Australian National University

Fuzzy Logic

Fuzzy Clustering

COMP4660/8420 - Neural Networks, Deep Learning and Bio-inspired Computing

Human Centred Computing

Schedule

- Introduction and Basics
- Fuzzy rules and rule base systems
- Fuzzy clustering
- Sparse and hierarchical fuzzy systems
- Fuzzy signatures

Outline

- Clustering overview
- Fuzzy (soft) vs hard partitions
  - k-means overview
- Fuzzy c-means algorithm
- Example

What is clustering?

- Discover structures or patterns in a data set
  - Exploration
- Objects in each cluster are similar
- Objects in different clusters are dissimilar
- Unsupervised

Cluster Analysis

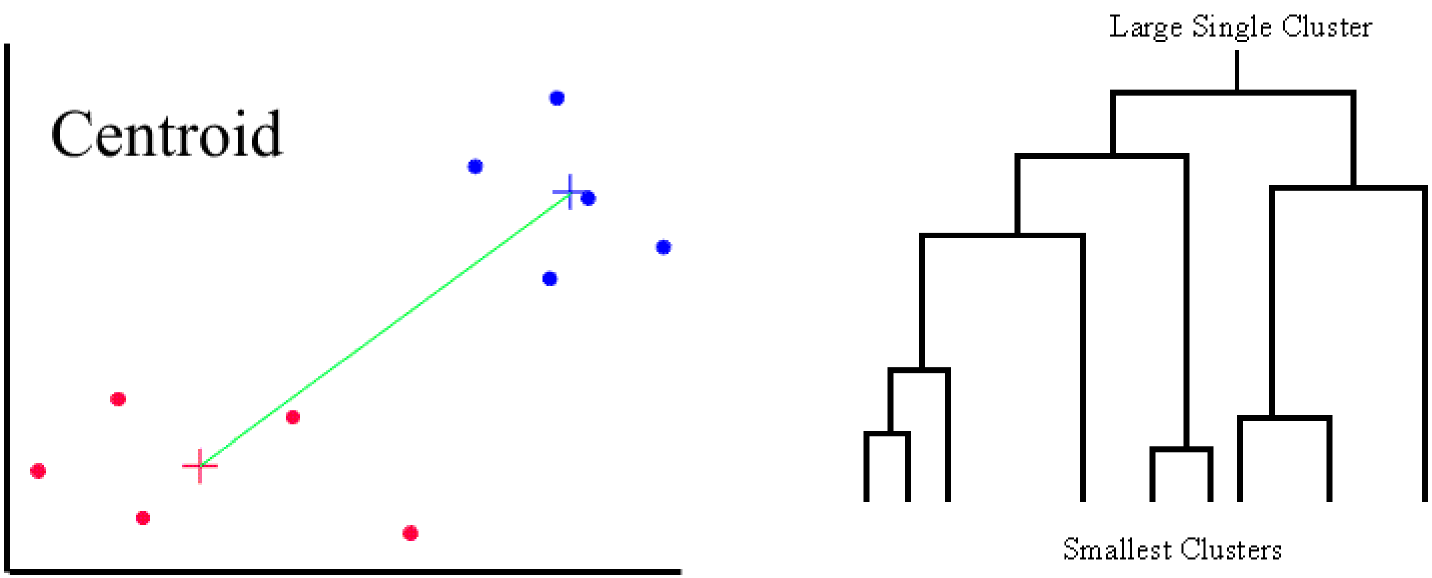
How many clusters?

Six Clusters

Four Clusters

Clustering Methods

- Hierarchical
- Centroid based
- Density based
- Distribution based

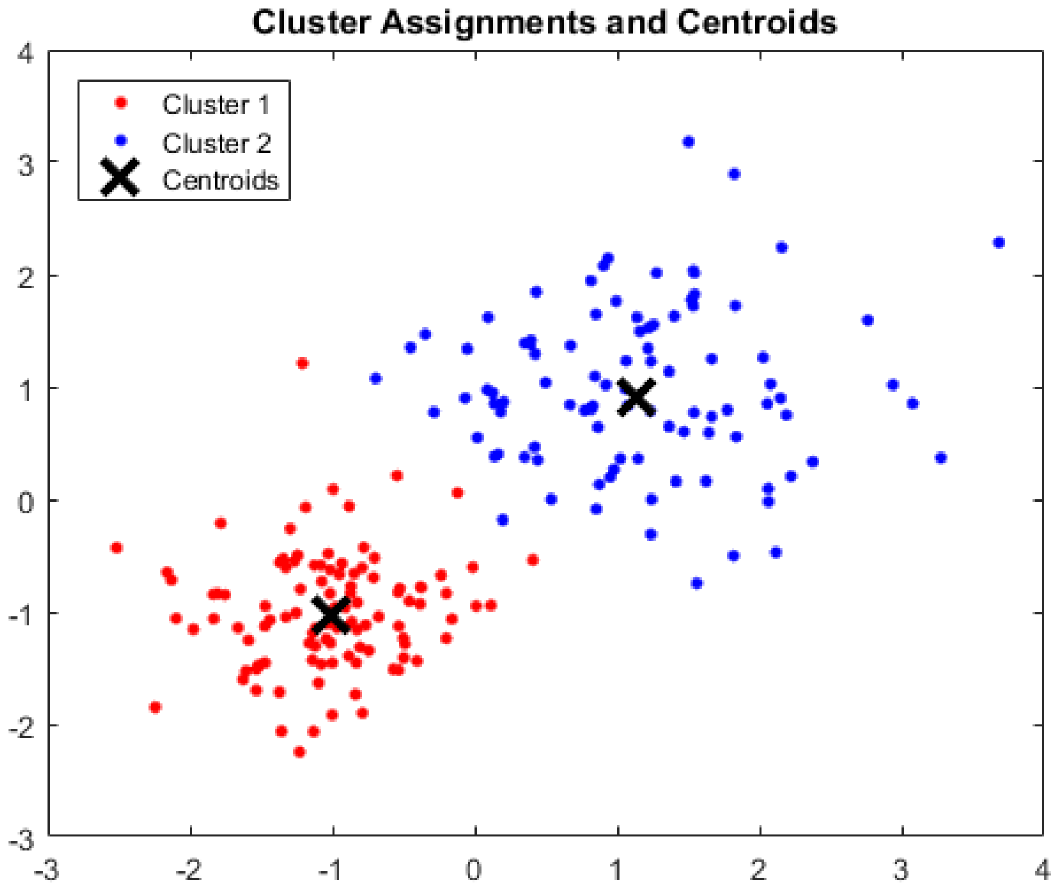


Examples of a a) centroid and b) hierarchical clustering

Partitional clustering

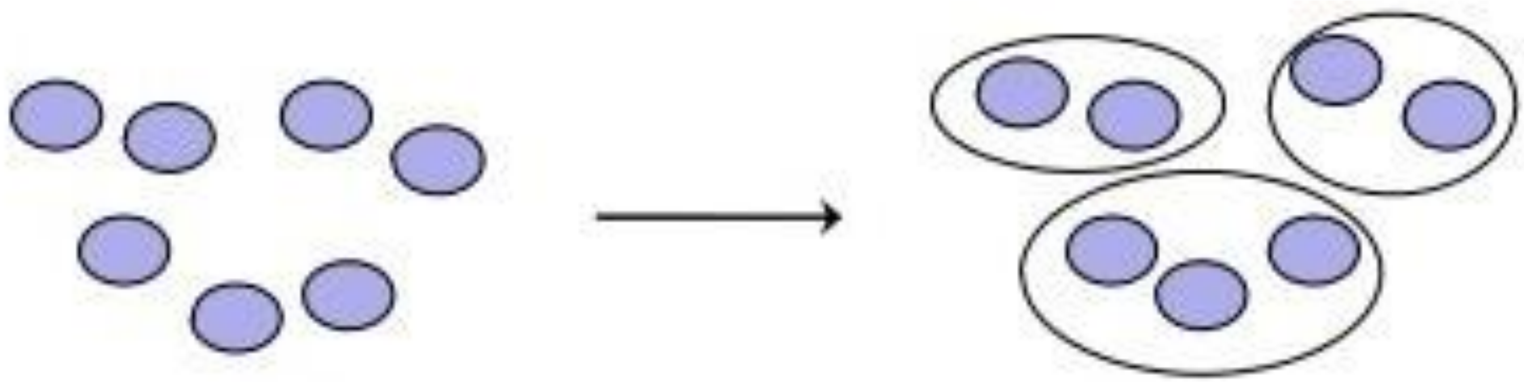
- Partition the data
- Centroid based
  - Objects in a cluster are similar to the “centre” of that cluster and dissimilar to the centres of other clusters
  - The centre of a cluster is called centroid
  - Each point is assigned to the cluster with the closest centroid
  - The number of clusters usually specified

Example of centroid clustering



Hard clustering

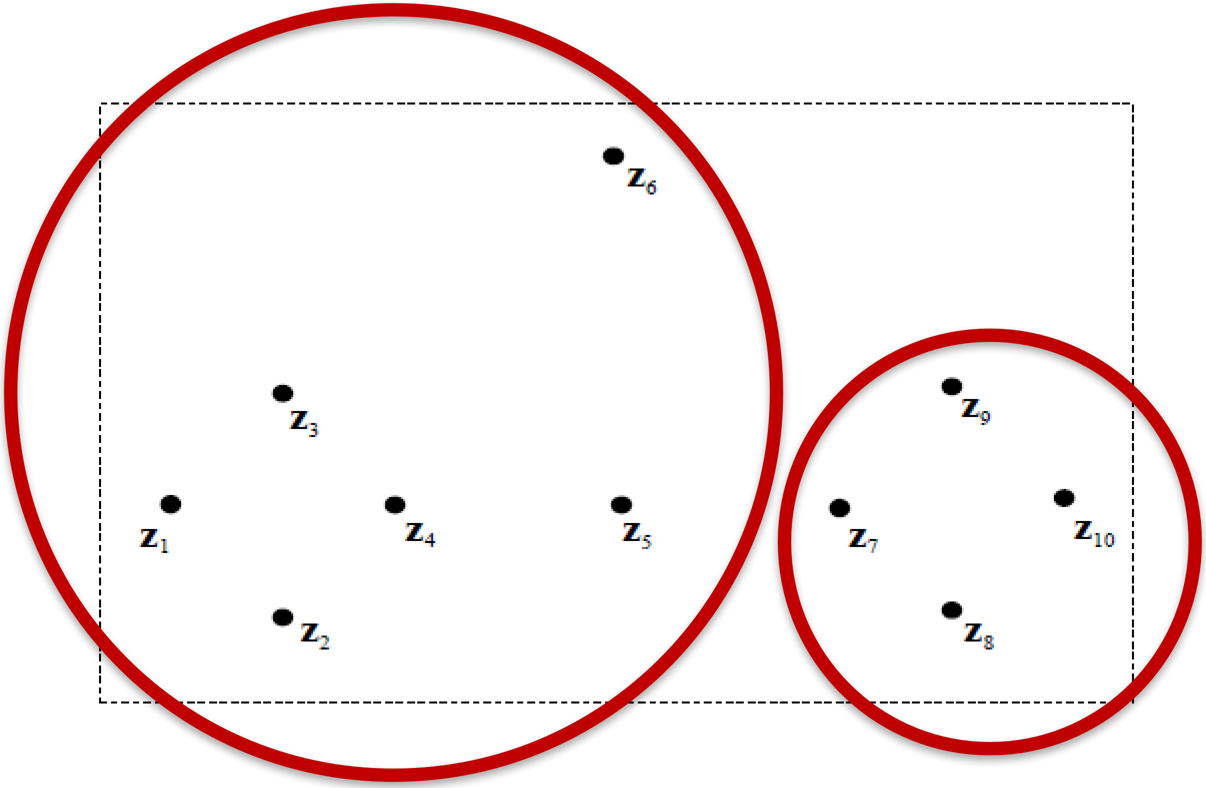
- each data point can belong to only one of the clusters



Hard Partitions

- Partition the data set X into *k* clusters.
- Assume that *k* is known, based on prior knowledge
- Membership in each cluster is 0 or 1
    - Each point can only belong to one cluster

Hard Partitions: Example

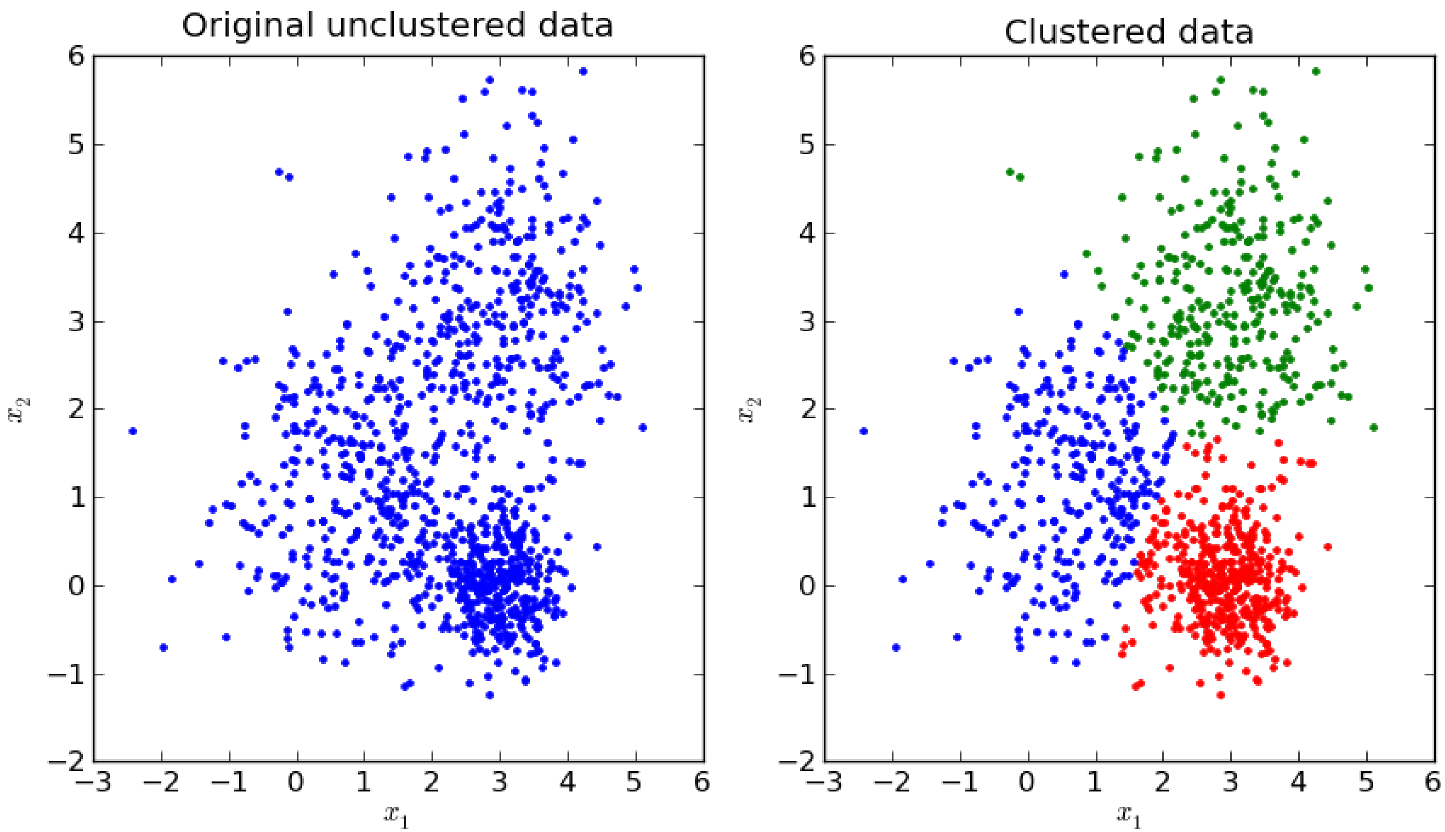


$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Example: k-means Clustering

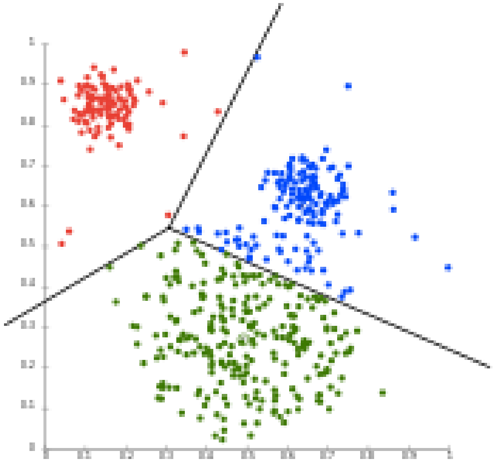
- Choose a number of clusters *k*
- Initialize cluster centers $c_1, \ldots, c_k$
  - Could pick *k* data points and set cluster centers to these points
  - Or could randomly assign points to clusters and take means of clusters
- For each data point, compute the cluster center it is closest to (using some distance measure) and assign the data point to this cluster
- Re-compute cluster centers (mean of data points in cluster)
- Stop when there are no new re-assignments

Example of k-means



Downside of k-means

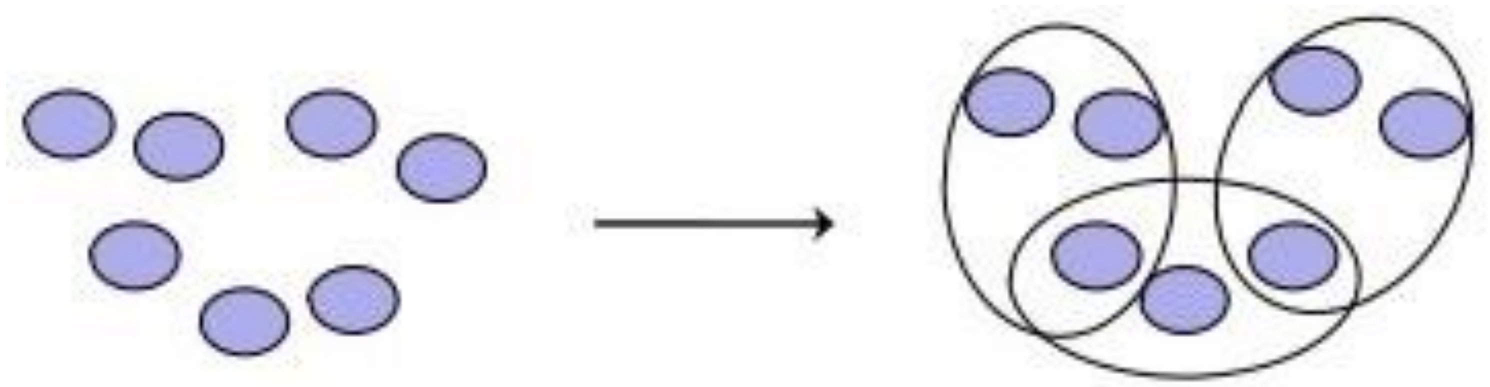
- One of the problems of the k-means algorithm is that it gives a hard partitioning of the data, that is to say that each point is attributed to one and only one cluster. But points on the edge of the cluster, or near another cluster, may not be as much in the cluster as points in the center of cluster.



Australian National University

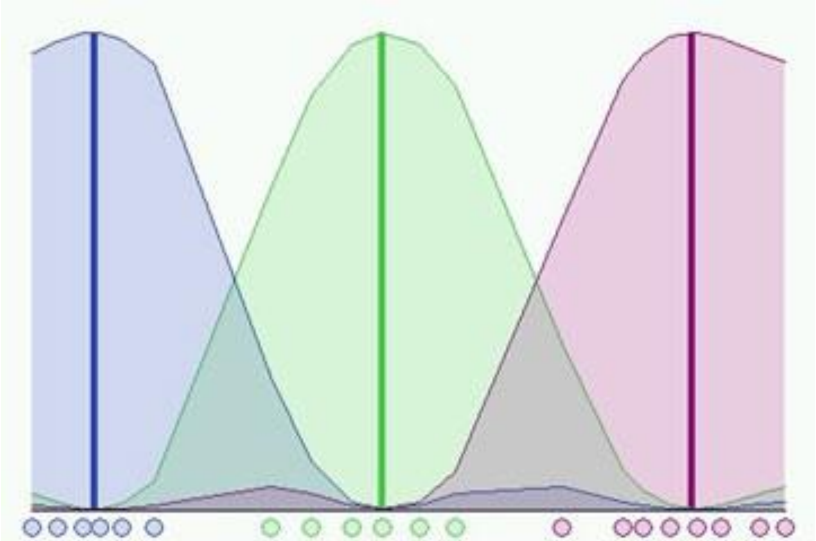
Soft clustering

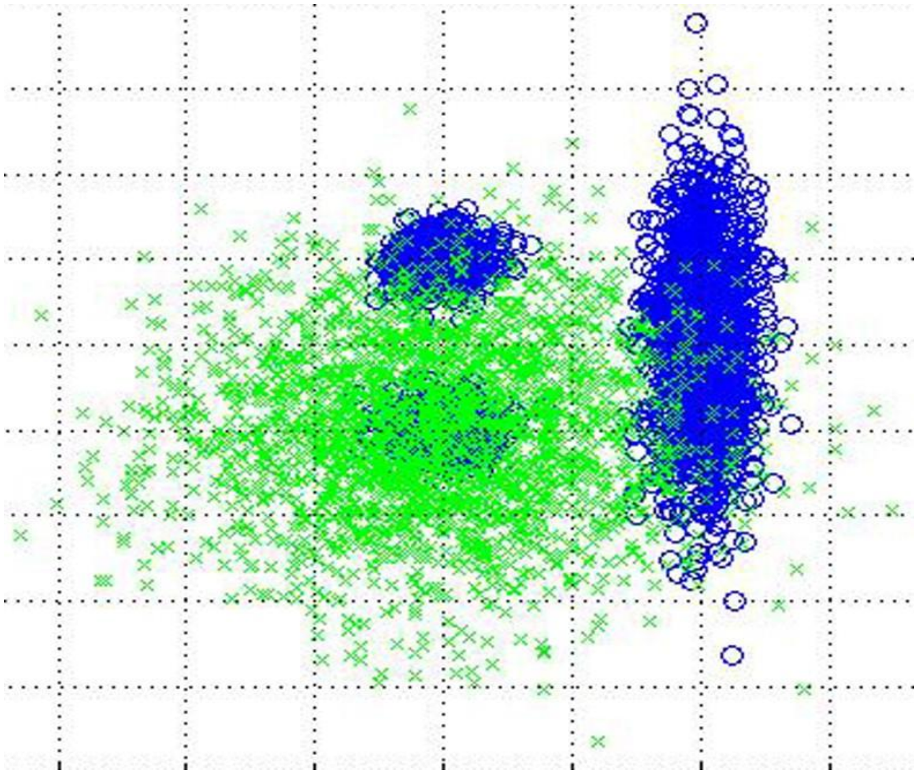
- Each data point can belongs to each cluster to a certain degree
  - Probablistic
  - Fuzzy!



Fuzzy clustering

- Every object belongs to every cluster with a membership between 0 and 1
- Clusters are therefore treated as fuzzy sets





Fuzzy Partition

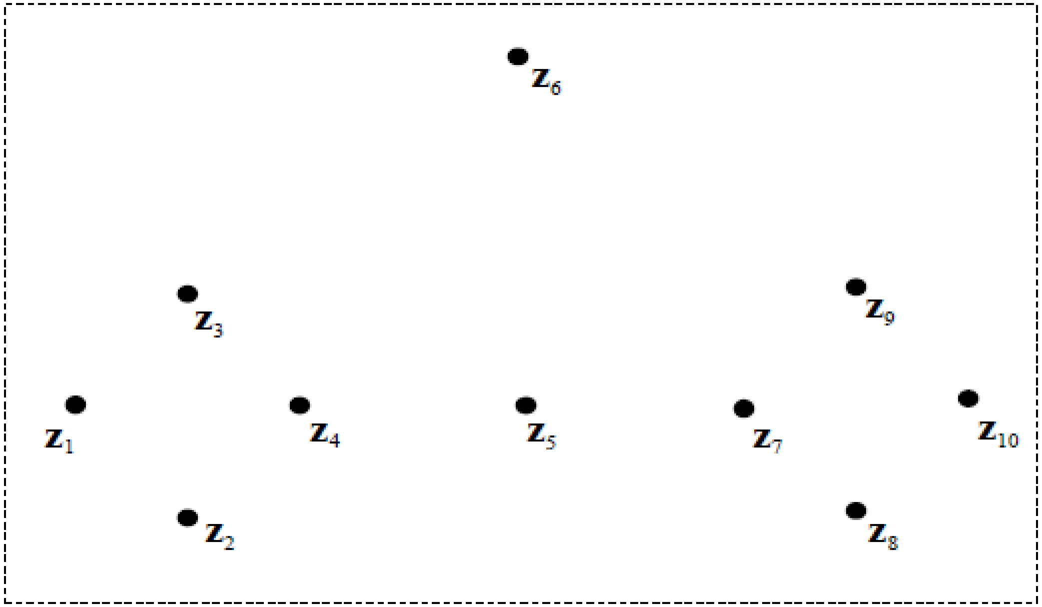
- Generalization of the hard partition to the fuzzy case follows directly by allowing μ_{ik} to attain real values in [0, 1].
- Conditions for a *fuzzy partition matrix* are given by Ruspini (1970)

$$\mu_{ik} \in [0,1], \quad 1 \le i \le c, \quad 1 \le k \le N,$$

$$\sum_{i=1}^{c} \mu_{ik} = 1, \quad 1 \le k \le N,$$

$$0 < \sum_{k=1}^{N} \mu_{ik} < N, \quad 1 \le i \le c.$$

Fuzzy Partitions: Example



$$U = \begin{bmatrix} 1.0 & 1.0 & 1.0 & 0.8 & 0.5 & 0.5 & 0.2 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.2 & 0.5 & 0.5 & 0.8 & 1.0 & 1.0 & 1.0 \end{bmatrix}$$

Why use fuzzy clustering?

- Useful in fuzzy modeling
  - Identification of fuzzy rules / sets to describe a black box system
- Fuzzy C-Means (Bezdek, 1981)
- Possibilistic C-Means (Krisnapuram & Keller, 1993)

Fuzzy c-means

- Most widely used
- Partitions to the data set into *c* fuzzy clusters
- Similar to k-means
- Goals
  - Maximise similarity within clusters
  - Minimise similarity between clusters

The algorithm in a nut shell

- Choose a number of clusters, *c*
- Assign randomly to each point coefficients for being in the clusters
- For each point, compute its coefficients of being in the clusters
- Compute the centroid for each cluster
- Repeat until the algorithm has converged
  - i.e., the coefficients' change between two iterations is no more than ε, the given sensitivity threshold

Objective function

- An objective function measures the overall dissimilarity within clusters
- By minimizing the objective function we can obtain the optimal partition

Objective function

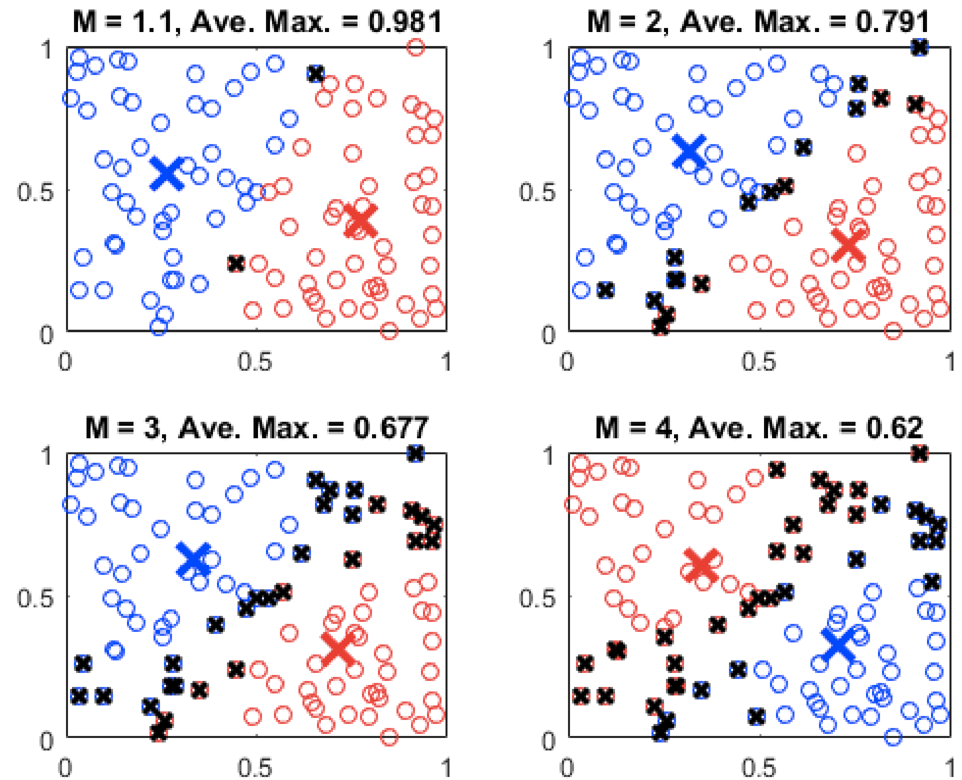
- The algorithm is based on minimization of the objective function:

$$J_m = \sum_{i-1}^{N} \sum_{j-1}^{C} u_{ij}^{m} \left\| x_i - c_j \right\|^2$$

- *m* is the fuzziness exponent and any real number greater than 1
- *N* is the number of data
- c is the number of clusters
- u_{ij} is the degree of membership of x_i in cluster j
- x_i is the *i*th of *d*-dimensional measured data
- c_j is the d-dimension centre
- || || is any norm expressing the similarity between any measured data and the centre

m – adjusting fuzzy overlap

- A max. membership value of 0.5 indicates that the point belongs to both clusters equally
- The data points marked with a black **x** have maximum membership values below 0.6.
  - These points have a greater degree of uncertainty in their cluster membership



Update the membership matrix and cluster centers

- Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership u_{ij} and the cluster centers c_j by:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}$$

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^{m}}$$

Stopping criteria

This iteration will stop when

$$max_{ij}\{\left|u_{ij}^{k+1} - u_{ij}^{k}\right|\} < \varepsilon$$

Where ε is a termination criterion between 0 and 1, whereas k are the iteration steps. This procedure converges to a local minimum or a saddle point of J_m.

Input

- Unlabelled data
- Set parameters as well:
  - the number of clusters, *c*, a real number that is greater than 1
  - The fuzziness exponent, *m*, which is a real number greater than 1
    - m = 1 is crisp
    - m = 2 is typical

Output

- A list of c centres, $C = \{c_1, \dots, c_c\}$
- A membership matrix U such that u_{ij} for $i = 1, \dots, c$ and $j = 1, \dots, n$
  - where m_{ij} is a value between [0,1] and is the degree to which element x_j belongs to the ith cluster
  - Summation of the membership of each point equals 1

Pseudocode

1. Randomly initialize the cluster membership values, μ_{ij}.
2. Calculate the cluster centers:

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m \cdot x_i}{\sum_{i=1}^{N} u_{ij}^m}$$

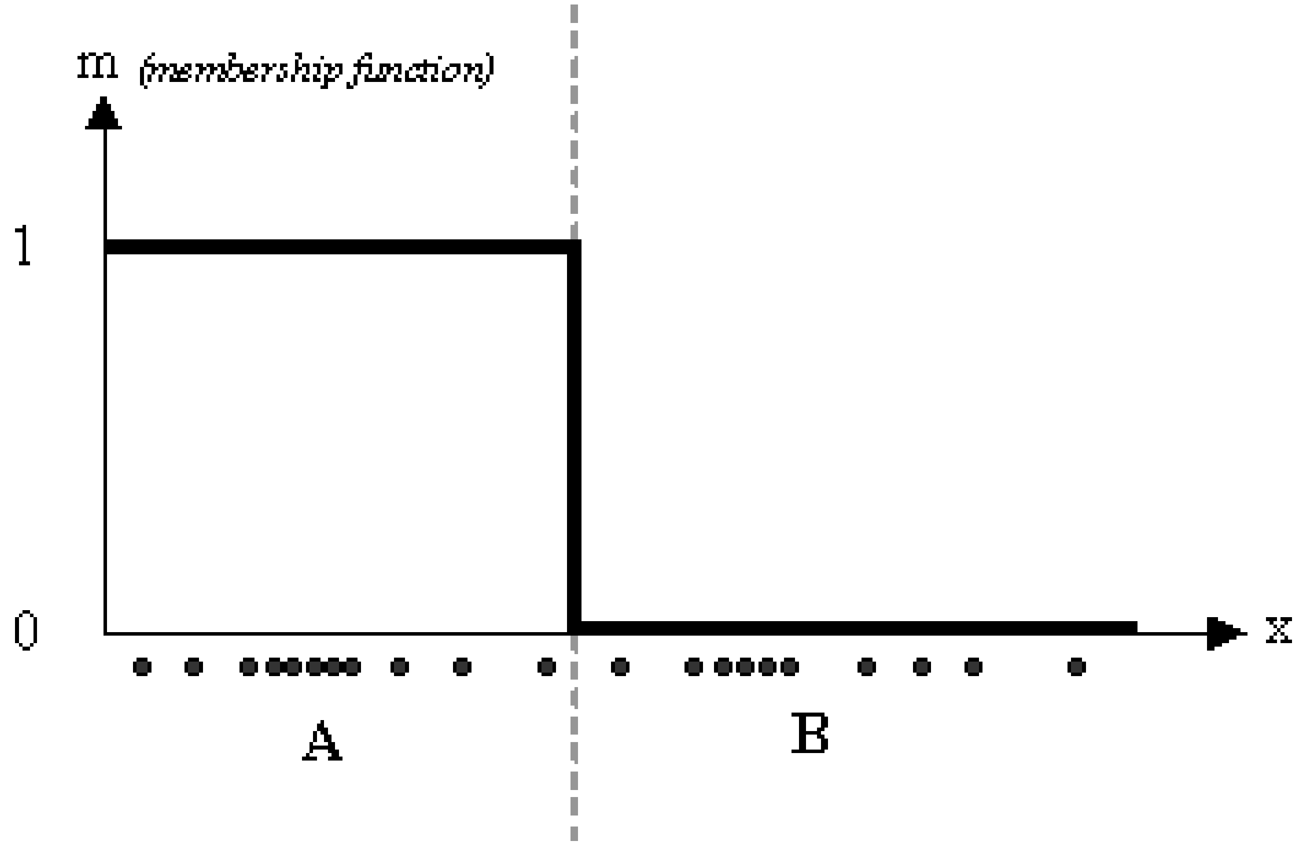
3. Update μ_{ij} according to the following:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\| x_i - c_j \|}{\| x_i - c_k \|} \right)^{\frac{2}{m-1}}}$$

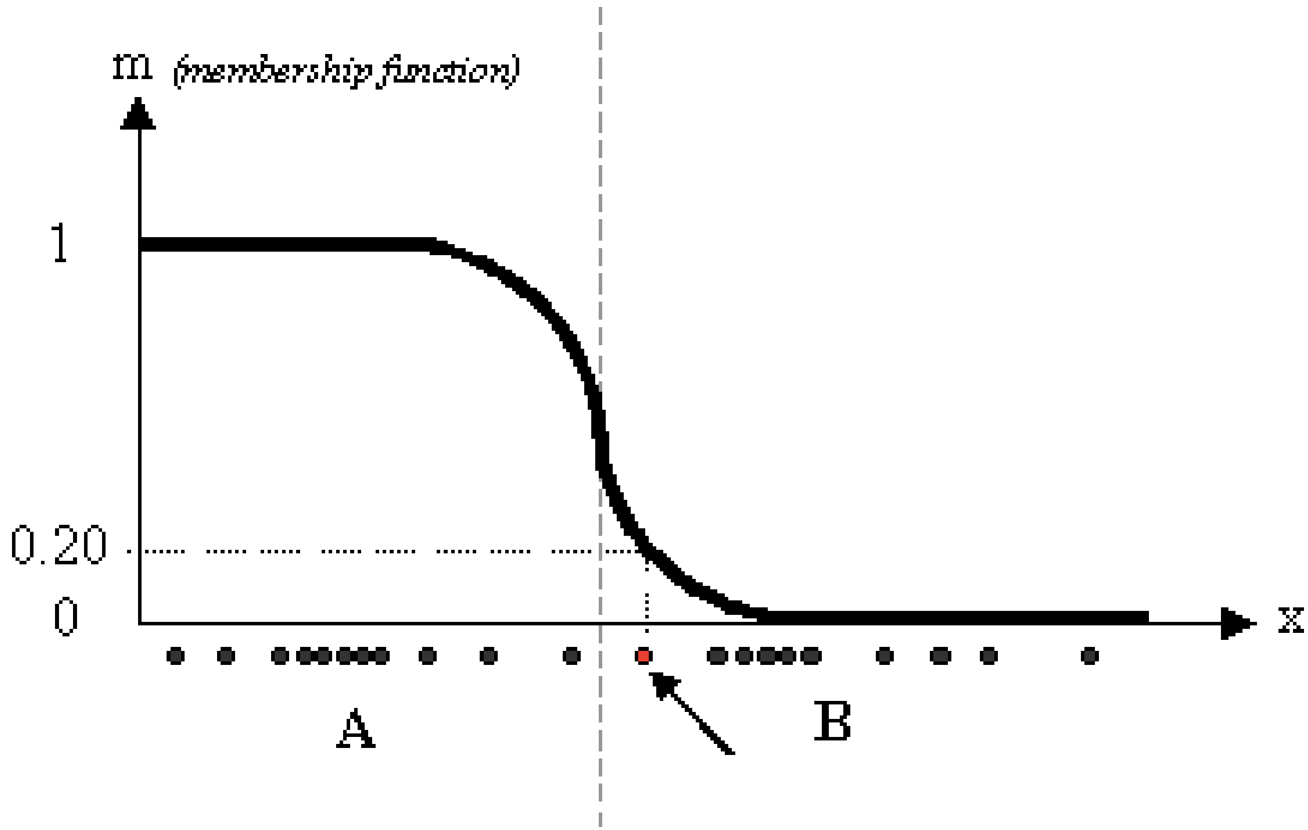
4. Calculate the objective function, J_m.

5. Repeat steps 2–4 until J_m improves by less than a specified minimum threshold or until after a specified maximum number of iterations.

K-means example



Fuzzy c-means example

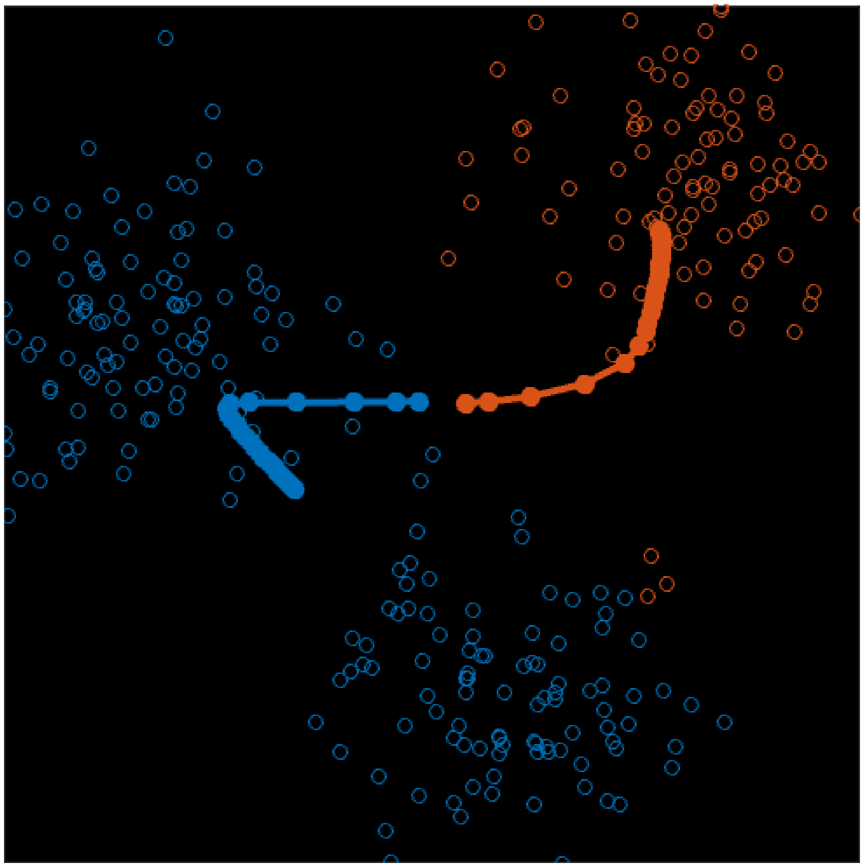


Pros and cons

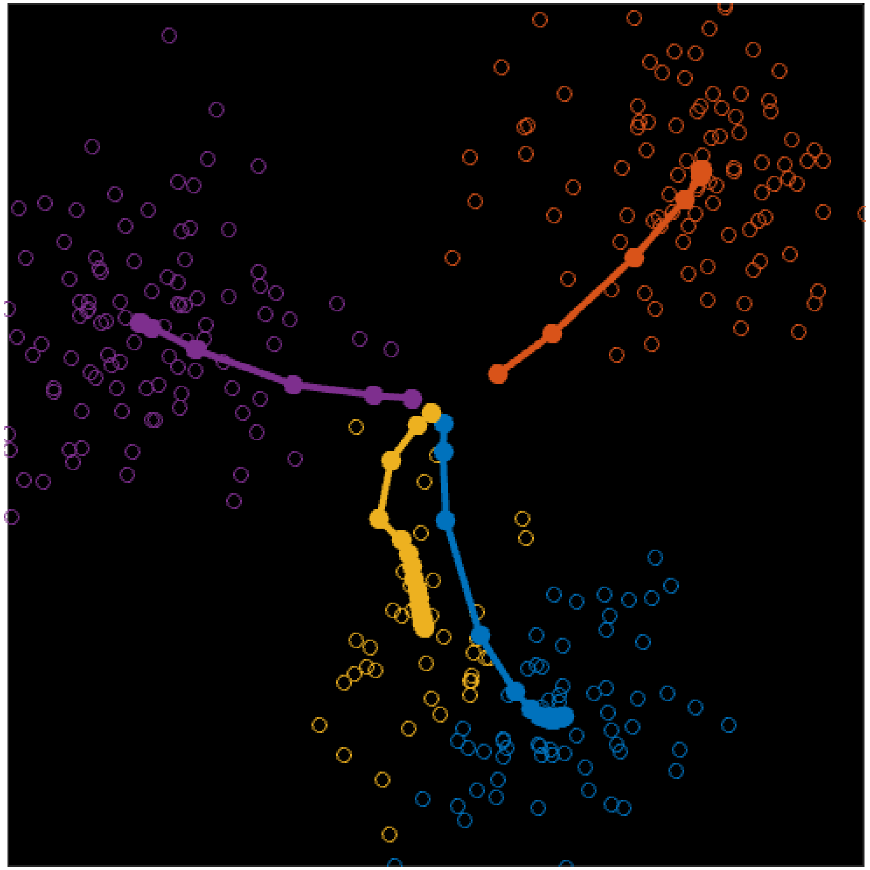
- Advantages
  - Unsupervised
  - Always converges
- Disadvantages
  - Specification of *c* clusters
  - Long computational time
  - Sensitivity to the initial guess (speed, local minima)
  - Sensitivity to noise
    - One expects low (or even no) membership degree for outliers (noisy points)

Cons: wrong number of clusters

2 clusters

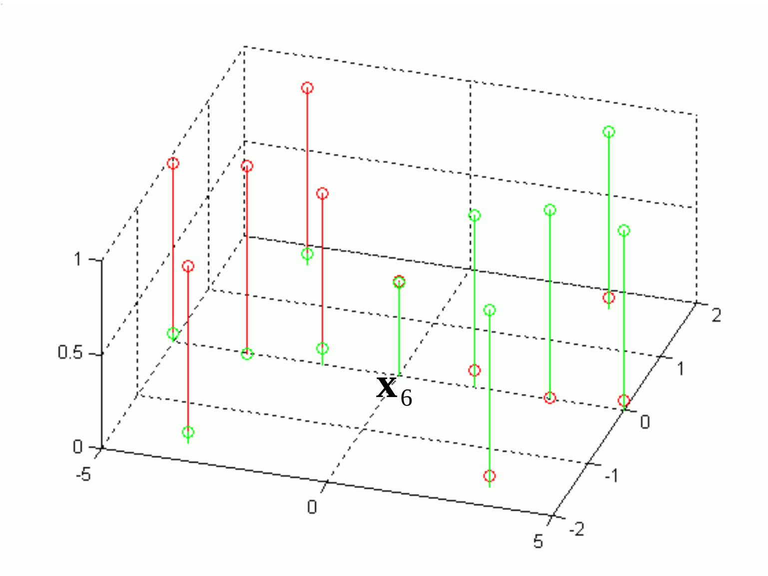


4 clusters

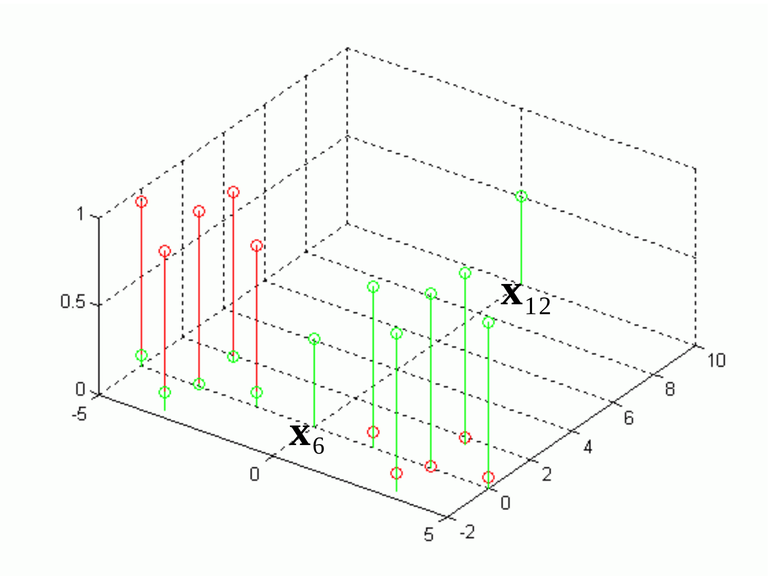


Cons: outliers

FCM on x_6
$u_1(x_6)= 0.5$ and $u_2(x_6) = 0.5$



FCM on x_6 & x_{12}
$u_1(x_6) = 0.5$ and $u_2(x_6) = 0.5$
$u_1(x_{12}) = 0.5$ and $u_2(x_{12}) = 0.5$



- X_{12} is an outlier but has the same membership degrees as X_6