CLUSTERING
Methods

Prof. Sukhendu Das
Deptt. of CS&E,
IIT Madras,
Chennai, INDIA.

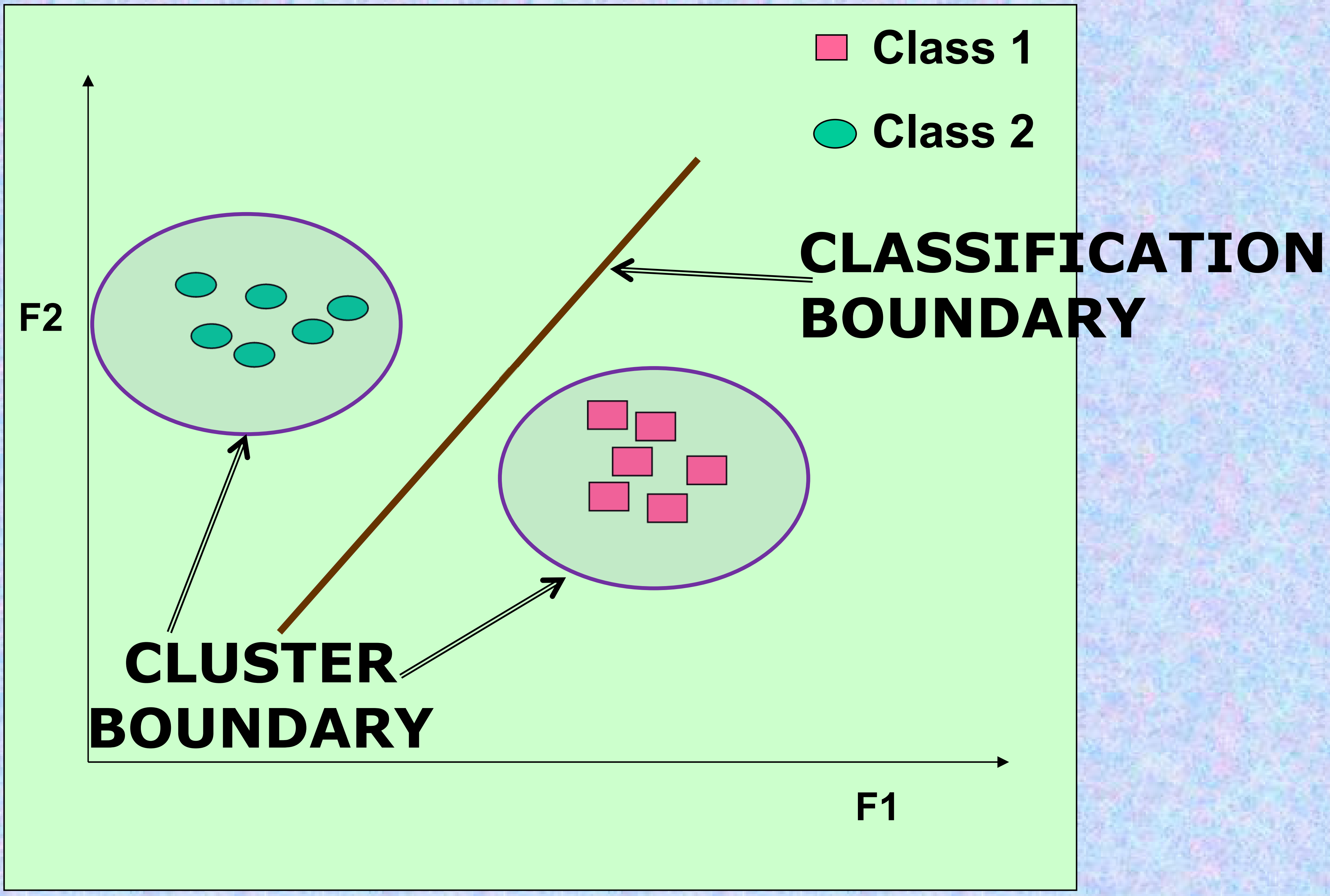
Email: sdas@cse.iitm.ac.in

What is Cluster Analysis?

- Cluster: A collection of data objects
  - similar (or related) to one another within the same group
  - dissimilar (or unrelated) to the objects in other groups
- Cluster analysis (or *clustering*, *data segmentation, …*)
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
- Unsupervised learning: no predefined classes (i.e., *learning by observations* vs. learning by examples: supervised)
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

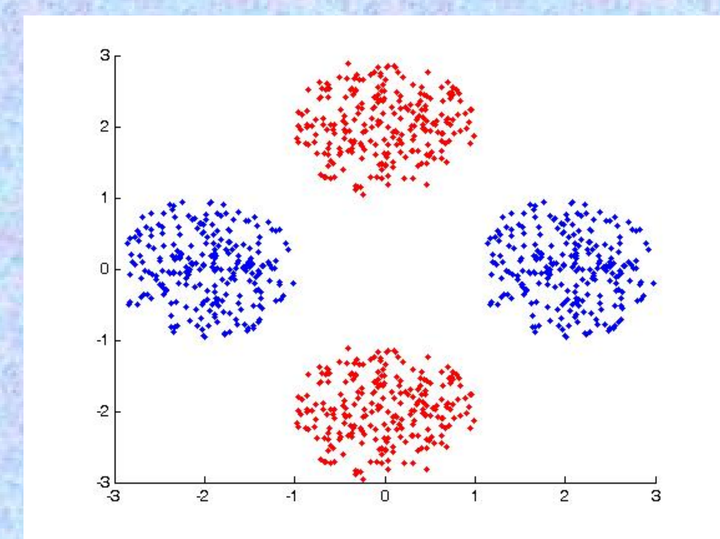
Clustering: Application Examples

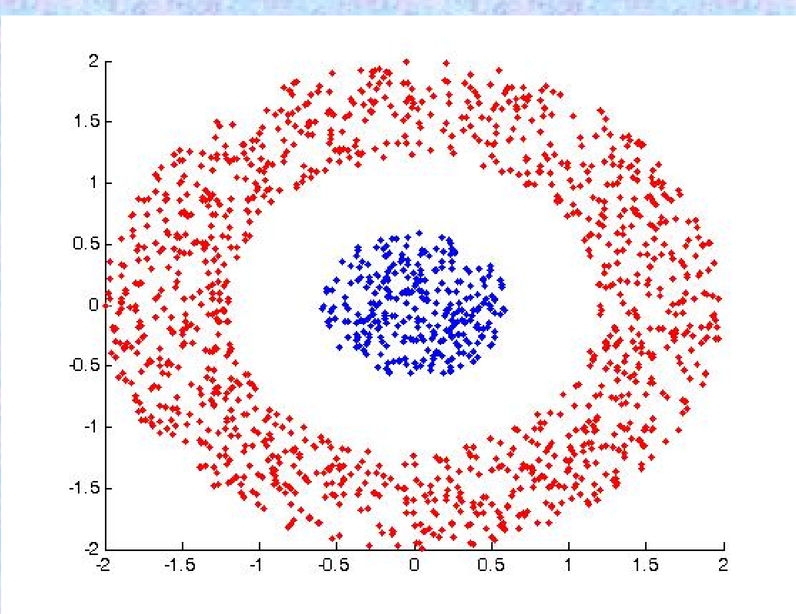
- **Biology**: taxonomy of living things: kingdom, phylum, class, order, family, genus and species
- **Information retrieval**: document clustering
- **Land use**: Identification of areas of similar land use in an earth observation database
- **Marketing**: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- **City-planning**: Identifying groups of houses according to their house type, value, and geographical location
- **Earth-quake studies**: Observed earth quake epicenters should be clustered along continent faults
- **Climate**: understanding earth climate, find patterns of atmospheric and ocean
- **Economic Science**: market research

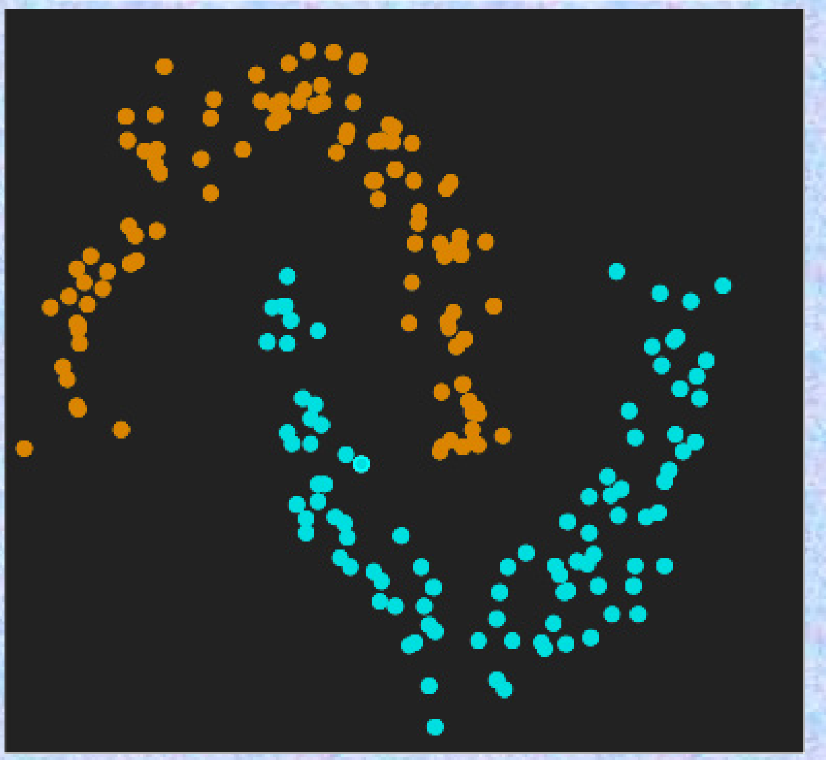


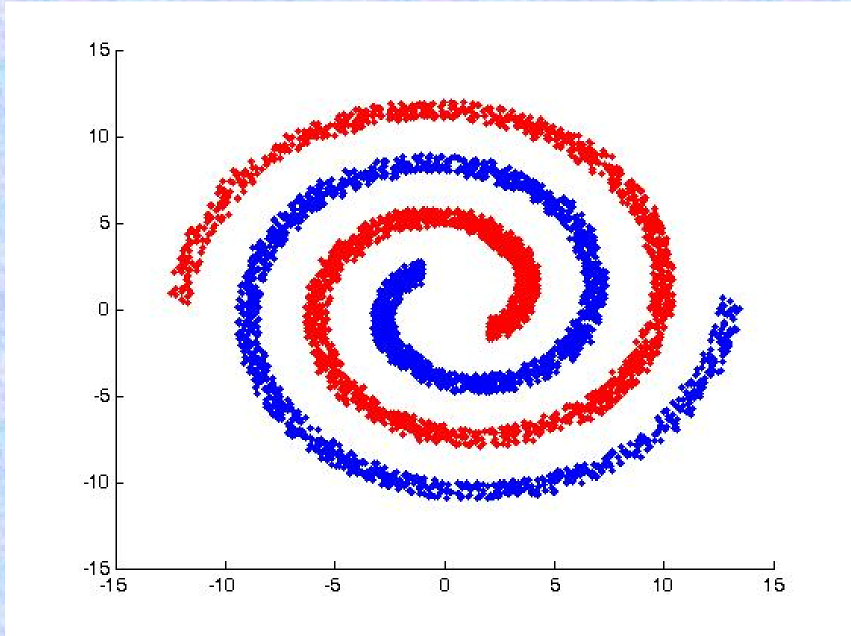
Sample points in a two-dimensional feature space

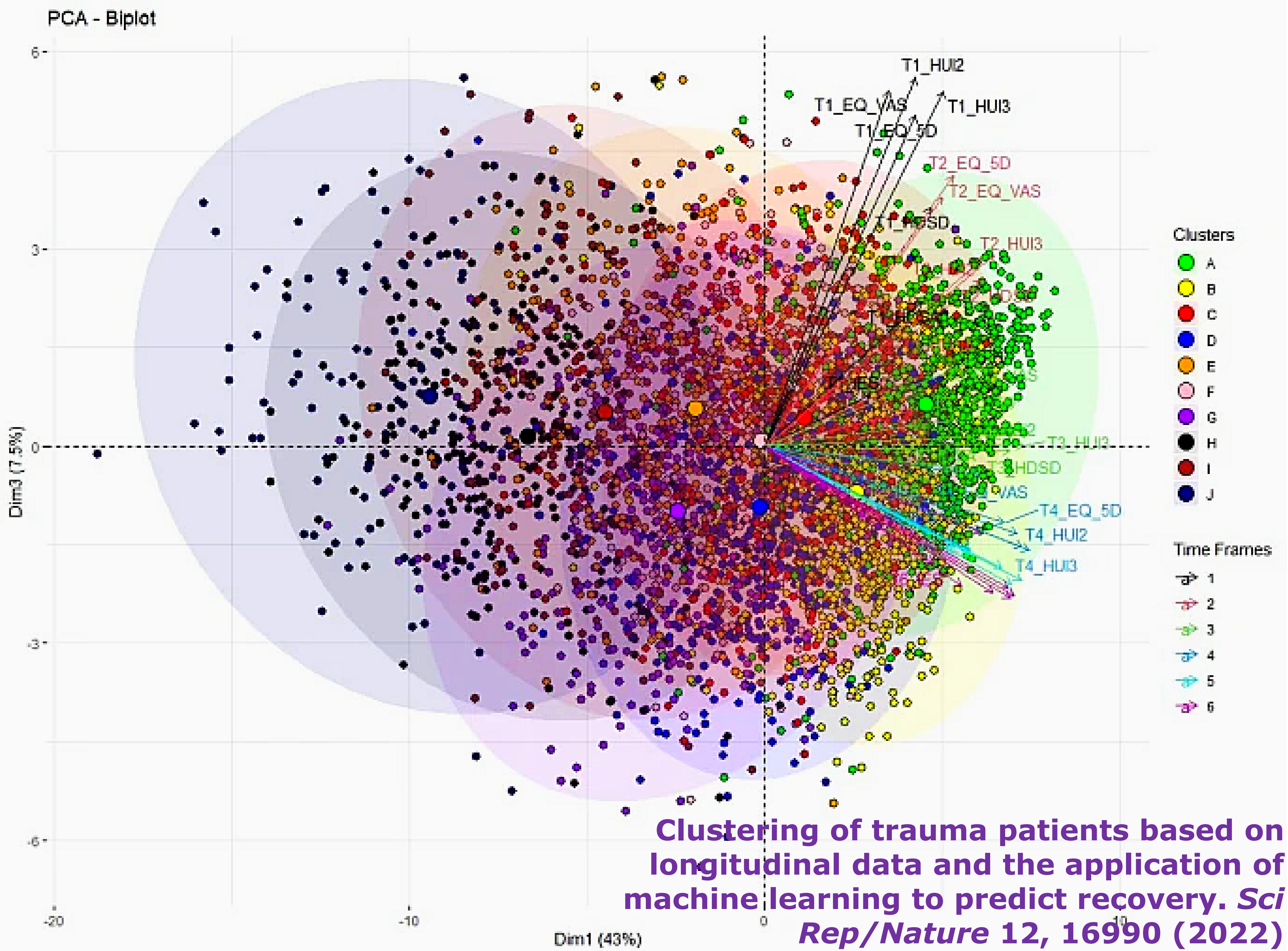
Complex cases of classification and clustering











Clustering of trauma patients based on longitudinal data and the application of machine learning to predict recovery. ***Sci Rep/Nature*** **12, 16990 (2022)**

CLUSTERING	CLASSIFICATION
Data Points have no labels	Most data points have labels

METHODS OF

CLUSTERING AND CLASSIFICATION

- REPRESENTATIVE POINTS
- Split & MERGE
- LINKAGE
- SOM
- MODEL-BASED
- VECTOR QUANTIZATION

Quality: What Is Good Clustering?

- A good clustering method will produce high quality clusters
  - high intra-class similarity: cohesive within clusters
  - low inter-class similarity: distinctive between clusters
- The quality of a clustering method depends on
  - the similarity measure used by the method
  - its implementation, and
  - Its ability to discover some or all of the hidden patterns

Considerations for Cluster Analysis

- **Partitioning criteria**
  - **Single level vs. hierarchical partitioning (often, multi-level hierarchical partitioning is desirable)**
- **Separation of clusters**
  - **Exclusive (e.g., one customer belongs to only one region) vs. non-exclusive (e.g., one document may belong to more than one class)**
- **Similarity measure**
  - **Distance-based (e.g., Euclidian, road network, vector) vs. connectivity-based (e.g., density or contiguity)**
- **Clustering space**
  - **Full space (often when low dimensional) vs. subspaces (often in high-dimensional clustering)**

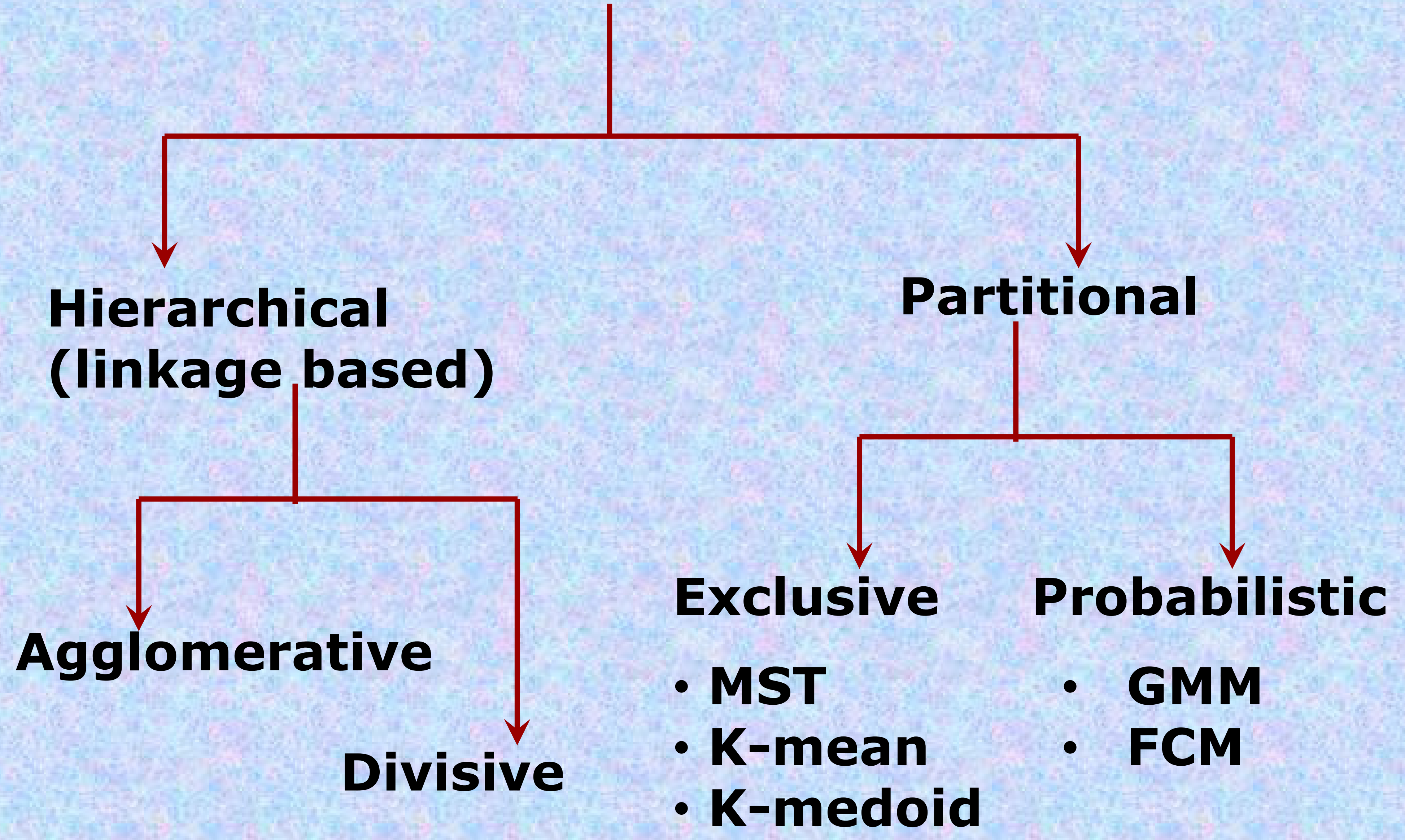
Major Clustering Approaches (I)

- <u>Partitioning approach</u>:
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
  - Typical methods: k-means, k-medoids, CLARANS
- <u>Hierarchical approach</u>:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Diana, Agnes, BIRCH, CAMELEON
- <u>Density-based approach</u>:
  - Based on connectivity and density functions
  - Typical methods: DBSCAN, OPTICS, DenClue
- <u>Grid-based approach</u>:
  - based on a multiple-level granularity structure
  - Typical methods: STING, WaveCluster, CLIQUE

Major Clustering Approaches (II)

- **Model-based:**
  - A model is hypothesized for each of the clusters and tries to find the best fit of that model to each other
  - Typical methods: EM, SOM, COBWEB
- **Frequent pattern-based:**
  - Based on the analysis of frequent patterns
  - Typical methods: p-Cluster
- **User-guided or constraint-based:**
  - Clustering by considering user-specified or application-specific constraints
  - Typical methods: COD (obstacles), constrained clustering
- **Link-based clustering:**
  - Objects are often linked together in various ways
  - Massive links can be used to cluster objects: SimRank, LinkClus

GENERAL CATEGORIES of CLUSTERING DATA



Alternative view of
Algorithms for CLUSTERING

- **Unupervised Learning/Classification:**
  - **K-means; K-medoid**

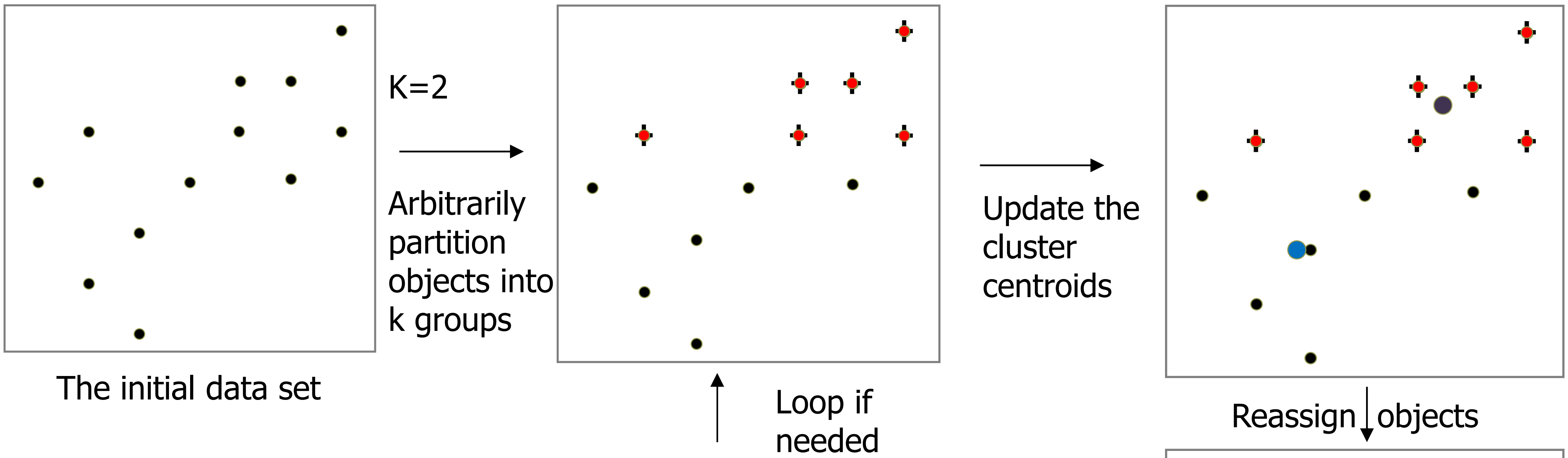
- **Density Estimation :**
  (i) Parametric
  - **Gaussian**
  - **MOG (Mixture of Gaussians)**
  - **Dirichlet, Beta etc.**
  - **Branch and Bound Procedure**
  - **Piecewise Quadratic Boundary**
  - **Nearest Mean Classifier**
  - **MLE (maximum Likelihood Estimate)**

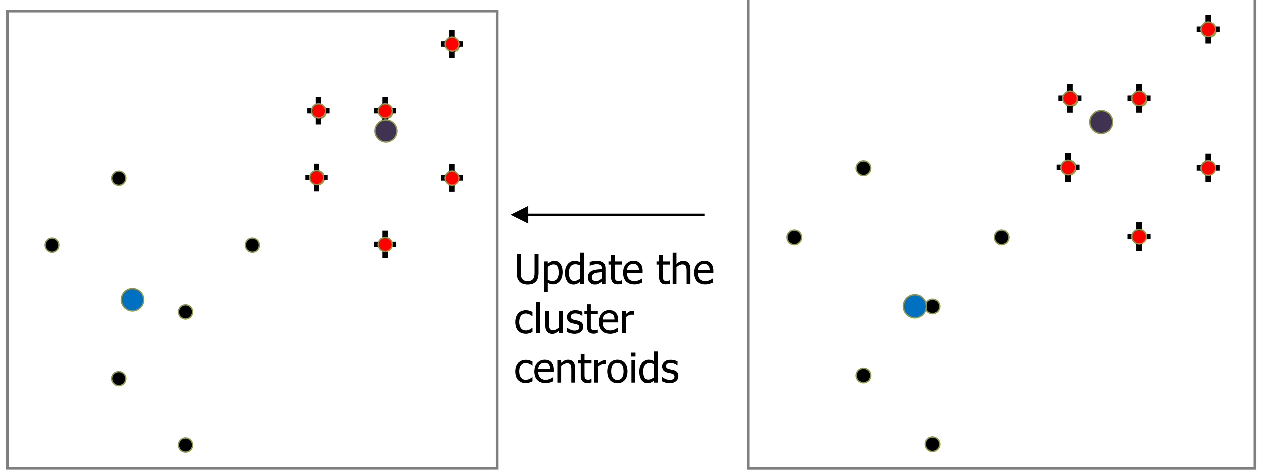
- Density Estimation :

(ii) Non-Parametric

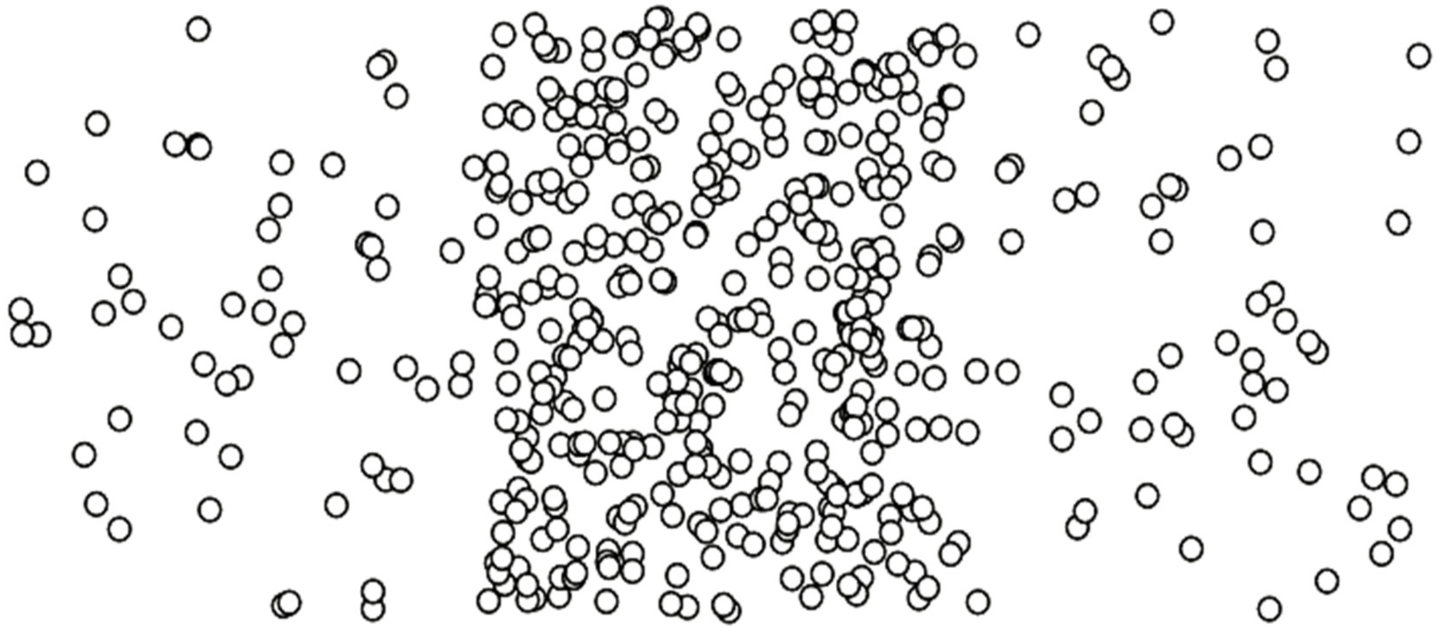
- **Histogram**
- **Neighborhood**
- **Kernel Methods**
- **Graph Theoretic**
- **Iterative Valley Seeking**

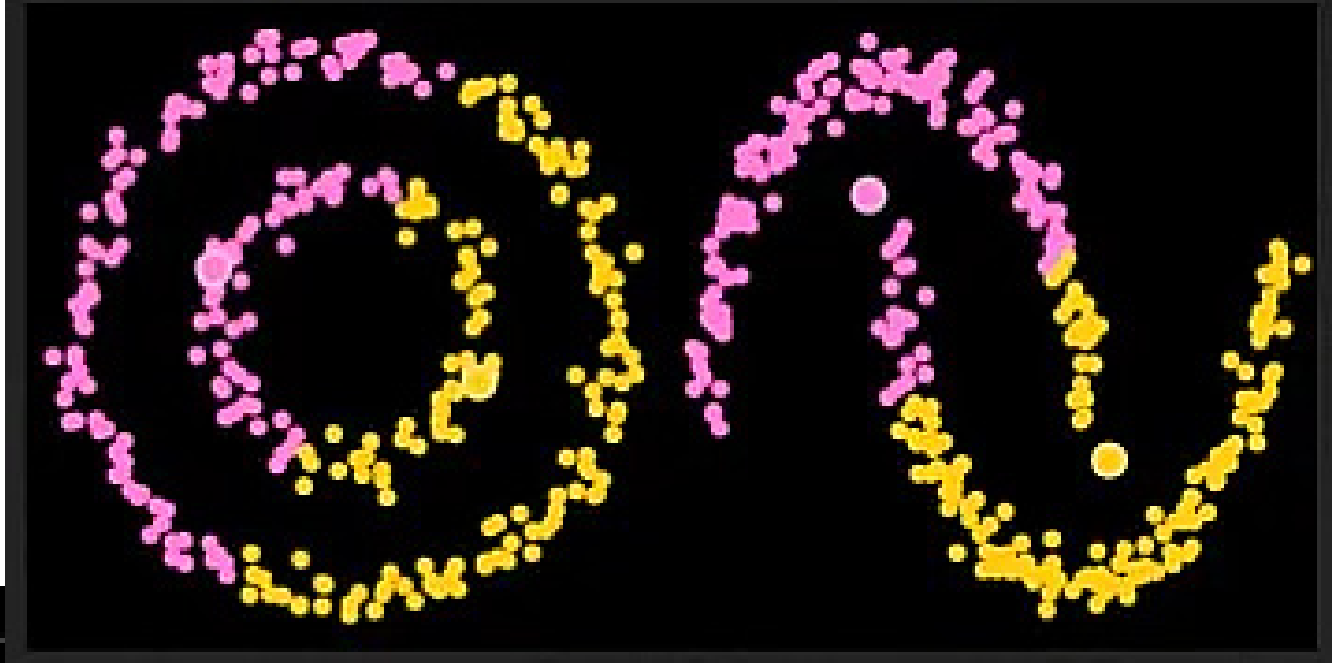
An Example of *K-Means* Clustering

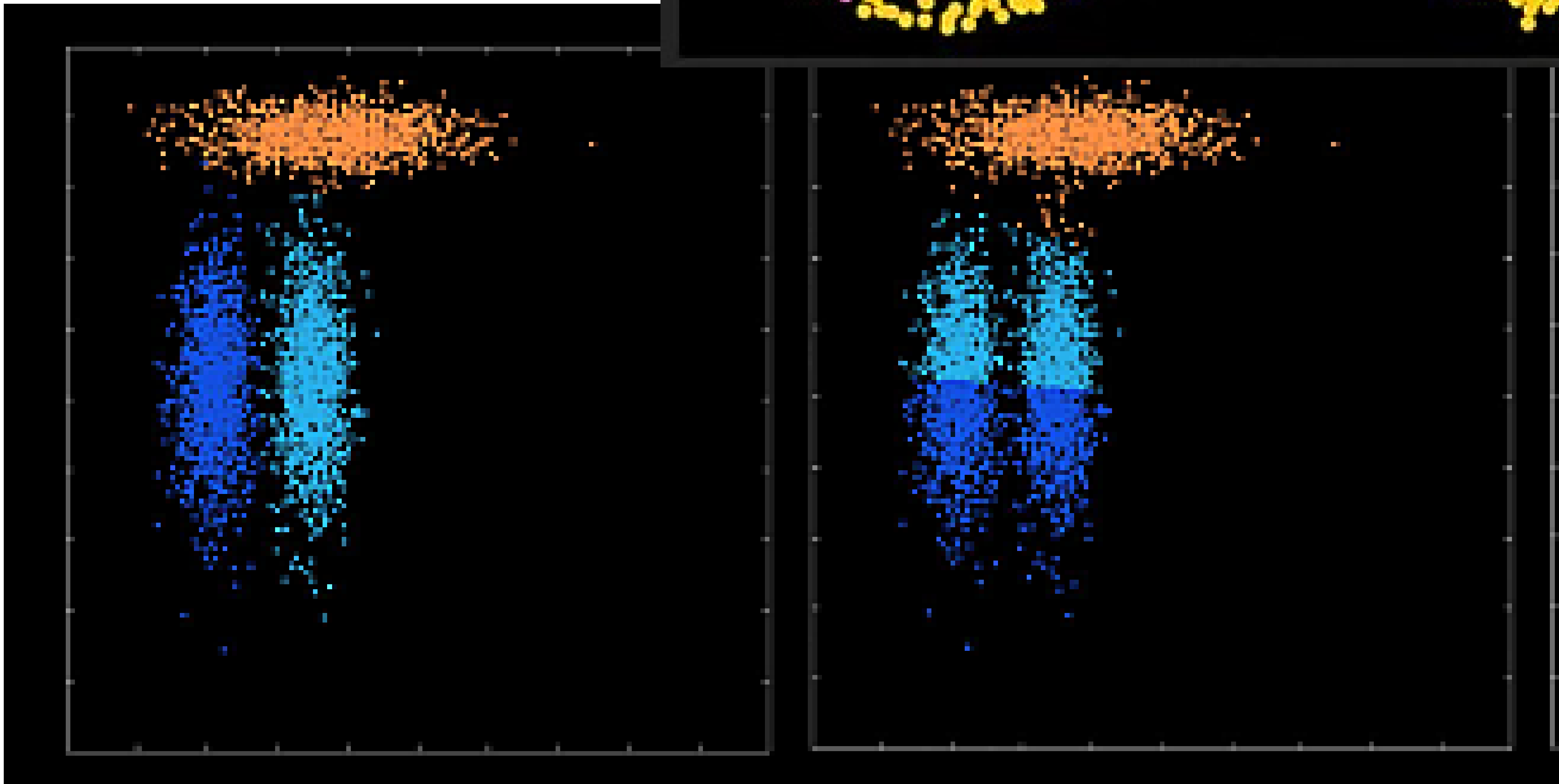




- Partition objects into *k* nonempty subsets
- Repeat
  - Compute centroid (i.e., mean point) for each partition
  - Assign each object to the cluster of its nearest centroid
- Until no change







(a) Generated synthetic data

(b) K-means

FCM - Fuzzy C-Means Clustering

FCM

- A method of clustering which allows one piece of data to belong to two or more clusters.

- Objective function to be minimized:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - \mu_j \right\|^2, \qquad 1 \leq m < \infty$$

Where

- u_{ij} is the degree of membership of x_j in the cluster j.
- x_i is d-dimensional observation
- μ_j is d-dimensional center of cluster j

Updation

- FCM is an iterative optimization approach.

- At each step, the membership u_{ij} and the cluster centers μ_j are updated as follows:

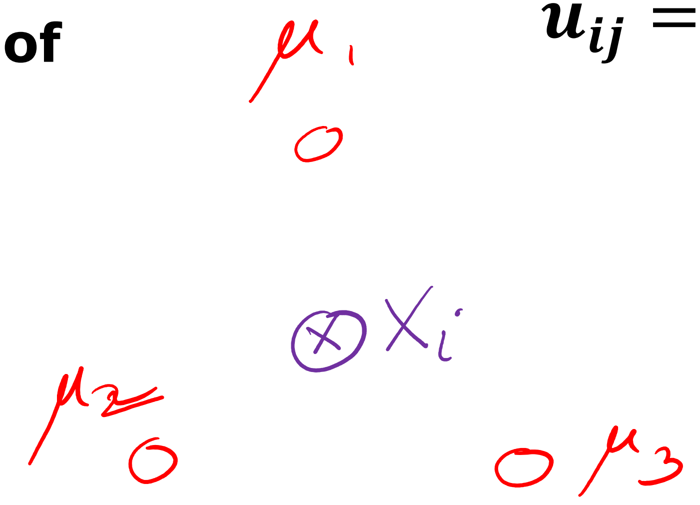
$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_i - \mu_j\|}{\|x_i - \mu_k\|} \right)^{\frac{2}{m-1}}},$$

$$\mu_j = \frac{\sum_{i=1}^{N} u_{ij}^m . x_i}{\sum_{i=1}^{N} u_{ij}^m}$$

Let C = 3; d = 2;
Class Means on vertices of an Equilateral Triangle.

$$u_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{\|x_i - \mu_j\|}{\|x_i - \mu_k\|}\right)^{\frac{2}{m-1}}}$$

$u_{ij} = 1/3$



$m > 1$

$\ell = 2/(m-1)$

u_{i1}

$u_{i2/3} =$

Let C = 3; d = 2;
Class Means on vertices of an Equilateral Triangle.

$$u_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{\|x_i-\mu_j\|}{\|x_i-\mu_k\|}\right)^{\frac{2}{m-1}}}$$

$$u_{ij} = 1/3$$

$$m > 1$$

$$\ell = 2/(m-1)$$

$$\alpha = \frac{|x_i - \mu_k|}{|x_i - \mu_j|}, \quad k \neq j$$

$$u_{i1} = \frac{1}{1 + \left(\frac{1}{\alpha}\right)^{\ell} + \left(\frac{1}{\alpha}\right)^{\ell}} = \frac{\alpha^{\ell}}{\alpha^{\ell}+2} \simeq 1 \qquad \alpha \gg 1$$

$$u_{i2/3} = \frac{1}{\left(\frac{1}{\alpha}\right)^{\ell} + 1 + 1} \simeq \alpha^{\ell} \qquad \alpha \ll 1$$

Go ahead;
Plot them

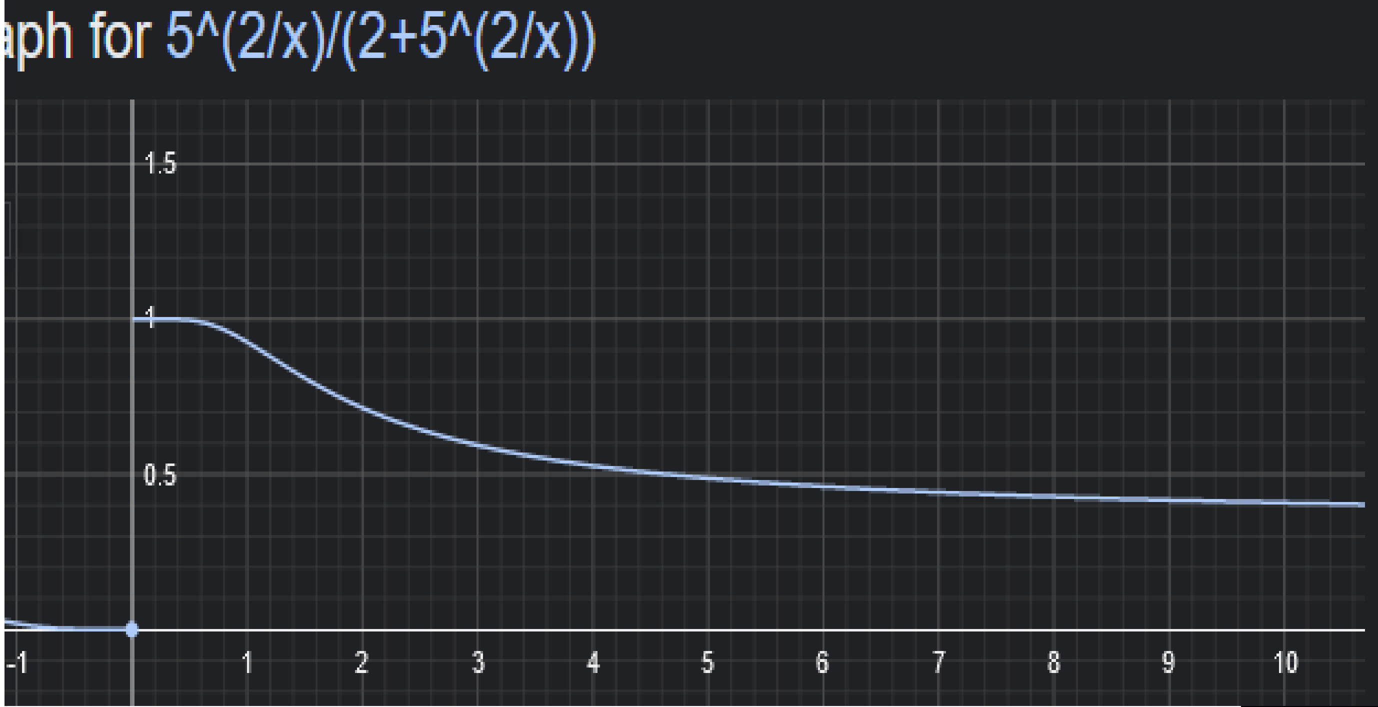
***U* vs *x* = (*m*-1)**

$$y = \frac{5^{2/x}}{2+5^{2/x}}$$

$$\frac{\alpha^{t}}{\alpha^{t}+2}$$

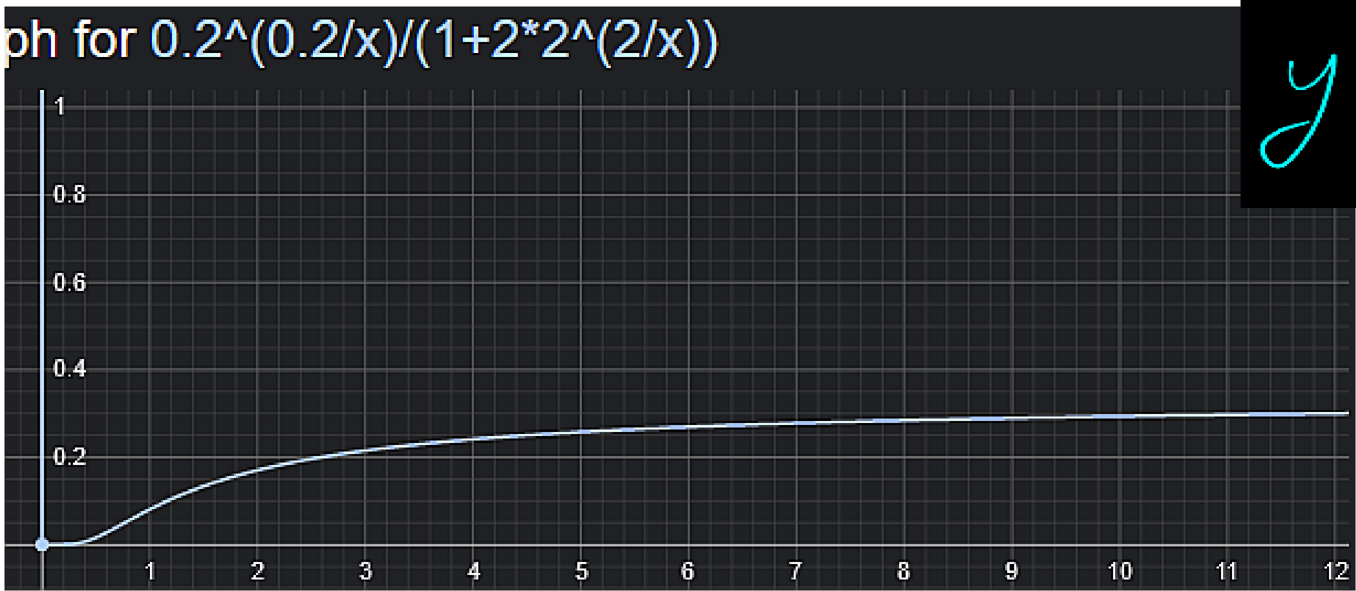
$$y = \frac{0.2^{2/x}}{1+2(0.2^{2/x})}$$

$$\frac{1}{\left(\frac{1}{\alpha}\right)^{t}+1+1}$$



$$y = \frac{5^{2/x}}{2+5^{2/x}}$$

$$y = \frac{0\cdot 2^{2/x}}{1+2\left(2^{2/x}\right)}$$



Termination Criterion

- Iteration stops, when

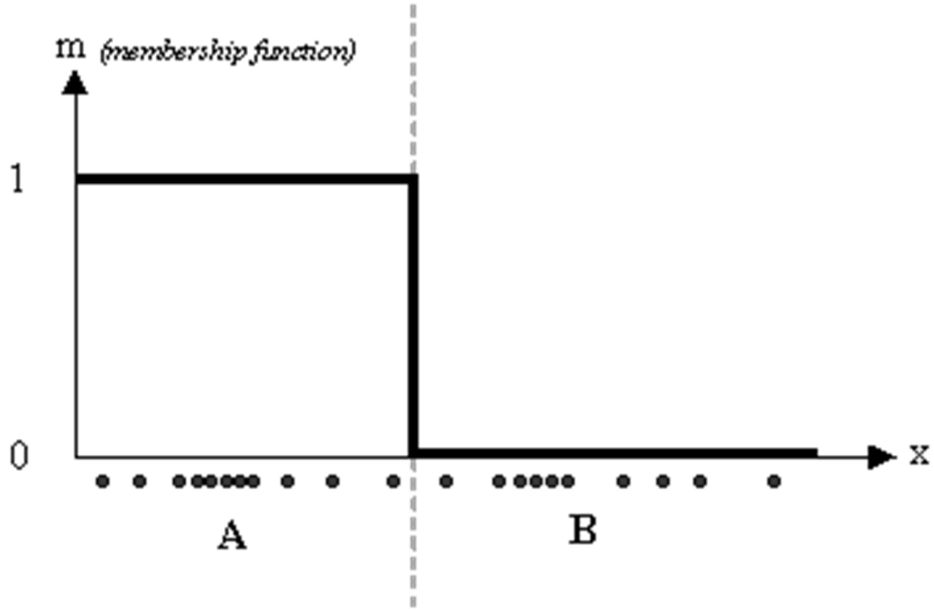
$$\max_{ij}\left\{\left|u_{ij}^{(k+1)} - u_{ij}^{(k)}\right|\right\} < \epsilon$$

Where k is the iteration number.

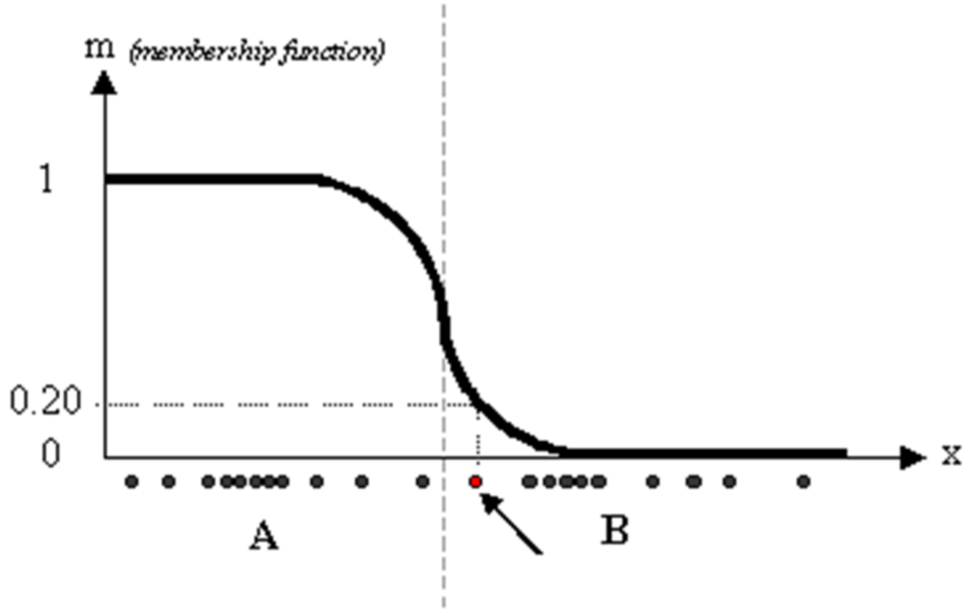
ϵ is between 0 and 1

K-means Vs FCM

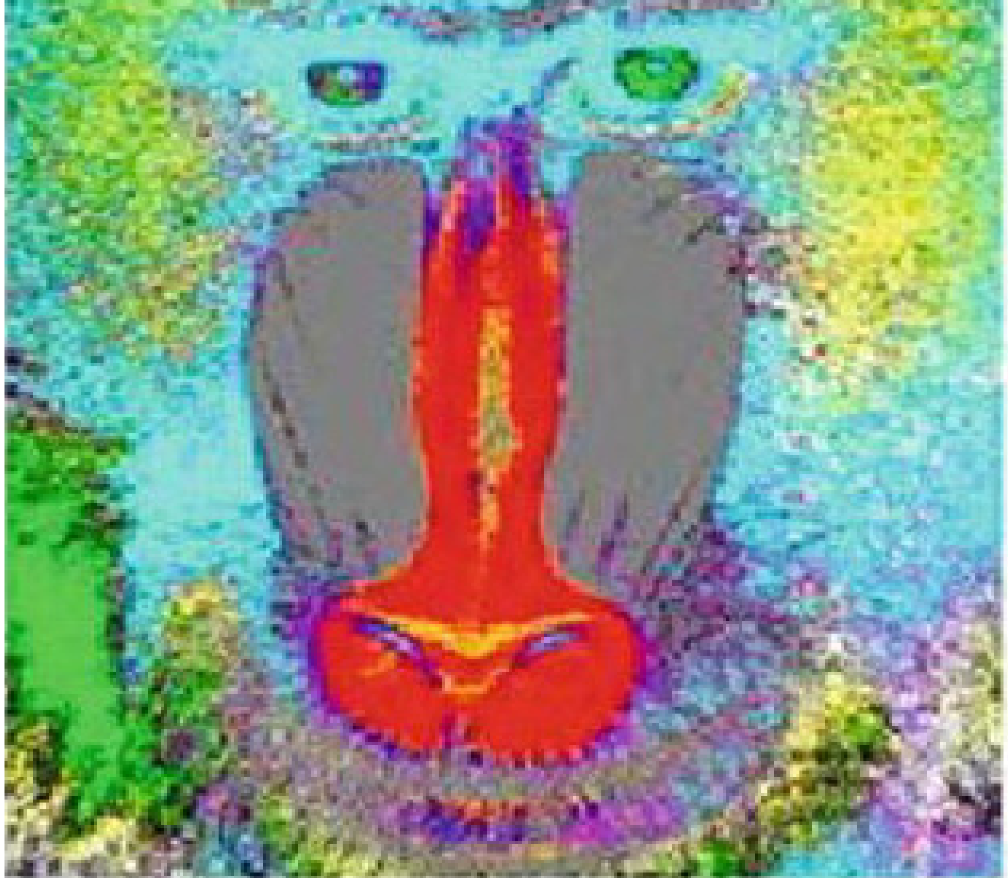
K-means



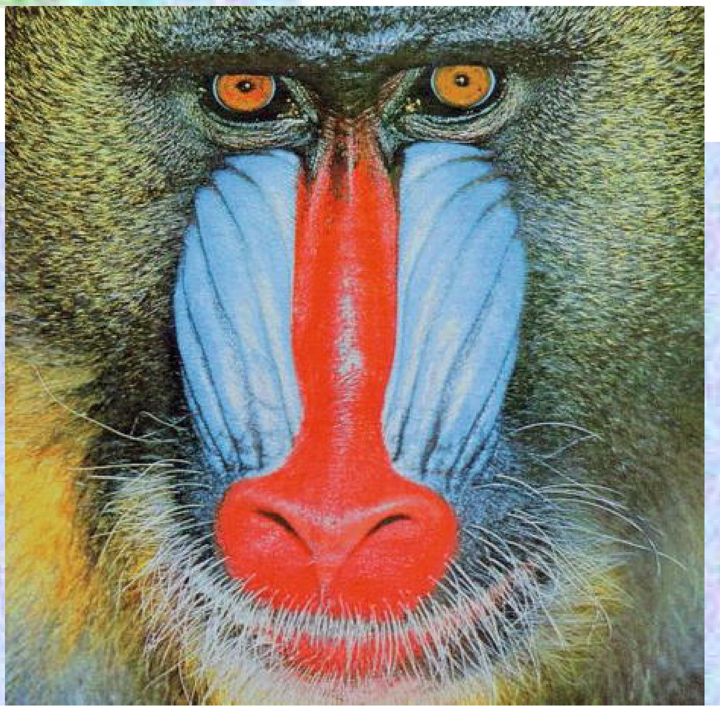
FCM



Read about K-medoids

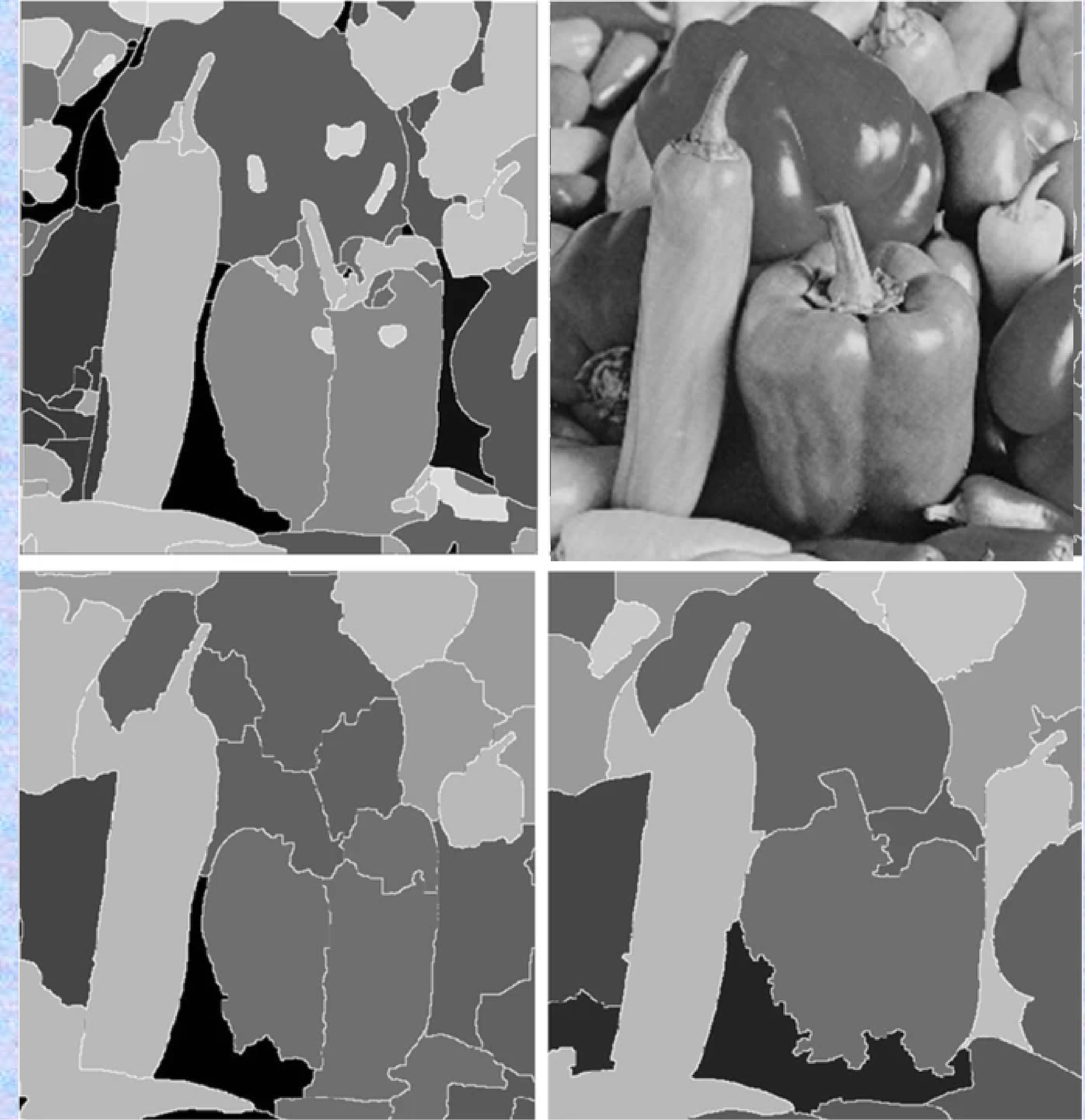


K-Means



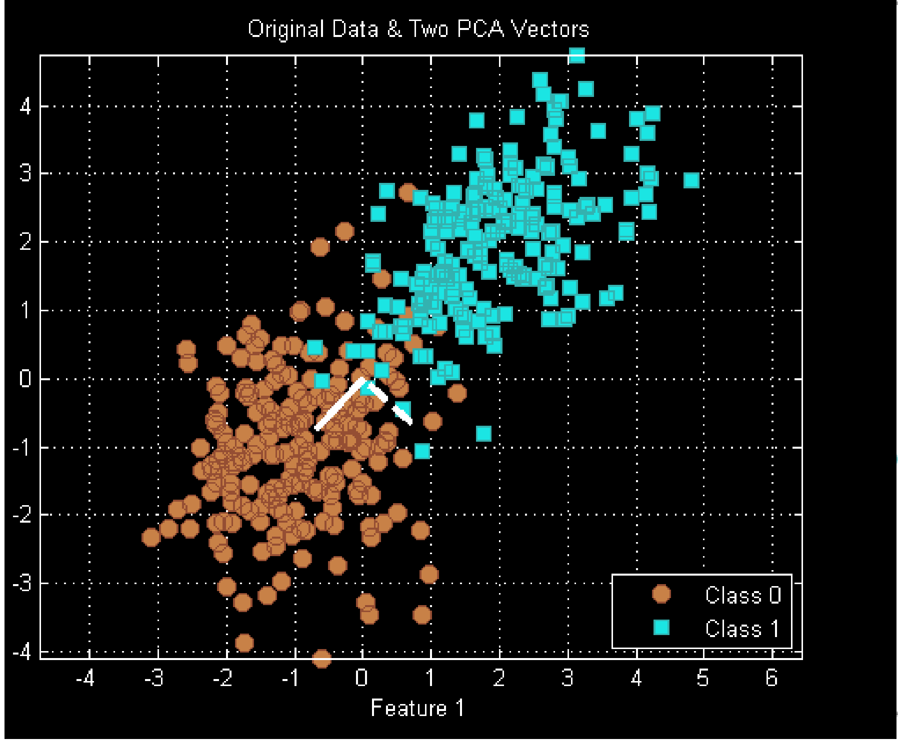
FCM

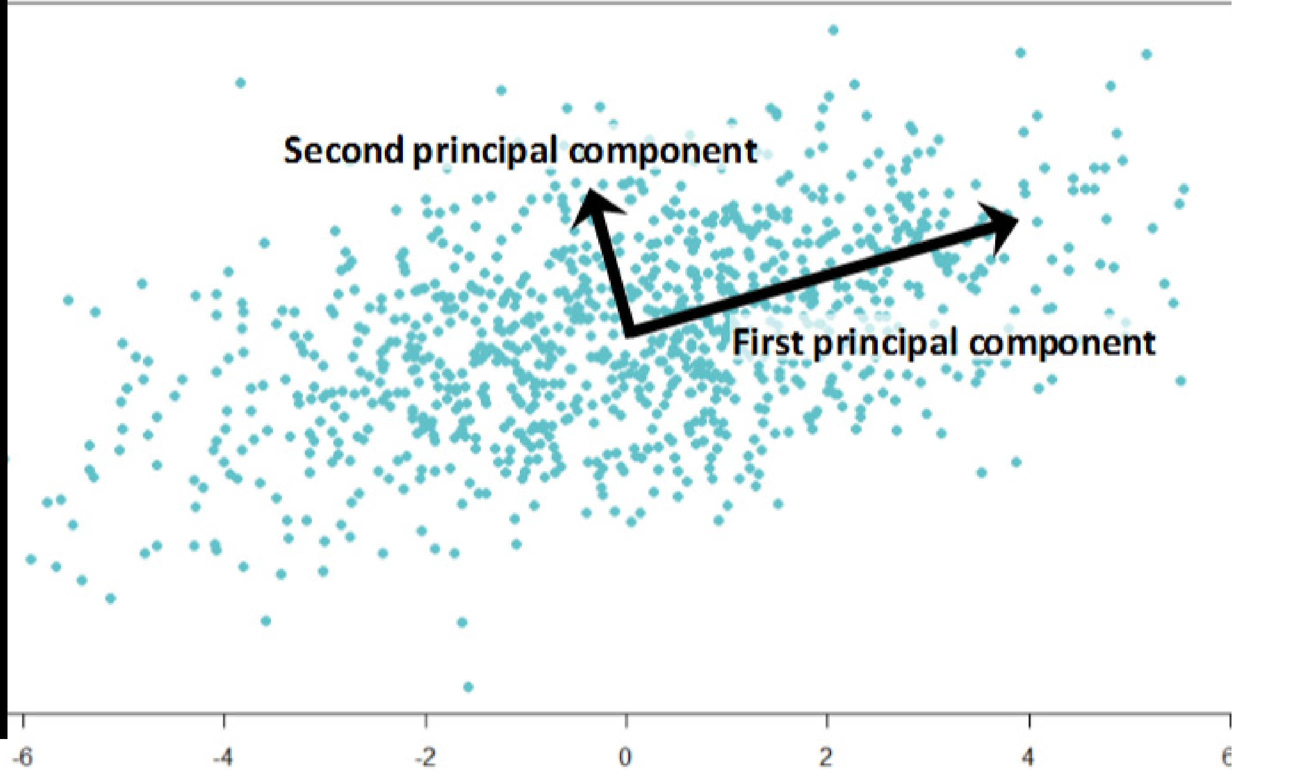
EURASIP 2019

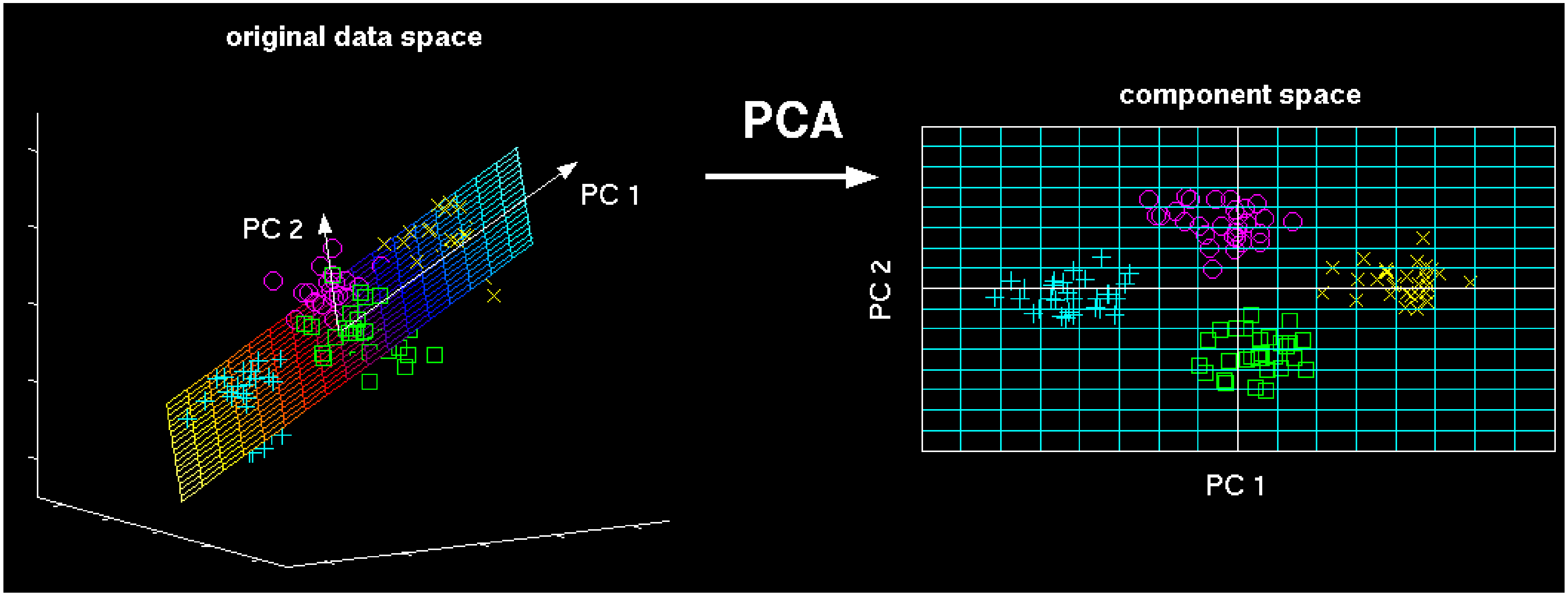


K-means

FCM


Original Data & Two PCA Vectors
4
3
2
1
0
-1
-2
-3
-4
-4 -3 -2 -1 0 1 2 3 4 5 6
Feature 1
Class 0
Class 1


Second principal component
First principal component
-6 -4 -2 0 2 4 6


original data space
PC 2
PC 1
PCA
component space
PC 2
PC 1

Hierarchical Clustering

Hierarchical Clustering

- Builds hierarchy of clusters

- Types:
  - Bottom Up - *Agglomerative*
    - *Starts by considering each observation as a cluster of it's own*
    - *Clusters are merged as we move up the hierarchy*

  - Top Down - *Divisive*
    - *Starts by considering all observations in one cluster*
    - *Clusters are divided as we move down the hierarchy*

Distance Functions

Certain mathematical properties are expected of any distance measure, or *metric*:

1. $d(x, y) \geq 0$ for all x, y.
2. $d(x, y) = 0$ iff $x = y$.
3. $d(x, y) = d(y, x)$ (symmetry)
4. $d(x, y) \leq d(x, z) + d(z, y)$ for all x, y, and z. (triangle inequality)

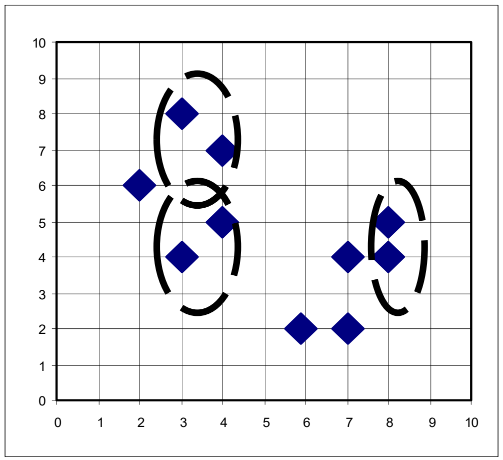
Euclidean distance $d(x, y) = \sqrt{\Sigma_{i=1}^{d} |x_i - y_i|^2}$ is probably the most commonly used metric. Note that it weights all features/dimensions "equally".

Some commonly used Metrics

- Euclidean distance
- Squared Euclidean distance
- Manhattan distance
- Maximum distance
- Mahalanobis distance

Agglomerative clustering

- Each node/object is a cluster initially
- Merge clusters that have the **least** dissimilarity
  - Ex: single-linkage, complete-linkage, etc.
- Go on in a non-descending fashion
- Eventually, all nodes belong to the same cluster



Linkage Criteria

- Determines the distance between sets of observations as a function of the pairwise distances between observations.

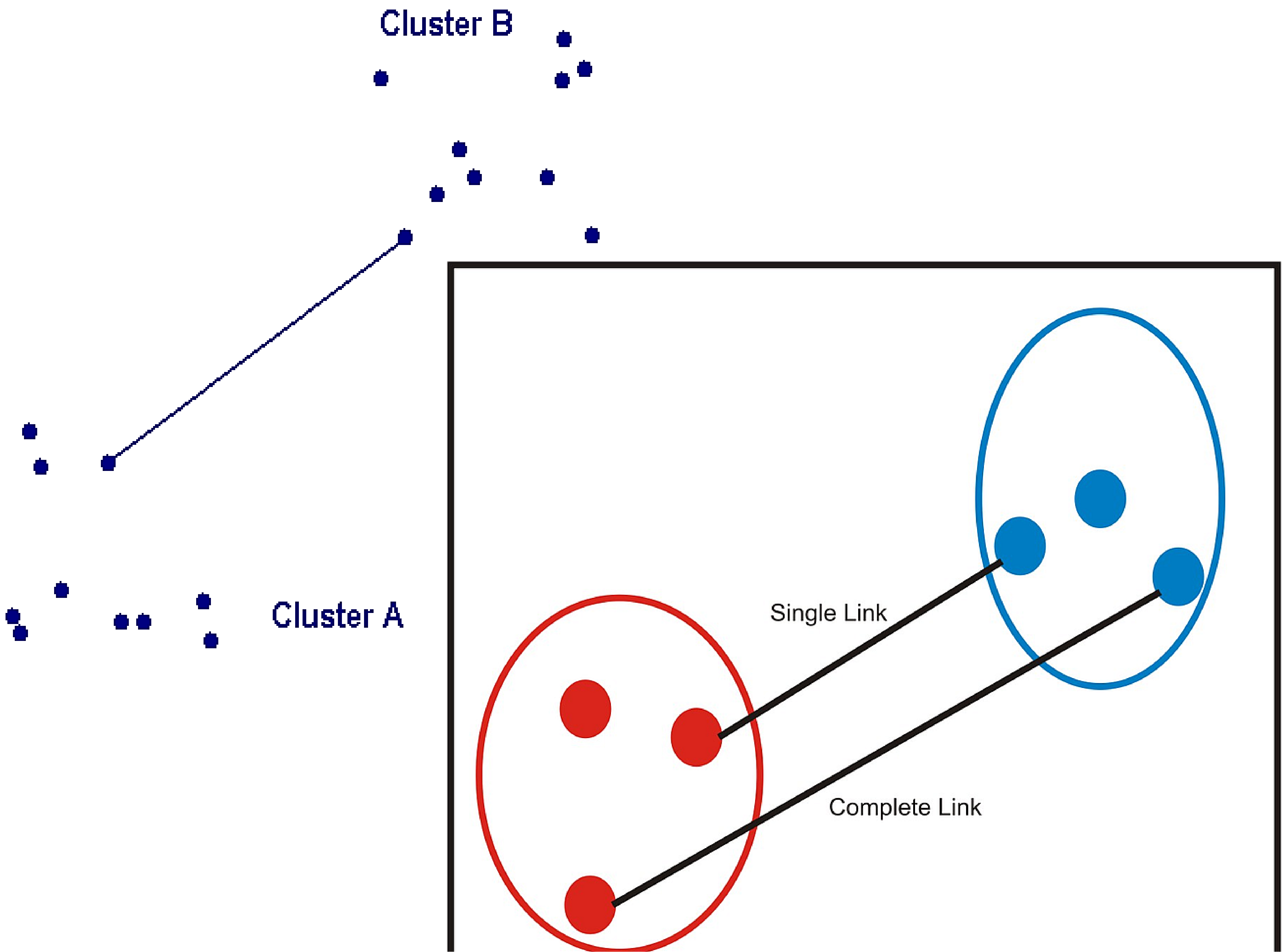
- Some commonly used criterias:
  - *Single Linkage:* Distance between two clusters is the **smallest** pairwise distance between two observations/nodes, each belonging to different clusters.

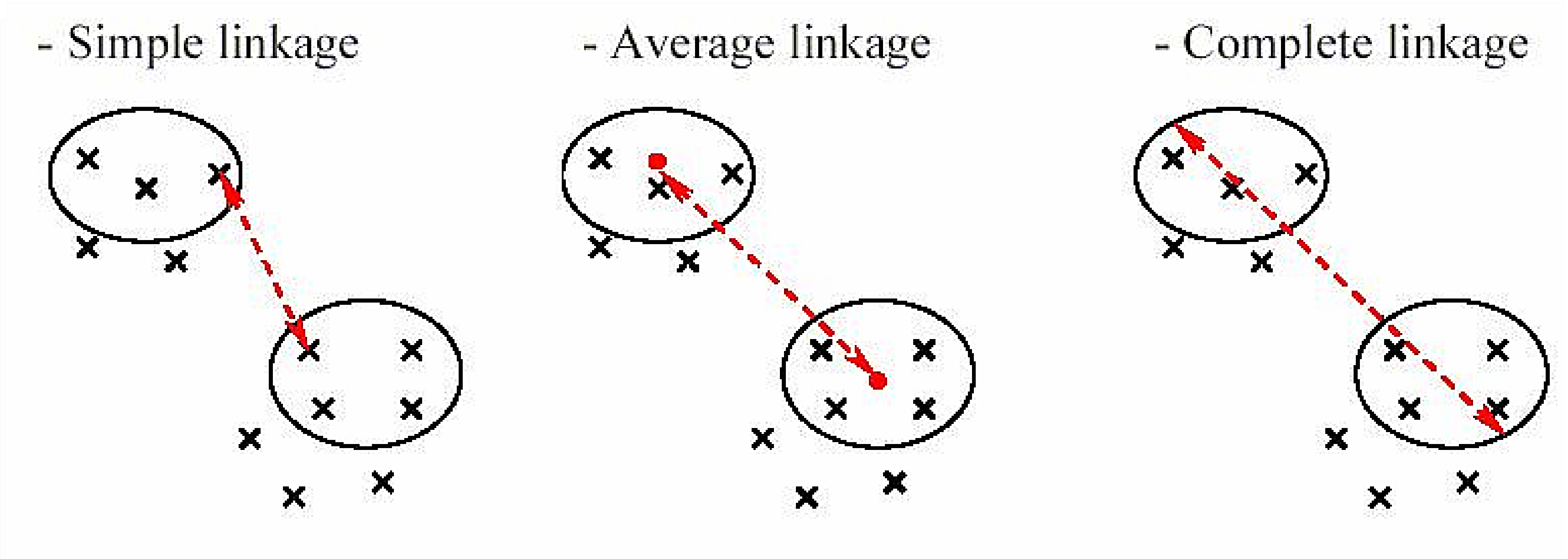
  - *Complete Linkage:* Distance between two clusters is the **largest** pairwise distance between two observations/nodes, each belonging to different clusters.

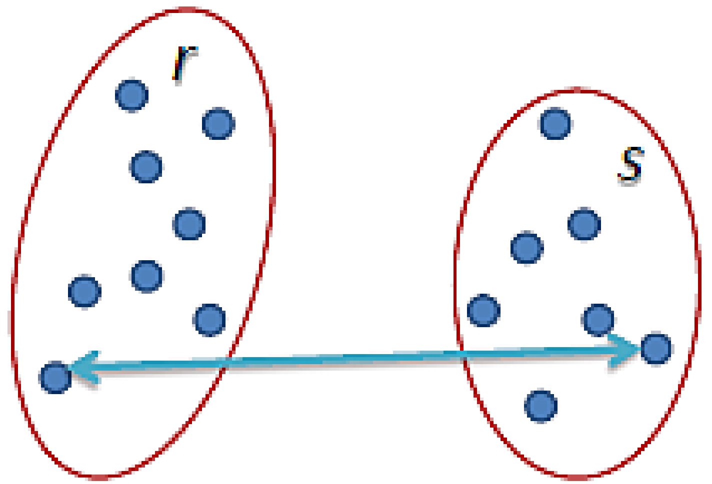
  - *Mean or average linkage clustering:* Distance between two clusters is the **average** of all the pairwise distances, each node/observation belonging to different clusters.

  - *Centroid linkage clustering:* Distance between two clusters is the **distance between their centroids**.

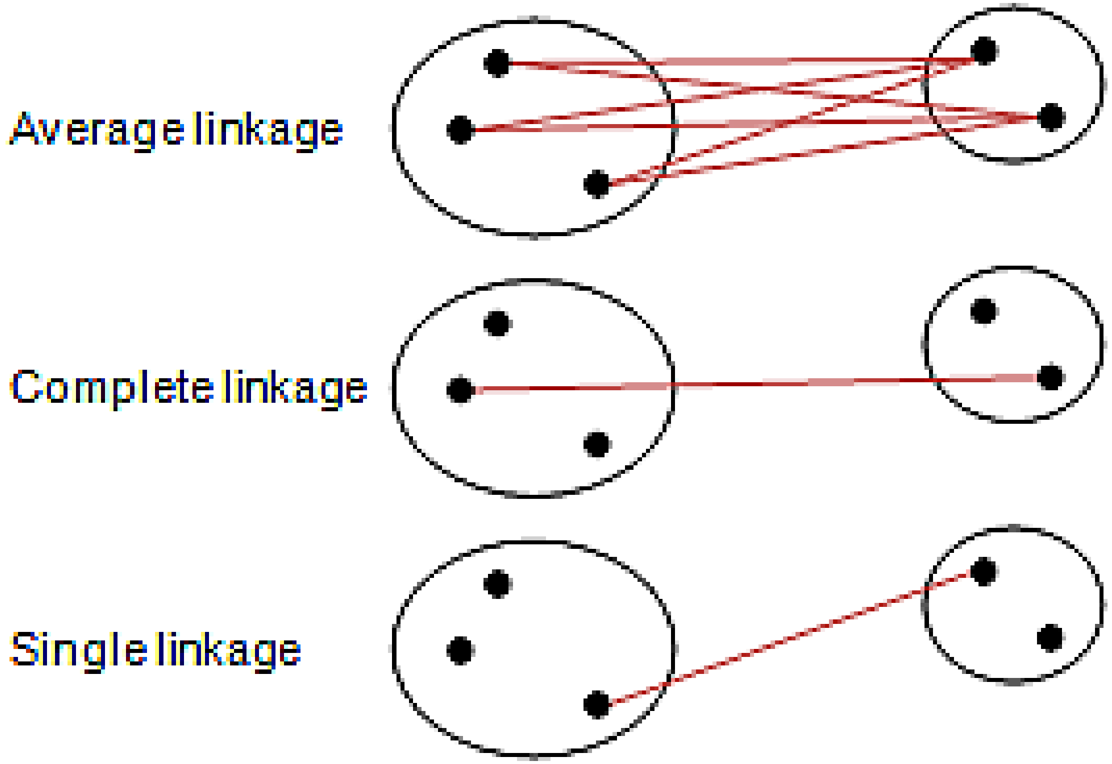
Maximum or complete-linkage clustering	$\max_{a\in A,\, b\in B} d(a,b)$
Minimum or single-linkage clustering	$\min_{a\in A,\, b\in B} d(a,b)$
Unweighted average linkage clustering (or UPGMA)	$\frac{1}{\lvert A\rvert\cdot\lvert B\rvert}\sum_{a\in A}\sum_{b\in B} d(a,b).$
Weighted average linkage clustering (or WPGMA)	$d(i\cup j,k)=\frac{d(i,k)+d(j,k)}{2}.$
Centroid linkage clustering, or UPGMC	$\Vert\mu_A-\mu_B\Vert^2$ where μ_A and μ_B are the centroids of A resp. B.
Median linkage clustering, or WPGMC	$d(i\cup j,k)=d(m_{i\cup j},m_k)$ where $m_{i\cup j}=\frac{1}{2}(m_i+m_j)$
Versatile linkage clustering[6]	$\sqrt[p]{\frac{1}{\lvert A\rvert\cdot\lvert B\rvert}\sum_{a\in A}\sum_{b\in B} d(a,b)^p}, p\neq 0$
Ward linkage,[7] Minimum Increase of Sum of Squares (MISSQ)[8]	$\frac{\lvert A\rvert\cdot\lvert B\rvert}{\lvert A\cup B\rvert}\Vert\mu_A-\mu_B\Vert^2=\sum_{x\in A\cup B}\Vert x-\mu_{A\cup B}\Vert^2-\sum_{x\in A}\Vert x-\mu_A\Vert^2-\sum_{x\in B}\Vert x-\mu_B\Vert^2$
Minimum Error Sum of Squares (MNSSQ)[8]	$\sum_{x\in A\cup B}\Vert x-\mu_{A\cup B}\Vert^2$
Minimum Increase in Variance (MIVAR)[8]	$\frac{1}{\lvert A\cup B\rvert}\sum_{x\in A\cup B}\Vert x-\mu_{A\cup B}\Vert^2-\frac{1}{\lvert A\rvert}\sum_{x\in A}\Vert x-\mu_A\Vert^2-\frac{1}{\lvert B\rvert}\sum_{x\in B}\Vert x-\mu_B\Vert^2$ $=\mathrm{Var}(A\cup B)-\mathrm{Var}(A)-\mathrm{Var}(AB)$
Minimum Variance (MNVAR)[8]	$\frac{1}{\lvert A\cup B\rvert}\sum_{x\in A\cup B}\Vert x-\mu_{A\cup B}\Vert^2=\mathrm{Var}(A\cup B)$
Mini-Max linkage[9]	$\min_{x\in A\cup B}\max_{y\in A\cup B} d(x,y)$
Hausdorff linkage[10]	$\max_{x\in A\cup B}\min_{y\in A\cup B} d(x,y)$
Minimum Sum Medoid linkage[11]	$\min_{m\in A\cup B}\sum_{y\in A\cup B} d(m,y)$ such that m is the medoid of the resulting cluster
Minimum Sum Increase Medoid linkage[11]	$\min_{m\in A\cup B}\sum_{y\in A\cup B} d(m,y)-\min_{m\in A}\sum_{y\in A} d(m,y)-\min_{m\in B}\sum_{y\in B} d(m,y)$
Medoid linkage[12][13]	$d(m_A,m_B)$ where m_A, m_B are the medoids of the previous clusters
Minimum energy clustering	$\frac{2}{nm}\sum_{i,j=1}^{n,m}\Vert a_i-b_j\Vert_2-\frac{1}{n^2}\sum_{i,j=1}^{n}\Vert a_i-a_j\Vert_2-\frac{1}{m^2}\sum_{i,j=1}^{m}\Vert b_i-b_j\Vert_2$


Cluster B
Cluster A
Single Link
Complete Link



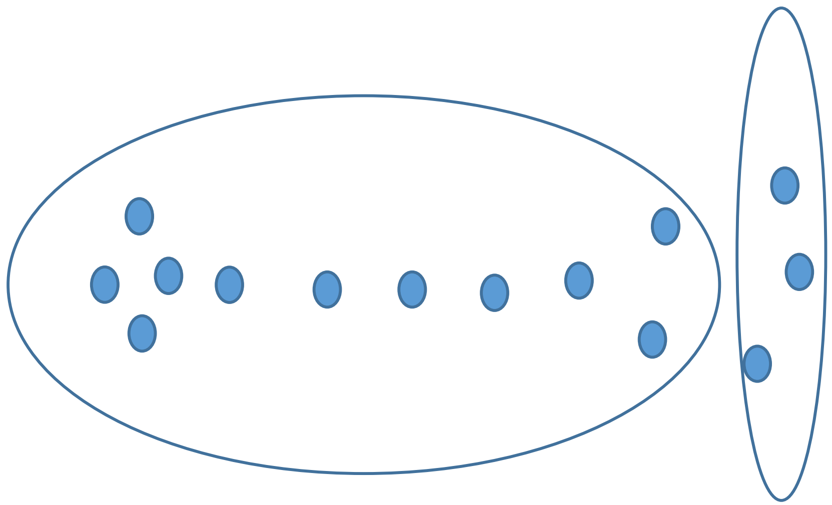


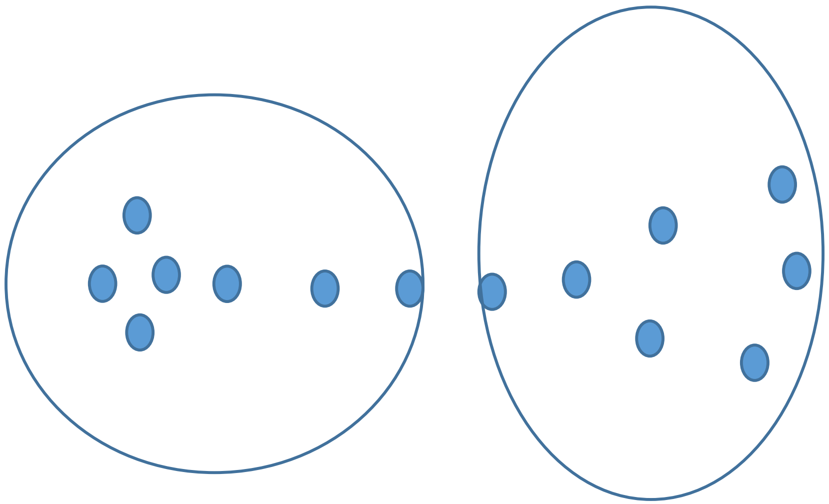
$$L(r,s) = \max(D(x_{ri}, x_{sj}))$$



Single Linkage vs. Complete Linkage

Single linkage

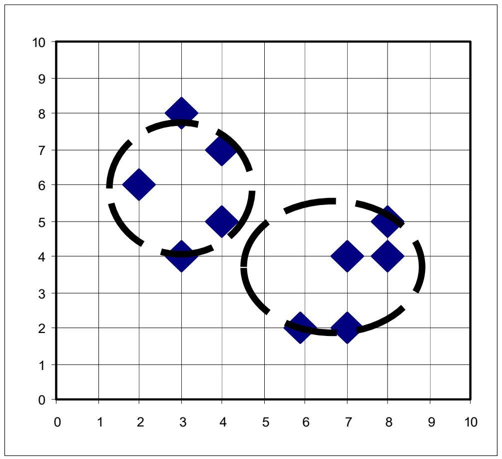




Complete linkage: Minimizes the diameter of the new cluster

Divisive Clustering

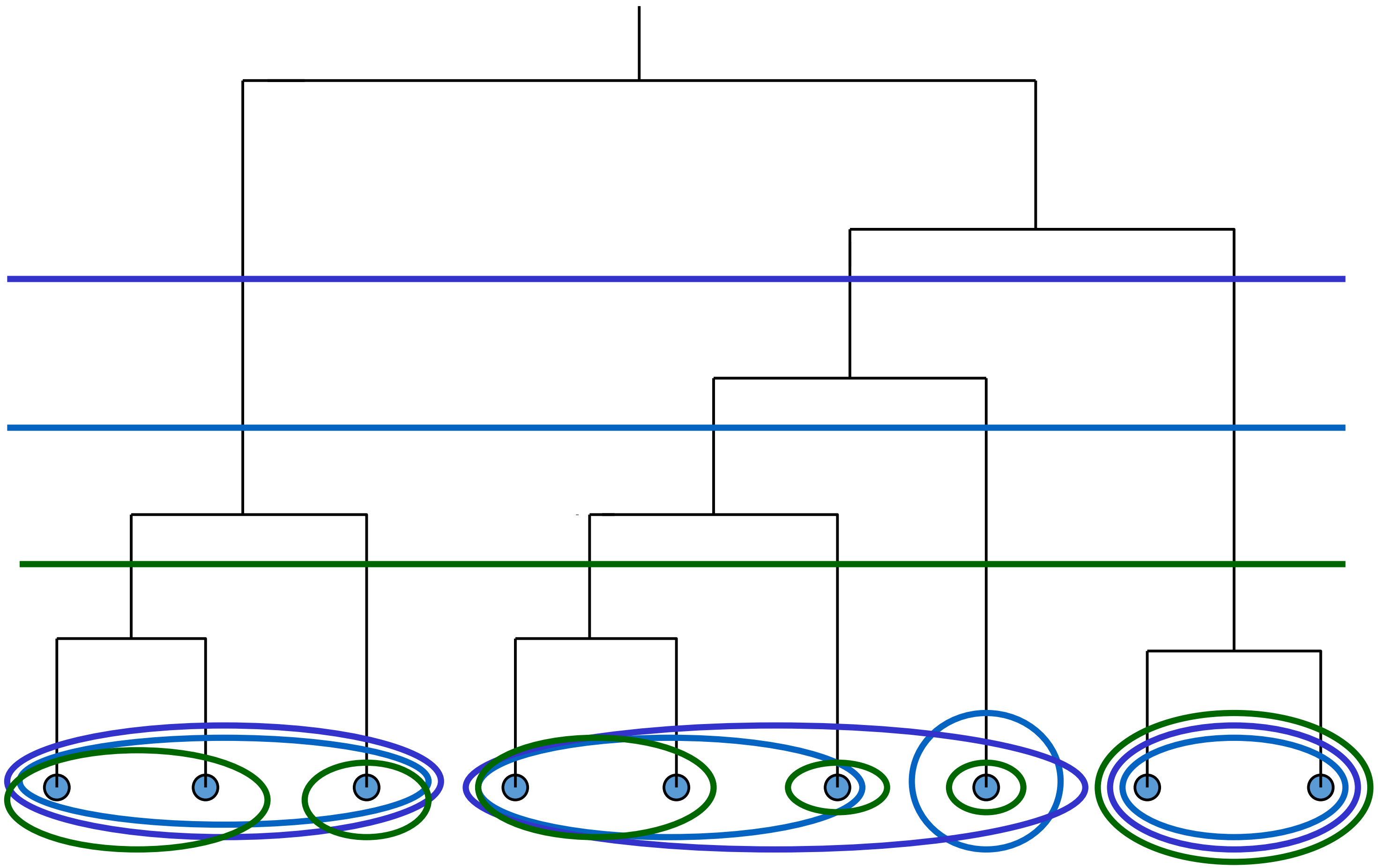
- Initially, all data is in the same cluster
- The largest cluster is split until every object is separate.

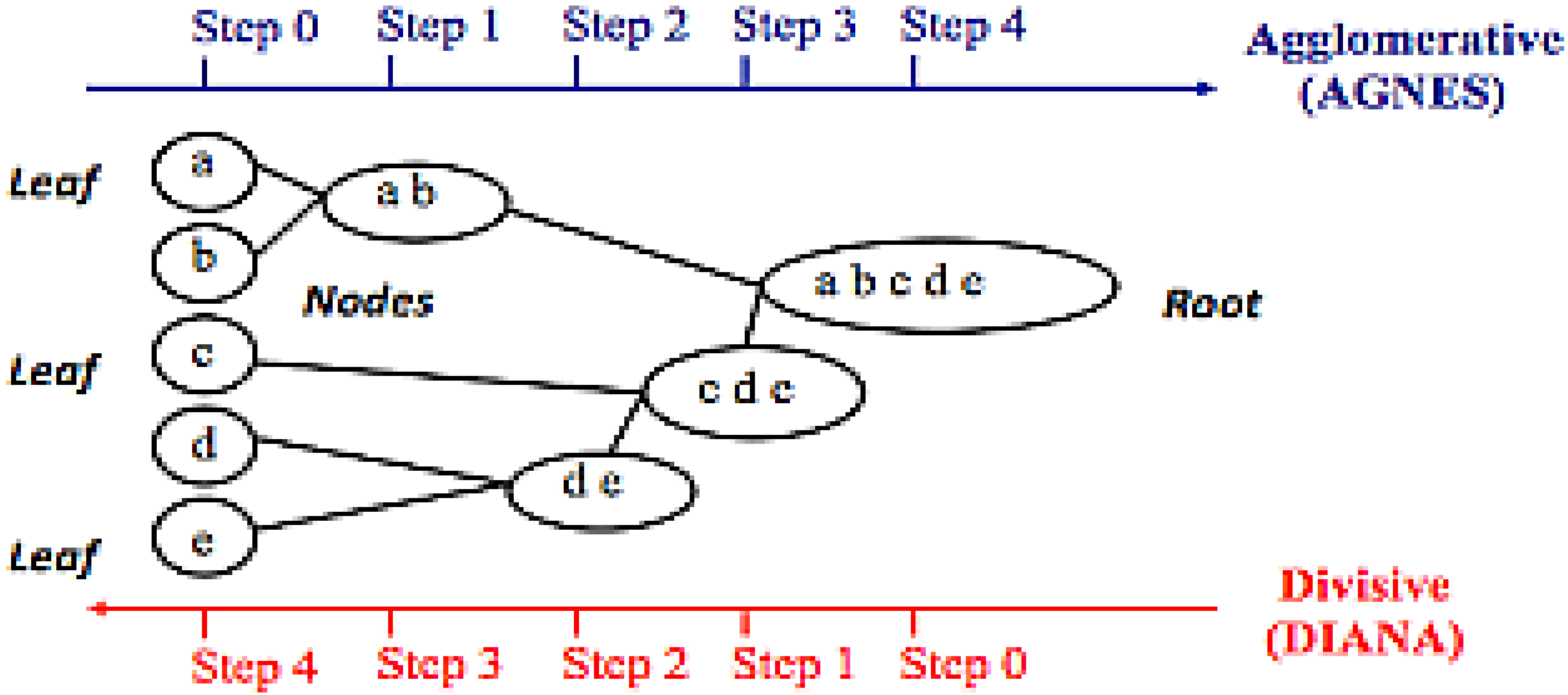


What are the true number of clusters?

- Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram.

- A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.





Use a pair of objects as seeds for the new bipartition. Choice was to select the two objects that are most dissimilar, and then to build up the two subclusters according to distances (or function of distances) to these seeds.

Another divisive algorithm was based on the usual k-means method for partitioning a set of objects. Called the *Bisecting k-means* (Steinbach et al. 2000) - this procedure builds up the successive dichotomies by a *2-means algorithm* with either a random initial partition or with any partition procedure.

Heard of ***BSP***, *Girvan–Newman* algorithm for graph partitioning?

Bisecting k-means Algorithms

Bisecting k-Means is like a combination of k-Means and hierarchical clustering. Instead of partitioning the data into 'k' clusters in each iteration, Bisecting k-means splits one cluster into two sub clusters at each bisecting step (by using k-means) until k clusters are obtained.

Basic Bisecting K-means Algorithm for finding K-Clusters

1. Pick a cluster to split.
2. Find 2 sub-clusters using the basic K-means algorithm. (Bisecting step)
3. Repeat step 2, the bisecting step, for ITER times and take the split that produces the clustering with the highest overall similarity.
4. Repeat steps 1, 2 and 3 until the desired number of clusters is reached.

Bisecting K-means Algorithm follows

Divisive hierarchical clustering method using K means

```
For i=1 to k-1 do {
     Pick a leaf cluster C to split
       For j=1 to Iteration do
       {
               Use K-Means split to two sub clusters C1 and C2
             Choose the best of the above splits and make it permanent
     }
}
```

DBSCAN : Density Based Spatial Clustering of Applications with Noise

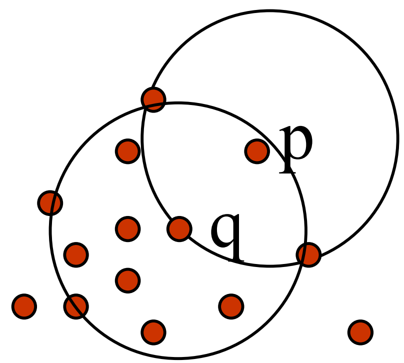
Density-Based Clustering Methods

- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - Need density parameters as termination condition
- Several interesting studies:
  - <u>DBSCAN:</u> Ester, et al. (KDD'96)
  - <u>OPTICS</u>: Ankerst, et al (SIGMOD'99).
  - <u>DENCLUE</u>: Hinneburg & D. Keim (KDD'98)
  - <u>CLIQUE</u>: Agrawal, et al. (SIGMOD'98) (more grid-based)

Density-Based Clustering: Basic Concepts

- Two parameters:
  - *Eps*: Maximum radius of the neighborhood
  - *MinPts*: Minimum number of points in an *Eps*-neighborhood 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* w.r.t. *Eps*, *MinPts* if
  - *p* belongs to $N_{Eps}(q)$
  - core point condition:

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

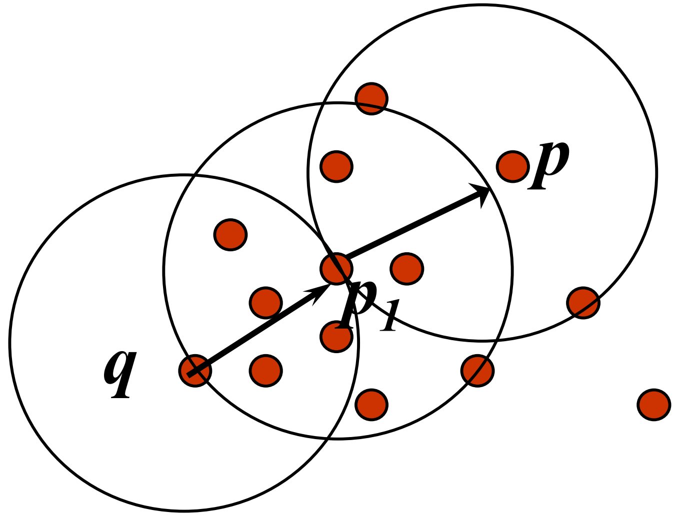


MinPts = 5

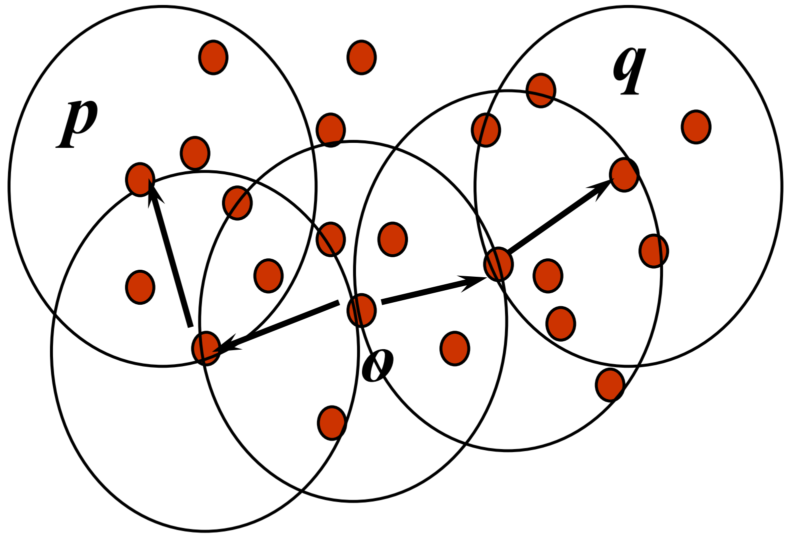
Eps = 1 cm

Density-reachable & Density-connected

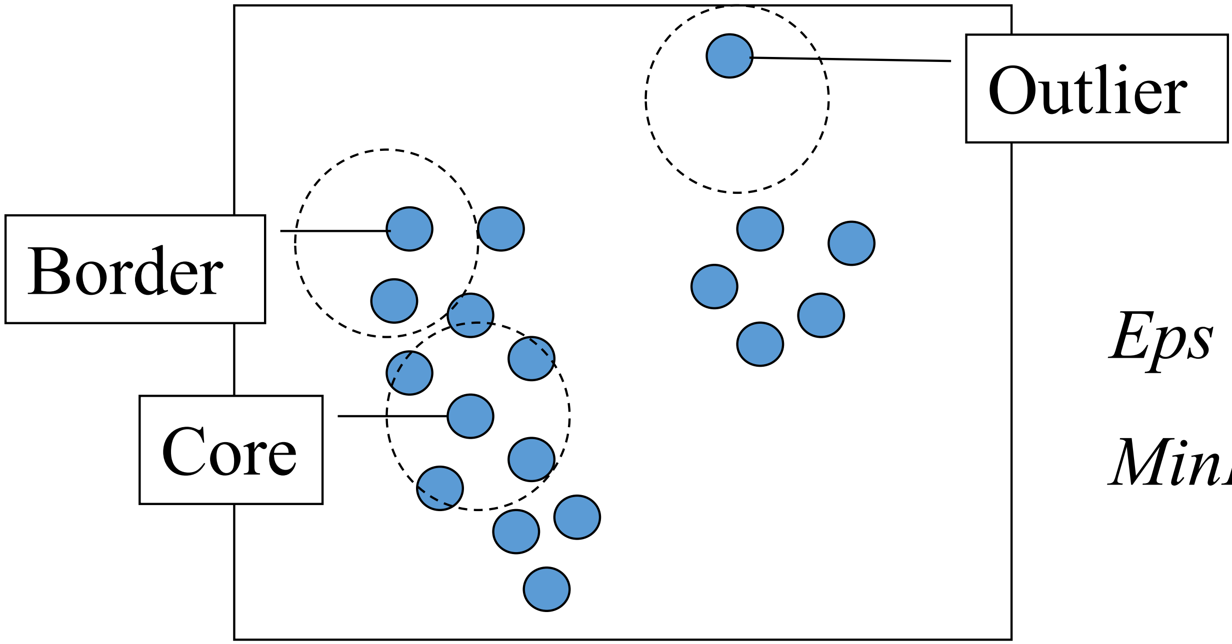
- Density-reachable:
  - A point p is density-reachable from a point q if there is a chain of points $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i
  - This is not symmetric



- Density-connected
  - A point p is density-connected to a point q w.r.t. *Eps*, *MinPts*, if there is a point o such that both p and q are density-reachable from o w.r.t. *Eps* and *MinPts*



DBSCAN



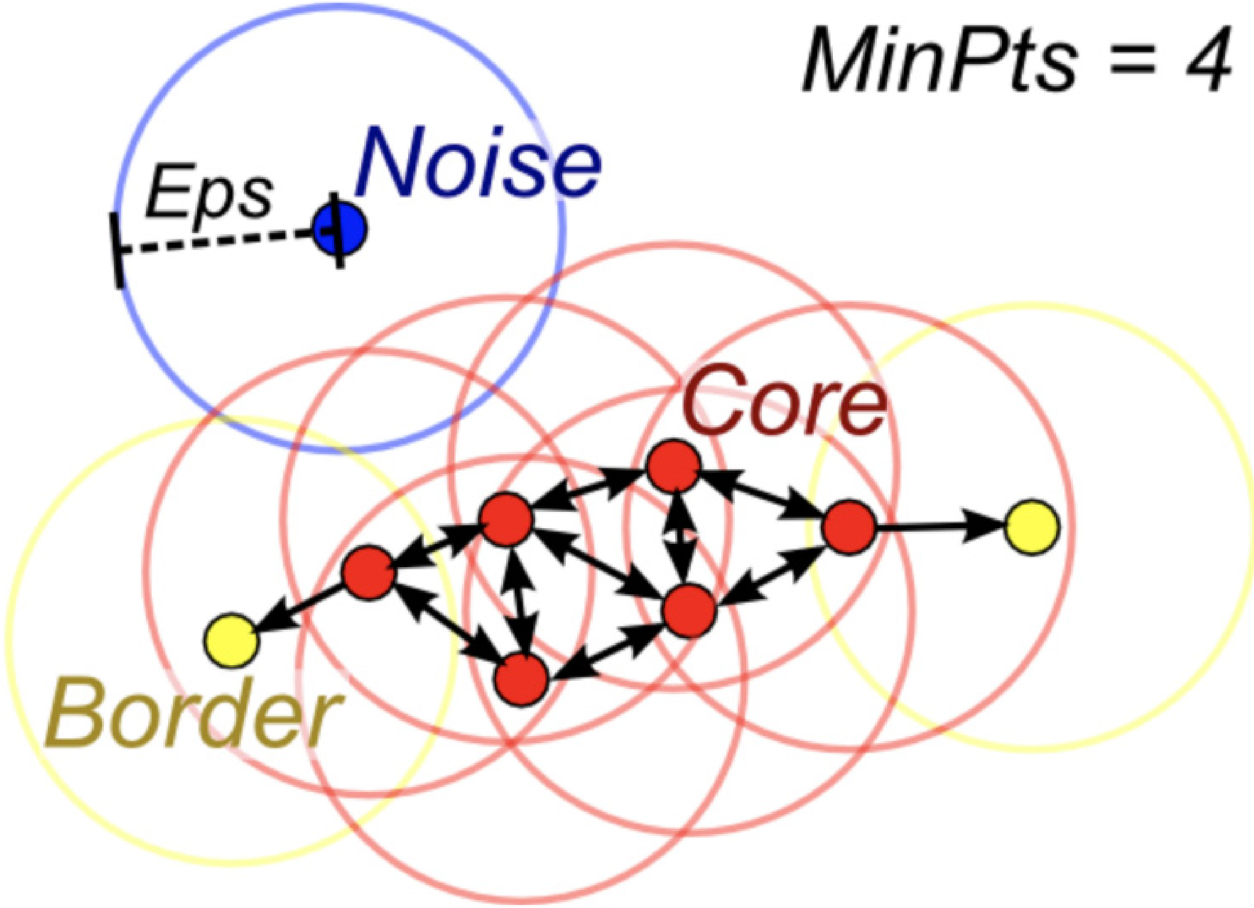
Given *Eps* and *MinPts*, categorize the objects into three exclusive groups.

$Eps = 1\text{cm}$

$MinPts = 5$

- A point is a core point if it has more than a specified number of points (*MinPts*) within *Eps*—These are points that are at the interior of a cluster.
- A border point has fewer than *MinPts* within *Eps*, but is in the neighborhood of a core point.
- A noise point is any point that is not a core point nor a border point.

DBSCAN – Core, border and noise points – Illustration - I

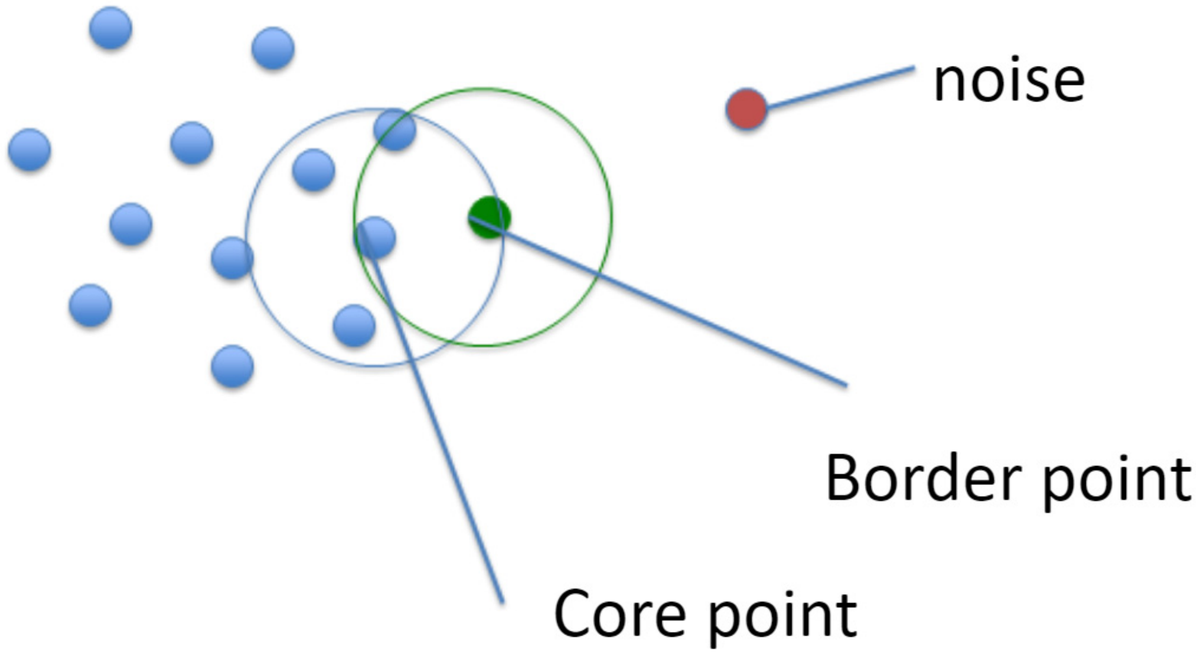


Red: Core Points

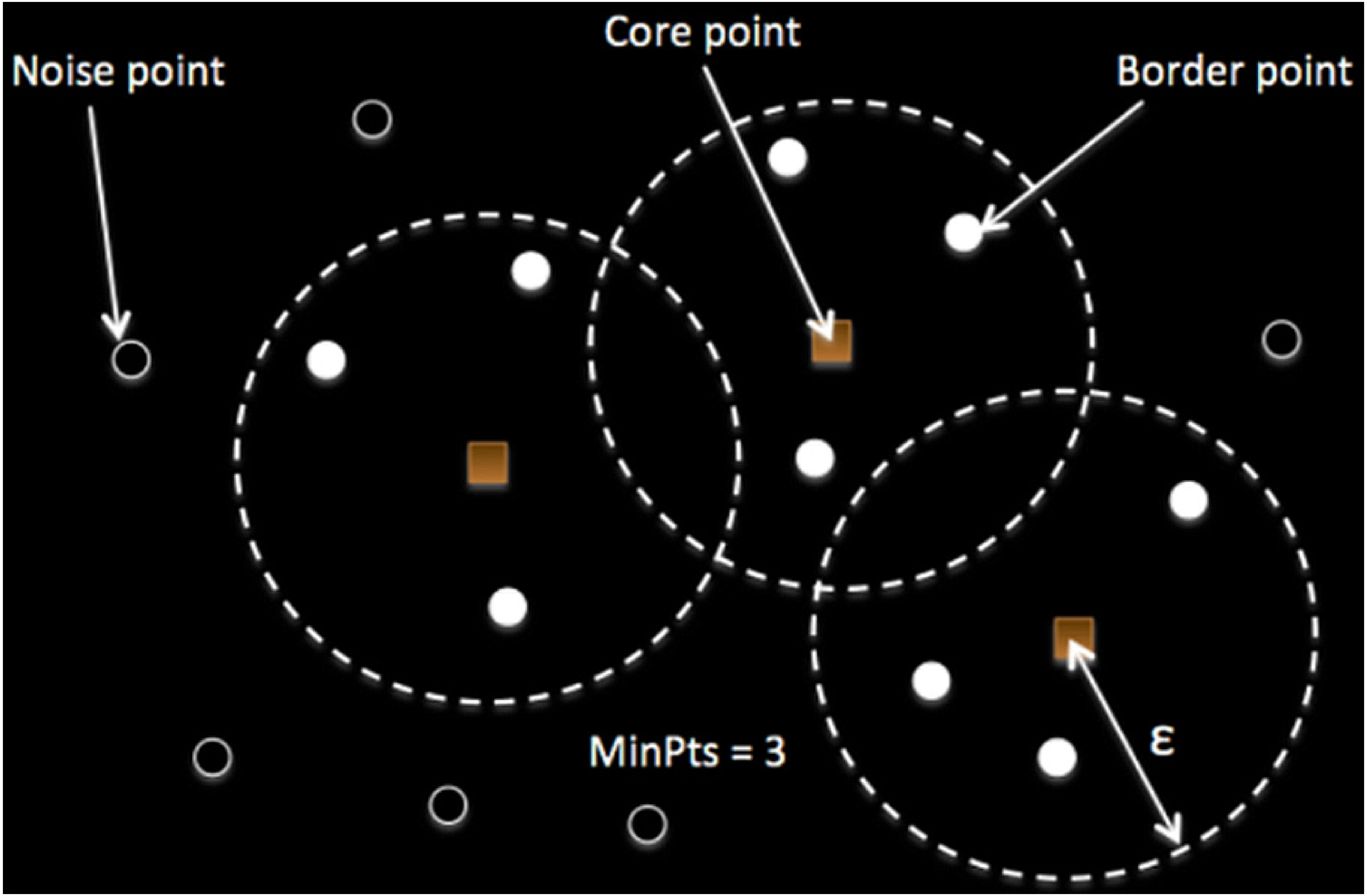
Yellow: Border points. Still part of the cluster because it's within epsilon of a core point, but not does not meet the min_points criteria

Blue: Noise point. Not assigned to a cluster

DBSCAN – Core, border and noise points – Illustration - II



MinPts = 4

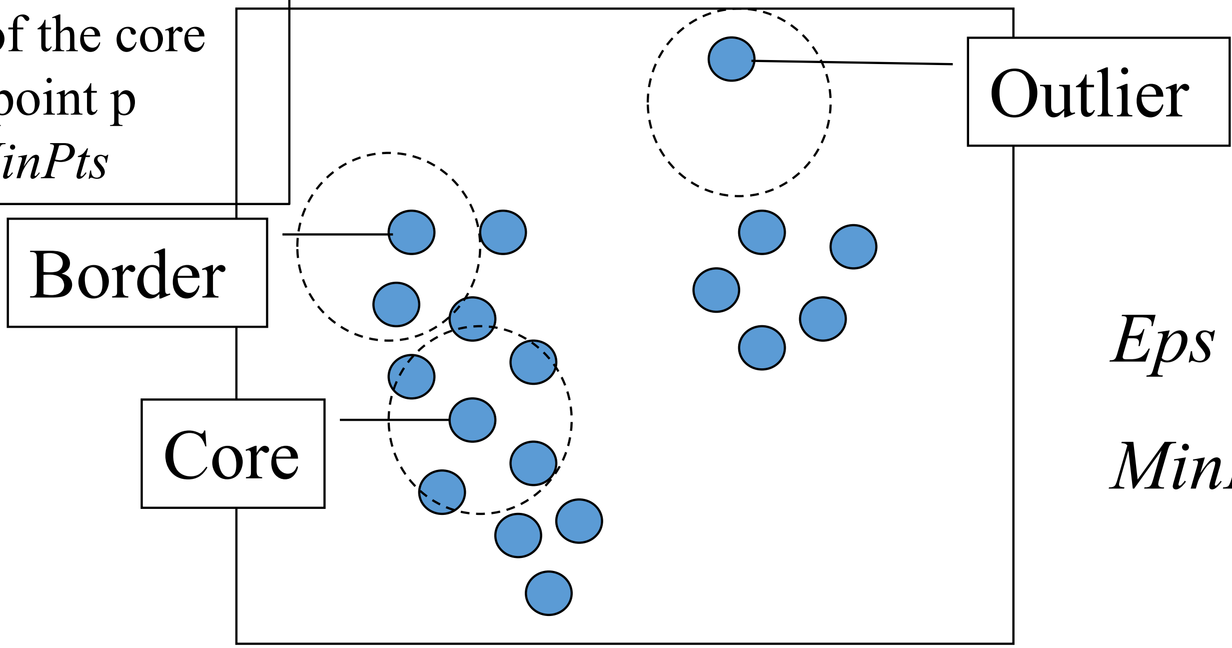

Core point
Noise point
Border point
MinPts = 3
ε

DBSCAN

- A set of points C is a cluster, if
  - For any two points $p, q \in C$, p and q are density-connected
  - There does not exist any pair of points, $p \in C$ and $s \notin C$ such that p and s are density-connected.

Border points are points that are reachable from any of the core points. For a border point p

$$|N_{Eps}(p)| < MinPts$$

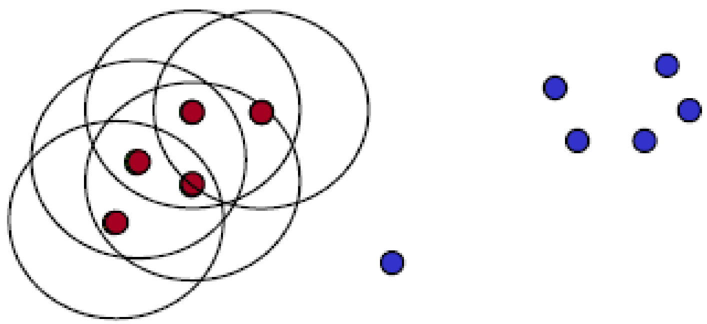


$Eps = 1\text{cm}$

$MinPts = 5$

DBSCAN Algorithm with example

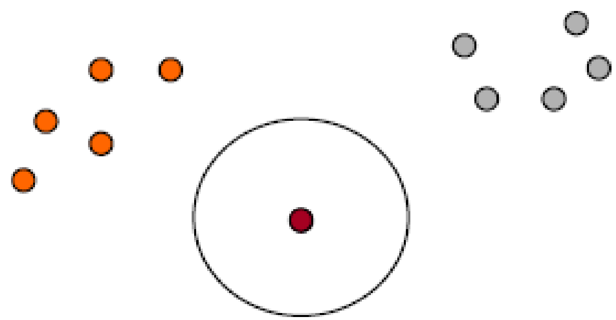
- Parameter: ε = 2, *MinPts* = 3



```
for each o ∈ D do
    if o is not yet classified then
        if o is a core-object then
            collect all objects density-reachable from o
            and assign them to a new cluster.
        else
            assign o to NOISE
```

DBSCAN Algorithm with example

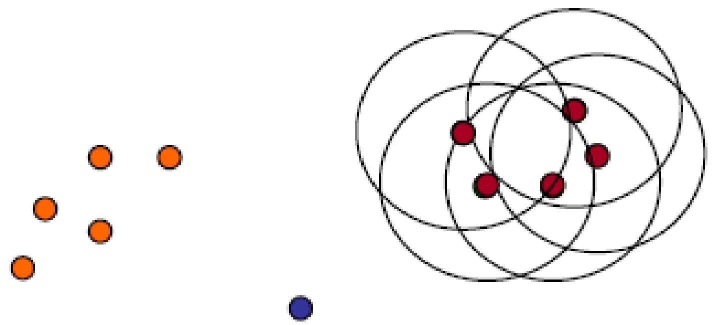
- Parameter: $\varepsilon = 2$, $MinPts = 3$



for each $o \in D$ **do**
    if o is not yet classified **then**
        if o is a core-object **then**
            collect all objects density-reachable from o
            and assign them to a new cluster.
        else
            assign o to NOISE

DBSCAN Algorithm with example

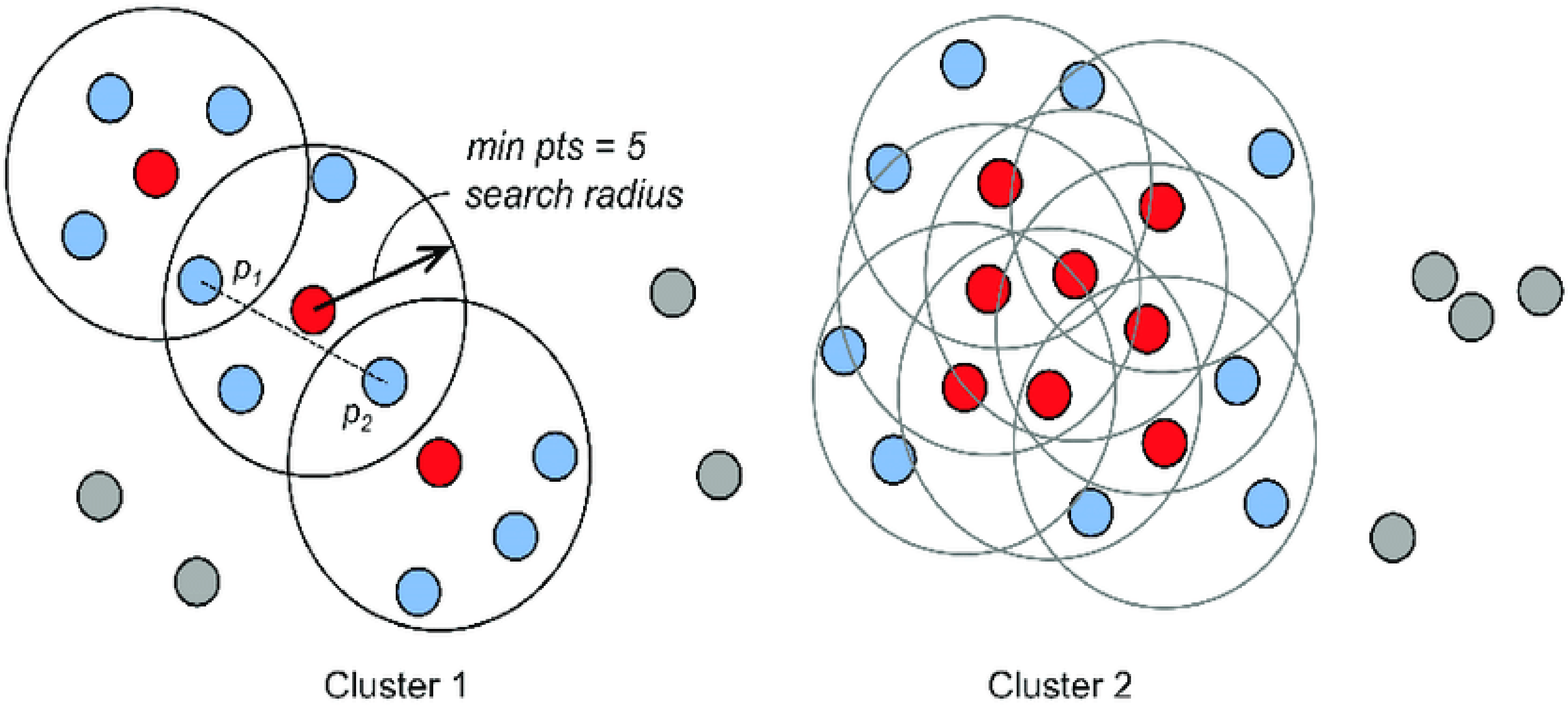
- Parameter: ε = 2, *MinPts* = 3



for each $o \in D$ **do**
  if o is not yet classified **then**
    if o is a core-object **then**
      collect all objects density-reachable from o
      and assign them to a new cluster.
    else
      assign o to NOISE

Algorithm

- **Select a point *p***
- **Retrieve all points directly density-reachable from *p* wrt. *Eps* and *MinPts*.**
- **If *p* is a not a core point, *p* is marked as noise**
- **Else a cluster is initiated.**
  - ***p* is marked as classified with a cluster ID**
  - ***seedSet* = all directly reachable points from *p.***
  - **For each point p_i in *seedSet* till it is empty**
    - **If p_i is a noise point, assign p_i to the current cluster ID**
    - **If p_i is unclassified, identify if it is a core point. If yes, then add all directly reachable point to seed set and add p_i to cluster ID**
    - **Delete p_i from *seedSet***


min pts = 5
search radius
p1
p2
Cluster 1
Cluster 2

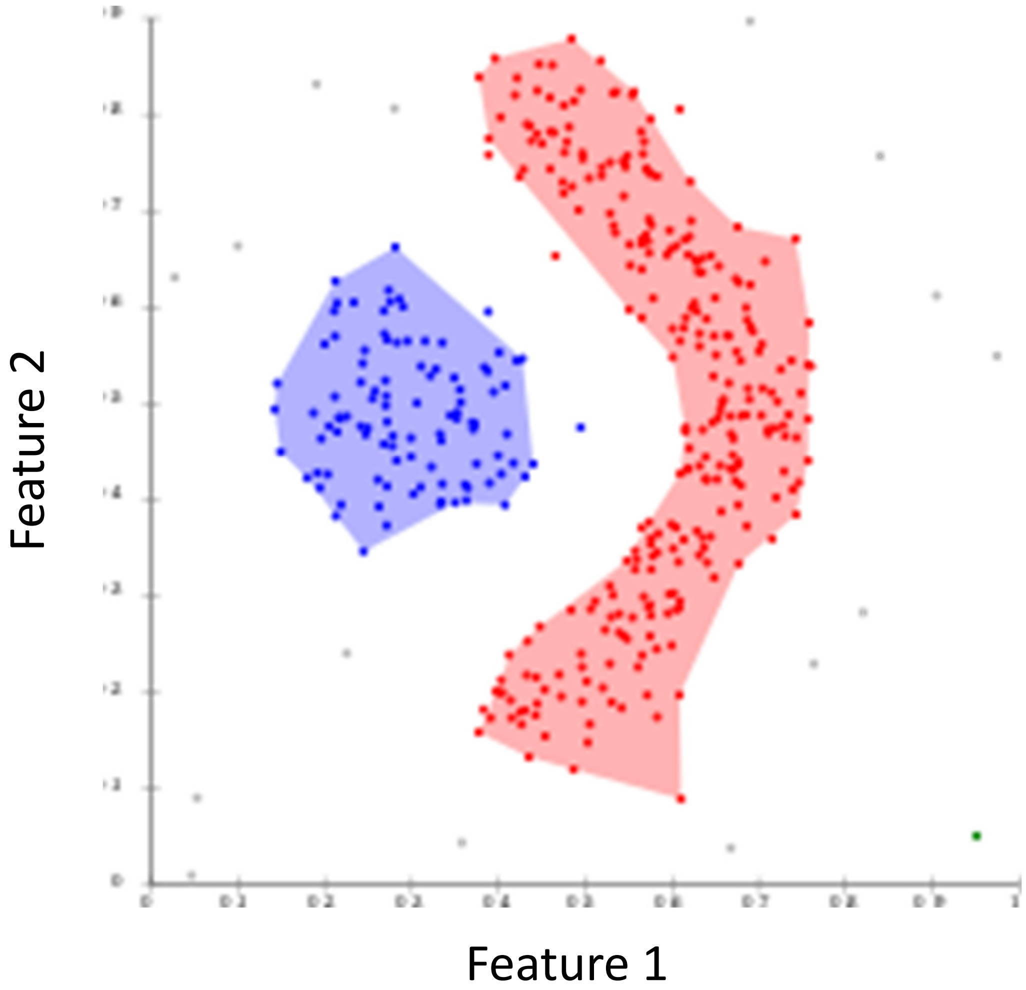
DBSCAN: Properties

- Can discover clusters of arbitrary shapes
- Complexity
  - Time
    - $O(n^2)$
    - $O(n\log^{d-1}n)$ with range tree. But requires more storage
      - d dimensions
- Weakness:
  - Parameter sensitive

Apply PCA to reduce dimension, and then use DBSCAN

Algorithm Name	Merits
DBSCAN	Finding clusters of arbitrary shapes and handling noise effectively.
VDBSCAN	Identify clusters with varied density.
LDBSCAN	Finding similar local density clusters and detects noises effectively.
ST-DBSCAN	Discovers clusters on spatial-temporal data.
DVBSCAN	Finding clusters of varied density.
DBSCAN-DLP	Discovers clusters of different densities.
PACA-DBSCAN	Discovers clusters from multidimensional datasets.
DMDBSCAN	Discovers clusters of different density levels.
MDBSCAN	Discovers natural clusters based on MST objective function.
MR-DBSCAN	Finding clusters on heavily skewed data.

DBSCAN - non-linearly separable clusters



How to pick the initial centroids?

I'll Choose

Randomly

Farthest Point

What kind of data would you lik

Uniform Points

Gaussian Mixture

Smiley Face

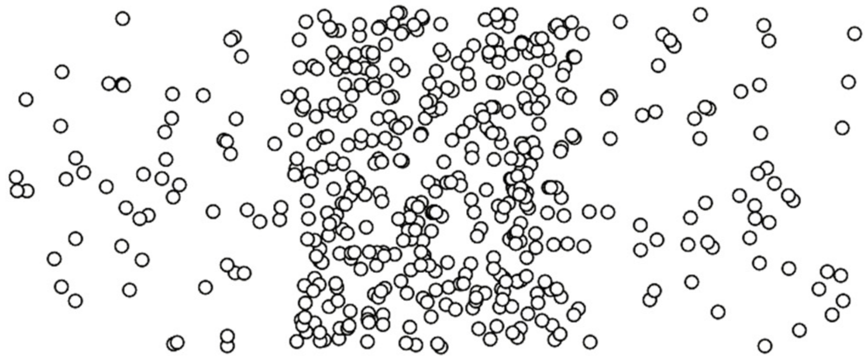
Density Bars

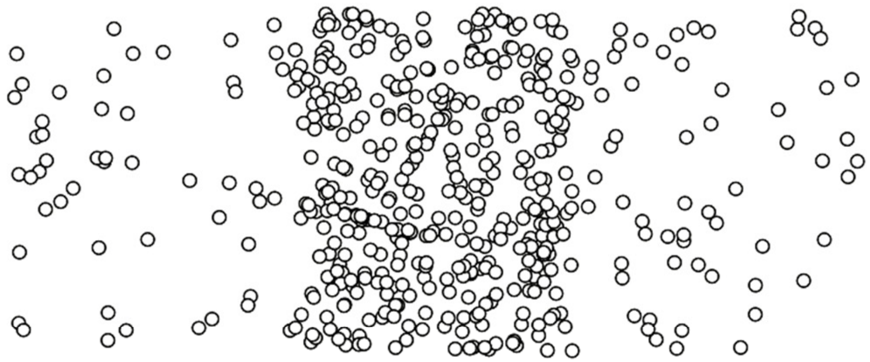
Packed Circles

Pimpled Smil

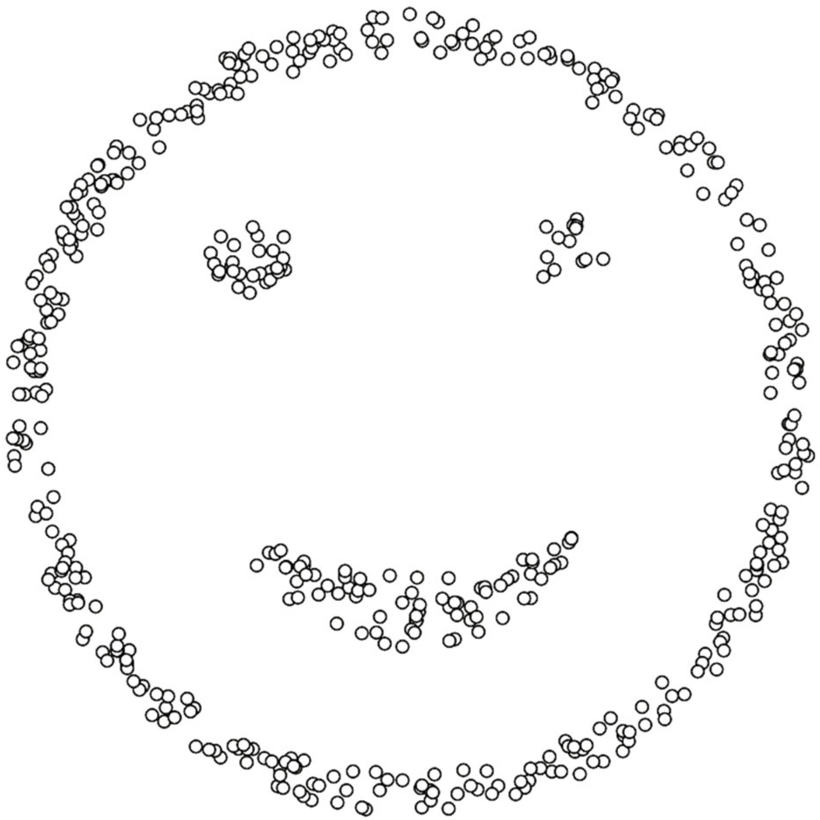
DBSCAN Rings

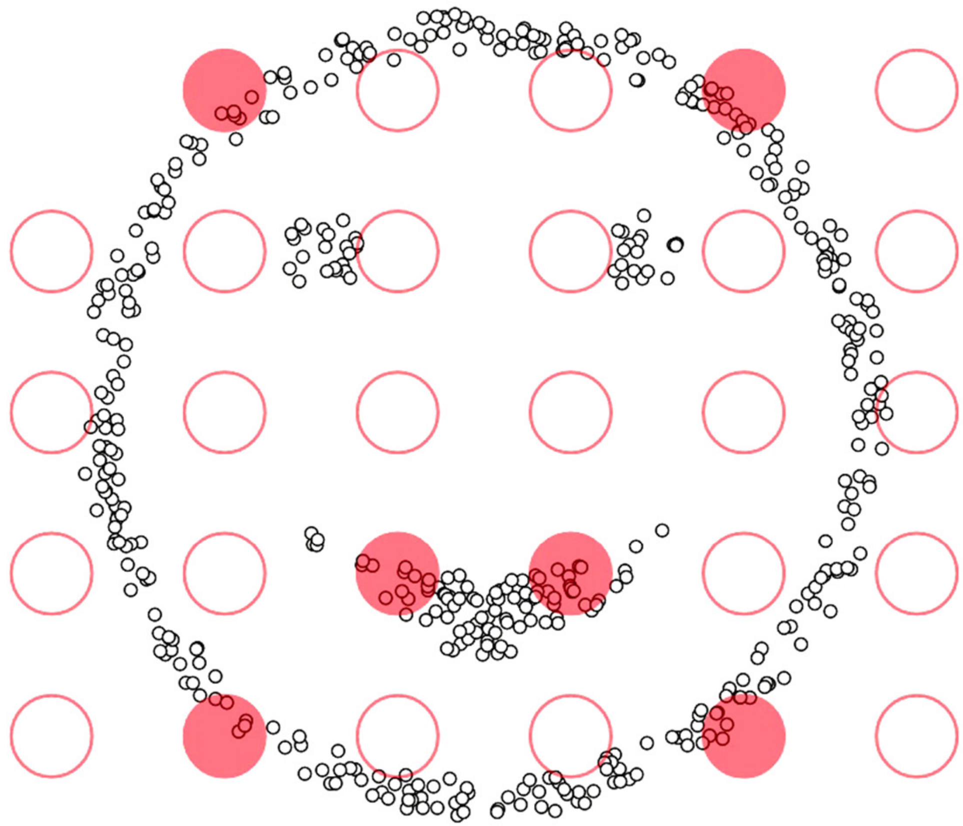
Example A



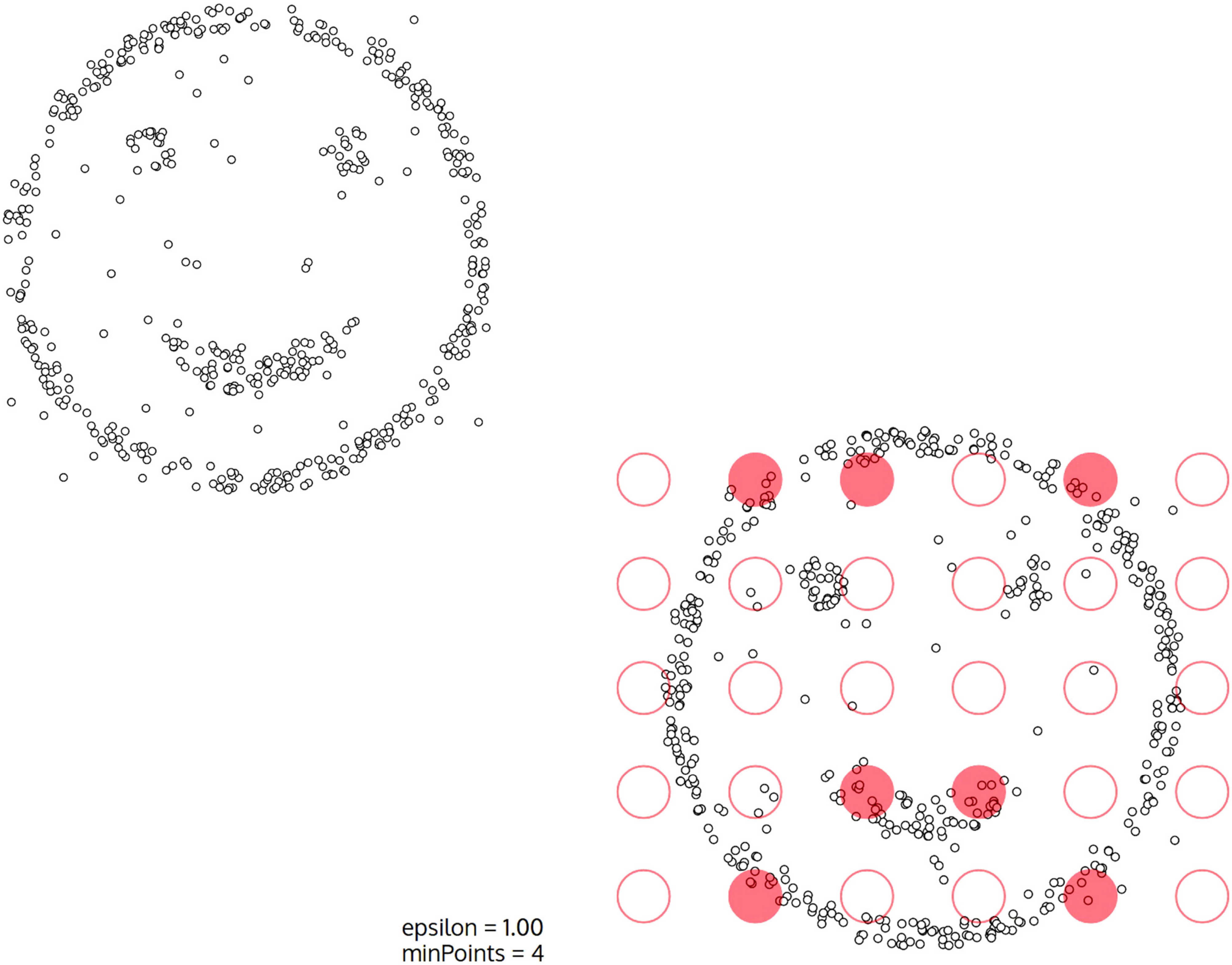


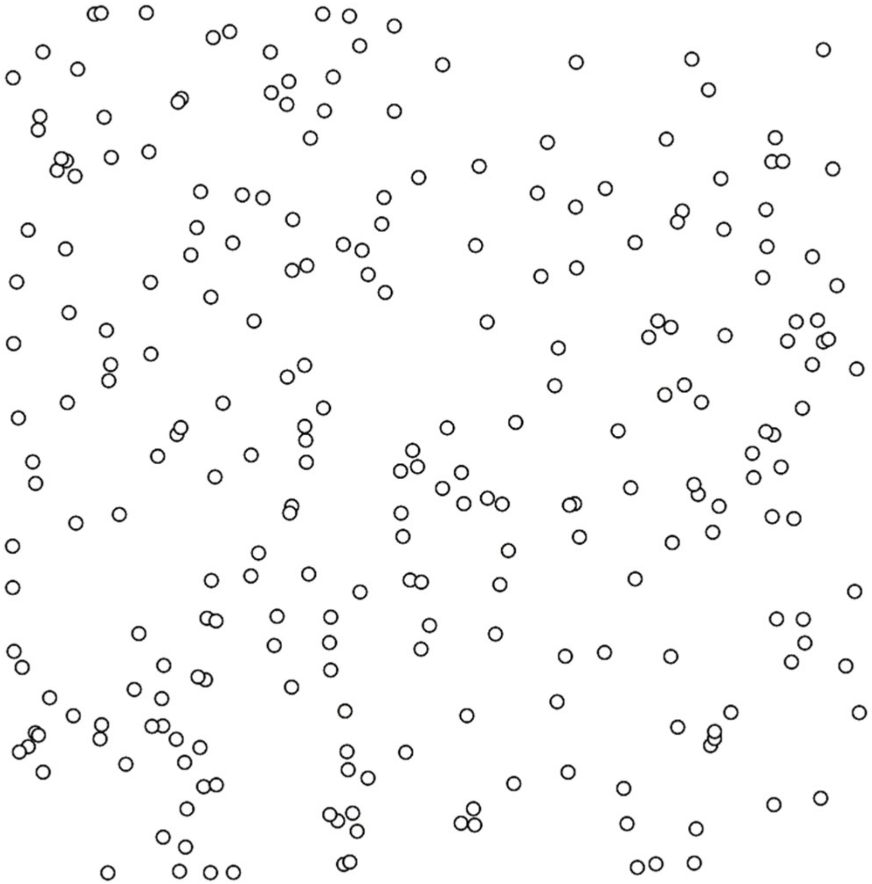
epsilon = 1.00
minPoints = 4





epsilon = 1.00
minPoints = 4


epsilon = 1.00
minPoints = 4



at kind of data would you like?

Uniform Points

Gaussian Mixture

Smiley Face

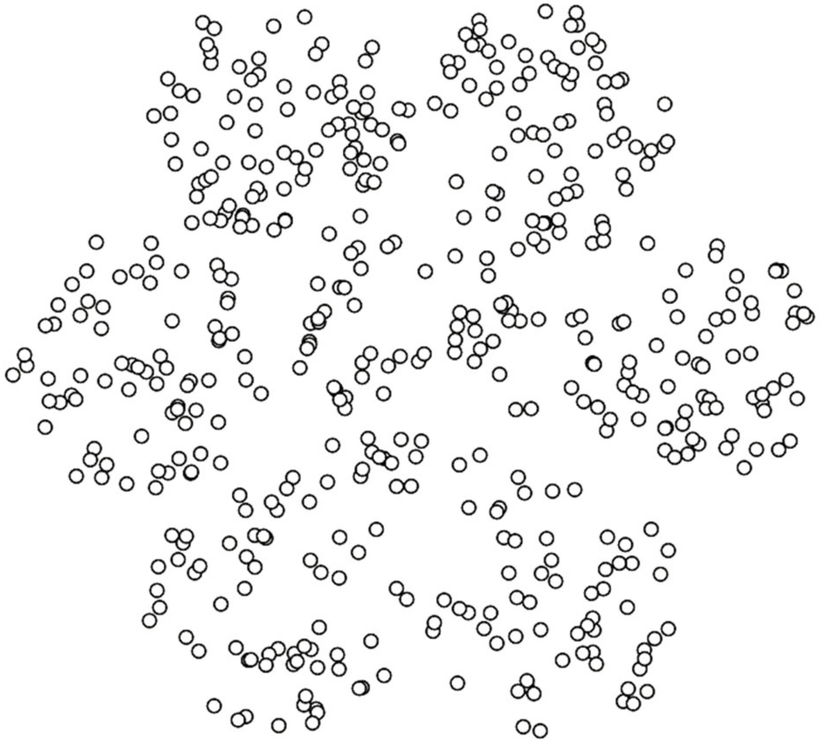
Density Bars

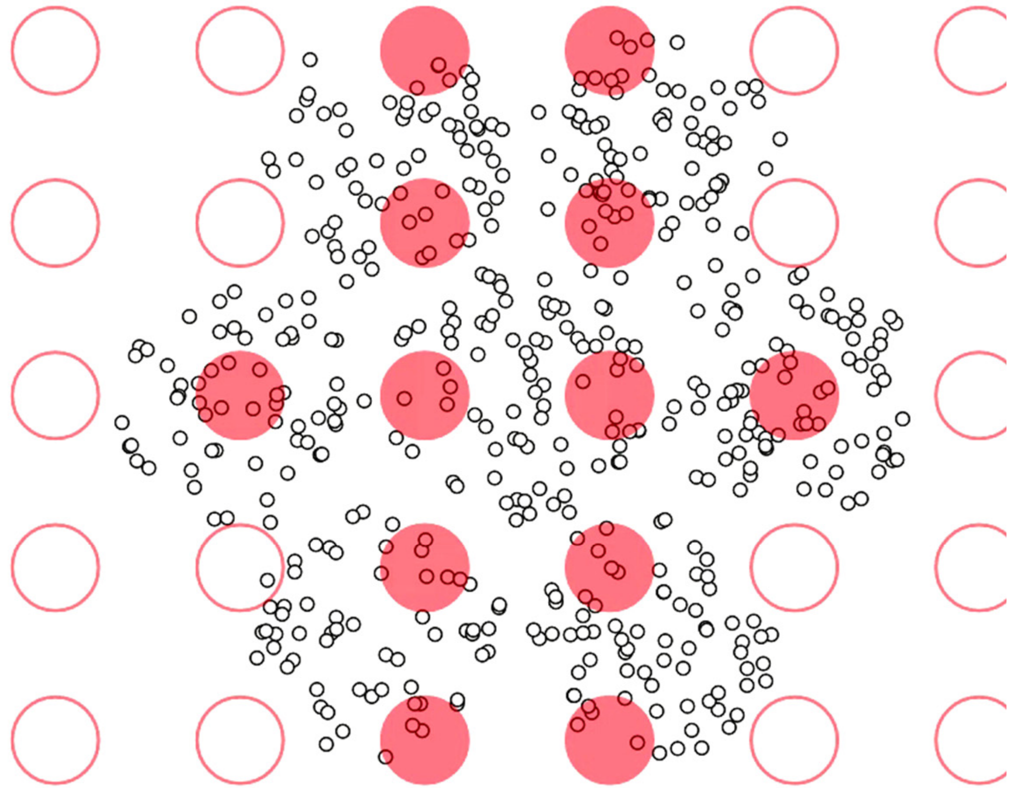
Packed Circles

Pimpled Smiley

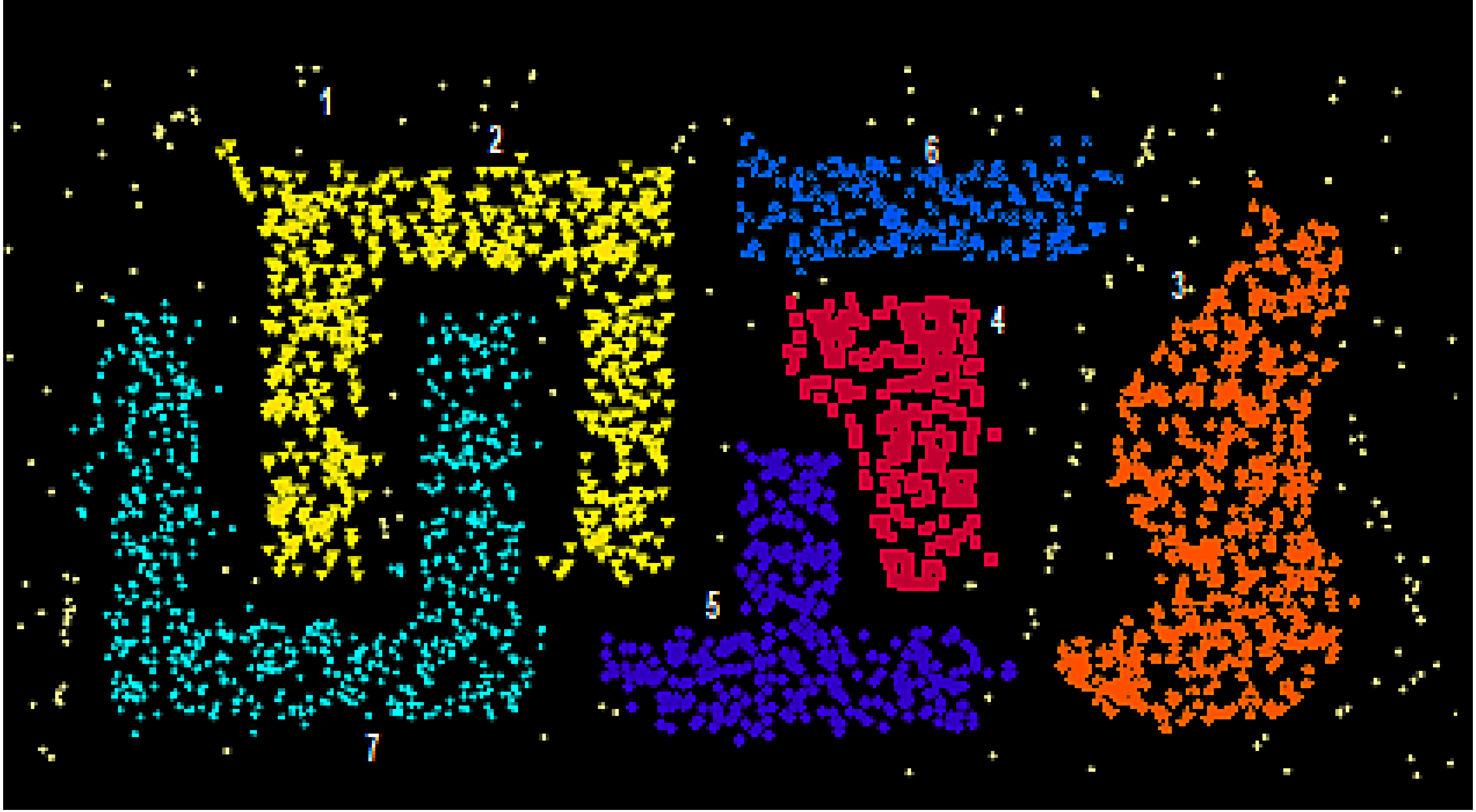
DBSCAN Rings

Example A



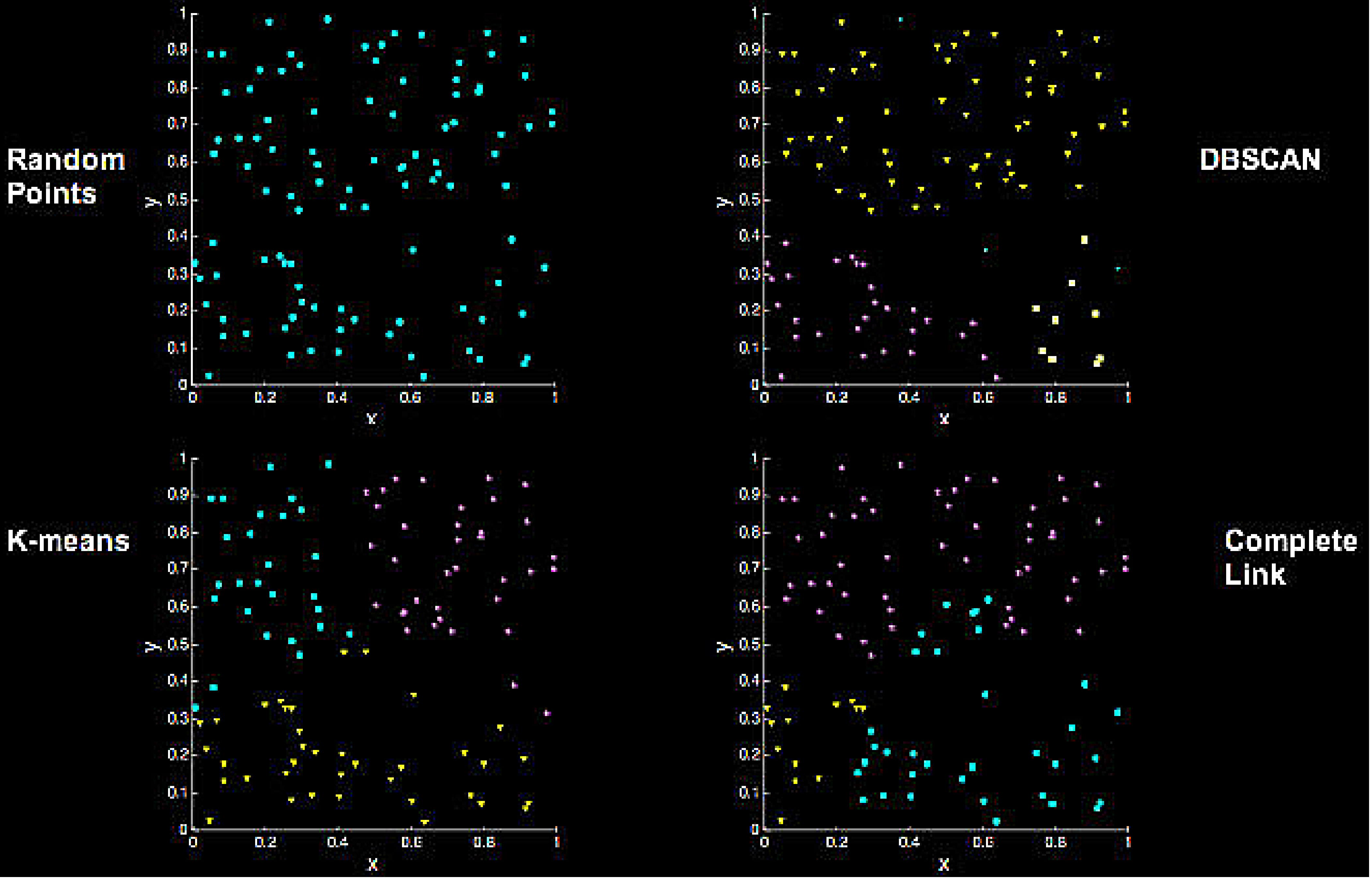


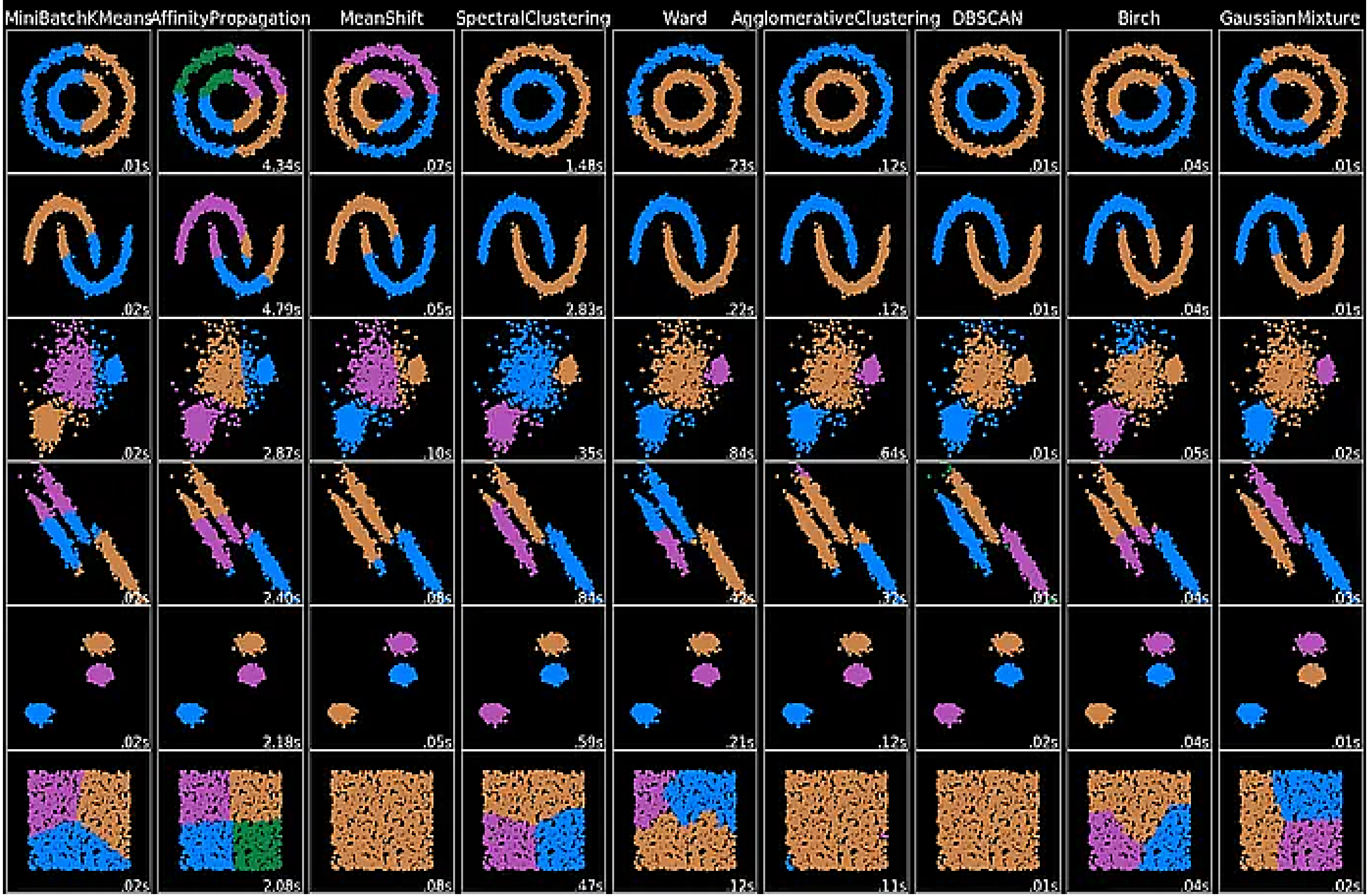
epsilon = 1.00
minPoints = 4

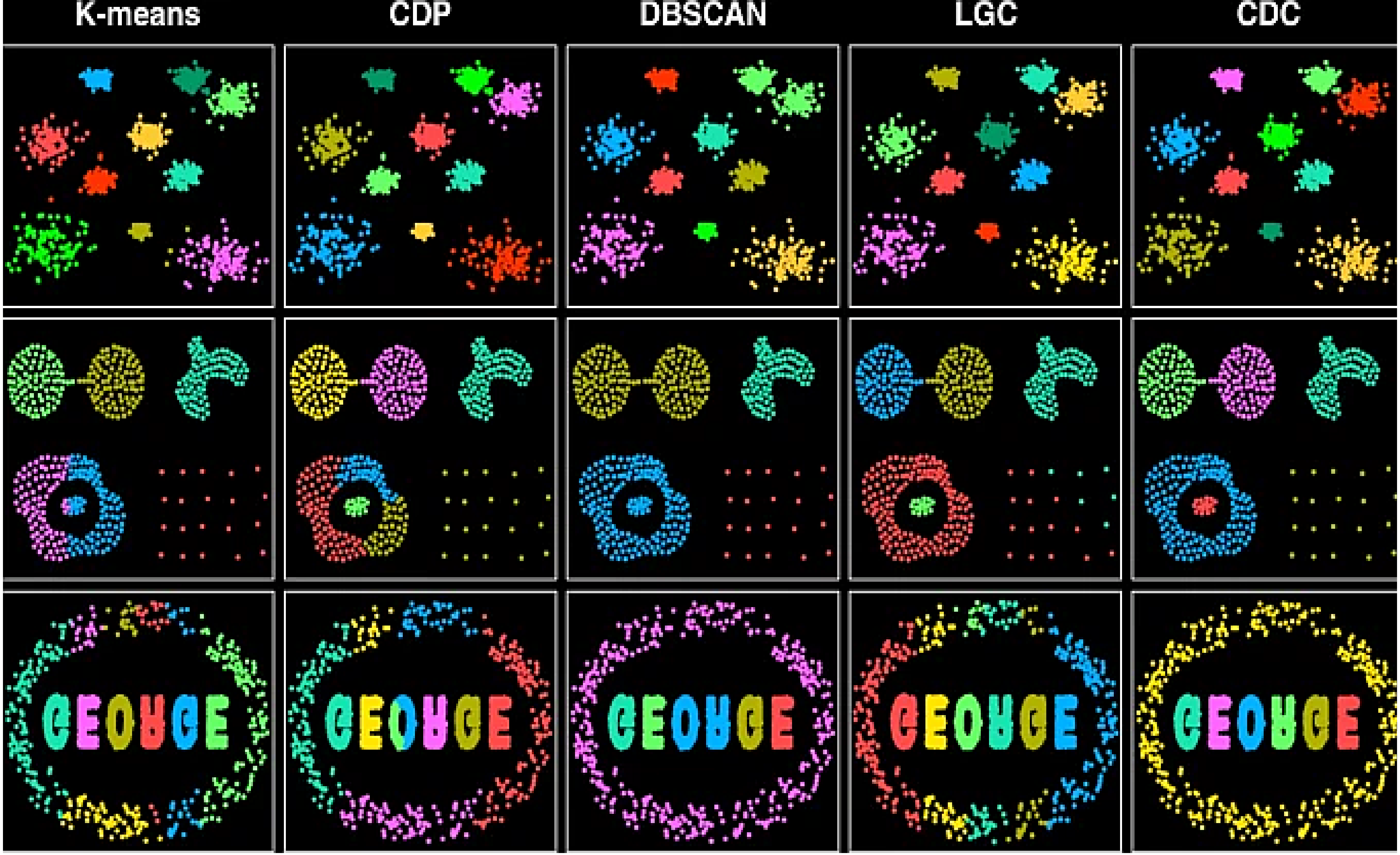


DBSCAN

Clusters are natural in purely random data



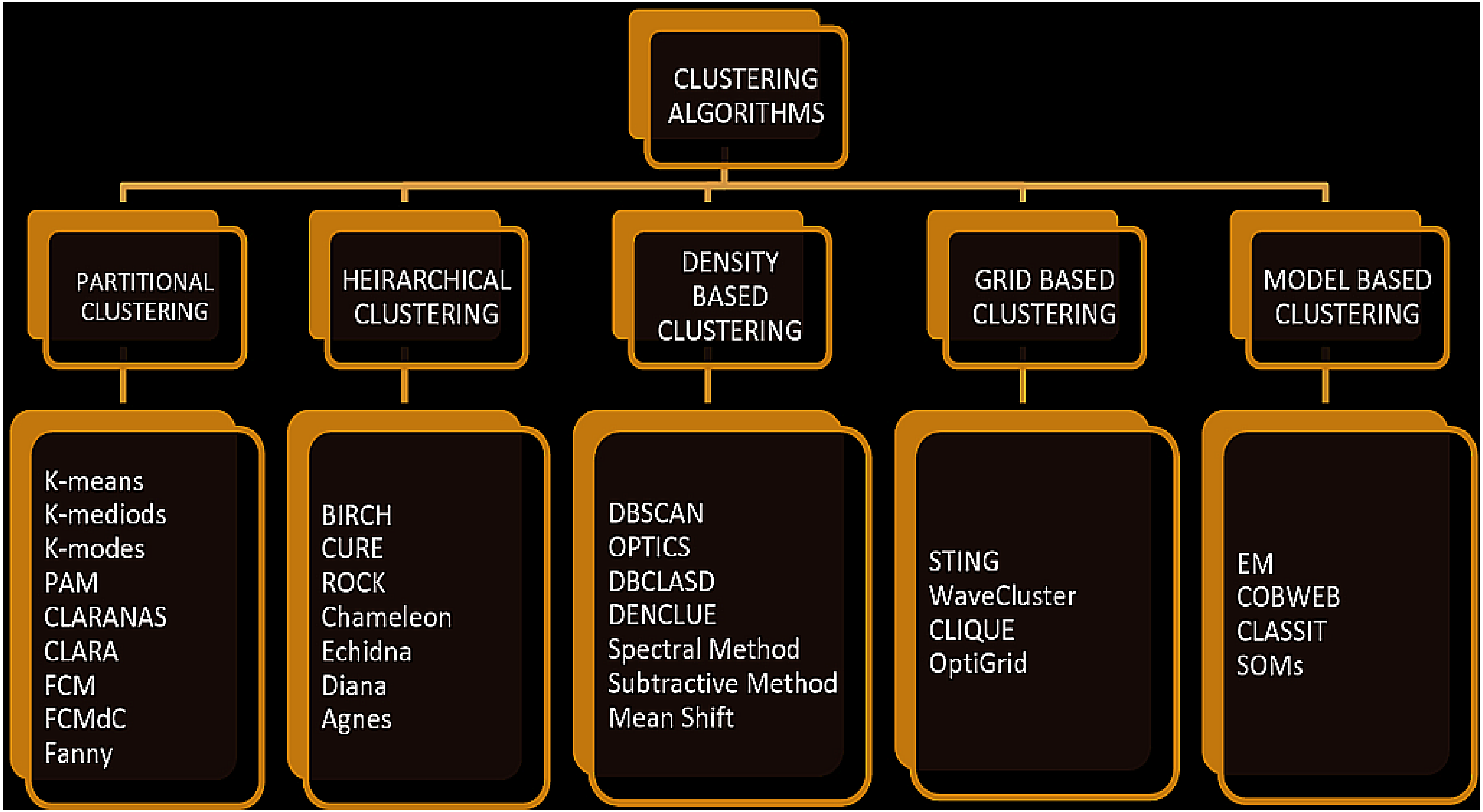

MiniBatchKMeans
AffinityPropagation
MeanShift
SpectralClustering
Ward
AgglomerativeClustering
DBSCAN
Birch
GaussianMixture
.01s
4.34s
.07s
1.48s
.23s
.12s
.01s
.04s
.01s
.02s
4.79s
.05s
2.83s
.22s
.12s
.01s
.04s
.01s
.02s
2.87s
.10s
.35s
.84s
.64s
.01s
.05s
.02s
.02s
2.40s
.08s
.84s
.42s
.37s
.01s
.04s
.03s
.02s
2.18s
.05s
.59s
.21s
.12s
.02s
.04s
.01s
.02s
2.08s
.08s
.47s
.12s
.11s
.01s
.04s
.02s



Clustering by finding Density Peaks (CDP); Science 2014;

Clustering by measuring local direction centrality for data (CDC); Nature Communications, 2022.

Local Gravitation Clustering (LGC) - 2002



Others – Agglomerative; GMM,

CLUSTER QUALITY – VALIDITY INDEX

- **Silhouette Score**

(nc-ic)/max(ic,nc)

where,

ic = mean of the intra-cluster distance

nc = mean of the nearest-cluster distance

- **Dunn Index**

$$D = \frac{\min_{1 \leq i < j \leq n} d(i,j)}{\max_{1 \leq k \leq n} d'(k)},$$

where *d*(*i*,*j*) represents the distance between clusters *i* and *j*, and *d* '(*k*) measures the intra-cluster distance of cluster *k*. The inter-cluster distance *d*(*i*,*j*) between two clusters may be any number of distance measures, such as the distance between the centroids of the clusters. Similarly, the intra-cluster distance *d* '(*k*) may be measured in a variety ways, such as the maximal distance between any pair c

- **Davies Bouldin index**

$$DB = \frac{1}{n}\sum_{i=1}^{n}\max_{j \neq i}\left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}\right)$$

where *n* is the number of clusters, c_i is the centroid of cluster i, σ_i is the average distance of all elements in cluster i to centroid c_i, and $d(c_i, c_j)$ is the distance between centroids c_i and c_j.

Other Indices:

- **Calinski–Harabasz**
- **Gamma index**
- **C-Index**
- **Score Function**
- **Symmetry, COP, SV, OS Indices etc.**

REFERENCES

K. Fukunaga; Introduction to Statistical Pattern Recognition, Second Edition, Academic Press, Morgan Kaufmann, 1990.

Richard O. Duda, Peter E. Hart, and David G. Stork. "Pattern Classification." Wiley, 1973.

Christopher M. Bishop, "Pattern Recognition and Machine Learning." Springer, 2006.

M. R. Anderberg. Cluster Analysis for Applications. Academic Press, 1973.

A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Prentice Hall, 1988.

L. Kaufman and P. J. Rousseeuw. Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons, 1990.

G. J. McLachlan and K.E. Bkasford. Mixture Models: Inference and Applications to Clustering. John Wiley and Sons, 1988.

Demo

Visualizing DBSCAN Clustering

Link: https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

https://theanlim.rbind.io/post/clustering-k-means-k-means-and-gganimate/

References (1)

- R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. SIGMOD'98
- M. R. Anderberg. Cluster Analysis for Applications. Academic Press, 1973.
- M. Ankerst, M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure, SIGMOD'99.
- Beil F., Ester M., Xu X.: "Frequent Term-Based Text Clustering", KDD'02
- M. M. Breunig, H.-P. Kriegel, R. Ng, J. Sander. LOF: Identifying Density-Based Local Outliers. SIGMOD 2000.
- M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases. KDD'96.
- M. Ester, H.-P. Kriegel, and X. Xu. Knowledge discovery in large spatial databases: Focusing techniques for efficient class identification. SSD'95.
- D. Fisher. Knowledge acquisition via incremental conceptual clustering. Machine Learning, 2:139-172, 1987.
- D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamic systems. VLDB'98.
- V. Ganti, J. Gehrke, R. Ramakrishan. CACTUS Clustering Categorical Data Using Summaries. KDD'99.

References (2)

- D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamic systems. In Proc. VLDB'98.
- S. Guha, R. Rastogi, and K. Shim. Cure: An efficient clustering algorithm for large databases. SIGMOD'98.
- S. Guha, R. Rastogi, and K. Shim. ROCK: A robust clustering algorithm for categorical attributes. In *ICDE'99*, pp. 512-521, Sydney, Australia, March 1999.
- A. Hinneburg, D.l A. Keim: An Efficient Approach to Clustering in Large Multimedia Databases with Noise. KDD'98.
- A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Printice Hall, 1988.
- G. Karypis, E.-H. Han, and V. Kumar. CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling. *COMPUTER*, 32(8): 68-75, 1999.
- L. Kaufman and P. J. Rousseeuw. Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons, 1990.
- E. Knorr and R. Ng. Algorithms for mining distance-based outliers in large datasets. VLDB'98.

References (3)

- G. J. McLachlan and K.E. Bkasford. Mixture Models: Inference and Applications to Clustering. John Wiley and Sons, 1988.
- R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. VLDB'94.
- L. Parsons, E. Haque and H. Liu, Subspace Clustering for High Dimensional Data: A Review, SIGKDD Explorations, 6(1), June 2004
- E. Schikuta. Grid clustering: An efficient hierarchical clustering method for very large data sets. Proc. 1996 Int. Conf. on Pattern Recognition,.
- G. Sheikholeslami, S. Chatterjee, and A. Zhang. WaveCluster: A multi-resolution clustering approach for very large spatial databases. VLDB'98.
- A. K. H. Tung, J. Han, L. V. S. Lakshmanan, and R. T. Ng. Constraint-Based Clustering in Large Databases, ICDT'01.
- A. K. H. Tung, J. Hou, and J. Han. Spatial Clustering in the Presence of Obstacles, ICDE'01
- H. Wang, W. Wang, J. Yang, and P.S. Yu. Clustering by pattern similarity in large data sets, SIGMOD' 02.
- W. Wang, Yang, R. Muntz, STING: A Statistical Information grid Approach to Spatial Data Mining, VLDB'97.
- T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH : An efficient data clustering method for very large databases. SIGMOD'96.
- Xiaoxin Yin, Jiawei Han, and Philip Yu, "LinkClus: Efficient Clustering via Heterogeneous Semantic Links", in Proc. 2006 Int. Conf. on Very Large Data Bases (VLDB'06), Seoul, Korea, Sept. 2006.