ch-10 Cluster Analysis (Kamber - 3rd Ed.)

Clustering is the process of grouping a set of data objects into multiple groups or 'clusters' so that objects within a cluster have high similarity, but very dissimilar to objects in other clusters.

Clustering as a data mining has application areas such as:-

- Biology
- security.
- business intelligence
- web search.

clustering techniques are divided into categories :-

(I) Partitioning methods.
(II) hierarchical methods.
(III) density-based methods.
(IV) grid-based methods.

Objectives / use of cluster Analysis :-

1) It is a data mining tool to gain knowledge about the distribution of data. (How data is distributed.)
2) Used to observe characteristics of each cluster.
3) It can also be used to focus on a particular set of clusters for further analysis.
4) serve as a preprocessing step for algorithms such as characterization, attribute subset selection & classification.
5) clustering can automatically find the groupings.

Clustering as Automatic classification :- (6)

Since, cluster is a collection of data objects that are similar to one another within a cluster & dissimilar to objects in other clusters, thus cluster of data objects can be treated as an implicit class. That is why, clustering is sometimes called automatic classification.

clustering as data segmentation (7)

In some applications, clustering partitions large data sets into groups according to their similarity

clustering can also be used for outlier detection. (8)
(where values are far away from any cluster)

clustering is known as unsupervised learning because class label information is not present. That is why, clustering is a form of learning by observation, rather than learning by examples.

Requirements for Cluster Analysis

1) Scalability :- Highly scalable clustering algorithms are needed as many of the clustering algorithms work well on small data sets but not on large data sets.

2) Ability to deal with different types of attributes :- Algorithms are designed to cluster numeric, binary, nominal, ordinal, or mixture of these data types.

3) Discovery of clusters with arbitrary shape :- Many clustering algorithms determine clusters based on Euclidean distance measures. Algorithms based on these measures tend to find spherical clusters with similar size & density, But a cluster can be of any shape. Thus it is important to develop algorithms that can detect clusters of arbitrary shape.

4) Ability to deal with noisy data :-

Clustering Algorithms can be sensitive to noise (outliers, missing or erroneous data) & may produce poor-quality clusters. Thus, we need clustering methods that are robust (strong) to noise.

5) Requirements for domain knowledge to determine input parameters:-

many clustering algorithms require users to provide domain knowledge in the form of input parameters such as desired number of clusters.

6) Incremental clustering and insensitivity to input order:-

clustering Algorithms may return different clusters depending on the order in which objects are presented.

7) Capability of clustering high-dimensionality data

Finding clusters of data objects in a high-dimensional space is challenging especially when data is very sparse & highly distorted (skewed).

8) Constraint-based clustering :-

challenging task is to find data groups with good clustering behavior that satisfy specified constraints.

9) Interpretability & usability :- users want clustering results to be interpretable, comprehensible & usable.

Orthogonal aspects (Independent) with which clustering methods can be compared :-

1) The Partitioning Criteria :- objects can be partitioned on various criterias (say) ① where no hierarchy exist among clusters ie all clusters are at the same level. for eg. - grouping customers into groups so that each group has its own manager.
② where hierarchy exists :- where clusters can be formed at different levels. for eg:- different games can exist as subtopics of sports.

2) Separation of clusters:- whether the clusters are mutually exclusive or not.

3) Similarity measures :- methods are to be used to determine the similarity b/w two objects so that they can lie within the same cluster. for eg. the ① distance b/w these objects. or ② connectivity based on density & not on distance.

4) clustering space :- search for clusters within different subspaces of the same data set.

Requirements of clustering Algorithms

1) scalability & ability] to deal with different types of attributes noisy data, incremental updates, clusters of arbitrary shapes & contraints.

2) Interpretability & usability are important.

3) clustering methods can differ with respect to the partitioning level, whether or not clusters are mutually exclusive, the similarity measures used & whether or not subspace clustering is performed.

Overview of Basic clustering methods

(1) Partitioning Methods :- Given a set of 'n' objects, a partitioning method constructs 'k' partitions of the data, where each partition represents a cluster & $k \leq n$.

- It find mutually exclusive clusters of spherical shape.
- most partitioning methods are distance based.
- Given 'k', (the no. of partitions of construct), partitioning method creates an initial partitioning. It then uses an 'iterative relocation technique' to improve partitioning by moving objects from one group to another.

- Criteria of good partitioning :- objects in same cluster are close to each other whereas objects in different cluster are "far apart" or very different.
- Greedy approaches like k-means & k-medoids algorithms are used to improve the clustering quality & approach a local optimum. Also known as heuristic methods.
- Partitioning methods work well for finding spherical-shaped clusters in small-to-medium size databases.

(2) Hierarchical methods

- creates a hierarchical decomposition of the given data set objects. (ie multiple levels).
- Hierarchical methods are classified as :-
  - (a) Agglomerative method (Bottom up approach) [merges]
  - (b) Divisive method (Top-down approach) [split]
- These methods cannot correct erroneous merges or splits.
- These methods can be distance-based or density- and-continuity-based.
- May incorporate other techniques like microclustering or consider object "linkages".

(2) Density-based methods :-

- Continue growing a given cluster as long as the density (no. of objects or data points) in the "neighborhood" exceeds some threshold.
for eg :- for each data point within a given cluster the neighborhood of a given radius has to contain at least a minimum no. of points. Such methods help to filter out noise or outliers & discover clusters of arbitrary shape.

- clusters are dense regions of objects in space that are separated by low-density regions.
- cluster density :- Each point must have a minimum number of points within its "neighborhood".
- Density based methods can be extended from full space to subspace clustering.

(3) Grid-based methods :-

- These methods quantize the # object space into finite number of cells that form a grid structure. ie it uses multiresolution grid data structure.
- Advantage of this approach is fast processing time, which is independent of the number of data objects & dependent only on the number of cells in each dimension in the quantized space.
Application :- Spatial Data mining problems.

Clustering :

Given a dataset D with 'n' points in a d-dimensional space, $D = \{x_i\}_{i=1}^{n}$ and given a number of desired clusters K, the goal of representative-based clustering is to partition the dataset into 'K' groups of clusters is called as CLUSTERING.

Denoted by $C = \{C_1, C_2, \ldots, C_K\}$.

for each cluster C_i, there exists a representative point that summarizes the cluster (common choice is mean of all points in cluster / also called centroid " μ_i ")

- Intracluster distance → min
- Intercluster " → max

$$\mu_i = \frac{1}{n_i} \sum_{x_j \in C_i} x_j$$

Here $n_i = |C_i|$ (is the no. of points in cluster C_i).

Brute force Approach / exhaustive approach :-

- Generate all possible partitions of n points into 'k' clusters, evaluate some optimization score for each of them & retain the clustering that gives the best score.
- Infeasible because no. of clusterings of n points K^n is $O(K^n/K!)$
- eg. n = 3, K = 2 = $2^3/2 = 4$

3 < 4. ∴ $O(K^n/K!)$

Exact no. of ways of partitioning n points into k (non-empty, disjoint) parts is given by

stirling numbers of the second kind :-

$$S(n,k) = \frac{1}{K!} \sum_{t=0}^{k} (-1)^t \binom{K}{t} (K-t)^n$$

Informally, each point can be assigned to any of k clusters, thus K^n possible clusterings and any permutation of K clusters within a given clustering gives an equivalent clustering.

$\therefore O(K^n/K!)$ of 'n' points into 'k' groups.

Advantages :-

1) understanding large d/B.
2) summarize large data set.