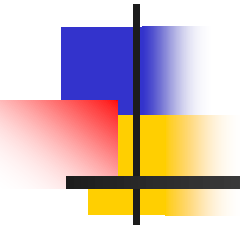
Odd Sem., AY 2024-25

Lecture: 18, 19, & 20

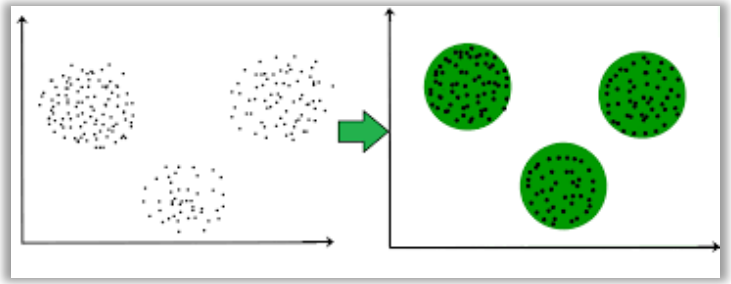
Fond. Of DS (DSPC201)



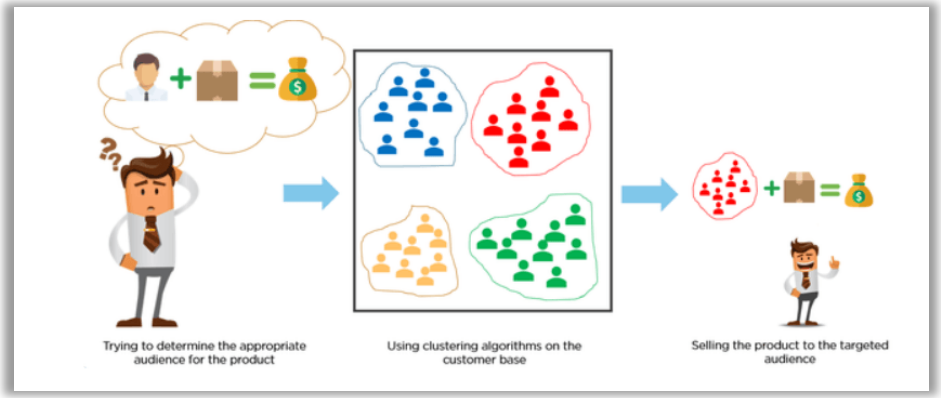
Course Instructor: Dr Vikram Singh

Data Clustering : scheme

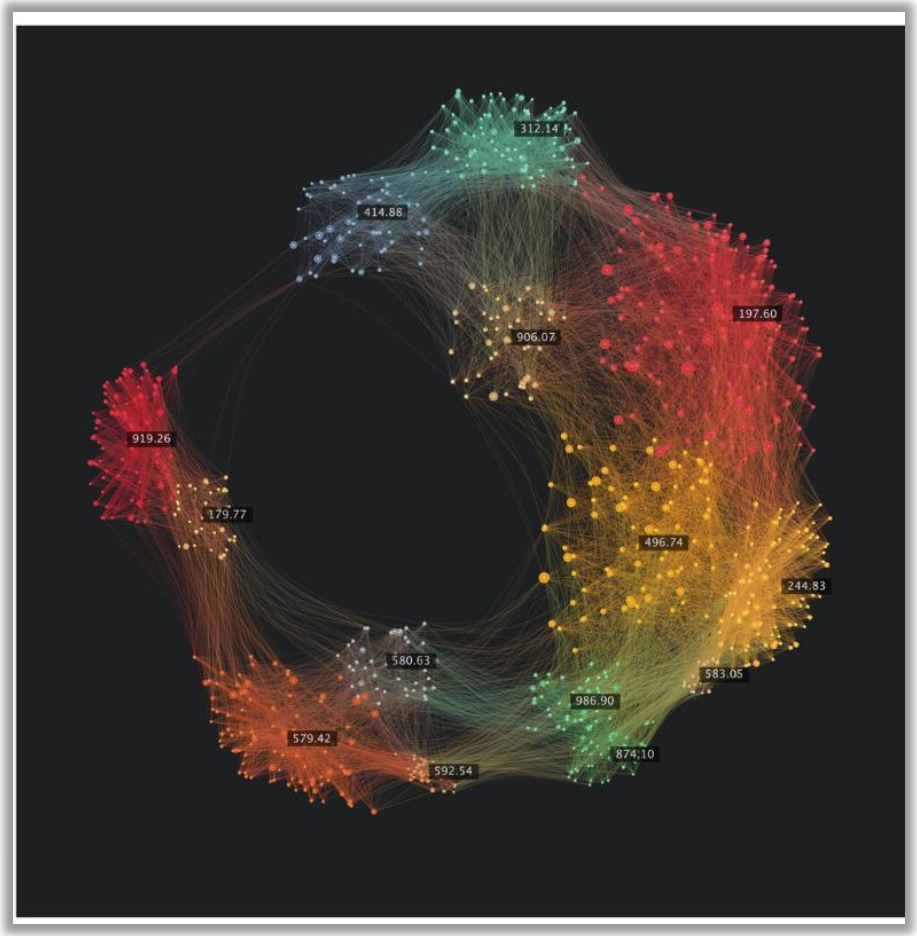
➢For Unsupervised grouping/Categorizing



➢For Decision making/ Learn by Observations



➢For effective visualization of distributions & relations of data objects



What is Cluster Analysis?

- A collection of data objects: **Cluster**
  - 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/dissimilar 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 for Data Understanding & Applications

- 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

Clustering as a Preprocessing Tool

- Summarization:
  - Preprocessing for regression, PCA, classification, and association analysis
- Compression:
  - Image processing: *vector quantization*
- Finding K-nearest Neighbors
  - Localizing search to one or a small number of clusters
- Outlier detection
  - Outliers are often viewed as those “far away” from any cluster

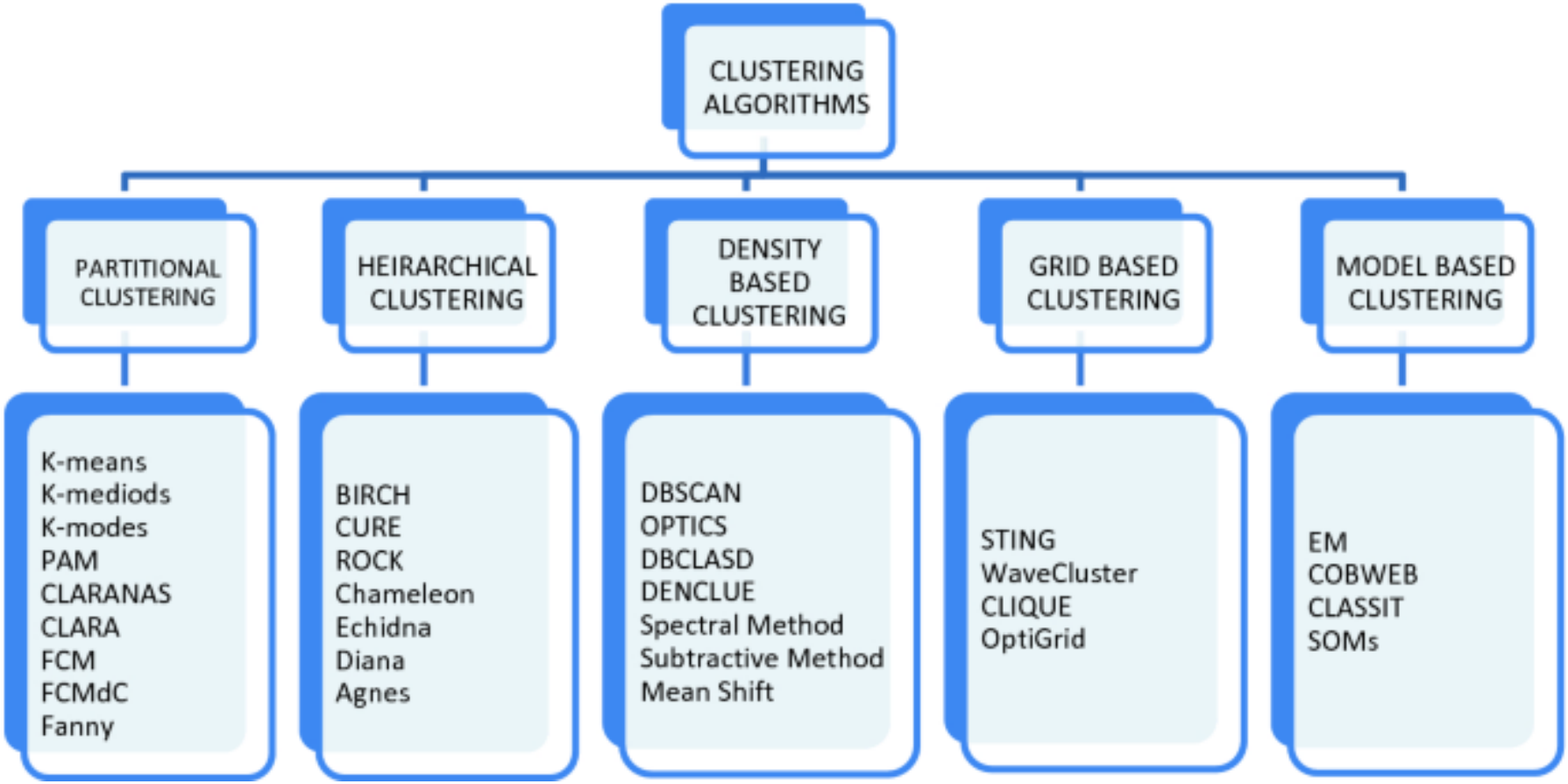
Quality: What Is Good Clustering?

- 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

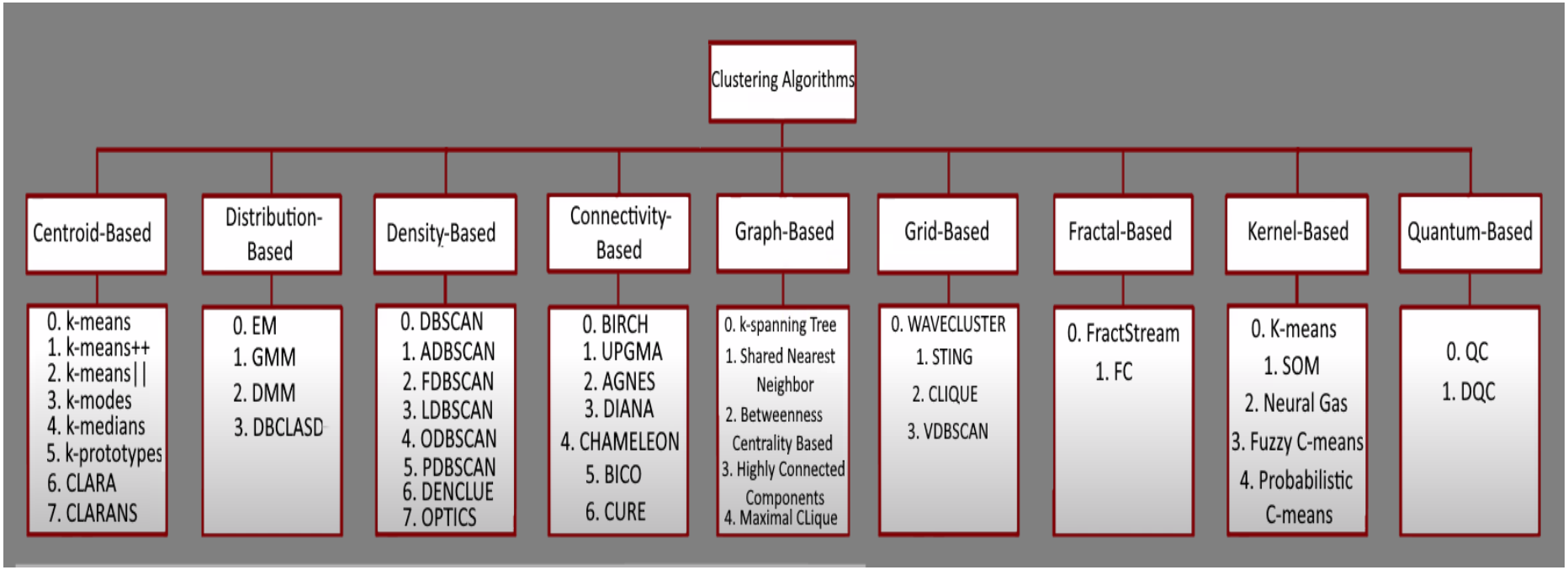
Measure the Quality of Clustering

- Dissimilarity/Similarity metric
  - Similarity is expressed in terms of a distance function, typically metric: $d(i, j)$
  - The definitions of distance functions are usually rather different for **interval-scaled, boolean, categorical, ordinal ratio, and vector variables**
- Quality of clustering:
  - There is usually a separate "quality" function that measures the "goodness" of a cluster.
  - It is hard to define "similar enough" or "good enough" : The answer is typically highly subjective.

Clustering : **Approaches & techniques/ Algos**



Clustering : **Approaches & techniques/ Algos**



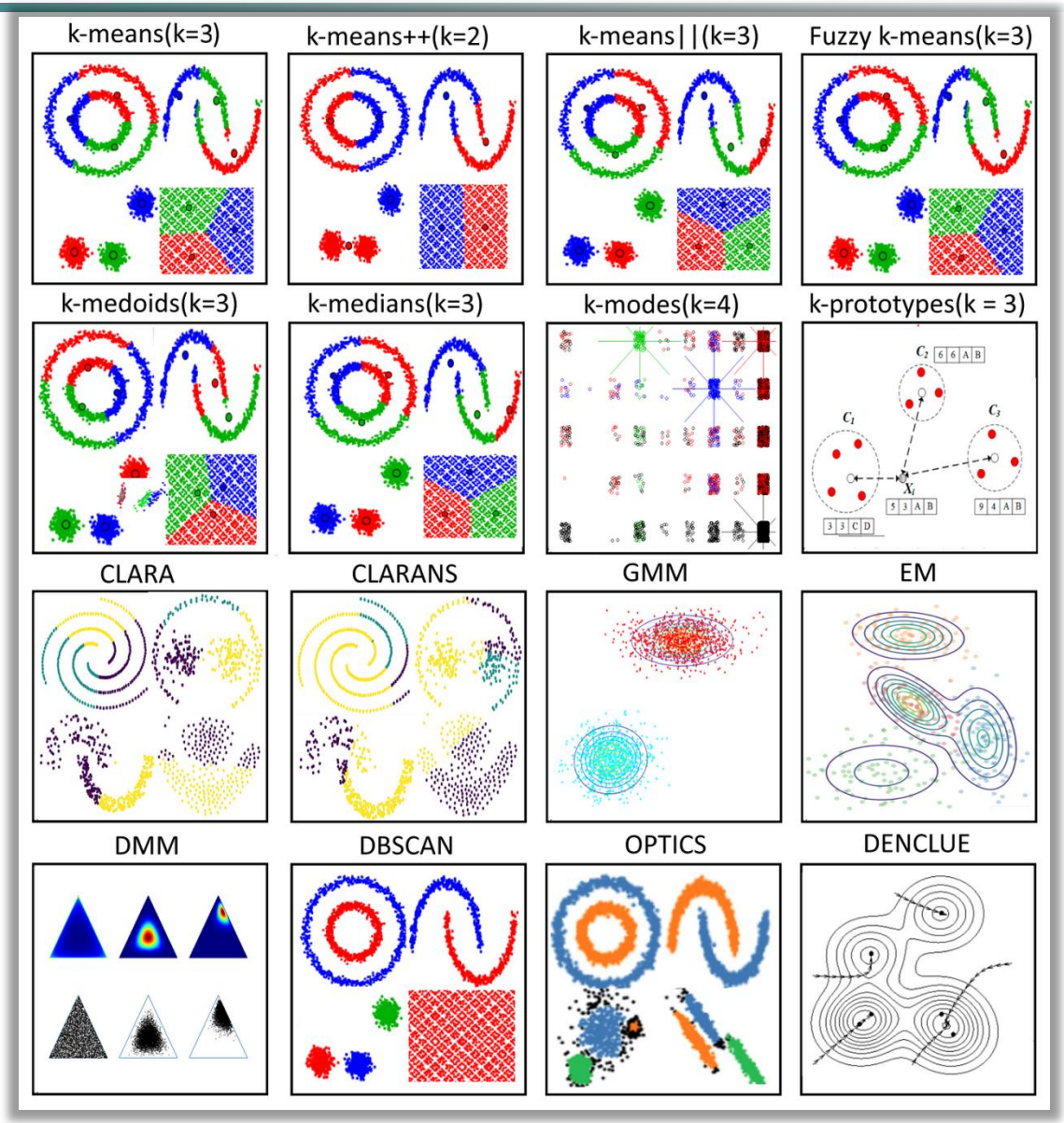
Clustering Approaches

- **Partitioning approach**:
  - 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*
- **Hierarchical approach**:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: *Diana, Agnes, BIRCH, CAMELEON*
- **Density-based approach**:
  - Based on connectivity and density functions
  - Typical methods: *DBSACN, OPTICS, DenClue*
- **Grid-based approach**:
  - based on a multiple-level granularity structure
  - Typical methods: *STING, WaveCluster, CLIQUE*

Clustering Approaches

- **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

Cluster Formation: **Cluster Shape & Feasibility**



Considerations for Cluster Analysis

- Partitioning criteria
  - Single level vs. hierarchical partitioning (often, multi-level hierarchical partitioning is desirable)
- Separation/Overlap 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)

Similarity /Dissimilarity measures

- Also referred as *Distance*
- Distance Based *(e.g. Euclidian, road network, etc).*
- Connectivity Based *(e.g. density, continuty, etc.)*
- More working examples:
  - click here
  - click here

Similarity and Dissimilarity: Distance

- **Similarity**
  - Numerical measure of how alike two data objects are
  - Value is higher when objects are more alike
  - Often falls in the range [0,1]
- **Dissimilarity** (e.g., distance)
  - Numerical measure of how different two data objects are
  - Lower when objects are more alike
  - Minimum dissimilarity is often 0
  - Upper limit varies
- **Proximity** refers to a similarity or dissimilarity

Similarity & Dissimilarity Estimation: Data Matrix

Data matrix : (or object-by-attribute structure): This structure stores the n data objects in the form of a relational table, or n-by-p matrix (n objects p attributes):

- n data points with p dimensions
- Two modes *(contains Feature and Values of each objects)*

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}.$$

Similarity & Dissimilarity Estimation: Dissimilarity matrix

Dissimilarity matrix: (or object-by-object structure): This structure stores a collection of proximities that are available for all pairs of n objects. It is often represented by an n-by-n table:

- n data points, but registers only the distance
- Is s close to 0 when objects *i and j are* highly similar or near" each other.
- A triangular matrix
- Single mode *(contains distance values)*

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix},$$

Similarity/Dissimilarity for Nominal Attribute:

- Nominal Attributes (Non-numeric) : Values States
- Two or more states
- e.g. *Color names: Blues, red, Yellow, etc.*
- Distance is referred as **Proximity**

Similarity & Dissimilarity Estimation

For Nominal Attribute

- For Nominal Attributes
  - Similarity / Dissimilarity Measure is referred as '**Proximity**'.
  - A nominal attribute can take on two or more states.
  - **e.g.** *map color is a nominal attribute that may have, say, five states: red, yellow, green, pink,* and *blue.*

Object Identifier	test-1 (nominal)
1	code A
2	code B
3	code C
4	code A

$$d(i, j) = \frac{p - m}{p},$$

With p is no. of nominal attributes & m is no. of matches

Dissimilarity Matrix

$$\begin{bmatrix} 0 & & & \\ d(2,1) & 0 & & \\ d(3,1) & d(3,2) & 0 & \\ d(4,1) & d(4,2) & d(4,3) & 0 \end{bmatrix}.$$

$$\begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ 1 & 1 & 0 & \\ 0 & 1 & 1 & 0 \end{bmatrix}.$$

All object are dissimilar, except d(4,1)

Similarity/Dissimilarity for Binary Attribute:

- Binary attribute (Non-numeric)
- Only two states in attribute.
- Required **Contingency table**
- **Two Types: Symmetric Binary & Asymmetric Binary.**
- e.g. Male & Female, 1 &0, Y/N
- Distance is referred as **Proximity**

Similarity & Dissimilarity Estimation

For Binary Attribute

- For Binary Attributes
  - Symmetric *(equal weights to both values, e.g Male/Female).*
  - Asymmetric nature *(one value is of lesser importance, 1/0).*
  - *Contingency table is required, to represent the* organize and compare the frequencies of occurrences for different combinations of attribute values between two objects, w.r.t. similar or dissimilar instances.

Relational Table Where Patients Are Described by Binary Attributes

name	gender	fever	cough	test-1	test-2	test-3	test-4
Jack	M	Y	N	P	N	N	N
Jim	M	Y	Y	N	N	N	N
Mary	F	Y	N	P	N	P	N
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Contingency Table for Binary Attributes

		Object j		
		1	0	sum
Object i	1	q	r	$q+r$
	0	s	t	$s+t$
	sum	$q+s$	$r+t$	p

Dissimilarity (on Symmetric Bin Attribute)

$$d(i,j) = \frac{r+s}{q+r+s+t}.$$

Similarity (on Asymmetric Bin Attributes)

$$sim(i,j) = \frac{q}{q+r+s} = 1 - d(i,j).$$

q *is the number of attributes that equal 1 for both data objects i and j, etc*

Similarity/Dissimilarity for Ordinal attribute:

- Ordinal attribute (Non-numeric)
- meaningful order or ranking about them,.
- e.g. for a Size attribute *small, medium, large.*
- thus required **Normalization** of ranked values into

States/ Categories, in [0.0, 1.0] range.

Similarity & Dissimilarity Estimation

For Ordinal Attribute

Dataset

Data object	Attribute
1	Excellent
2	Fair
3	Good
4	Excellent

Step-1: Discertization

Discertization (mapping vales into Ranks/Numeric)	
Fair	1
Good	2
Excellent	3

↓

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}.$$

↓

Step-2:**Normalization**

Normalizing the rank (between 0.0 to 1.0)	
Rank	***Value***
1	0
2	0.5
3	1

Dissimilarity Matrix

$$\begin{bmatrix} 0 & & & \\ 1.0 & 0 & & \\ 0.5 & 0.5 & 0 & \\ 0 & 1.0 & 0.5 & 0 \end{bmatrix}$$

Similarity and Dissimilarity of Numeric attributes:

- Numeric attribute:
- Distance Based *(e.g. Euclidian, road network, etc).*
- Connectivity Based *(e.g. density, continuity, etc.)*
- In some cases, Data is to be normalized as well (Ordinal Attribute).

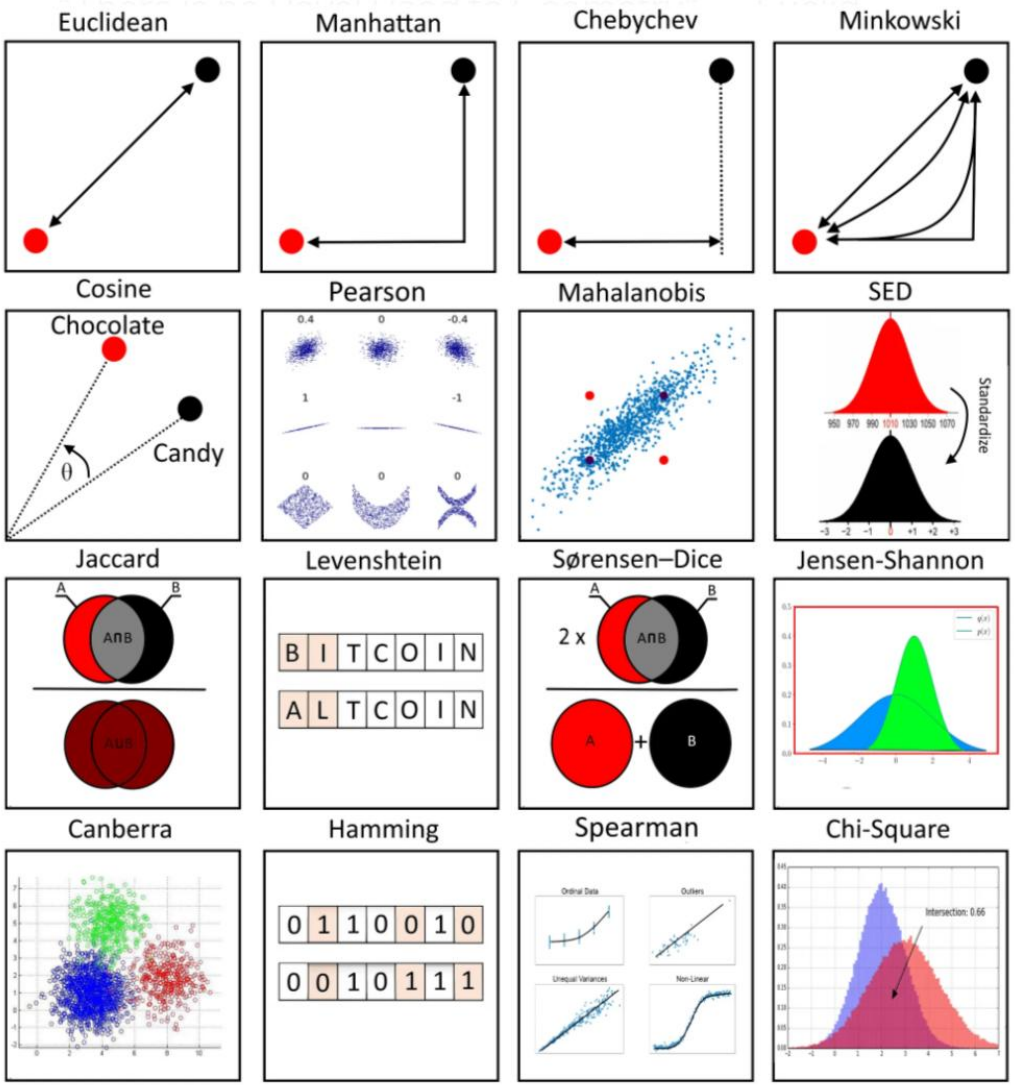
Similarity & Dissimilarity Estimation

For Numeric Attribute

- For Distance functions :
  - *Euclidean Distance*
  - *Manhattan Distance*
  - *Minkowski Distance*
  - *Hamming Distance*
  - *Cosine Similarity,*
  - *Jaccard Sim. etc.*
- Click here to see example

Similarity & Dissimilarity Estimation

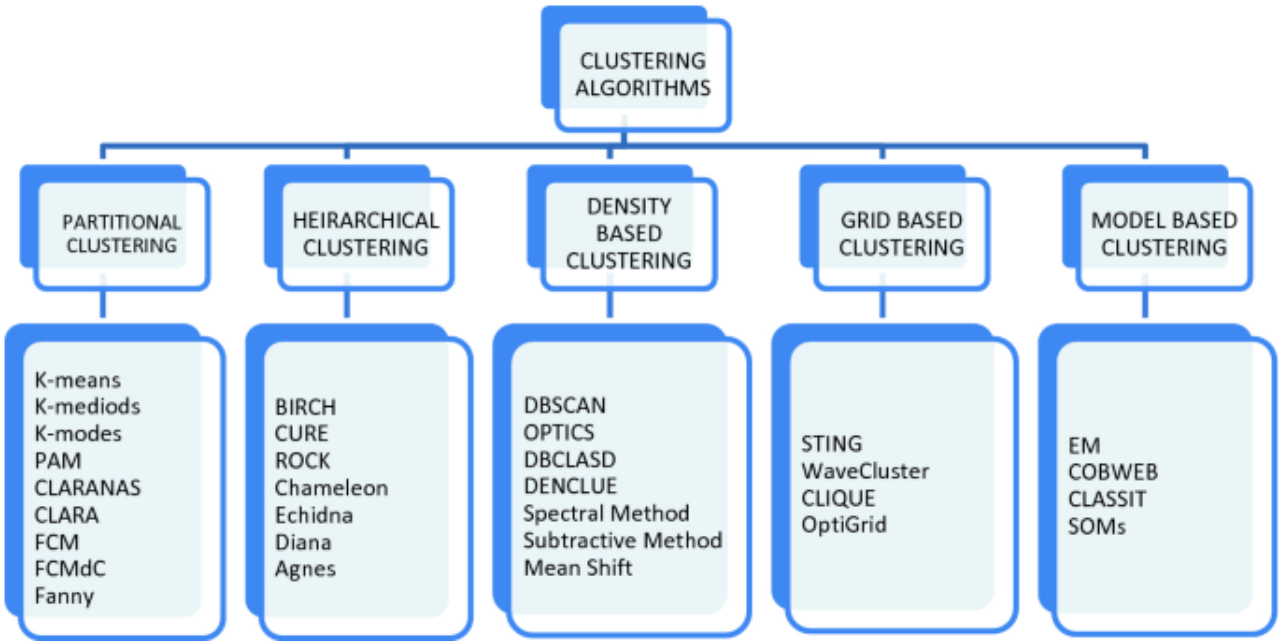
Distance Function scheme



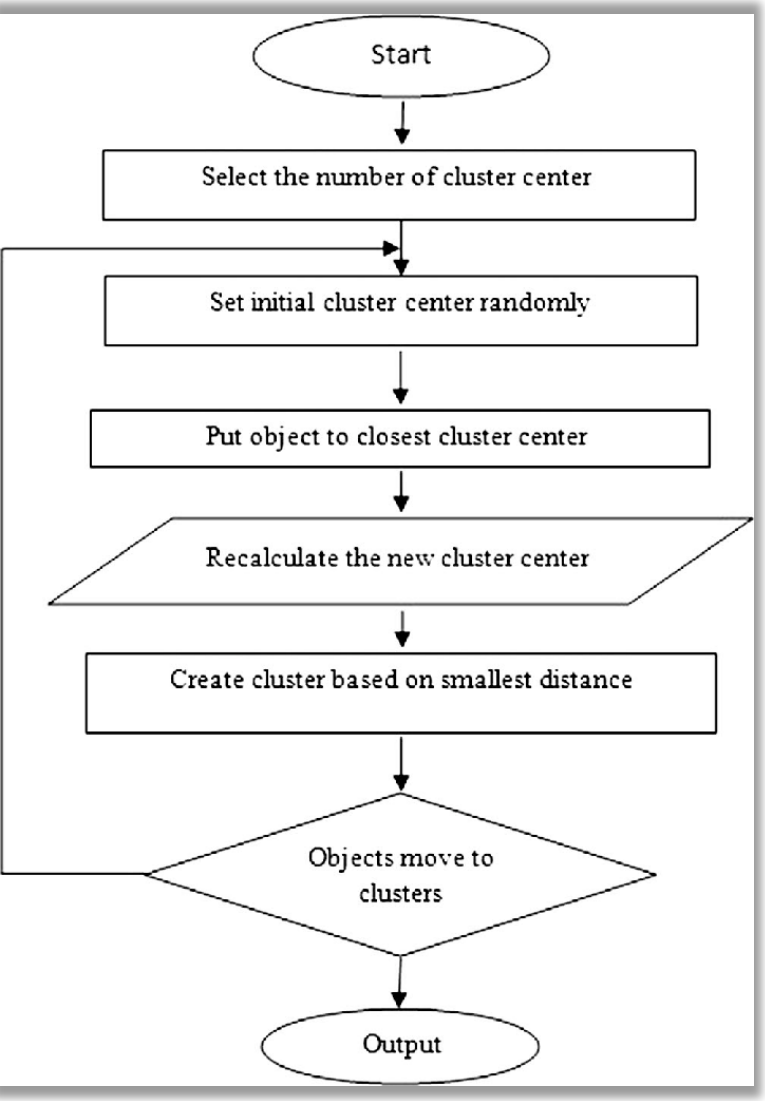
- Click here read more

Partitioning-based Clustering Approach

- Based on partitioning of data objects into smaller and cohesive data groups/categories.
- Data analyst supply the no. of required Groups/Cluster.
- *Techniques:*



K-Means Clustering



- Randomly assign K objects from the dataset(D) as cluster centres(C).
- (Re) Assign each object to which object is most similar based upon mean values.
- Update Cluster means, i.e., Recalculate the mean of each cluster with the updated values.
- Repeat Step 2 until no change occurs

Click here to see an example

K-Means Clustering

Step-I : Choose the number of clusters (K): define the *clusters (K) of interest.*

Step-II: Initialize cluster centroids: *Randomly select **K** points from the dataset* to serve as the initial centroids (mean points) of the clusters. *These centroids represent the center of each cluster.*

Step-III: Assign each point to the nearest centroid (cluster assignment step): For each data point in the dataset, calculate its distance to each of the **K** centroids. ***Assign the point to the nearest centroid.*** This forms **K** clusters where each point belongs to the cluster of the closest centroid.

Step-IV: 4. Recalculate the centroids (update step): Once all points are assigned to clusters, update the centroids of each cluster. *The new centroid is calculated by taking the mean (average) of all the points assigned to that cluster. This step moves the centroids to a new position.*

Step-V: Repeat steps 3 and 4:

Repeat the process of assigning points to clusters and recalculating centroids.

This continues until:

> **Convergence:** No point changes its cluster assignment, or
>
> The change in the positions of the centroids is minimal (i.e., the centroids stop moving significantly).

Step-VI: Final clusters: After convergence, *the clusters are considered final, and each point is now grouped with others that are most similar to it, based on the chosen distance metric.*

K-Means Clustering: Working Example

Given Data Points: (2,10),(2,5),(8,4),(5,8),(7,5),(6,4),(1,2),(4,9)(2,10), (2,5), (8,4), (5,8), (7,5), (6,4), (1,2), (4,9)(2,10),(2,5),(8,4),(5,8),(7,5),(6,4),(1,2),(4,9)

Step 1: Choose Initial Centroids

Let's assume **K = 2** (two clusters) and the initial centroids are:

C1=(2,10),C2=(5,8) C_1 = (2,10) and C_2 = (5,8)

Step 2: Compute Euclidean Distance

Step 3: Recalculate Centroids

For **C1**, take the mean of cluster points:

C1′=Mean of points in

For **C2**, take the mean of its cluster points:

Continue

*This **iteratively** until centroids don't change.*

The **Euclidean Distance** formula:

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

Compute the distance of each point to centroids C_1 and C_2:

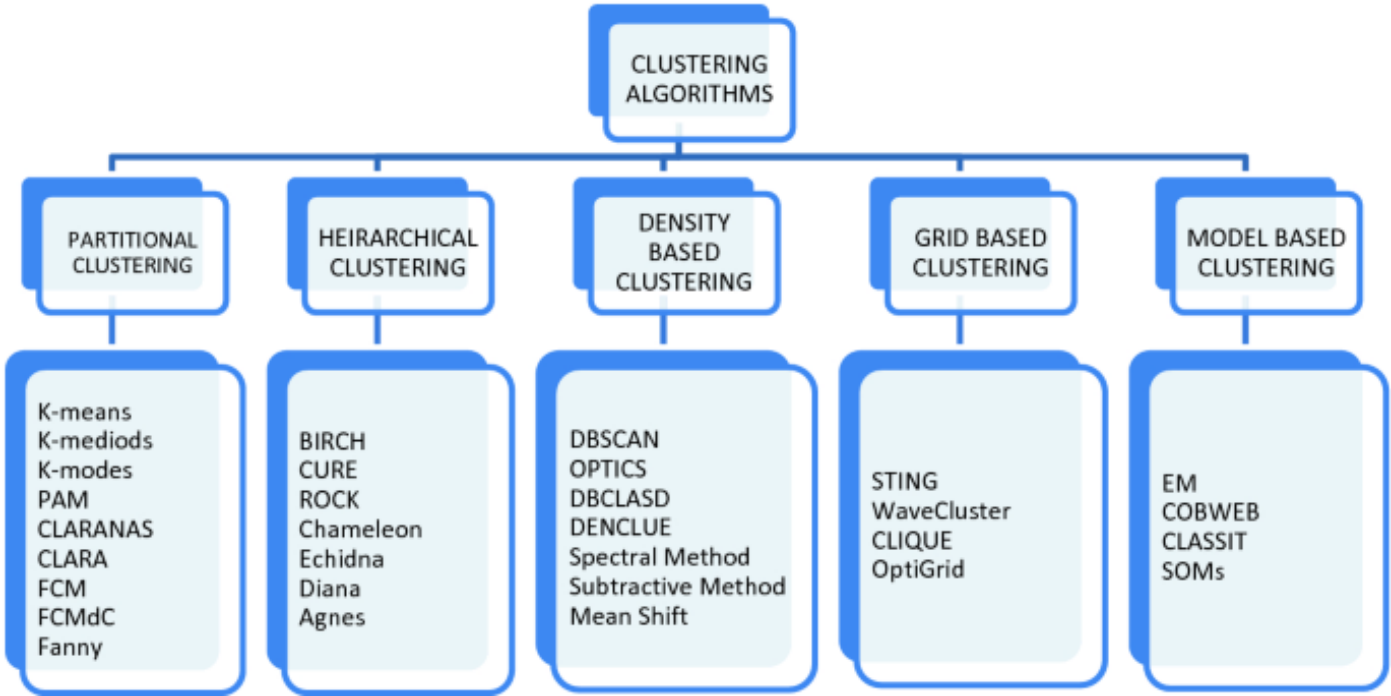
Point	Distance to $C_1(2, 10)$	Distance to $C_2(5, 8)$	Cluster
(2,10)	0.0	3.61	C1
(2,5)	5.0	3.61	C2
(8,4)	9.43	3.61	C2
(5,8)	3.61	0.0	C2
(7,5)	7.81	2.83	C2
(6,4)	8.94	4.47	C2
(1,2)	8.06	7.28	C2
(4,9)	2.24	1.41	C2

K-Means Clustering (limitation)

- **Initial centroid selection:** The algorithm can converge to a suboptimal solution if the initial centroids are not selected appropriately.
- **Outliers:** K-means is sensitive to outliers, which can skew the clusters.
- **Number of clusters**: It can be difficult to choose the optimal number of clusters, or value of K, for a given dataset. Different values of K can lead to different results.
- **Cluster shape**: K-means assumes that clusters are spherical and have similar variance, which *may not be suitable for complex or irregular clusters*.
- **Cluster size**: K-means may struggle to capture the structure of the data when clusters are of different sizes.
- **Scaling**: K-means is sensitive to scaling.
- **Repeatability:** K-means does not produce the same result every time.
- **Random seeds**: Different random seeds can cause the algorithm to converge to different local minima instead of the global minimum.

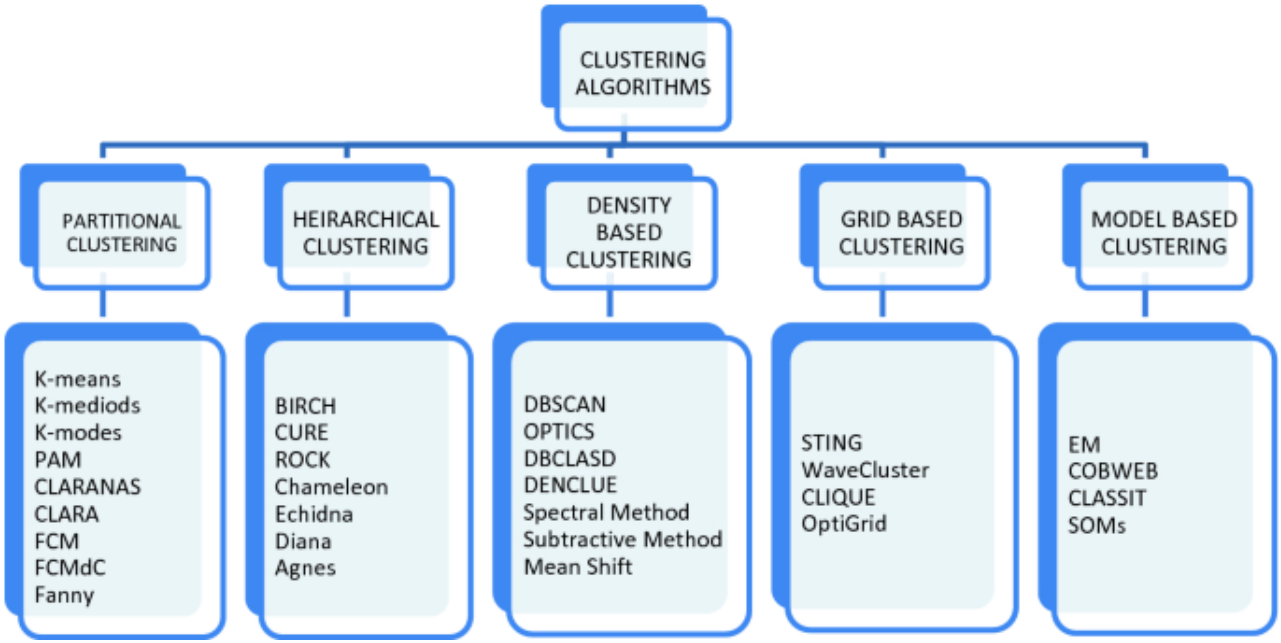
Partitioning Clustering approach

- ***C-Fuzzy Means Clustering*** : Read Assignment
- C-Fuzzy Means (FCM), is a clustering algorithm similar to K-Means but with **soft clustering**. Instead of assigning each data point to a single cluster, **FCM assigns membership probabilities** to all clusters.

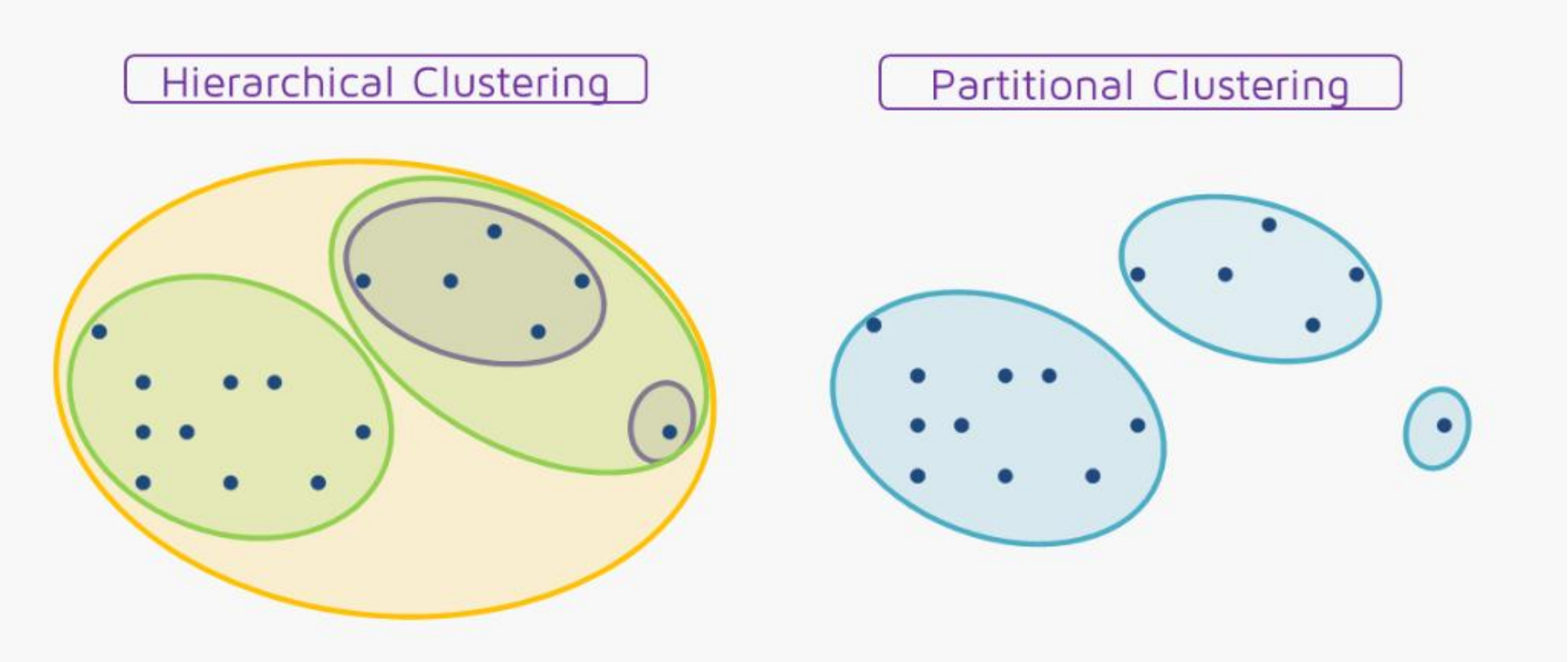


Hierarchical Clustering approach

- Create a *hierarchical decomposition* of the set of data (or objects) using some criterion
- Typical methods: ***Diana, Agnes, BIRCH, CAMELEON***



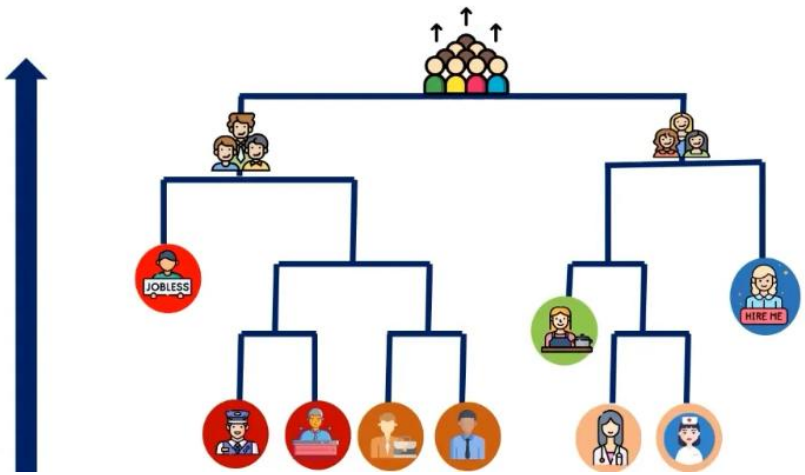
Partitioning Clustering approach Vs Hierarchical Clustering

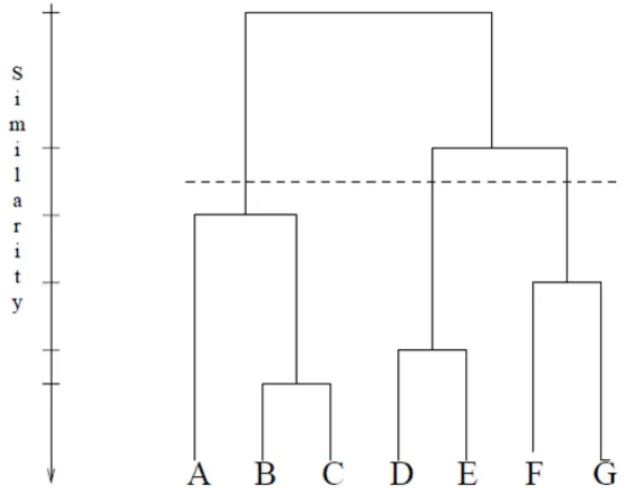


https://quantdare.com/hierarchical-clustering/

Hierarchical Clustering approach

- Is an unsupervised learning technique used for grouping data points into clusters ***without specifying the number of clusters beforehand***.
    - It builds a tree-like structure called a **dendrogram** that visually represents **data grouping at different levels of similarity.**
- Types: Agglomerative (Bottom-Up) & Divisive (Top-Down)





Hierarchical Clustering approach:

Agglomerative Vs Divisive

Agglomerative (Bottom-Up Approach)

- Each data point starts as its own cluster.
- Clusters **merge iteratively** based on similarity.
- The process continues until all points belong to a single cluster.

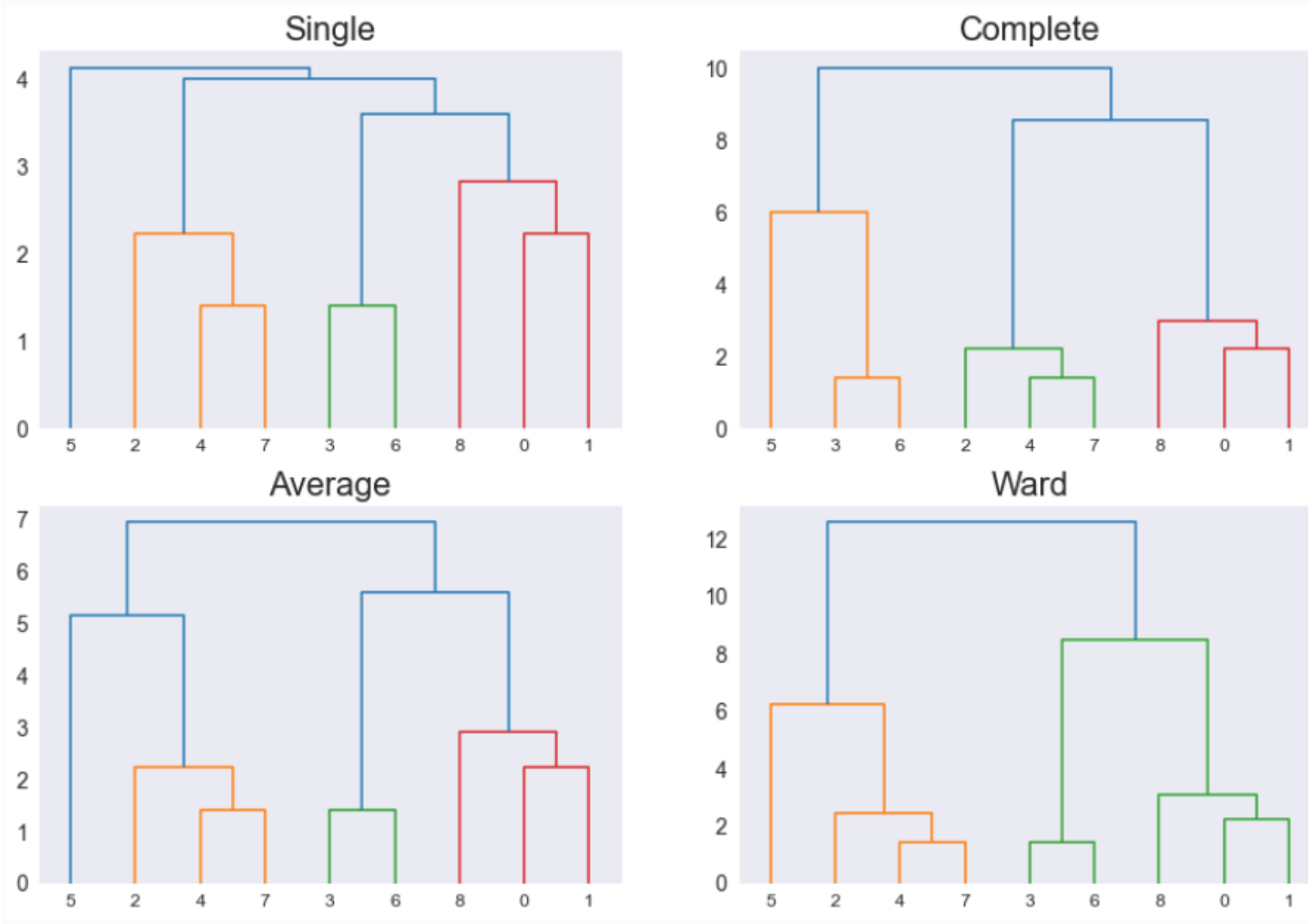
- **Common linkage methods**:
  - **Single Linkage**: Minimum distance between points in clusters.
  - **Complete Linkage**: Maximum distance between points.
  - **Average Linkage**: Mean distance between points.
  - **Ward's Method**: Minimizes variance within clusters.

Divisive (Top-Down Approach)

- Starts with **one large cluster** containing all data points.
- Splits iteratively into smaller clusters.
- Rarely used compared to **Agglomerative**.

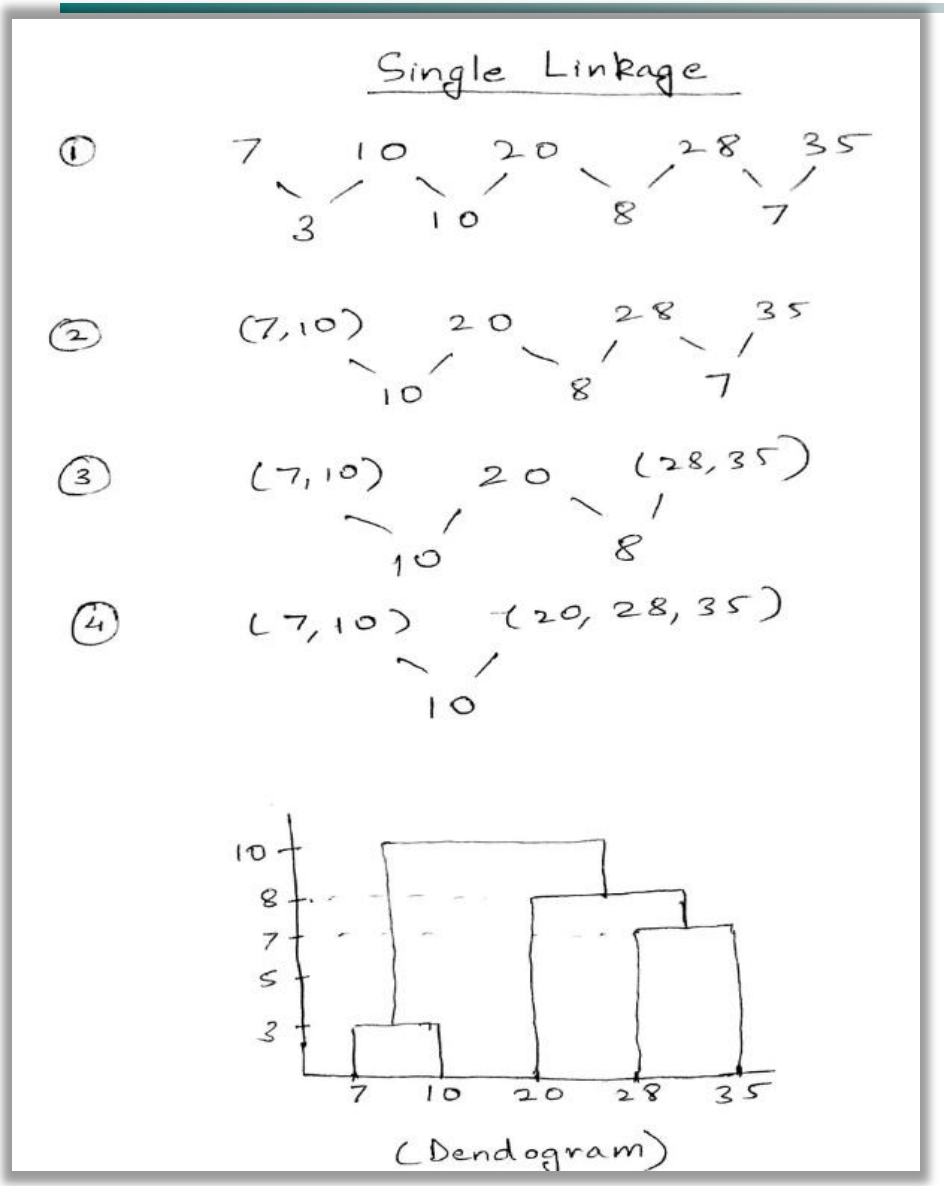
Hierarchical Clustering approach

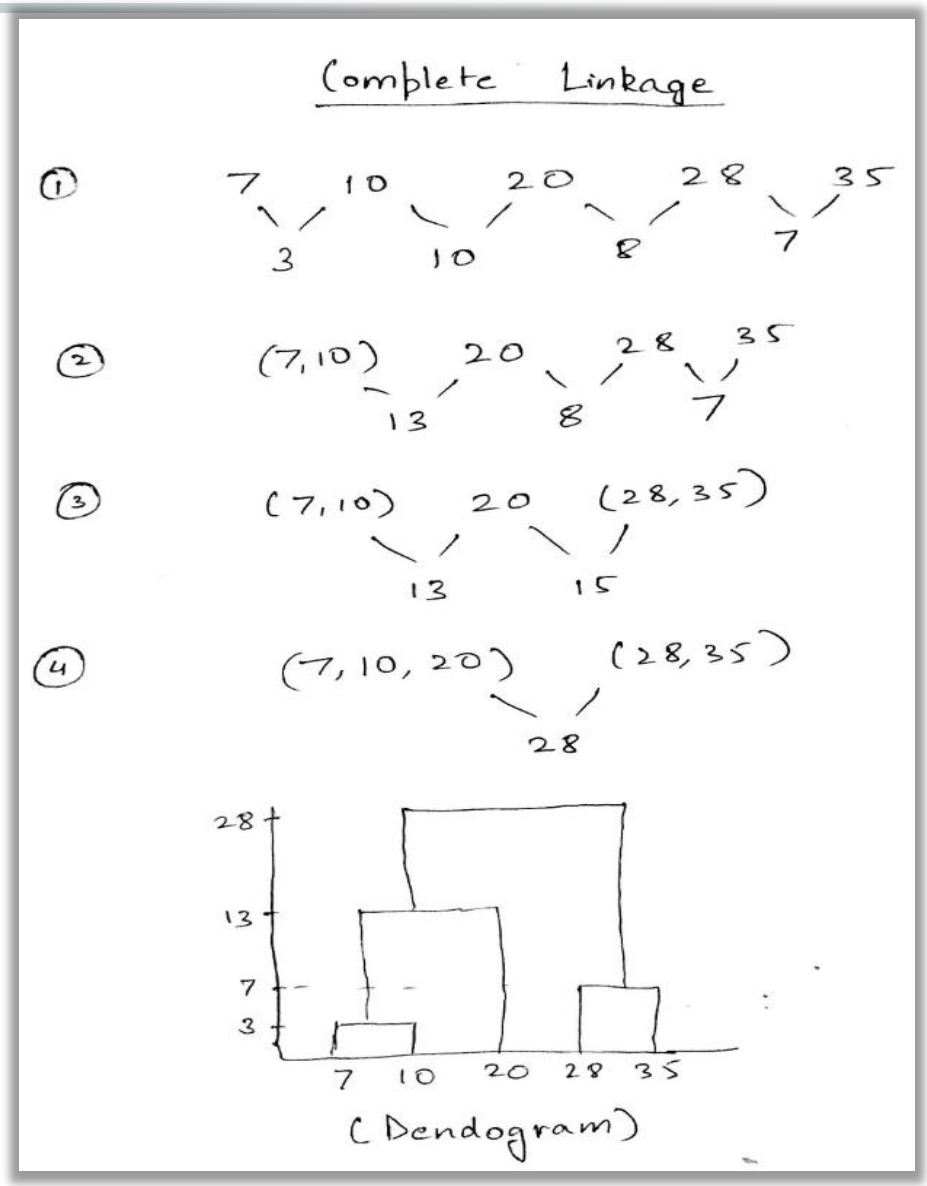
- Agglomerative (Bottom-Up Approach)



Hierarchical Clustering approach: Agglomerative

Single-linkage Vs Complete linkage





Hierarchical Clustering approach:

Agglomerative

Step-I: Initialization: Treat each data point as a separate cluster. Thus, if you have N data points, you start with N clusters, each containing just one data point.

Step-II: Compute Distance Matrix: Calculate the distance between each pair of clusters. *The choice of distance metric can significantly affect the outcome of the clustering*. This results in an N×N distance matrix, where the distance between a cluster and itself is zero.

Step-III: Find the Closest Clusters: Identify the two clusters that are closest to each other based on the distance matrix.

Step-IV: Merge Clusters: Combine the two closest clusters into a single cluster and this steps reduces the total number of clusters by one.

StepV: Update Distance Matrix: Recalculate the distances between the new cluster and all the existing clusters and the method of recalculating the distance depends on the **linkage criterion** used.

Hierarchical Clustering approach: Agglomerative

Single-linkage Vs Complete linkage

Working Examples

https://medium.com/@sachinsoni600517/mastering-hierarchical-clustering-from-basic-to-advanced-5e770260bf93

https://medium.com/@rohanjoseph_91119/learn-with-an-example-hierarchical-clustering-873b5b50890c

https://www.learndatasci.com/glossary/hierarchical-clustering/

Hierarchical Clustering approach (limitations)

Limitation	Description	Impact
Dependent on Linkage Method	Different linkage criteria (single, complete, average, Ward's) affect results.	**Choosing the wrong linkage may lead to incorrect clusters**
No Reassignment of Points	Once a point is merged into a cluster, it cannot move.	**Not flexible like K-Means or FCM**
Imbalanced Cluster Sizes	Some clusters may contain very few points while others are large.	**Leads to biased clustering**
Computationally Expensive for High Dimensions	Distance calculation in high-dimensional spaces becomes inefficient.	**Not suitable for large feature spaces (e.g., images, text)**
High Computational Cost	Requires computing pairwise distances, leading to O(n2)O(n^2)O(n2) or worse complexity.	**Slow for large datasets (>10,000 samples)**
High Memory Usage	Stores a distance matrix of O(n2)O(n^2)O(n2) size.	**Consumes large memory for big data**
Difficult to Choose Clusters (K)	Requires manual interpretation of dendrogram.	**No automatic way to determine K**
Sensitive to Noise & Outliers	Outliers can distort clustering results.	**Poor handling of noisy data**

Comparison

Partitioning *(K-means)* vs Hierarchical *(Agglomerative)*

Feature	Hierarchical Clustering	K-Means Clustering
Cluster Shape	Any shape (depends on linkage)	Spherical clusters
Number of Clusters (K)	No need to predefine	Must be specified
Handles Outliers?	**No, sensitive to outliers**	**No, outliers distort centroids**
Memory Usage	**High** (Stores distance matrix)	**Low** (Stores only centroids)
Cluster Assignment	**Once merged, cannot move**	**Can reassign in iterations**
Works on Large Datasets?	**No, due to high complexity**	**Yes, efficient for large data**
Works on Noisy Data?	**No, easily affected**	**No, sensitive to noise**
Dendrogram Visualization?	**Yes** (Great for hierarchical structure)	No
Best Used For	Small datasets, exploratory analysis	Large datasets with well-defined clusters