Clustering
(Unsupervised Learning)

- Cluster: a collection of data objects
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters
- Cluster analysis
  - Grouping a set of data objects into clusters
- Clustering is unsupervised: no predefined classes
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

General Applications of Clustering

- Pattern Recognition
- Spatial (Geographic) Data Analysis
- Image Processing
- Market research
- Document classification
- Cluster Web-log data to discover patterns

What Is Good Clustering?

- A good clustering method will produce high quality clusters with:
  - high intra-class similarity
  - low inter-class similarity
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation.
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns.

Requirements of Clustering

- Discovery of clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise/ outliers
- Insensitive to the order of input records
- High dimensionality
- Incorporation of user-specified constraints
- Interpretability and usability

Similarity and Dissimilarity Between Objects

- Distances are normally used to measure the similarity or dissimilarity between two data objects
- Some popular ones include: *Minkowski distance*:

$$d(i,j) = \sqrt[q]{(|x_{i_1} - x_{j_1}|^q + |x_{i_2} - x_{j_2}|^q + ... + |x_{i_p} - x_{j_p}|^q)}$$

where *i* = (x_{i1}, x_{i2}, …, x_{ip}) and *j* = (x_{j1}, x_{j2}, …, x_{jp}) are two *p*-dimensional data objects, and *q* is a positive integer

- If *q = 1*, *d* is Manhattan distance/ City-block distance/ snake distance:

$$d(i,j) = |x_{i_1} - x_{j_1}| + |x_{i_2} - x_{j_2}| + ... + |x_{i_p} - x_{j_p}|$$

- *If q = 2, d* is Euclidean distance:

$$d(i,j)=\sqrt{(|x_{i_1}-x_{j_1}|^2+|x_{i_2}-x_{j_2}|^2+...+|x_{i_p}-x_{j_p}|^2)}$$

- Properties
  - *d(i,j)* ≥ 0; Distance is a non negative number
  - *d(i,i)* = 0; Distance of an object to itself is 0.
  - *d(i,j) = d(j,i); Distance is symmetric function.*

Major Clustering Approaches

Major Clustering Approaches

- Partitioning algorithms: Construct various partitions and then evaluate them by some criterion
- Hierarchy algorithms: Create a hierarchical decomposition of the set of data (or objects) using some criterion
- Density-based: based on connectivity and density functions
- Grid-based: based on a multiple-level granularity structure
- Model-based: A model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other

Partitioning Methods

In this, ‘n’ data points are partitioned into ‘k’ clusters where k<n.

There are two basic conditions for this:

1. One data point should only belongs to one cluster (we use iterative relocation technique for this where a data point is removed from one cluster and added to new cluster).
2. There should not be any cluster having no data point

Partitioning Algorithms: Basic Concept

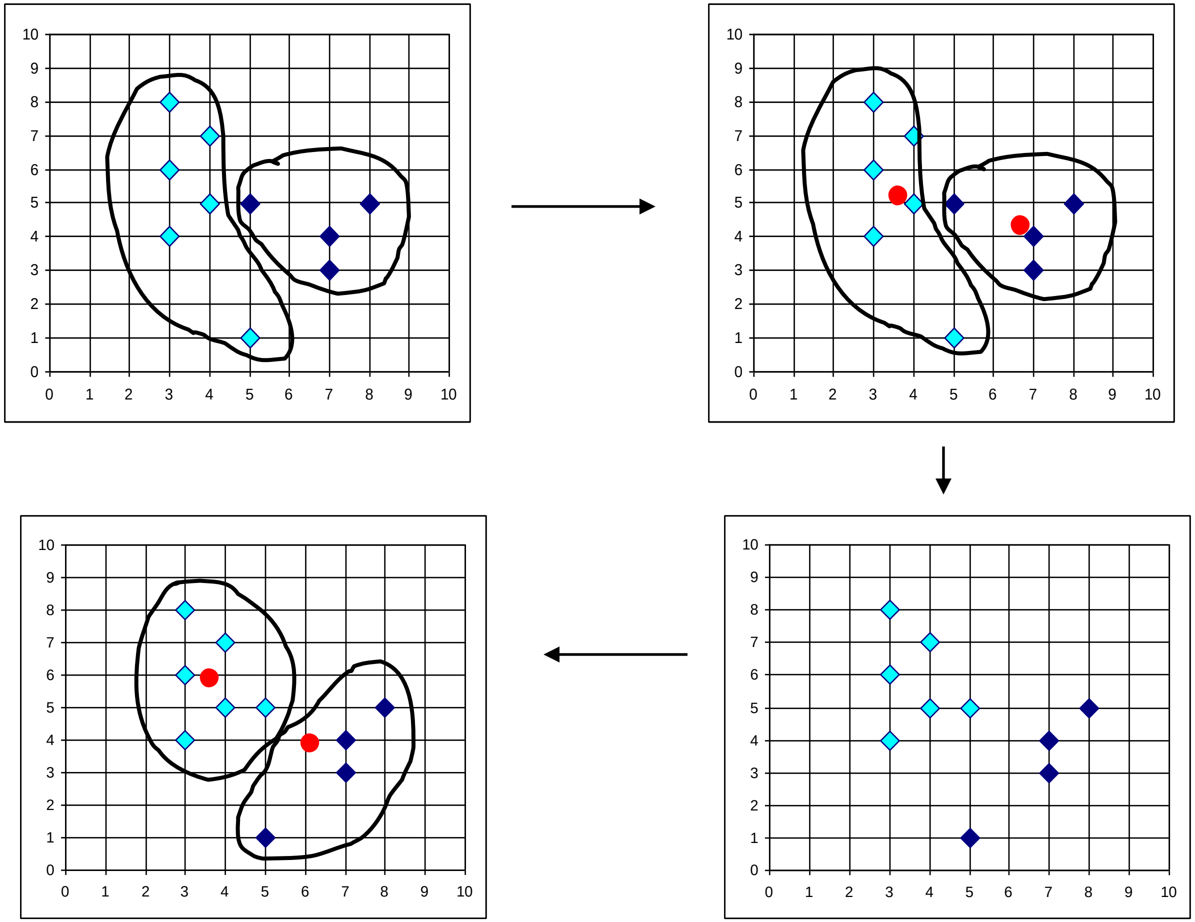
- Partitioning method: Construct a partition of ***n*** objects into a set of ***k*** clusters
- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means* and *k-medoids* algorithms
  - *k-means* (MacQueen'67): Each cluster is represented by the center of the cluster
  - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster

The *K-Means* Clustering Method

- Given *k*, the *k-means* algorithm is implemented in 4 steps:
  - Partition objects into *k* nonempty subsets
  - Compute seed points as the centroids of the clusters of the current partition. The centroid is the center (mean point) of the cluster.
  - Assign each object to the cluster with the nearest seed point.
  - Go back to Step 2, stop when no more new assignment.

The *K-Means* Clustering Method

- Example



K-Means Clustering: Example 1

- Given data points are x1(8,5), x2(4,1), x3(3,7), x4(2,1), x5(7,2) and x6(4,2)
- Number of clusters: 2

- Final Solution: Cluster 1(x1,x3,x5) and Cluster 2(x2,x4,x6)

K-Means Clustering: Example 2

- Given data points are A(2,10), B(2,5), C(8,4), D(5,8), E(7,5), F(6,4), G(1,2) and H(4,9)

- Number of clusters: 3

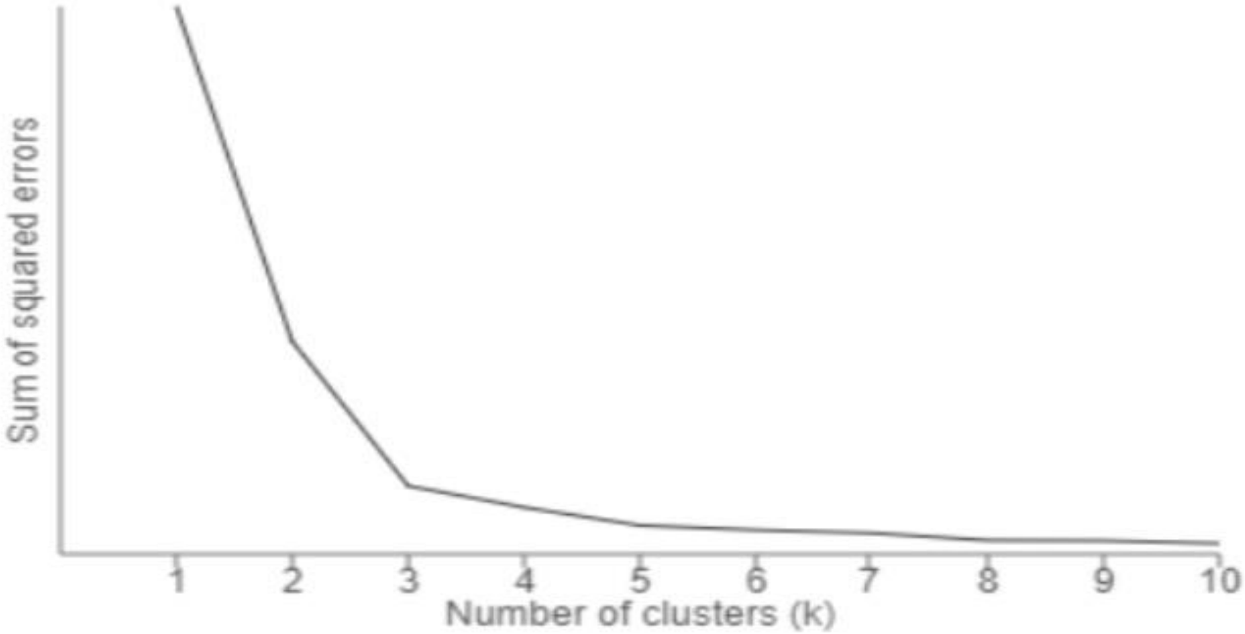
- Final Solution: ??

Choosing the value of K in K-Means Clustering

- One method of choosing value K is *the elbow method*

- *r*un K-Means clustering for a range of K values and find Sum of Squared Error (SSE). SSE is calculated as the mean distance between data points and their cluster centroid.

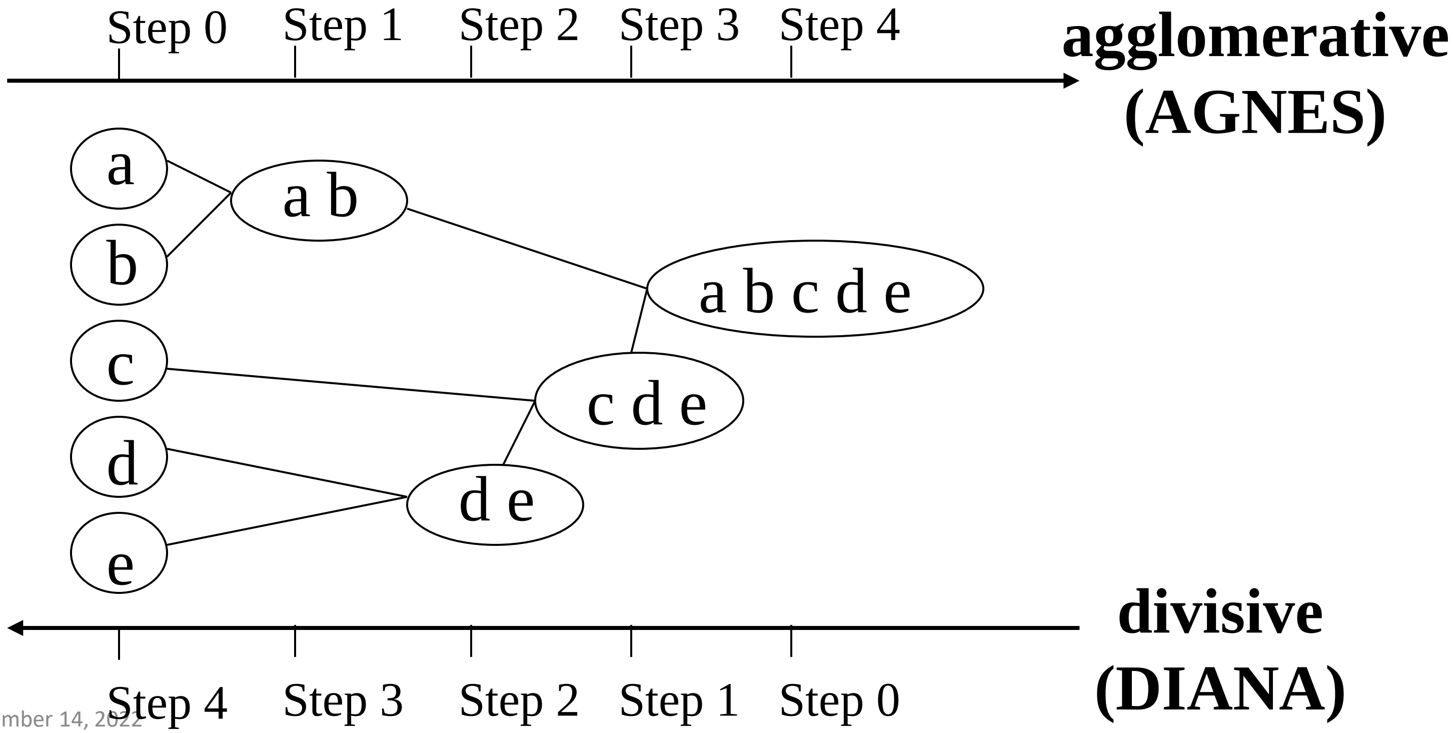
- Plot a line chart between SSE values and K. Elbow on the arm is the value of K which is to be selected.



Hierarchical Clustering

Hierarchical Clustering

- Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition



Types of Agglomerative Clustering

- ***Single Link Technique:*** Where the proximity of two clusters is identified using the minimum of the distance between the points belonging to two different clusters.

- ***Complete Link Technique:*** Where the proximity of two clusters is identified using the maximum of the distance between the points belonging to two different clusters.

- ***Average Link Technique:*** Where the proximity of two clusters is identified using the average of the distance between the points belonging to two different clusters.

Numerical example: Single Link Technique

- Illustrate single link technique for clustering using the following dataset. Use Euclidean distance:

a(2,7), b(2,5), c(3,6), d(8,5), e(7,4), f(8,3)

Results:

Level 1: ‘a’ and ‘c’ will be merged to form (a,c) [Value: 1.41]

Level 2: ‘d’ and ‘e’ will be merged to form (d,e) [Value: 1.41]

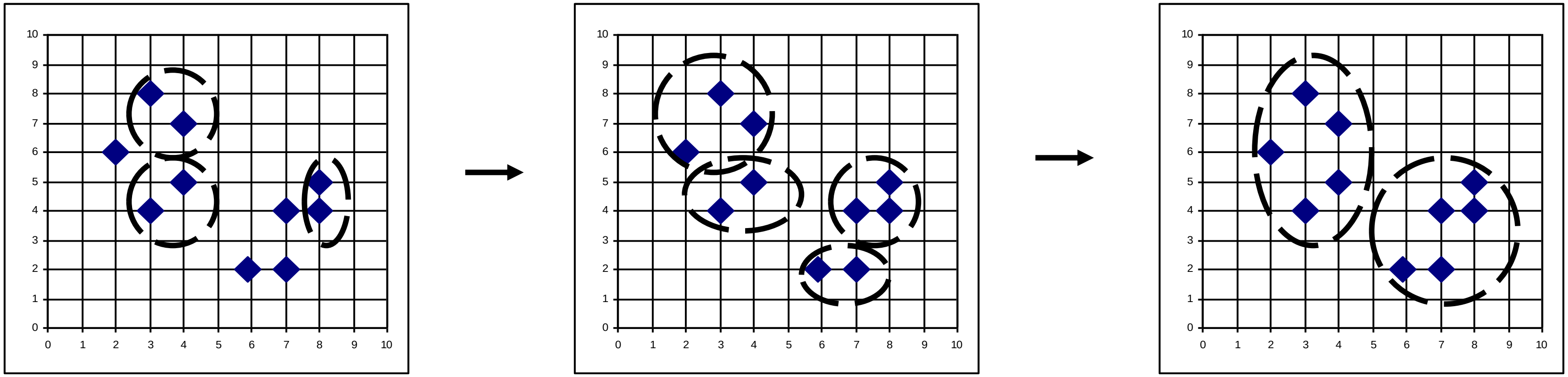
Level 3: (d,e) and ‘f’ will be merged to form ((d,e),f) [Value: 1.41]

Level 4: (a,c) and ‘b’ will be merged to form ((a,c),b) [Value: 1.41]

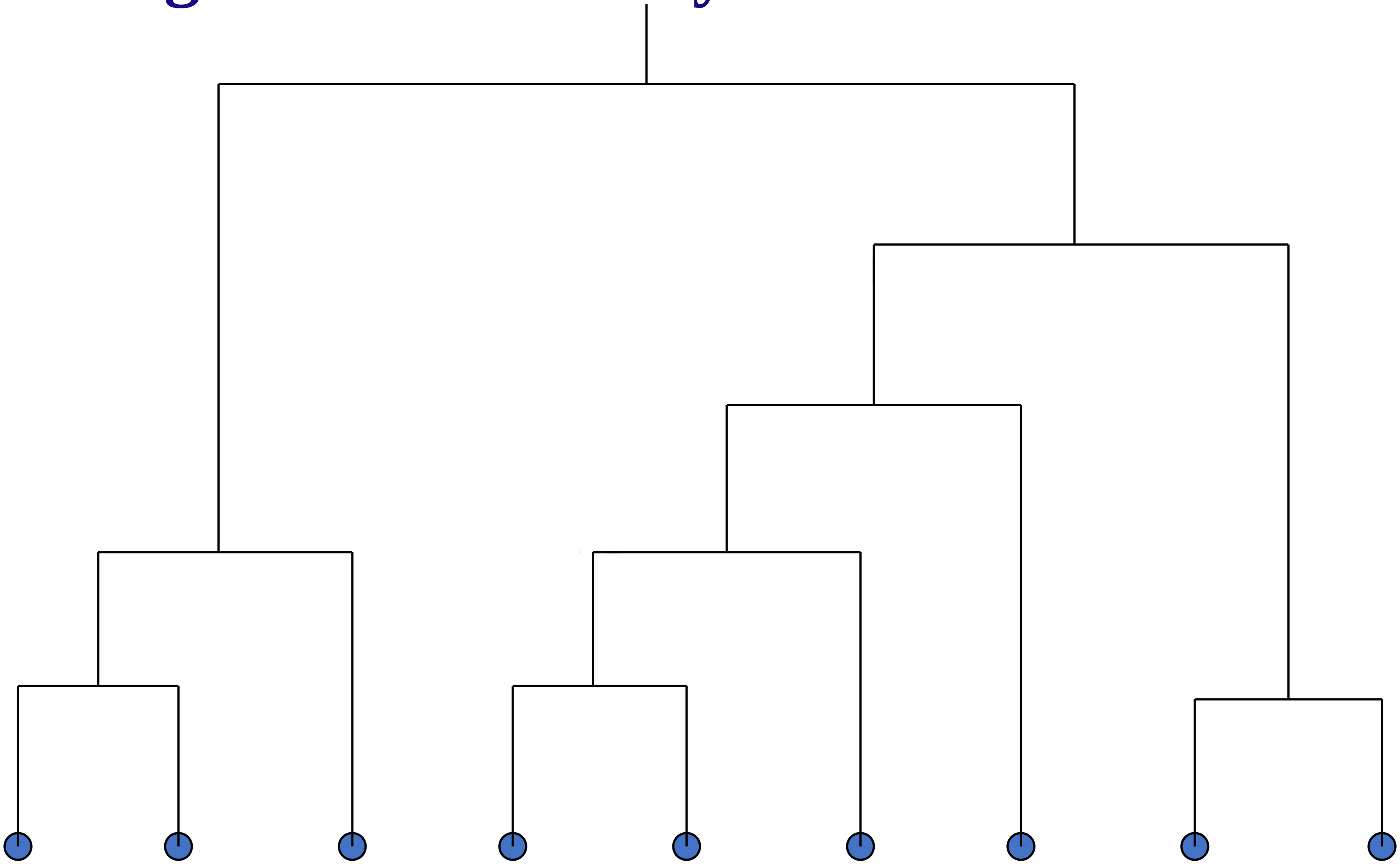
Level 5: ((a,c),b) and ((d,e),f) will be merged to form one cluster [Value: 4.47]

AGNES (Agglomerative Nesting): Bottom-up approach

- Introduced in Kaufmann and Rousseeuw (1990)
- Use the Single-Link method and the dissimilarity matrix.
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



A *Dendrogram* Shows How the Clusters are Merged Hierarchically



Density-Based Clustering Methods

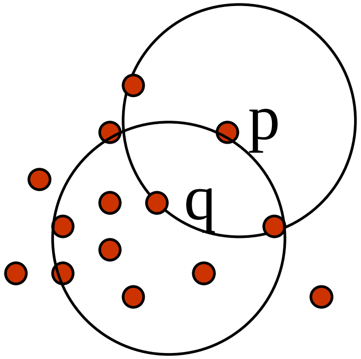
Density-Based Clustering Methods

- Based on density which means number of data points in a given area. More data point means high density and less data points means low density.
- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition
- Examples are DBSCAN, OPTICS, DENCLUE, CLIQUE

Density-Based Clustering: Background

- Two parameters*:*
  - ***Eps***: Maximum radius of the neighbourhood
  - ***MinPts***: Minimum number of points in an Eps-neighbourhood of that point
- $N_{Eps}(p)$: $\{q \text{ belongs to } D \mid dist(p,q) <= Eps\}$
- Directly density-reachable: A point ***p*** is directly density-reachable from a point ***q*** wrt. ***Eps***, ***MinPts*** if
  - 1) ***p*** belongs to $N_{Eps}(q)$
  - 2) core point condition:

$$|N_{Eps}(q)| >= MinPts$$

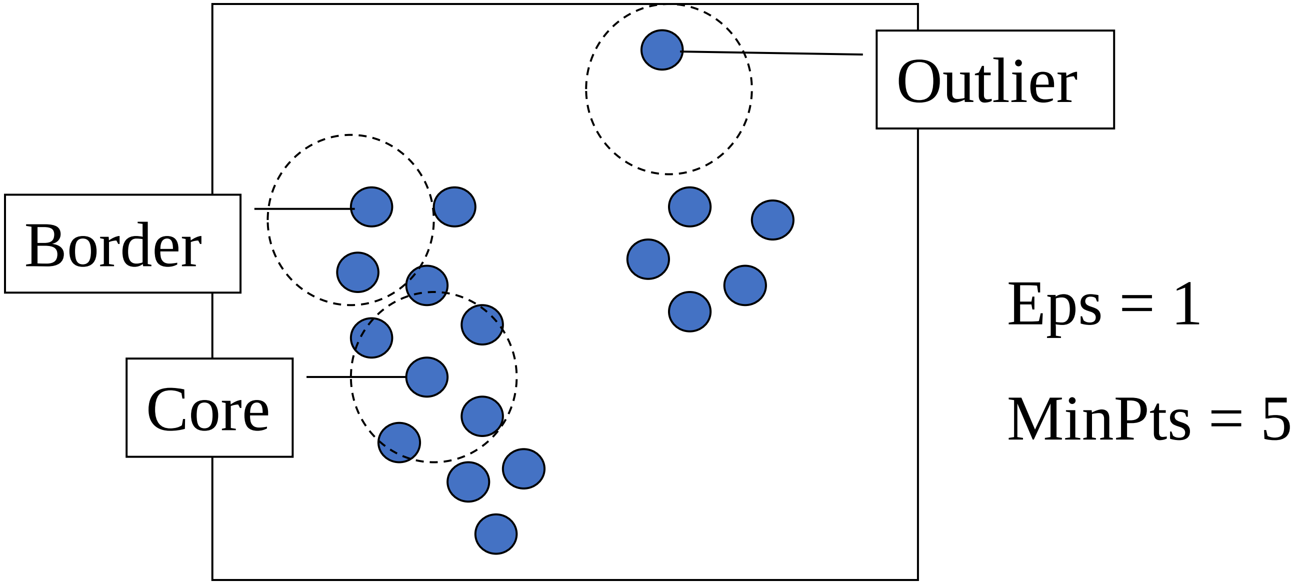


MinPts = 5

Eps = 1

DBSCAN: Density Based Spatial Clustering of Applications with Noise

- Relies on a *density-based* notion of cluster: A *cluster* is defined as a maximal set of density-connected points
- Discovers clusters of arbitrary shape in spatial databases with noise



DBSCAN: The Algorithm

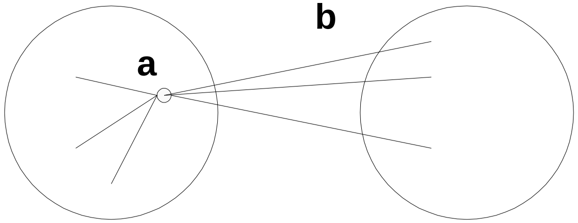
- Arbitrary select a point ***p***
- Retrieve all points density-reachable from ***p*** wrt ***Eps*** and ***MinPts***.
- If ***p*** is a core point, a cluster is formed.
- If ***p*** is a border point, no points are density-reachable from ***p*** and DBSCAN visits the next point of the database.
- Continue the process until all of the points have been processed.

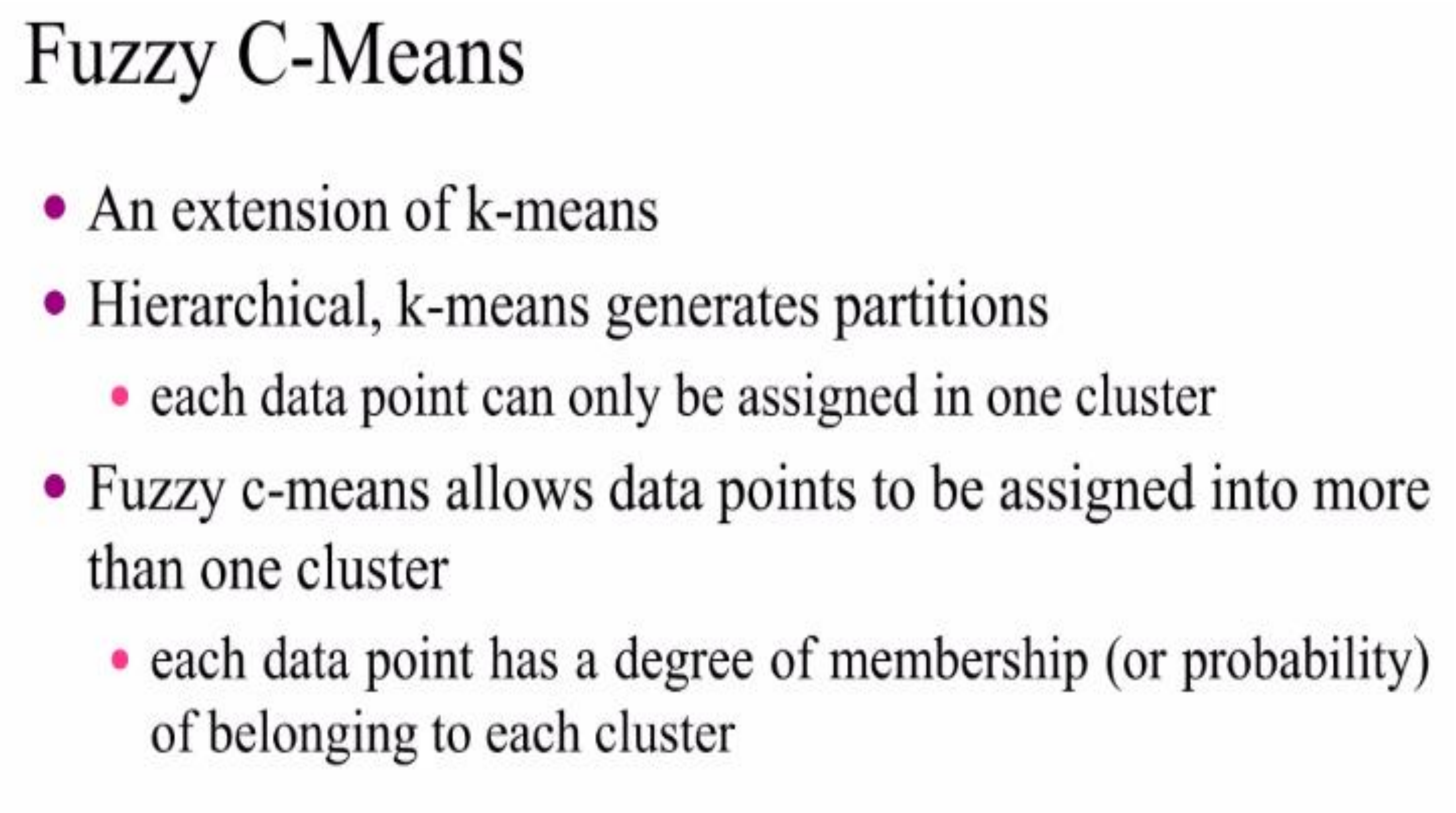
Silhouette Coefficient/ Scores

- Silhouette Coefficient is used for validation of clusters and to compare clustering algorithms.
- It combines ideas of both cohesion and separation for individual points.
- For an individual point, *I*
  - ***a*** = average distance of ***i*** to the points in the same cluster
  - ***b*** = min (average distance of ***i*** to points in another cluster)
  - silhouette coefficient of **i**:

    s = 1 – a/b if **a < b**

  - Typically between 0 and 1.
  - The closer to 1 the better



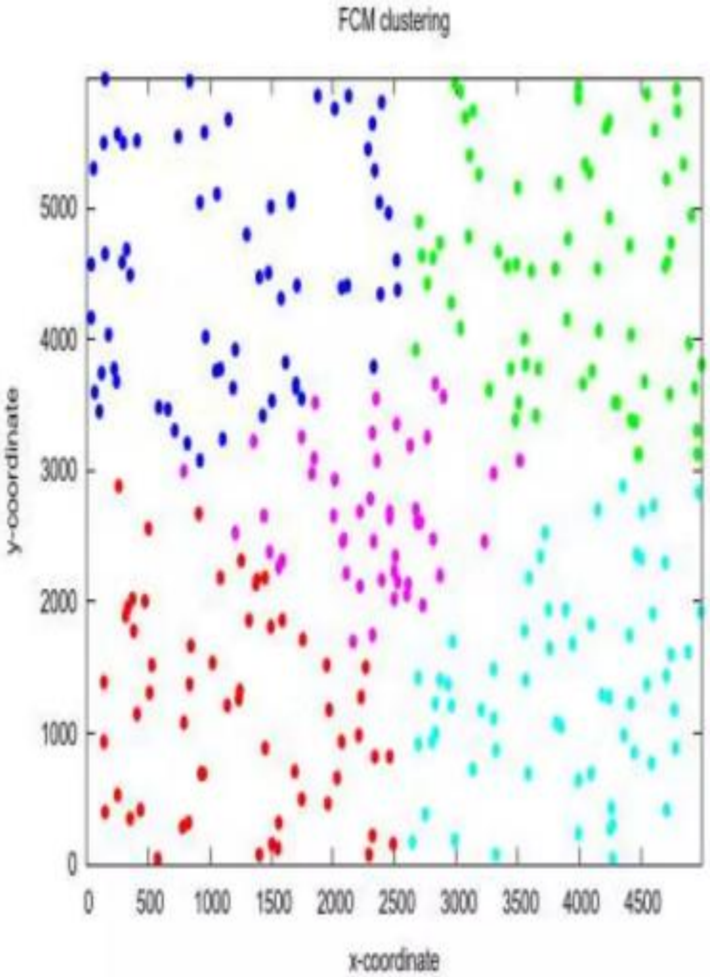


Fuzzy C-Means

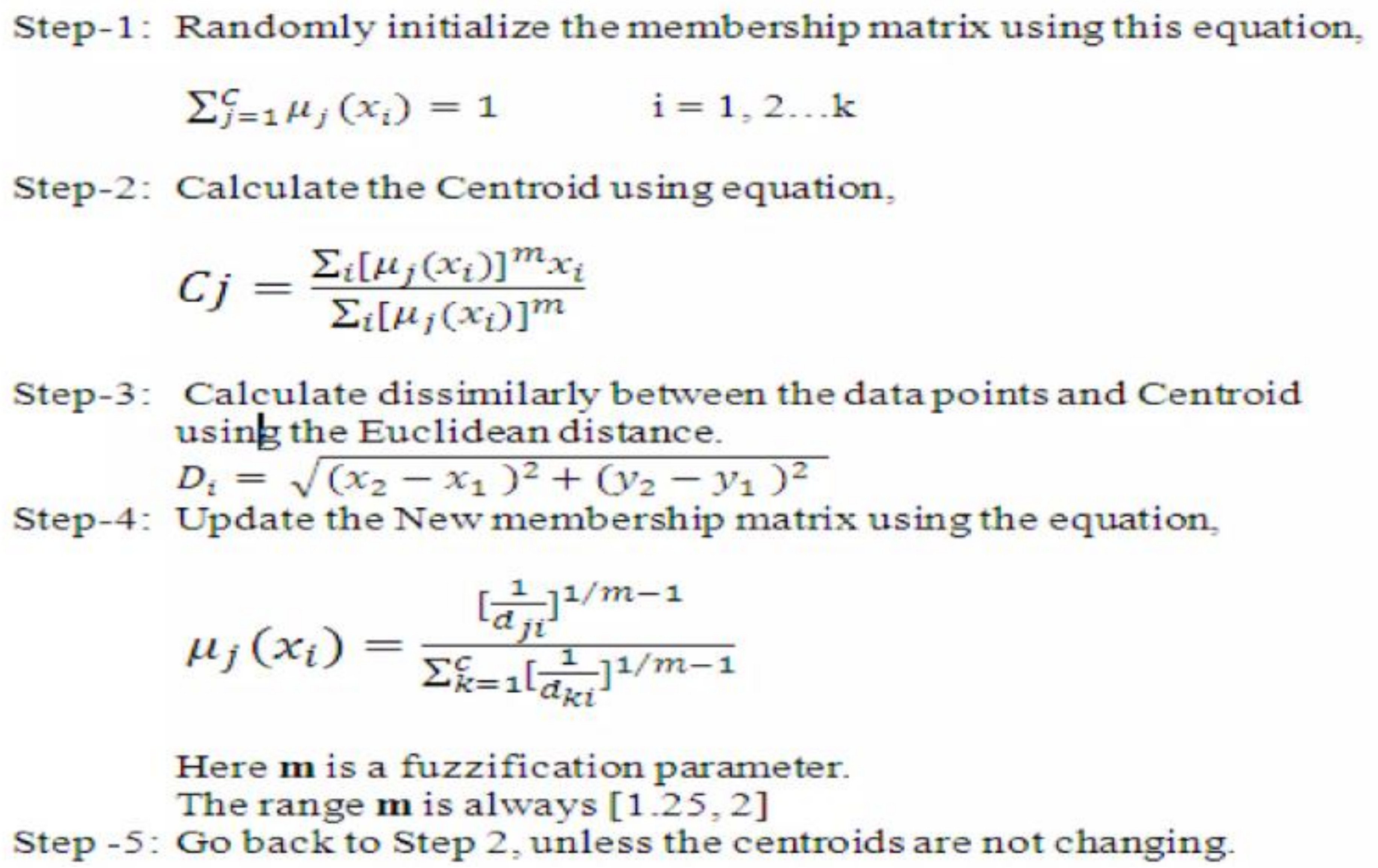
- An extension of k-means
- Hierarchical, k-means generates partitions
  - each data point can only be assigned in one cluster
- Fuzzy c-means allows data points to be assigned into more than one cluster
  - each data point has a degree of membership (or probability) of belonging to each cluster

Fuzzy c-means clustering(FCM)

Fuzzy c-means (FCM) is a method of clustering which allows one piece of data to belong to two or more clusters. This method is frequently used in pattern recognition. It is based on minimization of the objective function !



Fuzzy C Means Algorithm



Step-1: Randomly initialize the membership matrix using this equation,

$$\sum_{j=1}^{c} \mu_j(x_i) = 1 \qquad i = 1, 2 \ldots k$$

Step-2: Calculate the Centroid using equation,

$$Cj = \frac{\sum_i [\mu_j(x_i)]^m x_i}{\sum_i [\mu_j(x_i)]^m}$$

Step-3: Calculate dissimilarly between the data points and Centroid using the Euclidean distance.

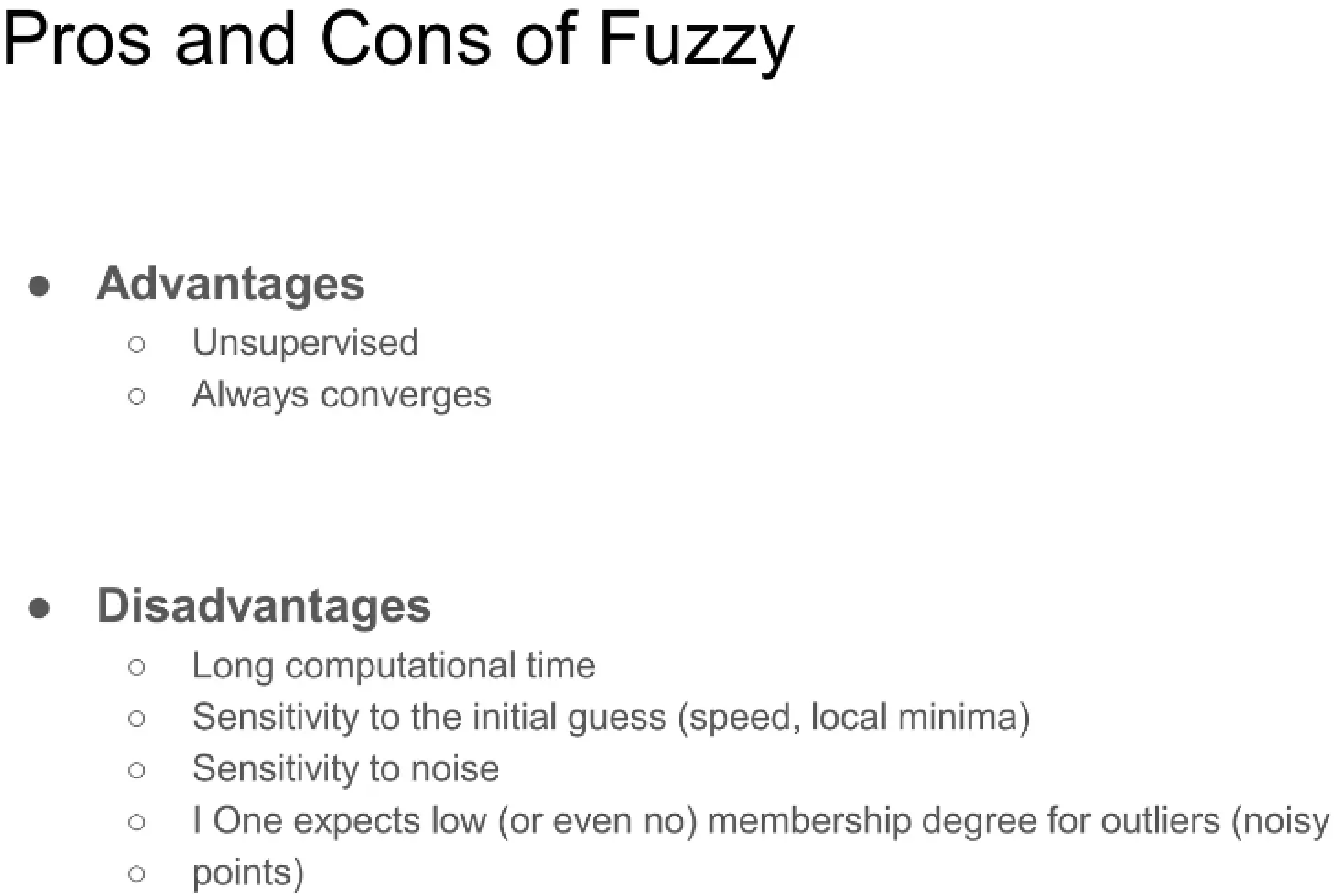
$$D_i = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Step-4: Update the New membership matrix using the equation,

$$\mu_j(x_i) = \frac{[\frac{1}{d_{ji}}]^{1/m-1}}{\sum_{k=1}^{c}[\frac{1}{d_{ki}}]^{1/m-1}}$$

Here **m** is a fuzzification parameter.
The range **m** is always [1.25, 2]

Step -5: Go back to Step 2, unless the centroids are not changing.



Pros and Cons of Fuzzy

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

A novel kernelized fuzzy C-means algorithm with application in medical image segmentation

Dao-Qiang Zhang[1,2] and Song-Can Chen[1,2*]

[1] *Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, P.R. China*

[2] *National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, 100080, P.R. China*

Summary

- Cluster analysis groups objects based on their similarity and has wide applications
- Clustering algorithms can be categorized into partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods
- Outlier detection and analysis are very useful for fraud detection, etc. and can be performed by statistical, distance-based or deviation-based approaches