Clustering Algorithms :
Kmeans, Gaussian Mixture Model, and DBSCAN

HY Eric Kim

University of Seoul, Department of Statistics

2019-01-24

Clustering : Introduction

Cluster Analysis(Clustering)

A task of grouping a set of objects in such a way that objects in the same group (cluster) are more similar to each other than to those in other groups.

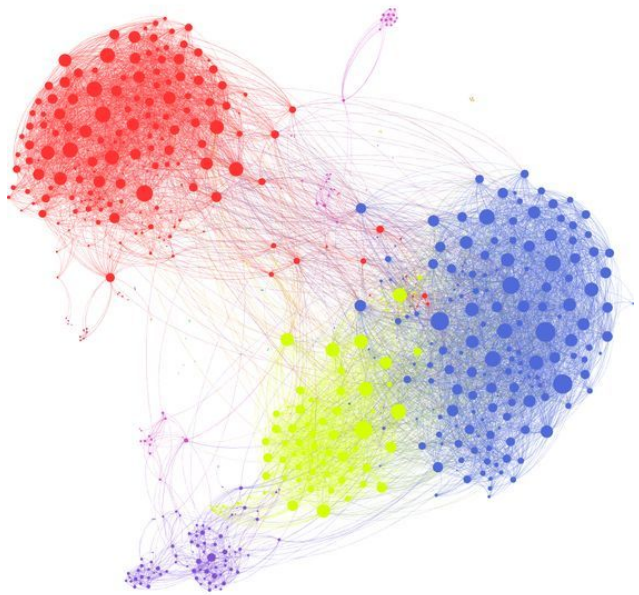


Figure: Clustering Algorithm

Clustering : Algoritms

- Clustering algorithms can be categorized based on their cluster model.

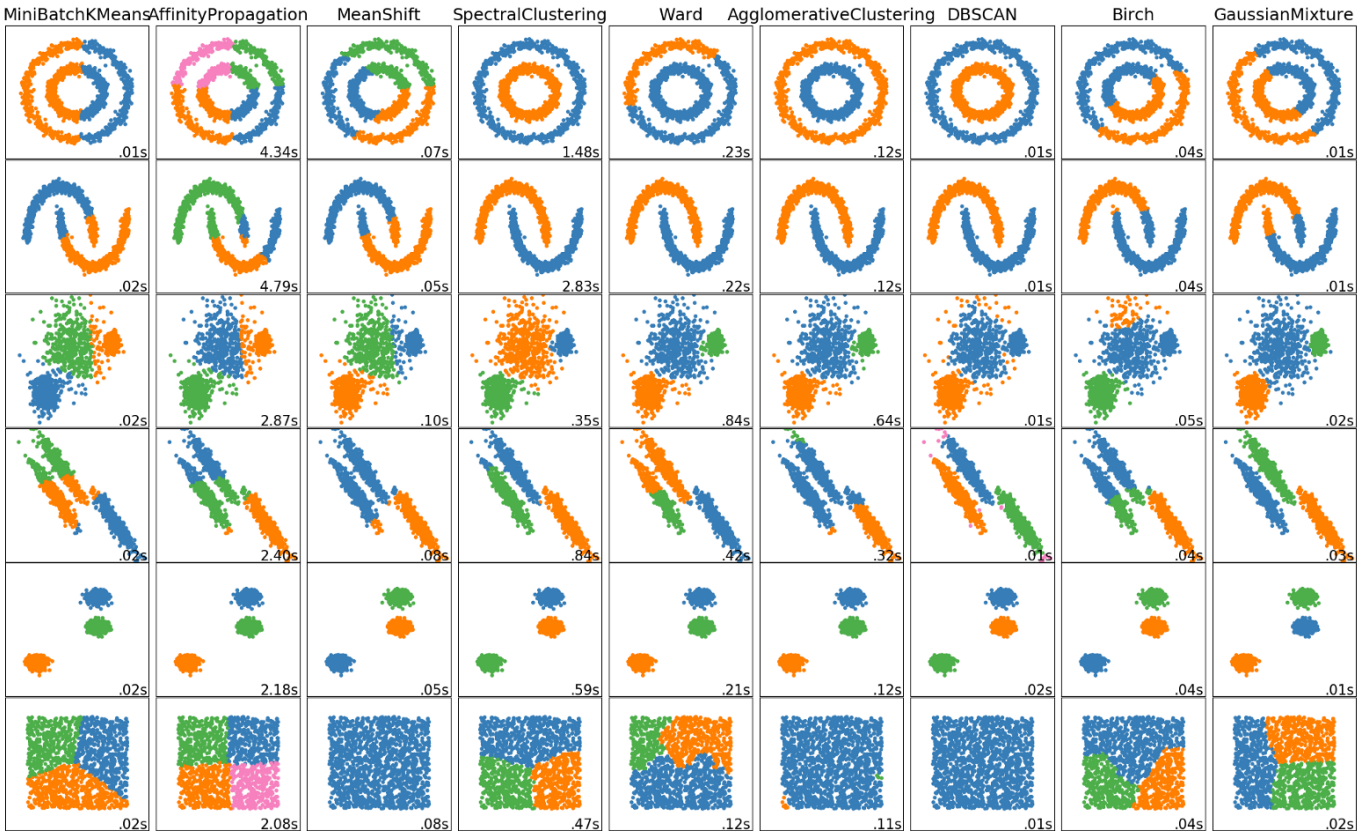


Figure: Various Algorithms

Clustering : Category

- Centroid-based Clustering

  : Kmeans Clustering

- Distribution-based Clustering

  : Gaussian-Mixture-Model

- Density-based Clustering

  : DBSCAN

- Connectivity-based Clustering

  : Hierarchical Clustering

Centroid-based Clustering : Kmeans Clustering

Kmeans Clustering

It aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean.

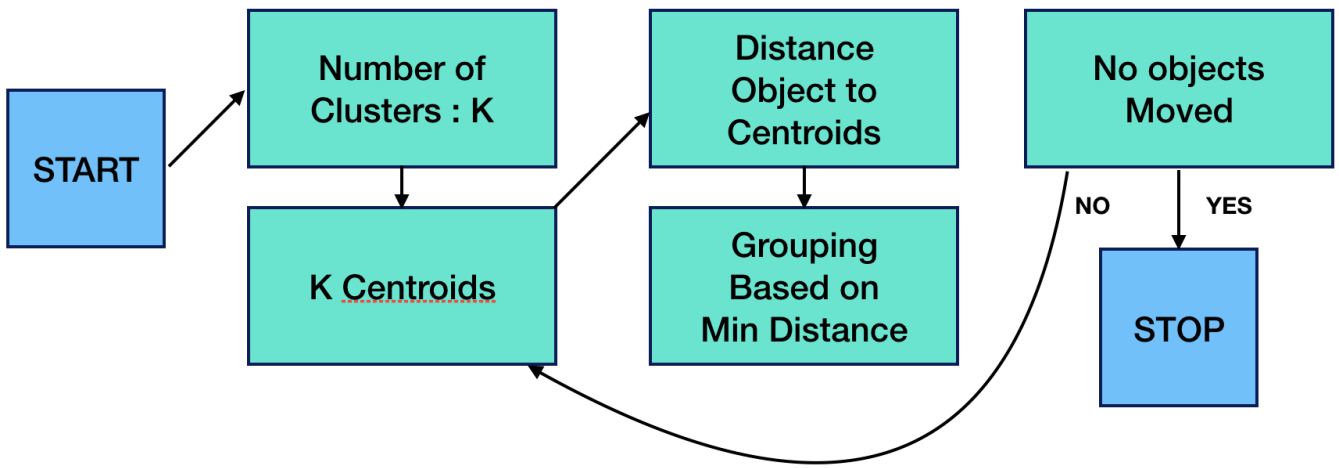


Figure: Kmeans Algorithm

Kmeans Clustering : Algorithm

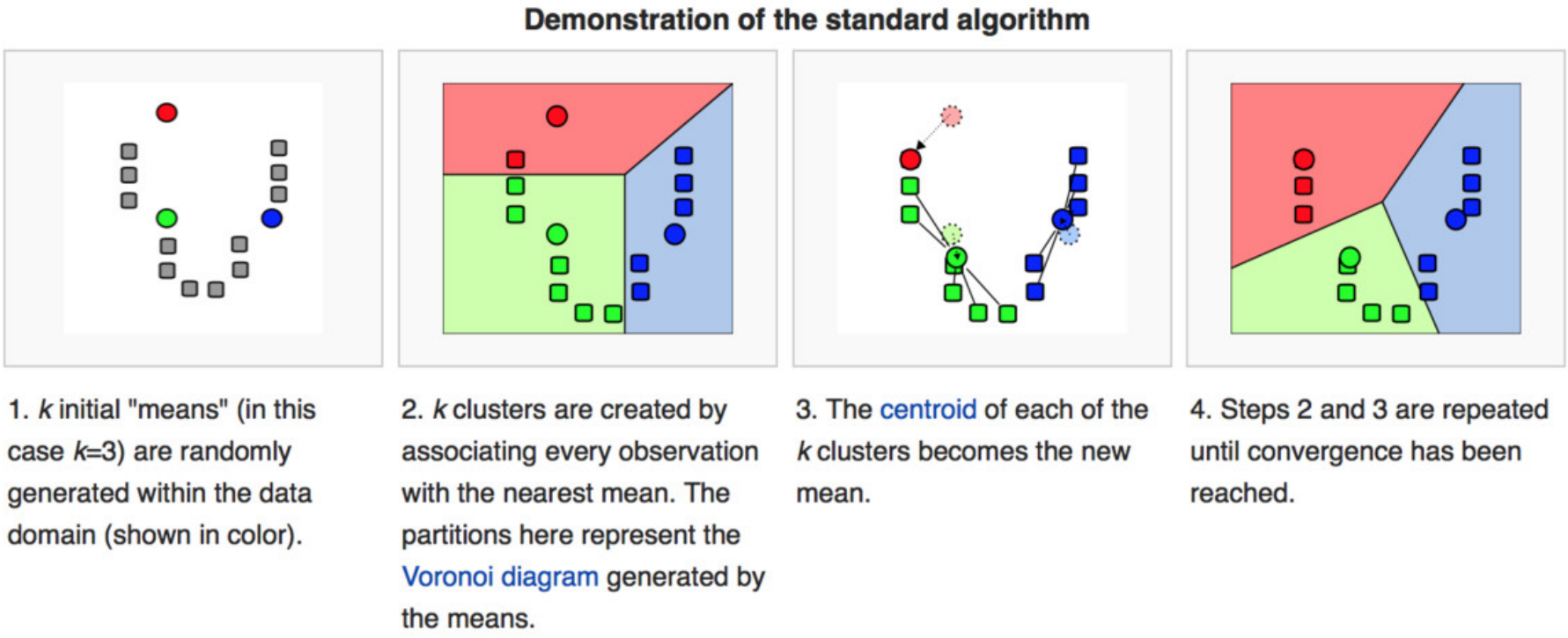


Figure: Basic Steps in Kmeans Algorithm

Kmeans Clustering : Discussion

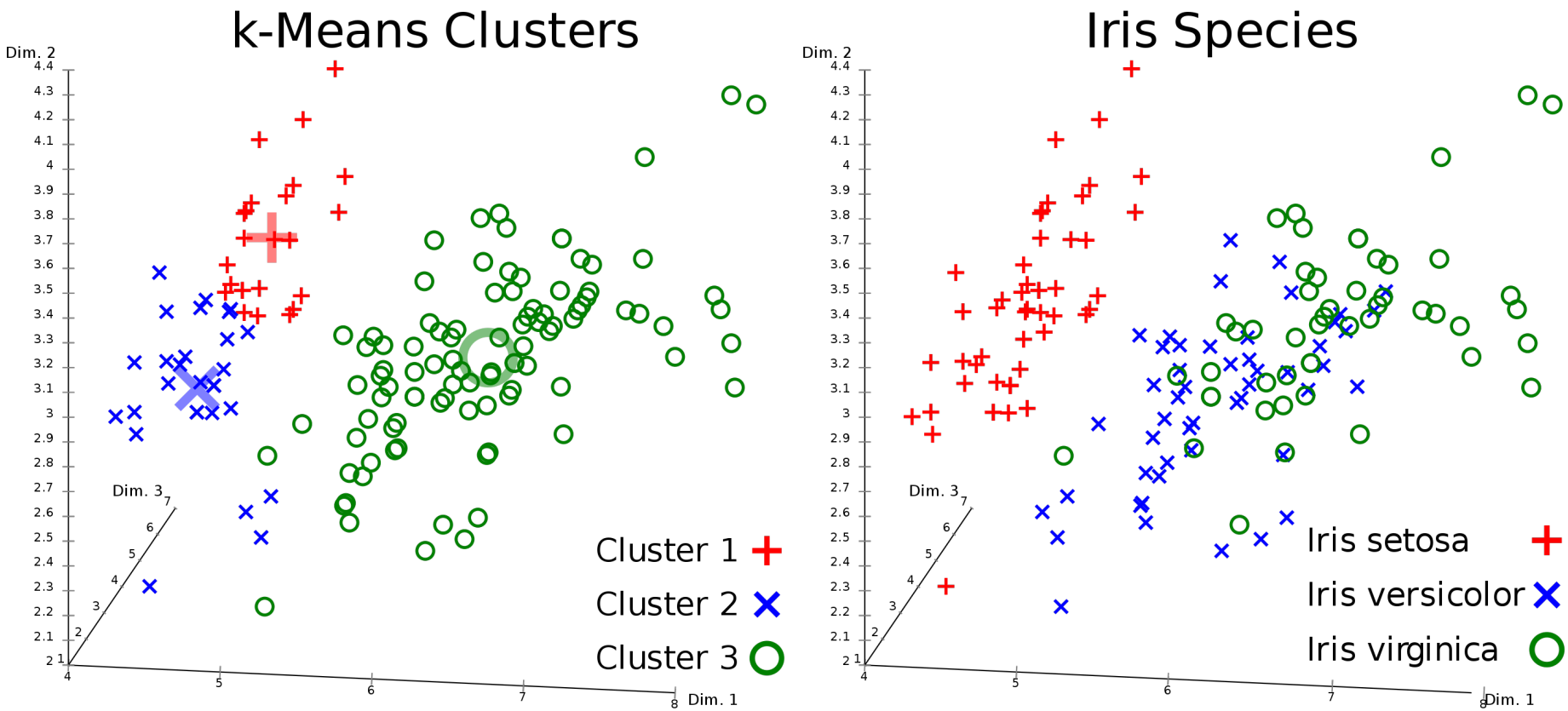


Figure: Kmeans Example from Iris data

Distribution-based Clustering : GMM

Gaussian Mixture Model

A Gaussian Mixture Model (GMM) is a parametric probability density function represented as a weighted sum of Gaussian component densities.

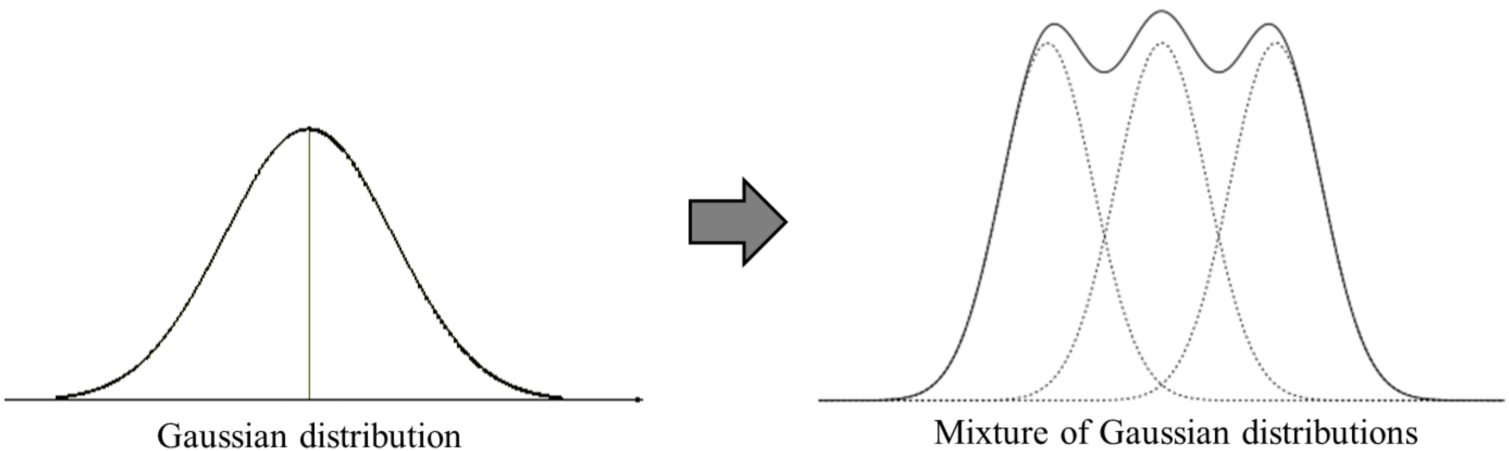


Figure: Mixture of Gaussian Distribution

GMM : Introduction

$$p(x) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \Sigma_k)$$

- x is a D-dimensional continuous-valued data vector.
- π_k is called Mixing Coefficient : The probability that decides which kth Gaussian Distribution is chosen.
  - $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$

- Learning GMM is equivalent to learning π_k, μ_k, Σ_k.

GMM : Introduction

- Using GMM, we try to find which distribution x_n came from.
- To make this happen, we define $\gamma(z_{nk})$.

  $$\gamma(z_{nk}) = p(z_{nk} = 1|x_n)$$

  - $z_{nk} \in \{0, 1\}$: Binary variable which assigns x_n to 1 if it is from kth Dist or 0.
- We assign x_n to kth Dist which has the highest $\gamma(z_{nk})$ as we compute every $\gamma(z_{nk})$ given x.

GMM : EM algorithm

Algorithm 1: EM algorithm for GMM

Input : a given data $X = \{x_1, x_2, ..., x_n\}$

Output : $\pi = \{\pi_1, \pi_2, ..., \pi_K\}$,
$\mu = \{\mu_1, \mu_2, ..., \mu_K\}$,
$\Sigma = \{\Sigma_1, \Sigma_2, ..., \Sigma_K\}$

Randomly initialize π, μ, Σ
for $t = 1 : T$ **do**
  // E-step
  for $n = 1 : N$ **do**
    for $k = 1 : K$ **do**
      $\gamma(z_{nk}) = \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n|\mu_j, \Sigma_j)}$
    end
  end
  // M-step
  for $k = 1 : K$ **do**
    $\mu_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) x_n}{\sum_{n=1}^{N} \gamma(z_{nk})}$
    $\Sigma_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^{N} \gamma(z_{nk})}$
    $\pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma(z_{nk})$
  end
end

Algorithm 2: GMM classification

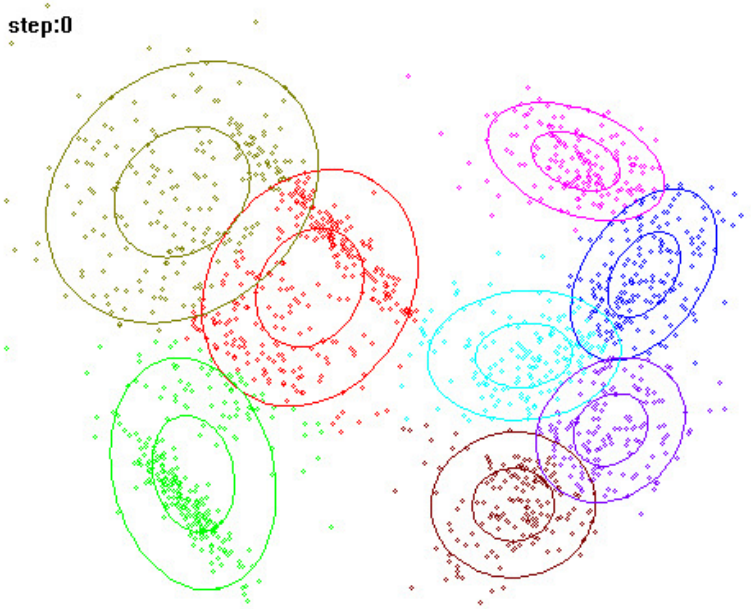
Input : a given data $X = \{x_1, x_2, ..., x_n\}$,
$\pi = \{\pi_1, \pi_2, ..., \pi_K\}$,
$\mu = \{\mu_1, \mu_2, ..., \mu_K\}$,
$\Sigma = \{\Sigma_1, \Sigma_2, ..., \Sigma_K\}$

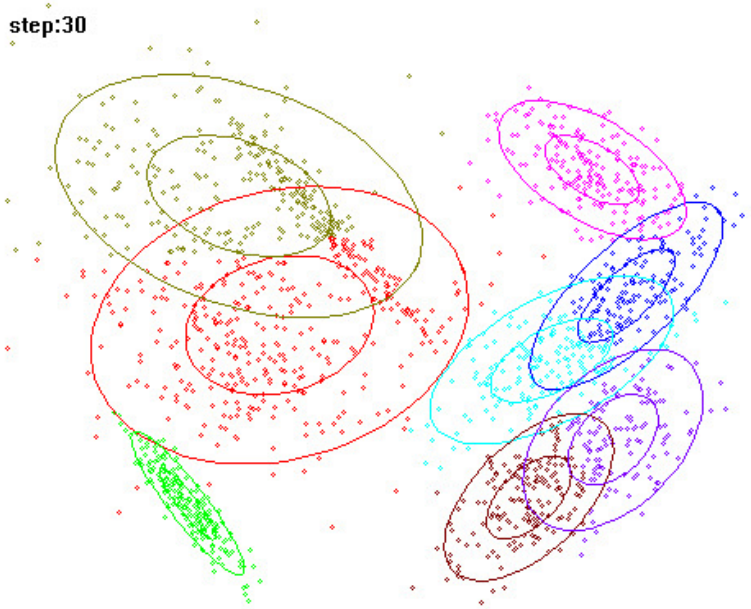
Output : class labels $y = \{y_1, y_2, ..., y_N\}$ for X

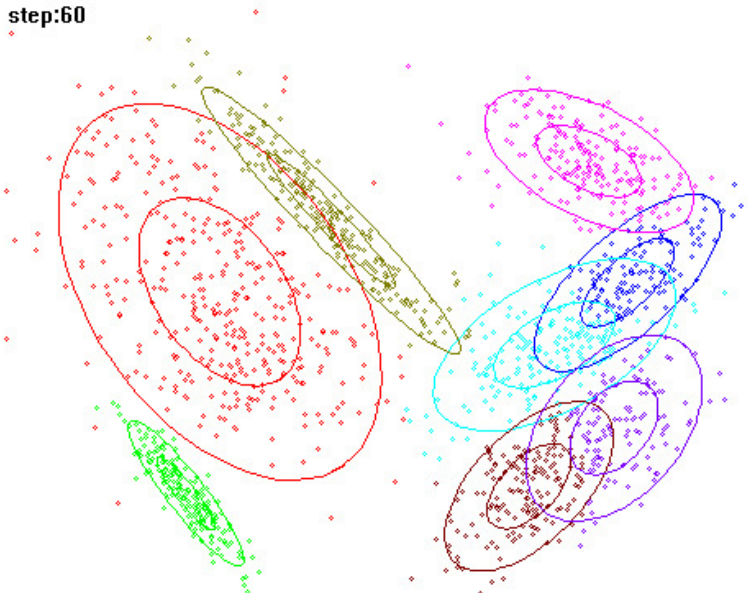
for $n = 1 : N$ **do**
  $y_n = \arg\max_k \gamma(z_{nk})$
end

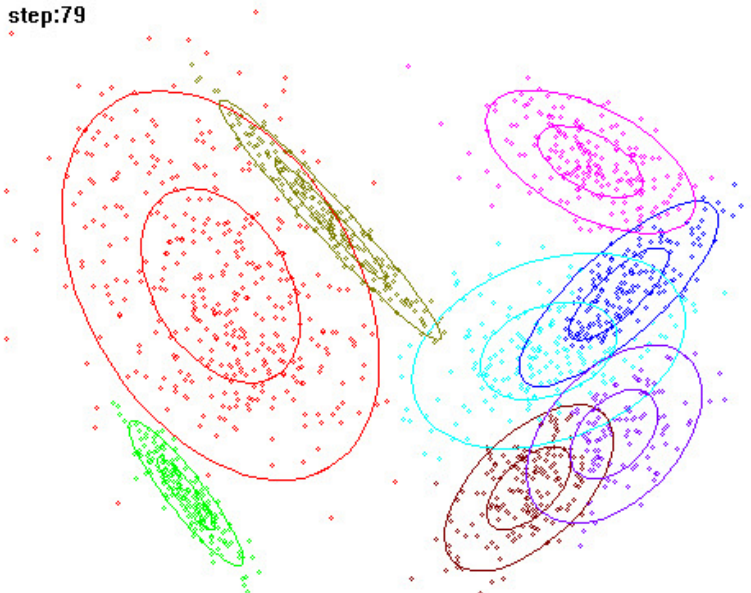
- We can estimate μ_k, π_k, Σ_k using above algorithm.

GMM : Demonstration









Density-based Clustering : DBSCAN

Density Based Algorithm for Discovering Discovering Clusters (DBSCAN)

It is an algorithm discovering clusters relying on a density-based notion of clusters.

- Practically, it requires one parameter.
- It can discover clusters of arbitrary shape.
- It is efficient in large dataset.

DBSCAN : A Density-based Notion of Clusters

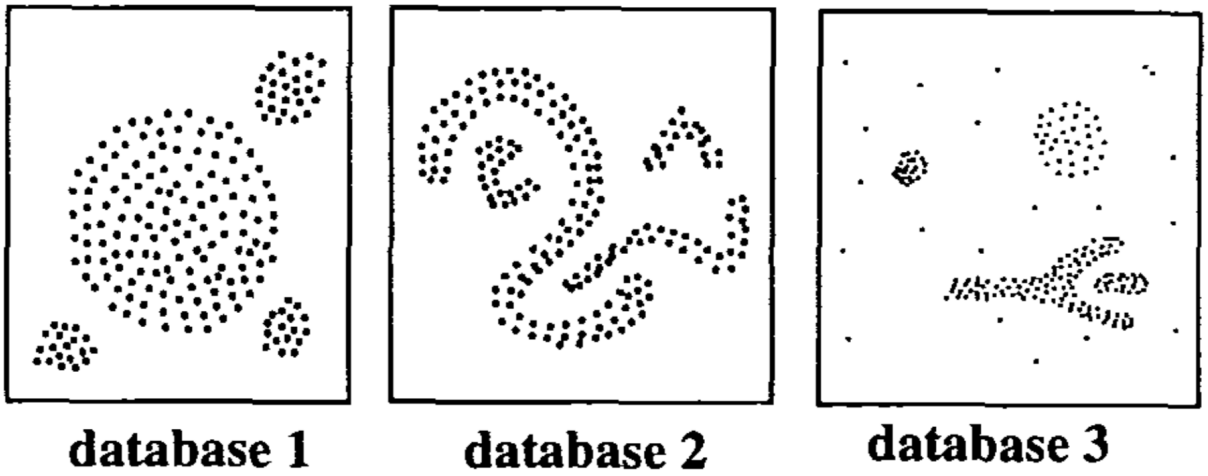


Figure: Sample databases

- The main reason we can recognize the clusters is that within each cluster we have a typical density of points which is considerably higher than outside the cluster.
- The density within the areas of noise is lower than the density in any of the clusters.

DBSCAN : Keywords

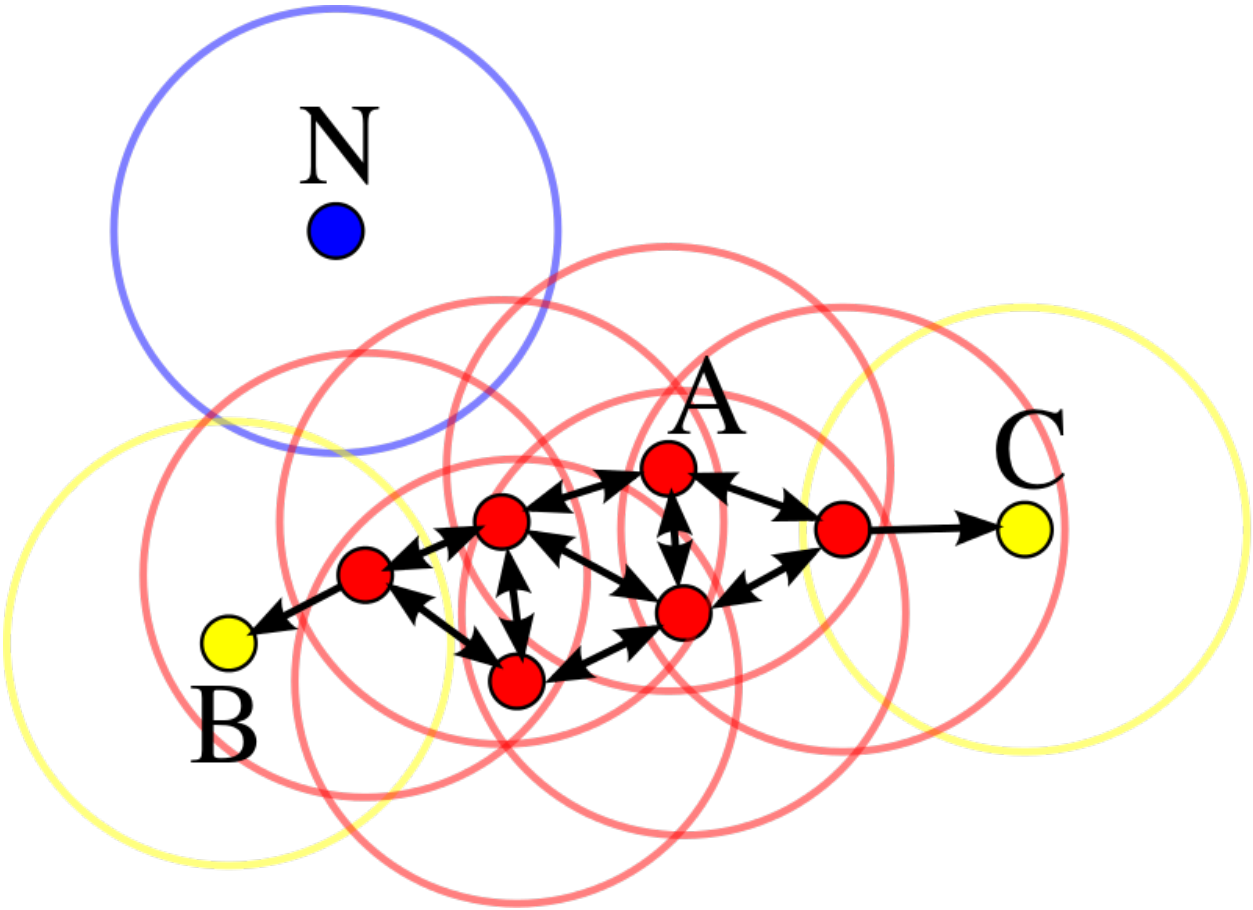
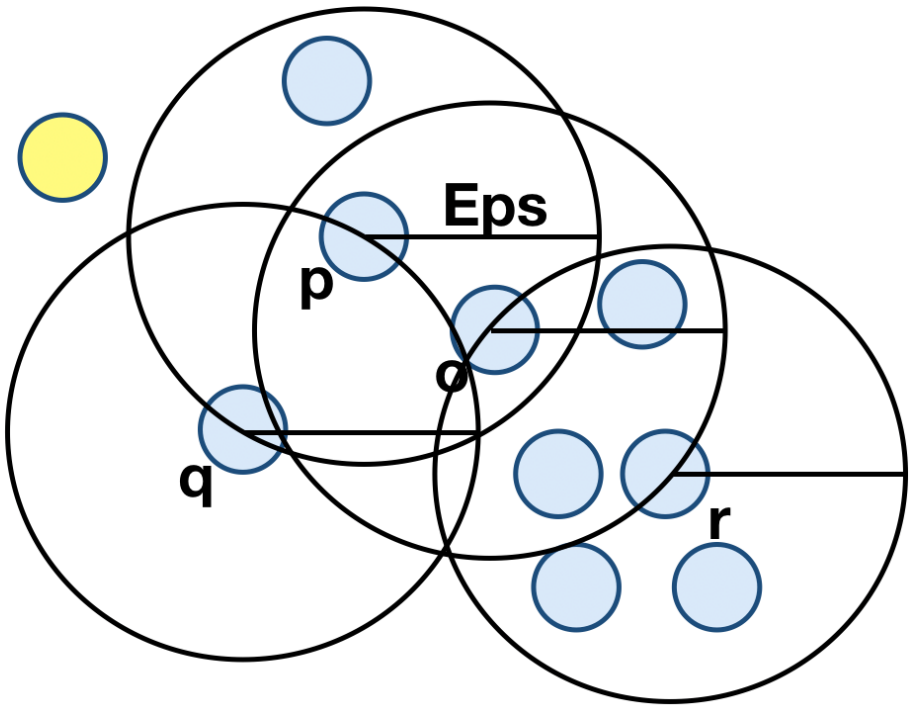


Figure: Core, Border Points and Noise

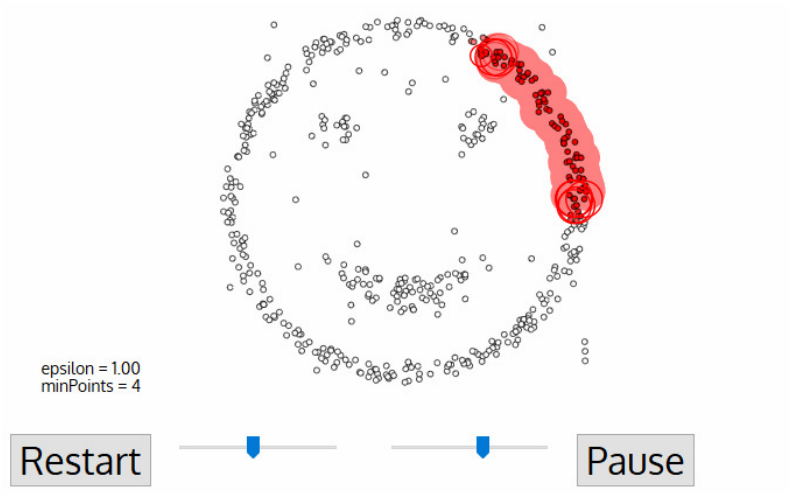
DBSCAN : Keywords

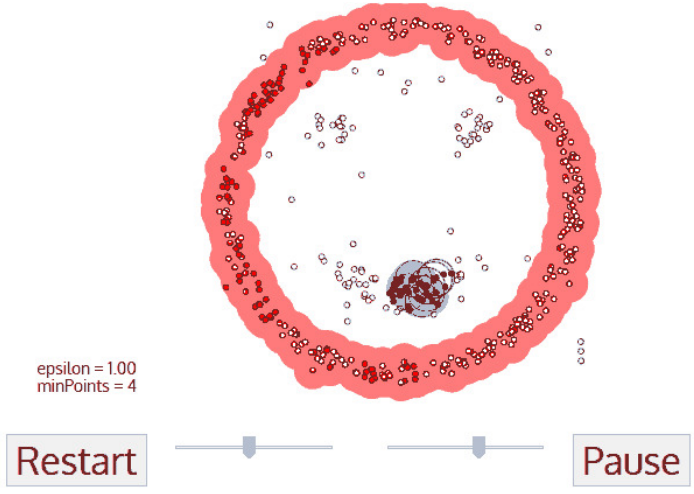


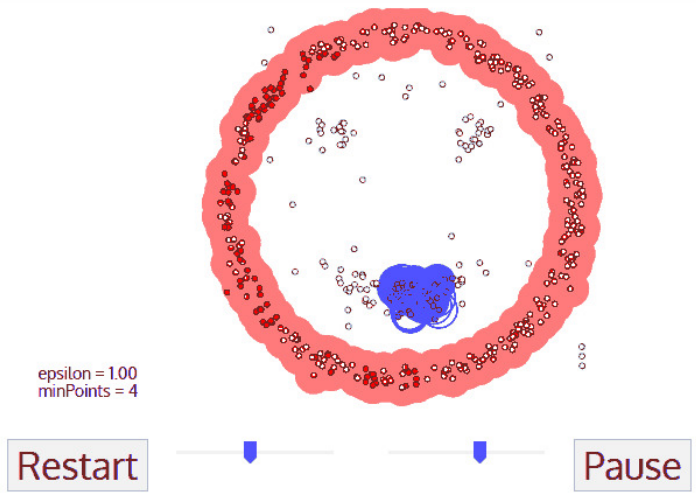
- **q is directly density reachable from p**
- **p is NOT directly density reachable from p**
- **p and r is density connected to each other by o**

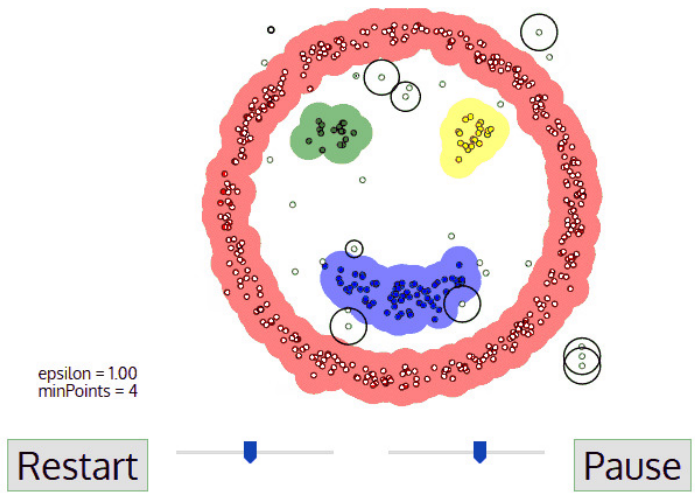
Figure: Reachability and Connectivity

DBSCAN : Demonstration









DBSCAN : Algorithm

INPUT : N Objects, *MinPts*, *Eps*

OUPUT : Clusters of objects

1. Select point P randomly.
2. Retrieve all 'density reachable' points from p w.r.t. *MinPts*, *Eps*.

3 - 1. If p is a core point, a cluster is formed.

3 - 2. if p is not a core point, DBSCAN visits the next points.

4. Continue till all of the points have been processed.

DBSCAN : Determining the parameters

- k-dist function : computes the dist between every points p and kth nearest point.
- DBSCAN computes k-dist function for every points and graphs it with sorted order. (*sorted k-dist graph*)

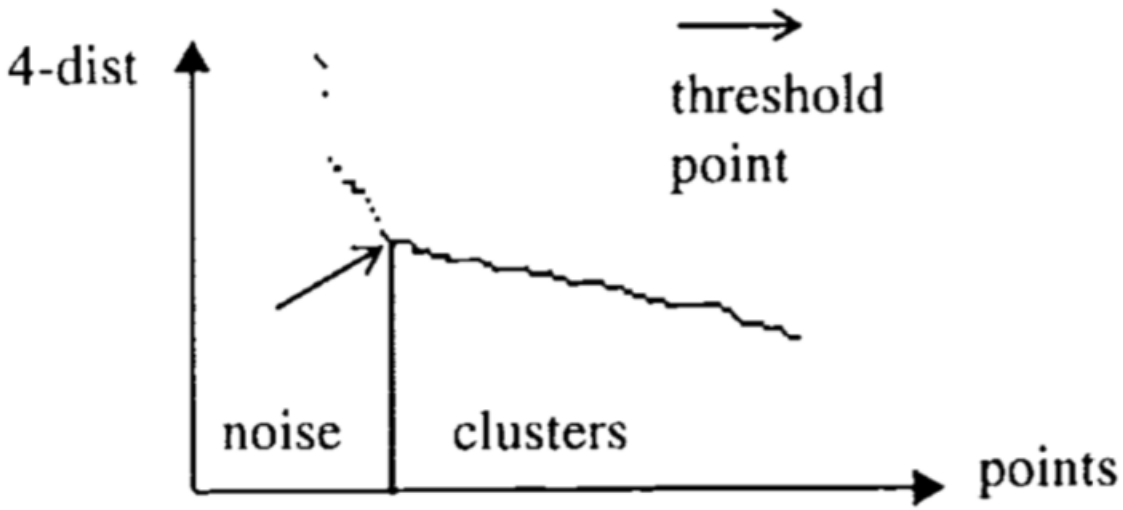
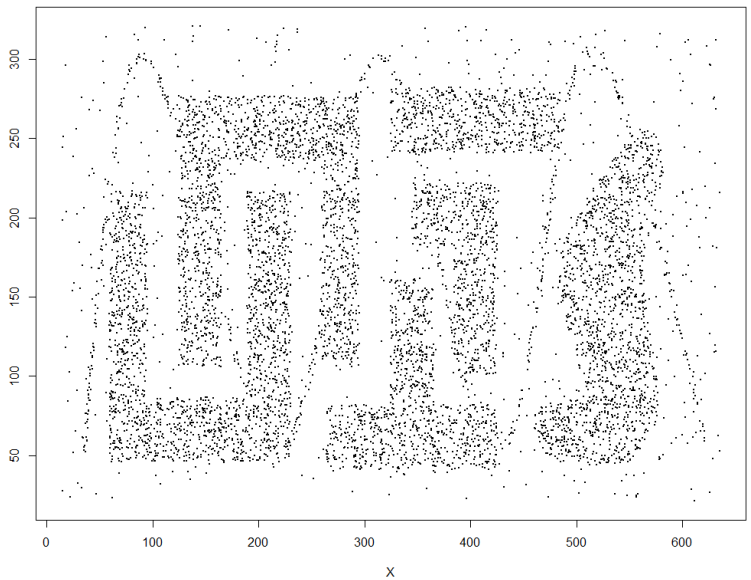


Figure: sorted 4-dist graph

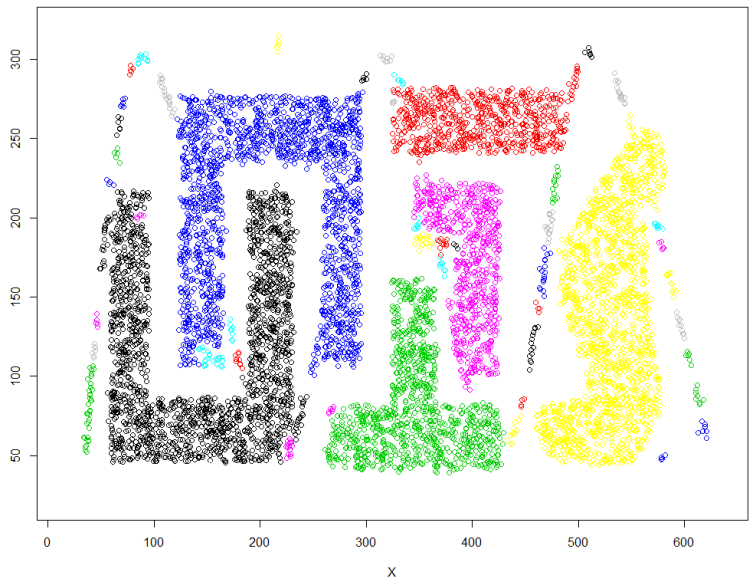
DBSCAN : Determining the parameters

- k= 4 is default since greater value than 4-dist-graph do not differ significantly. (Set MinPts to 4)
- The threshold for Eps can be chosen by the "valley" of the graph.
  - It is difficult for DBSCAN to catch but the user can easily find this in graphical representation.

DBSCAN : Demonstration - DS3 Data



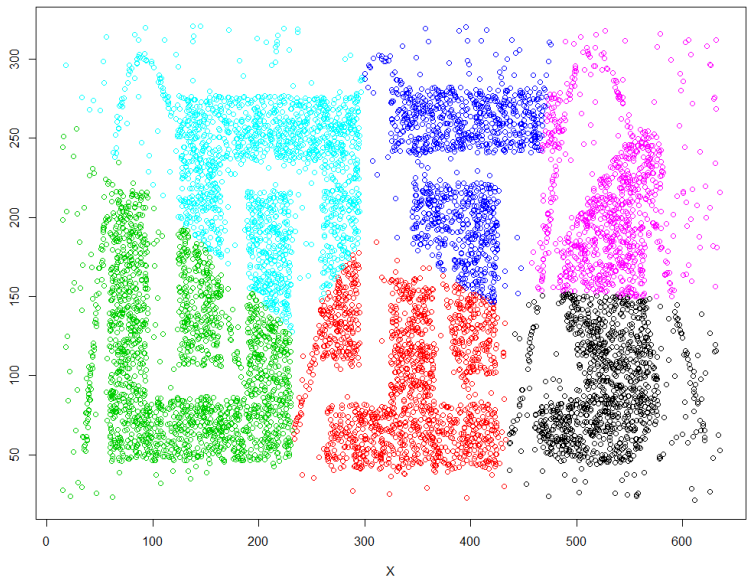
(a) True DS3 Cluster



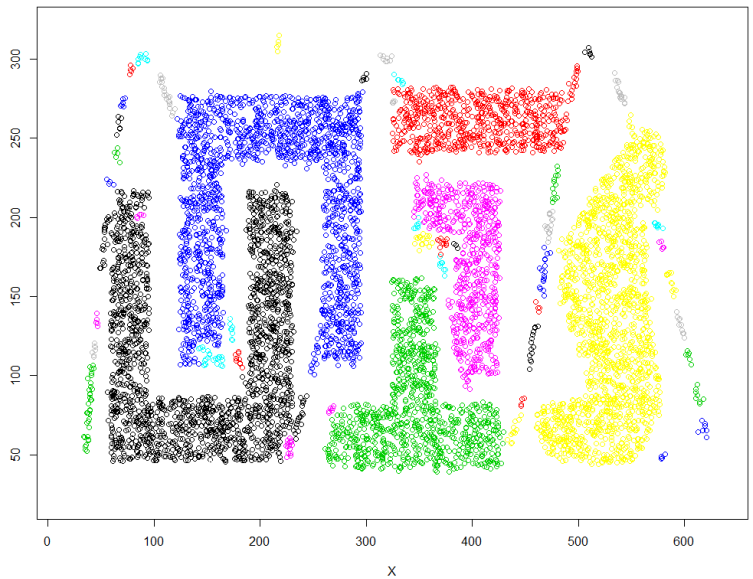
(b) DBSCAN DS3 Cluster

- The original data consists of 6 clusters

DBSCAN : Comparison with Kmeans



(a) Kmeans DS3 Cluster



(b) DBSCAN DS3 Cluster

- DBSCAN definitely does better than Kmeans clustering in this data

DBSCAN : Summary

Advantages

1. Does not require the number of clusters.
2. It can form an arbitrary shape of any cluster.
3. It has a notion of noise : robust to outliers

Disadvantages

1. Not entirely deterministic : a border point can be part of any cluster by ordering of the data
2. Has problems of identifying varying densities

Hierachical Clustering

Hierachical Clustering

Hierarchical clustering is a method of cluster analysis which seeks to build a hierarchy of clusters.

- Strategies for hierarchical clustering generally fall into two types.
  - Agglomerative(bottom-up): each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.
  - Divisive(top-down): all observations start in one cluster, and splits are performed

Hierachical Clustering : Two Types

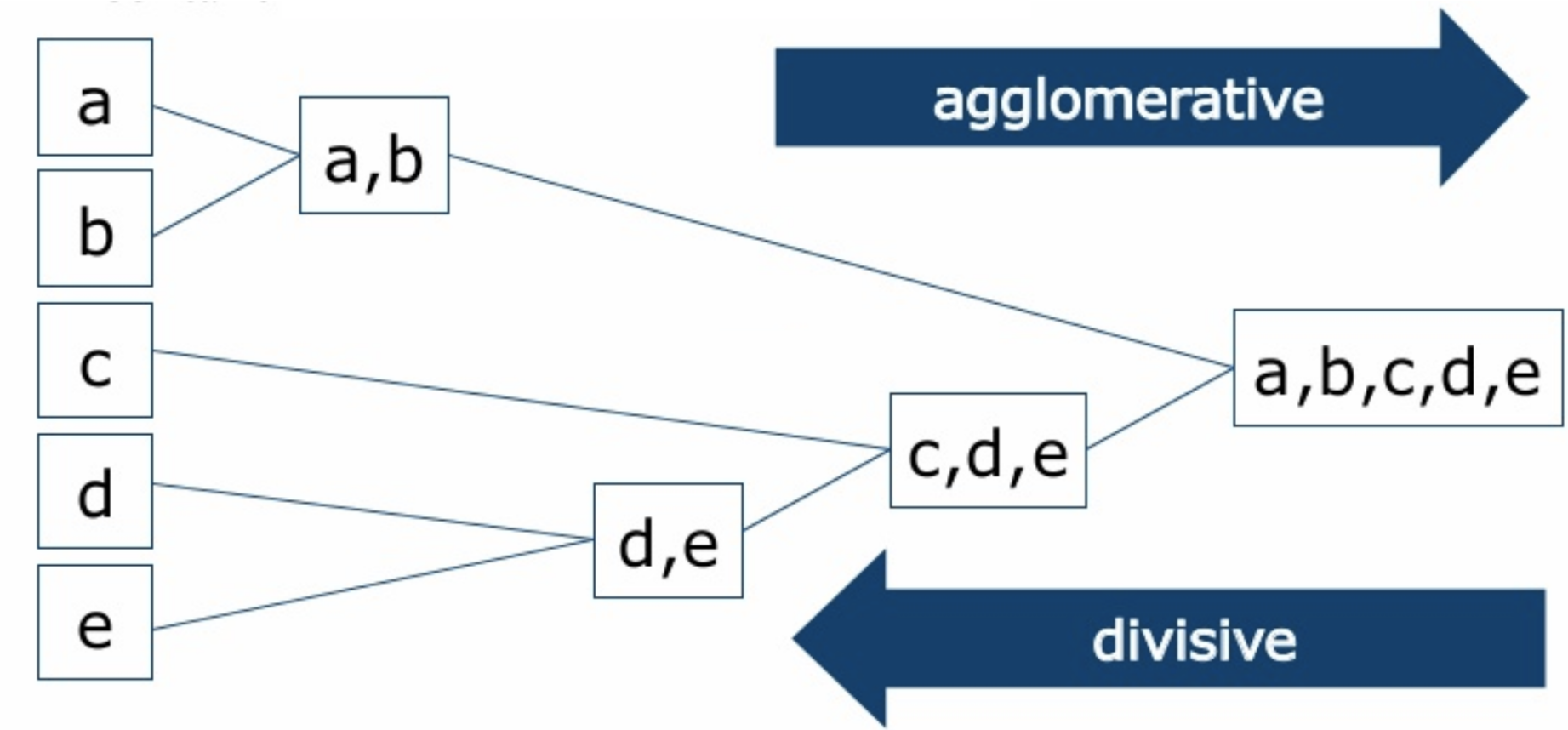


Figure: Two types in Hierachical Clustering

Hierachical Clustering : Cluster dissimilarity

- **Distance**

  The choice of an appropriate metric will influence the shape of the clusters.

  - Euclidean distance, ..

- **Linkage Criteria**

  The distance between sets of observations.

  - Maximum(Complete), Minimum(Single), Average linkage, ..

Hierachical Clustering : Linkage Criteria

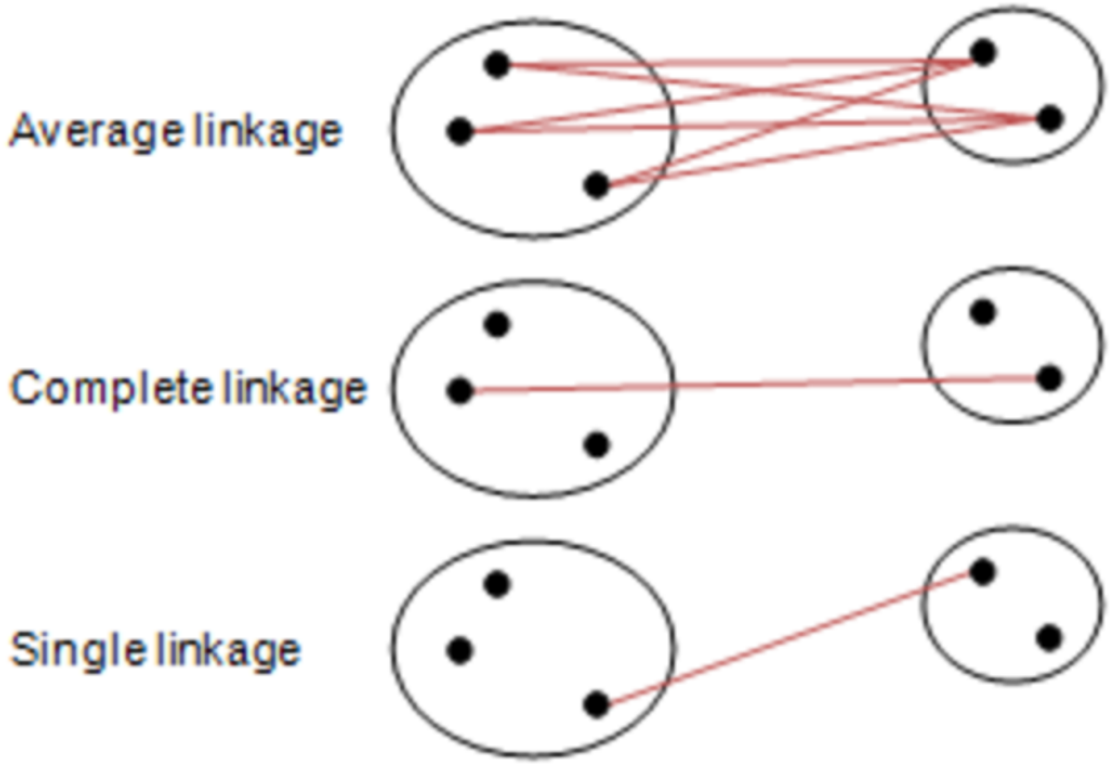


Figure: Three types of linkage criteria

Hierachical Clustering : Dendrogram

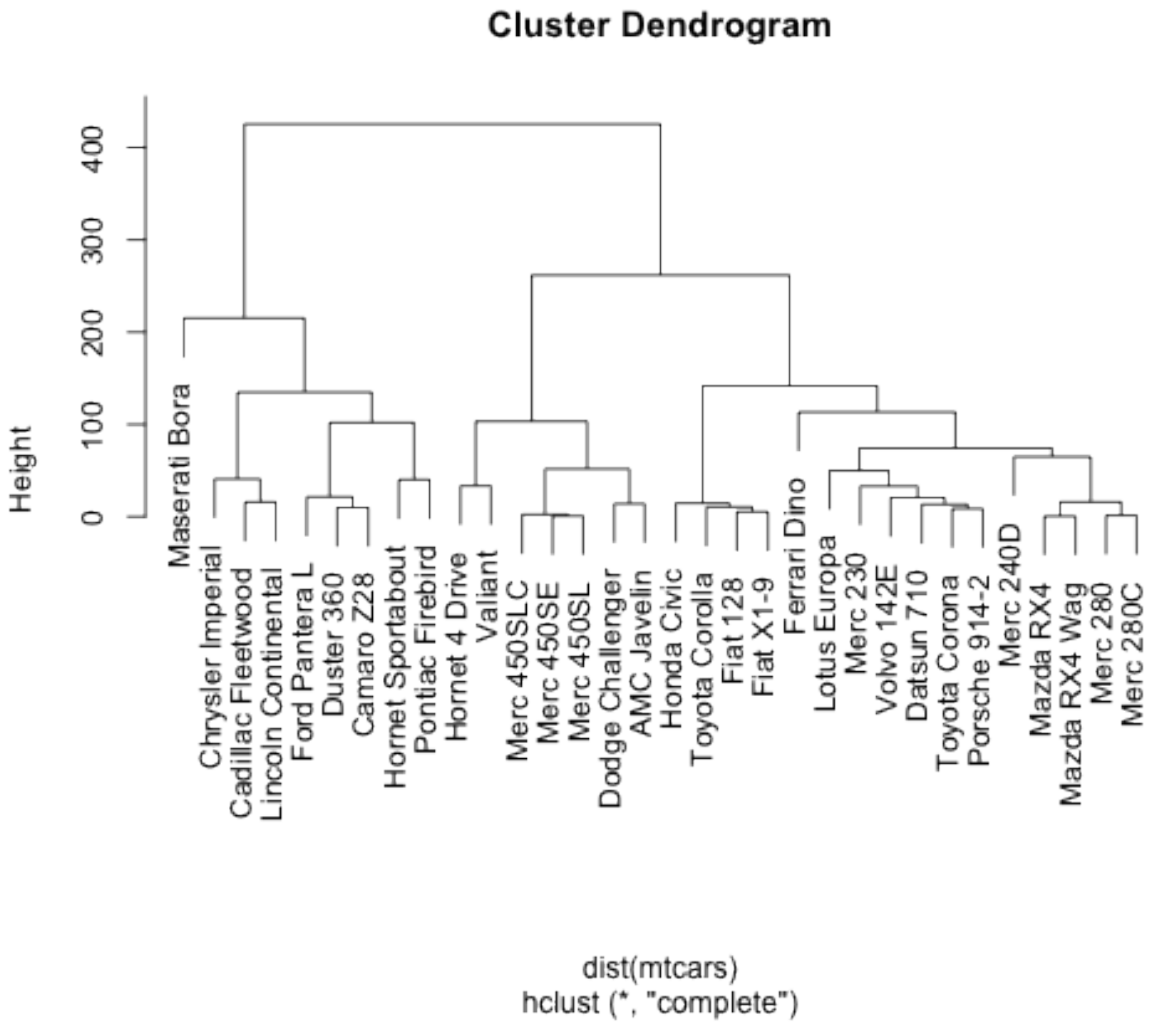


Figure: Tree using Compete Linkage

Hierachical Clustering : Summary

Advantages

1. Does not require the number of clusters.
2. They may correspond to meaningful taxonomies

Disadvantages

1. Sensitive to outliers
2. Difficult to determine the number of clusters

Spectral Clustering

Spectral Clustering

Algorithms that cluster points using eigenvectors of matrices derived from the data

- It can be done by following steps.

  1) Compute the Similarity Matrix S.

  2) Get Laplacian Matrix from S.

  3) Find the k smallest eigenvectors

  4) Perform standard Kmeans Clustering Algorithm.

- package *kernlab* has **specc** function to do the work.

Spectral Clustering : Laplacian Matrix

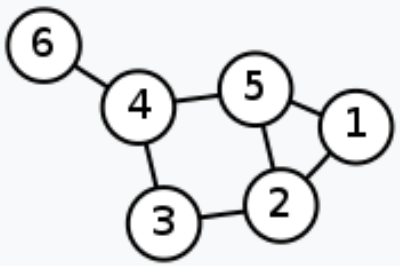
Labeled graph	Degree matrix	Adjacency matrix	Laplacian matrix
	$\begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 2 & -1 & 0 & 0 & -1 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & -1 & 0 & -1 & 3 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}$

Figure: Making Laplacian Matrix

Spectral Clustering : Laplacian Matrix

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	1.0000000	0	0.7429016	0.6319343	0.0000000	0.0000000	0.0000000	0
[2,]	0.0000000	1	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0
[3,]	0.7429016	0	1.0000000	0.0000000	0.0000000	0.0000000	0.6657756	0
[4,]	0.6319343	0	0.0000000	1.0000000	0.0000000	0.0000000	0.7195922	0
[5,]	0.0000000	0	0.0000000	0.0000000	1.0000000	0.7765565	0.0000000	0
[6,]	0.0000000	0	0.0000000	0.0000000	0.7765565	1.0000000	0.0000000	0
[7,]	0.0000000	0	0.6657756	0.7195922	0.0000000	0.0000000	1.0000000	0
[8,]	0.0000000	0	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	1

(a) Simility Matrix using Gaussian Kernel

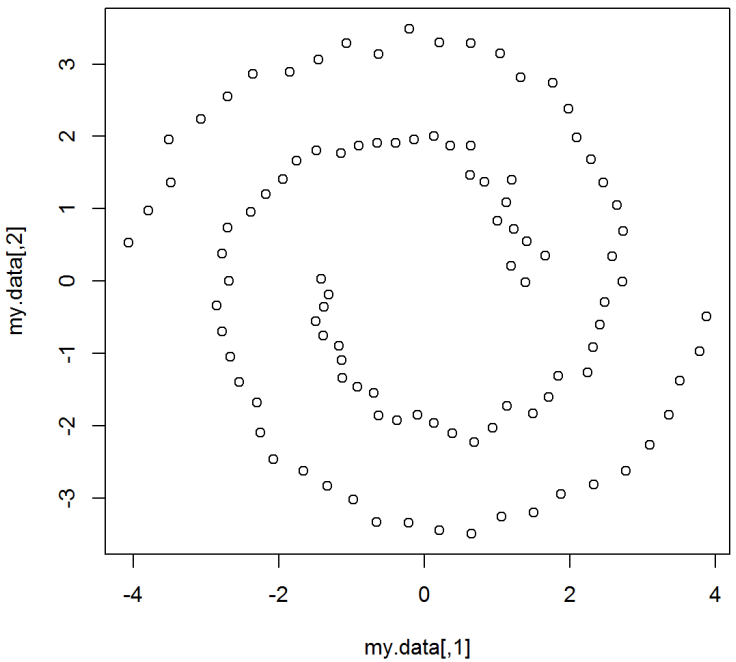
	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	2.374836	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
[2,]	0.000000	2.597451	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
[3,]	0.000000	0.000000	2.408677	0.000000	0.000000	0.000000	0.000000	0.000000
[4,]	0.000000	0.000000	0.000000	2.351526	0.000000	0.000000	0.000000	0.000000
[5,]	0.000000	0.000000	0.000000	0.000000	2.523175	0.000000	0.000000	0.000000
[6,]	0.000000	0.000000	0.000000	0.000000	0.000000	2.519936	0.000000	0.000000
[7,]	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	3.170424	0.000000
[8,]	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	2.302241

(b) Affinity Matrix

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]	[,11]	[,12]
[1,]	1.4	0.0	-0.7	-0.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
[2,]	0.0	1.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
[3,]	-0.7	0.0	1.4	0.0	0.0	0.0	-0.7	0.0	0.0	0.0	0.0	0.0
[4,]	-0.6	0.0	0.0	1.4	0.0	0.0	-0.7	0.0	0.0	0.0	0.0	0.0
[5,]	0.0	0.0	0.0	0.0	1.5	-0.8	0.0	0.0	0.0	0.0	0.0	0.0
[6,]	0.0	0.0	0.0	0.0	-0.8	1.5	0.0	0.0	0.0	0.0	0.0	0.0
[7,]	0.0	0.0	-0.7	-0.7	0.0	0.0	2.2	0.0	0.0	-0.8	0.0	0.0
[8,]	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.3	0.0	0.0	0.0	0.0
[9,]	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.5	0.0	0.0	0.0
[10,]	0.0	0.0	0.0	0.0	0.0	0.0	-0.8	0.0	0.0	1.6	-0.8	0.0
[11,]	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.8	1.5	-0.8
[12,]	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.8	1.5

(c) Laplacian Matrix

Spectral Clustering



(a) Simulation Data

(b) Clustering Using Specc Function

Reference

- Wikipedia : Clustering Analysis
- Wikipedia : Kmeans Clustering
- Douglas Reynolds : Gaussian Mixture Models
- Luca : mclust 5: Clustering, Classification and Density Estimation Using Gaussian Finite Mixture Models
- Martin Ester(1996) : A Density based algorithm for discovering clusters
- Wikipedia : Hierachical Clustering
- AY NG(2002) : On Spectral Clustering: Analysis and an algorithm