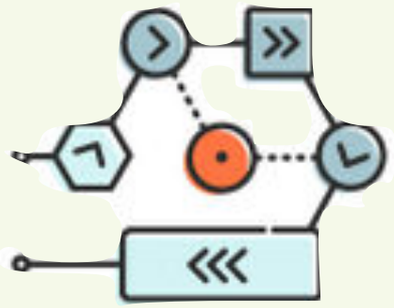
DEEP LEARNING



ALGORITHM

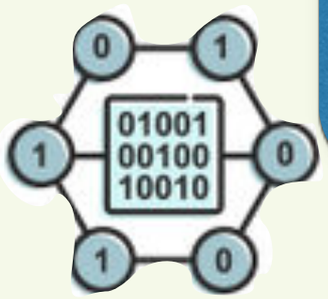
LEARNING

AI

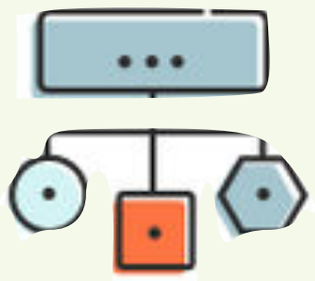
COURSE
"APLICAÇÕES DE INTELIGÊNCIA ARTIFICIAL"
University of Aveiro

Mestrado em Eng. Computational

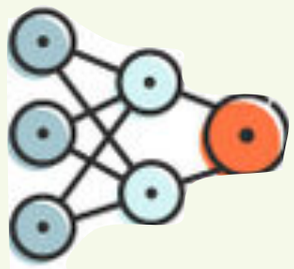
IMPROVES



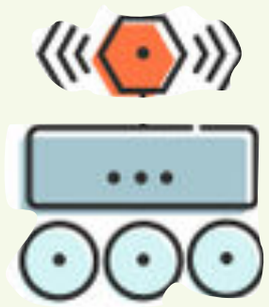
DATA MINING



CLASSIFICATION



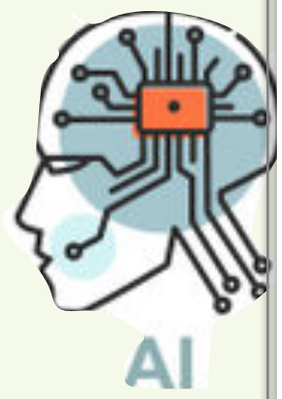
NEURAL NETWORKS

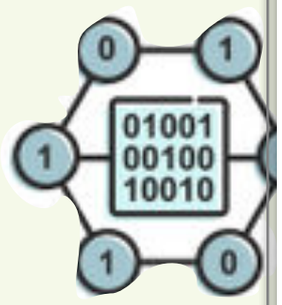


AUTONOMUS

ANALYZE

EXTRA INFO - this label will appear in slides with complementary information





SUMMARY: About Neuro-Fuzzy (continuation):

[T] Hard Clustering and Fuzzy clustering; Neuro-Fuzzy.

[P] Exploring scikit-fuzzy and fylearn.

EXTRA INFO - this label will appear in slides with complementary information

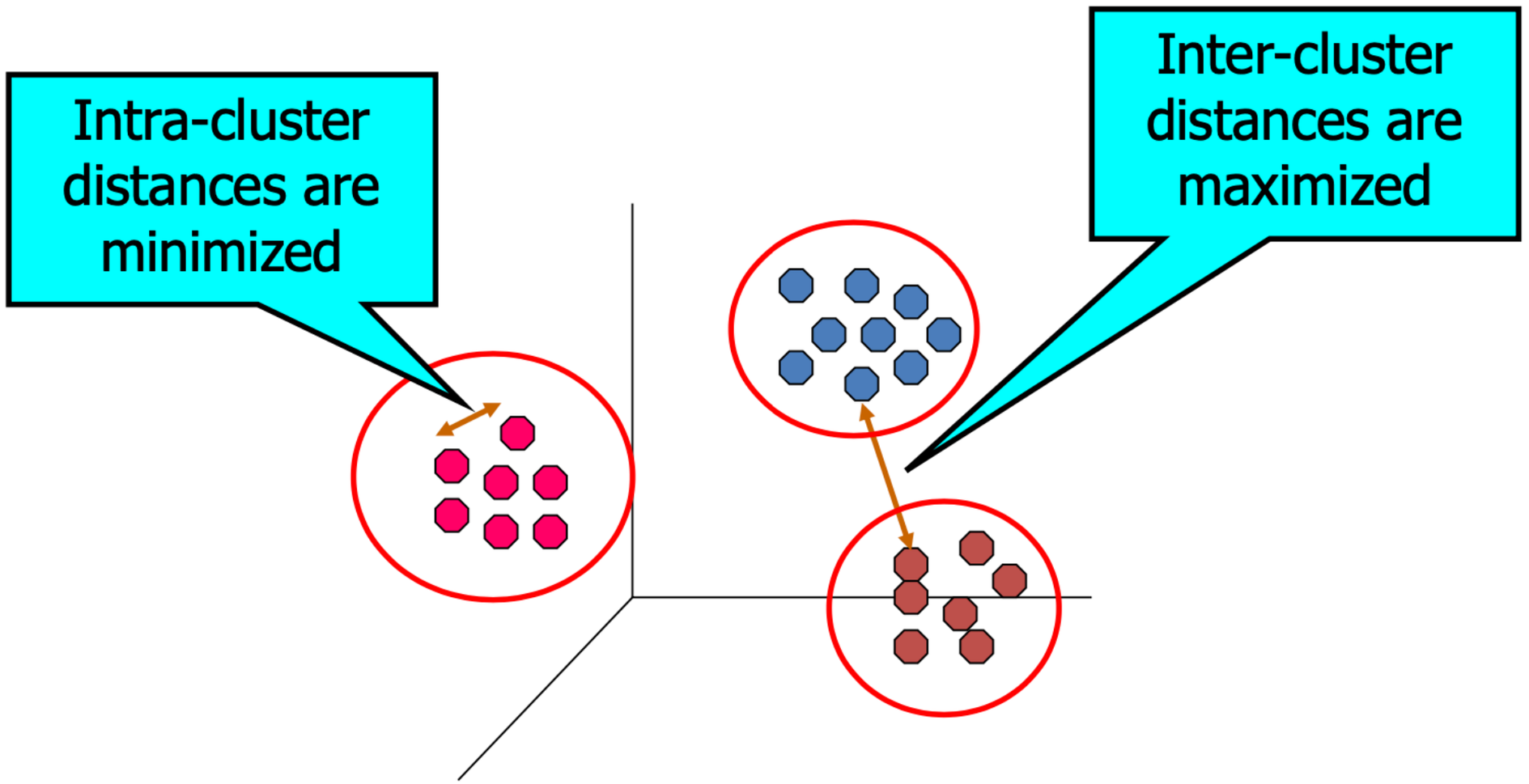
FIS optimization

- **Why to optimise?**
  - trial and error tuning is laborious
  - it can be impossibly complicated if number of input parameters are large (100's or more)
  - far too many parameters to tune: number of rules, membership functions, rule consequents
  - arbitrariness of FIS is eliminated
  - if not optimised, the FIS performance is not optimum

- Optimisation methods:
  - Adaptive Neuro-Fuzzy systems
    multilayer perceptron neural networks (ANFIS, FuNeI)
  - evolutionary techniques (genetic algorithms)
  - clustering methods (cluster analysis, Hard C-means, Fuzzy C-means)
  - ...

What is a Clustering?

- In general a grouping of objects such that the objects in a group (cluster) are similar (or related) to one another and different from (or unrelated to) the objects in other groups



Early applications of cluster analysis

- John Snow, London 1854

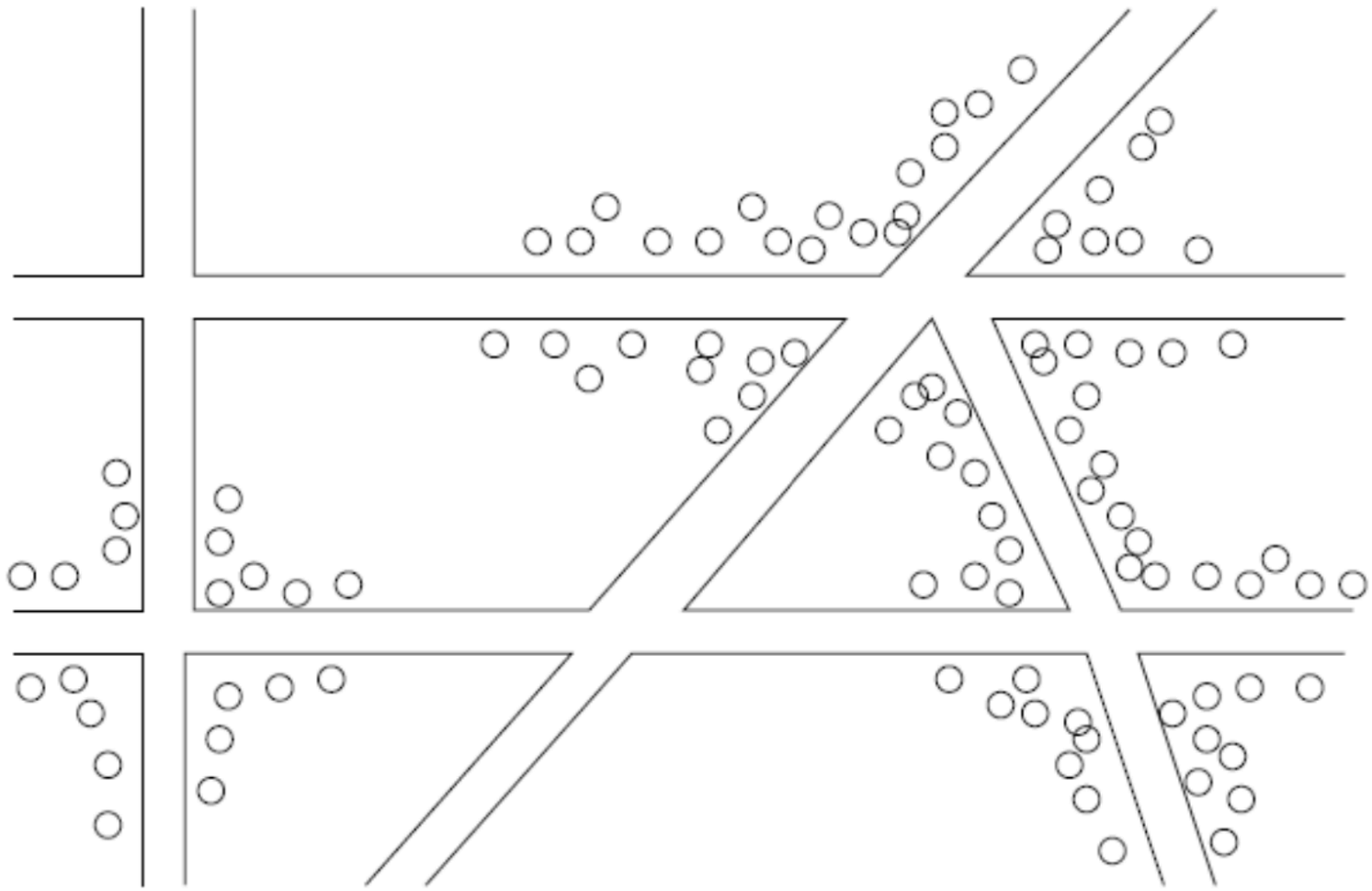
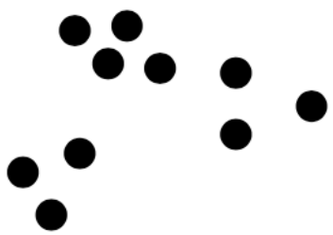
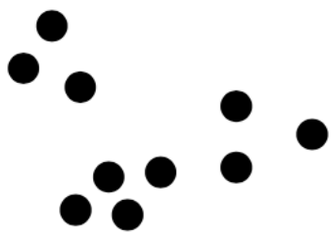


Figure 1.1: Plotting cholera cases on a map of London

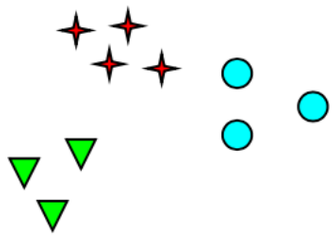
Notion of a Cluster can be Ambiguous



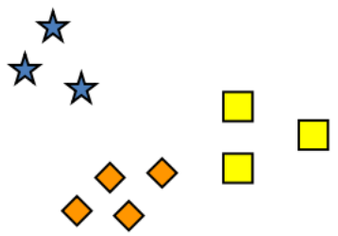
How many clusters?



Two Clusters



Six Clusters

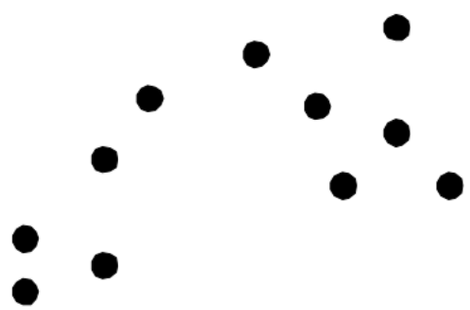


Four Clusters

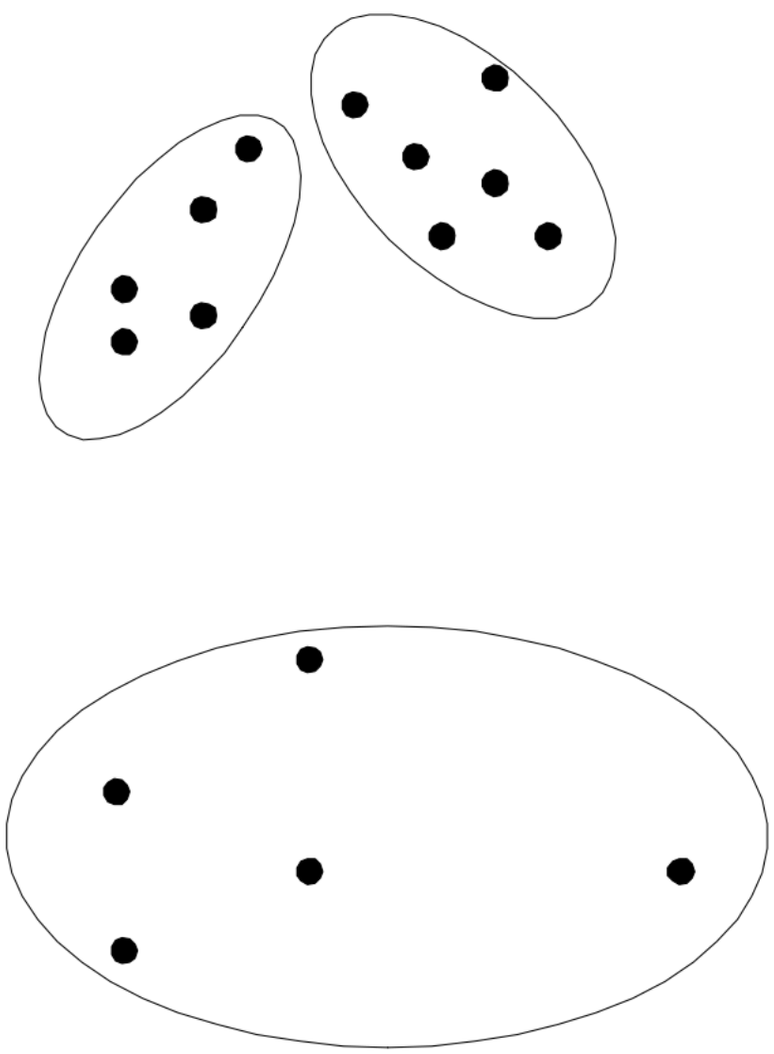
Types of Clusterings

- A clustering is a set of clusters
- Important distinction between hierarchical and partitional sets of clusters
  - Partitional Clustering
    - A division data objects into subsets (clusters) such that each data object is in exactly one subset
  - Hierarchical clustering
    - A set of nested clusters organized as a hierarchical tree

Partitional Clustering

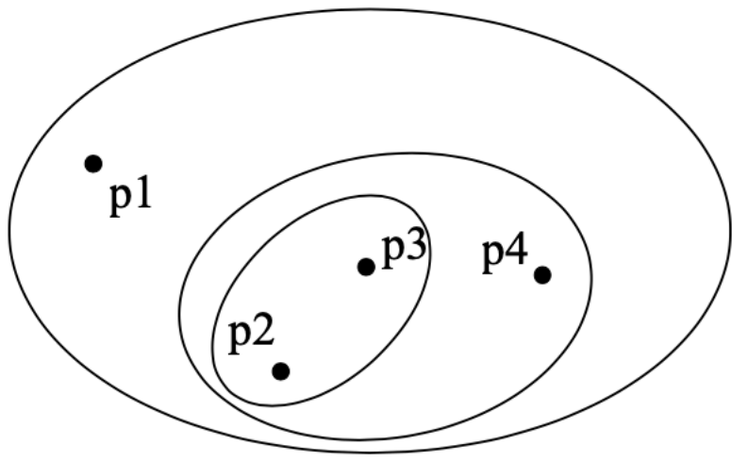


Original Points

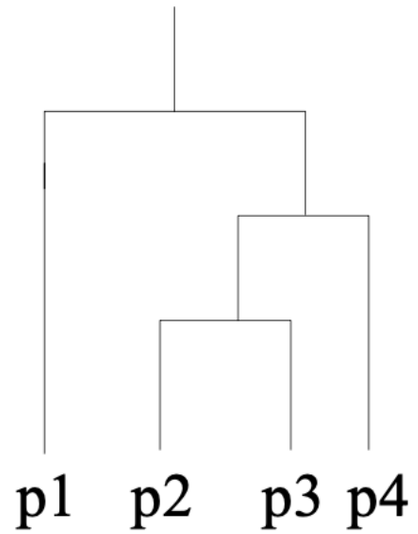


A Partitional Clustering

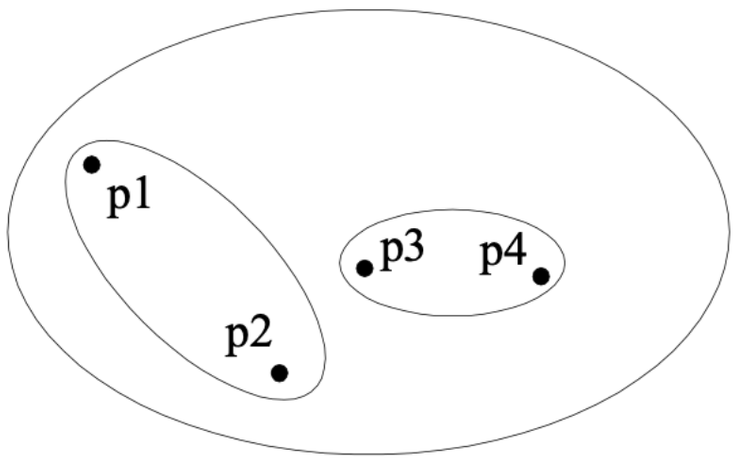
Hierarchical Clustering



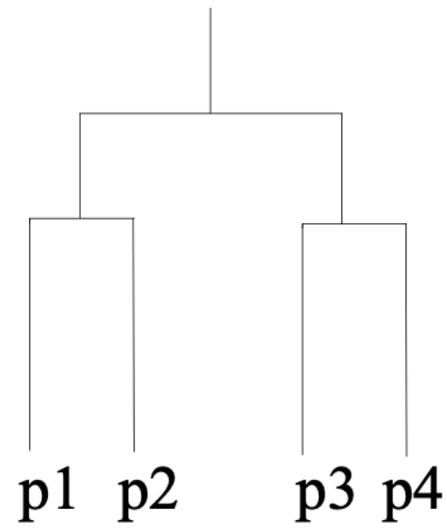
Traditional Hierarchical Clustering



Traditional Dendrogram



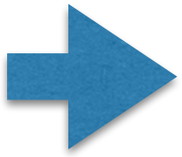
Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

Other types of clustering

- Exclusive (or non-overlapping) versus non-exclusive (or overlapping)
  - In non-exclusive clusterings, points may belong to multiple clusters.
    - Points that belong to multiple classes, or 'border' points



- Fuzzy (or soft) versus non-fuzzy (or hard)
  - In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1

- Partial versus complete
  - In some cases, we only want to cluster some of the data

Types of Clusters: Objective Function

- Clustering as an optimization problem
  - Finds clusters that minimize or maximize an objective function.
  - Enumerate all possible ways of dividing the points into clusters and evaluate the `goodness' of each potential set of clusters by using the given objective function. (NP Hard)
  - Can have global or local objectives.
    - Hierarchical clustering algorithms typically have local objectives
    - Partitional algorithms typically have global objectives
- A variation of the global objective function approach is to fit the data to a parameterized model.
  - The parameters for the model are determined from the data, and they determine the clustering
  - E.g., Mixture models assume that the data is a ‘mixture' of a number of statistical distributions.

K-means Clustering

- Partitional clustering approach
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K, must be specified
- The objective is to minimize the sum of distances of the points to their respective centroid

K-means Clustering

- **Problem:** Given a set X of n points in a d-dimensional space and an integer K group the points into K clusters $C= \{C_1, C_2,\ldots,C_k\}$ such that

$$Cost(C) = \sum_{i=1}^{k} \sum_{x \in C_i} dist(x, c)$$

is minimized, where c_i is the centroid of the points in cluster C_i

K-means Clustering

- Most common definition is with euclidean distance, minimizing the Sum of Squares Error (SSE) function
  - Sometimes K-means is defined like that

- **Problem:** Given a set X of n points in a d-dimensional space and an integer K group the points into K clusters $C= \{C_1, C_2,\ldots,C_k\}$ such that

$$Cost(C) = \sum_{i=1}^{k} \sum_{x \in C_i} (x - c_i)^2$$

is minimized, where c_i is the mean of the points in cluster C_i

Sum of Squares Error (SSE)

Complexity of the k-means problem

- NP-hard if the dimensionality of the data is at least 2 (**d>=2**)
  - Finding the best solution in polynomial time is infeasible

- For **d=1** the problem is solvable in polynomial time (how?)

- A simple iterative algorithm works quite well in practice

K-means Algorithm

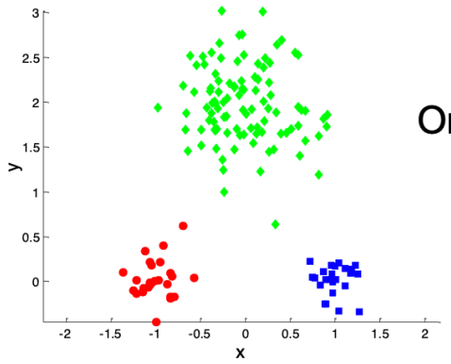
- Also known as Lloyd's algorithm.
- K-means is sometimes synonymous with this algorithm

1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

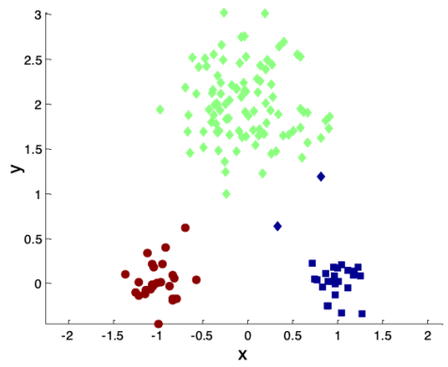
K-means Algorithm – Initialization

- Initial centroids are often chosen randomly.
  - Clusters produced vary from one run to another.

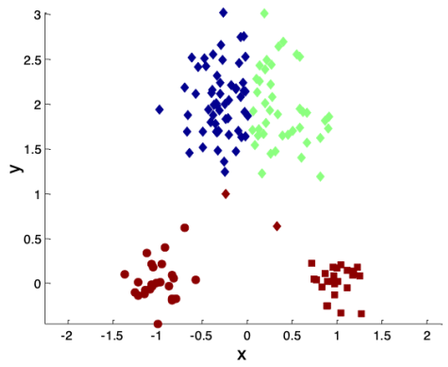
Two different K-means Clusterings



Original Points



Optimal Clustering



Sub-optimal Clustering

Dealing with Initialization

- Do multiple runs and select the clustering with the smallest error

- Select original set of points by methods other than random . E.g., pick the most distant (from each other) points as cluster centers (K-means++ algorithm)

K-means Algorithm – Centroids

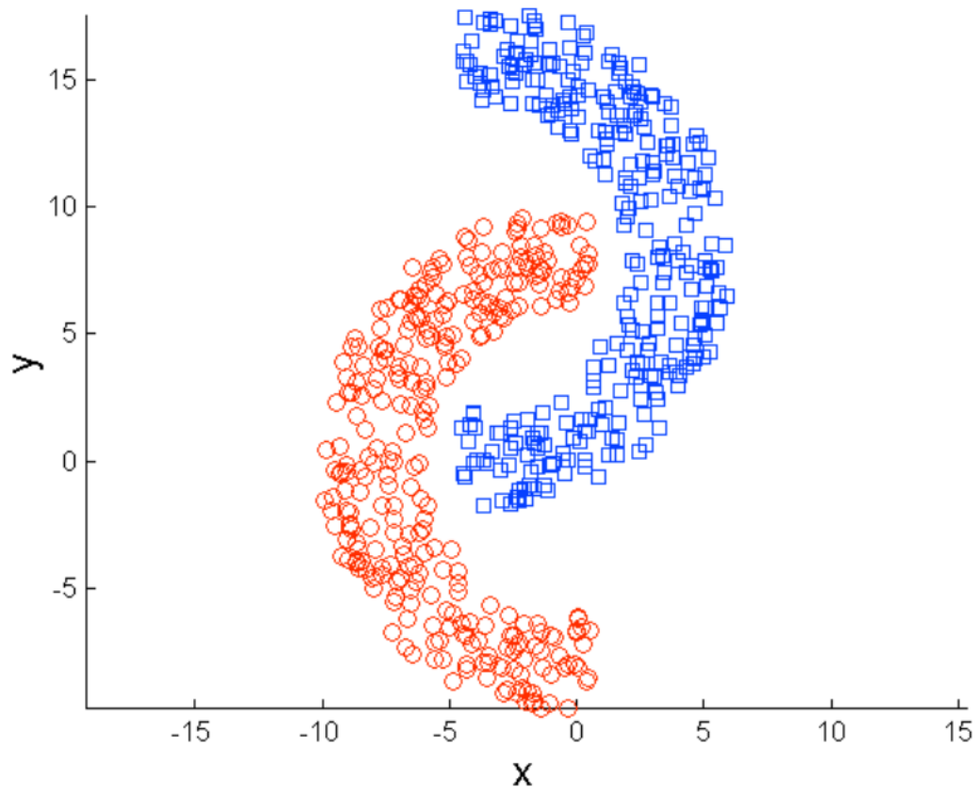
- The centroid depends on the distance function
  - The minimizer for the distance function
- 'Closeness' is measured by Euclidean distance (SSE), cosine similarity, correlation, etc.
- Centroid:
  - The mean of the points in the cluster for SSE, and cosine similarity
  - The median for Manhattan distance.

- Finding the centroid is not always easy
  - It can be an NP-hard problem for some distance functions
    - E.g., median form multiple dimensions

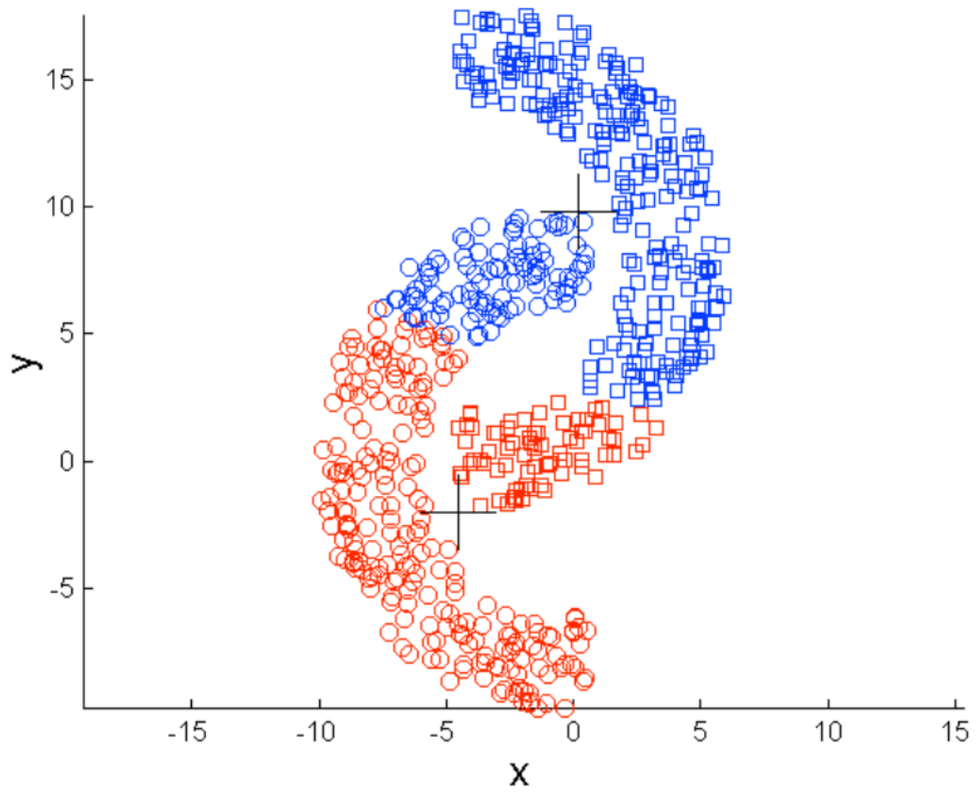
K-means Algorithm – Convergence

- K-means will converge for common similarity measures mentioned above.
  - Most of the convergence happens in the first few iterations.
  - Often the stopping condition is changed to ‘Until relatively few points change clusters’
- Complexity is O(n * K * I * d)
  - n = number of points, K = number of clusters, I = number of iterations, d = dimensionality
- In general a fast and efficient algorithm

Limitations of K-means: Non-globular Shapes

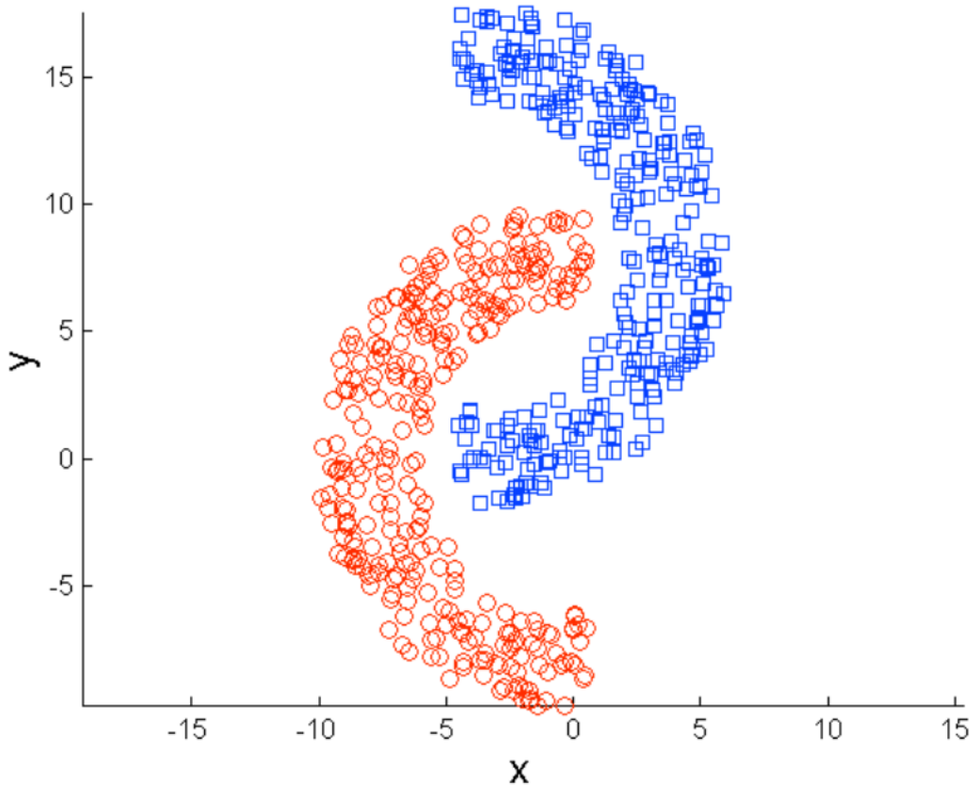


Original Points

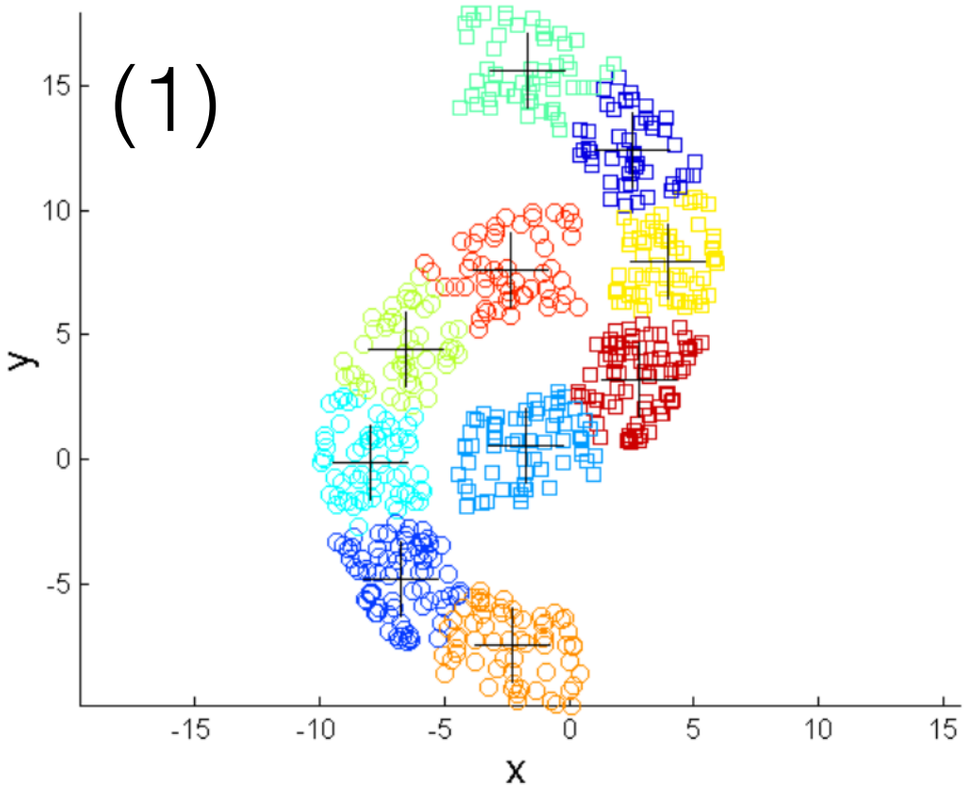


K-means (2 Clusters)

Overcoming K-means Limitations

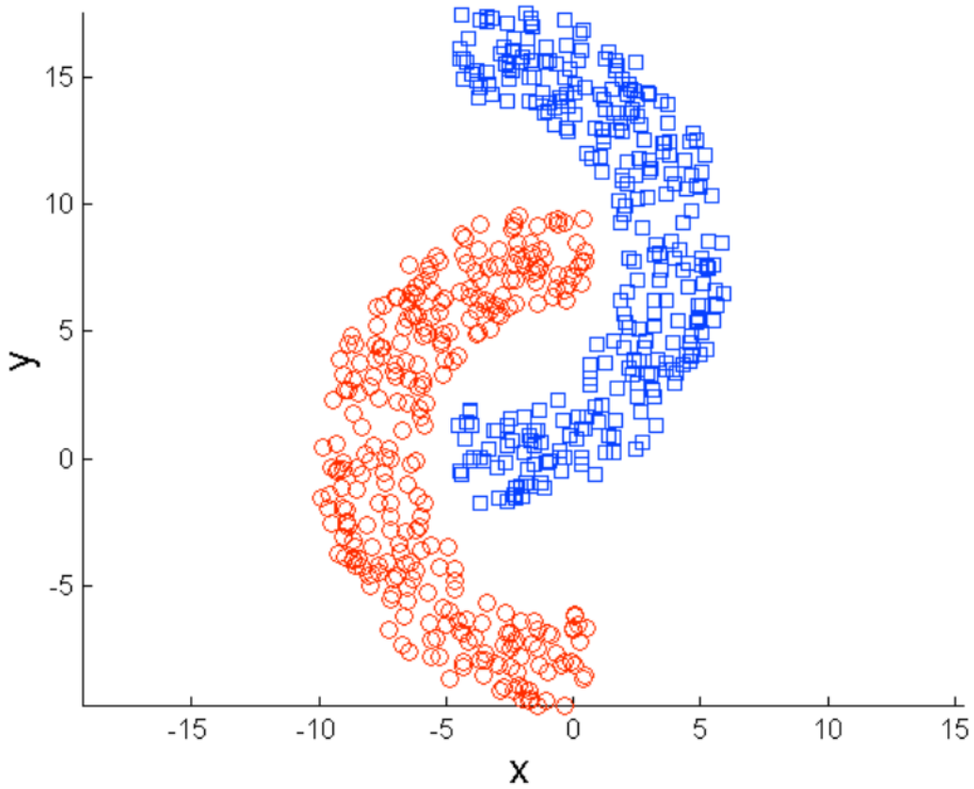


Original Points



K-means Clusters

Overcoming K-means Limitations



Original Points

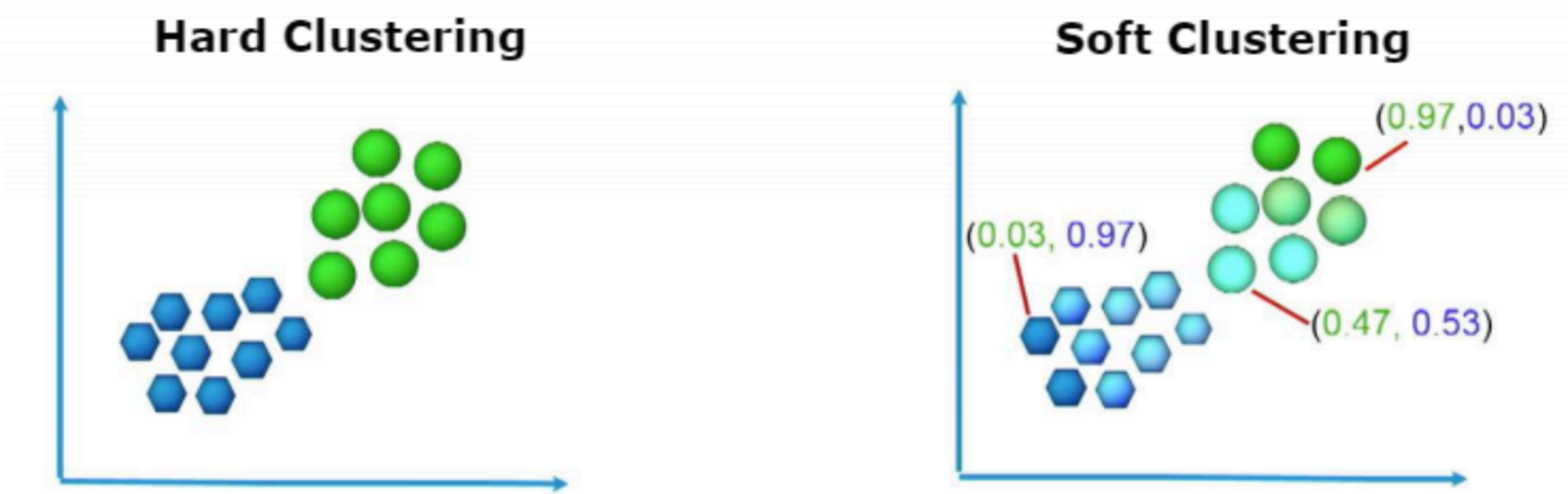
(2) Changing the distance/similarity

e.g. Mahalanobis distance

$$D_M(\vec{x}) = \sqrt{(\vec{x} - \vec{\mu})^\mathsf{T} \mathbf{S}^{-1} (\vec{x} - \vec{\mu})}.$$

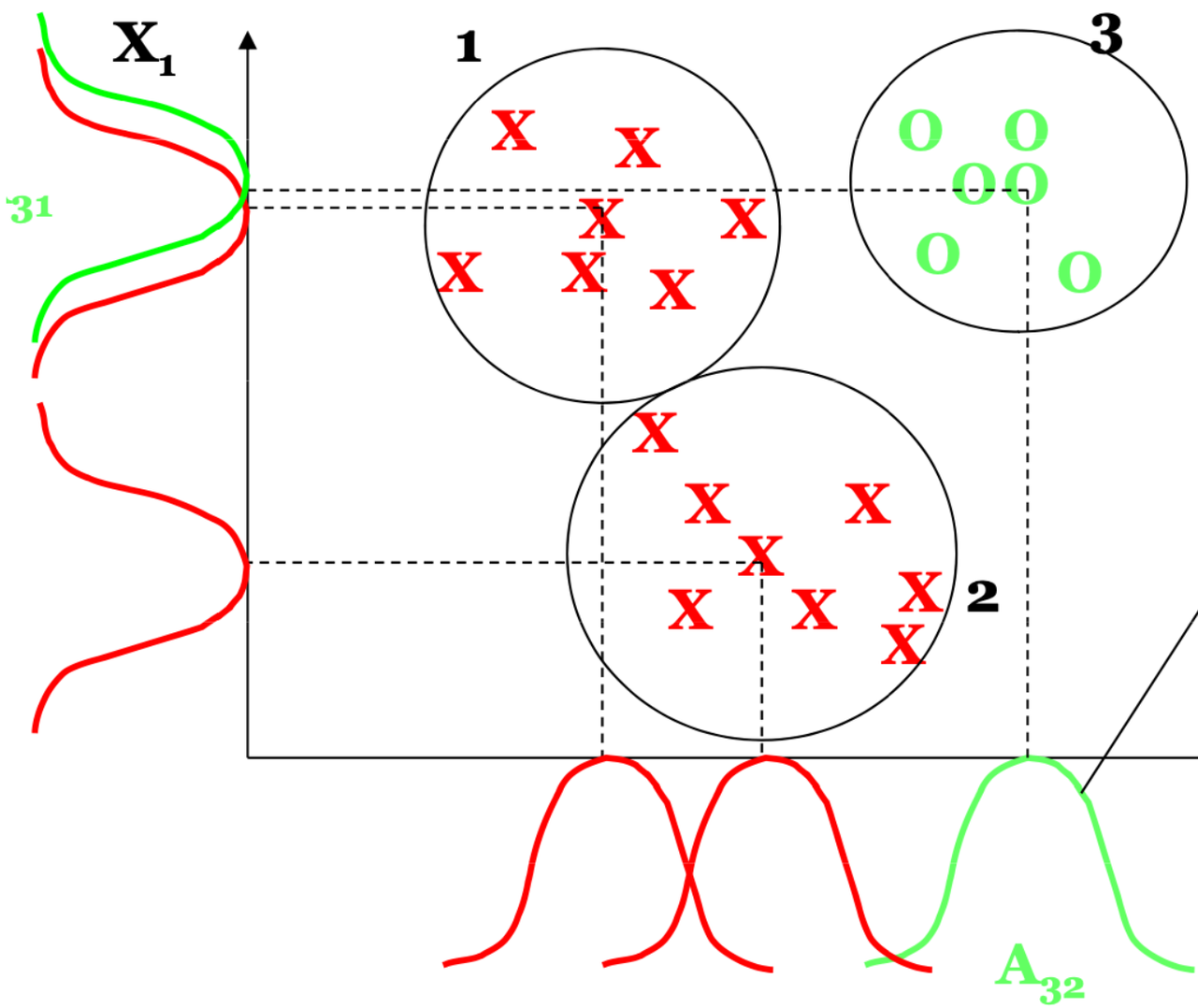
$$\operatorname{cov}[X_i, X_j] = \operatorname{E}[(X_i - \operatorname{E}[X_i])(X_j - \operatorname{E}[X_j])]$$

Fuzzy Clustering (idea)



Fuzzy Clustering (idea)

- fuzzy rules generation

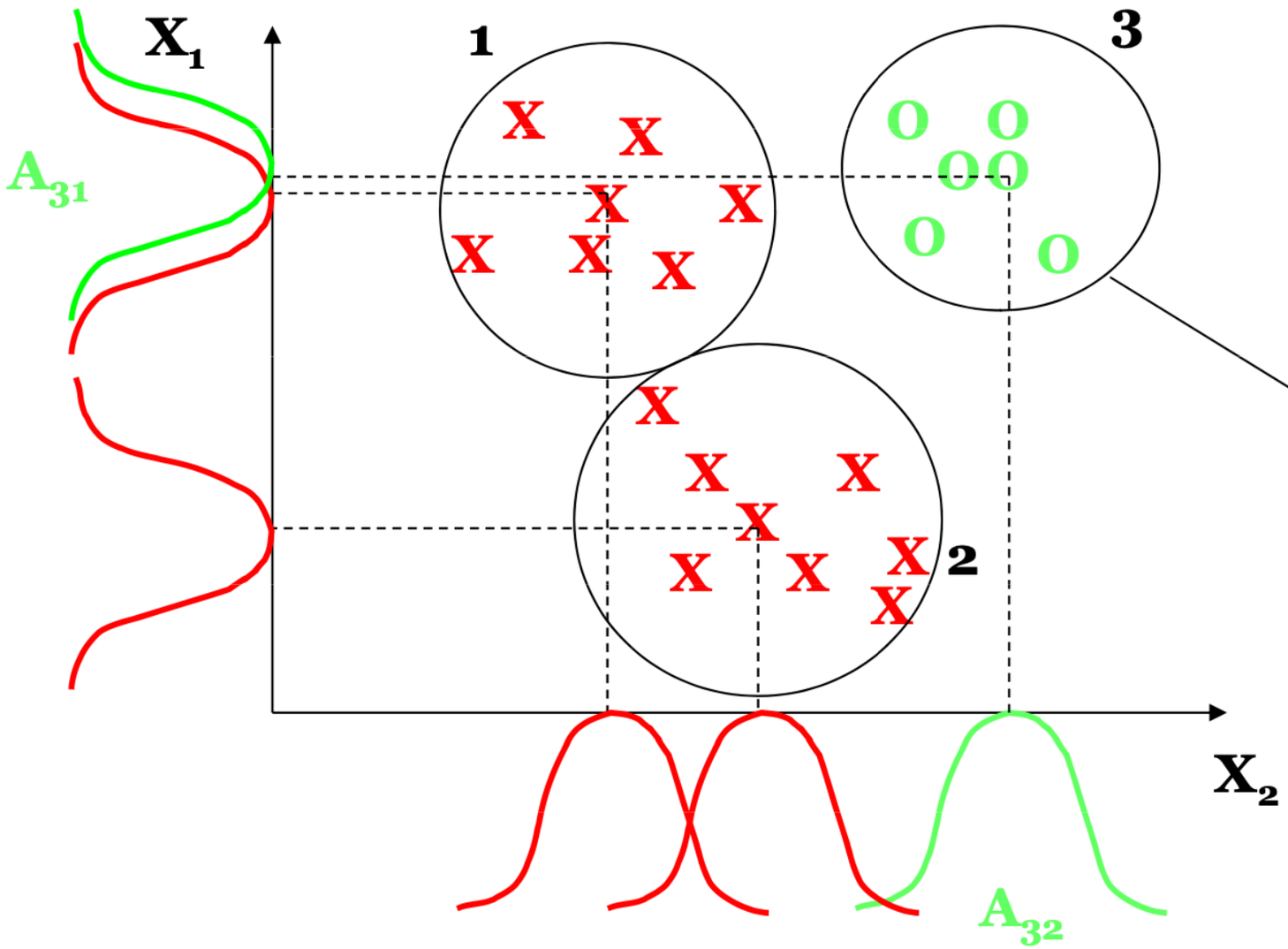


Gaussian fuzzy membership function

$$A_{32}(X_2) = e^{-\frac{1}{2}\left(\frac{X_2 - x_{32}}{\sigma_{32}}\right)^2}$$

Fuzzy Clustering (idea)

- fuzzy rules generation



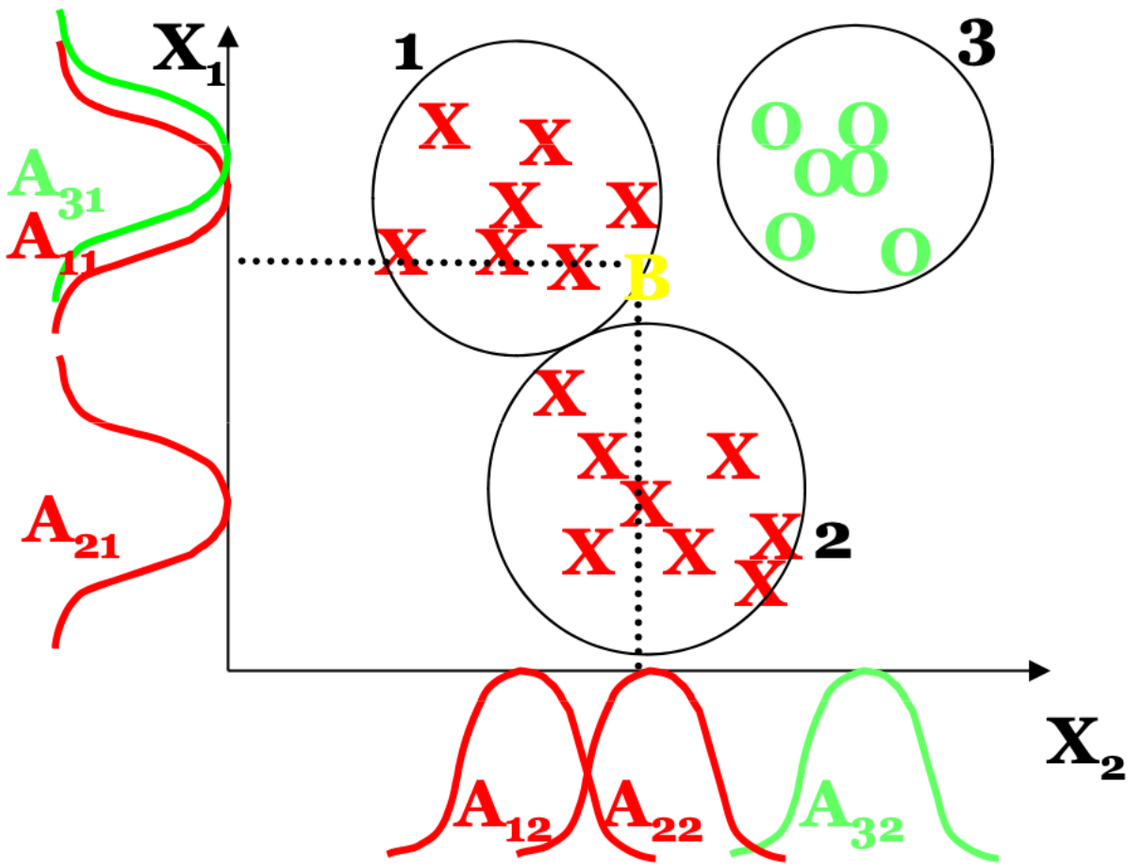
If X_1 is A_{31} and X_2 is A_{32}

Then

Class is O

Fuzzy Clustering (idea)

- Classification of an unseen vector B



If X_1 is A_{11} and X_2 is A_{12} Then Class is X
$A_{11}(X_1) * A_{12}(X_2) = 0.8 * 0.2 = 0.16$

If X_1 is A_{21} and X_2 is A_{22} Then Class is X
$A_{21}(X_1) * A_{22}(X_2) = 0.01 * 0.9 = 0.009$

If X_1 is A_{31} and X_2 is A_{32} Then Class is O
$A_{31}(X_1) * A_{32}(X_2) = 0.1 * 0.01 = 0.001$

B belongs to X

FCM

$$\min \; J_{\mathrm{FCM}}(U,V) = \sum_{j=1}^{N}\sum_{i=1}^{C}(u_{ij})^{q}(d_{ji})^{2} \qquad \text{with}$$

$(d_{ji})^2 = \|x_j - v_i\|^2$

u_{ij} = Membership level x_j in cluster to i;
N = Total data;
C = Total cluster;
q = Fuzzifier parameter, $q > 1$.

Step 1. Initialization Vector centroid, v_i (prototypes).

Step 2. Calculate the distance between feature vector (X) and the centroid vector $(V)[X \rightarrow V]$. Feature vectors with the closest distance to one of the centroid vectors then expressed as a cluster member.

Step 3. Calculate membership level of all feature vectors in all clusters by using the formula:

$$u_{ij} = \frac{1}{\sum_{k=1}^{K}\left[\frac{(d_{ji})^2}{(d_{jk})^2}\right]^{1/(q-1)}} = \frac{\left[\frac{1}{(d_{ji})^2}\right]^{1/(q-1)}}{\sum_{k=1}^{K}\left[\frac{1}{(d_{jk})^2}\right]^{1/(q-1)}}. \qquad (3)$$

FCM

$$\min \; J_{\mathrm{FCM}}(U,V) = \sum_{j=1}^{N}\sum_{i=1}^{C}(u_{ij})^{q}(d_{ji})^{2} \qquad \text{with}$$

$(d_{ji})^2 = \|x_j - v_i\|^2$

u_{ij} = Membership level x_j in cluster to i;
N = Total data;
C = Total cluster;
q = Fuzzifier parameter, $q > 1$.

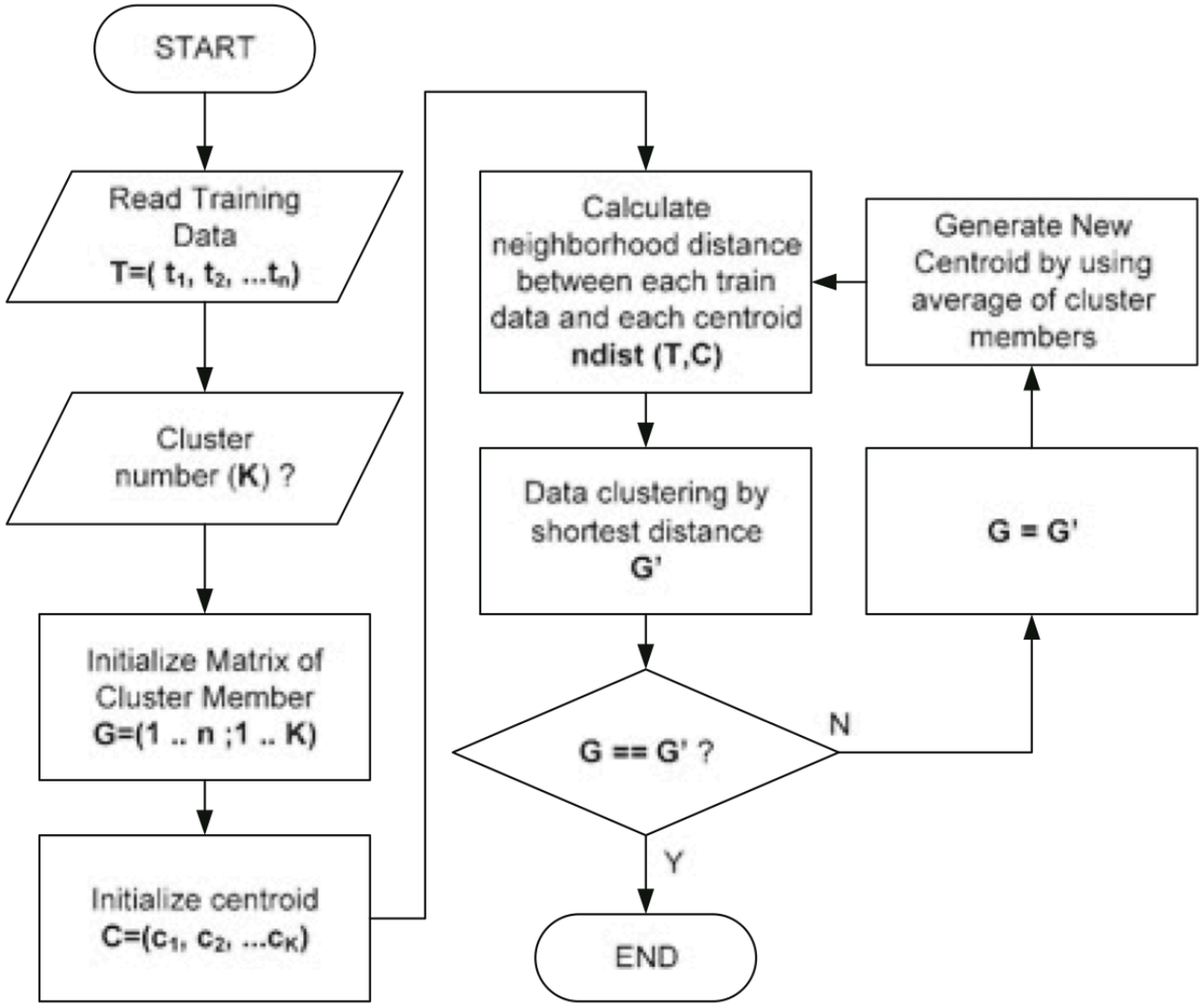
Step 4. Calculate new centroid using Eq. (4).

$$\hat{V}_i = \frac{\sum_{j=1}^{N}(u_{ij})^{q}X_j}{\sum_{j=1}^{N}(u_{ij})^{q}}. \tag{4}$$

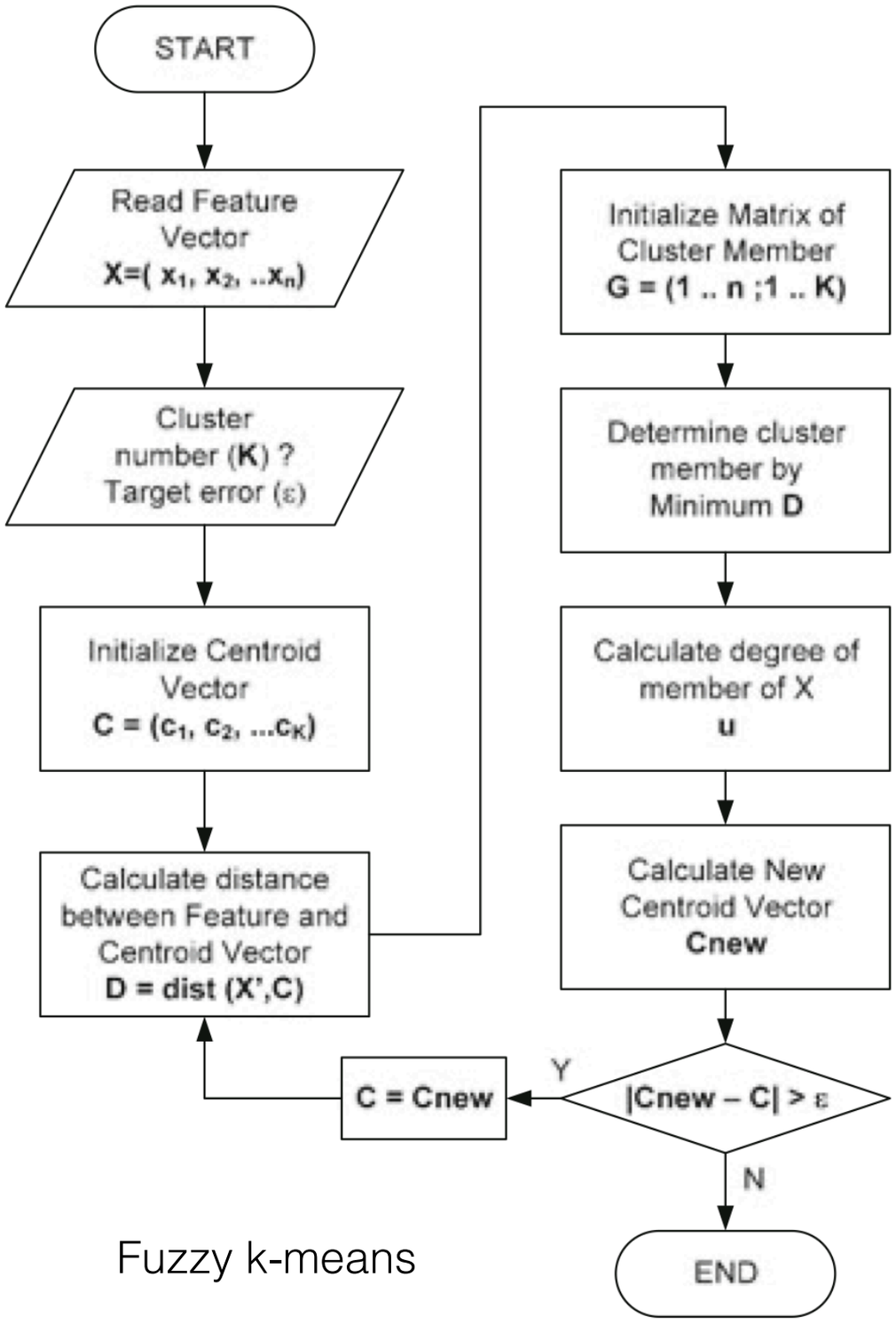
Step 5. Recalculate the step 4, $u_{ij} \rightarrow \hat{u}_{ij}$ If, $\max_{ij}|u_{ij} - \hat{u}_{ij}| < \varepsilon$, where ε is termination criteria between 0 and 1. Then, the iteration process is stopped. If not go back to step 5.

U = Fuzzy datasets K-partition;
V = Set of prototype centroid;
$V = \{v_1, v_2, \cdots, v_C\} \subset R^P$;

Fuzzy Clustering (idea)



Classical k-means



Fuzzy k-means

Fuzzy Clustering (idea)

Evaluation of features — For the partitioning process, both versions use the same measures, entropy and information gain, in order to select the features to be used in the test nodes of the tree;

Induction process — Both versions use the same approach: repeated subdivision of the feature space using the most informative features until a leaf node is reached or no features or examples remain;

Some benefits

- Rule set reduction:

1. **IF** *Compactness* is ≤ 95 **AND** ... **AND** *Compactness* is ≤ 89
2. **IF** *Compactness* is ≤ 95 **AND** ... **AND** *Compactness* is > 89
3. **IF** *Compactness* is > 95
4. **IF** *Compactness* is ≤ 102
5. **IF** *Compactness* is > 102
6. **IF** *Compactness* is ≤ 109 **AND** ... **AND** *Compactness* is ≤ 106
7. **IF** *Compactness* is ≤ 109 **AND** ... **AND** *Compactness* is > 106
8. **IF** *Compactness* is > 109
9. **IF** *Compactness* is ≤ 82 **AND** ... **AND** *Compactness* is ≤ 81
10. **IF** *Compactness* is ≤ 82 **AND** ... **AND** *Compactness* is > 81
11. **IF** *Compactness* is > 82 **AND** ... **AND** *Compactness* is ≤ 84
12. **IF** *Compactness* is > 82 **AND** ... **AND** *Compactness* is > 84

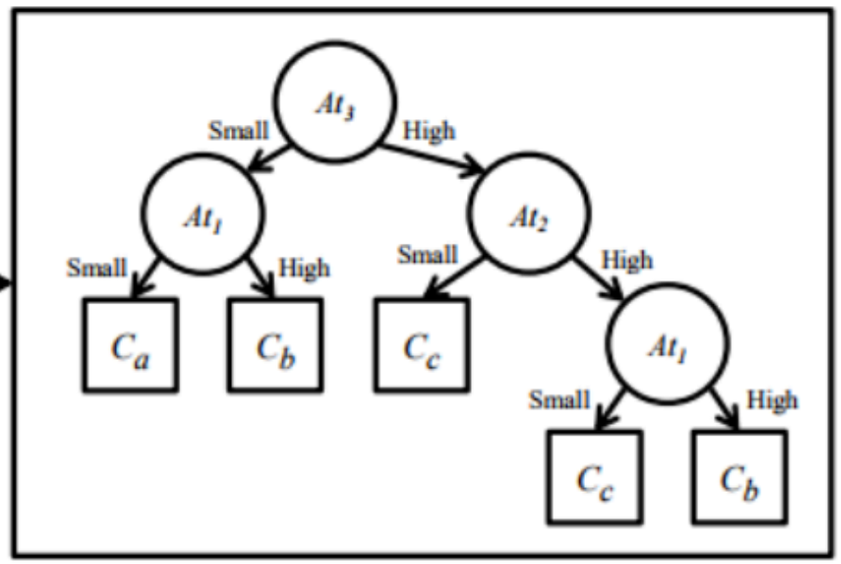
Fuzzy Machine Learning

#	At_1	At_2	At_3	Class
1	15	0.6	3.1	C_a
2	17	0.9	2.2	C_b
3	19	0.4	2.3	C_c
		...		
n	16	0.8	2.9	C_a

Original dataset

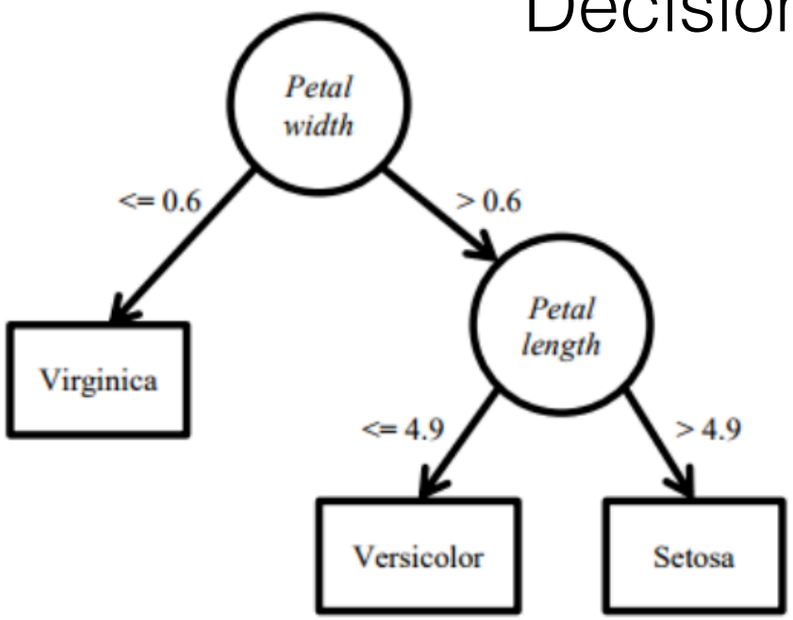
#	At_1	At_2	At_3	Class
1	Small	Low	High	C_a
2	High	High	Low	C_b
3	High	Low	Low	C_c
	...	...	...	...
n	Low	High	High	C_a

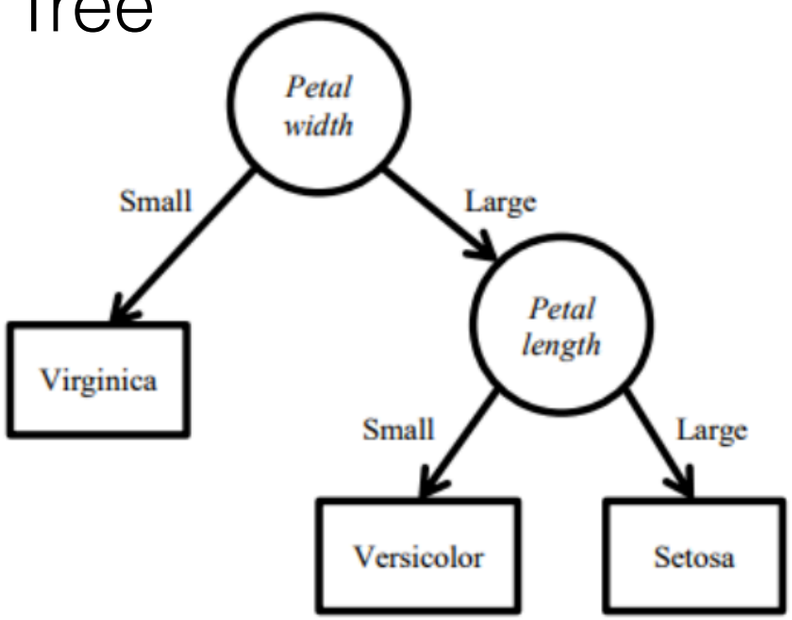
Fuzzyfied dataset



Induced Tree

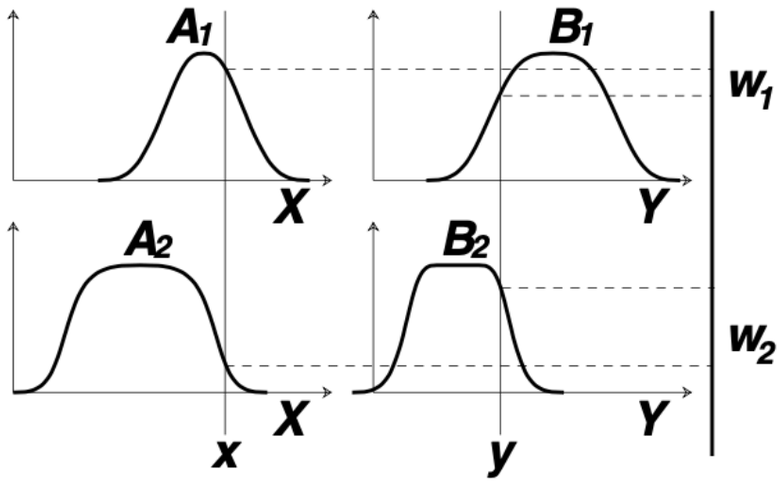
Decision Tree





Neuro-Fuzzy

Fuzzy Model



$$f_1 = p_1 x + q_1 y + r_1$$

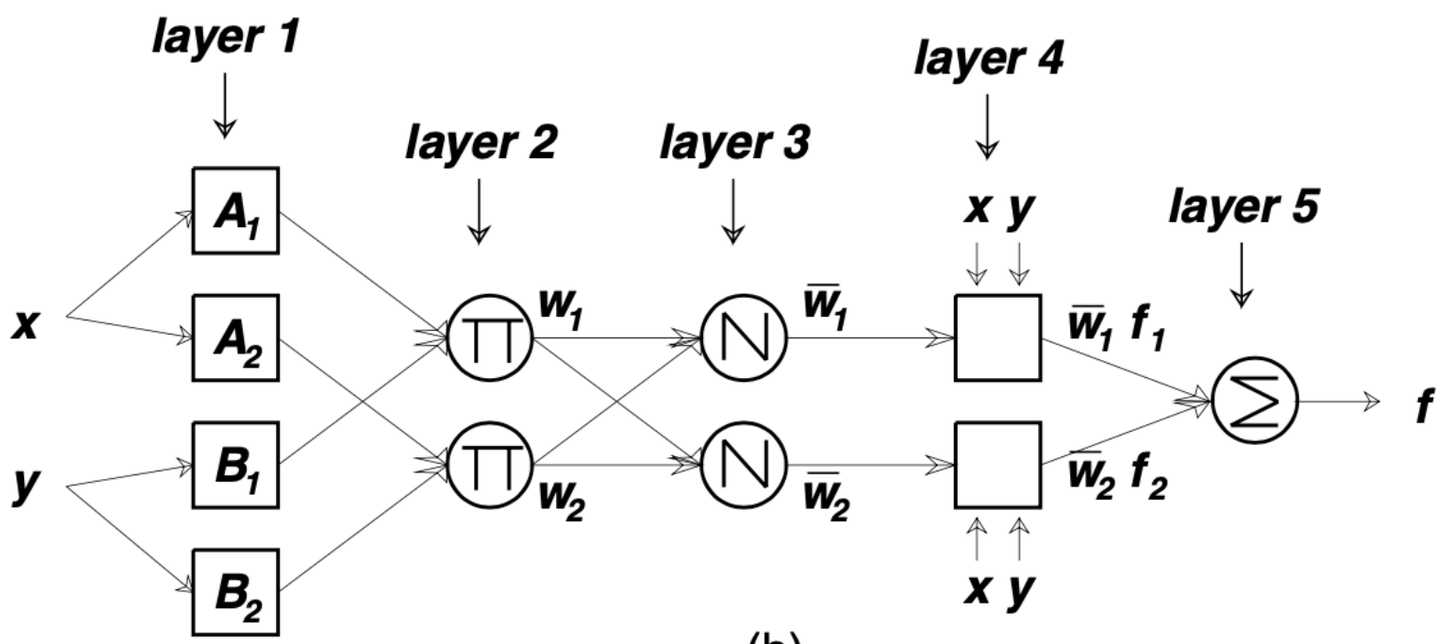
$$f_2 = p_2 x + q_2 y + r_2$$

$$f = \frac{w_1 f_1 + w_2 f_2}{w_1 + w_2}$$

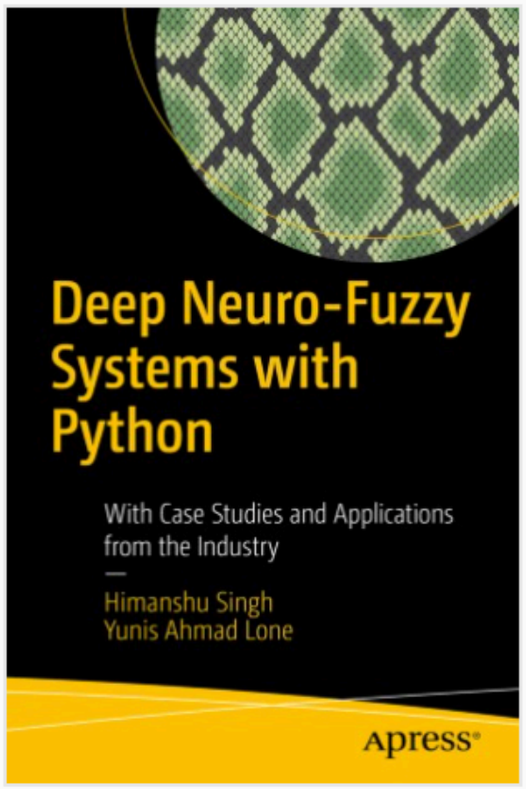
$$= \bar{w}_1 f_1 + \bar{w}_2 f_2$$

(a)

Neuro Network



(b)



Deep Neuro-Fuzzy Systems with Python

With Case Studies and Applications from the Industry

Authors (view affiliations)

Himanshu Singh, Yunis Ahmad Lone

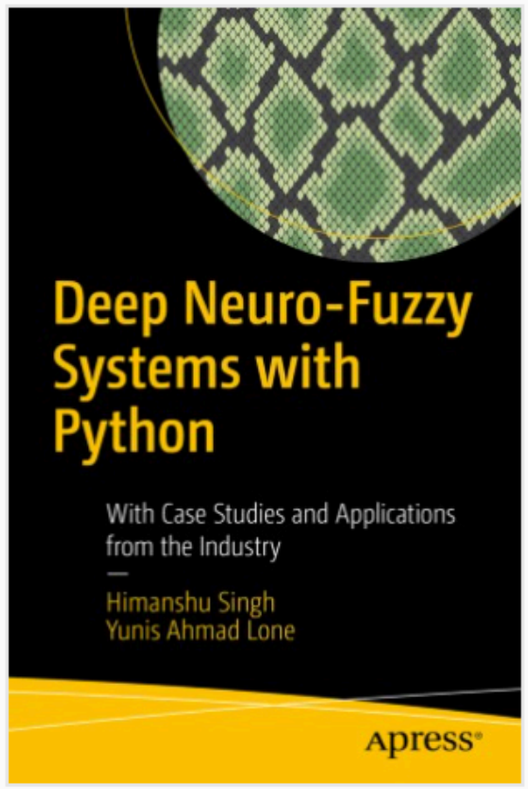
Explains deep neuro-fuzzy systems with applications and mathematical details

Implementations of all the applications using Python

Covers the recent applications of neuro fuzzy inference systems in industry

Apress®

https://github.com/Apress/deep-neuro-fuzzy-systems-w-python



Apress®

Deep Neuro-Fuzzy Systems with Python

Introduction to Fuzzy Set Theory
Himanshu Singh, Yunis Ahmad Lone
Pages 1-34

Fuzzy Rules and Reasoning
Himanshu Singh, Yunis Ahmad Lone
Pages 35-92

Fuzzy Inference Systems
Himanshu Singh, Yunis Ahmad Lone
Pages 93-127

Introduction to Machine Learning
Himanshu Singh, Yunis Ahmad Lone
Pages 129-156

Artificial Neural Networks
Himanshu Singh, Yunis Ahmad Lone
Pages 157-198

Fuzzy Neural Networks
Himanshu Singh, Yunis Ahmad Lone
Pages 199-221

Advanced Fuzzy Networks
Himanshu Singh, Yunis Ahmad Lone
Pages 223-251

Back Matter
Pages 253-260

https://github.com/Apress/deep-neuro-fuzzy-systems-w-python

https://github.com/twmeggs/anfis

Pratical Tasks

1. Implement a “standard” sklearn kmeans
2. Explore FuzzyLogic04.ipynb
3. Create a new notebook FuzzyLogic05.ipynb and implement a Neuro-Fuzzy system.