

Clustering Algorithms

Kmeans, Gaussian Mixture Model, and DBSCAN

Gunwoong Park

Department of Statistics
University of Seoul.

HY Eric Kim

Department of Statistics
University of Seoul.

Clustering and Classification

Clustering and Classification

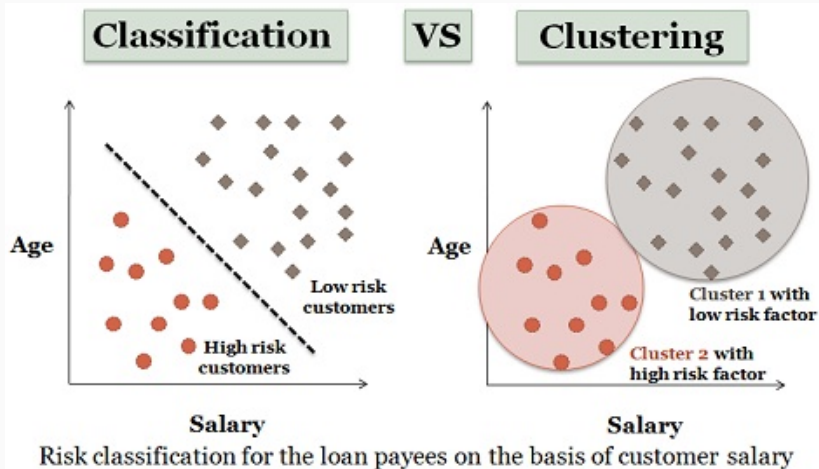


Figure 1: Classification vs Clustering

Clustering and Classification

Basis for Comparison	Classification	Clustering
Basic	It classifies data into numerous already defined definite classes	It maps the data into one of the multiple clusters where the arrangement of data relies on the similarities between them
Involved in	Supervised Learning	Unsupervised Learning
Training Sample	Provided	Not Provided

Clustering and Classification

- Classification categorizes the data with the help of provided training data.
- On the other hand, clustering uses different similarity measures to categorize the data.
- Reference : Difference between clustering and classification [[Techdifferences.com](https://www.techdifferences.com)]

Clustering

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.

- Common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, bioinformatics, data compression, and computer graphics.

Clustering : Introduction

- Cluster analysis itself is not one specific algorithm but the general task to be solved.
 - Various algorithms that differ significantly in their understanding of what constitutes a cluster and how to efficiently find them.
 - Popular notions of clusters include groups with small distances between cluster members, dense areas of the data space, intervals or particular statistical distributions.

Clustering : Algorithms

- Clustering algorithms can be categorized based on their cluster model.
 - There are possibly over 100 published clustering algorithms.
- There is no objectively “correct” clustering algorithm.
 - The most appropriate clustering algorithm for a particular problem often needs to be chosen experimentally.
Unless there is a mathematical reason to prefer one cluster model over another.
- An algorithm that is designed for one kind of model will generally fail on a data set that contains a radically different kind of model.

Clustering : Category

- Centroid-based Clustering
: Kmeans Clustering
- Distribution-based Clustering
: Gaussian-Mixture-Model
- Density-based Clustering
: DBSCAN
- Connectivity-based Clustering
(Hierarchical Clustering)

Kmeans Clustering

Centroid-based Clustering : Kmeans Clustering

- Clusters are represented by a central vector
: which may not necessarily be a member of the data set.
- Kmeans clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean.

Kmeans Clustering : Introduction

- Basic Concept

Given a set of observation (x_1, x_2, \dots, x_n) , where each observation is a d -dimensional real vector, k -means clustering aims to partition the n observation into $k \leq n$ sets $S = \{S_1, S_2, \dots, S_k\}$ so as to minimize the within-cluster sum of squares.

- Formally, the object is to find

$$\arg \min_S \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2$$

where μ_i is the mean point of S_i .

Kmeans Clustering : Algorithm

- Kmeans Algorithm : an iterative algorithm
 - Given an initial set of kmeans $m_1^{(1)}, m_2^{(1)}, \dots, m_k^{(1)}$
 - (ASSIGN STEP) Assign each observation to the cluster whose mean has the least squared Euclidean distance
$$S_i^{(t)} = \{x_p : \|x_p - m_i^{(t)}\|^2 \leq \|x_p - m_j^{(t)}\|^2, \quad \forall j, 1 \leq j \leq k\}$$
where x_p is assigned to exactly one $S^{(t)}$
 - (UPDATE STEP) Calculate the new means to be the centroids of the observations in the new clusters
$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$
- The algorithm has converged when the assignments no longer change.
 - This does not guarantee to find the optimum.
(Hartigan, J, A (1979) : "Algorithm AS 136: A k-Means Clustering Algorithm")

Kmeans Clustering : Initialization

- There are several methods for initialization.

The most known methods are Forgy and Random Partition

- The Forgy method randomly chooses k observations from the dataset and uses these as the initial means.
- The Random Partition method first randomly assigns a cluster to each observation and then proceeds to the update step.

Thus computing the initial mean to be the centroid of the cluster's randomly assigned points.

Kmeans Clustering : Discussion

- The number of clusters k is an input parameter: an inappropriate choice of k may yield poor results.
- Convergence to a local minimum may produce counter-intuitive ("wrong") results.

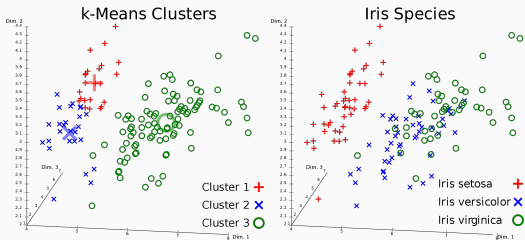


Figure 2: Kmeans Example from Iris data

Gaussian Mixture Model

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.

- GMM parameters are estimated from training data using the iterative Expectation-Maximization (EM) algorithm or Maximum A Posteriori (MAP) estimation.
- GMMs are commonly used as a parametric model of the probability distribution of continuous measurements or features in a biometric system, such as vocal-tract related spectral features in a speaker recognition system.

GMM : Introduction

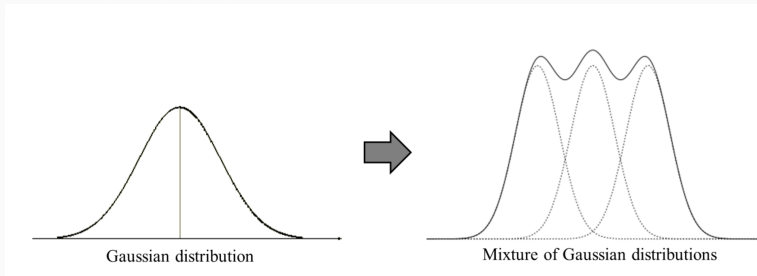


Figure 3: Mixture of Gaussian Distribution

- A Gaussian mixture model is a weighted sum of K component Gaussian densities as given by the equation.

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

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 k th 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 k th Dist or 0.
- We assign x_n to k th Dist which has the highest $\gamma(z_{nk})$ as we compute every $\gamma(z_{nk})$ given x .

GMM : Introduction

- $\gamma(z_{nk})$ can be archived by using Bayes' Theorem.

$$\begin{aligned}\gamma(z_{nk}) &= p(z_{nk} = 1 | x_n) = \frac{p(z_{nk}=1)p(x_n|z_{nk}=1)}{\sum_{j=1}^K p(z_{nj}=1)p(x_n|z_{nj}=1)} \\ &= \frac{\pi_k N(x_n | \mu_k, \Sigma_k)}{\sum_{k=1}^K \pi_k N(x_n | \mu_k, \Sigma_k)}\end{aligned}$$

- we can estimate μ_k, π_k, Σ_k from the idea of MLE.

$$\begin{aligned}- \mu_k &= \frac{\sum_{n=1}^N \gamma(z_{nk} x_n)}{\sum_{n=1}^N \gamma(z_{nk})} \\ - \Sigma_k &= \frac{\sum_{n=1}^N \gamma(z_{nk}) (x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^N \gamma(z_{nk})} \\ - \pi_k &= \frac{1}{N} \sum_{n=1}^N \gamma(z_{nk})\end{aligned}$$

GMM : EM Algorithm

- EM Algorithm : an iterative method to find maximum likelihood.
- The basic idea of the EM algorithm is, beginning with an initial model λ , to estimate a new model $\bar{\lambda}$ such that $p(x|\lambda) \leq p(x|\bar{\lambda})$.

GMM : EM algorithm

Algorithm 1: EM algorithm for GMM

Input : a given data $X = \{x_1, x_2, \dots, x_n\}$
Output: $\pi = \{\pi_1, \pi_2, \dots, \pi_K\}$,
 $\mu = \{\mu_1, \mu_2, \dots, \mu_K\}$,
 $\Sigma = \{\Sigma_1, \Sigma_2, \dots, \Sigma_K\}$
1 Randomly initialize π, μ, Σ
2 **for** $t = 1 : T$ **do**
3 // E-step
4 **for** $n = 1 : N$ **do**
5 **for** $k = 1 : K$ **do**
6 $\gamma(z_{nk}) = \frac{\pi_k N(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j N(x_n | \mu_j, \Sigma_j)}$
7 **end**
8 **end**
9 // M-step
10 **for** $k = 1 : K$ **do**
11 $\mu_k = \frac{\sum_{n=1}^N \gamma(z_{nk}) x_n}{\sum_{n=1}^N \gamma(z_{nk})}$
12 $\Sigma_k = \frac{\sum_{n=1}^N \gamma(z_{nk}) (x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^N \gamma(z_{nk})}$
13 $\pi_k = \frac{1}{N} \sum_{n=1}^N \gamma(z_{nk})$
14 **end**
15 **end**

Algorithm 2: GMM classification

Input : a given data $X = \{x_1, x_2, \dots, x_n\}$,
 $\pi = \{\pi_1, \pi_2, \dots, \pi_K\}$,
 $\mu = \{\mu_1, \mu_2, \dots, \mu_K\}$,
 $\Sigma = \{\Sigma_1, \Sigma_2, \dots, \Sigma_K\}$
Output: class labels $y = \{y_1, y_2, \dots, y_N\}$ for X
1 **for** $n = 1 : N$ **do**
2 $y_n = \arg \max_k \gamma(z_{nk})$
3 **end**

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

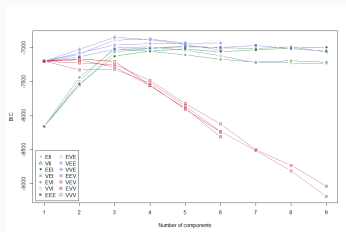
GMM : Demonstration

- Applying GMM is presented using Mclust package and wine data from gclus package.

STEP 1. Find the appropriate model for GMM by using Mclust function.

```
> summary(mod$BIC)
Best BIC values:
          VVE,3      EVE,4      VVE,4
BIC      -6849.391 -6873.61648 -6885.47222
BIC diff      0.000   -24.22499   -36.08073
```

(a) Summary of Mclust\$BIC



(b) Plot of Mclust\$BIC

GMM : Demonstration

STEP 2. Check the model.

```
> summary(mod)
-----
Gaussian finite mixture model fitted by EM algorithm
-----
Mclust VVE (ellipsoidal, equal orientation) model with 3 components:

log.likelihood  n  df      BIC      ICL
-3015.335 178 158 -6849.391 -6850.734

Clustering table:
 1  2  3
59 69 50
```

(a) Summary of Mclust

```
> table(Class, mod$classification)
```

Class	1	2	3
Barolo	59	0	0
Grignolino	0	69	2
Barbera	0	0	48

(b) Table : GMM vs Actual

GMM : Demonstration

STEP 3. Plot the result clustering.

```
> drmod <- MclustDR(mod, lambda = 1)
> summary(drmod)
```

Dimension reduction for model-based clustering and classification

Mixture model type: Mclust (VVE, 3)

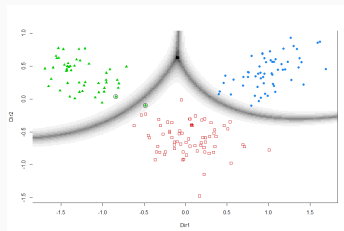
Clusters n
1 59
2 69
3 50

Estimated basis vectors:

	Dir1	Dir2
Alcohol	0.13399009	0.19209123
Malic	-0.03723778	0.06424412
Ash	-0.01313103	0.62738796
Alcalinity	-0.04299147	-0.03715437
Magnesium	-0.00053971	0.00051772
Phenols	-0.13507235	-0.04687991
Flavanoids	0.51323644	-0.13391186
Nonflavanoid	0.68462875	-0.61863302
Proanthocyanins	-0.07506153	-0.04652587
Intensity	-0.08855450	0.04877118
Hue	0.28941727	-0.39564601
OD280	0.36197696	-0.00779361
Proline	0.00070724	0.00075867

Eigenvalues 1.6189 1.292
Cum. % 55.6156 100.000

(a) Data Projection using
MclustDR



(b) Cluster Plot

DBSCAN

Density-based Clustering : DBSCAN

DBSCAN : Density Based Algorithm for Discovering Discovering Clusters

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

- It requires one parameter.
- It can discover clusters of arbitrary shape.
- It is efficient in large dataset.
- There are no well-known clustering algorithms offering solutions to the combination of these requirements.

DBSCAN : A Density-based Notion of Clusters

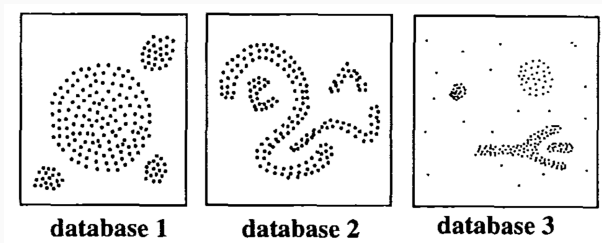


Figure 8: Sample databases

- 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 : Key Idea

- The key idea
 - For each points of a cluster, the neighborhood of a given radius has to contain at least minimum number of points i.e. the density in the neighborhood needs to exceed some threshold.
 - The shape of neighborhood is determined by the choice of a distance function for two points p and q denoted by $dist(p, q)$.

DBSCAN : Keywords in DBSCAN

Definition

Eps-Neighborhood of a point : The *Eps-Neighborhood* of a point p , denoted by $N_{Eps}(p) = \{q \in D | dist(p, q) \leq Eps\}$

- A naive approach : There are minimum number(*MinPts*) of points in an Eps-Neighborhood of that point.

For each point in a cluster

- This approach fails since there are 2 kinds of points
 - a) Core points : points inside of the cluster.
 - b) Border points : points on the border of the cluster.
- In general, border points contain significantly less points than core points.

DBSCAN : Keywords in DBSCAN

- Thus, we must set the minimum number of points to a relatively low value in order to include all points belonging to the same cluster.
 - This value is not characteristic for the respective cluster. (particularly in the presence of noise)
 - Therefore, for every point p in Cluster C , there's q in C such that p is inside $N_{Eps}(q)$, and $N_{Eps}(q)$ contains at least $MinPts$.
- This Definition is elaborated in the paper.
 - The paper defines several rules to define cluster.

DBSCAN : Cluster and Noise

Definition

Cluster C wrt. $Eps, MinPts$: non-empty subset of D satisfying

1) $\forall p, q$: if $p \in C$ and q is density reachable from p then $q \in C$. "Maximality"

2) $\forall p, q$: p is density-connected to q wrt. $Eps, MinPts$.
"Connectivity"

where D is dataset of points

Definition

Noise : There are k clusters wrt. $Eps_i, Minpts_i$, Noise is a set in D not belonging to C_i

$$\text{Noise} = \{p \in D \mid \forall i, p \notin C_i\}$$

DBSCAN : Cluster and Noise

- Cluster C contains at least $MinPts$ points.
 - Since it contains at least one p , and p must satisfy the core condition.
- Given parameters, we can find a cluster in two step
 - 1) Choose an arbitrary point from D satisfying core point condition as a seed.
 - 2) Retrieve all points that are density-reachable from the seed.

DBSCAN : In Real Application

- Ideally, we would have to know the appropriate parameter *Eps* and *MinPts* for each clusters and at least one point from respective cluster.
- Since it is impossible to get this information in advance for all cluisters of the database.
- There is simpe effective heuristic to determine the parametesrs of the thinnest cluster in the database.

Therefore, DBSCAN uses global values for all clusters.

- This may merge 2 close clusters.
 - Let $dist(S_1, S_2) = \min\{dist(p, q) | p \in S_1, q \in S_2\}$.
These sets will be seperated from each other only if $dist(S_1, S_2) > Eps$.

DBSCAN : Determining the parameters

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

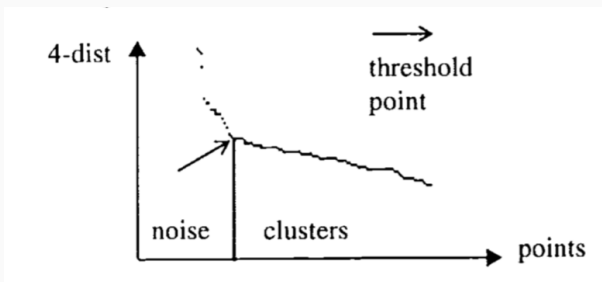


Figure 9: sorted 4-dist graph

DBSCAN : Determining the parameters

- The reason it used 4 as k is that the author found out that greater value than 4 dist graph do not differ significantly. (Set MinPts to 4)
 - Furthermore, it requires more computation using greater value. "Not worth it"
- 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 - Iris Data

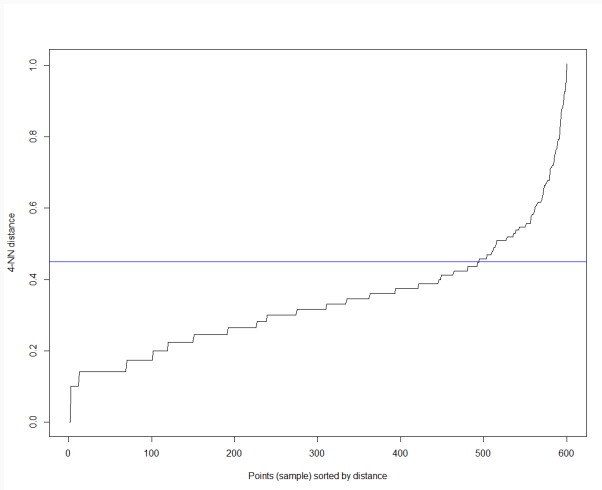
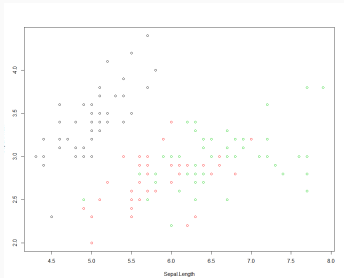
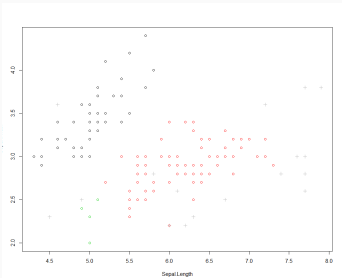


Figure 10: sorted 4-dist graph for iris

DBSCAN : Demonstration - Iris Data

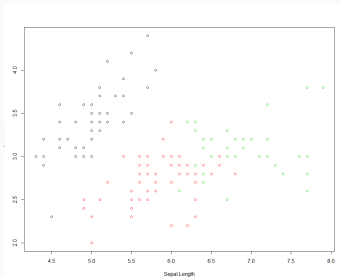


(a) True Iris Cluster

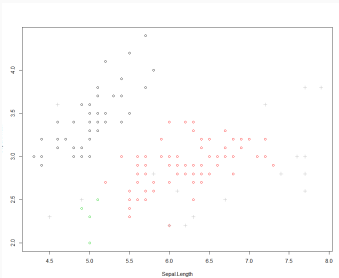


(b) DBSCAN Iris Cluster

DBSCAN : Comparison with Kmeans



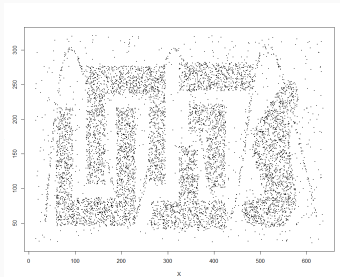
(a) Kmeans Iris Cluster



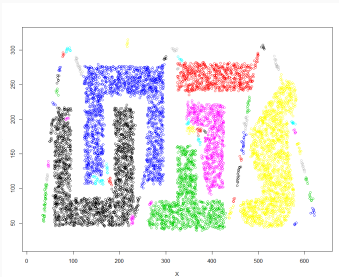
(b) DBSCAN Iris Cluster

- Can't say for sure which one is better

DBSCAN : Demonstration - DS3 Data



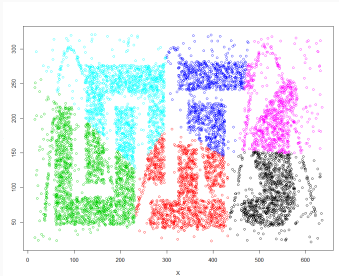
(a) True DS3 Cluster



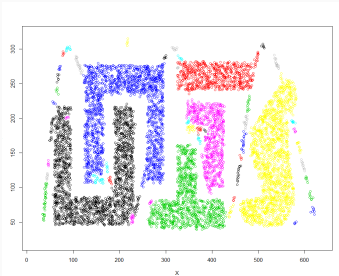
(b) DBSCAN DS3 Cluster

- 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

Hierachical Clustering

Hierarchical Clustering : Tree

- **Hierarchical 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
- Their standard algorithm has $O(n^3)$ complexity
 - There are several methods to solve it or adapts heuristic.

Tree : Cluster dissimilarity

- **Distance**

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

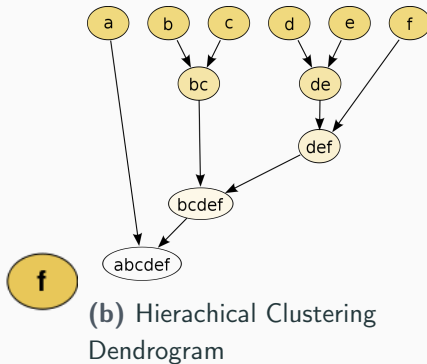
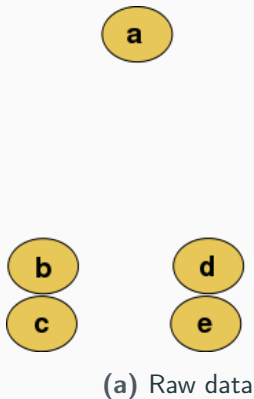
- Euclidean distance, Manhattan distance, ..

- **Linkage Criteria**

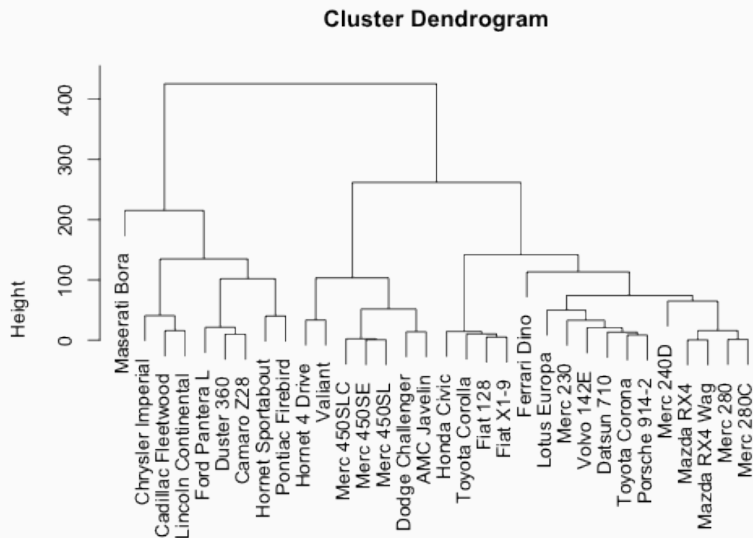
The distance between sets of observations.

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

Tree : Agglomerative



Tree : Data from mtcars in R



Spectral Clustering

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

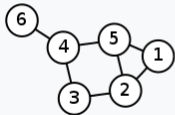
Labeled graph	Degree matrix	Adjacency matrix	Laplacian matrix
	$\begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1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

Figure 17: Making Laplacian Matrix

Spectral Clustering : Laplacian Matrix

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,] 1.000000 0 0.7429016 0.6319343 0.0000000 0.0000000 0.0000000 0
## [2,] 0.000000 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,] 0.000000 0 0.0000000 0.0000000 1.0000000 0.7765565 0.0000000 0
## [6,] 0.000000 0 0.0000000 0.0000000 0.7765565 1.0000000 0.0000000 0
## [7,] 0.6657756 0 0.7195922 0.0000000 0.0000000 1.0000000 1.0000000 0
## [8,] 0.000000 0 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 1
```

(a) Simlity Matrix using
Gaussian Kernel

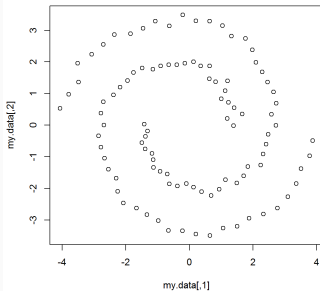
```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7]
## [1,] 2.374896 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
## [2,] 0.000000 2.597451 0.000000 0.000000 0.000000 0.000000 0.000000
## [3,] 0.000000 0.000000 2.408877 0.000000 0.000000 0.000000 0.000000
## [4,] 0.000000 0.000000 0.000000 2.351526 0.000000 0.000000 0.000000
## [5,] 0.000000 0.000000 0.000000 0.000000 2.523175 0.000000 0.000000
## [6,] 0.000000 0.000000 0.000000 0.000000 0.000000 2.519936 0.000000
## [7,] 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 3.170424
## [8,] 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
##      [,8]
## [1,] 0.000000
## [2,] 0.000000
## [3,] 0.000000
## [4,] 0.000000
## [5,] 0.000000
## [6,] 0.000000
## [7,] 0.000000
## [8,] 2.302241
```

(b) Affinity Matrix

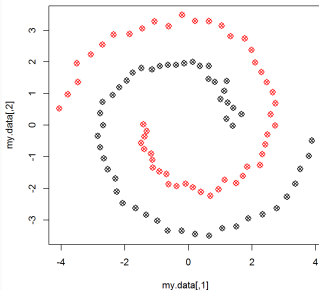
```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
## [1,] 1.4 0.0 -0.7 -0.6 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
## [2,] 0.0 1.6 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
## [3,] -0.7 0.0 1.4 0.0 0.0 0.0 -0.7 0.0 0.0 0.0 0.0 0.0
## [4,] -0.6 0.0 0.0 1.4 0.0 0.0 -0.7 0.0 0.0 0.0 0.0 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

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 : Hierarchical Clustering
-
- AY NG(2002) : On Spectral Clustering: Analysis and an algorithm