UNIT-(V) :- (1) clustering

Clustering :- A Collection of data objects
- → similiar [one another within same group] [related]
- → dissimilar [to objects in other groups] [unrelated]

Cluster Analysis : [clustering, data segmentation]
- → Finding similarities between data according to the characteristics found in the data grouping similar data objects into clusters.

unsupervised learning :-
- → no predefined classes (ie, learning by observations vs learning by examples : supervised)
- → Applications :-
  - → A stand-alone-tool to get Insight into data distribution
  - → As a preprocessing step for Other Algorithms
  - → Biology :- Animal kingdom
  - → Information retrieval : document clustering
  - → land use : similar lands
  - → Marketing :- Targeted marketing programs by discovering distinct groups
  - → city planning & earth quake studies :
  - → Economic science :

Clustering as a preprocessing tool (utility)
- Summarization → Preprocessing for regression
- Compression → Image processing
- finding k-nearest Neighbors
- Outlier detection

Quality (what is good clustering)

- A good clustering method will produce high quality cluster
  - → high Intra-class similarity : cohesive within clusters
  - → lower Inter-class similarity: distinctive between clusters.
- The quality of a clustering method depends on
  - → The similarity measure used by the method
  - → its implementation
  - → The ability to discover some or all hidden patterns

Measuring the quality of the cluster

→ dissimilarity / similarity metric

- Similarity expressed in terms of a distance function.
- The definitions of distance functions are usually rather different for interval scaled, boolean, categorical, ordinal ratio and vector variables.
- weights should be associated with different variables.

→ quality of clustering:

- This is usually seperate "quality" function that measures the "goodness" of a cluster.
- It is hard to define "similar enough" or "good enough"
  - because they are subjective

Considerations of cluster Analysis

- partitioning criteria
  - Single level vs hierarchial partitioning
- Seperation of clusters
  - exclusive (one customer belongs to only one region) vs non-exclusive (eg:- one document belong to more than one class)
- Similarity ~~&~~ measure.
  - distance based (euclidean, vector) vs Connectivity based (density, contiguity)

- Clustering space
  - full space (often when low dimensional) VS subspaces (often in high dimensional clustering)

Requirements & challenges

- Scalability :-
  - Clustering all the data Instead of samples
- Ability to deal with different kind of attributes :-
  - Numerical, binary, categorical, ordinal etc
- Constraint based clustering :-
  - → user may give inputs on constraints
  - → use domain knowledge to determine input parameters
- Interpretability and usability :
- Others :
  - discovery of cluster with arbitrary shape
  - ability to deal with noisy data
  - Incremental clustering and insensitivity to input order
  - High dimensionality

Major clustering approaches

- Partitioning approach :
  - Construct various partitions and then evaluate them by some criterion
    - eg: minimizing the sum of square errors
  - Typical methods : ~~K-means~~ K-means, K-medoids, CLARANS
- Hierarchial approach :
  - creates a hierarchial decomposition of set of data by some criteria.
  - → Typical methods : diana, Agnes, BIRCH, Chameleon

- **Density based approach :**

  - Based on connectivity & density function
  - Typical methods : DBSACN, OPTICS, DenClue

- **Grid based approach :**

  - Based on multiple-level granularity structure
  - Typical methods : STING, WaveCluster, CLIQUE

- **Model Based :-**

  - A model is hypothesized for each of the clusters and tries to find the best fit of model to each other
  - Typical method :- EM, SOM, COBWEB

- **Frequent pattern Based :-**
  - Based on analysis of Frequent patterns.
  - Typical methods : P-cluster

- **User Guided | Constraint Based**
  - Clustering by considering user-specified on 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

$\boxed{\text{Partition Based Methods}}$

- **Partition method** :- Partition a database D of n objects into a set of k clusters, such that sum of squared distances is minimized. (where ci is the centroid or medoid of cluster)

$$E = \sum_{i=1}^{k} \sum_{p \in c_i} (p - c_i)^2$$

- Given k, find a partition of k clusters that optimizes the chosen partition criteria

→ global optimal :- exhaustively enumerate all partitions

→ Heuristic partitions : Each cluster is represented by the centre of cluster or one of objects (k-means, k-medoids)

→ k-means : each cluster is represented by center of cluster

→ K-Medoids or PAM (Partition around medoids) each cluster is represented by one of objects in the cluster.

k-means clustering method

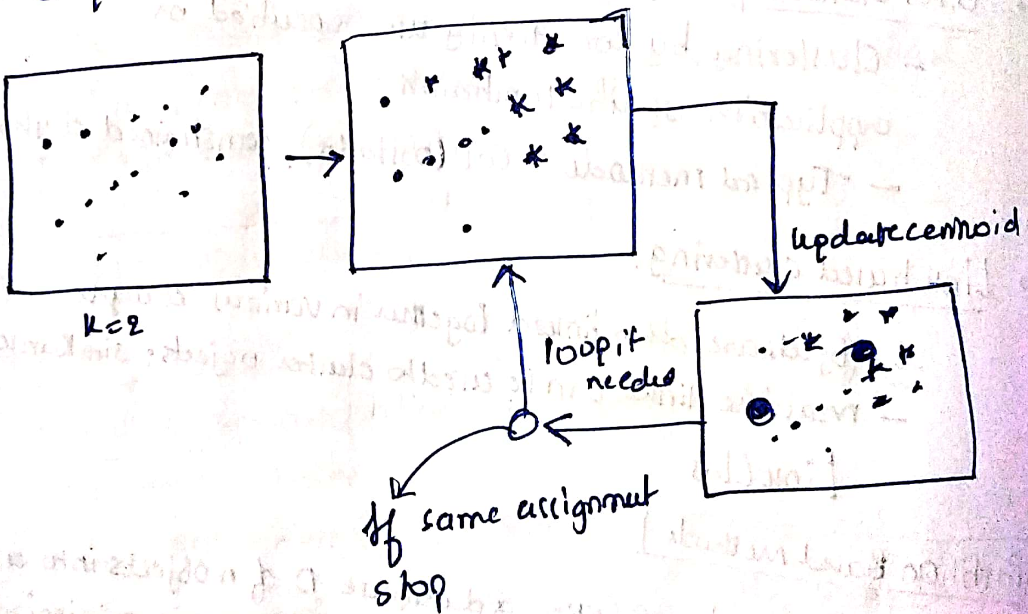
: 4 steps

→ Partition objects into k-nonempty subsets

→ Compute seedpoints as the centroids of clusters of the current partitioning (centroid is the centre, mean of cluster)

→ Assign each object to cluster with nearest seedpoint

→ goback to step 2, when the assignment doesnot change stop.



Advantages:-

- efficient : O(tkn), # of objects → n, k → # of clusters, t → # of iterations.

  Mostly k, t << n

  Pam (O(k(n-k)^2))

  CLARA : O(ks^2+k(n-k))

disadvantage :

→ terminates often at local optimal
→ sensitive to noisy data & outliers
→ NOT suitable to discover clusters with non-convex shapes

K-means Problems ?

Variations of K-means

- most of the variants of K-means differ in
  - selection of the initial k means
  - dissimilarity calculations
  - strategies to calculate cluster means
- Handling categorical data : K-modes
  - replacing means with modes
  - A mixture of categorical & numerical data : K-prototype method

why K-medoids ?

- K-means algorithm is sensitive to outliers
- But instead of taking the mean, the medoids can be used (which are centrally located object in a cluster)

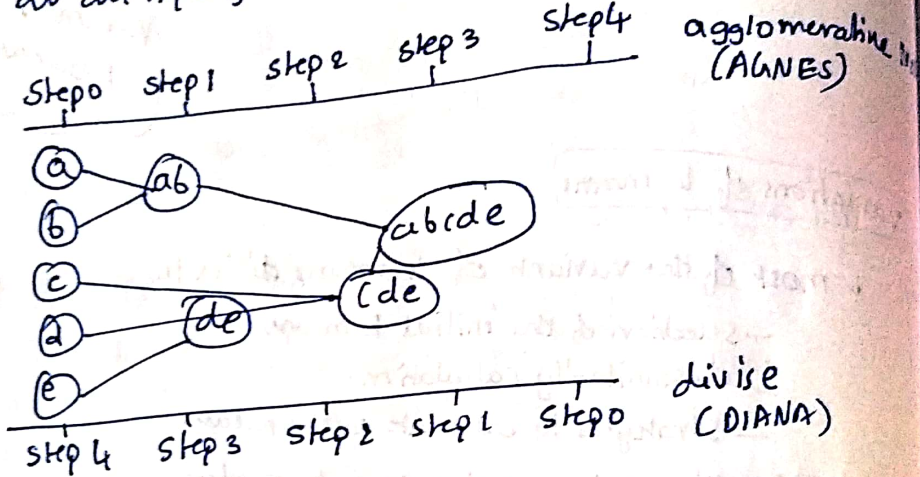
PAM : A typical K-mediods problem

- Start from an initial set of medoids & iteratively replace one of medoids by one of non-mediods iff it improves the total distance of the resulting clustering
- PAM works effectively for small data sets, but does not scale well for large data sets (due to computational complexity)

→ Improvements on PAM
- → CLARA : PAM on samples
- → CLARANS : Randomized re-sampling

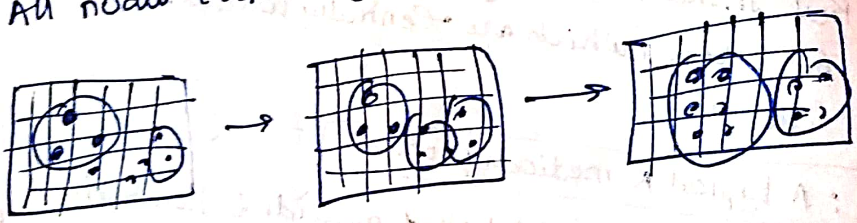
Hierarchial methods

- use distance matrix as a clustering criteria.
- This method does not require number of clusters K as an input, but needs an terminating condition



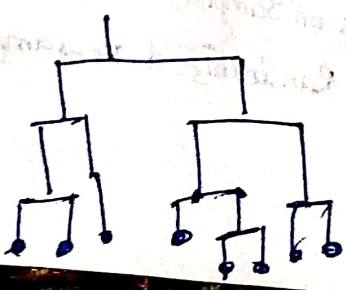
Agnes (Agglomerative nesting)

- Implemented in statistical packages Splus
- use single link method & dissimilarity matrix
- merges nodes with least dissimilarity.
- goes on a non-descending fashion
- All nodes eventually belong to same cluster



dendogram: How clusters are merged

- Decompose data objects into several levels of nested partitioning (tree of clusters), called a dendogram
- A clustering of data objects is obtained by cutting the dendogram at the desired level, then each connected component forms a cluster



DIANA (divisive Analysis)

- Inverse order of Agnes
- eventually each node forms a cluster of its own.

Distance between clusters

- Sing link: smallest distance between an element in one cluster and another element in another cluster.

  $dist(K_i, K_j) = \min(t_{ip}, t_{iv})$

- Complete link: longest distance between an element in one cluster & an other element in another cluster.

  $dist(K_i, K_j) = \max(t_{ip}, t_{iv})$

- Average: Average distance between an element in one cluster and element in another cluster $dist(K_i, K_j) = avg(t_{ip}, t_{iv})$
- Centroid: distance between two centroids in two clusters
- medoid: distance between two medoids in two clusters.

— centroid: $C_m = \dfrac{\sum_{i=1}^{N}(t_{ip})}{N}$

— Radius: Square root of average distance from any point in cluster to its centroid.

$$R_m = \sqrt{\frac{\sum_{i=1}^{N}(t_{ip} - C_m)^2}{N}}$$

— Diameter: Square root of average mean squared distance between all pairs of points in cluster.

$$D_m = \sqrt{\frac{\sum_{i=1}^{N}\sum_{i=1}^{N}(t_{ip} - t_{iv})^2}{N(N-1)}}$$

Extensions to Hierarchical clustering

- major weakness in agglomerative clustering: -
  - Can never undo what's done previously
  - Do not scale well

- Integration of hierarchical & distance based clustering :

① Birch(1996) :-

(Balanced iterative reducing & clustering using hierarchies)

- Incrementally construct a CF (clustering feature) tree, a hierarchical datastructure for multiphase clustering

Phase 1 :- Scan DB to build an initial in memory CF tree (A multi-level compression of data that tries to preserve the inherent clustering structure of the data)

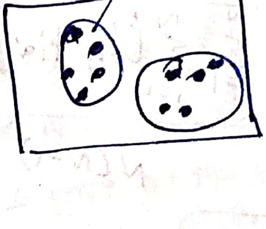
Phase 2 : use an arbitrary clustering algorithm to cluster the leaf nodes of C-F Tree

- scales linearly
- handles only numeric data
- clustering feature vector in Birch

(CF) = (N, LS, SS)

- N → Number of data points
- LS → Linear sum of N points $\sum_{i=1}^{N} X_i$
- SS → square sum of N points $\sum_{i=1}^{N} X_i^2$

$\boxed{CF = (5, (16,30), (54,190))}$

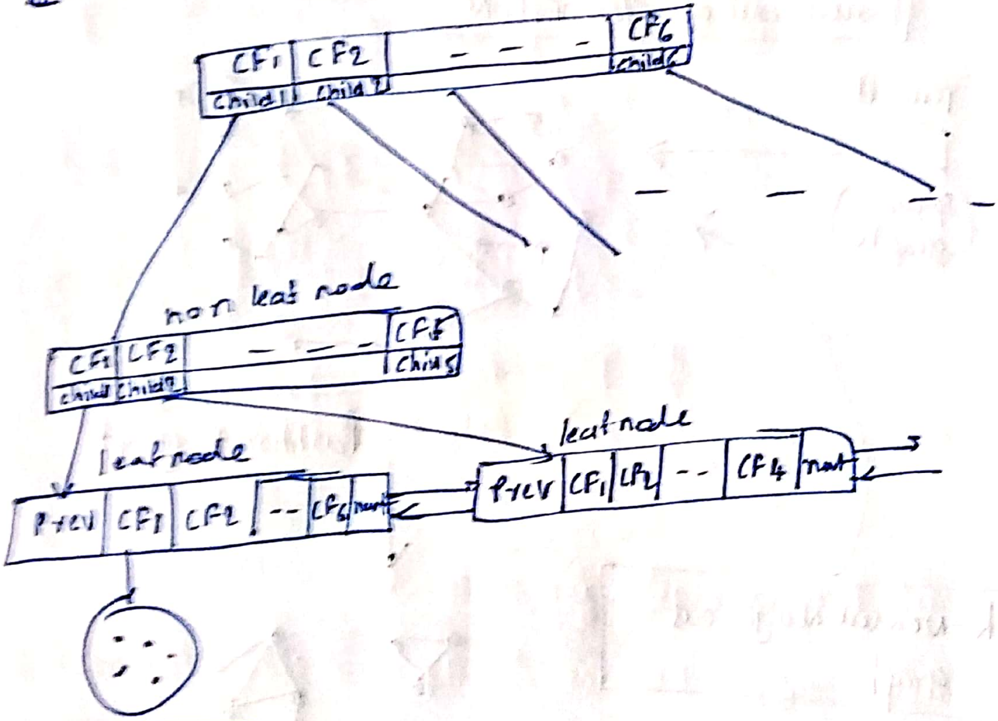


(3, 4)
(2, 6)
(4, 5)
(4, 7)
(3, 8)

- A CF Tree is a height balanced tree that stores the clustering features for a hierarchial clustering.
  - A non-leaf node in a tree has children
  - The nonleaf node stores the sums of CFs of their children.

- A CF tree has two parameters
  - Branching factor: Max # of children
  - Threshold: Max diameter.

B = 7
L = 6



$\boxed{\text{Birch Algorithm}}$:

- cluster diameter:

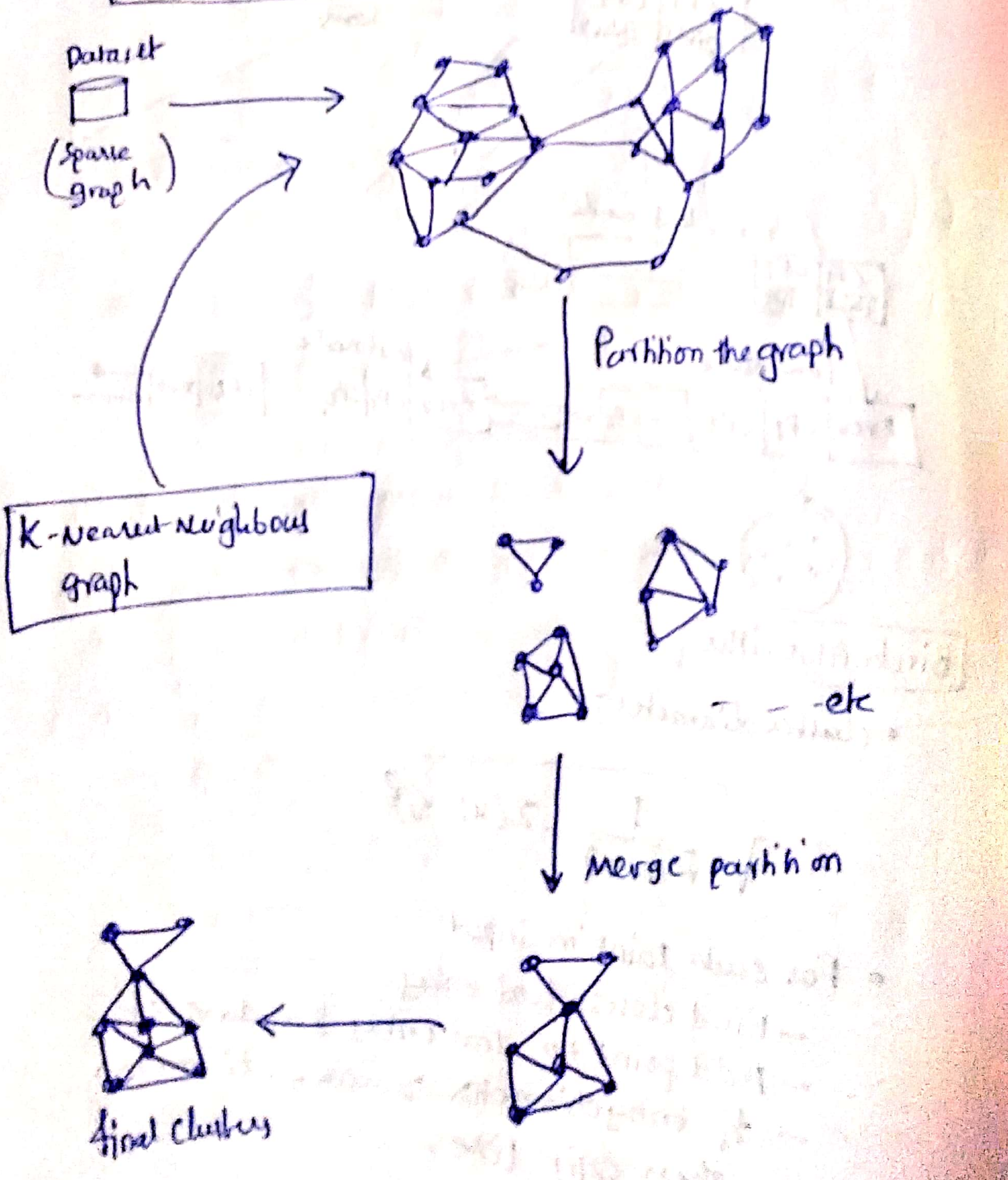
$$\sqrt{\frac{1}{n(n-1)}\Sigma(x_i - x_j)^2}$$

- For each point in input
  - Find closest leaf entry
  - Add point to leaf entry & update CF
  - If entry diameter > max_diameter then split leaf.
- Algorithm: O(n)
- Sensitive to insertion order of data points

CHAMELEON (hierarchial clustering using dynamic modelling)

- measures the similarity based on a dynamic model
  - Two clusters are merged only if
    If (Interconnectivity & Proximity between two clusters) are high;
- graph-Based two phase Algorithm:
  - use a graph-partitioning algorithm:- cluster objects into a large number of relatively small sub-clusters
  - use an agglomerative hierarchial clustering algorithm: find the genuine clusters by repeatedly combining these sub-clusters.

Framework of CHAMELEON



Probabilistic Hierarchial clustering

- Algorithmic hierarchial clustering
  - Non trivial to choose a good distance measure
  - hard to handle missing values
  - Optimization goal not clear : heuristic, local search
- Probabilistic hierarchial clustering :
  - use probabilistic models to measure distance between clusters
  - Generative model : set of data objects to be clustered as a sample of underlying data generation mechanism to be analyzed.
  - easy to understand, same efficiency as Algorithemic hierarchial clustering, but can handle partly observed data

eg: Gaussian distribution or Bernoulli distribution

Generative model :-

- let 1-D points $x = \{x_1, \ldots x_n\}$ and assume are generated by gaussian distribution

$$N(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)}$$

- Probability that point $x_i \in X$ is generated by model

$$P(x_i / \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x_i-\mu)^2}{2\sigma^2}}$$

- The maximimum likelihood (Task of generative model learning)

$$N(\mu_0, \sigma_0^2) = \arg \max \{ L(N(\mu, \sigma^2)) : X \}$$

A probabilistic Hierarchical Clustering Algorithm

- $Q\{c_1 \dots c_m\} = \prod_{i=1}^{m} P(c_i)$

  Maximum likelihood

- distance between clusters c_1 and c_2

$$dist(c_1, c_2) = -\log\left(\frac{P(c_1 \cup c_2)}{P(c_1)\,P(c_2)}\right)$$

- Algorithm :- Progressively merge points & clusters

  Input: $D = \{o_1 \dots o_n\}$

  output: A hierarchy of clusters

  Method:

  create a cluster for each object $c_i = \{o_i\}$ $1 \le i \le n$.

  for $i = 1$ to n

  {

  find pair of clusters c_i & c_j such that

$$c_i, c_j = \arg\max_{i \ne j}\left\{\log\left(\frac{P(c_1 \cup c_2)}{P(c_1)\,P(c_2)}\right)\right\}$$

  If $\log\left(\frac{P(c_1 \cup c_2)}{P(c_1)\,P(c_2)}\right) > 0$, merge c_i & c_j

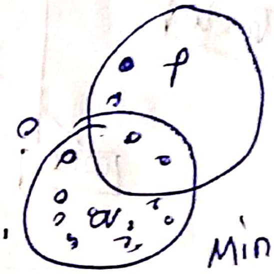
  }

Density Based clustering methods

- clustering based on the density (local cluster criterion) such as density connected points
- Major features
  - one scan
  - handle noise
  - discover clusters of arbitary shape
  - requires density parameters as terminating condition

- Basic concepts
  - **Eps : (Parameter)** :- Maximum radius of the neighbourhood
  - **MinPts (Parameter)** : minimum number of points in an Eps neighbourhood of that point

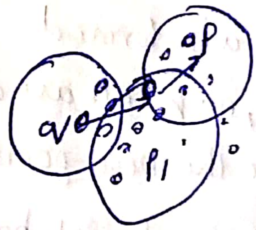
- $N_{Eps}(P) : \{ q$ belongs $D \mid dist(P, q) \leq Eps \}$
- Direct density reachable : A point p is directly density reachable from a point q wrt Eps, minpts if
  - p belongs to $N_{Eps}(q)$
  - core point condition
    $|N_{Eps}(q)| \geq Minpts$



Min pts = 5
Eps = 1 cm

Density - Reachable & density connected

- Density reachable :-



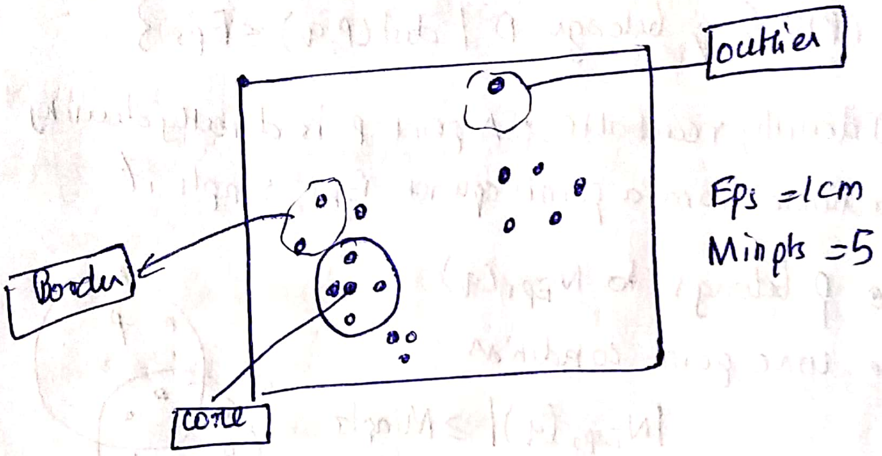
A point p is density reachable from a point q wrt Eps, minpts if there is a chain of points $p_1, p_2 \ldots p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density reachable from p_i

- Density-Connected

A point p is density connected to a point q wrt Eps, minpts if there is a point o such that both p & q are density Reachable from o wrt Eps & minpts

DBSCAN (density-Based spatial clustering of Applications with Noise)

- Relies on a density based notion of cluster. A cluster is defined as maximal set of density connected points
- Discovers clusters of arbitary shape in spatial database with Noise



DBscan : The Algorithm

- Arbitary select a point p
- Retreive all points density-reachable from pwrt Eps & minEpts
- If p is a core point cluster's formed
- If p is a border point, no points are density reachable from p & DBSCAN visit the next point of the database
- Continue the process until all of points have been Processed

OPTICS: A cluster-ordering method

(ordering points to identify clustering structure)

- good for both automatic & interactive cluster Analysis, including finding Intrinsic clustering structure
- Can be represented graphically or using visualization Techniques

- optics → added as an extension to DBSCAN

- Index based :-

  K = Number of dimensions

  N = 20

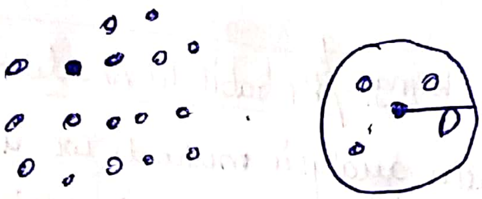
  P = 75%

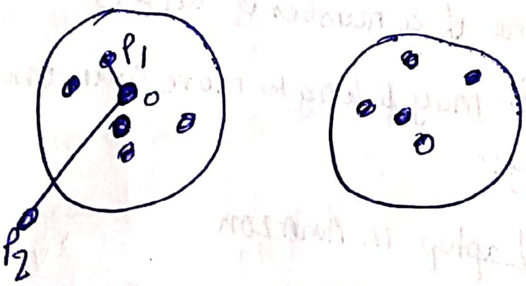
  M = N (1-P) = 5   Complexity O(N log N)

- Core distance :-
  min eps st point is core

- Reachability distance :-

  Max (Core-distance (o), d(o,p))





$\boxed{\text{DENCLUE : using statistical density functions}}$

- $f_{Gaussian}(x\,y) = e^{-\frac{d(x,y)^2}{2\sigma^2}}$ Influence of Y on X

$$f^{D}_{Gaussian}(x) = \sum_{i=1}^{N} e^{-\frac{d(x_i, x)^2}{2\sigma^2}}$$ total influence on x

$$\nabla f^{D}_{Gaussian}(x, x_i) = \sum_{i=1}^{N} (x_i - x) \cdot e^{-\frac{d(x, x_i)^2}{2\sigma^2}}$$

gradient of x in direction of x_i

Major features of DENCLUE

- Solid Mathematical foundation
- good for datasets with large amounts of noise
- Needs larger parameters
- Faster than DBSCAN

Grid-Based Clustering method

- using multi-resolution grid datastructure
- Several Interesting methods
  - STING (Statistical Information Grid approach)
  - wavecluster (A multi resolution clustering approach)
  - CLIQUE (Both grid-based & subspace clustering)

Model Based clustering :: **Probabilistic model-Based clustering**

- In all the cluster analysis methods we use
  - data object for each to be assigned to only one of a number of clusters
- Sometimes it may belong to more than one cluster

eg:-

Laptop in AMAZON

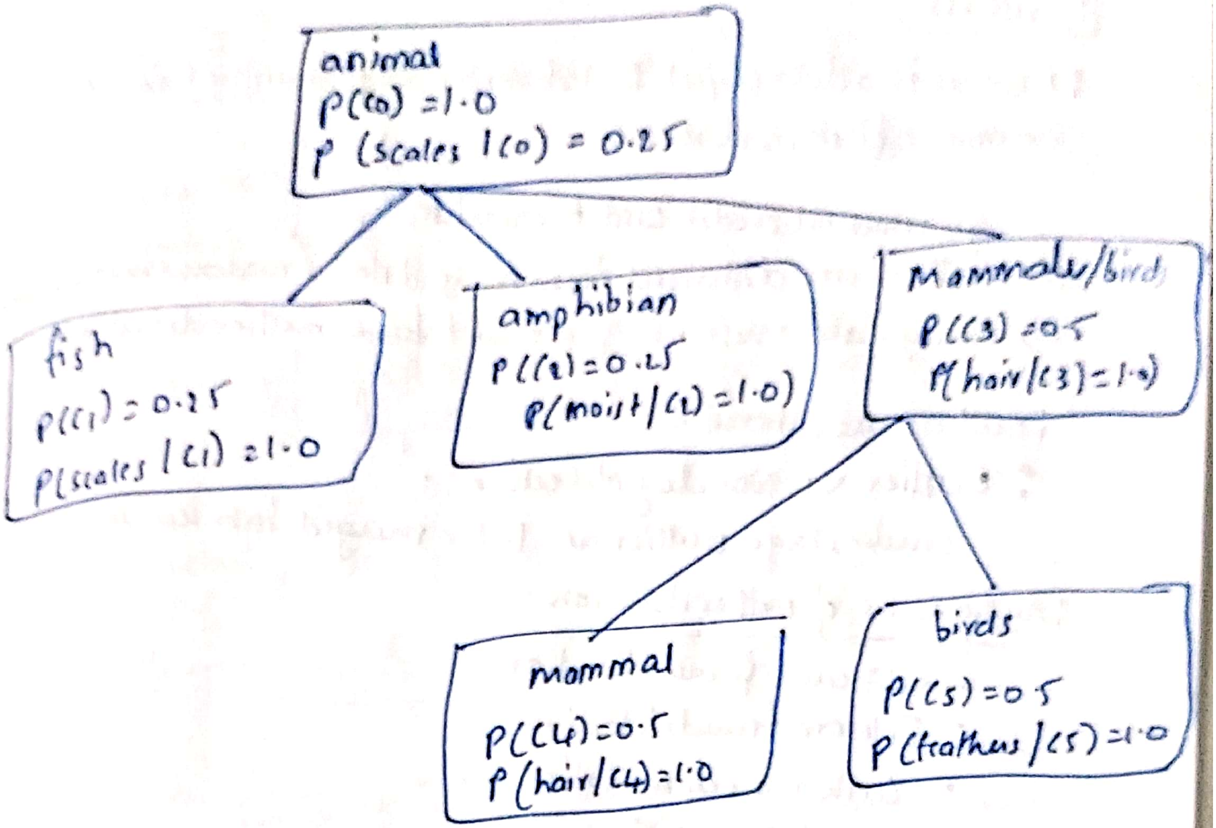
→ Electronics, PC, Gaming

Model Based clustering Attempts to optimize the fit between the data & Some Mathematical mode by using Statistical & AI approach

— Conceptual clustering
(finds characteristic description for each concept (class))

— COBWEB (Fishing 7)
- (Incremental conceptual learning)
- creates hierarchical clustering in form of classification tree
- each node refers to a concept &

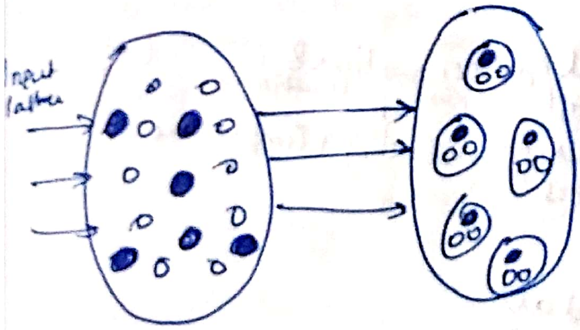
Contains probabilistic description for that object.



- limitations of COBWEB
  - not suitable for clustering large database
  - assumption that attributes are not co-related may not be true for all

— Class IT
  (an extension of COBWEB) but as failure as COBWEB

— Competitive learning



*hidden layers may also be present

— Self-organizing feature maps

- clustering is performed by objects competing for a class
- winner & its neighbours learn by adjusting their weights
- Similar to processing in human brain

Outliers

- A outlier is a data object that deviates significantly from the normal object in dataset

eg: unusual credit card transaction

→ Outliers are different from noisy data (random error, variance)

→ noisy data must be removed before outlier detection

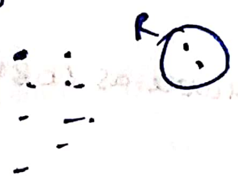
- outliers are interesting
- outlier vs Novelty detection:-
  early stage outlier but later merged into the model

Applications of outlier detection:-

- Credit card fraud detection
- Telecom fraud detection
- Customer segmentation
- Medical analysis

Types of outliers

global outlier



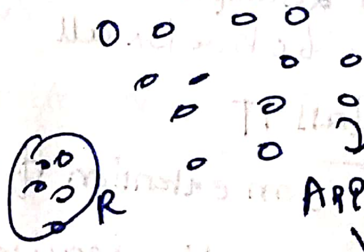
Contextual outlier (conditional) (outlier)

If data objects are divided into two groups then

- contextual attributes (defines context) → time & location
- behaviour attributes (characteristic) → Temperature

can be viewed as generalisation of local outliers, whose density significantly

collective outliers



Applications → Intrusion detection (unexpected prediction detection)

Challenges of outlier detection

- modeling normal objects & outliers properly
- Application - specific outlier detection
- Handling noise in outlier detection
- understandability

Outlier detection Methods

Based on user labelled examples of outliers

Supervised methods

- modelling outlier detection as a classification problem
- Methods for learning a classifier for outlier detection effectively

-challenges :-

- * Imbalanced classes

  Sol: make artificial outliers

- * catch as many outliers as possible

Unsupervised Methods

- Already assuming normal objects are clustered into groups each having distinct features
- An outlier is expected to be far away from the clusters

-challenges:

- * In virus detection it may have high false positive rate
- * very costly

Semi supervised Methods

If labelled normal objects are available

- Then train model using unlabelled data objects by using labelled examples
- Those not fitting lines are outliers

If labelled outlier objects are available

- Then take help of unsupervised methods

Based on assumptions between normal & Outlier data

Statistical methods (or) Model Based methods

mean, variance

Grubbs Test

- find gaussian distribution to model the normal data

-alternative

* Parametric vs non-parametric

$f(x,\theta)$

Proximity Based methods

- An object is outlier if the nearest neighbours of object are far away

distance based

- challenges:

  - hard to find outliers close to each other

- Nested loop algorithm
- grid cell algorithm
- Index based algorithm

density based

- local outlier factor (LOF)

Clustering Based Methods

- normal data belong to large & dense clusters
- outliers belong to small or sparse clusters or donot belong to any clusters