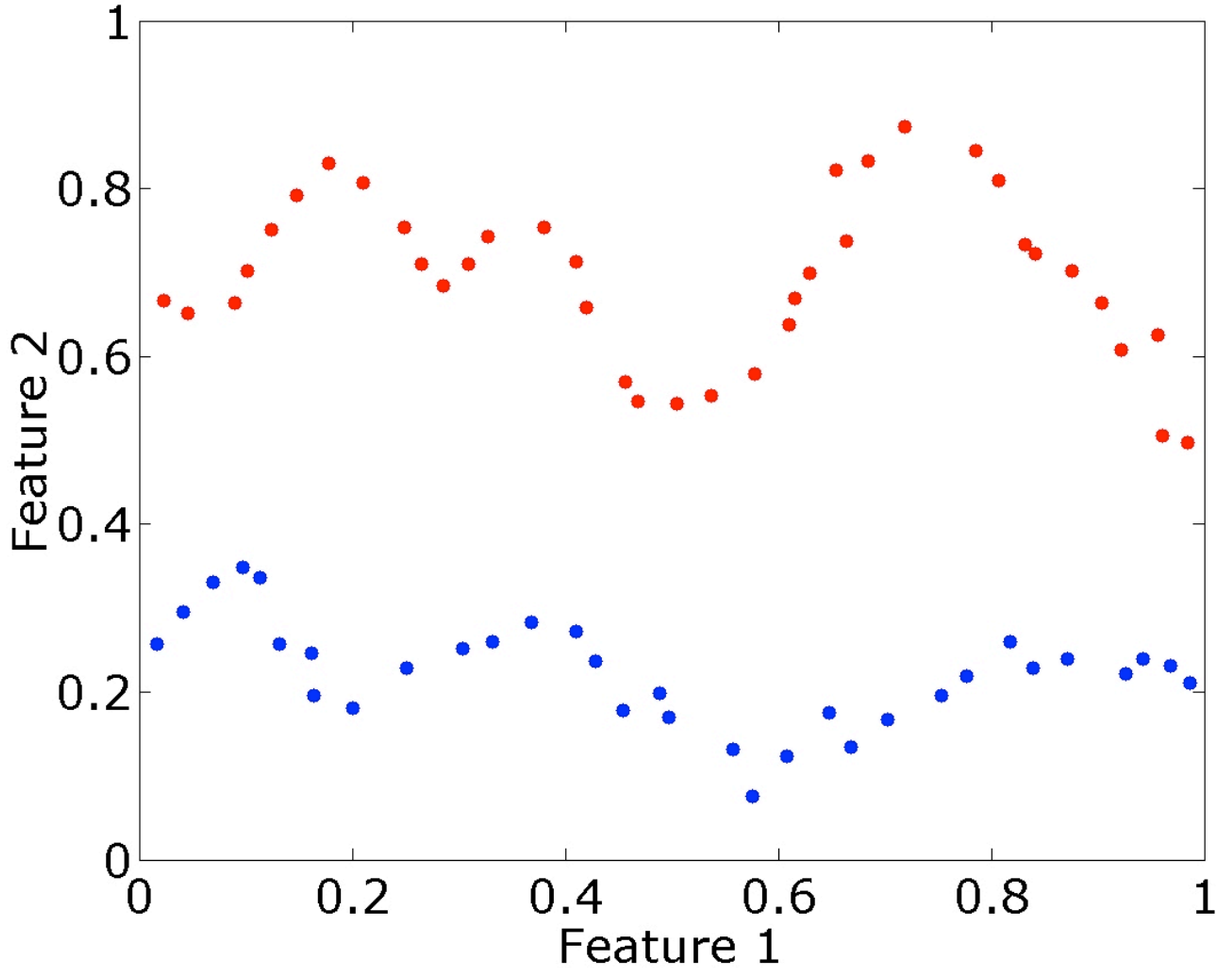


TU Delft

Hierarchical clustering examples (2)

Euclidean, single linkage

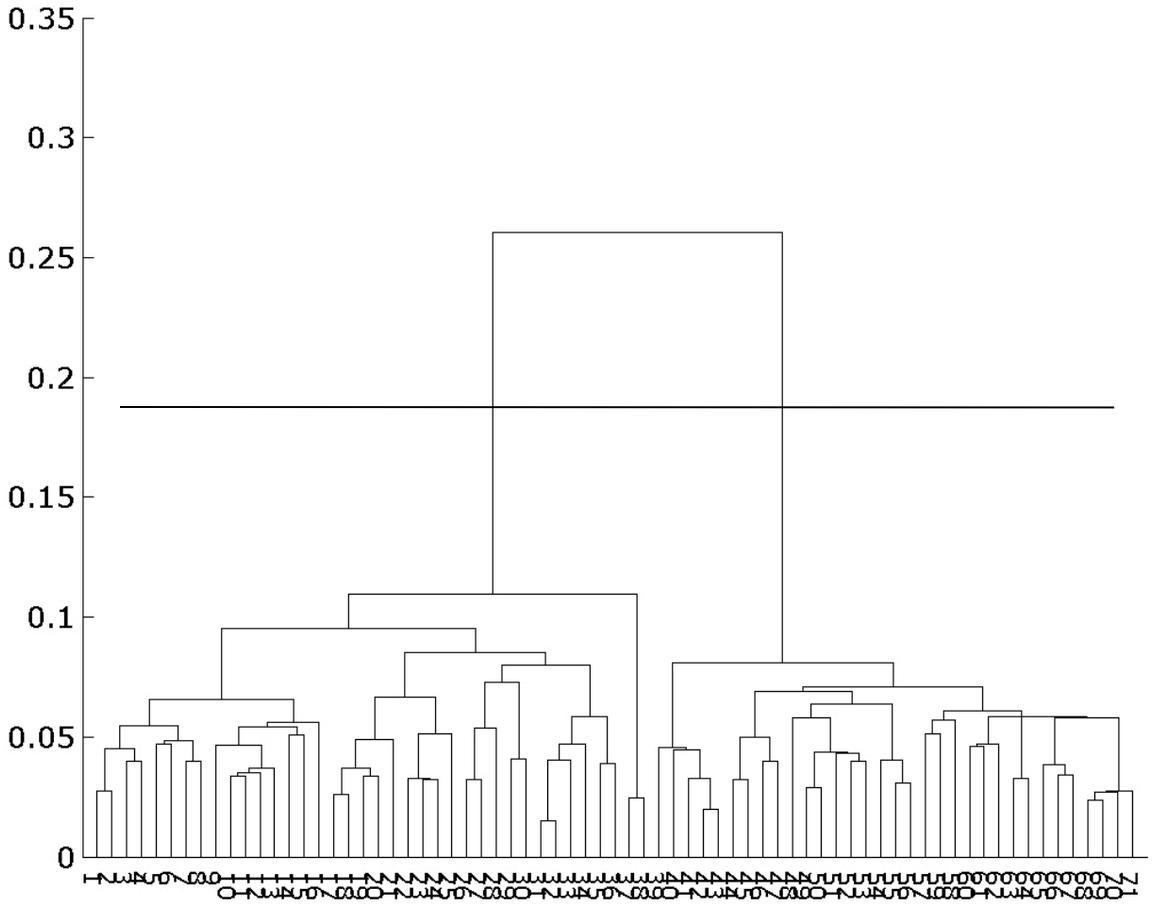


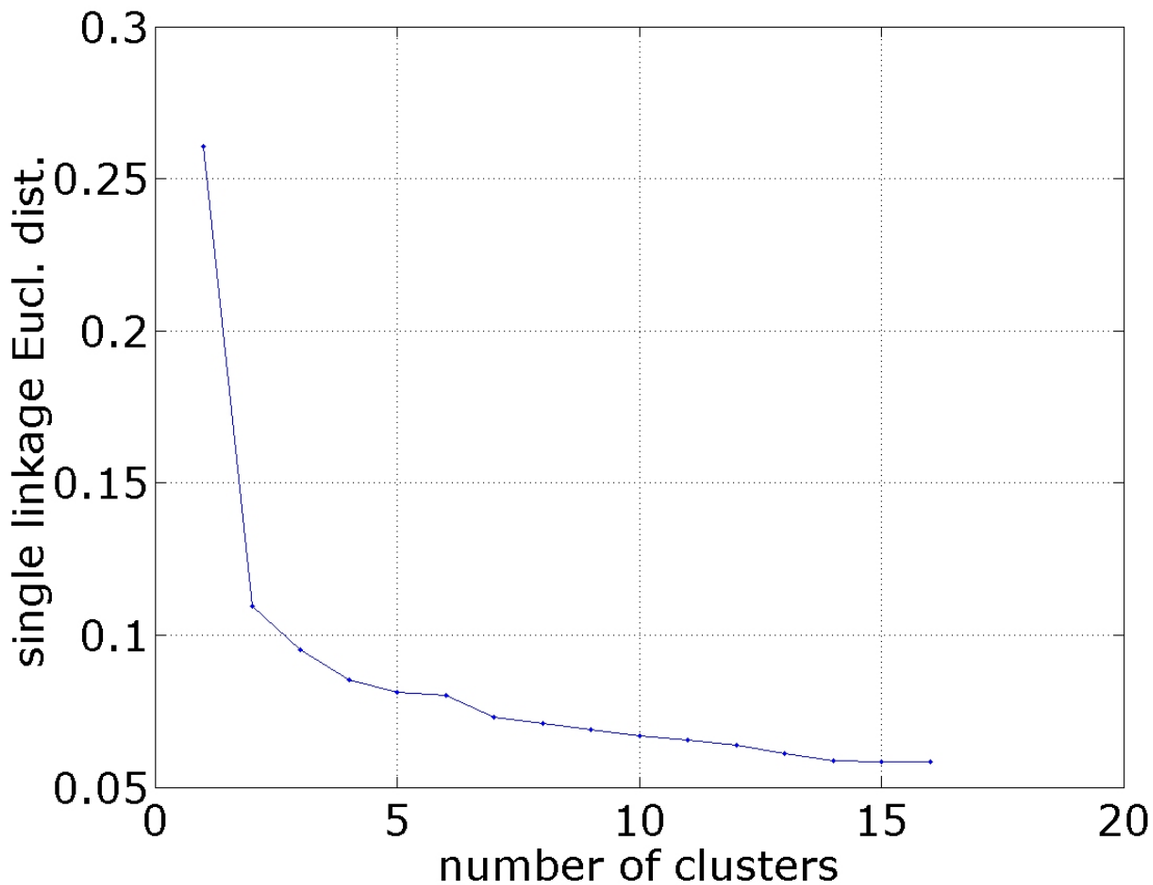


TU Delft

Hierarchical clustering examples (2)

Euclidean, single linkage

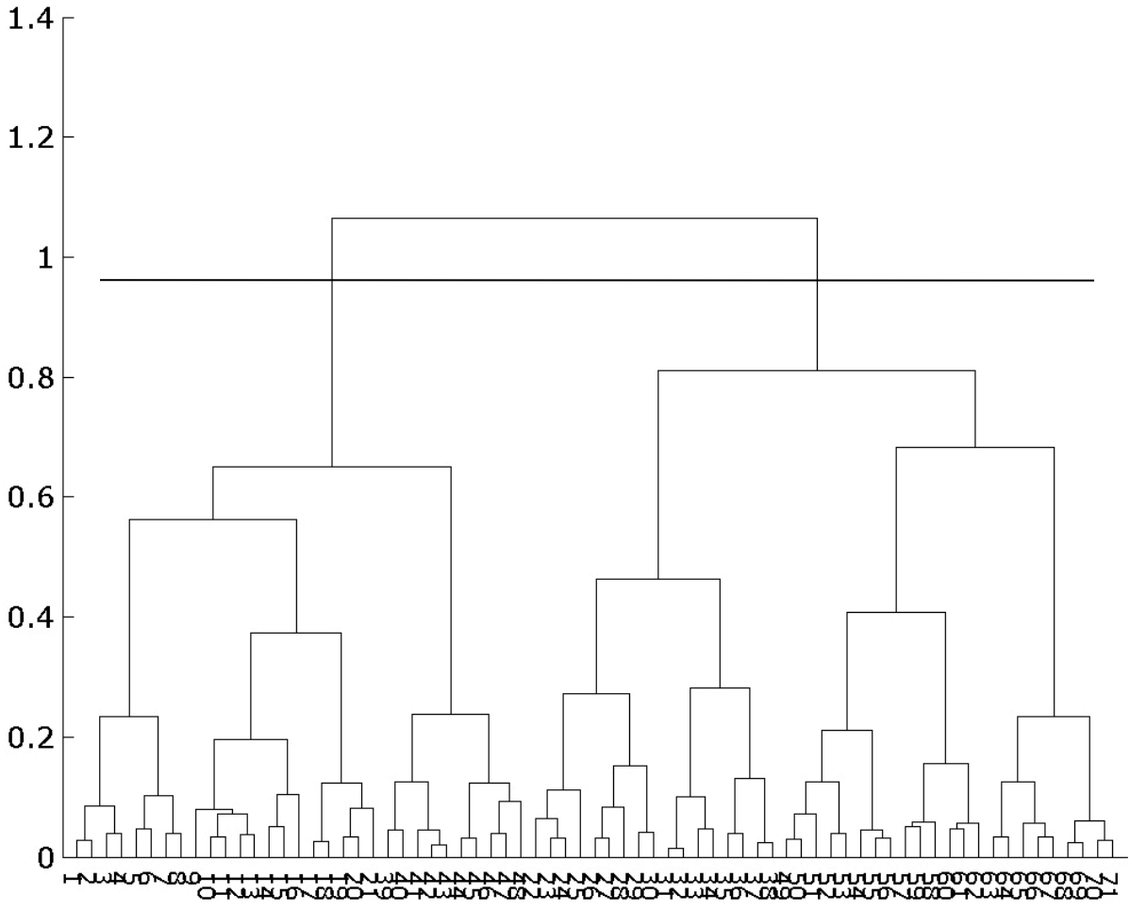


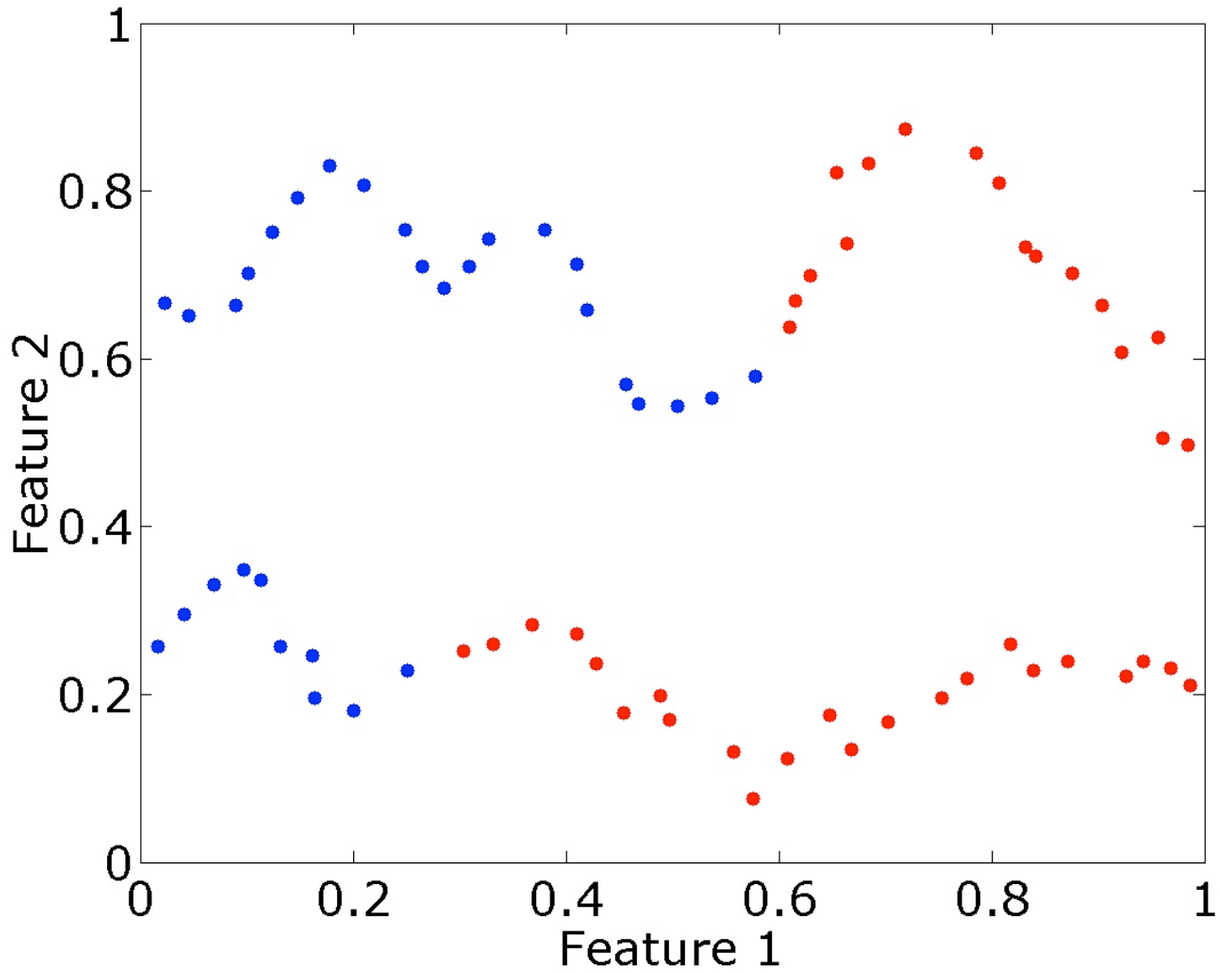


TU Delft

Hierarchical clustering examples (3)

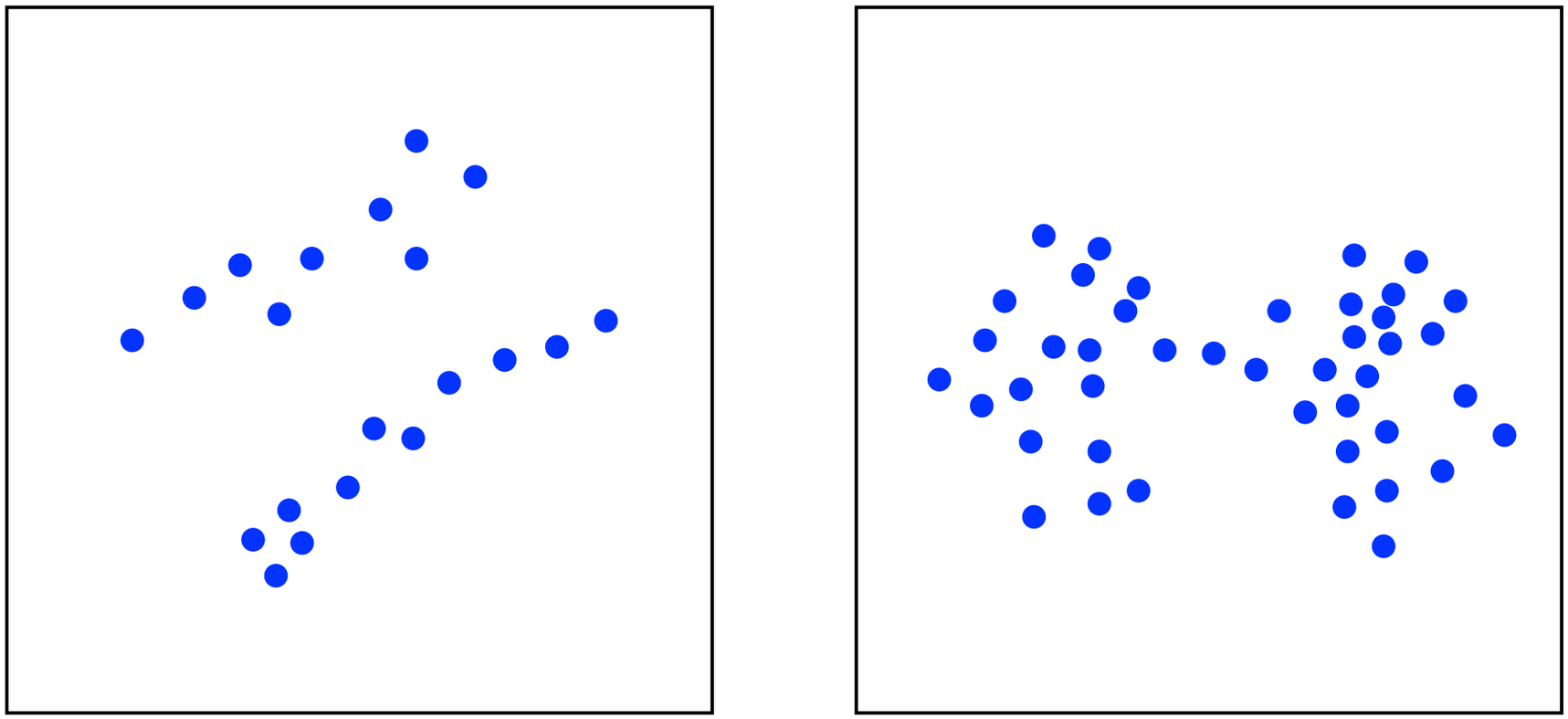
Euclidean, complete linkage





TU Delft

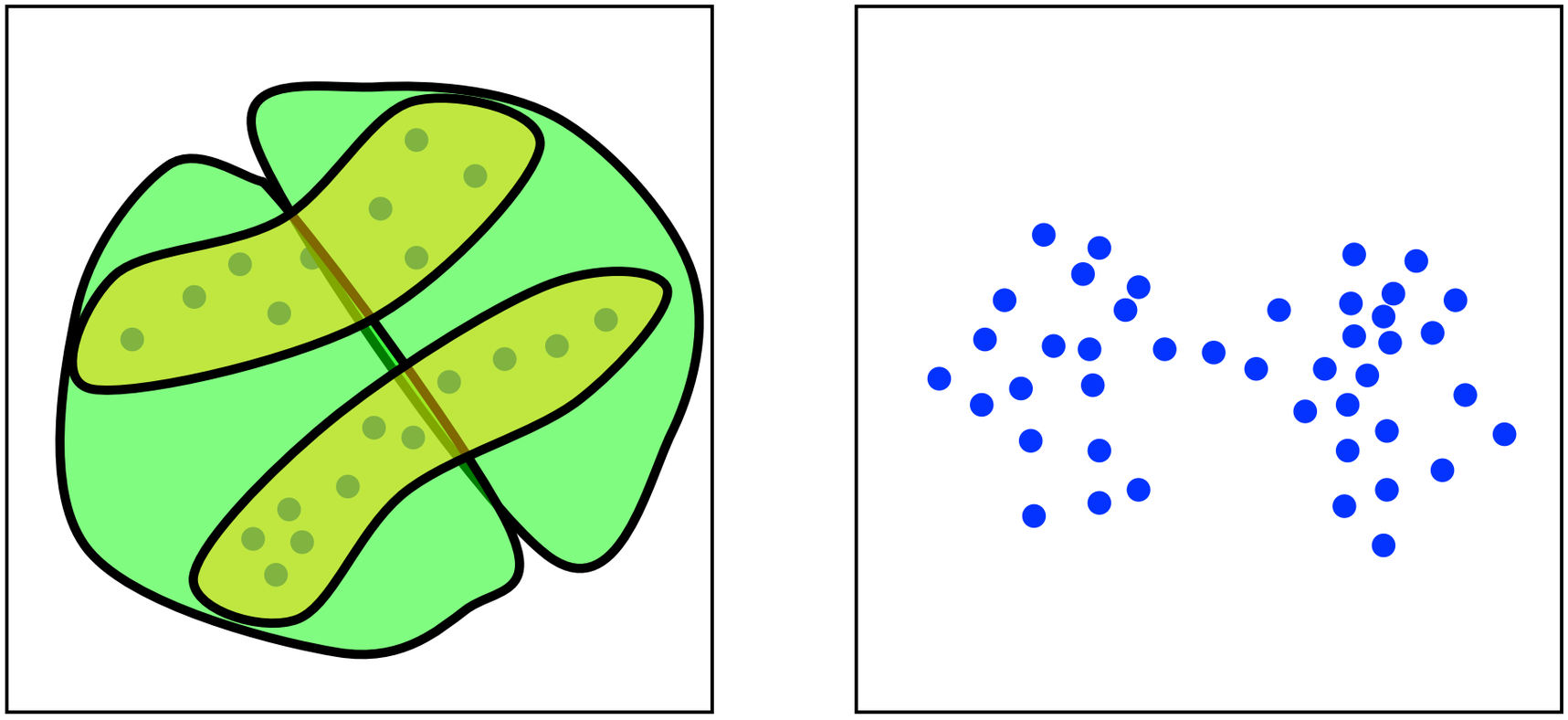
Linkage and cluster shape



complete linkage

single linkage

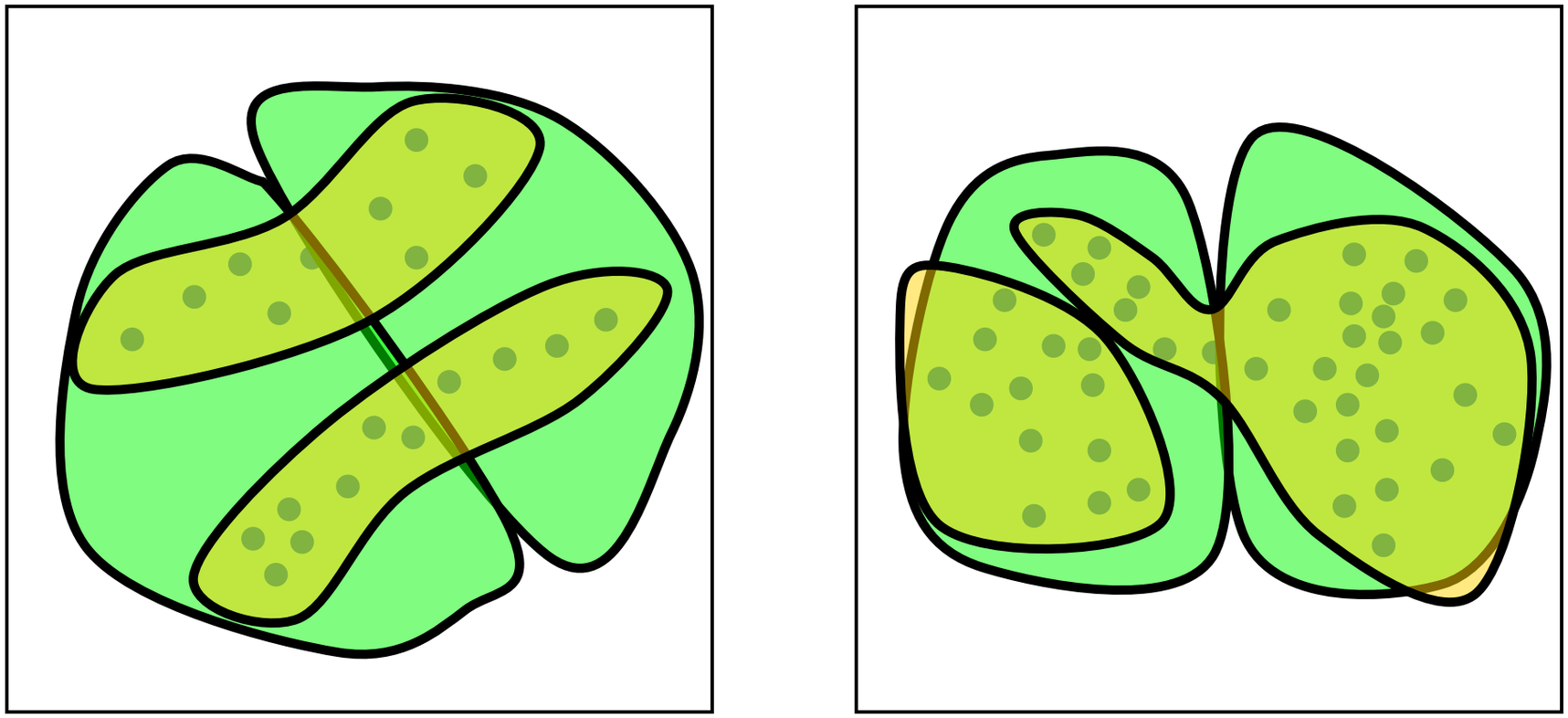
Linkage and cluster shape (2)



Complete linkage

Single linkage

Linkage and cluster shape (3)

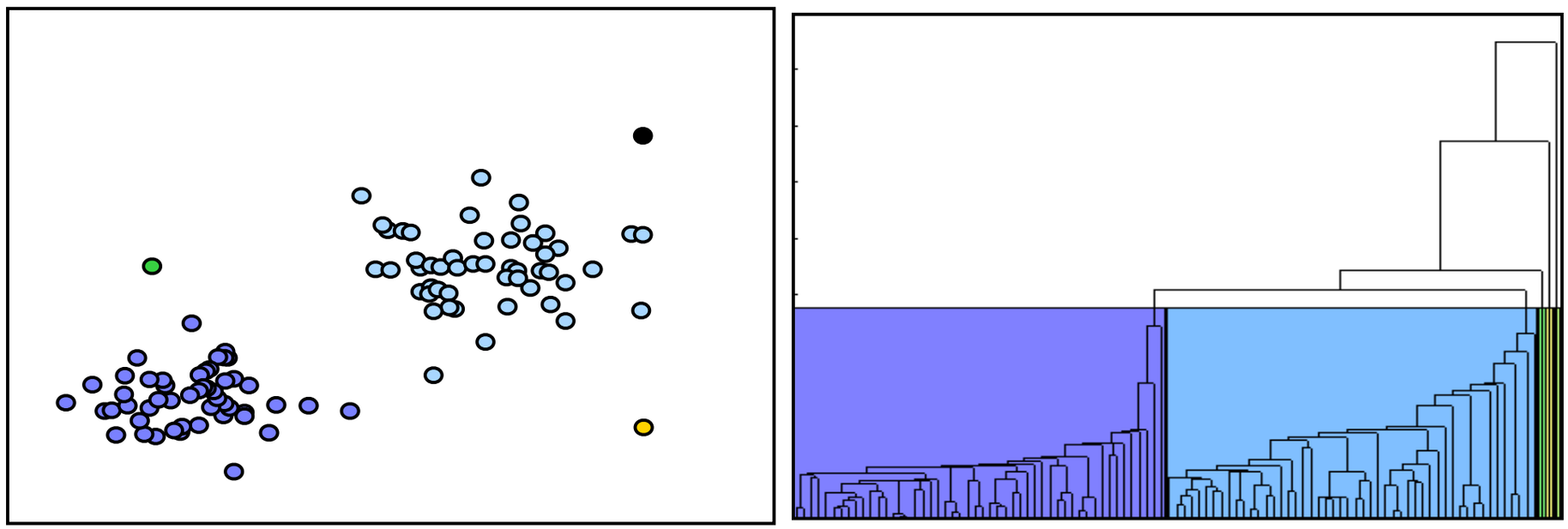


Complete linkage

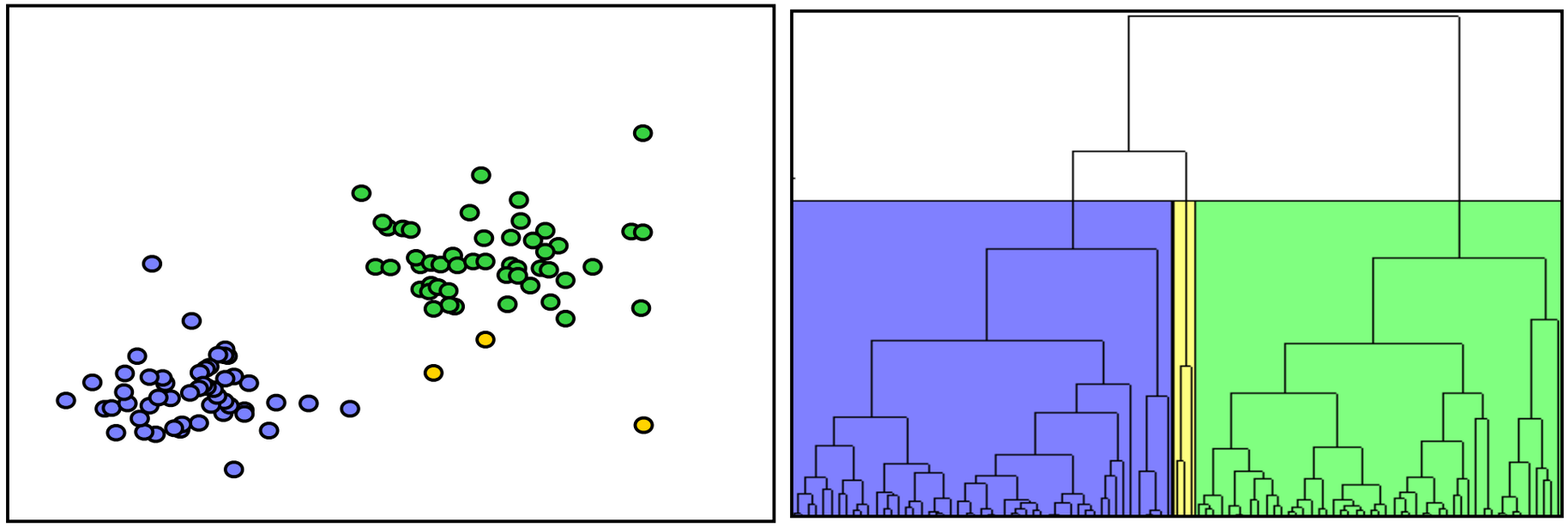
Single linkage

Linkage and outliers

Single linkage



Complete linkage

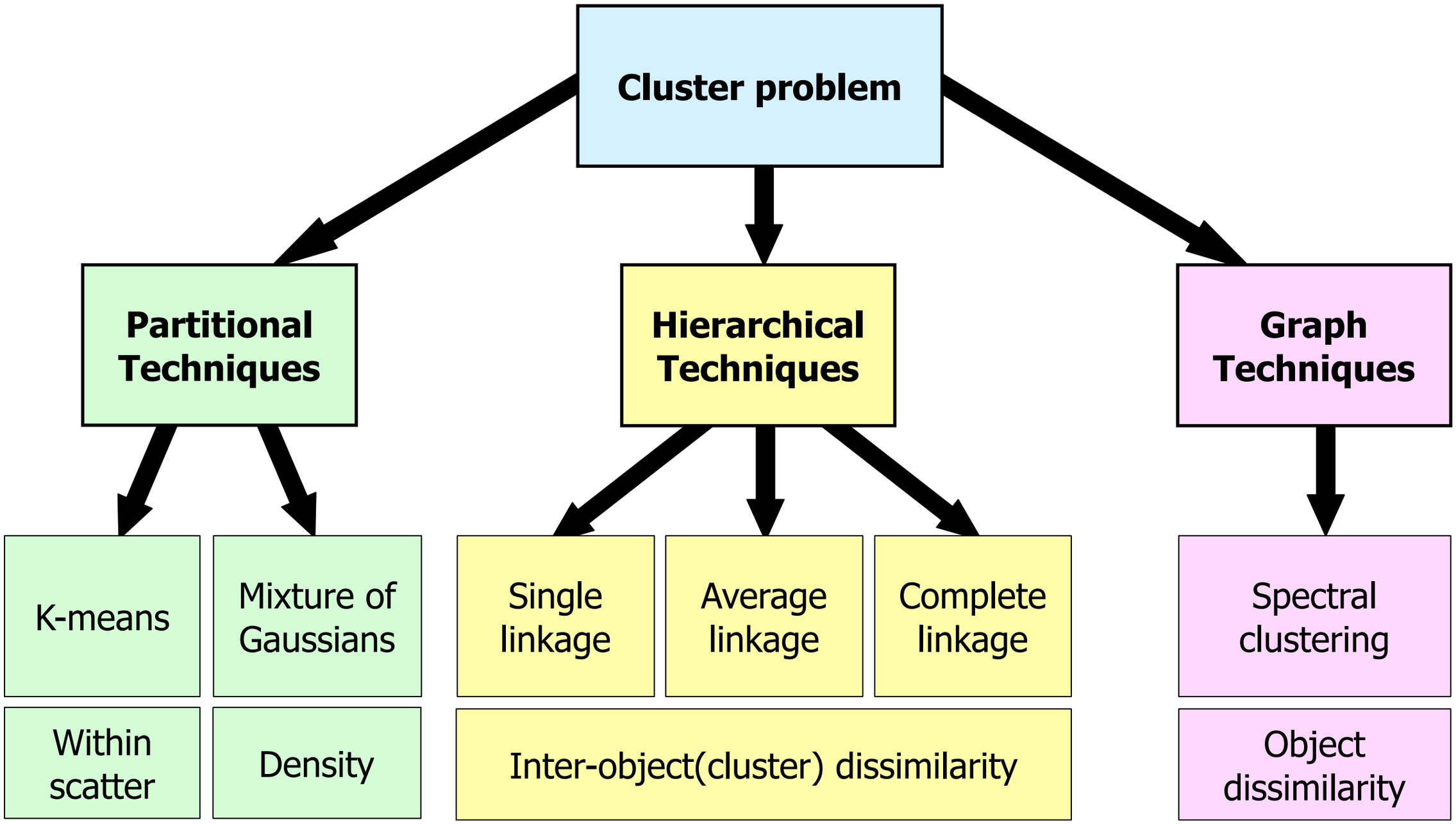


Hierarchical Clustering

13.2

- +
  - Dendrogram gives overview all possible clusterings
  - Linkage type allows to find clusters of varying shapes [convex and non-convex]
  - Different dissimilarity measures can be used
- –
  - Computationally intensive : $O(n^2)$ in complexity and memory
  - Clusterings limited to “hierarchical nestings”

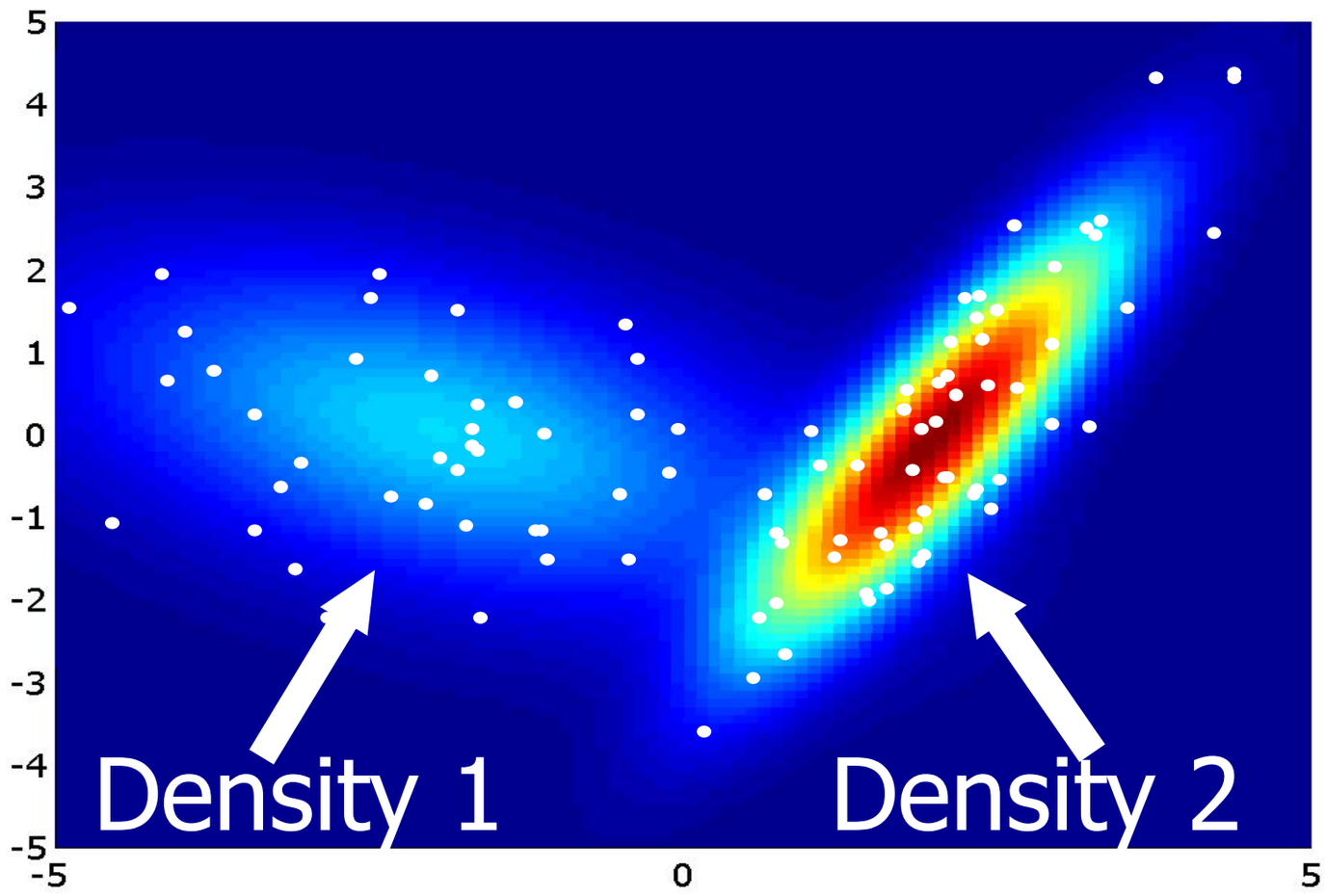
Clustering techniques



Probabilistic Mixture Model

14.2

- Each cluster is described by a probability density
- Total dataset is described by a **mixture** of densities
- Clustering = maximizing the mixture fit



TU Delft

Probabilistic Mixture Model

14.2

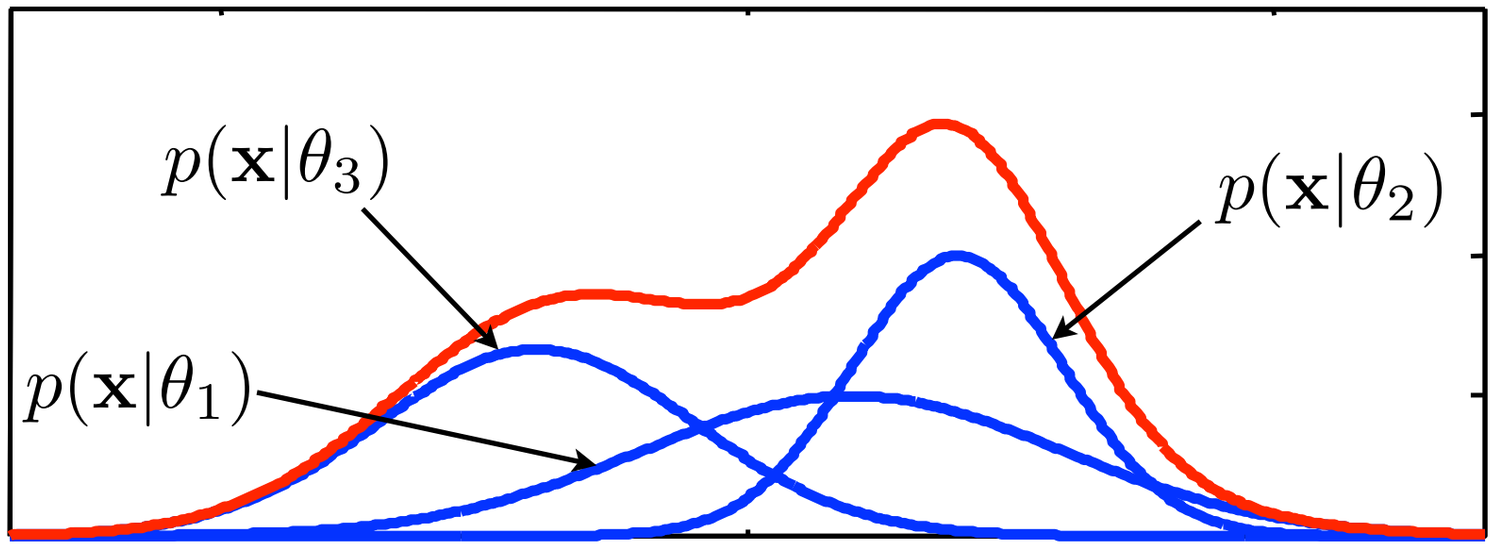
- Probabilistic mixture model : $p(\mathbf{x}|\Theta) = \sum_{j=1}^{m} u_j p(\mathbf{x}|\theta_j)$
- Mixing proportions : $u_j \geq 0, \quad \sum_{j=1}^{m} u_j = 1$
  - Probabilistic clustering allows for overlapping clusters
  - Model parameters are usually estimated by maximum likelihood approach using expectation-maximization [EM] algorithm

Mixture of Gaussians

- Choose Gaussian as component density

$$p(\mathbf{x}|\theta_j) = \frac{1}{\sqrt{2\pi^d \det(\Sigma_j)}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_j)^T \Sigma_j^{-1}(\mathbf{x}-\mu_j)\right)$$

- Describe the complete dataset as a mixture:



Maximum Likelihood Estimation

14.2

- Model and parameters : $p(\mathbf{x}|\Theta) = \sum_{j=1}^{m} u_j p(\mathbf{x}|\theta_j)$

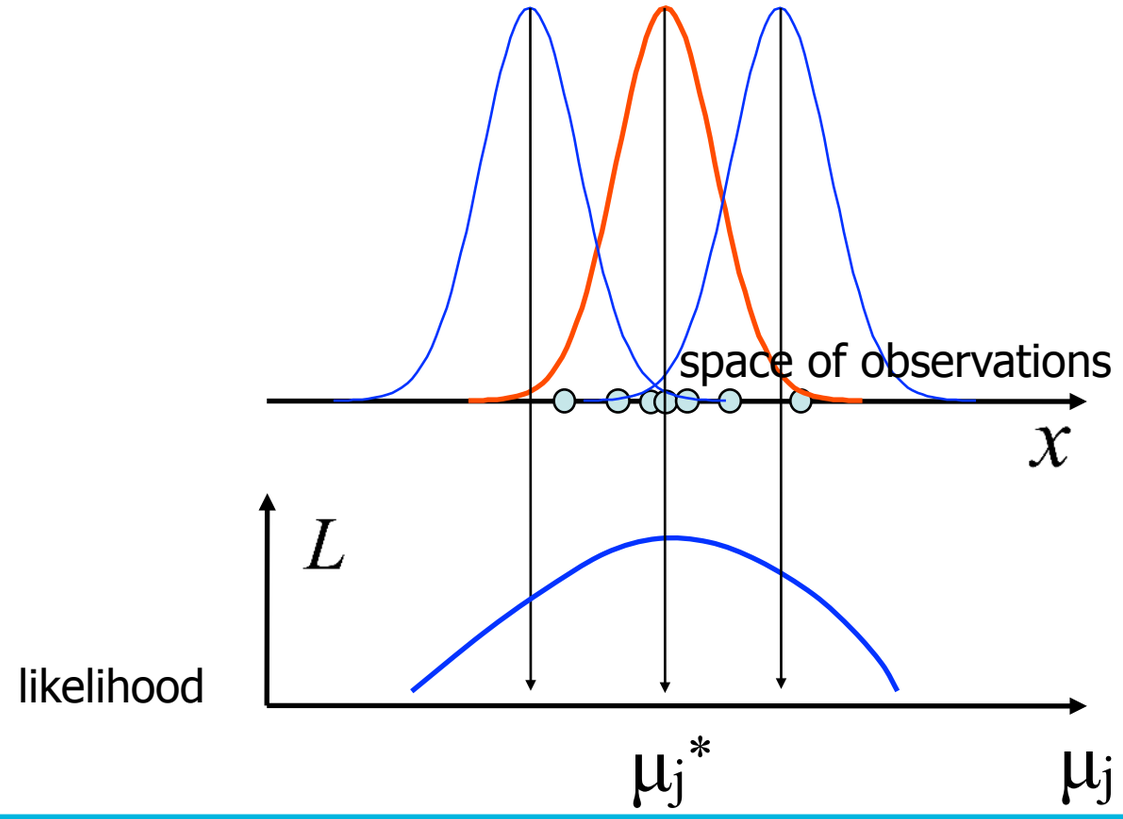
u_j, μ_j, Σ_j

- Likelihood : $L(\Theta|\mathbf{X}) = \prod_{i=1}^{N} p(\mathbf{x}_i|\Theta)$

  - Likelihood is a function of parameters, data samples remain fixed

- Introduce membership:

$$p(C_k|\mathbf{x};\Theta)$$



TU Delft

EM for Mixture Models

14.2

- EM = expectation maximization

- E-step computes cluster membership $p(C_k|\mathbf{x};\Theta)$ of each object based on current model
- M-step updates maximum likelihood estimates of parameters based on cluster membership

- Process is iterated…

EM for Mixture Models

14.2

- EM clustering
  - Assumes apriori known number of clusters: m
  - Need to define a cluster density (typically Gaussian)
  - Guarantees finding of local optimum only
  - May converge slowly
  - Is dependent on initialization
- but:
  - it can use prior knowledge on the cluster distribution
  - gives a general framework for any density mixture

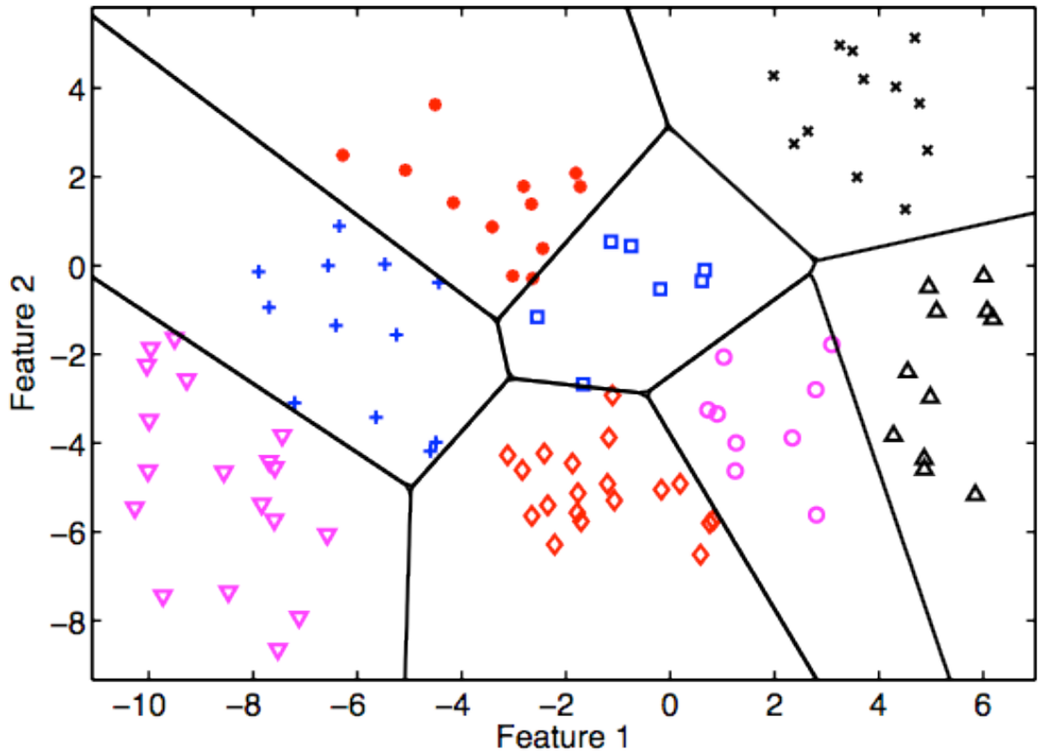
"Generalized" EM Clustering

14.2

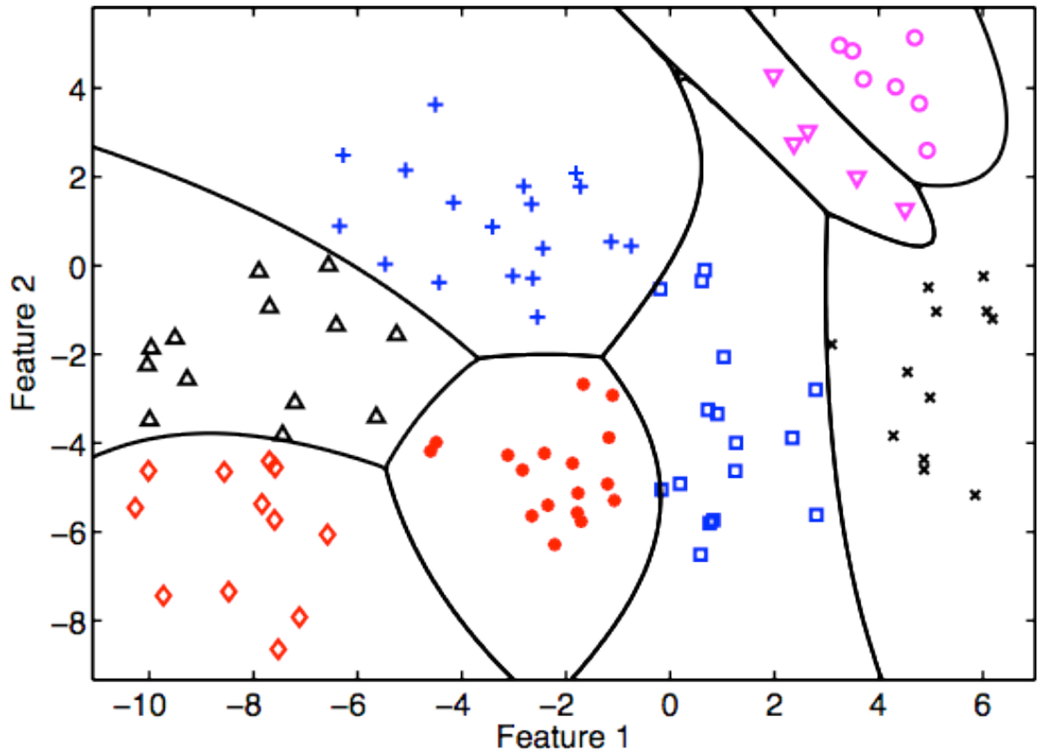
- Replace probability model by arbitrary classifier
- E-step : assign each observation **x** by classifier C to one of the classes
- M-step : use the labels to train new classifier C
- Stopping criterion : Labels do not change

"Generalized" EM Clustering

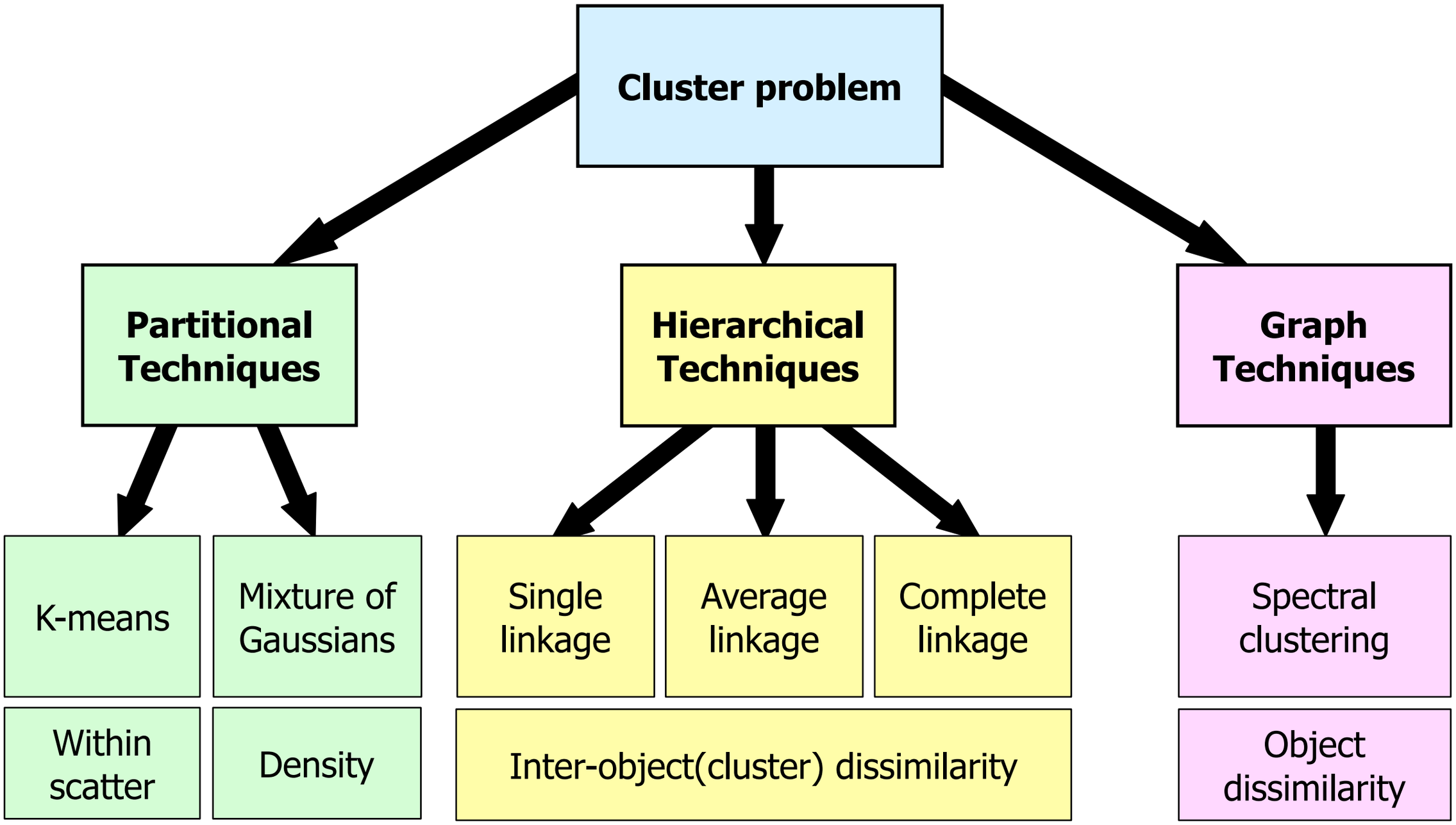
- `nmc` : assuming Gaussian densities with equal covariances



- `qdc` : assuming Gaussian densities with full covariance matrices



Clustering techniques



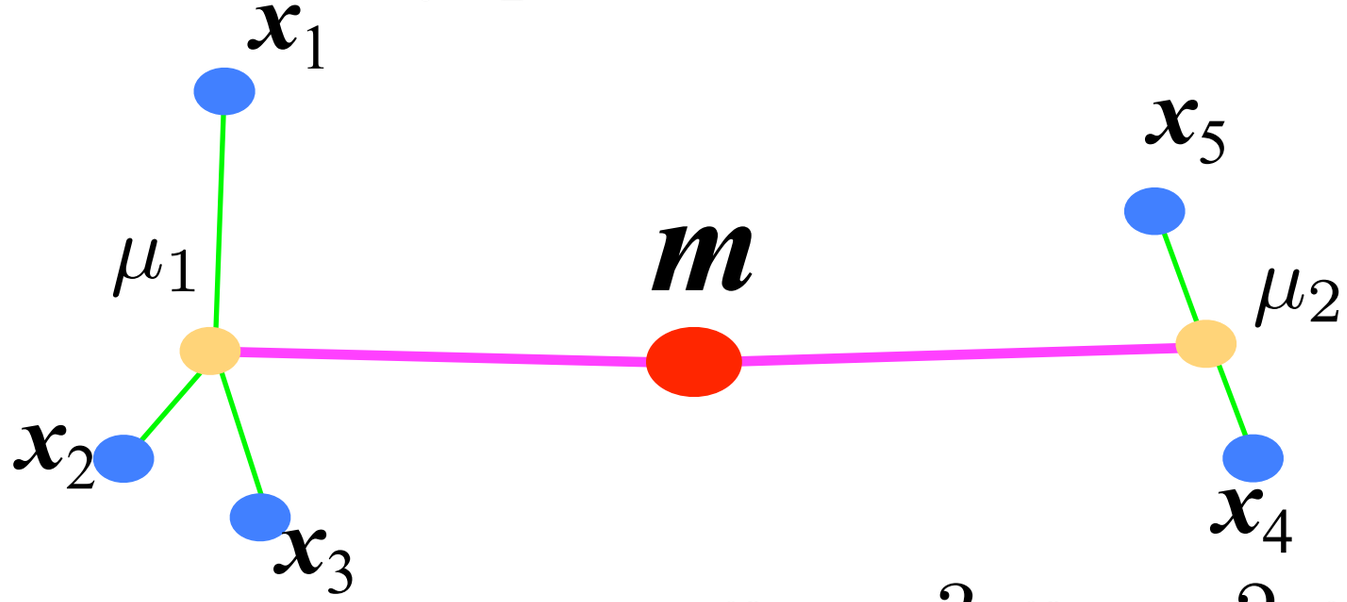
Sum-of-squares clustering

5.6.3

- Within and between scatter:

$$\mathbf{S}_w = \sum_{i=1}^{m} \frac{n_i}{n} \Sigma_i$$

$$\mathbf{S}_B = \sum_{i=1}^{m} \frac{n_i}{n} (\mu_i - \mathbf{m})(\mu_i - \mathbf{m})^T$$



$$n_1 = 3, n_2 = 2, n = 5, m = 2$$

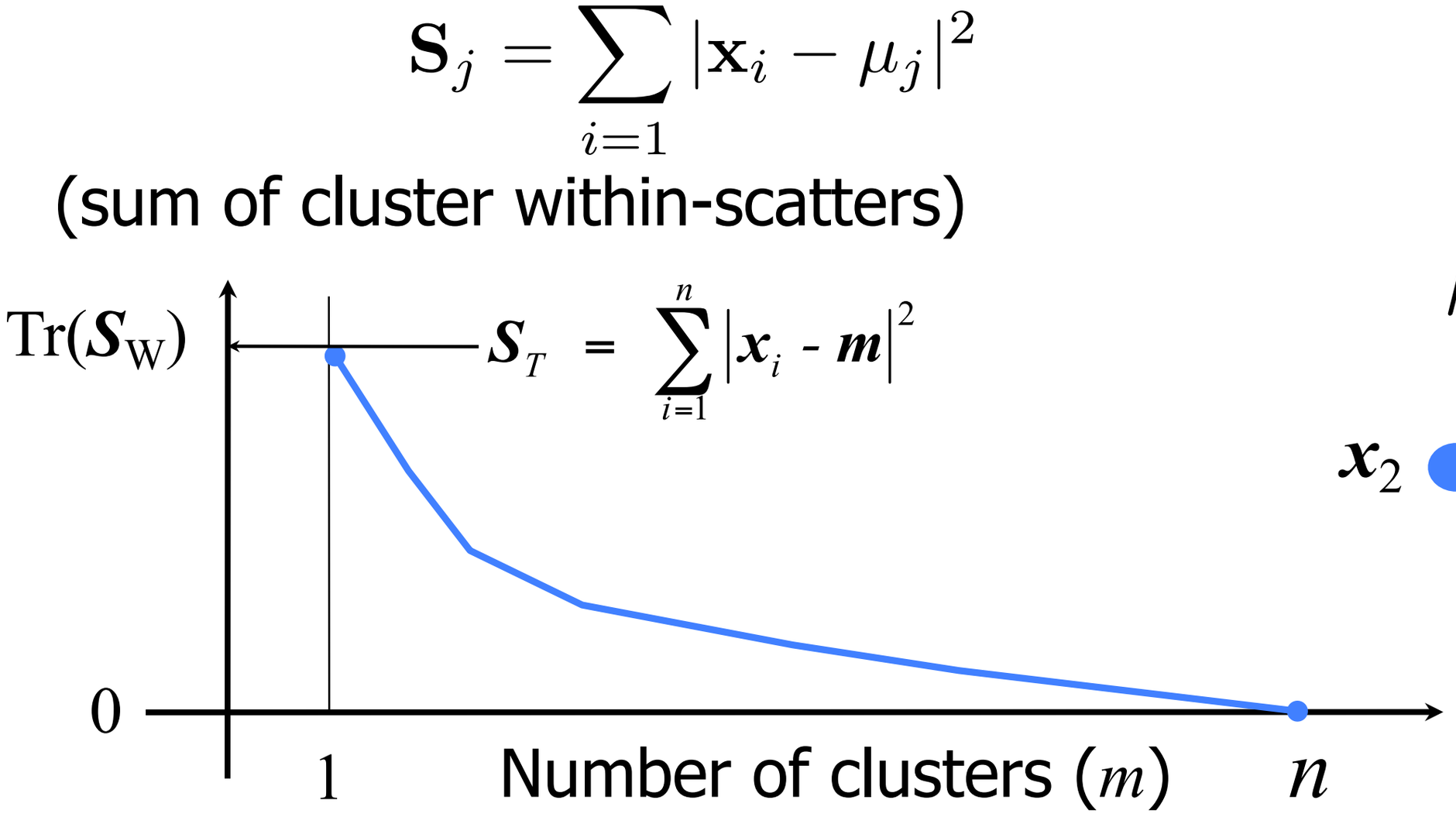
TU Delft

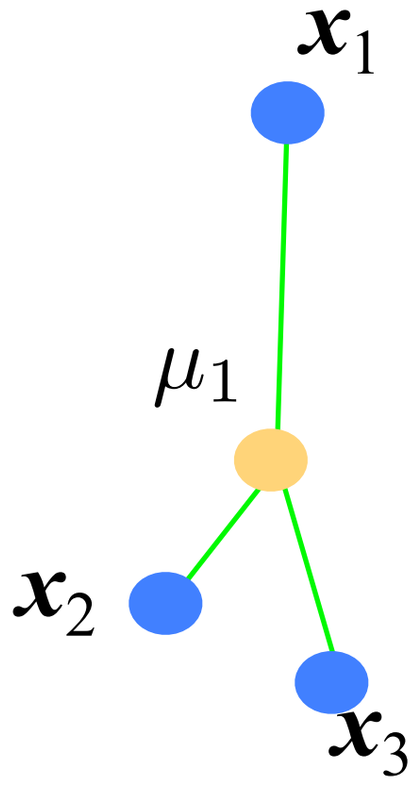
K-means

- Minimize: $\mathrm{Tr}\{S_w\} = \frac{1}{n}\sum_j \mathbf{S}_j$

$$\mathbf{S}_j = \sum_{i=1}^{n_j} |\mathbf{x}_i - \mu_j|^2$$

(sum of cluster within-scatters)





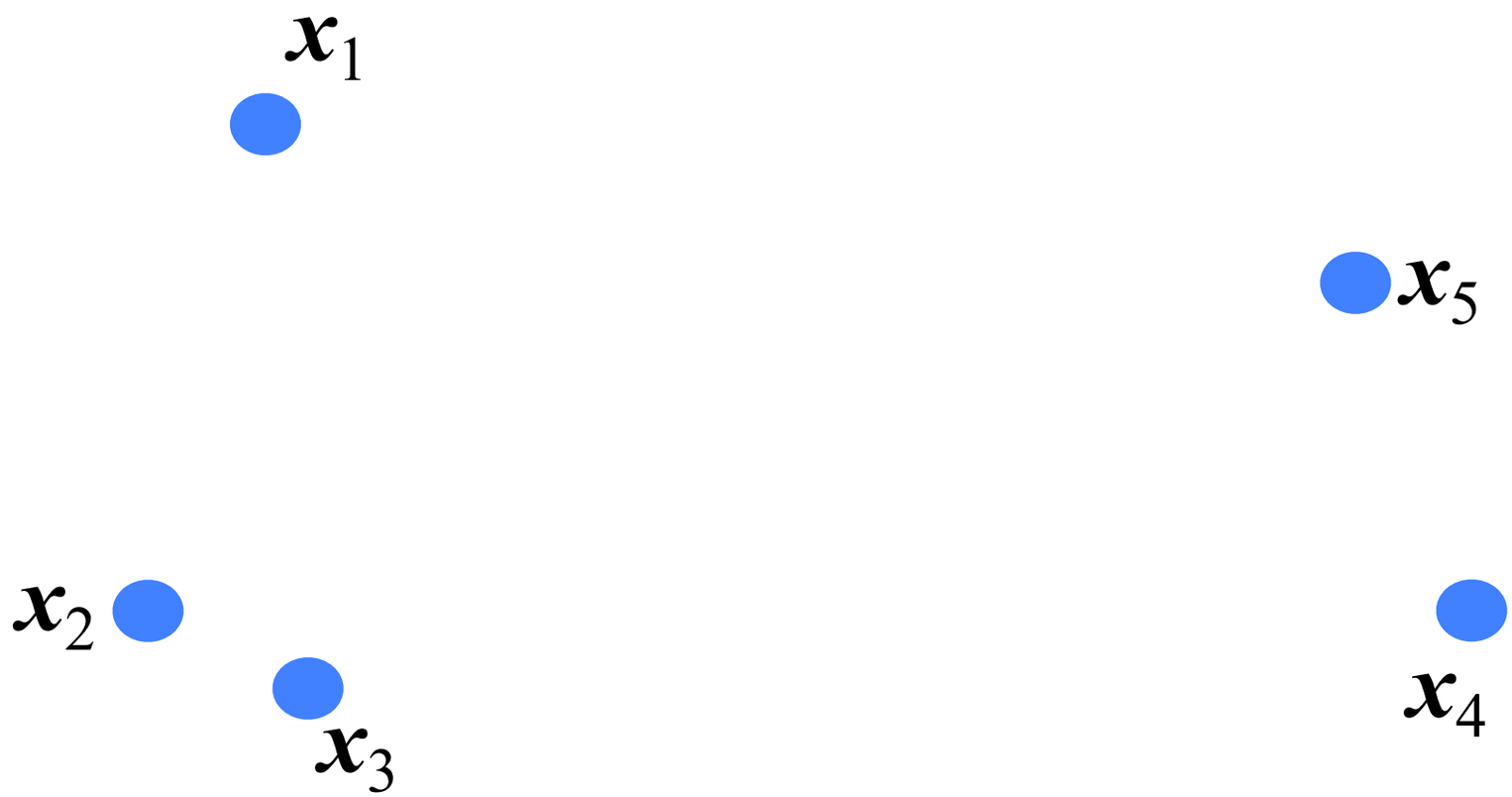
K-means

- Iterative procedure to search for $\min(\mathrm{Tr}(\boldsymbol{S}_\mathrm{W}))$:
  1. choose number of clusters (m)
  2. position prototypes $(m_j,\ j=1,\ldots,m)$ 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 (2)

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

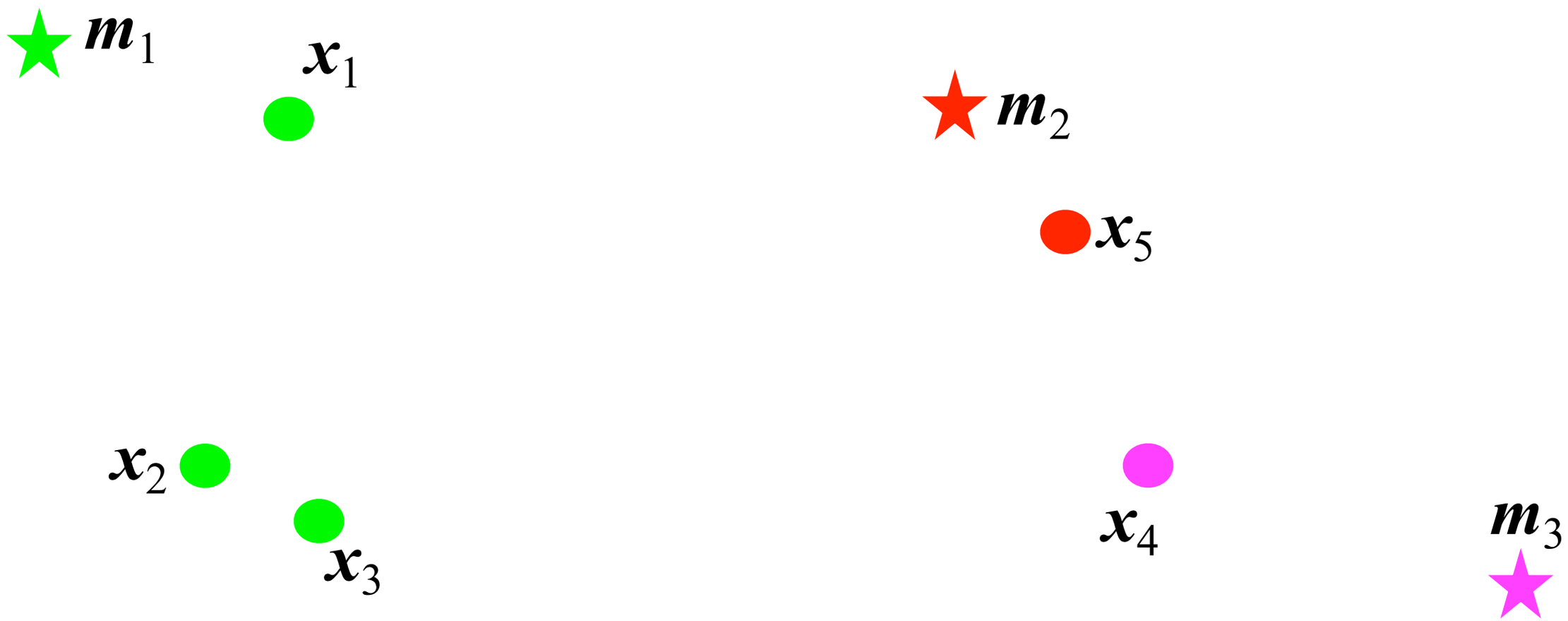


K-means (2)

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

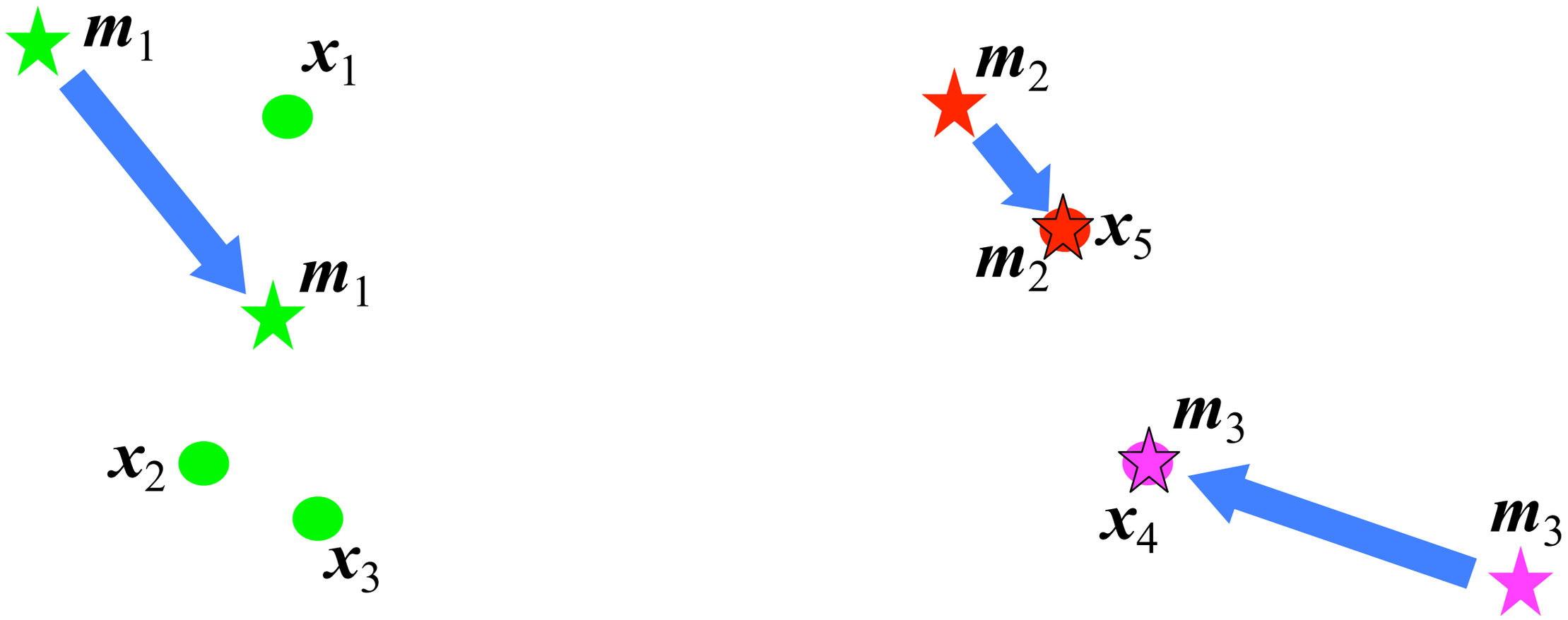
K-means (3)

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



K-means (4)

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

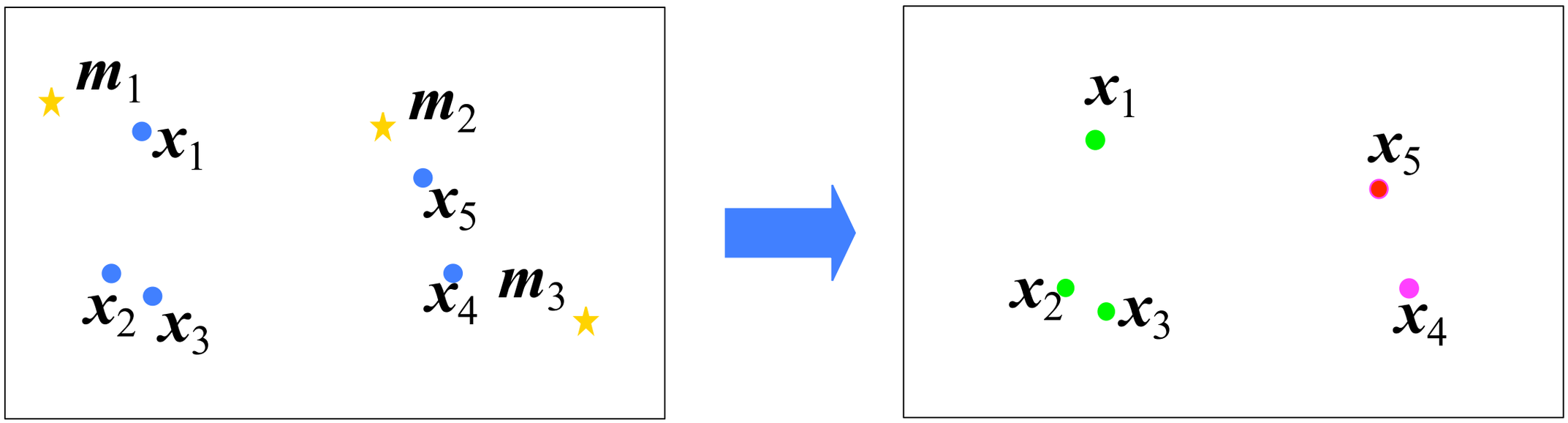


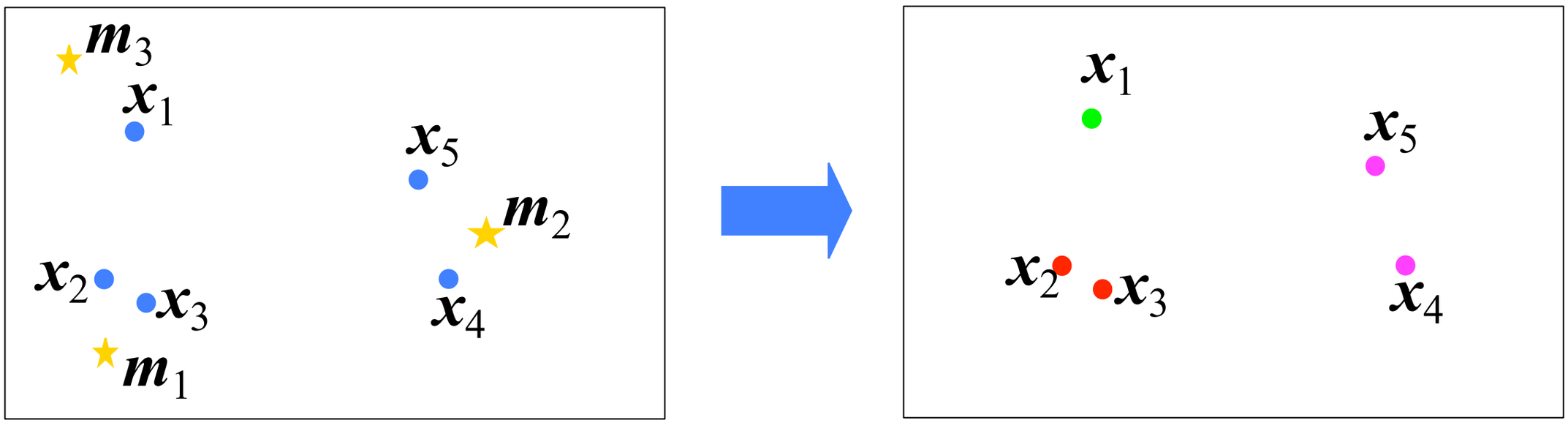
K-means (5)

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

K-means problems

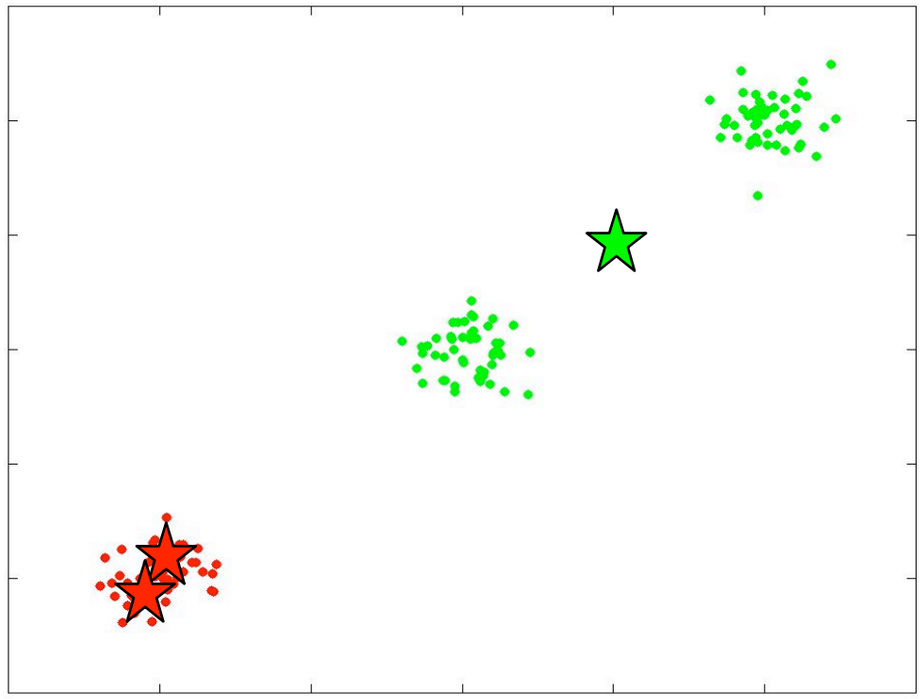
- Clustering depends on initialization





K-means problems (2)

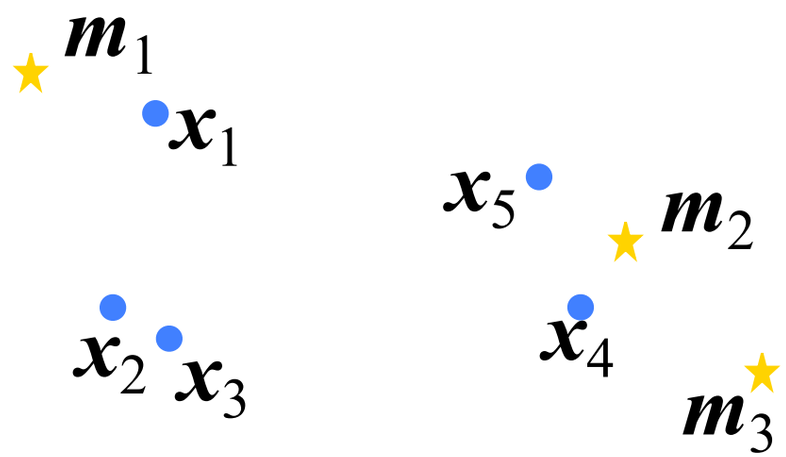
- Algorithm can get stuck in local minima



- 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 problems (3)

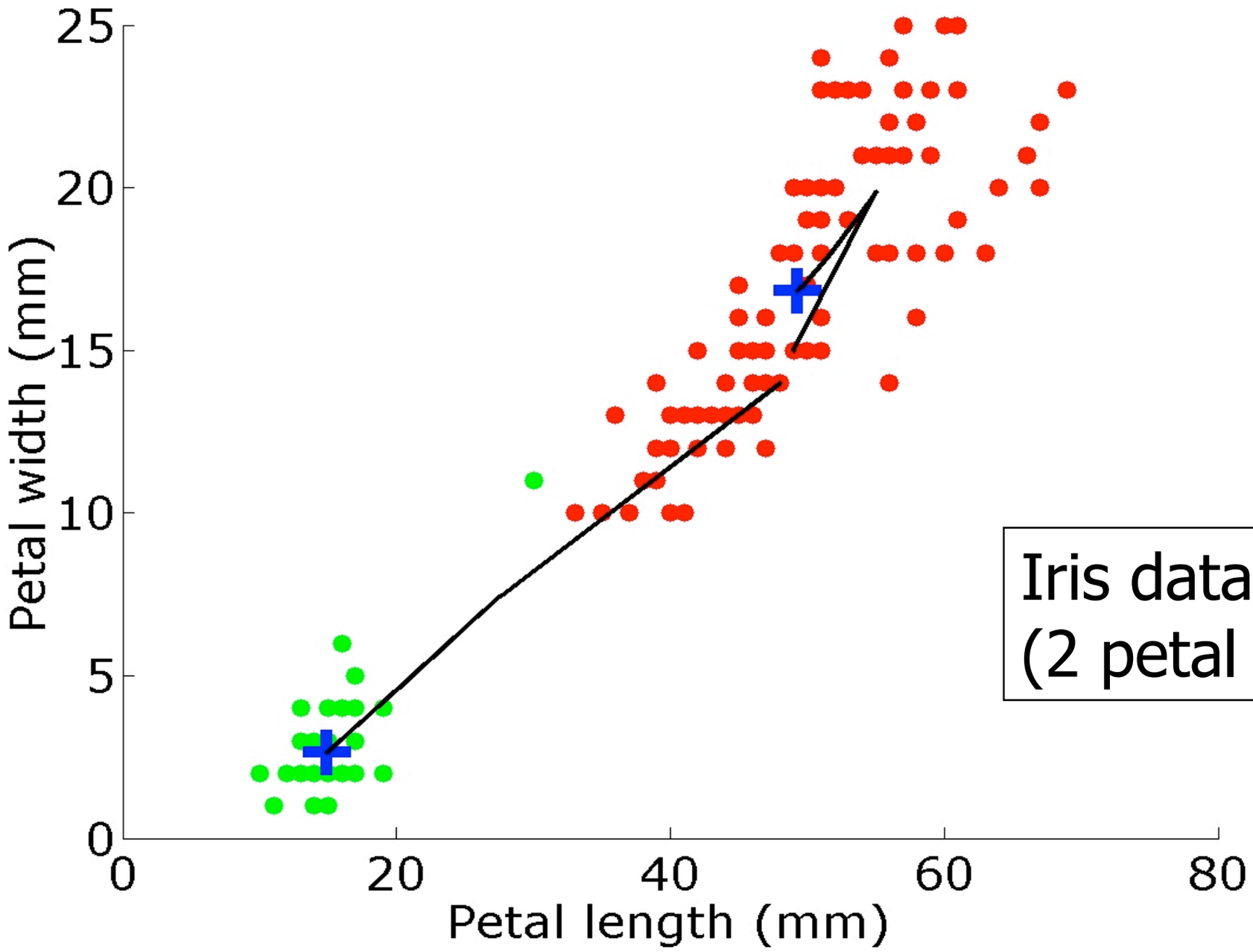
- Clusters can lose all samples



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

K-means example

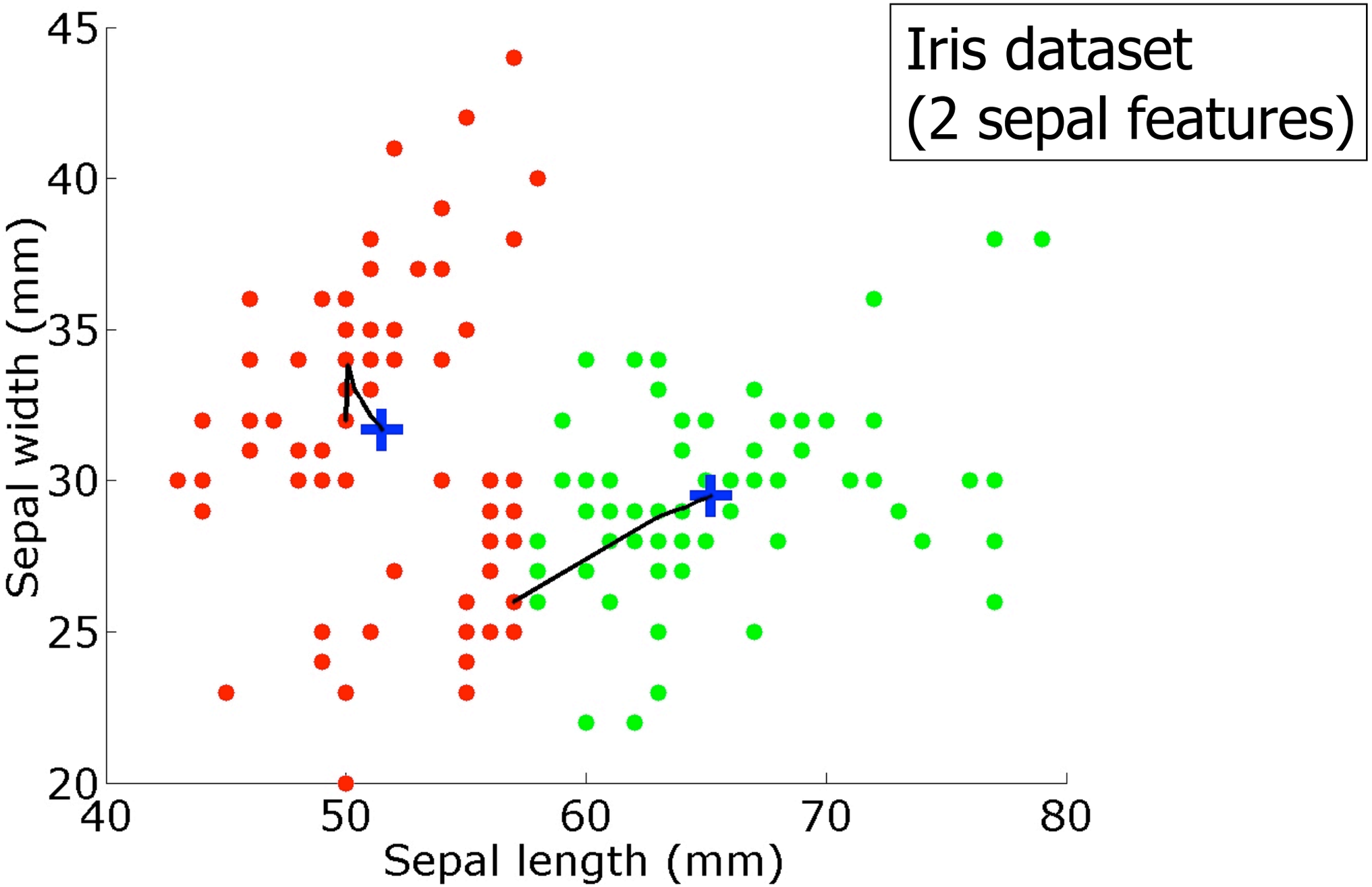
14.5



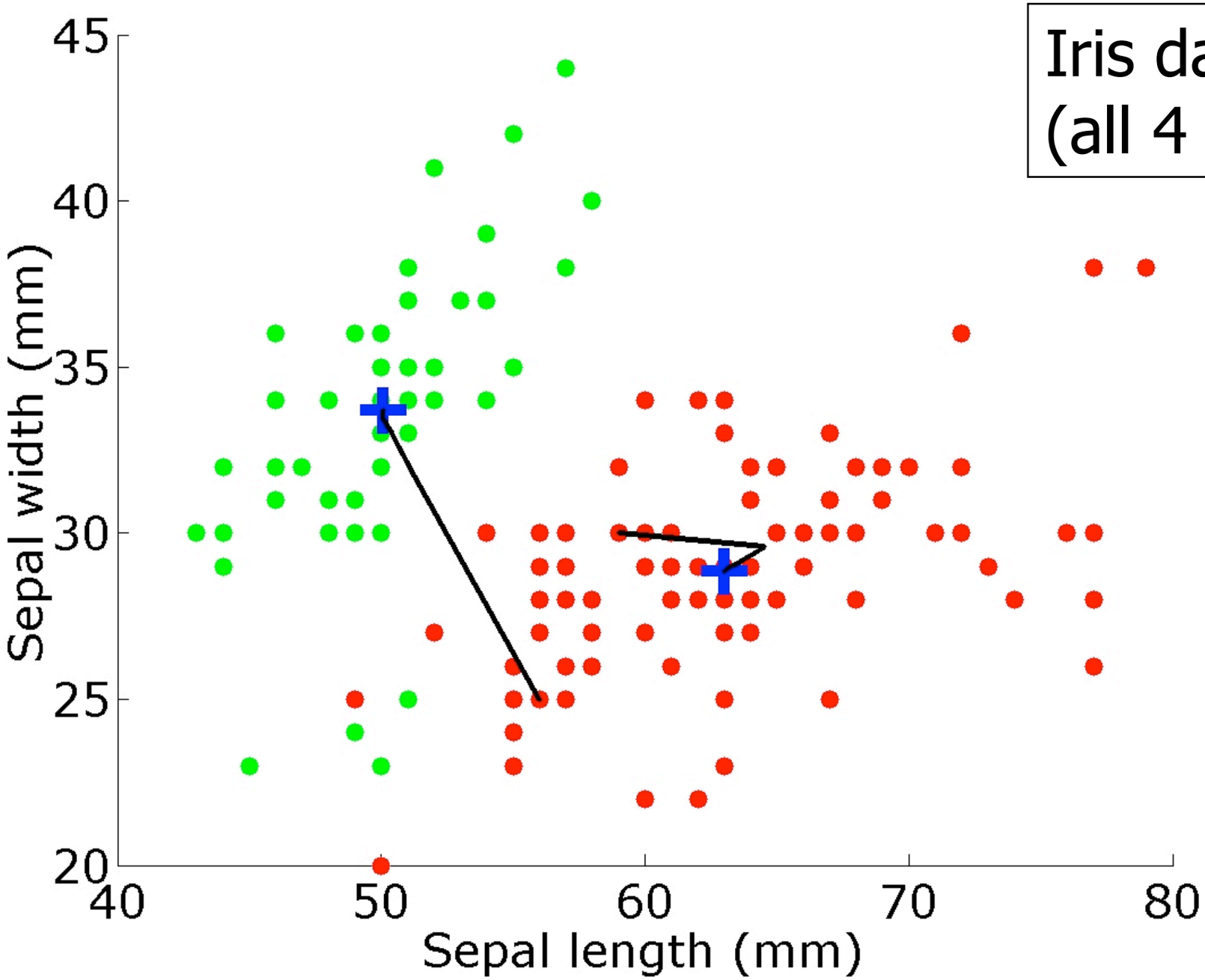
Iris dataset
(2 petal features)

TU Delft

K-means example (2)



K-means example (3)



K-means summary

- Disadvantages:
  - Finds only convex clusters ("round shapes")
  - Sensitive to initialization
  - Can get stuck in local minima

- Advantages:
  - Very simple
  - Fast

K-means & the EM algorithm

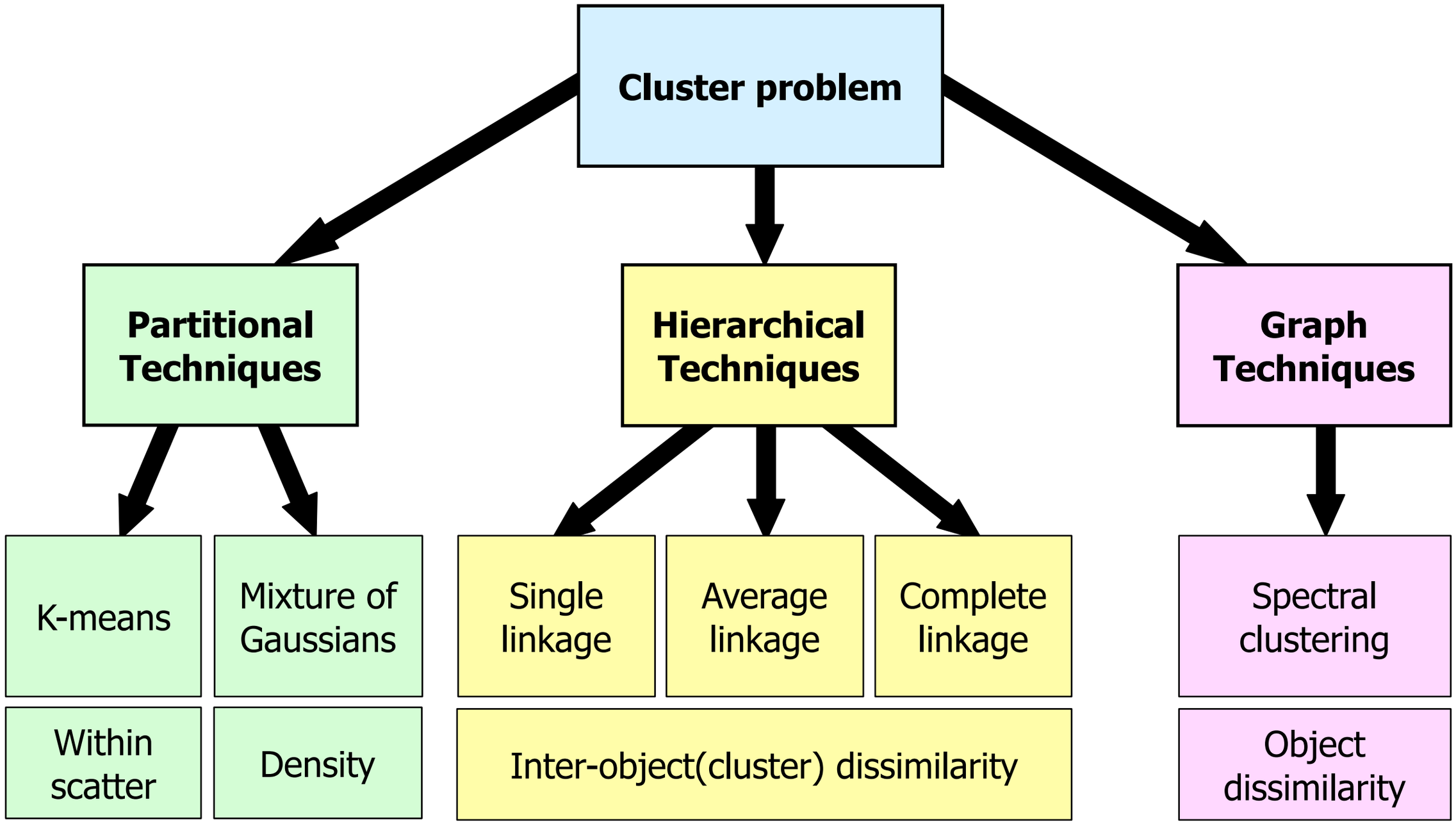
- If...
  - all clusters are spherical
  - the variance of each cluster is infinitely small

$$\Sigma = \begin{bmatrix} \varepsilon^2 & 0 & 0 \\ 0 & \varepsilon^2 & 0 \\ 0 & 0 & \varepsilon^2 \end{bmatrix}, \quad \varepsilon \to 0$$

  then the EM algorithm simplifies to the K-means algorithm (samples are always assigned to the closest cluster!)

The difference: k-means uses crisp labels, EM uses soft labels...

Clustering techniques



Conclusion

- We can classify when we don't have (training) labels: clustering
- Definition of cluster is vague, several methods have been devised:
  - Hierarchical clustering
  - Mixture of Gaussians
  - k-means clustering
  - (Spectral clustering)
  - …
- How to know what to use? Which one is the best?