Module - 5

Clustering Analysis

Nithin Kumar
Asst professor
Dept of CSE&E
VVCE, Mysuru

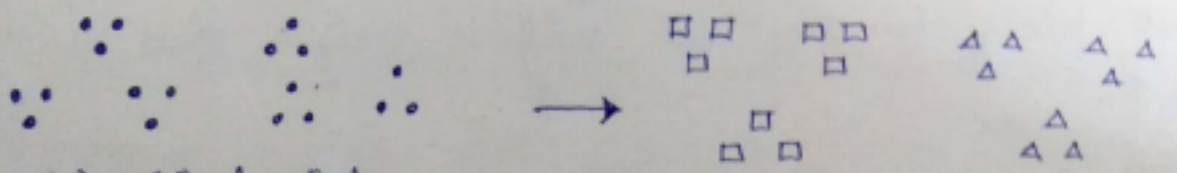
Cluster Analysis groups data objects based only on information found in the data that describes the objects & their relationships.

→ The goal is that the objects within a group be similar to one another & different from the objects in other groups.

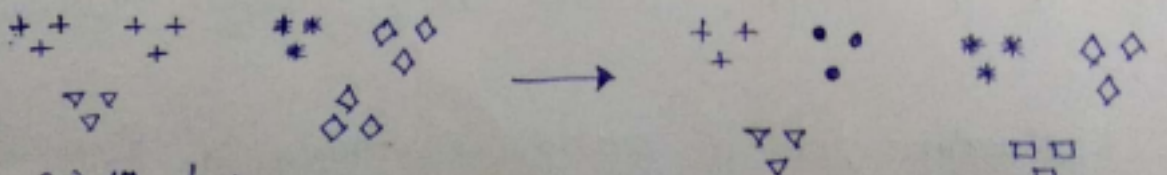
The greater the similarity within a group & the greater the difference b/w groups the better or more distinct the clustering.

* Different types of clusterings :- The various types of clusterings are

* Hierarchical Vs partitional :- 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.



(a) original points (b) 2 - clusters



(c) 4 - clusters (d) 6 - clusters

→ "Hierarchical clustering" is a set of nested clusters that

are organized as a tree.
Each node (cluster) in the tree is a union of its children (sub clusters) & root of the tree is the cluster containing all objects but leafs of trees are single clusters of individual data objects.

* Exclusive V/s overlapping V/s Fuzzy :- In "Exclusive clustering", we assign each object to a single cluster. There are many situations in which a point could reasonably be placed in more than one cluster & these situations are better addressed by Non-Exclusive clustering.

→ "Non-Exclusive" or "overlapping" clustering is used to reflect the fact that an object can simultaneously belong to more than one group (class)
For Ex, A person at a University can be both an enrolled student & Employee of University.

→ "Fuzzy" clustering, where every objects belongs to every cluster with membership weight that is between "0 & 1"
"0" (Absolutely doesn't belong) & "1" (Absolutely belongs).
clusters are treated as fuzzy sets i.e one in which an object belong to any set with a weight that is between 0 & 1.

* Complete V/s partial :- A complete clustering assigns every object to a cluster, where as 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.

→ For Example, Some Newspaper Stories May have a Common theme, while other Stories are More Generic or "one-of-a-kind".

* Different Type of clusters :-

Clustering Aims to find useful Groups of objects (clusters), where Usefulness is defined by the Goals of Data Analysis.

→ There are Several different Notations of clusters that prove -ve useful in practice

* Well-Separated :- The data objects with a Cluster Must have Small distance & distance b/w two clusters Must be High.

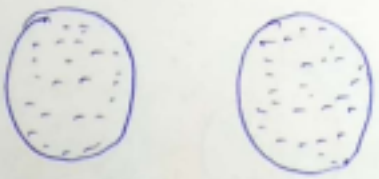


fig:- Well-Separated clusters

Sometimes a threshold is used to Specify that all the objects in a cluster Must be Sufficiently close (Similar) to one Another.

* prototype-Based clusters (Center-Based) :- A cluster is a Set of objects in which Each object is closer (Similar) to the prototype that defines the cluster than to the prototype of Any other clusters.

fig:- proto-type-Based Cluster

* Graph - Based clustering (Contiguity - Based) :- If the data is represented as graph, where the nodes are objects & the links represent connection among objects then a cluster can be defined as Connected Component.

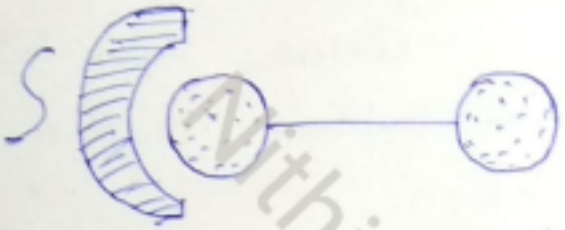


fig: Graph - Based cluster

* Density - Based cluster :- A cluster is a dense region of objects that is surrounded by a region of low density

A density Based def^{n} of cluster is often Employed when the clusters are irregular or intertwined & when Noise & outliers are present.

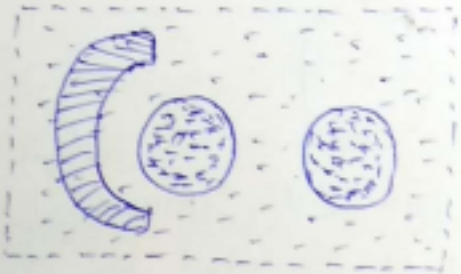


fig: Density - Based cluster

* Shared - property clusters (Conceptual) :- More Generally, A cluster as a set of objects that share some property

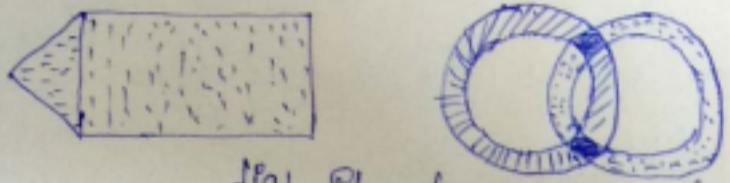


fig: Shared - property clusters

A clustering Algo would need a very specific concept of a cluster to successfully detect these clusters, the process of finding such clusters is called "Conceptual clustering"

* K-Means :-

K-Means clustering intends to partition 'n' objects into k-clusters in which each object belongs to the cluster with the nearest mean.
This method produces exactly "k" different clusters of greatest possible distinction.

→ It is a prototype-based clustering technique. Creates a one-level partitioning of data objects.
There are a no of such techniques but two of most prominent are "K-Means" & "K-Medios".

→ The objective of K-Means clustering is to minimize total "intra-cluster variance" or "squared error function".

→ In K-means, we first choose "k" initial centroids, where "k" is a user-specified parameter, namely the no of clusters desired.
Each point is then assigned to closest centroid & each collection of points assigned to a centroid is a cluster.

→ The centroid of each cluster is then updated based on the points assigned to the cluster.
We repeat the assignment & update steps until no point changes clusters or equivalently until the centroids remain the same.

* 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.

k-Means Algorithm was stated generally as "recompute the Centroid of each cluster".

→ "Data In Euclidean Space", Euclidean distance is one which Measures the quality of a clustering. using "Sum of Squared Error (SSE)" which is also known as "Scatter"

In other words, we calculate the Error of each data point i.e its Euclidean distance to the closest centroid & then compute the total Sum of Squared Errors.

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

where "dist" is the Standard Euclidean distance b/w two objects in Euclidean Space.

→ The Centroid (Mean) of i^{th} cluster is defined by Eqn"

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

→ "Document Data", our objective in k-Means is to Maximize the Similarity of the Documents in a cluster to cluster Centroid.

This quality is known as "cohesion" of cluster.

The total cohesion is given by = $\sum_{i=1}^{K} \sum_{x \in C_i} cosine(x, c_i)$

* k-Means Additional Issues :- There are two Strategies that decrease the total SSE by increasing the no of clusters

* "Split a cluster" → Cluster with the largest SSE is usually chosen, but we could split the cluster with large deviat

-ion for one particular attribute.

* "Introduce a new cluster' Centroid" → often the point that is farthest from any cluster will be chosen.
Another Approach is to choose randomly from all points or from the points with highest SSE.

There are two Strategis that decrease the no of clusters, while trying to Minimize the total SSE are

* "Dispose a cluster" → this is accomplished by removing the Centroid that corresponds to cluster & reassigning the point to other clusters.

* "Merge two clusters" → The cluster with closest centroids are typically chosen, although another Approach to Merge the two clusters that result in Small Increase in total SSE.

* Bisecting K-Means :- Algorithm is a Straight forward Extension of basic K-Mean Algo that is based on Simple Idea to obtain k clusters.

→ Split the Set of all points into two clusters, Select one of these clusters to Split & so on.
until k-clusters have been produced.

Algorithm

1. Initialize the list of clusters to contain the cluster containing of all points.
2. Repeat
3. Remove a cluster from the list of clusters
4. for i="1" to "number of trials" do
5. Bisect the Selected cluster using basic K-Means
6. End for
7. Select the two clusters from Bisection with lowest total SSE

8. Add these clusters to the list of clusters
9. until the list of clusters contains k clusters.

→ Consider the foll Example of Bisecting K-Means.

Iteration 1 Iteration 2 Iteration 3

→ The "Time Complexity" & "Space Complexity" of K-Means Algo is

* Space complexity → $O((m+k)n)$
* Time Complexity → $O(I*k*m*n)$

I = no of iterations m = no of points n = no of attributes.

* Strengths & Weakness:- K-Means is simple & can be used for a wide variety of data types, It is also quite efficient even though multiple runs are often performed.
* K-Means are even more efficient & less susceptible to "initialization problems".
* K-Means is not suitable for all types of data.
* K-Means also have trouble in clustering data that contains outliers.
* K-Means is restricted to data for which there is a notion of center (centroid).

* K-Means Example problem:- Consider the foll data set

$K = \{2, 3, 4, 10, 11, 12, 20, 25, 30\}$

& No of clusters to formed = 2 i.e $k = 2$, Random Mean value m_1 & m_2 = 4 & 12

⇒ $k = \{2, 3, 4, 10, 11, 12, 20, 25, 30\}$ $k = 2$ $M_1 = 4$ $M_2 = 12$

Assign Each point to the closest Centroid (M_1 & M_2) & form $k = 2$ clusters

$\therefore k_1 = \{2, 3, 4\}$ $k_2 = \{10, 11, 12, 20, 25, 30\}$

$$k_1 = \frac{2+3+4}{3}$$ $$k_2 = \frac{10+11+12+20+25+30}{6}$$

new $m_1 = 3$ new $m_2 = 18$

→ Again Assign Each point to the closest new Centroid (M_1 & M_2) & form $k = 2$ clusters

$M_1 = 3$ $M_2 = 18$

$\therefore k_1 = \{2, 3, 4, 10\}$ $k_2 = \{11, 12, 20, 25, 30\}$

$$m_1 = \frac{2+3+4+10}{4} = 4.75$$ $$m_2 = \frac{11+12+20+25+30}{5} = 19.6$$

new $m_1 = 5$ new $m_2 = 20$

→ Again Assign Each point to the closest new Centroid (M_1 & M_2) & form $k = 2$ clusters

$M_1 = 5$ $M_2 = 20$

$\therefore k_1 = \{2, 3, 4, 10, 11, 12\}$ $k_2 = \{20, 25, 30\}$

$$m_1 = \frac{2+3+4+10+11+12}{6}$$ $$m_2 = \frac{20+25+30}{3}$$

new $m_1 = 7$ new $m_2 = 25$

→ Again Assign Each point to the closest new Centroid (M_1 & M_2) & form $k = 2$ clusters

$M_1 = 7$ $M_2 = 25$

$\therefore k_1 = \{2, 3, 4, 10, 11, 12\}$ $k_2 = \{20, 25, 30\}$

$$m_1 = \frac{2+3+4+10+11+12}{6} = 7 //$$ $$m_2 = \frac{20+25+30}{3} = 25 //$$

Thus we are getting the same Mean value we have to Stop.

∴ Two well Separated Clusters of K are

$K_1 = \{2, 3, 4, 10, 11, 12\}$ $K_2 = \{20, 25, 30\}$ //

* Agglomerative Hierarchical Clustering :-

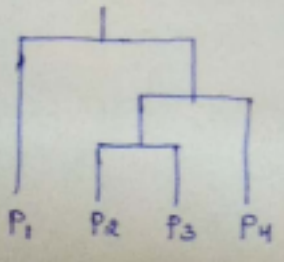
Hierarchical Clustering techniques are a second important Category of Clustering Methods. These Approaches are relatively old compared to many Clustering Algorithms but they still enjoy widespread use.

→ There are two basic Approaches for generating a Hierarchical Clustering

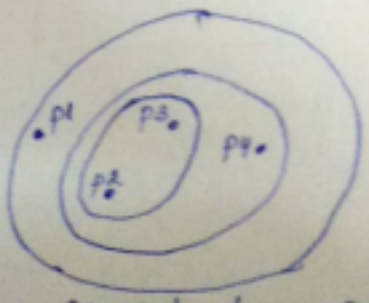
* Agglomerative → Start with point as individual clusters & at each Step, "Merge" the closest pair of clusters.

* Divisive → Start with one, all inclusive cluster & at each Step "Split" a cluster until only Singleton clusters of individual points remain.

→ A Hierarchical Clustering is often displayed graphically using a tree-like diagram called "dendrogram" which displays both cluster-subcluster relationships & order in which the clusters were Merged or Split.



(Dendrogram)



(Nested-cluster Diagram)

A Hierarchical Clustering can also be Graphically represented

Using a "Nested cluster diagram".

* Agglomerative Hierarchical clustering Algorithm :- Starting with individual points as clusters, successively merge the two closest clusters until only one cluster remains.

The Algorithm is as follows :-

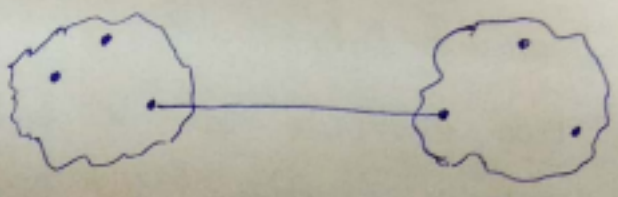
1. Compute the proximity Matrix, if necessary
2. Repeat
3. Merge the two closest clusters
4. Update the proximity Matrix to reflect the proximity between New cluster & the original cluster
5. Repeat until only one cluster remains.

* Defining proximity between clusters :- Cluster proximity is typically defined with a particular type of cluster in mind.

→ The different proximity Measures in Agglomerative clustering are

* MIN
* MAX
* Group Average

* MIN → Defines clusters proximity as the proximity between the closest two points that are in different clusters or the shortest Edge between two nodes in different Subset of nodes.



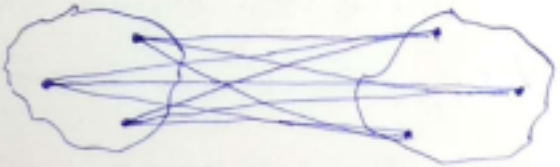
(MIN)

* MAX → Defines clusters proximity as the proximity between

the farthest two points that are in different clusters or the longest Edge between two nodes in different Subset of nodes.

(MAX)

* Group Average → Defines cluster proximity to be the Average pairwise proximities of all pairs of points from different clusters.



(Group Average)

→ An alternative technique "Ward Method" also assumes that a cluster is represented by its Centroid, but it Measures proxi-mity between two clusters in terms of "SSE" that results from Merging the two clusters like "K-Means".

* Strength & Weakness of Agglomerative clustering are

* Agglomerative Algorithm can produce Better-Quality Clusters.
* Agglomerative Algorithm are Expensive in terms of their Computation & Storage requirements.
* All Merges can cause trouble of Noise in High-Dimension Data Such as Document Data.

Above Mentioned two problems can be addressed to some degree by first partially clustering the Data using Another Technique Such as K-Means.

* DBSCAN :-

Density - Based clustering locates region of high Density that are separated from one another by region of low - Density

→ DBSCAN is a Simple & Effective Density - Based clustering Algorithm that illustrates a no of important Concepts that are important for any Density - Based clustering Approach.

* Center - Based Approach :- There are several distinct Methods but DBSCAN is based on Center - Based Approach

→ In the "Center - Based Approach", density is estimated for a particular point in data set by counting the no of points within Specified radius "Eps" of that point (this includes point itself).

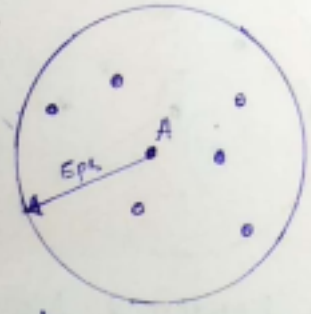


fig: Center - Based Approach

The no of points within a radius of Eps of point A is "7".

* Classification of points According to Center - Based Approach :- The Center - Based Approach to density allows us to classify a point into all three categories

* Core point
* Border point
* Noise or Background point

* Core points → These points are in the interior of density Based cluster.

"A point is a core point, if the no of points within a Given neighborhood around the point as determined by user Specified parameter (Distance Function (Eps)) Exceeds a Certain threshold "Minpts" which is also user-Specified parameter"

Ex:- point A is core point for radius Eps if Minpts ≤ 7

* Borden points → A Borden point is not core point, but Falls within the neighbourhood of core point.
A Borden point can fall within neighbourhoods of Several core points.

* Noise points → A Noise point is any point that is Neither a core point nor a Borden point.

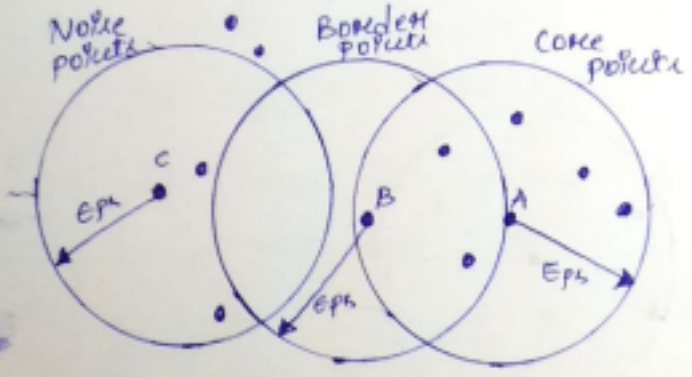


Fig:- Core, Borden & Noise points

* DBSCAN Algorithm :- Can be Informally described as follow
Any two core points that are close Enough within a distance Eps of one Another are put in Same Cluster.
→ Any Borden point that is close Enough to a core point is put in Same cluster as core point.
Noise points are Discarded.
→ "DBSCAN Algorithm"
1. label all points as core, Borden & Noise
2. Eliminate Noise points.

3. put an Edge between all core points that are within Eps
4. Make Each group of Connected Core points to Separate Cluster.
5. Assign Each Border point to one of cluster of its Associated Core points.

* Strength & Weakness :- DBSCAN uses a density-based defn of a cluster
* It is relatively resistant to Noise & can handle clusters of arbitary Shapes & Sizes
* DBSCAN has trouble when the clusters have widely varying densities.
* It also has trouble with high-dimensional data because density is more difficult to define for such data.
* DBSCAN can be Expensive when the computation of Nearest Neighbours requires computing all pairwise proxi-mities.

* Cluster Evaluation :- (Cluster Validation)

The Evaluation of the resulting classific-ation Model is an Integral part of the process of developi-ng a classification Model & there are well-Separated Mea-sures & procedures Ex "Accuracy & Cross-validation" Etc.

→ "Cluster Evaluation" should be a part of any cluster Analysis, A Key Motivation is that almost Every clustering Algo will find clusters in a data set, Even if that data Set has no Natural Cluster Structure.