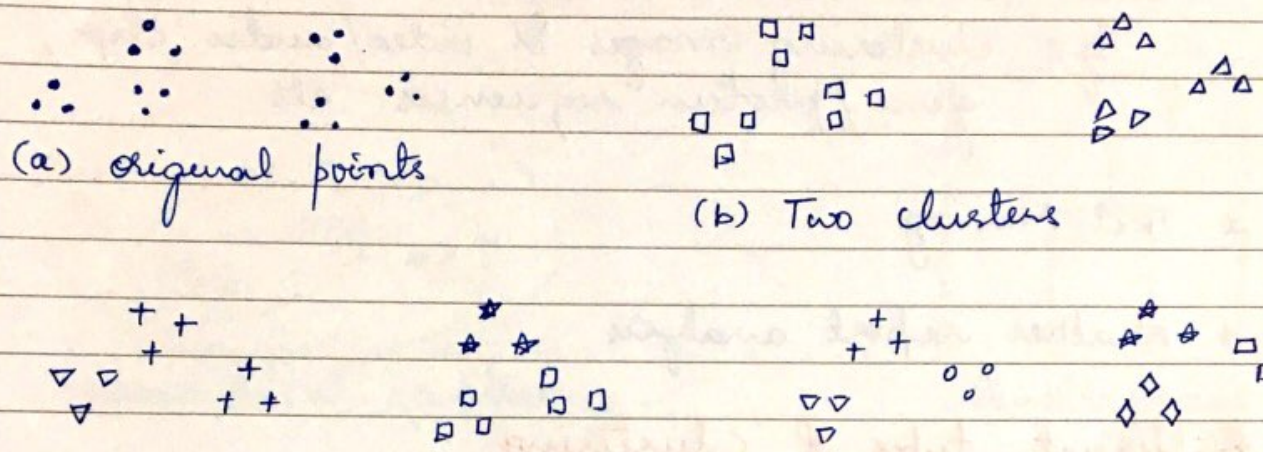
Unit - 4 : CLUSTERING

Date No.

What is cluster Analysis ?

Cluster Analysis groups data objects based only on information found in the data that describes the objects and their relationships



(a) Original points

(b) Two clusters

(c) Four clusters

(d) Six clusters

fig: Different ways of clustering the same set of points

Cluster Analysis Applications

* A Key intermediate step for other data mining tasks
  - Generating a compact summary of data for classification, pattern discovery, hypothesis generation and testing. etc
  - Outlier detection: outliers - those "far away" from any cluster

* Data Summarization, compression and reduction
  Eg:- Image processing : Vector quantization

* Collaborative filtering, recommendation systems, or customer segmentation
  - Find like-minded users or similar products

* Dynamic trend detection
  - clustering stream data and detecting trends and patterns

* Multimedia data analysis, biological data analyses and Social Network Analysis
  Eg:- clustering images or video/audio clips, gene/protein sequences etc

* Text Mining

* Weather report analysis

Different types of clustering

Entire collection of clusters is commonly referred to as clustering

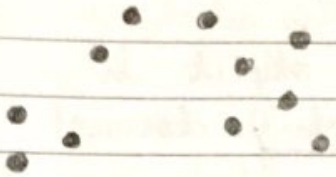
Various types of clusterings are
→ Hierarchical Versus Partitional
→ Exclusive versus overlapping versus Fuzzy
→ Complete versus Partial

Hierarchical versus Partitional

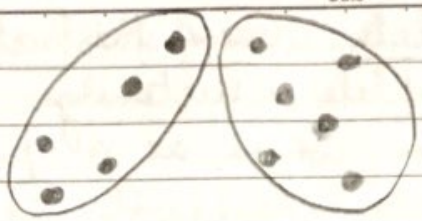
The most common distinction among different types of clustering is whether the sets of clusters are nested or unnested

A partitional clustering is simply a division of the set of data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset

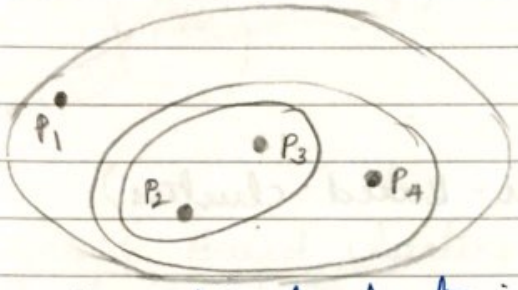
Hierarchical clustering is a set of nested clusters that are organized as a tree



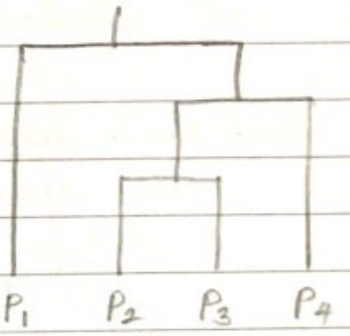
Original points



A partitional clustering



Hierarchical clustering



Dendrogram

fig: Hierarchical V/s Partitional Clustering

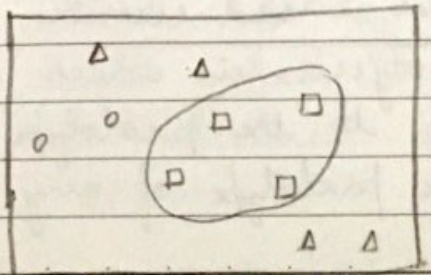
Exclusive versus Overlapping versus Fuzzy clustering

Exclusive clustering assign each object to a single cluster

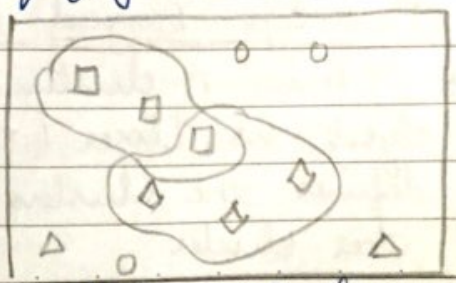
Non-Exclusive / Overlapping clustering is used to reflect the fact that an object can simultaneously belong to more than one group (class)

Eg:- Person at a University can be both an enrolled student & an employee of the University

In a Fuzzy clustering, every object belongs to every cluster with a membership weight that is between 0 (absolutely doesn't belong) and 1 (absolutely belongs) i.e, clusters are treated as fuzzy ~~sets~~ sets



Exclusive clustering



overlapping clustering

Complete versus Partial Clustering

Complete clustering assigns every object to a cluster, where as a partial clustering doesnot

The motivation for a partial clustering is that some objects in a dataset may not belong to well-defined groups

Different types of clusters

Well Separated
Prototype-Based (Center-Based clusters)
Graph-Based
Density-Based
Shared Property (Conceptual clusters)

Well-Separated Clusters

A cluster is a set of objects in which each object is closer (or more similar) to every other object in the cluster than to any object not in the cluster.

Sometimes, a threshold is used to specify that all the objects in a cluster must be sufficiently close (or similar) to one another

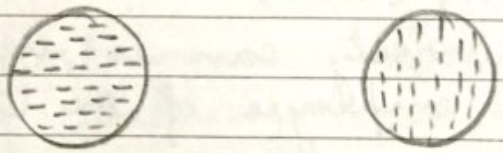


fig: well separated cluster: Each point is closer to all of the points in its cluster than to any point in another cluster

Prototype-Based clusters / Center based clusters

* Here, A cluster is a set of objects in which each object is closer (more similar) to the prototype that defines the clusters than to the prototype of any other cluster

Date No.

* For data with continuous attributes, the prototype of a cluster is often a centroid (mean of all the points in the cluster)
* For data with categorical attributes, the prototype is often a medoid (most representative point of a cluster)

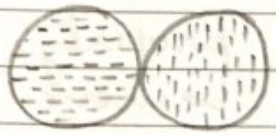


fig: center-based clusters: Each point is closer to the center of its cluster than to the center of any other cluster

Graph-Based Clusters

* If the data is represented as a graph, where the nodes are objects and the links represent connections among objects, then a cluster can be defined as a connected component i.e; a group of objects that are connected to one another, but that have no connection to objects outside the group

* Example for graph based clusters are Contiguity-based clusters, where two objects are connected only if they are within a specified distance of each other.

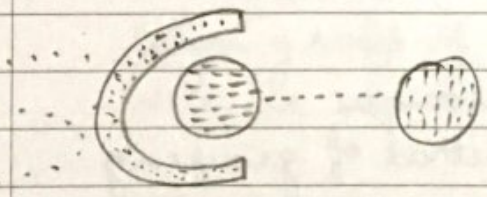


fig:- Contiguity based clusters: Each point is closer to atleast one point in its cluster than to any point in another cluster

Density-Based Clusters

* Here, a cluster is a dense region of objects that is surrounded by a region of low density.
* Used when the clusters are irregular or intertwined and when noise & outliers are present

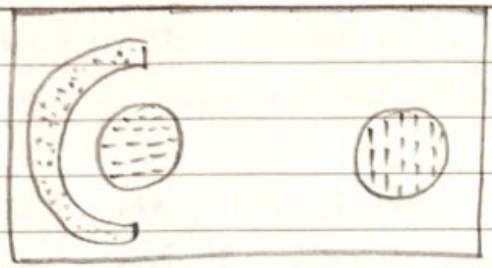


fig: Density-Based clusters: clusters are regions of high-density separated by regions of low density.

Shared property / Conceptual clusters

More generally, we can define a cluster as a set of objects that share some property. this definition encompasses all the previous definitions of a cluster

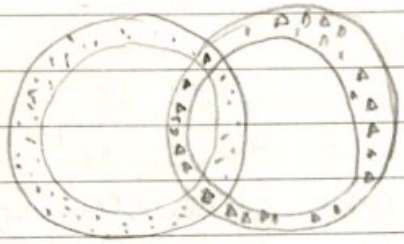


fig: Conceptual clusters: Points in a cluster share some general property that derives from the entire set of (points in the intersection of the circles belongs to both)

K-Means

* Exploratory data analysis technique
* Implements non hierarchical method of grouping objects together
* Determines the centroid using the Euclidean method for distance calculation
* Groups the objects based on minimum distance

K-Means is a prototype based, partitional clustering technique that attempts to find a user-specific number of clusters (K), which are represented by their centroids

Date No.

Basic K-Means Algorithm

1. Select K - points as initial centroids
2. **repeat**
3: Form K - clusters by assigning each point to its closest centroid
4: Recompute the centroid of each cluster
5: **until** Centroids do not change

* In the first step, shown in fig (a), points are assigned to the initial centroids, which are all in the larger group of points.
  Here, we use the mean as the centroid

* After points are assigned to the centroid, the centroid is then updated.

* In the second step, points are assigned to the updated centroids, & the centroids are updated again.

* In step 2, 3 and 4 which are shown in fig (b), (c) and (d) respectively, two of the centroids move to the small group of points.

MATRIKAS

Date No.

* when K-means algorithm terminates, in fig (d) (because no more changes occur), the centroids have identified the natural grouping of points

Assigning points to the closest centroid

* To assign a point to the closest centroid, we need a proximity measure that quantifies the notion of "closest" for the specific data under consideration

Centroids and objective functions

* The centroid can vary depending upon the proximity measure for the data and the goal of the clustering
* The goal of the clustering is typically expressed by an objective function that depends on the proximities of the points to one another or to the cluster centroids.

Table: K-Means: Common choices for proximity, centroids and objective functions

Proximity function	Centroid	objective function
Manhattan (L_1)	median	Minimize sum of the L_1 distance of an object to its cluster centroid
Squared Euclidean (L_2^2)	mean	Minimize sum of the squared L_2 distance of an object to its cluster centroid
Cosine	mean	Maximize sum of the cosine similarity of an object to its cluster centroid
Bregman divergence	mean	Minimize sum of the Bregman divergence of an object to its cluster centroid

Date No.

Table of notion

Symbol	Description
x	An object
C_i	The i^{th} cluster
c_i	The centroid of cluster C_i
c	The centroid of all points
m_i	The number of objects in the i^{th} cluster
m	The number of objects in the dataset
K	The number of clusters

Example 1: Data in Euclidean Space

→ Consider data whose proximity measure is Euclidean distance

→ For our objective function, we use the sum of the squared error (SSE), which is also known as scatter. In other words, we calculate the error of each data points (i.e., its Euclidean distance to the closest centroid) and then compute the total sum of squared errors.

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist(c_i, x)^2$$

→ It can be shown that the centroid that minimizes the SSE of the cluster is the mean.

→ The centroid (mean) of the i^{th} cluster is defined by

$$c_i = \frac{1}{m_i} \sum_{x \in C_i} x$$

Example 2: Document Data

→ Consider document data and the cosine proximity measure.

Date No.

→ Our objective here is to maximize the similarity of the documents in a cluster to the cluster centroid. this quantity is known as the cohesion of the cluster

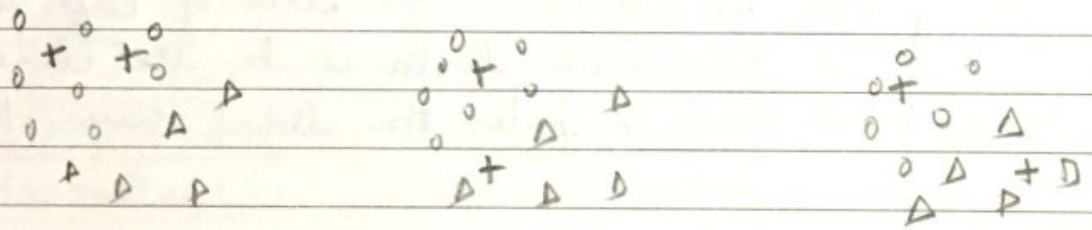
→ For this objective, it can be shown that the cluster centroid is the mean

$$\text{Total cohesion} = \sum_{i=1}^{K} \sum_{x \in C_i} \text{Cosine}(x, C_i)$$

Choosing Initial Centroids

* when random initialization of centroids is used, different runs of K-means typically produce different total SSEs

Eg:



(a) (b) (c)

* Choosing the proper initial centroids is the key step of the basic K-means procedure.
A common approach is to choose the initial centroid randomly, but the resulting clusters are often poor.
Therefore, other techniques are often employed for initialization

Date No.

1. One approach is to take a sample of points and cluster them using a hierarchical clustering technique. K clusters are extracted from the heirarchical clustering & The centroids of those clusters are used as the initial centroids.
This approach is practical only if
a) The sample is relatively small
b) K is relatively small compared to the sample size

2. One more approach,
→ Select the first point at random or take the centroid of all points
→ Then, for each successive initial centroid, select the point that is farthest from any of the initial centroids already selected.

Disadvantage
* This approach can select outliers, rather than points in dense regions
* Expensive to compute the farthest point from the current set of initial centroids

Agglomerative Hierarchical Clustering

* There are two basic approaches for generating a hierarchical clustering.

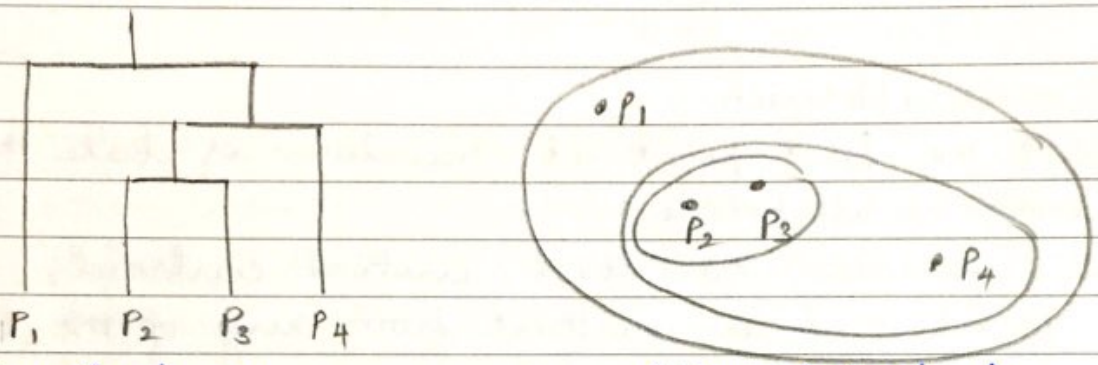
Agglomerative: Start with the points as individual clusters and at each step, merge the closest pair of clusters. This requires defining a notion of cluster proximity.

Divisive: Start with one, all-inclusive cluster and, at each step, split a cluster until only singleton cluster of individual points remain. In this case, we need to decide which cluster to split at each step and how to do the splitting.

Date No.

* A hierarchical clustering is often displayed graphically using a tree-like diagram called Dendrogram, which displays both the cluster-subcluster relationship and the order in which clusters were merged.

Eg:- Figure shows a heirarchical clustering of four points shown as a dendrogram and as nested clusters



(a) Dendrogram (b) Nested cluster

Basic Agglomerative Hierarchical clustering Algorithm

Algorithm: Basic agglomaratie hierarchical clustering
1: Compute the proximity matrix if necessary
2: **repeat**
3: Merge the closest two clusters
4: Update the proximity matrix to reflect the proximity between the new cluster and the original clusters
5: **until** only one cluster remains

Defining proximity between clusters

* The key operation of above algorithm is the computation of the proximity between two clusters.
The definition of cluster proximity differentiates the various agglomerative hierarchical techniques.

* cluster proximity is typically defined with a particular type of cluster in mind

Date No.

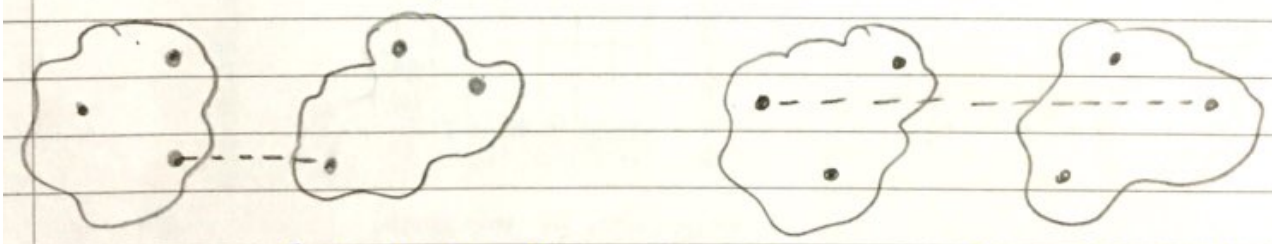
Eg:- Many agglomerative hierarchical clustering techniques such as MIN, MAX and group average, come from a graph based view of clusters.

* MIN defines cluster proximity as the proximity between the closest two points that are in different clusters (using graph terms, the shortest edge between two nodes in different subsets of nodes.)

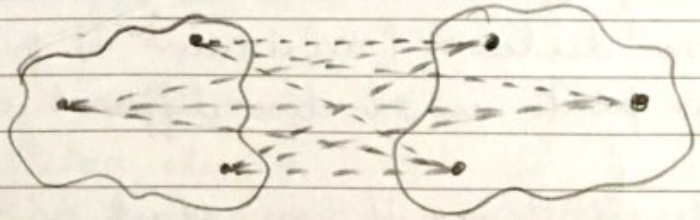
* MAX takes the proximity between the farthest two points in different clusters to be the cluster proximity (using graph terms, the longest edge between two nodes in different subsets of nodes)

* Group average defines cluster proximity to be the average pair wise proximities (average length of edges) of all pairs of points from different clusters.

NOTE:- MIN and MAX techniques have alternative names as single link & complete link respectively



(a) MIN (single link) (b) MAX (complete link)



(c) Group average

fig:- Graph-based definitions of cluster proximity

MATRIKAS

* An alternative technique, Ward's method measures the proximity between two clusters in terms of the increase in the SSE that results from merging of the two clusters.

Specific Techniques

To illustrate the behavior of various hierarchical clustering algorithms, we shall use sample data that consists of 6 2-D points, which are shown below.

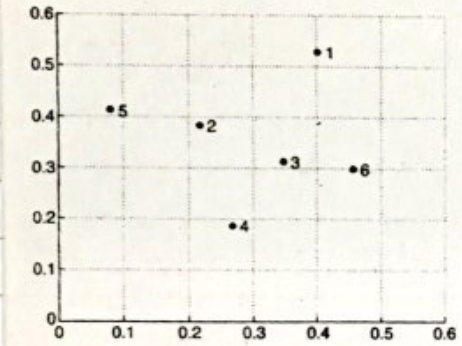


Figure 8.15. Set of 6 two-dimensional points.

Point	x Coordinate	y Coordinate
p1	0.40	0.53
p2	0.22	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

Table 8.3. xy coordinates of 6 points.

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Table 8.4. Euclidean distance matrix for 6 points.

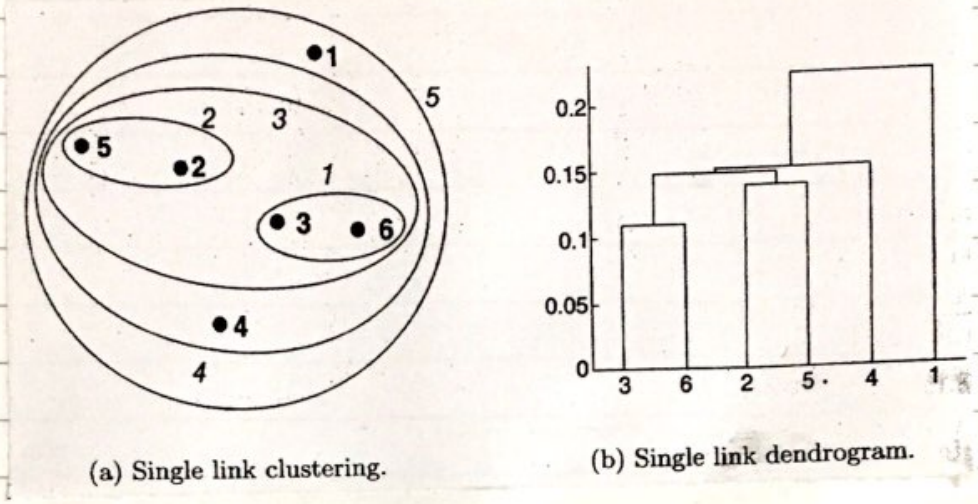
Single Link or MIN

* Here, the proximity of two clusters is defined as the minimum of the distance (maximum of the similarity) between any two points in the two different clusters

* Using graph terminology, if you start with all points as singleton clusters & add links between points one at a time, shortest links first, then these links combine the points into clusters.

Date No.

Eg:- The following figure shows result of applying single link technique to our example data set of 6 points



(a) Single link clustering.

(b) Single link dendrogram.

* Distance between points 3 & 6 is 0.11 (from previous table)
* Distance between clusters {3, 6} & {2,5} is given by

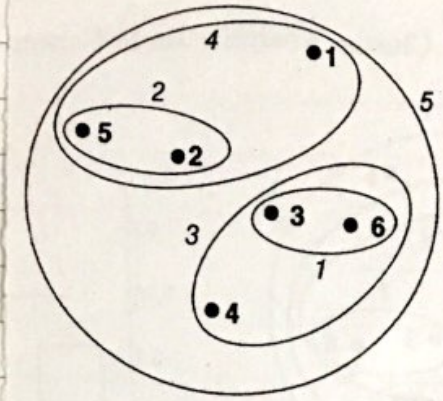
$$dist(\{3,6\},\{2,5\}) = \min\{dist(3,2), dist(6,2), dist(3,5), dist(6,5)\}$$
$$= \min\{0.15, 0.25, 0.28, 0.39\}$$
$$= 0.15$$

Complete link or MAX or CLIQUE

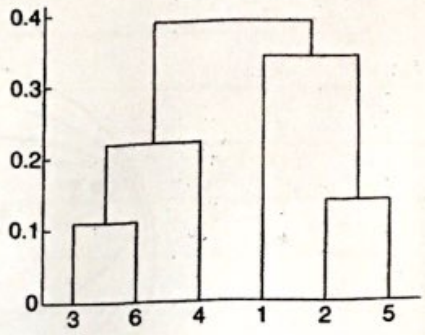
* For MAX version of hierarchical clustering, the proximity of two clusters is defined as the maximum of the distance (minimum of the similarity) between any two points in the two different clusters

* Using graph terminology, if you start with all points as singleton clusters and add links between points one at a time, shortest links first, then a group of points is not a cluster until all the points in it are completely linked i.e. form a clique

Eg:- The following figure shows result of applying MAX to the sample data of six points.



(a) Complete link clustering.



(b) Complete link dendrogram.

* As with MIN, point 3 & 6 are merged first
* However, {3,6} is merged with {4}, instead of {2,5} or {1} because

$$dist(\{3,6\},\{4\}) = \max(dist(3,4), dist(6,4))$$
$$= \max(0.15, 0.22)$$
$$= 0.22$$

$$dist(\{3,6\},\{2,5\}) = \max(dist(3,2), dist(3,5), dist(6,2), dist(6,5))$$
$$= 0.39$$

$$dist(\{3,6\},\{1\}) = \max(dist(3,1), dist(6,1))$$
$$= 0.23$$

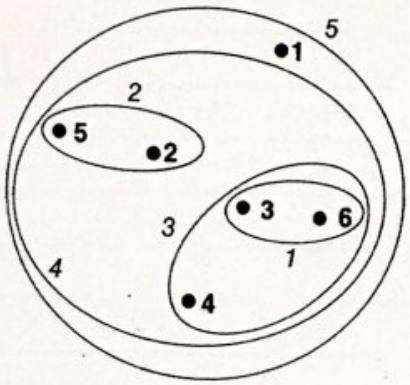
Group Average

* Here, the proximity of two clusters is defined as the average pairwise proximity among all pairs of points in the different clusters.

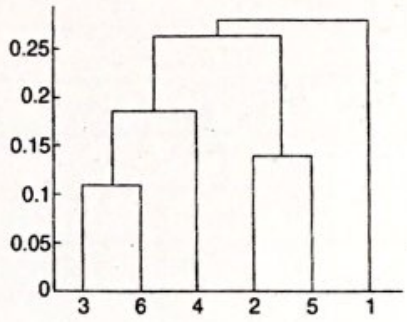
i.e., proximity (C_i, C_j) of clusters C_i & C_j which are of size m_i and m_j is

$$\text{proximity}(C_i, C_j) = \frac{\sum_{x \in C_i} \sum_{x \in C_j} \text{proximity}(x, y)}{m_i * m_j}$$

Eg:- The following figure shows result of applying group average to our sample dataset of 6 points



(a) Group average clustering.



(b) Group average dendrogram.

Here,

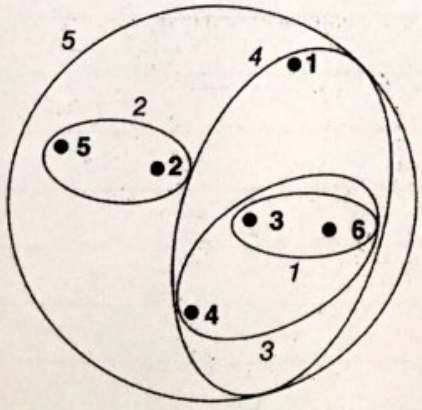
$$\text{dist}(\{3,6,4\}, \{1\}) = \frac{0.22 + 0.37 + 0.23}{3*1} = 0.28$$

$$\text{dist}(\{2,5\}, \{1\}) = \frac{0.2357 + 0.3421}{2*1} = 0.2889$$

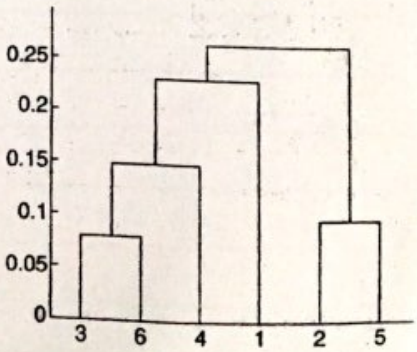
$$\text{dist}(\{3,6,4\}, \{2,5\}) = \frac{0.15 + 0.28 + 0.25 + 0.39 + 0.20 + 0.29}{6*2}$$

$$= 0.26$$

Ward's method & Centroid Method



(a) Ward's clustering.



(b) Ward's dendrogram.

MATRIKAS

Date No.

* Ward's method is very similar to the group average method when the proximity between two points is taken to be the square of distance between them.

* Centroid method calculate the proximity between two clusters by calculating the distance between the centroids of clusters.

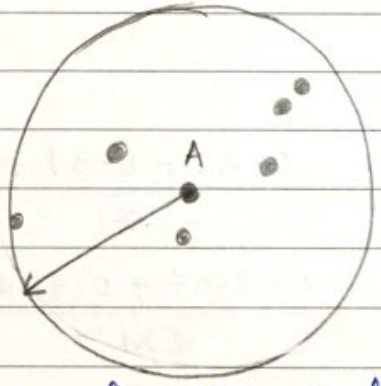
DBSCAN

It is a density-based algorithm

Density: no. of points within a specified radius

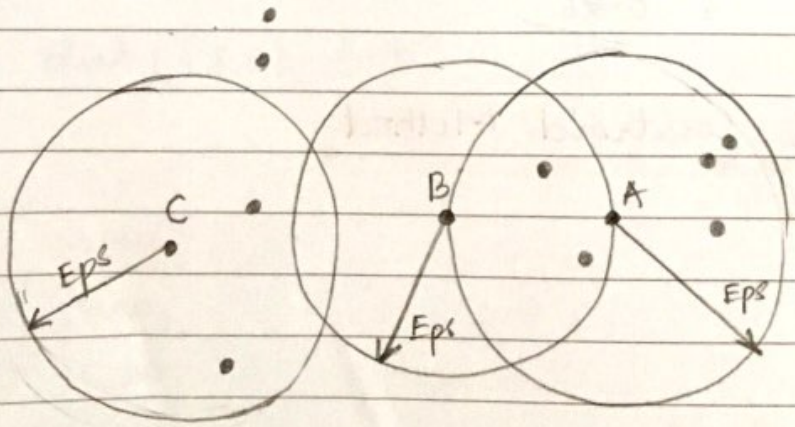
Density-Based clustering

locates regions of high density that are separated from one another by regions of low density.



density = 7 (including A)

center-based density



Point A → Core point
Point B → Border point
Point C → Noise point

Core, border and noise points

In center-based density, the points (number of points) within a radius of Eps of point A is ≠ (including A itself). This approach is called center-based approach on which DBSCAN is based upon.

Classification of points according to Center-Based Density

1. **Core points:** (In the interior of a dense region)
   A point is a core point if, it has more than a specified number of points (i.e; certain threshold, MinPts) within Eps.

2. **Border Point:** (on the edge of a dense region)
   It has fewer than MinPts with Eps, but is in the neighbourhood of a corepoint.

3. **Noise points** (In a sparsely occupied region)
   It is neither a core point nor a border point.

The DBSCAN Algorithm

Algorithm: DBSCAN Algorithm
1: Label all points as core, border or noise points
2: Eliminiate noise points
3: Put an edge between all core points that are within Eps of each other
4: Make each group of connected core points into a separate cluster.
5: Assign each border point to one of the clusters of its associated core points.

Selection of DBSCAN parameters

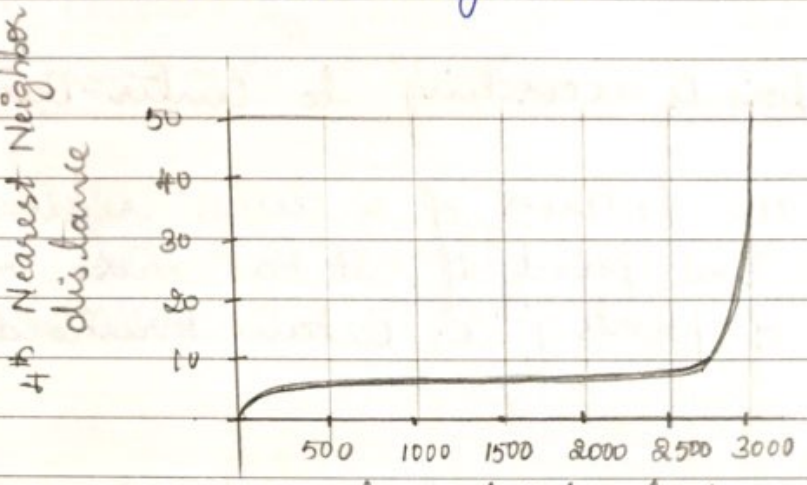
How to determine Eps and MinPts?
→ The idea is that for points in a cluster, their K^{th} nearest neighbors are at roughly the same distance.

MATRIKAS

we call it as K-dist

→ Noise points have the K^{th} - nearest neighbor at farther distance

→ So plot sorted distance of every point to its K^{th} nearest neighbor



Here, K=4

fig:- K-dist plot for sample data

We see a sharp change at the value of K-dist that corresponds to a suitable value of Eps.

We select this distance as Eps and we take the value of K as MinPts

Strength and weakness

Strength

* resistance to noise
* can handle clusters of different shapes & sizes.

weakness

* Doesnot work well for varying densities, high dimensional data

Cluster Evaluation

Several Issues for cluster validation / Several aspects of cluster validation / Cluster evaluation

1. Determining the clustering tendency of a set of data.

i.e; distinguishing whether (set of data) non-random structure actually exists in data.

2. Determining the correct number of clusters.

3. Evaluating how well the results of a cluster analysis fit the data without reference to external information.

4. Comparing the results of a cluster analysis to externally known results, such as externally provided class labels

5. Comparing two sets of clusters to determine which is better.

The evaluation measures (or indices) that are applied to judge various aspects of cluster validity are traditionally classified into following three types.

1. Unsupervised (or internal indices)
   Used to measure the goodness of a clustering structure without respect to external information.
   Eg:- SSE

2. Supervised (External indices)
   Measures the extent to which the clustering structure matches some external structure.
   Eg:- Entropy - measures how well cluster labels match externally supplied class labels

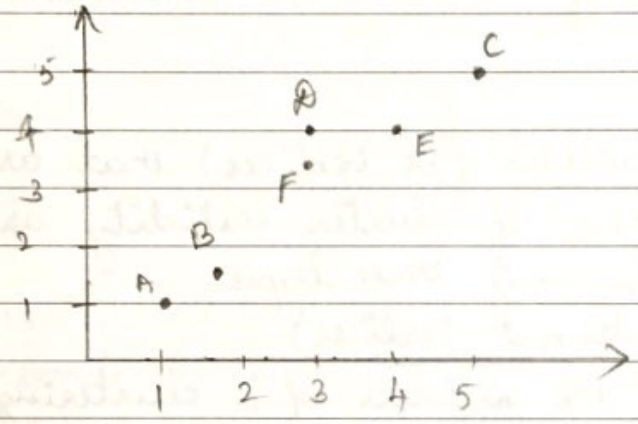
3. Relative Index
   Compares different clustering or clusters. A relative cluster evaluation measure is a supervised or unsupervised evaluation measure that is used for the purpose of comparision
   Eg:- Two K-means clusterings can be compared using either SSE or Entropy.

NOTE:

Unsupervised measures of cluster validity are often further divided into two classes

* Measures of cluster cohesion (Compactness, tightness) which determine how closely related the objects are in a cluster
* Measures of cluster separation (isolation) which determine how distinct or well-separated a cluster is from other clusters.

Solve using DBSCAN



A (1,1)
B (1.5, 1.5)
c (5, 5)
D (4, 4)
E (4, 4)
F (3, 3.5)

Cluster these points using DBSCAN

Find distance between these points using Euclidean distance

	A	B	c	D	E	F
A	0	0.71	5.66	3.61	4.24	3.20
B	0.71	0	4.95	4.95	3.54	2.50
C	5.66	4.95	0	2.24	1.41	2.50
D	3.61	2.92	2.24	0	1.00	0.50
E	4.24	3.54	1.41	1.00	0	1.2
F	3.20	2.50	2.50	0.50	1.2	0

Assume $\epsilon = 1.5$ & minpts = 3.

Lets randomly select F, with ϵ radius of F The neighbours are F, D & E, hence F is a core point

Date No.

Similarly, E & D are core points.
Look at C (C is in the neighborhood of E, but doesnot satisfy minpts condition) hence it is a border point
A & B are noise points
Hence, one cluster (EFDC) is formed