⇒ Cluster Analysis

→ partitions large dataset into groups acc. to similarity.

→ used for outlier detection.

→ main focus → distance based cluster-analysis

→ unsupervised learning, learning by observation,

→ Requirements of clustering:

- Scalibility (should work on a large scale database)
- Ability to deal diff. data types: numeric, binary, nominal, categorical, ordinal or mixture of them.
- discovery of arbitarily shaped clusters; should be able to detect arbitarily shaped clusters & not only spherical ones.
- Domain knowledge & input params
- Dealing with noisy data.
- Incremental clustering & independence from input order
- clustering high-dimensional data
- constraint-based clustering.
- Interpretability & usability

→ Clustering approach

- Partitioning criteria: with or without hierarchy
- separ^n of clusters: clusters may or may not be mutually exclusive
- similarity measures: distance function, or connectivity/density/contiguity based.

dist. based ⇒ adv. of optimization

connectivity ⇒ " of arbitary shaped clusters

- clustering space: which parameters/attributes to consider while clustering.

⇒ K-means:-

→ Form random k-partitions.

→ repeat:

- compute centroid of each cluster
- Assign obj to nearest centroid cluster

until no change.

→ O(nkt) terminates at local optimal.

→ Weakness:

→ sensitive to noise / outliers

→ Need to specify 'k' in advance.

⇒ Variations:

- selection of initial k means
- dissimilarity calculations
- strategies to calc. cluster means.

⇒ K-modes

→ variation of k-means, which can cluster nominal data.

→ replace mean by mode.

→ a different dissimilarity func^n

→ freq. based method to update mode.

⇒ K-mediods

→ basic idea : in K-means, rather than mean value, select an actual object most similar to mean value of each cluster.
→ NP-hard.
→ Partitioning Around Mediods (PAM)
  ∴ randomly select initial K - objects $O_1 \dots O_k$
  repeat ~~for~~ i;
    replace O_i by ~~a~~ O_r ~~random~~ if quality increased
    ~~random~~ r → all possible points.
  until no change in quality
⇒ effective for small dataset, not scalable

⇒ Density-Based Clustering
- to find arbitrarily shaped clusters.
- handles noise
- one scan
- needs density parameter ~~as~~ for termination

→ DBSCAN:
→ Two parameters: Esp: radius of ε neighborhood.
  MinPts: density threshold.
→ core-objects: An object with contains atleast MinPts objects in its ε neighborhood.
→ clustering: use core-objects & its ε neigh., to form dense region i.e clusters.

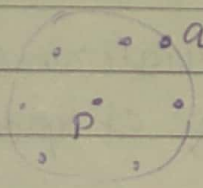
→ Some terminologies:

~ Directly-density reachable (ddr):
core-object q, object p; p is ddr from q if p belongs to ε-neighborhood of q.

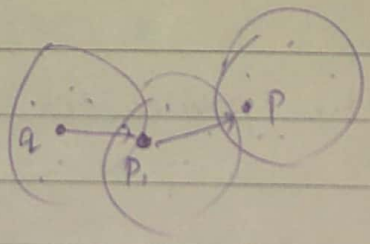
~ Density-reachable (dr):
~~For two points p & q are dr~~
A a point q is dr from p if for n points $p_1, p_2 \ldots p_n$, (where $p_1 = p$ & $p_n = q$) p_{i+1} is ddr from p_i.

~ Density connected:
Two points p & q are D.c. if ∃ a point O such that p is dr from O & q is also dr from O.

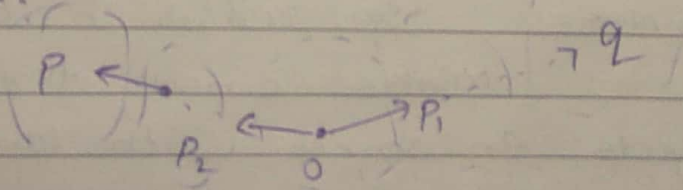
⇒ ddr:



⇒ dr:



⇒ dc:



⇒ DBSCAN Algorithm:

(*) (w.r.t Esp & MinPts)

∀ points p in database:

- retrieve all density-reachable points of p (*)
  - If p is core-point:
    - it is a cluster
  - else
    - donothing.

→ spatial indexing → O(n log(n)) else $O(n^2)$

⇒ OPTICS:

- users have to select/provide parameters for DBSCAN
- OPTICS: outputs a ordering of clusters.
- To construct clusters, object are processed in a specific ~~orb~~ order.
- Orders: select object that is dr with lowest ε (i.e high density clusters are first)
- core-distance: For an object p, core-distance ε' is the minimum value for which p̂ is a core-objec
- reachability-dist:- minimum radius value that makes p dr. From q
  = max{core-distance(q), dist(p,q)}.

⇒ **DENCLUE:-**

→ uses kernel density estimation (non-parametric)

→ general idea:

~~w~~ treat an observed object as an indicator of high-probability density in its surrounding

~~w~~ probability density ∝ dist. From this point to observed objects

For a random variable f,

Kernel density approx^n of P.D.F is

$$f_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x_i - x}{h}\right)$$

h → smoothing parameter

K → kernel func^n: $\int_{-\infty}^{\infty} K(u) \cdot du = 1$

$K(-u) = K(u)$

freq. used kernel func^n: $K\left(\frac{x_i - x}{h}\right) = \frac{1}{\sqrt{2\pi}} e^{\frac{-1}{2} \times \left(\frac{x_i - x}{h}\right)^2}$

(gaussian with $\mu = 0$ & $\sigma^2 = 1$)

~~w~~ a point x^* is density attractor

if → its a local maximum of estimated density func.

↳ $f(x^*) \geq$ some threshold.

~~de~~ these non-trivial x^*(s) are centres of clusters

Some func^n(s):

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

Total " on x = $f^D_{gaussian}(x) = \sum_{i=1}^{N} f(x, x_i)$

gradient of x in direction of x_i = $\nabla f^D_{gaussian}(x, x_i) = \sum_{i=1}^{N} (x - x_i) e^{\frac{-(d(x,x_i))^2}{2\sigma^2}}$

$= \sum_{i=1}^{N} (x - x_i) \cdot f_{gaussian}(x, x_i)$

⇒ STING:

→ divide the spatial area into rectangular cells

→ Arrange them in an hierarchy

- higher level cell ⇒ partitioned to smaller cells
- Use top-down approach to answer queries

→ remove irrelevant cells From further consid.

Advantages: Query-independent, incremental update, O(K), K → number of grids at lowest level.

⇒ CLIQUE:-

→ both density based & grid based.

- partitions m-dimensional data into non-overlaping rectangular units & each dimensional is partitioned into eq. number of intervals (of equal length)
- frac of data points ≥ input parameter for a unit to be dense contained in unit
- cluster ≈ maximal set of connect dense units in a subspace. dimensional

Adv. :- automatically finds subspace of highest
:- insensitive to order of records in input
:- linearly scalable.

Disadv:- trade off between accuracy & simplicity of method.

⇒ Evaluation of clustering :-

⇒ Assesing Clustering Tendency :

- check for existence of non-random because a clustering algo. may return random & non-meaningful clusters.
- statistical tests for spatial randomness :-

⇒ Hopkins Statistic :

dataset D

Step 1 : sample n random points $p_1 \cdots p_n$

$\forall i \in 1$ to n

$$x_i = \min_{v \in D} \{d(p_i, v)\}$$

Step 2 : 111 ^{ly} considers n points $q_1 \cdots q_n$

$\forall i \in 1$ to n

$$y_i = \min_{v \in D - \{q_1, q_2 \cdots q_n\}} \{d(q_i, v)\}$$

Step 3 :

$$H = \frac{\sum_{i=1}^{n} y_i}{\sum x_i + \sum y_i}$$

$H > 0.5$, (reject clustering)

⇒ Determining # clusters:

determining how many clusters to form, is a really difficult task.

⇒ most simple method ⇒ # clusters = $\sqrt{n/2}$

cluster size = $\sqrt{2n}$

→ Elbow Method

clusters ↑ ⇒ sum of intra-cluster variance ↓

Var(k) = total variance when k clusters are formed.

Plot curve of var against k. The first/most significant turning point of curve ~~is~~ suggest optimal value of k.