Unsupervised learning: clustering

5ARE0: DATA ANALYSIS & LEARNING METHODS (2025 – 2026)

Uzay Kaymak, Jheronimus Academy of Data Science, u.kaymak@tue.nl

Master: Artificial Intelligence & Engineering Systems

Includes slides courtesy of W. Kouw

Recap

- Feature selection
  - Filter-based methods
  - Wrapper methods
  - Embedded methods
- Feature extraction
  - Create augmented/derived variables (new features)
  - Linear change of features (PCA/ICA)
  - Nonlinear embedding methods have become popular in recent years

Outline

- **What is clustering**
- **When to cluster?**
- **Crisp clustering**
  - Hierarchical clustering
  - K-means clustering
- **Fuzzy clustering**
  - Fuzzy c-means clustering
  - Gustafson-Kessel clustering
  - Possibilistic c-means clustering

Clustering

Clustering refers to a family of techniques that groups data points based on similarity

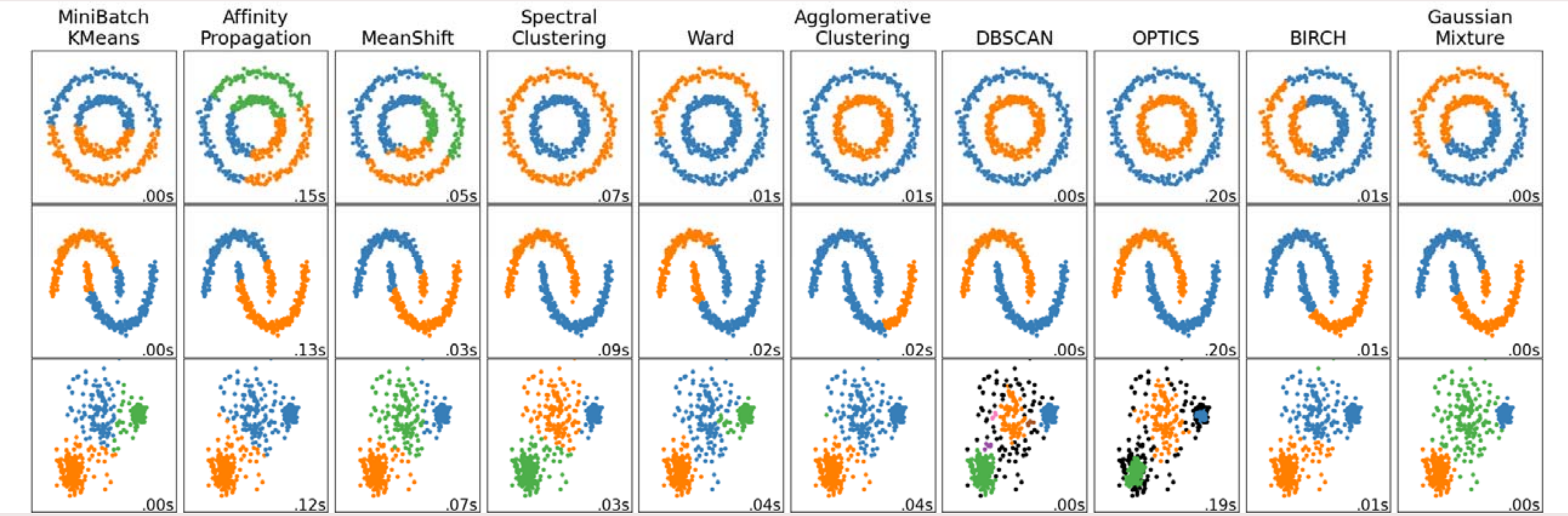
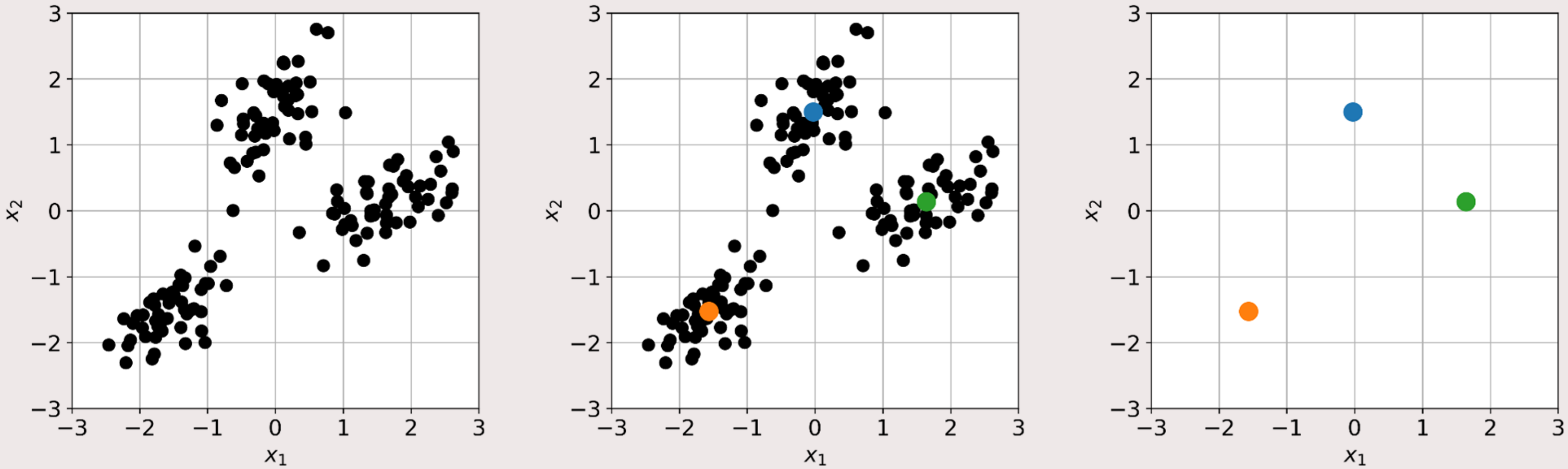


Figure taken from *Scikit-learn: Comparing clustering algorithms* (link).

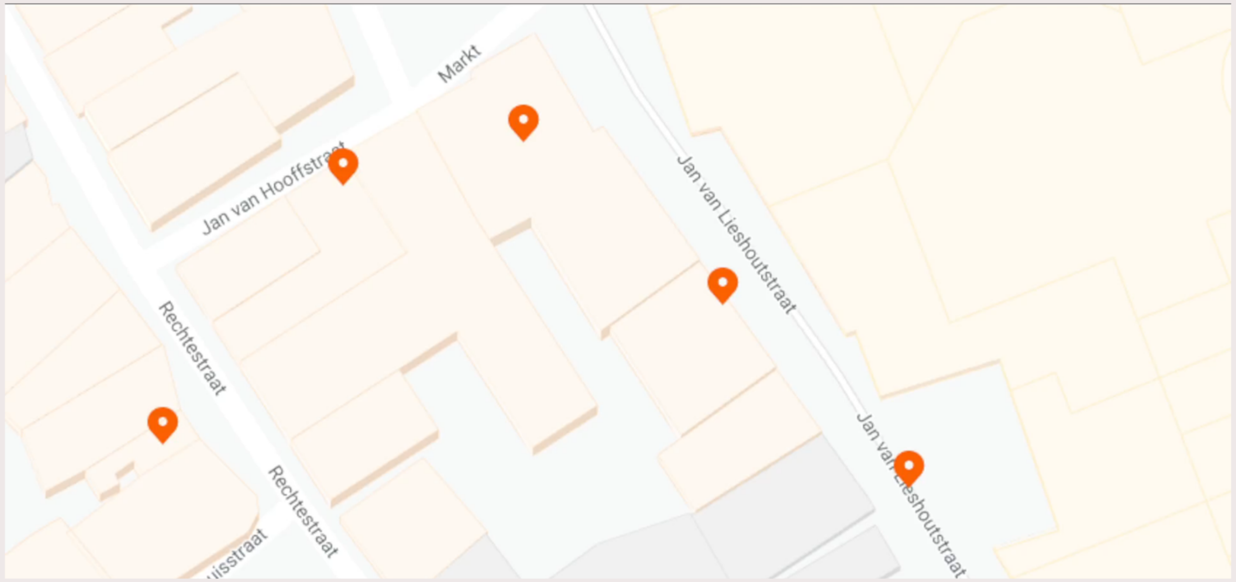
When to cluster?

Data compression: Groups of points can be replaced by representative samples



When to cluster?

Visualization: High-level overviews can be constructed by merging points in clusters



7

When to cluster?

Subgraph detection: communities in networks can be found by clustering nodes.

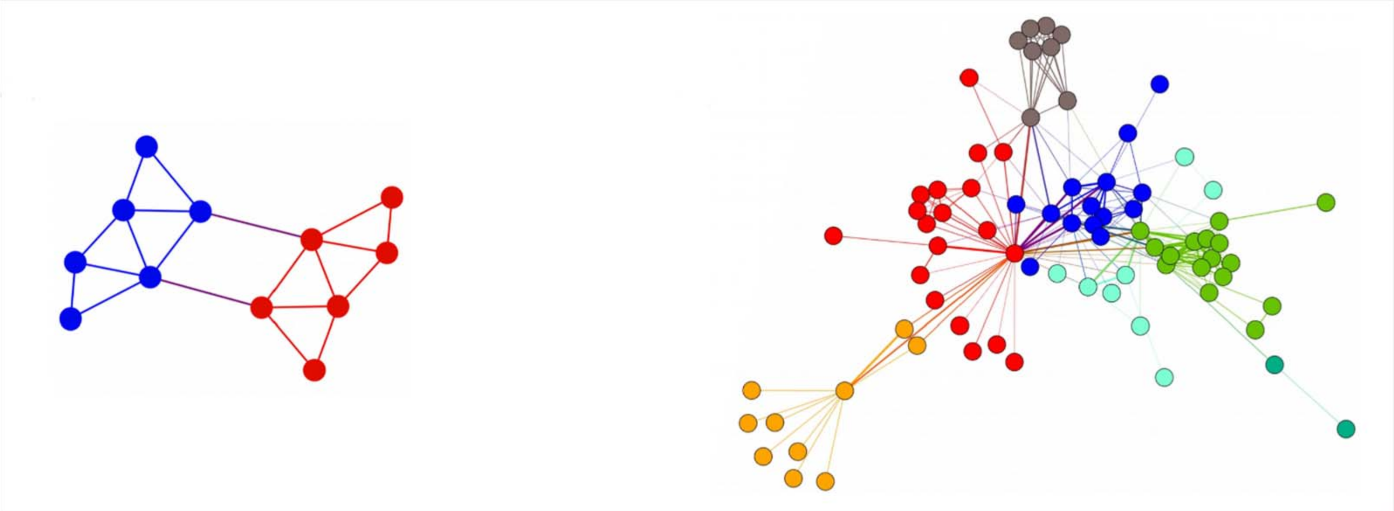
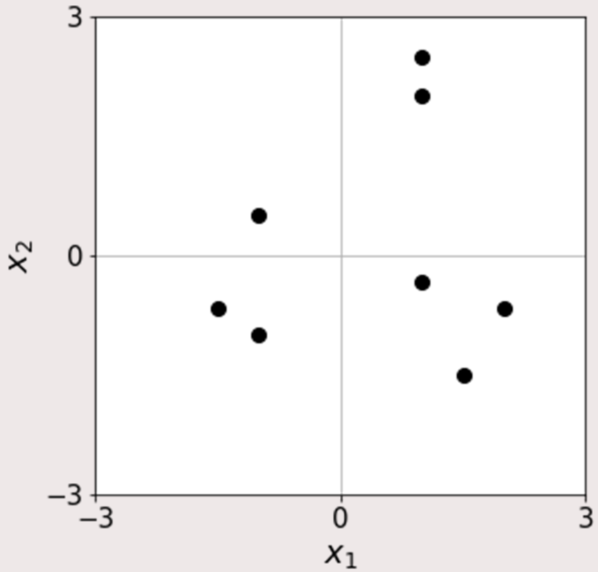


Figure taken from *Negre, Ushijima-Mwesigwa, Mniszewski (2020), Detecting multiple communities using quantum annealing on the D-Wave system.*

Clustering

Q: How would you cluster the following data set?



- **How many clusters to use?**
- **How big should every cluster be?**
- **What similarity metric?**
- **Select or create representative samples?**

9

Hierarchical clustering

“Hierarchical clustering” algorithms generate a *hierarchy* of clustering of points

- The hierarchy is typically encoded in a tree graph.
- *Agglomerative clustering* groups points from the bottom up.
- *Divisive clustering* groups points from the top down.
- Similarity is typically expressed in terms of distances between points.

Q: What depth will the hierarchy have?

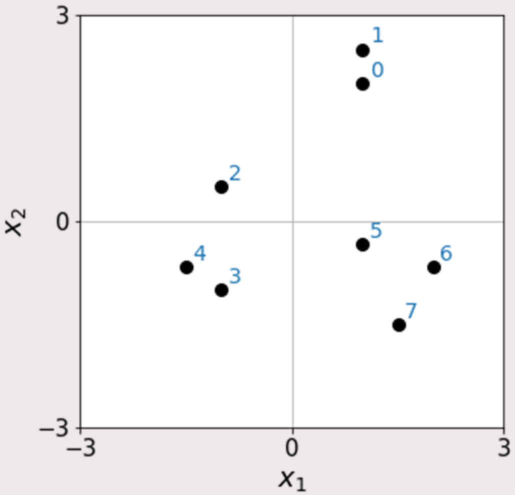
At the top is **1** cluster (all samples) and at the bottom are $\boldsymbol{N}$ clusters (1 for each sample).

10

Agglomerative clustering: single-linkage

Single-linkage is a similarity metric based on the distance between the closest two points.

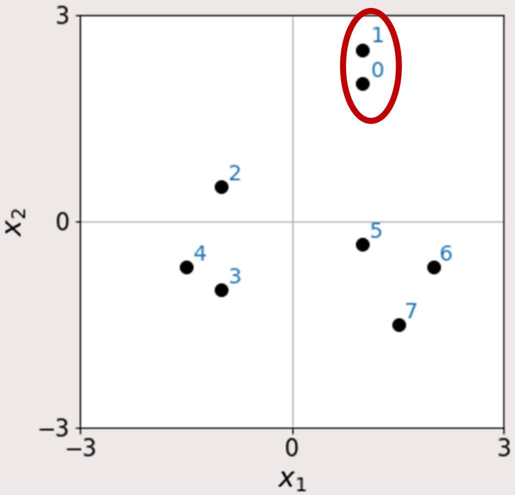
At L(0), each sample is a cluster and cluster similarities are pure pointwise distances:



	0	1	2	3	4	5	6	7
0	0	0.5	2.5	3.6	3.7	2.3	2.8	3.5
1	0.5	0	2.8	4	4	2.8	3.3	4
2	2.5	2.8	0	1.5	1.3	2.2	3.2	3.2
3	3.6	4	1.5	0	0.6	2.1	3	2.5
4	3.7	4	1.3	0.6	0	2.5	3.5	3.1
5	2.3	2.8	2.2	2.1	2.5	0	1.1	1.3
6	2.8	3.3	3.2	3	3.5	1.1	0	0.97
7	3.5	4	3.2	2.5	3.1	1.3	0.97	0

Agglomerative clustering: single-linkage

At L(1), we first find the closest pair of clusters based on our similarity matrix.

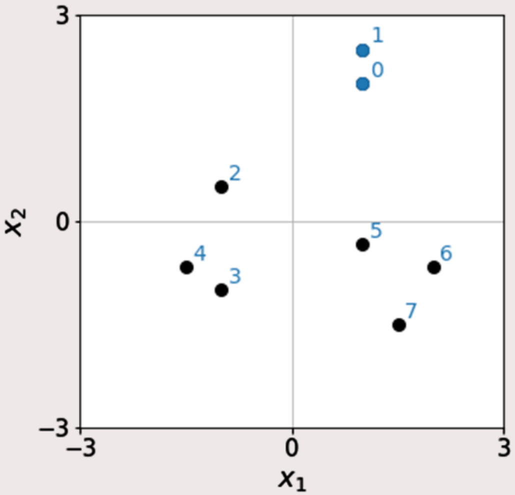


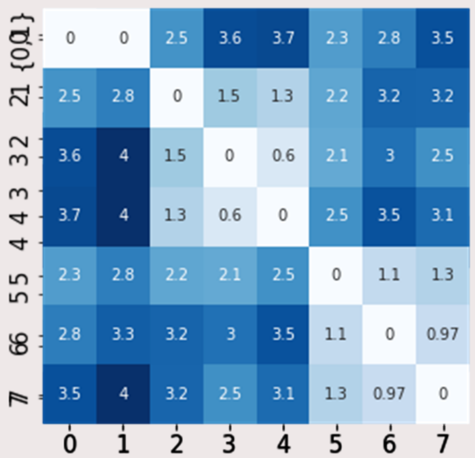
	0	1	2	3	4	5	6	7
0	0	0.5	2.5	3.6	3.7	2.3	2.8	3.5
1	0.5	0	2.8	4	4	2.8	3.3	4
2	2.5	2.8	0	1.5	1.3	2.2	3.2	3.2
3	3.6	4	1.5	0	0.6	2.1	3	2.5
4	3.7	4	1.3	0.6	0	2.5	3.5	3.1
5	2.3	2.8	2.2	2.1	2.5	0	1.1	1.3
6	2.8	3.3	3.2	3	3.5	1.1	0	0.97
7	3.5	4	3.2	2.5	3.1	1.3	0.97	0

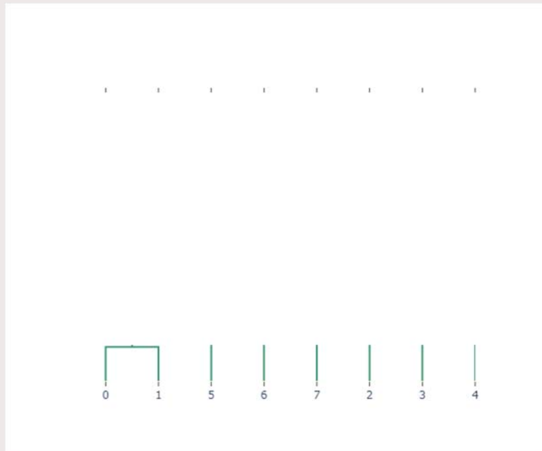
Agglomerative clustering: single-linkage

At L(1), we first find the closest pair of clusters based on our similarity matrix.

Then, we merge them based on minimal distances; $d\big(k,(i,j)\big) = \min\{d(k,i), d(k,j)\}$.



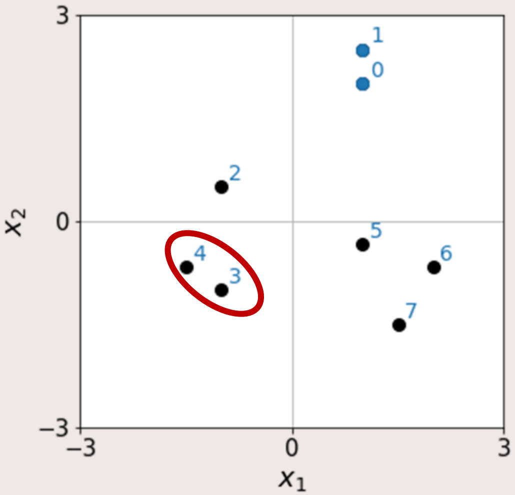


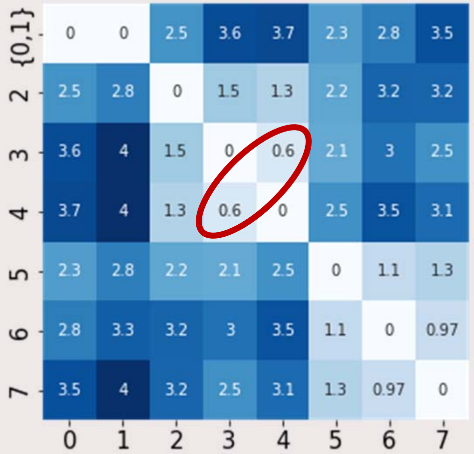


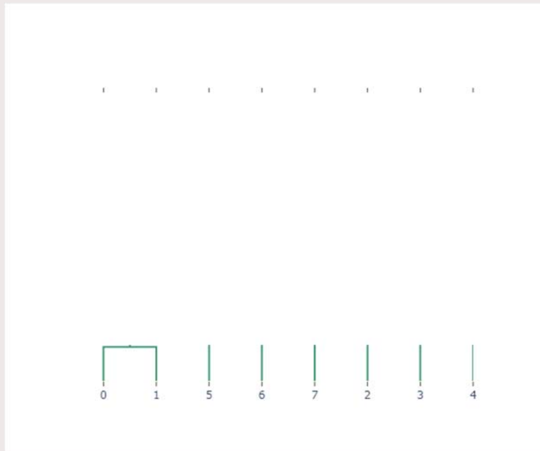
13

Agglomerative clustering: single-linkage

At L(2), we find the closest pair of clusters based on our updated similarity matrix.





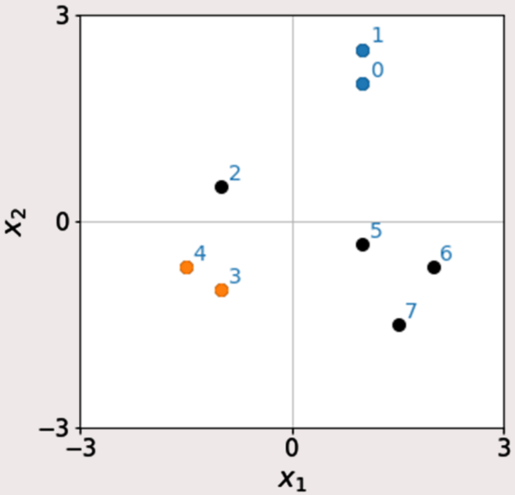


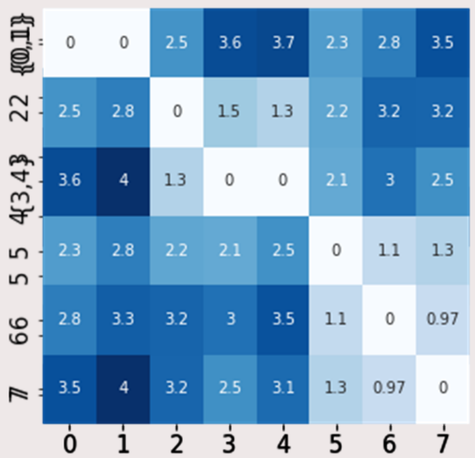
14

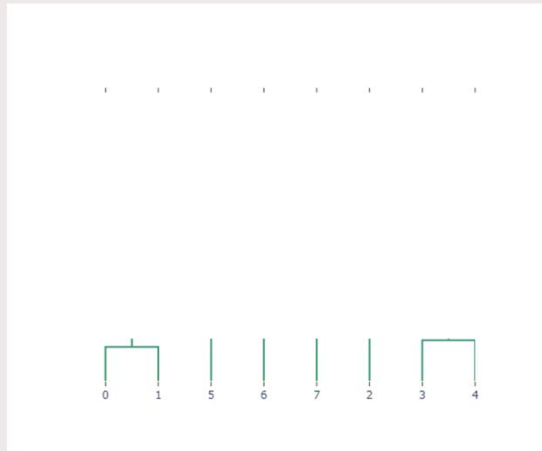
Agglomerative clustering: single-linkage

At L(2), we find the closest pair of clusters based on our updated similarity matrix.

Again, we merge the closest clusters into one, using the minima of distances.



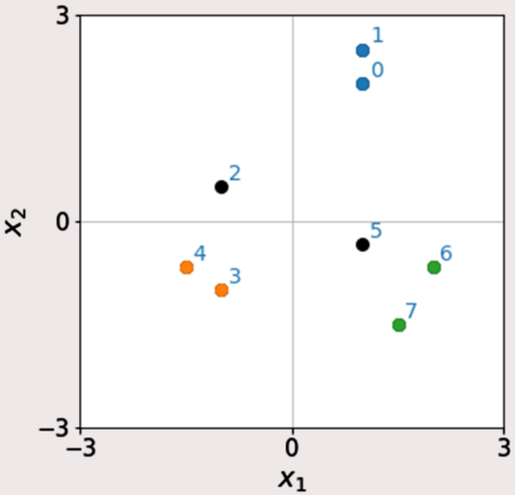


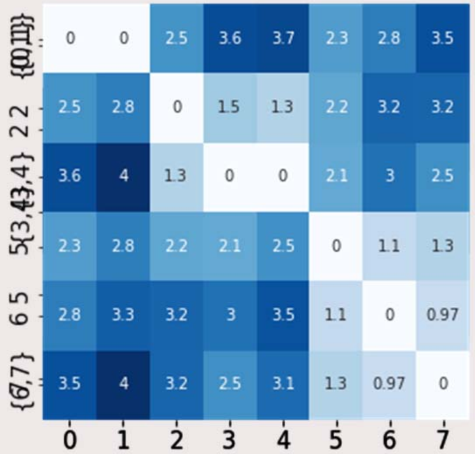


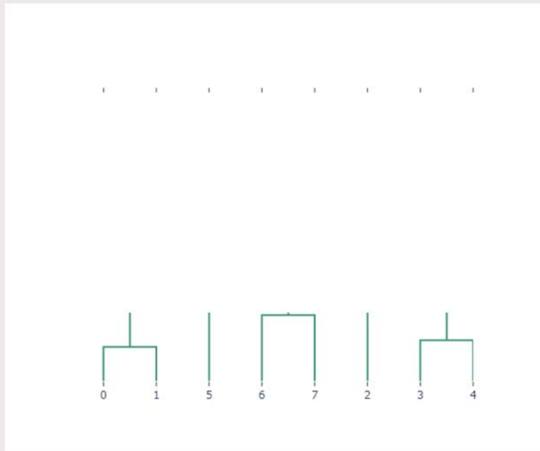
15

Agglomerative clustering: single-linkage

We repeat the same procedure to get L(3).



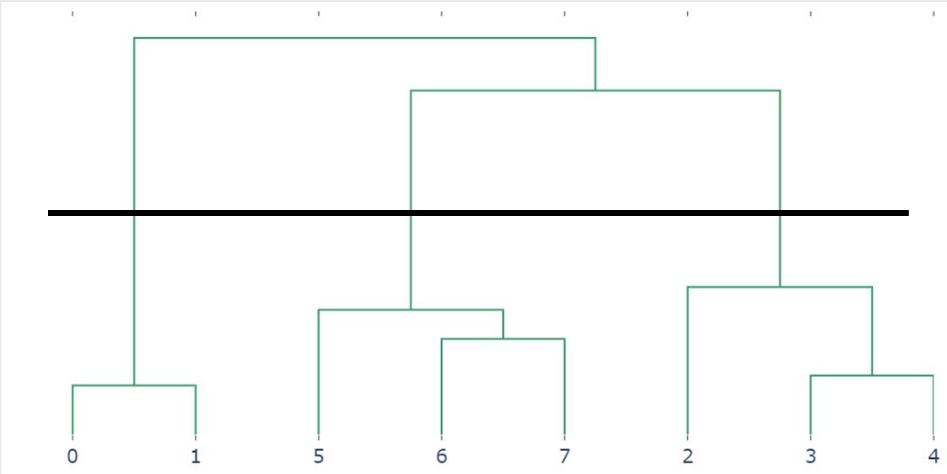


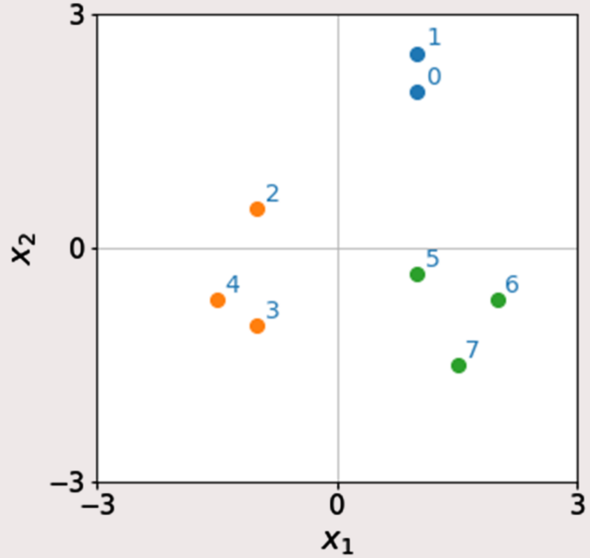


16

Agglomerative clustering: single-linkage

If we continue this procedure until L(N), then we obtain the following tree:





Cutting this tree at L(m), gives you m clusters.

Agglomerative clustering: linkage functions

Single-linkage is just one of many different functions used to merge clusters.

- **"Complete linkage":** $d(k, C) = \max_{c \epsilon C} d(k, c)$.
- **"Weighted average linkage":** $d(k, \{i, j\}) = \frac{1}{2}\big(d(k, i) + d(i, j)\big)$
- **"Unweighted average linkage":** $d\big(C_i, C_j\big) = \frac{1}{|C_i||C_j|} \sum_{k \in C_i} \sum_{l \in C_j} d(k, l)$
- **"Centroid linkage":** $d\big(C_i, C_j\big) = d(c_i, c_j)$ **where** c_i, c_j **are cluster centroids.**
- **"Ward's": this function describes the increase in variance with each merger.**

18

Limitations of hierarchical clustering

Q: What are the limitations of hierachical clustering algorithms?

1. **All pairwise distances are required before the algorithm can even start.**
   - Computing pairwise distances for all points is $O(N^2)$.
   - All those distances need to be kept in memory.
2. **You cannot skip steps by pre-specifying the number of clusters.**
3. **Similarity metrics are less intuitive for non-numeric data.**
   - For example, how do you compute similarity between genes?

19

Objective function-based clustering

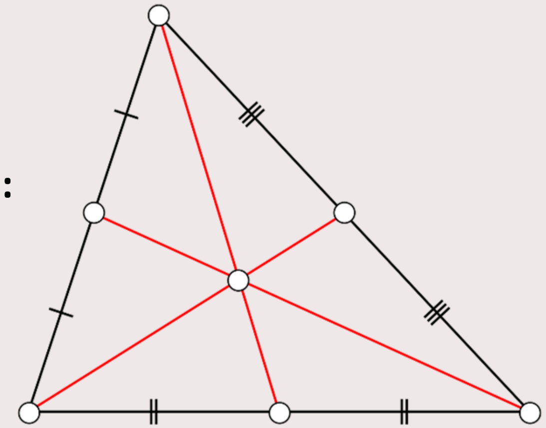
In hierarchical clustering, the data set is divided into sub-groups in a hierarchical way, using a similarity metric (or a dissimilarity metric, i.e distance)

In objective function-based clustering, an objective function is minimized to find the clustering solution

K-means clustering

K-means clustering is an algorithm that groups points based on distances to centroids.

- A *centroid* is the average of a set of points in high-dimensional space.
- Centroids are not part of the data set.

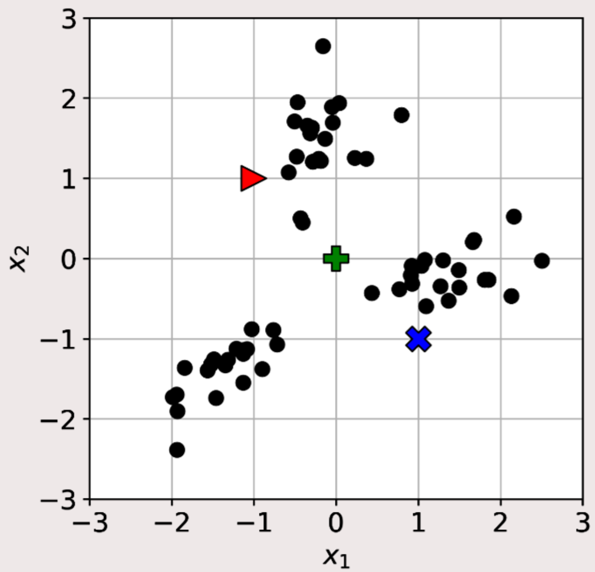


K-means is a simple iterative procedure consisting of two steps:

1. Assign points to clusters based on distance to centroids.
2. Update clusters centroids based on assigned points.

K-means clustering

K-means is best explained through a step-by-step demonstration.



First, we initialize a set of K centroids:

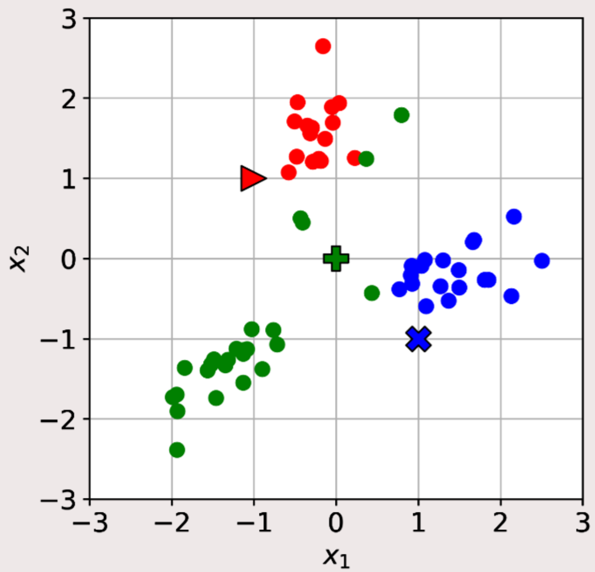
$$\boldsymbol{C = \{c_k : k = 1, \ldots K\}}.$$

These points should not start at the same coordinates.

If so, the distances to each centroid will be the same and all points will be tied for assignments.

K-means clustering

We then iterate the 2-step procedure:



1. **Compute the squared Euclidean distance between the data and the centroids,**

$$d(\boldsymbol{x}_i, \boldsymbol{c}_k) = \left|\left|\boldsymbol{c}_k - \boldsymbol{x}_i\right|\right|^2,$$

and assign each point to the closest centroid

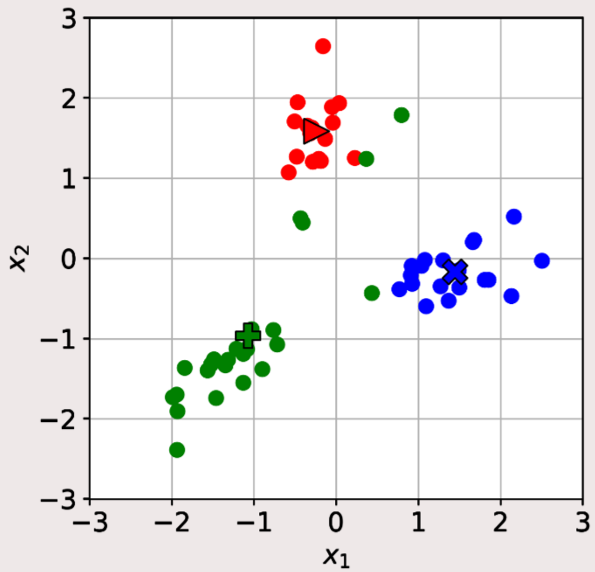
$$z_i = \arg\min_k d(\boldsymbol{x}_i, \boldsymbol{c}_k)$$

where z_i is the assignment variable.

23

K-means clustering

We then iterate the 2-step procedure:



2. **Update the centroids according to:**

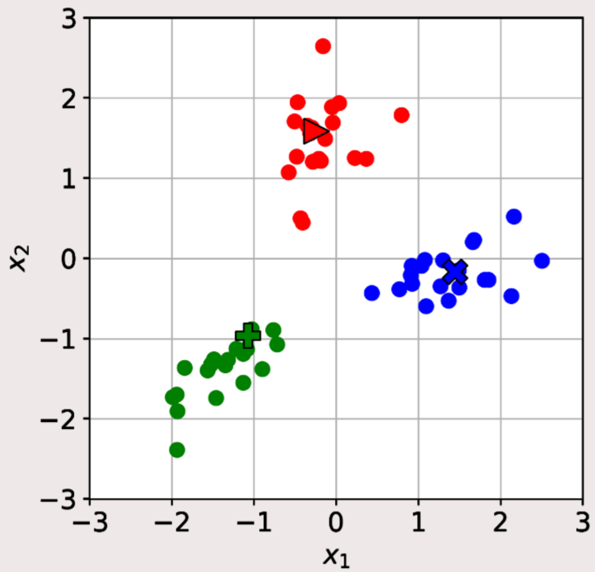
$$c_k = \frac{1}{|S_k|} \sum_{x_j \in S_k} x_j$$

where S_k is the set of points assigned to cluster k. This can alternatively be computed with:

$$c_k = \sum_{i=1}^{N} \frac{[z_i = k] \cdot x_i}{|z_i = k|}.$$

K-means clustering

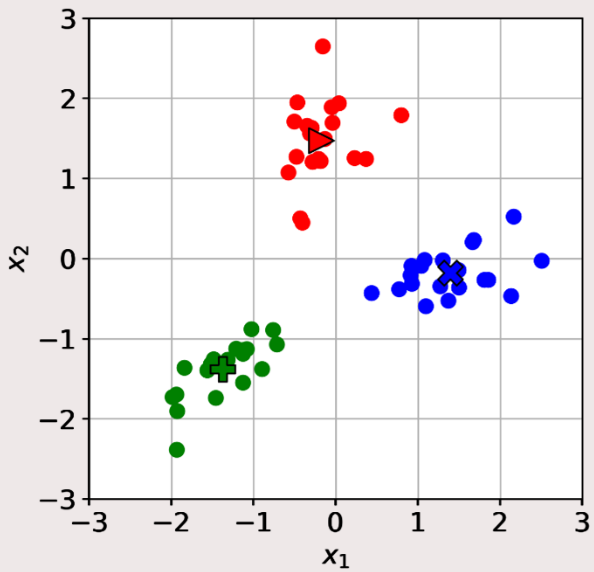
We then iterate the 2-step procedure until convergence:



1. **Assign each data point to a cluster based on minimal Euclidean distance.**
2. Update centroids according to the arithmetic mean of all points assigned to clusters.

K-means clustering

We then iterate the 2-step procedure until convergence:



1. Assign each data point to a cluster based on minimal Euclidean distance.
2. **Update centroids according to the arithmetic mean of all points assigned to clusters.**

26

Euclidean distance

The Euclidean distance between two points is the length of the vector traveling from one point to the other:

$$\boldsymbol{d(p,q)} = \sqrt{\sum_{j=1}^{M}\left(\boldsymbol{p_j} - \boldsymbol{q_j}\right)^2}$$

where $\mathbf{p}, \mathbf{q} \in \mathbb{R}^{M}$.

It is essentially a generalization of Pythagoras' theorem to higher dimensions.

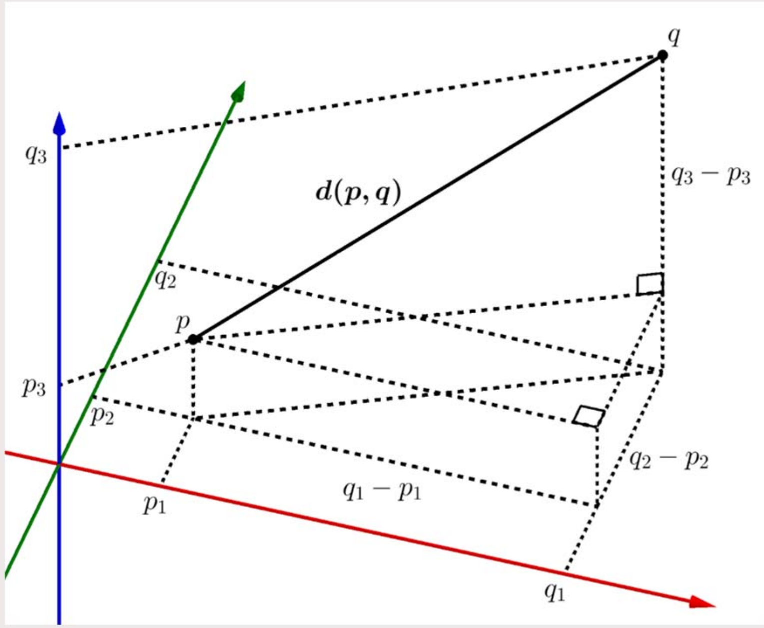
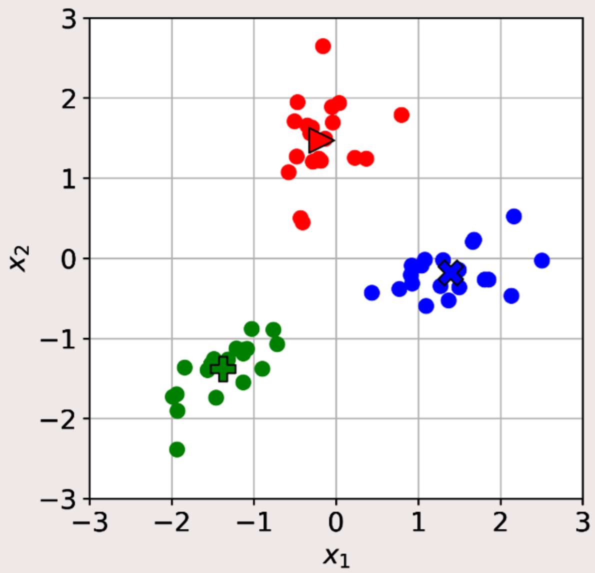


Figure adapted from *Wikipedia: Euclidean distance* (link).

K-means clustering

We then iterate the 2-step procedure until convergence:

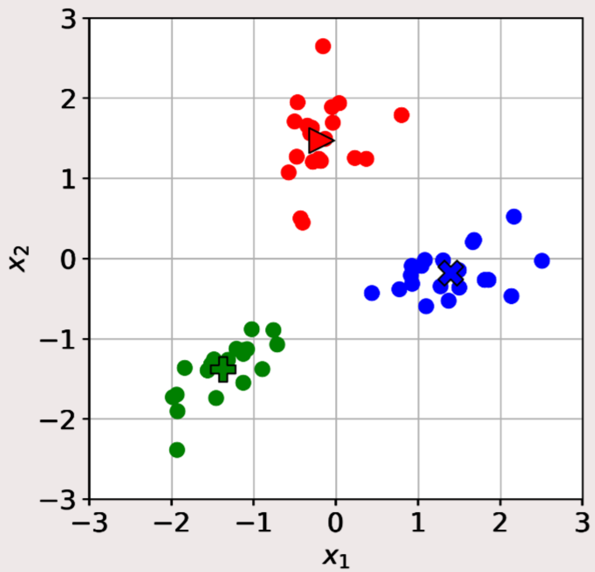


1. **Assign each data point to a cluster based on minimal Euclidean distance.**

2. Update centroids according to the arithmetic mean of all points assigned to clusters.

K-means clustering

We then iterate the 2-step procedure until convergence:



1. Assign each data point to a cluster based on minimal Euclidean distance.

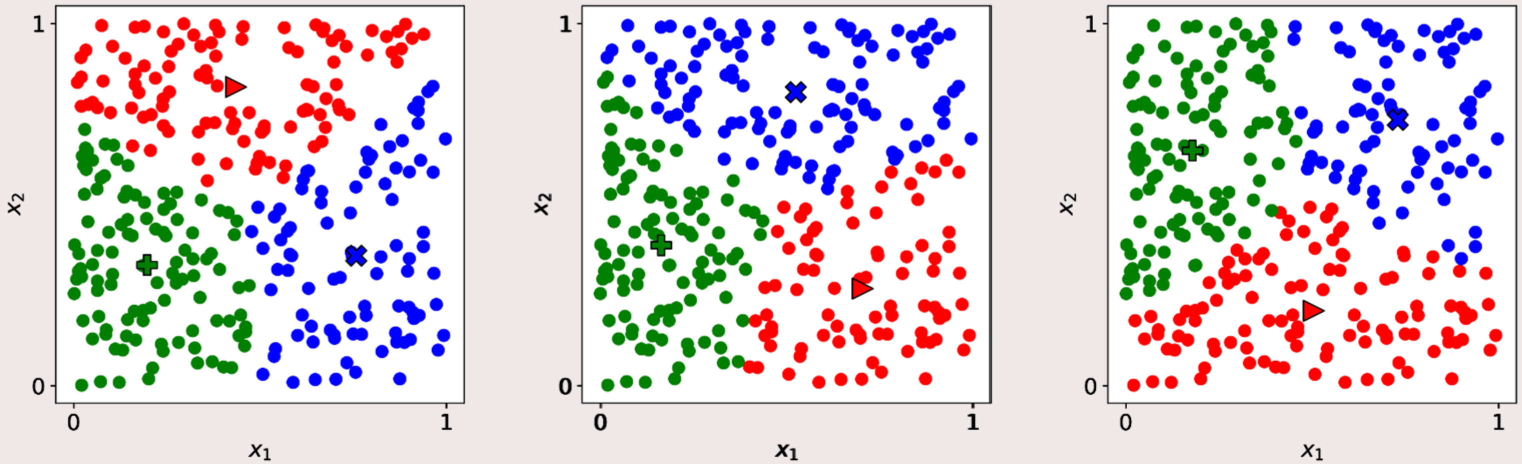
2. **Update centroids according to the arithmetic mean of all points assigned to clusters.**

29


Break
JADS
Jheronimus Academy of Data Science

Limitations of K-means

If the data is not clustered, then the centroids are random and meaningless

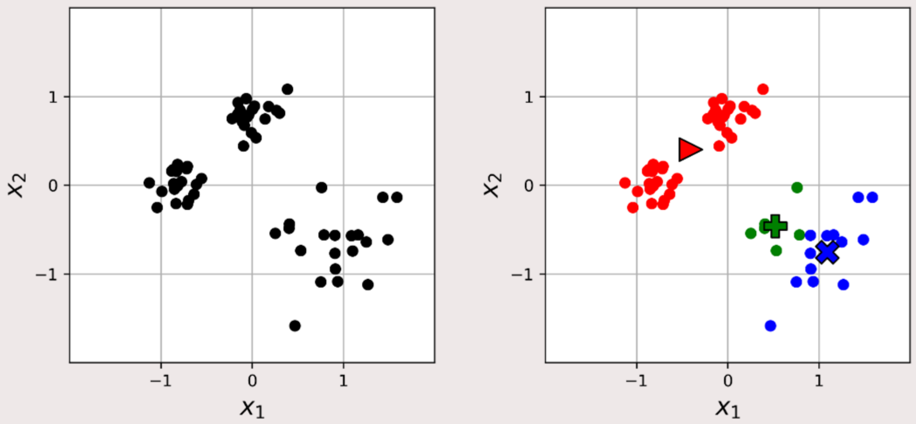


31

Limitations of K-Means

Q: What else could go wrong with K-Means?

K-Means is blind to cluster size and can get stuck in unintuitive solutions.



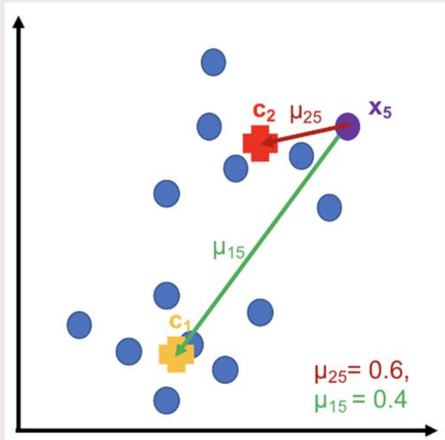
- **It merges the two smaller clusters on the top-left.**
- **It splits the larger cluster on the right-bottom.**

32

What about if the clusters cannot be separated well?

Crisp clustering algorithms
partition the data set into disjoint groups, i.e. each data point belongs to one cluster only
similarity is quantified using some metric

Fuzzy clustering algorithms
partition the data set into overlapping groups, i.e. each data point belongs to multiple clusters with varying degree of membership
similarity is quantified using some metric which is modified by membership values



https://towardsdatascience.com/fuzzy-c-means-clustering-with-python-f4908c714081

Fuzzy c-means

Partition data into overlapping sets based on similarity amongst patterns

Given the data $\mathbf{x}_k = [x_{1k}, x_{2k}, \ldots, x_{nk}]^T \in \Re^n, \quad k = 1, \ldots, N$

Find the fuzzy partition matrix:

$$\mathbf{U} = \begin{bmatrix} \mu_{11} & \cdots & \mu_{1N} \\ \vdots & \ddots & \vdots \\ \mu_{C1} & \cdots & \mu_{CN} \end{bmatrix}, \quad \mu_{ij} \in [0,1]$$

Divides *N* objects into *C* (overlapping) groups

and the cluster centres:

$$\mathbf{V} = \{\mathbf{v}_1, \ldots \mathbf{v}_C\}, \quad \mathbf{v}_i \in \Re^n$$

This is a generalization of (hard) k-means!

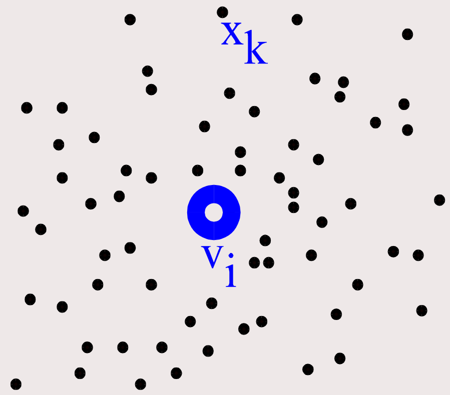
34

Solution with the Lagrangian method!

Fuzzy c-means clustering

Minimise objective function

$$J(\mathbf{X},\mathbf{U},\mathbf{V}) = \sum_{i=1}^{C}\sum_{k=1}^{N}\mu_{ik}^{m} d^2(\mathbf{x}_k,\mathbf{v}_i)$$



subject to

$$0 \le \mu_{ik} \le 1, \quad i = 1,\dots,C,\ k = 1,\dots,N$$ membership degree

$$\sum_{i=1}^{C}\mu_{ik} = 1, \quad k = 1,\dots,N$$ total membership

$$0 < \sum_{k=1}^{N}\mu_{ik} < N, \quad i = 1,\dots,C$$ no cluster empty

$m \in (1,\infty)$ is the fuzziness parameter

Fuzzy c-means algorithm

Initialization can be done either by initializing **V** or by initializing **U**

Repeat:

1. Compute cluster centers $\mathbf{v}_i = \frac{\sum_{k=1}^{N} \mu_{ik}^m \mathbf{x}_k}{\sum_{k=1}^{N} \mu_{ik}^m}$

Assumes partition matrix is fixed

2. Calculate distances $d_{ik}^2 = (\mathbf{x}_k - \mathbf{v}_i)^T (\mathbf{x}_k - \mathbf{v}_i)$

3. Update partition matrix $\mu_{ik} = \frac{1}{\sum_{j=1}^{C} (d_{ik}^2 / d_{jk}^2)^{1/(m-1)}}$

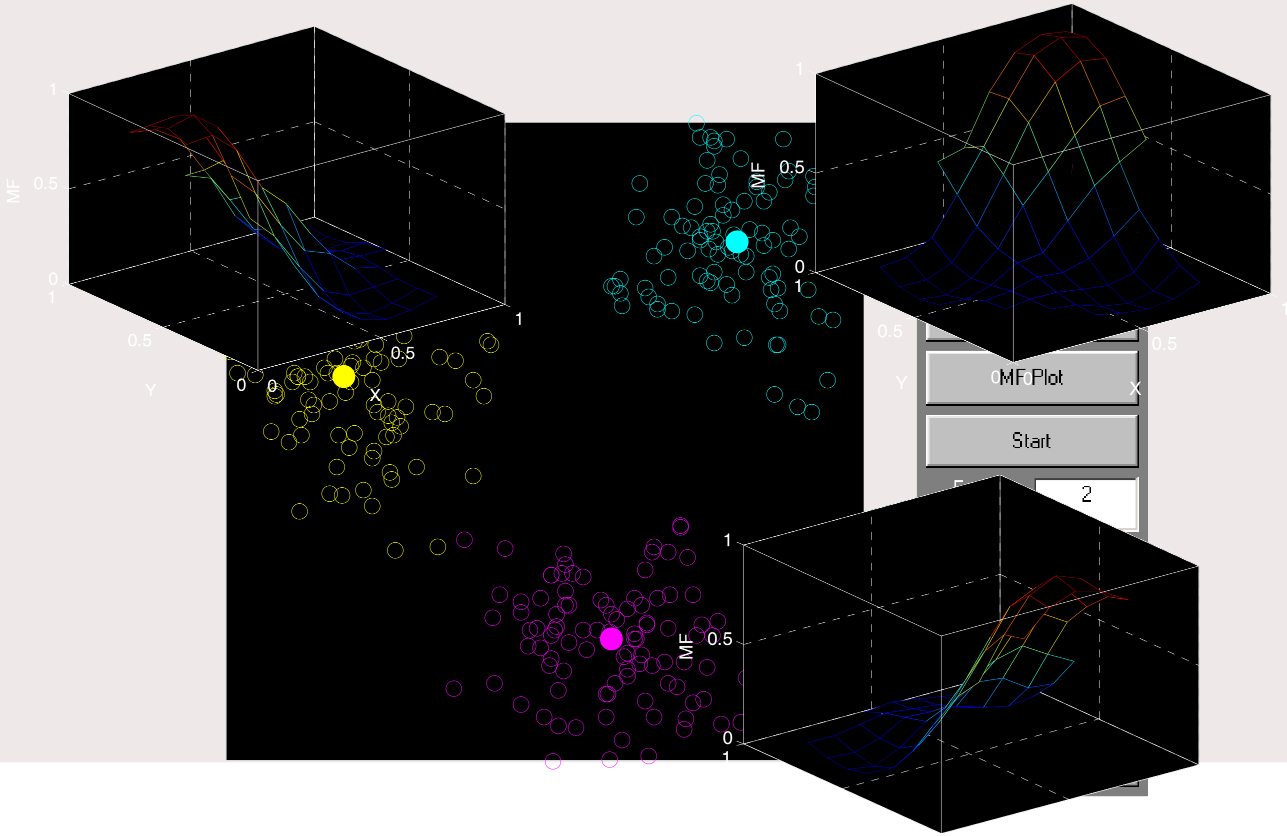
Assumes cluster centers are fixed

until $\|\Delta \mathbf{U}\| < \varepsilon$

Other stopping criteria are possible

36

Fuzzy c-means example (Matlab)



37

Example - Iris

150 Objects

4 Features

(a) Iris Setosa (b) Iris Versicolour (c) Iris Virginica

38

Distance measures

Euclidean norm:

$$d^2(\mathbf{x}_k, \mathbf{v}_i) = (\mathbf{x}_k - \mathbf{v}_i)^T (\mathbf{x}_k - \mathbf{v}_i)$$

Inner-product norm:

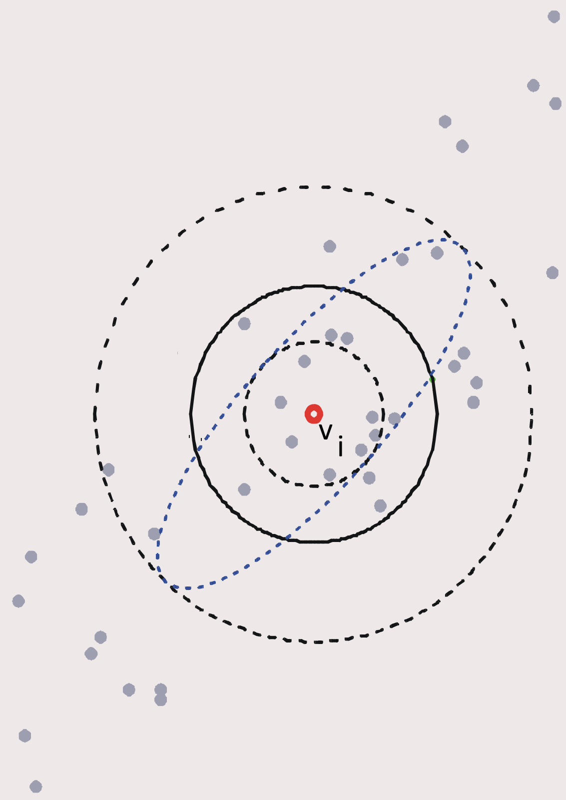
$$d^2(\mathbf{x}_k, \mathbf{v}_i) = (\mathbf{x}_k - \mathbf{v}_i)^T \mathbf{A} (\mathbf{x}_k - \mathbf{v}_i)$$

A is diagonal

Mahalanobis norm:

$$d^2(\mathbf{x}_k, \mathbf{v}_i) = (\mathbf{x}_k - \mathbf{v}_i)^T \mathbf{F}_i^{-1} (\mathbf{x}_k - \mathbf{v}_i)$$

Rotated clusters



Gustafson-Kessel clustering

Uses an adaptive distance metric

$$d^2(\mathbf{x}_k - \mathbf{v}_i) = (\mathbf{x}_k - \mathbf{v}_i)^T \mathbf{A}_i (\mathbf{x}_k - \mathbf{v}_i)$$

$$\mathbf{A}_i = |\mathbf{F}_i|^{1/n} \mathbf{F}_i^{-1}$$

Fuzzy covariance matrix

$$\mathbf{F}_i = \frac{\sum_{k=1}^{N} (\mu_{ik})^m (\mathbf{x}_k - \mathbf{v}_i)(\mathbf{x}_k - \mathbf{v}_i)^T}{\sum_{k=1}^{N} (\mu_{ik})^m}$$

Clusters are constrained by volume

Clusters adapt themselves to the shape and location of data

40 TU/e

GK algorithm

Repeat:

1. Compute cluster centers

$$\mathbf{v}_i = \frac{\sum_{k=1}^{N} \mu_{ik}^m \mathbf{x}_k}{\sum_{k=1}^{N} \mu_{ik}^m}$$

Assumes partition matrix is fixed

2. Calculate covariance matrices and distances

$$d_{ik}^2 = |\mathbf{F}_i|^{1/n} (\mathbf{x}_k - \mathbf{v}_i)^T \mathbf{F}_i^{-1} (\mathbf{x}_k - \mathbf{v}_i)$$

$$\mathbf{F}_i = \frac{\sum_{k=1}^{N} (\mu_{ik})^m (\mathbf{x}_k - \mathbf{v}_i)(\mathbf{x}_k - \mathbf{v}_i)^T}{\sum_{k=1}^{N} (\mu_{ik})^m}$$

3. Update partition matrix

$$\mu_{ik} = \frac{1}{\sum_{j=1}^{C} (d_{ik}^2 / d_{jk}^2)^{1/(m-1)}}$$

Assumes cluster centers are fixed

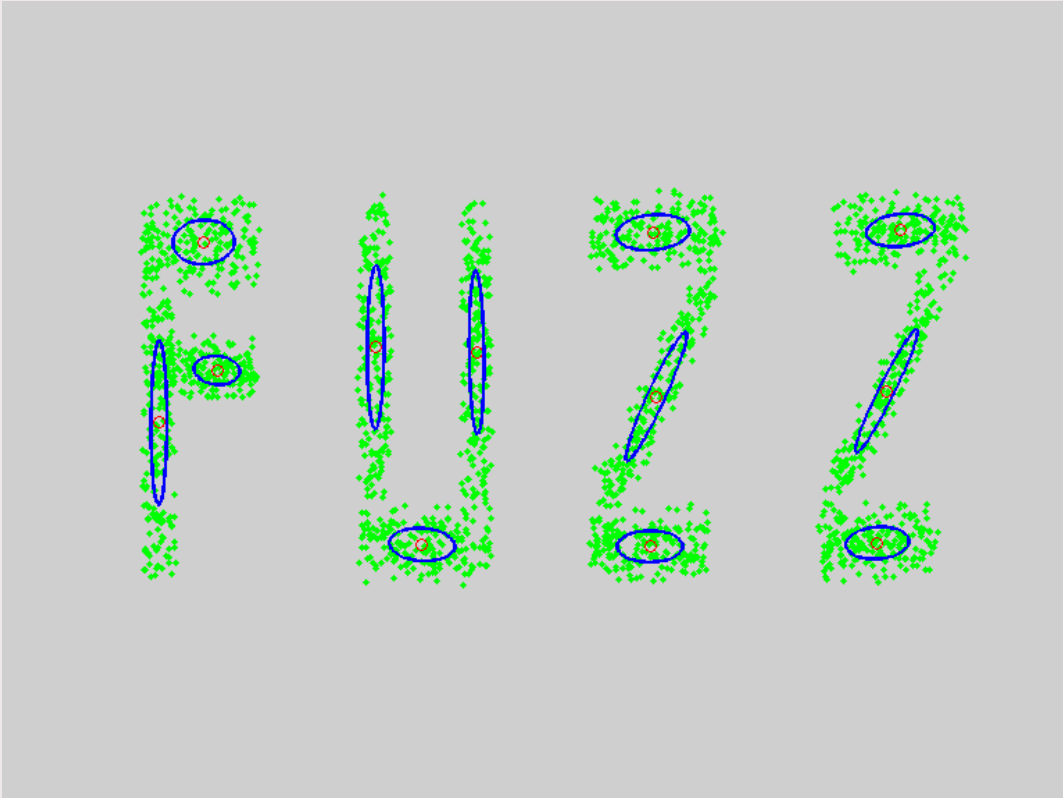
until

$$\|\Delta \mathbf{U}\| < \varepsilon$$

Other stopping criteria are possible

41

GK algorithm example



42

Issues in fuzzy clustering

- Normalization
- Initialization
- Cluster volumes
- Number of clusters
- Cluster convexity
- Selection of parameters (e.g. fuzziness m)
- Categorical variables (ordinal and nominal)
- Missing values

Normalization

Can you compare measurements on different scales? (c.f. contrast enhancement)

Data box normalization

$$x'_{jl} = \frac{x_{jl} - \min_j x_{jl}}{\max_j x_{jl} - \min_j x_{jl}}, \; j = 1,\ldots,N \text{ and } l = 1,\ldots,n$$

Standard deviation normalization (z-normalization)

$$x'_{jl} = \frac{x_{jl} - \overline{x_l}}{\sigma_l}, \; j = 1,\ldots,N \text{ and } l = 1,\ldots,n$$

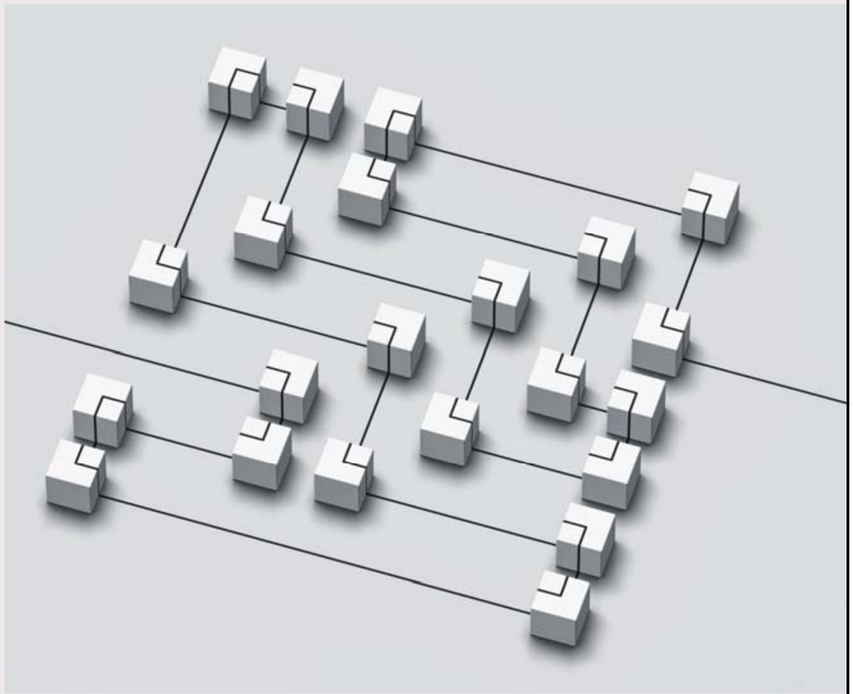
Adaptive distance metrics as in Gustafson-Kessel clustering are less sensitive to normalization

44

Initialization

How to avoid local minima during the optimization?

- **Randomly select a set of cluster prototypes V**
- **Randomly select a set of data points as cluster centers V**
- **Randomly initialize the partition matrix U**
- **Use information (e.g. cluster centre locations) from a separate clustering step**
- **Initialize centres far away from data**

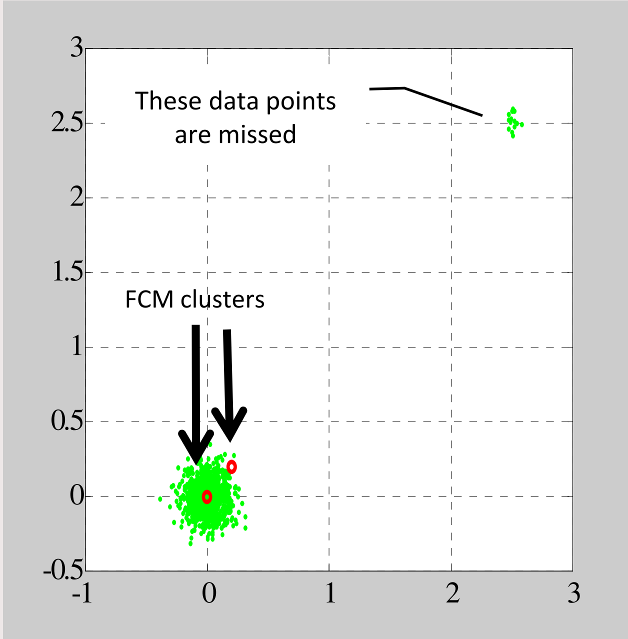


Cluster volumes

- How large should clusters be?
- Extent of clusters
- Data density and distribution
- Size of cluster prototypes

Cluster volume can be a parameter in Gustafson-Kessel clustering

FCM cluster centres



Cluster validity

- How good are the clustering results?
  - Correct number of clusters?
  - Well-separated clusters?
  - Compact clusters?
- Cluster validity measures try to quantify the answers to these questions in a formula
- Optimal number of clusters at a local minimum of the validity measure

47

Validity measures

Gath and Geva index

$$S_G = \sum_{i=1}^{C} \sqrt{\left| \frac{\sum_{k=1}^{N} (\mu_{ik})^m (\mathbf{x}_k - \mathbf{v}_i)(\mathbf{x}_k - \mathbf{v}_i)^T}{\sum_{k=1}^{N} (\mu_{ik})^m} \right|} + \beta C$$

Xie-Beni index

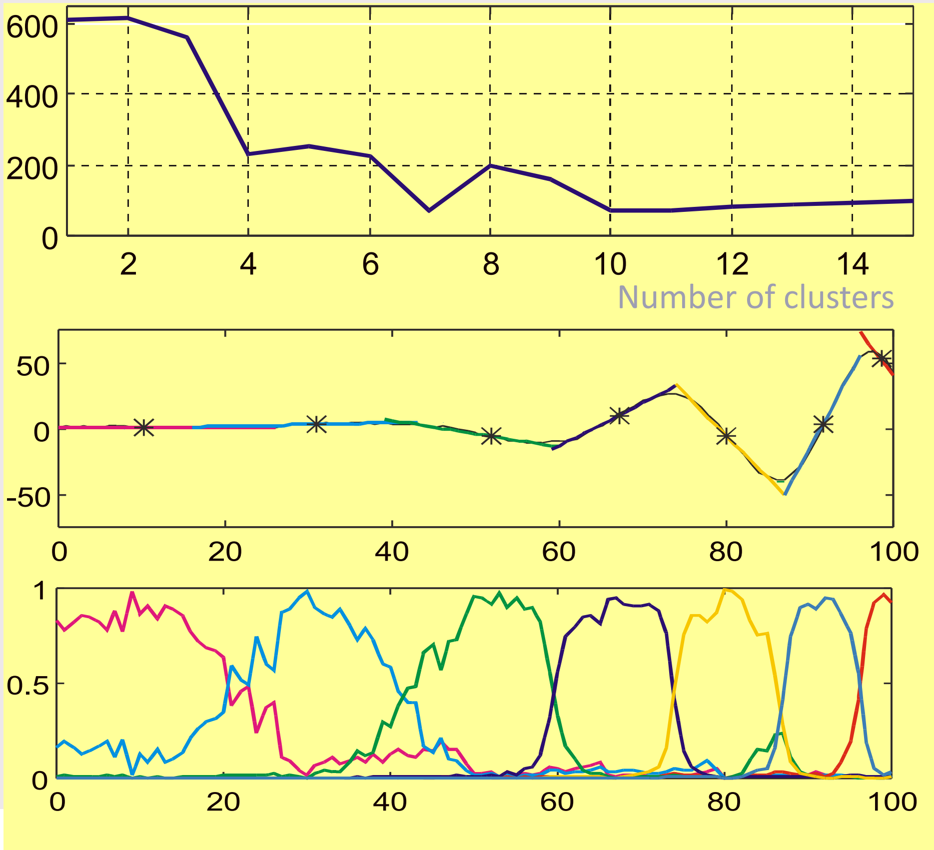
$$S_X = \frac{\sum_{i=1}^{C} \sum_{k=1}^{N} \mu_{ik}^m d^2(\mathbf{x}_k, \mathbf{v}_i)}{N \left(\min_{i,j,i \neq j} d^2(\mathbf{v}_i - \mathbf{v}_j) \right)}$$

Validity measures - example

Validity
(Gath and Geva)

Local Models
(Gustafson-Kessel)

Clusters



49

Cluster merging

Select

Select number of clusters larger than needed and do clustering

Merge

Merge clusters that are compatible

Repeat

Cluster again and continue merging until there are no compatible clusters

Measure

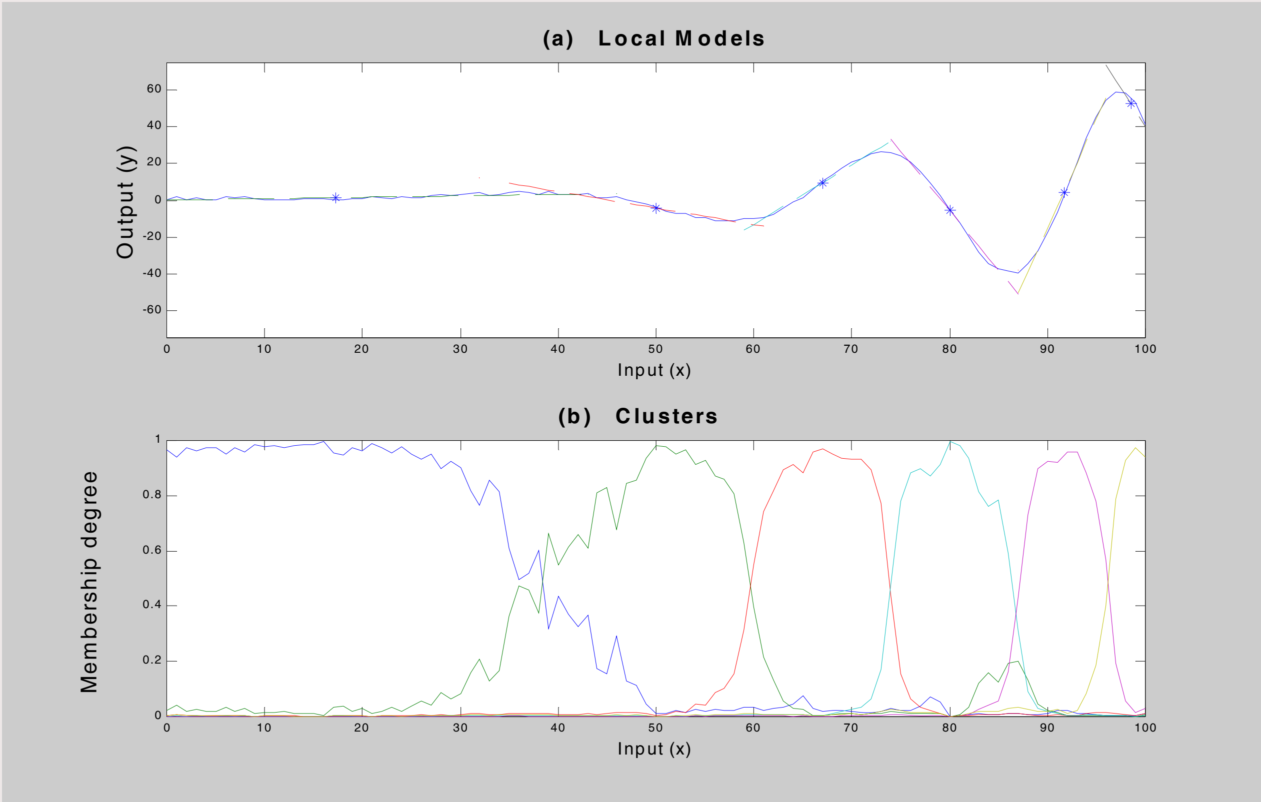
Cluster compatibility measured by

- Compatibility criteria How close are clusters? How similar are their characteristics? Etc.
- Similarity measures

50

TU/e

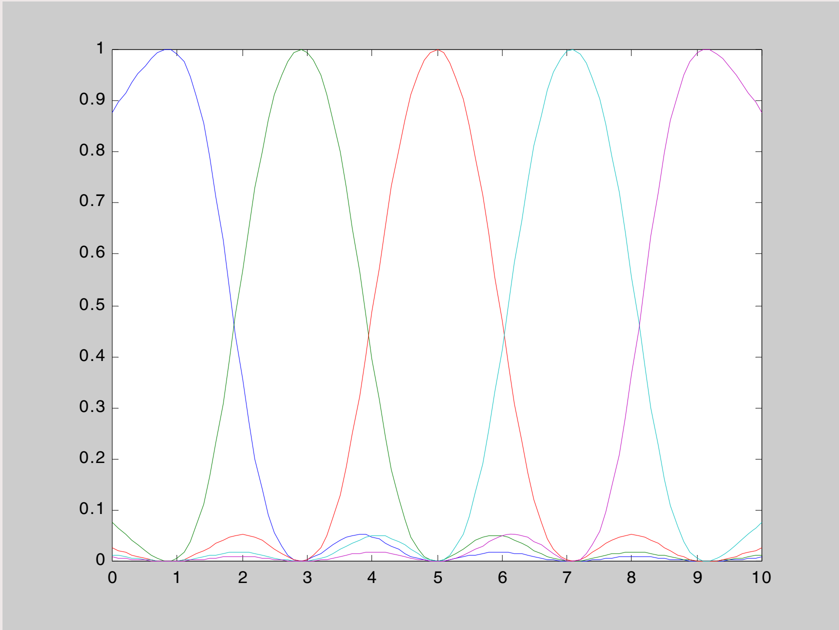
Cluster merging - example



Effect of probabilistic constraint

Probabilistic constraint:

$$\sum_{i=1}^{C} \mu_{ik} = 1, k = 1, \ldots, N$$



Problematic if a data point lies far away from all clusters (e.g. outliers)

Leads to non-convex clusters

Possibilistic clustering: possibilistic c-means

Minimize the objective function:

$$J(\mathbf{X},\mathbf{U},\mathbf{V},\boldsymbol{\eta}) = \sum_{i=1}^{C}\sum_{k=1}^{N} \mu_{ik}^{m} d^{2}(\mathbf{x}_k,\mathbf{v}_i) + \sum_{i=1}^{C}\eta_i \sum_{k=1}^{N}(1-\mu_{ik})^{m}$$

$m \in (1,\infty)$ is the fuzziness parameter

- η determine the size of the clusters
- suitable values from average inter-cluster distance

$$\eta_i = \frac{\sum_{k=1}^{N}\mu_{ik}^{m} d_{ik}^{2}}{\sum_{k=1}^{N}\mu_{ik}^{m}}$$

53

The optimization problem can now be decomposed into C independent optimization problems

Possibilistic clustering algorithm

Repeat:

1. Compute cluster centers $$\mathbf{v}_i = \frac{\sum_{k=1}^{N} \mu_{ik}^m \mathbf{x}_k}{\sum_{k=1}^{N} \mu_{ik}^m}$$

2. Calculate distances $$d_{ik}^2 = (\mathbf{x}_k - \mathbf{v}_i)^T \mathbf{A} (\mathbf{x}_k - \mathbf{v}_i)$$

3. Update partition matrix $$\mu_{ik} = \frac{1}{1 + \left(\frac{d_{ik}^2}{\eta_i^2} \right)^{\frac{1}{m-1}}}$$

Membership value does not depend on the membership to other clusters

until $\|\Delta \mathbf{U}\| < \varepsilon$

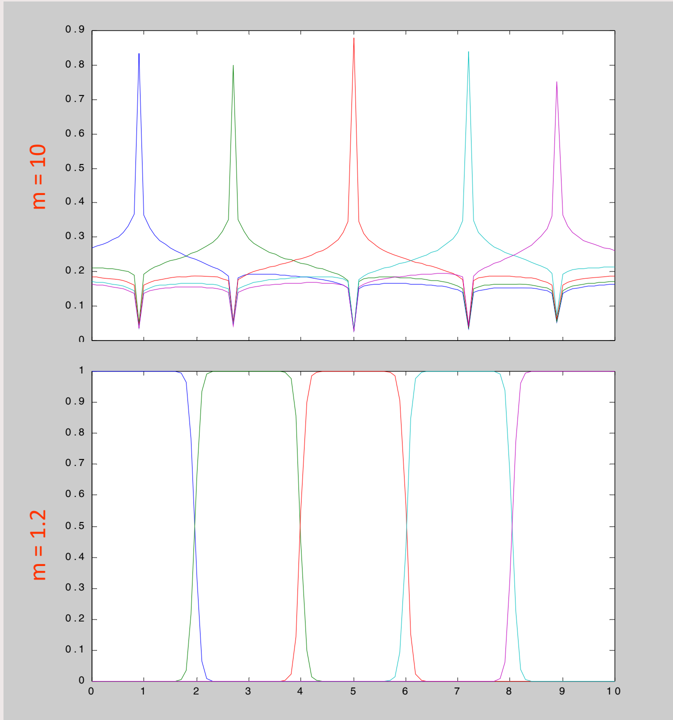
54

Effect of fuzziness index

As m increases, clusters overlap more; their centers become more isolated

As m decreases, clusters overlap less; they become crisp

Often m=2 is selected



Recap

- Clustering: group data such that within group similarity I slarger than between group similarity
- Hierarchical clustering: generates a tree graph
- K-means: minimizes total cluster scatter
- Fuzzy clustering: assigns data to multiple clusters with different membership
- Fuzzy clustering algorithms: FCM, GK, PCM
- Cluster validity measures help estimate correct number of clusters
- Cluster merging also helps determine correct number of clusters

Questions?

57