VIGNAN'S
Foundation for Science, Technology & Research
(Deemed to be University)
-Estd. u/s 3 of UGC Act 1956

Data Mining Techniques
Clustering
Unit V

Contents

VIGNAN'S
Foundation for Science, Technology & Research
(Deemed to be University)
-Estd. u/s 3 of UGC Act 1956

- ✓ **Introduction to Clustering**
- ✓ Similarity and distance measures
- ✓ Hierarchical Algorithms
- ✓ Partitioning Algorithms
- ✓ Clustering Large Databases
- ✓ Clustering with categorical attributes

What is Clustering

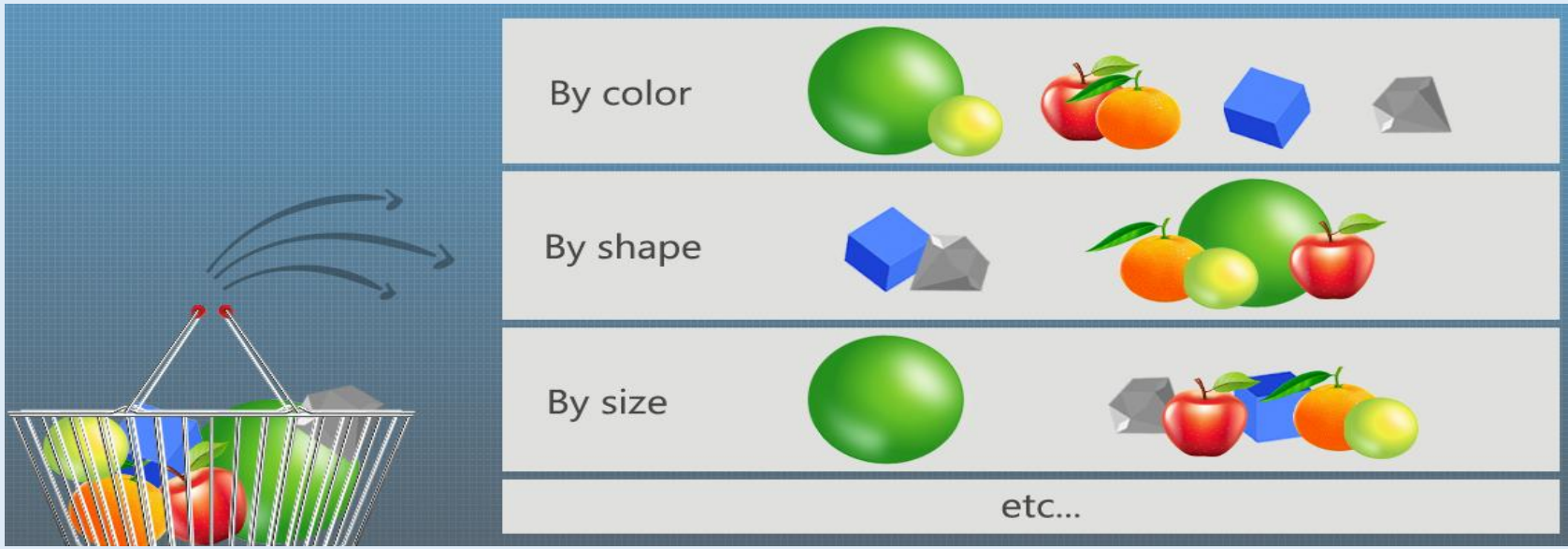
- ✓ Cluster analysis or Clustering is the process of Grouping a set of Data objects into different Groups based *on their similar characteristics*
- ✓ The objects in the same group (called a cluster) are more similar to each other than to those in other groups
- ✓ Clustering is also known as *unsupervised learning* because class Label information is not available.
- ✓ Maximize the intra-class similarity and minimize the inter-class similarity

What is Clustering

- ✓ Clustering is referred as ***Data Segmentation*** - Partitions large data sets into groups as per their similarity
- ✓ Clustering is also known as ***Partitioning*** because partitioning a set of data objects into subsets
- ✓ Finding groups of objects - objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups.

Applications of Clustering

- Identifying Fake News
- Spam filter
- Marketing and Sales
- Classifying network traffic
- Identifying fraudulent or criminal activity
- Document Analysis
- Fantasy Team Preparation in Sports



Types of Clustering

Exclusive versus Overlapping versus Fuzzy

- ✓ The clustering in which each object is assigned to a single cluster are called as exclusive clustering (same as partitioning)
- ✓ When an object is placed in more than one cluster, then the clustering is called as non-exclusive clustering or overlapping
- ✓ 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)

Types of Clustering

Complete versus Partial

- ✓ With complete clustering, every object is assigned to a cluster, whereas a partial clustering does not.
- ✓ The motivation for a partial clustering is that some objects in a data set may not belong to well-defined groups.
- ✓ Many times objects in the data set may represent noise, outliers, or uninteresting background

Types of Clustering

Well-Separated

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

Prototype-Based

A cluster is a set of objects in which each object is closer (more similar) to the prototype that defines the cluster than to the prototype of any other cluster.(in the example, the prototype is center)(center based).

Types of Clustering

Prototype based: Ex: Centroid based

A Cluster is a set of objects such that an object in a cluster is more closer to the center of a cluster than to the center of any other cluster.

The center of the cluster is often referred as centroid, the average of all the points in the cluster, or a medoid, the most representative point of a cluster.

Types of Clustering

Movie Recommendation systems are an example of:

- Classification
- **Clustering**
- Regression

Sentiment Analysis is not an example of:

- Regression
- Classification
- **Clustering**

Can decision trees be used for performing clustering?

- **True**
- False

Similarity & Dissimilarity Measures

- The term ***proximity*** is used to refer to either similarity or dissimilarity
- The proximity between two objects is a ***function of the proximity between the corresponding attributes of the two objects***.
- The similarity is a numerical measure of the degree to which the two objects are alike.

  Similarity is higher for pairs of objects that are more alike. In general, similarity is a non-negative and lies in between 0 (no similarity) and 1 (complete similarity).
- The dissimilarity between two objects is a numerical measure of the degree to which the two objects are different.

  Dissimilarities are lower for more similar pairs of objects.
- Frequently, the term distance is used as a synonym for dissimilarity.

Similarity & Dissimilarity Measures

Attribute Type	Dissimilarity	Similarity
Interval or ratio	d=|x-y|	s=-d, $s = \frac{1}{1+d'}$ s=e^{-d}, s=1-$\frac{d-\min\_d}{\max\_d-\min\_d}$
Nominal	d= $\begin{cases} 0 \; if \; x = y \\ 1 \; if \; x \neq y \end{cases}$	s= $\begin{cases} 1 \; if \; x = y \\ 0 \; if \; x \neq y \end{cases}$
Ordinal	d= $\frac{\|x-y\|}{(n-1)}$ (values mapped to integers 0 to n-1 where n is the number of values)	s= 1-d

Dissimilarity Measures

Dissimilarity is also termed as distance between two objects.

- Manhattan distance

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + ... + |x_{ip} - x_{jp}|$$

- Euclidean distance

$$d(i, j) = \sqrt{(|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + ... + |x_{ip} - x_{jp}|^2)}$$

- Minkowski distance

$$d(i, j) = \sqrt[q]{(|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + ... + |x_{ip} - x_{jp}|^q)}$$

- Supremum distance $d(i, j) = \sum_{i=1}^{n} \max(xi - yi)$

Properties of Distance Measure

- ✓ $d(i,j) \geq 0$ non negativity
- ✓ $d(i,i) = 0$
- ✓ $d(i,j) = d(j,i)$ symmetric
- ✓ $d(i,j) \leq d(i,k) + d(k,j)$ Triangular In equality

The distance must be positive definite.
The distance must be symmetric, so that the distance from x to y is the same as the distance from y to x. This is sometimes called the symmetry rule.
An object has a distance 0 from itself.
When considering three objects, x, y and z, the distance from x to z is always less than or equal to the sum of the distance from x to y and the distance from y to z. This is sometimes called the triangle rule.

Similarity/ Dissimilarity Measures

Observations (i.e., dependent variables) that occur in one of two possible states, often labeled zero and one. A contingency table for binary data

	1	0	*sum*
1	a	b	$a+b$
0	c	d	$c+d$
sum	$a+c$	$b+d$	p

Distance measure for symmetric binary variables: $$d(i, j) = \frac{b+c}{a+b+c+d}$$

Distance measure for asymmetric binary variables:

Jaccard coefficient (similarity measure for asymmetric binary variables): $$d(i, j) = \frac{b+c}{a+b+c}$$

Similarity/ Dissimilarity Measures

if x and y are 2 documents, then:

$$\cos(x,y)=\frac{x.y}{||x||||y||}$$

where . indicates the vector . product, $x.y = \sum_{k=1}^{n} x_k \cdot y_k$

and ||x|| is the length of the vector x, $||x|| = \sqrt{\sum_{k=1}^{n} x_k^2}$

Similarity/ Dissimilarity Measures

$$corr(x,y) = \frac{covariance(x,y)}{standard\ deviation(x) * standard\ deviation(y)} = \frac{s_{xy}}{s_x s_y}$$

where the standard statistical notations are used.

$$covariance(x,y) = s_{xy} = \frac{1}{n-1}\sum_{k=1}^{n}(x_k - \bar{x})(y_k - \bar{y})$$

$$\text{standard deviation(x)} = s_x = \sqrt{\frac{1}{n-1}\sum_{k=1}^{n}(x_k - \bar{x})^2}$$

$$\text{standard deviation(y)} = s_y = \sqrt{\frac{1}{n-1}\sum_{k=1}^{n}(y_k - \bar{y})^2}$$

Partitioning Methods

- Given a database of n objects, a partition clustering algorithm constructs k partitions of the data,
- In each cluster optimizes a clustering criterion, such as the minimization of the sum of squared distance from the mean within each cluster.
- Partitioning algorithms try to locally improve a clustering criterion.
- First, they compute the values of the similarity or distance, they order the results, and pick the one that optimizes the criterion.
- Hence, the majority of them could be considered as greedy-like algorithms.

Partitioning Methods – K Means

Input : 'k', the number of clusters to be partitioned; 'n', the number of objects.

Output: A set of 'k' clusters based on given similarity function.

Steps:

i) Arbitrarily choose 'k' objects as the initial cluster centers;

ii) Repeat,

a. (Re)assign each object to the cluster to which the object is the most similar; based on the given similarity function;

b. Update the centroid (cluster means), i.e., calculate the mean value of the objects for each cluster;

iii) Until no change

K Means – Pros & Cons

Strengths:

1. Relatively scalable and efficient in processing large data sets; complexity is O (ikn), where i is the total number of iterations, k is the total number of clusters, and n is the total number of objects. Normally, k<<n and i<<n.
2. Easy to understand and implement.

Weaknesses:

1. Applicable only when the mean of a cluster is defined; not applicable to categorical data.
2. Need to specify k, the total number of clusters in advance.
3. Not suitable to discover clusters with non-convex shape, or clusters of very different size.
4. Unable to handle noisy data and outliers.
5. May terminate at local optimum.
6. Result and total run time depends upon initial partition.

K Means – Problem

Cluster the following eight points (with (x, y) representing locations) into three clusters:

A1(2, 10), A2(2, 5), A3(8, 4), A4(5, 8), A5(7, 5), A6(6, 4), A7(1, 2), A8(4, 9)

Initial cluster centers are: A1(2, 10), A4(5, 8) and A7(1, 2).

The distance function between two points a = (x1, y1) and b = (x2, y2) is defined as-

$P(a, b) = |x2 - x1| + |y2 - y1|$

Use K-Means Algorithm to find the three cluster centers after the second iteration.

K Means – Iteration 1

Calculating Distance Between A1(2, 10) and C1(2, 10)-

$P(A1, C1) = |x2 - x1| + |y2 - y1| = |2 - 2| + |10 - 10| = 0$

Calculating Distance Between A1(2, 10) and C2(5, 8)-

$P(A1, C2) = |x2 - x1| + |y2 - y1| = |5 - 2| + |8 - 10| = 3 + 2 = 5$

Calculating Distance Between A1(2, 10) and C3(1, 2)-

$P(A1, C3) = |x2 - x1| + |y2 - y1| = |1 - 2| + |2 - 10| = 1 + 8 = 9$

K Means – Iteration 1

Given Points	Distance from center (2, 10) of Cluster-01	Distance from center (5, 8) of Cluster-02	Distance from center (1, 2) of Cluster-03	Point belongs to Cluster
A1(2, 10)	0	5	9	C1
A2(2, 5)	5	6	4	C3
A3(8, 4)	12	7	9	C2
A4(5, 8)	5	0	10	C2
A5(7, 5)	10	5	9	C2
A6(6, 4)	10	5	7	C2
A7(1, 2)	9	10	0	C3
A8(4, 9)	3	2	10	C2

K Means – Iteration 1

Cluster-01: First cluster contains points- A1(2, 10)

Cluster-02: Second cluster contains points-
- A3(8, 4)
- A4(5, 8)
- A5(7, 5)
- A6(6, 4)
- A8(4, 9)

Cluster-03: Third cluster contains points-
- A2(2, 5)
- A7(1, 2)

The new cluster center is computed by taking mean of all the points contained in that cluster.

For Cluster-01:
We have only one point A1(2, 10) in Cluster-01.
- So, cluster center remains the same.

For Cluster-02:
Center of Cluster-02
= ((8 + 5 + 7 + 6 + 4)/5, (4 + 8 + 5 + 4 + 9)/5)
= (6, 6)

For Cluster-03:
Center of Cluster-03
= ((2 + 1)/2, (5 + 2)/2)
= (1.5, 3.5)

The center of the three clusters are-

- C1(2,10)
- C2(6, 6)
- C3(1.5, 3.5)

K Means – Iteration 2

Given Points	Distance from center (2, 10) of Cluster-01	Distance from center (6, 6) of Cluster-02	Distance from center (1.5, 3.5) of Cluster-03	Point belongs to Cluster
A1(2, 10)	0	8	7	C1
A2(2, 5)	5	5	2	C3
A3(8, 4)	12	4	7	C2
A4(5, 8)	5	3	8	C2
A5(7, 5)	10	2	7	C2
A6(6, 4)	10	2	5	C2
A7(1, 2)	9	9	2	C3
A8(4, 9)	3	5	8	C1

K Means – Iteration 2

Cluster-01:
First cluster contains points-
•A1(2, 10)
•A8(4, 9)

Cluster-02:
Second cluster contains points-
•A3(8, 4)
•A4(5, 8)
•A5(7, 5)
•A6(6, 4)

Cluster-03:
Third cluster contains points-
•A2(2, 5)
•A7(1, 2)

For Cluster-01:
Center of Cluster-01
= ((2 + 4)/2, (10 + 9)/2)
= (3, 9.5)

For Cluster-02:
Center of Cluster-02
= ((8 + 5 + 7 + 6)/4, (4 + 8 + 5 + 4)/4)
= (6.5, 5.25)

For Cluster-03:
Center of Cluster-03
= ((2 + 1)/2, (5 + 2)/2)
= (1.5, 3.5)

The center of the three clusters are-

- C1(3, 9.5)
- C2(6.5, 5.25)
- C3(1.5, 3.5)

K Medoids Algorithm

Input: 'k', the number of clusters to be partitioned; 'n', the number of objects

Output: A set of 'k' clusters that minimizes the sum of the dissimilarities of all the objects to their nearest medoid.

i) Arbitrarily choose 'k' objects as the initial medoids;
ii) Repeat,

a. Assign each remaining object to the cluster with the nearest medoid;
b. Randomly select a non- medoid object;
c. Compute the total cost of swapping old medoid object.
d. If the total cost of swapping is less than zero, then perform that swap operation to form the new set of k- medoids.

iii) Until no change.

K Medoids - Problem

VIGNAN'S
Foundation for Science, Technology & Research
(Deemed to be University)
-Estd. u/s 3 of UGC Act 1956

Object	*X*	*Y*
1	8	7
2	3	7
3	4	9
4	9	6
5	8	5
6	5	8
7	7	3
8	8	4
9	7	5
10	**4**	**5**

Step 1:
Let randomly select 2 medoids, such as **C1 -(4, 5)** and **C2 -(8, 5)**

Step 2: Calculating cost.

Object	*X*	*Y*	*C1*	*C2*
1	8	7	6	**2**
2	3	7	**3**	7
3	4	9	**4**	8
4	9	6	6	**2**
5 – C2	8	5		
6	5	8	**4**	6
7	7	3	5	**3**
8	8	4	5	**1**
9	7	5	3	**1**
10 – C1	4	5		

$(8,7)$ & $(4, 5) = |8-4| + |7-5 \} = 6$

$(8,7)$ & $(8, 5) = |8-8| + |7-5| = 2$

K Medoids - Problem

The points 2, 3, 6 go to cluster C1 and 1, 4, 7, 8, 9 go to cluster C2.
The Cost = (3 + 4 + 4) + (3 + 1 + 1 + 2 + 2) = 20

Step 3: randomly select one non-medoid point and recalculate the cost.

Let the randomly selected point be (8, 4). The dissimilarity of each non-medoid point with the medoids – C1 (4, 5) and C2 (8, 4) is calculated and tabulated.
So, the points 2, 3, 6 go to cluster C1 and 1, 4, 5, 7, 9 go to cluster C2.
The New cost = (3 + 4 + 4) + (2 + 2 + 1 + 3 + 3) = 22
Swap Cost = New Cost – Previous Cost = 22 – 20 and 2 >0

As the swap cost is not less than zero, we undo the swap.

Object	X	Y	C1	C2
1	8	7	6	3
2	3	7	3	8
3	4	9	4	9
4	9	6	6	3
5	8	5	4	1
6	5	8	4	7
7	7	3	5	2
8 – C2	8	4		
9	7	5	3	2
10 – C1	4	5		

Pros & Cons of K Medoids

Strengths:

- More robust than k-means in the presence of noise and outliers; because a medoid is less influenced by outliers or other extreme values than a mean.

Weaknesses:

- Relatively more costly; complexity is O(i*k*(n-k)**2), where i is the total number of iterations, is the total number of clusters, and n is the total number of objects.
- Relatively not so much efficient.
- Need to specify k, the total number of clusters in advance.
- Result and total run time depends upon initial partition.

Hierarchical Clustering Algorithm

Also called **Hierarchical cluster analysis** or **HCA** is an unsupervised clustering algorithm which involves creating clusters that have predominant ordering from top to bottom.
For e.g: All files and folders on our hard disk are organized in a hierarchy.

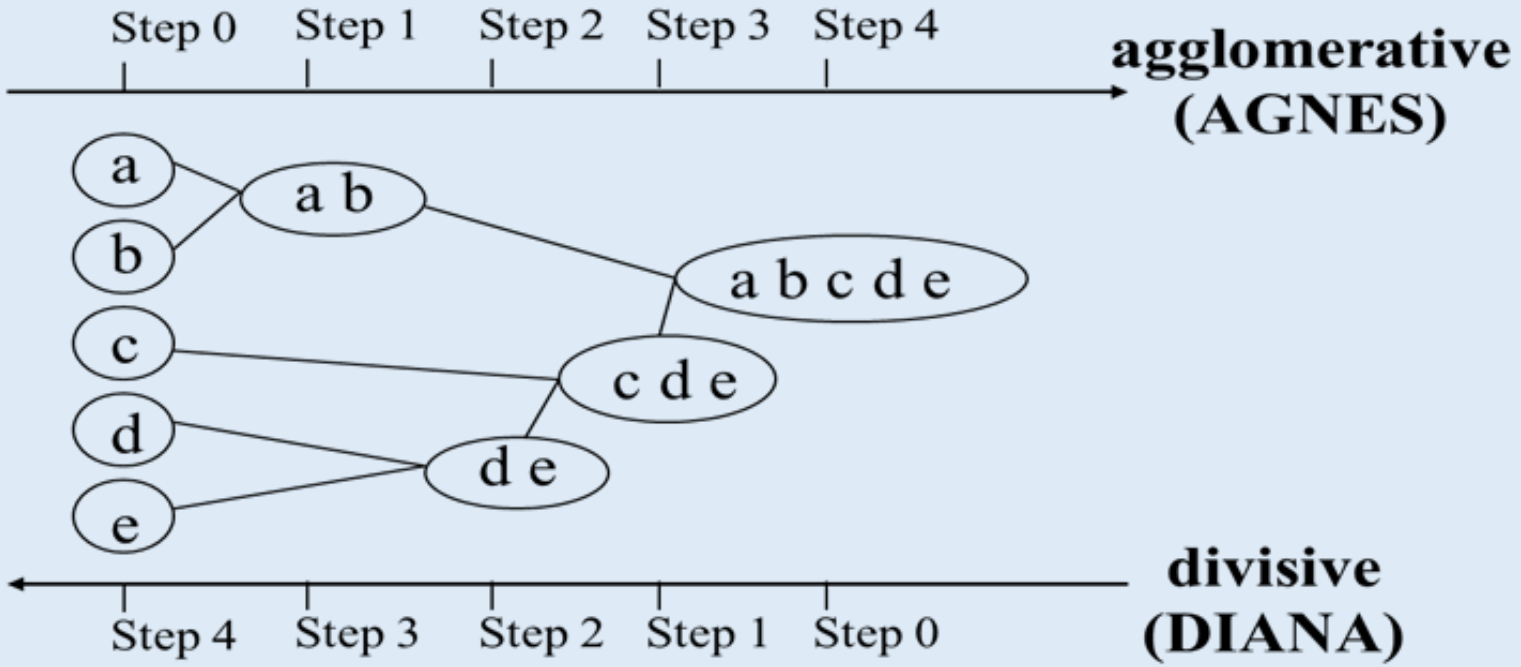
The algorithm groups similar objects into groups called ***clusters***.

The endpoint is a set of clusters or groups, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly similar to each other.

This clustering technique is divided into two types:

1. Agglomerative Hierarchical Clustering
2. Divisive Hierarchical Clustering

AGNES & DIANA



Min (Ab,Ac,Ad,Ae)
Ab
{(a,b), c,d,e}
{(a,b), c, (d,e)}
{(a,b), (c,d,e)}
(a,b,c,d,e)

Agglomerative Nesting

Algorithmic steps:

Let X = {x1, x2, x3, ..., xn} be the set of data points.

- Begin with the disjoint clustering having level L(0) = 0 and sequence number m = 0.
- Find the least distance pair of clusters in the current clustering, say pair (r), (s), according to d[(r),(s)] = min d[(i),(j)] where the minimum is over all pairs of clusters in the current clustering.
- Increment the sequence number: m = m +1. Merge clusters (r) and (s) into a single cluster to form the next clustering m. Set the level of this clustering to L(m) = d[(r),(s)].
- Update the distance matrix, D, by deleting the rows and columns corresponding to clusters (r) and (s) and adding a row and column corresponding to the newly formed cluster. The distance between the new cluster, denoted (r,s) and old cluster(k) is defined in this way: d[(k), (r,s)] = min (d[(k),(r)], d[(k),(s)]).
- If all the data points are in one cluster then stop, else repeat from step 2).

Divisive Analysis

Divisive Analysis called DIANA is just the reverse of Agglomerative Hierarchical approach.

Advantages

1) No apriori information about the number of clusters required.
2) Easy to implement and gives best result in some cases.

Disadvantages

1) Algorithm can never undo what was done previously.
2) Time complexity of at least $O(n^2 \log n)$ is required, where 'n' is the number of data points.
3) Based on the type of distance matrix chosen for merging different algorithms can suffer with one or more of the following:
    - Sensitivity to noise and outliers
    - Breaking large clusters
    - Difficulty handling different sized clusters and convex shapes
    - No objective function is directly minimized

Linkage Methods

There are several ways to measure the distance between clusters in order to decide the rules for clustering, and they are often called Linkage Methods.

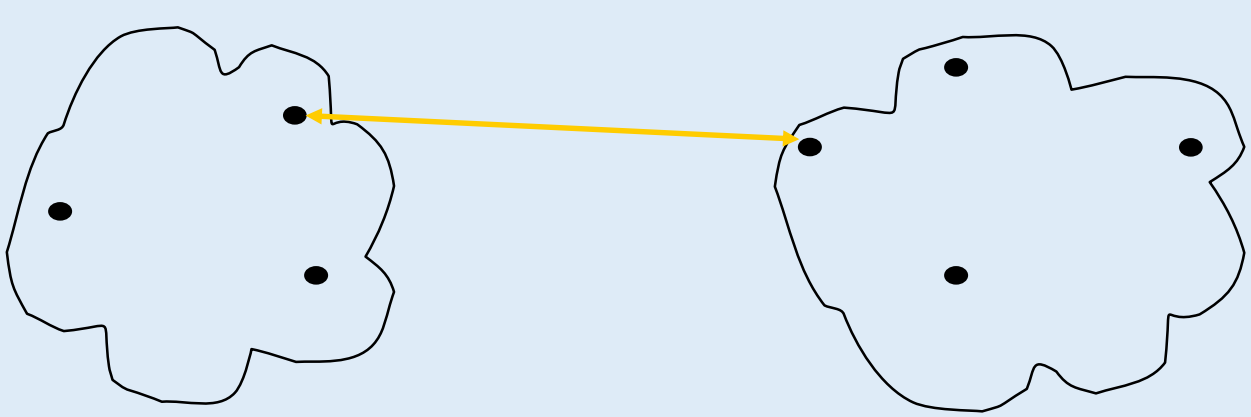
Single-linkage: the distance between two clusters is defined as the shortest distance between two points in each cluster. This linkage may be used to detect high values in your dataset which may be outliers as they will be merged at the end.

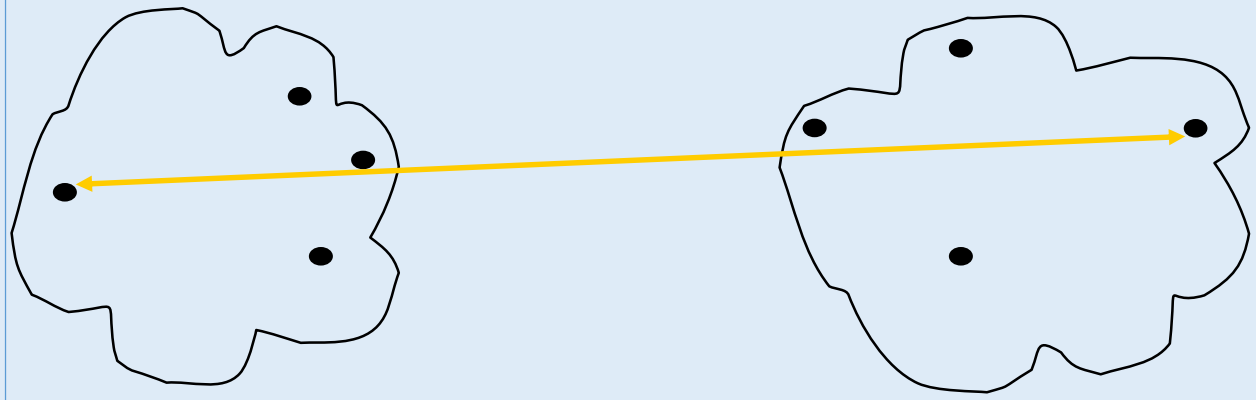
Complete-linkage: the distance between two clusters is defined as the longest distance between two points in each cluster.

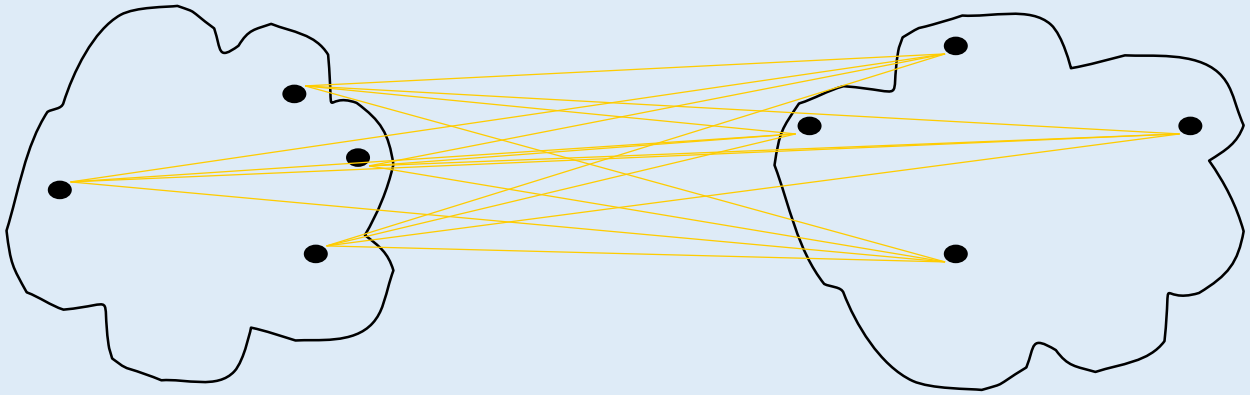
Average-linkage: the distance between two clusters is defined as the average distance between each point in one cluster to every point in the other cluster.

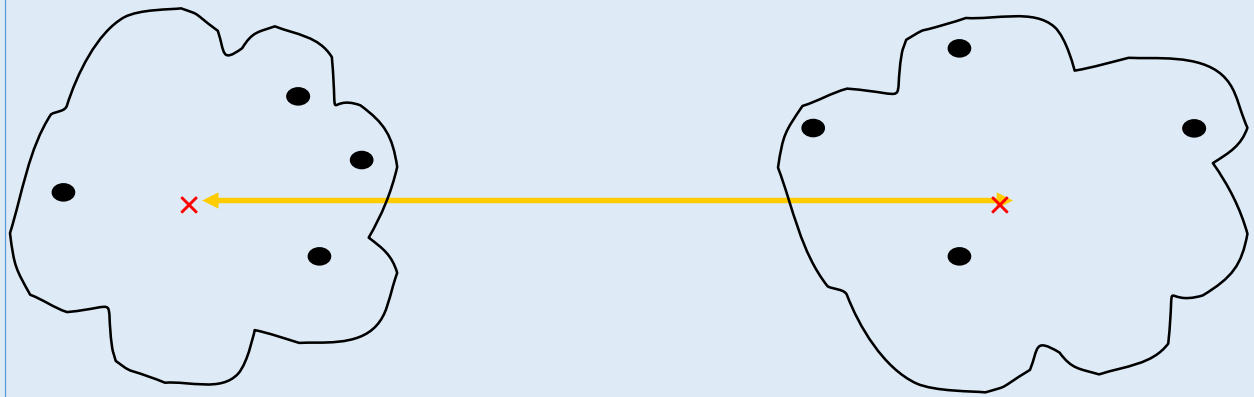
Centroid-linkage: finds the centroid of cluster 1 and centroid of cluster 2, and then calculates the distance between the two before merging.

Linkage Methods







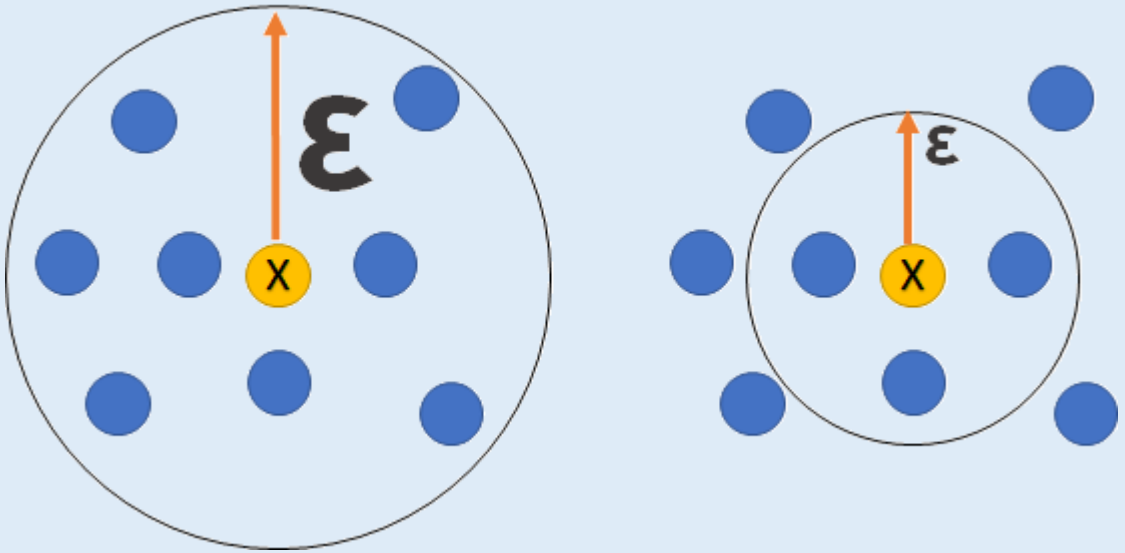


Density Based Spatial Clustering of Applications with Noise

Eps: Maximum radius of the neighborhood

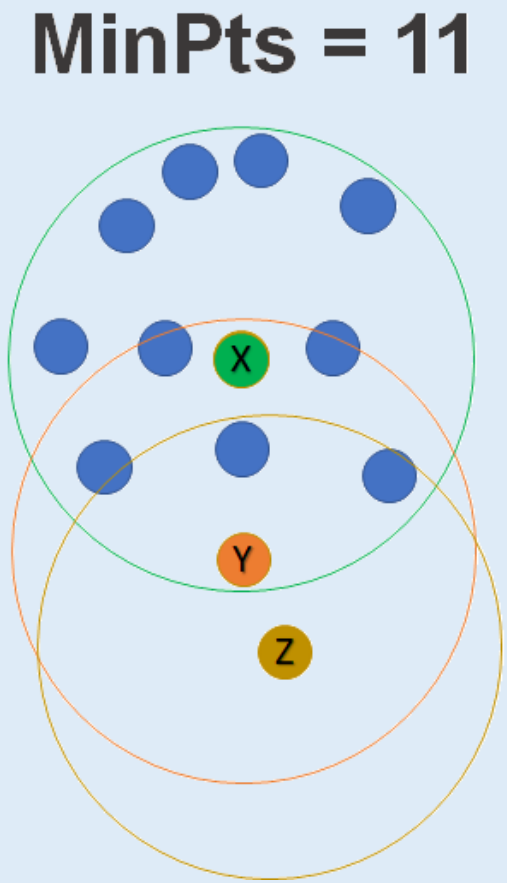
MinPts: Minimum number of points in an Eps-neighbourhood of that point

$N_{Eps}(p)$: *{q belongs to D | dist(p,q) <= Eps}*



Density Based Spatial Clustering of Applications with Noise

Directly density-reachable: A point *p* is directly density-reachable from a point *q* w.r.t. *Eps*, *MinPts* if *p* belongs to $N_{Eps}(q)$ core point condition: $|N_{Eps}(q)| >= MinPts$



Point y is directly density reachable from point x if:

- *Point y belongs to the ε-neighborhood of point x*
- *Point x is a core point.*

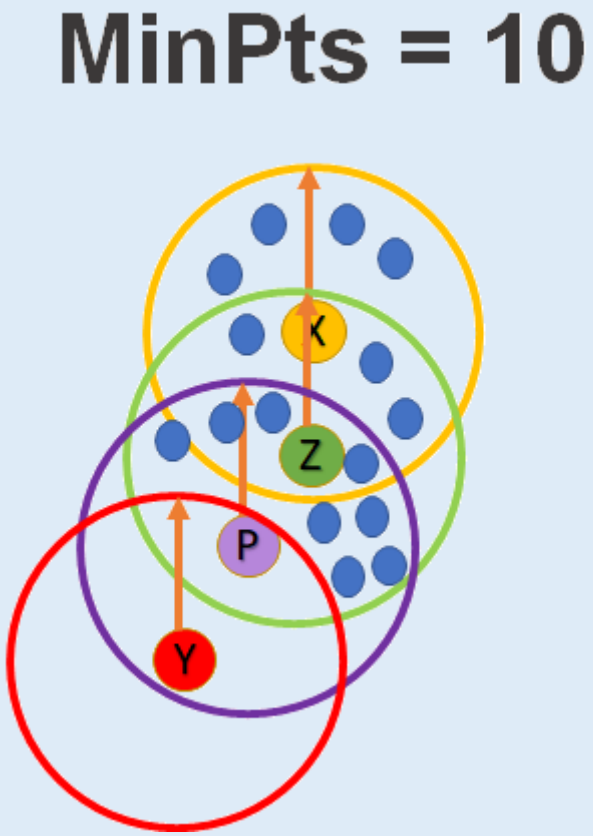
Density Based Spatial Clustering of Applications with Noise

Density-reachable: A point p is density-reachable from a point q w.r.t. *Eps*, *MinPts* if there is a chain of points $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i

Point y is density reachable from point x, if

there is a path of points between point x and point y, where each point in the path is directly reachable from the previous point.

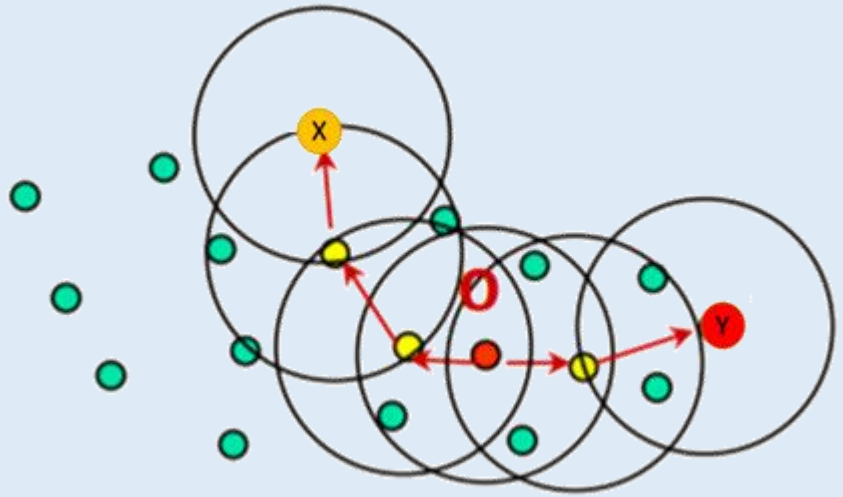
This means that all the points on the path are core points, with the possible exception of point y.



Density Based Spatial Clustering of Applications with Noise

Density-connected: A point *p* is density-connected to a point *q* w.r.t. *Eps*, *MinPts* if there is a point *o* such that both, *p* and *q* are density-reachable from *o* w.r.t. *Eps* and *MinPts*.

A point x is density-connected to a point y, if there is a point o that both x and y are density-reachable from.



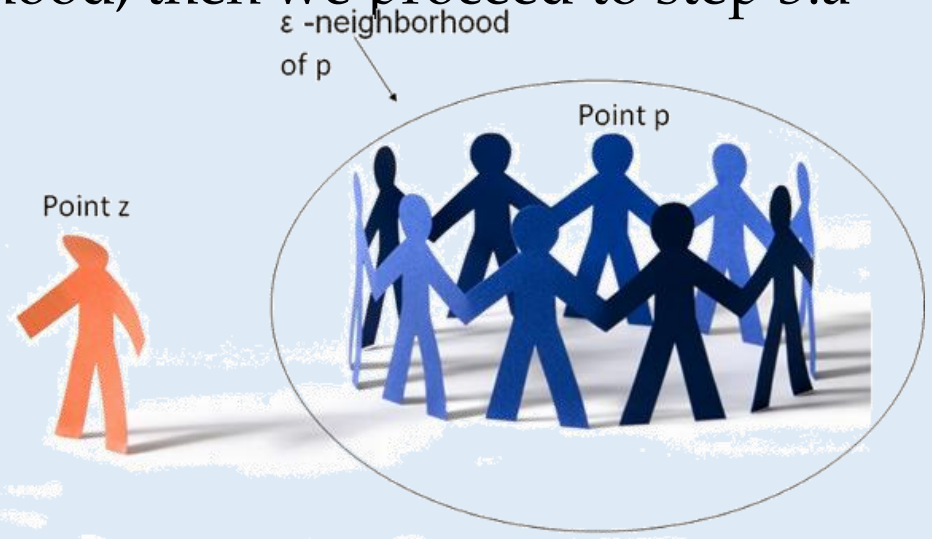
DBSCAN Algorithm

1) We mark all the points in the data as unvisited.

2) We choose a random unvisited point to visit, and mark it as visited. Let's refer to it as 'p'.

3) We check if the ε-neighborhood of p has at least MinPts points.

If 'p' doesn't have enough points in its ε-neighborhood, we mark it as noise and proceed to step 4. If point 'p' has enough points in its ε-neighborhood, then we proceed to step 3.a



DBSCAN Algorithm

3.a) We will create a new cluster C And we add 'p' to the cluster.

3.b) We will give the ε -neighborhood of p a new name — N.

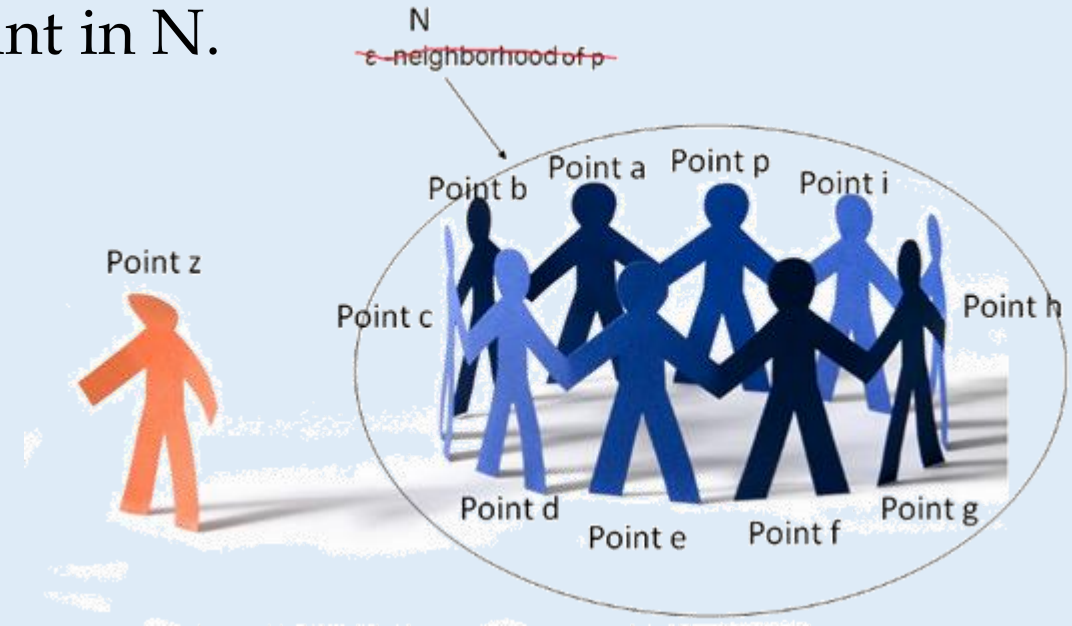
N will include the points: {a,b,c,d,e,f,g,h,i}

3.c) We'll follow the next steps for each point in N.

The first point is 'a'. If 'a' is unvisited

we will do steps 3.c.1 and 3.c.2.

Else, we will proceed to step 3.d.



DBSCAN Algorithm

3.c.1) We will mark point 'a' as visited.

3.c.2) We will check if point 'a' has at least MinPts in its ε-neighborhood. If so we will add these points to N. for example if the ε -neighborhood of 'a' includes the points {j ,k,l } and MinPts is 3, the new N would include the points: {a,b,c,d,e,f,g,h,i,j,k,l}. But, if the ε -neighborhood of 'a' would include just point {j}, N would stay the same.

3.d) If point 'a' 'doesn't belong to any cluster, we will add it to cluster C, and now cluster C will include points 'a' and 'p'.

3.e) We will move to the next point in N (point 'b') and go back to step 3.c. We will finish to repeat steps 3.c-3.e after we checked all the points in N.

4. We're done with 'p'. We will go back to step 2, visit the next unvisited point and repeat these steps. We will finish when all the points in the dataset have been visited.

CURE (Clustering using Representative) Algorithm

First partitions the random sample and partially clusters the data points in each partition. After eliminating outliers, the pre clustered data in each partition is then clustered in a final pass to generate the final clusters.

(1) The clustering algorithm can recognize arbitrarily shaped clusters (e.g., ellipsoidal),

(2) The algorithm is robust to the presence of outliers

(3) The algorithm uses space that is linear in the input size n and has a worst-case time Complexity of $O(n^2 \log n)$. For lower dimensions (e.g., two), the complexity can be shown to further reduce to $O(n^2)$.

(4) It appropriate for handling large data sets.

BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)

BIRCH is an agglomerative hierarchical clustering algorithm proposed by Charikar et al. in 1997.

It is especially suitable for very large databases.

It is used to minimize the number of I/O operations.

BIRCH incrementally and dynamically clusters incoming multi-dimensional metric data points to try to produce the best quality clustering with the available resources.

BIRCH can typically find a good clustering with a single scan of the data, and improve the quality further with a few additional scans.

BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)

BIRCH is also the first clustering algorithm proposed in the database area to handle "noise" (data points that are not part of the underlying pattern) effectively.

The data pre-processing algorithm BIRCH groups the data set into compact sub clusters that have summary statistics (called Clustering Features (CF)) associated to each of them.

These CF's are computed and updated as the sub clusters are being constructed.

Clustering Feature is a triplet defined as CF = (N, LS, SS)

N: Number of data points

LS: $\sum^{N}_{i=1}=X_i$

SS: $\sum^{N}_{i=1}=X_i^2$

Clustering with categorical attributes

VIGNAN'S
Foundation for Science, Technology & Research
(Deemed to be University)
-Estd. u/s 3 of UGC Act 1956

The traditional clustering algorithms focused on clustering numeric data by exploiting the inherent geometric properties of the dataset for calculating distance functions between the points to be clustered.

The distance based approach did not fit into clustering real life data containing categorical values.

The focus of research then shifted to clustering such data and various categorical clustering algorithms are proposed till date.

The clustering of categorical data turns complex because of the absence of a natural order on the individual domains, high dimensionality of data and the existence of subspace clusters in the categorical datasets.

Clustering with categorical attributes

VIGNAN'S
Foundation for Science, Technology & Research
(Deemed to be University)
-Estd. u/s 3 of UGC Act 1956

The similarity between data points is calculated through a similarity/distance measure.
The very first of the proposed clustering algorithms concentrated on clustering numeric data through the use of derived ideas from statistics and geometry.
With changing requirements and time, it was observed that the real life data contains categorical values and not numeric values and hence limited the scope of the existing clustering algorithms to numeric data only.
Categorical data is different from numeric data in the sense that it groups the data into categories and not any numeric values.
For example, a set of {male, female, children} can be used to categorize a group of people.
This set cannot be clustered based on the distance between the people present.