USN											CMRIT

Internal Assessment Test 3 –July2021

Sub:	DataMining and DataWarehousing					Sub Code:	17CS651/ 18CS561	Branch:	CSE
Date:	28/07/2021	Duration:	90 min's	Max Marks:	50	Sem / Sec:	VI- 15,16,17 Scheme		OBE

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1	State and explain K-Means clustering algorithm K-means clustering algorithm **computes the centroids and iterates until we it finds optimal centroid**. It assumes that the number of clusters are already known. It is also called flat clustering algorithm. The number of clusters identified from data by algorithm is represented by 'K' in K-means. Kmeans Algorithm **Kmeans** algorithm is an iterative algorithm that tries to partition the dataset into *K*pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to **only one group**. It tries to make the intra-cluster data points as similar as possible while also keeping the clusters as different (far) as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid (arithmetic mean of all the data points that belong to that cluster) is at the minimum. The less variation we have within clusters, the more homogeneous (similar) the data points are within the same cluster. The way kmeans algorithm works is as follows: 1. Specify number of clusters *K*.	[10]	CO3	L3

2. Initialize centroids by first shuffling the dataset and then randomly selecting *K* data points for the centroids without replacement.

3. Keep iterating until there is no change to the centroids. i.e assignment of data points to clusters isn't changing.

- Compute the sum of the squared distance between data points and all centroids.
- Assign each data point to the closest cluster (centroid).
- Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster.

The approach kmeans follows to solve the problem is called **Expectation-Maximization**. The E-step is assigning the data points to the closest cluster. The M-step is computing the centroid of each cluster. Below is a break down of how we can solve it mathematically (feel free to skip it).

The objective function is:

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} \|x^i - \mu_k\|^2 \qquad (1)$$

where wik=1 for data point xi if it belongs to cluster *k*; otherwise, wik=0. Also, μk is the centroid of xi's cluster.

It's a minimization problem of two parts. We first minimize J w.r.t. wik and treat μk fixed. Then we minimize J w.r.t. μk and treat wik fixed. Technically speaking, we differentiate J w.r.t. wik first and update cluster assignments (*E-step*). Then we

differentiate J w.r.t. μk and recompute the centroids after the cluster assignments from previous step (*M-step*). Therefore, E-step is:

$$\frac{\partial J}{\partial w_{ik}} = \sum_{i=1}^{m} \sum_{k=1}^{K} \|x^i - \mu_k\|^2$$

$$\Rightarrow w_{ik} = \begin{cases} 1 & \text{if } k = argmin_j \|x^i - \mu_j\|^2 \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

In other words, assign the data point xi to the closest cluster judged by its sum of squared distance from cluster's centroid.

And M-step is:

$$\frac{\partial J}{\partial \mu_k} = 2 \sum_{i=1}^{m} w_{ik}(x^i - \mu_k) = 0$$

$$\Rightarrow \mu_k = \frac{\sum_{i=1}^{m} w_{ik} x^i}{\sum_{i=1}^{m} w_{ik}} \quad (3)$$

Which translates to recomputing the centroid of each cluster to reflect the new assignments.

Few things to note here:

- Since clustering algorithms including kmeans use distance-based measurements to determine the similarity between data points, it's recommended to standardize the data to have a mean of zero and a standard deviation of one since almost always the features in any dataset would have different units of measurements such as age vs income.
- Given kmeans iterative nature and the random initialization of centroids at the start of the algorithm, different initializations may lead to different clusters since kmeans algorithm may *stuck in a local optimum and may not converge to global optimum*. Therefore, it's recommended to run the

	algorithm using different initializations of centroids and pick the results of the run that that yielded the lower sum of squared distance. • Assignment of examples isn't changing is the same thing as no change in within-cluster variation: $$\frac{1}{m_k}\sum_{i=1}^{m_k}\|x^i - \mu_{c^k}\|^2 \qquad (4)$$			
2.	Explain OLAP server architecture: ROLAP,MOLAP,HOLAP	[10]	CO2	L2
3.	What are Bayesian Classifiers? Explain Bayes' theorem for classification. A Bayesian classifier is based on the idea that the role of a (natural) class is to predict the values of features for members of that class. ... A Bayesian classifier is a **probabilistic model** where the classification is a latent variable that is probabilistically related to the observed variables. Bayesian classification is based on **Bayes'** Theorem. Bayesian classifiers are the statistical classifiers. Bayesian classifiers can predict class membership probabilities such as the probability that a given tuple belongs to a particular class. Bayesian classification uses Bayes theorem **to predict the occurrence of any event**. ... P(Y/X) is a conditional probability that describes the occurrence of event Y is given that X is true. P(X) and P(Y) are the probabilities of observing X and Y independently of each other. This is known as the marginal probability. In numerous applications, the connection between the attribute set and the class variable is non- deterministic. In other words, we can say the class label of a test record cant be assumed with certainty even though its attribute set is the same as some of the training examples. These circumstances may emerge due to the noisy data or the presence of certain confusing factors that influence classification, but it is not included in the analysis. For example, consider the task of predicting the occurrence of whether an individual is at risk for liver illness based on individuals eating habits and working efficiency. Although most people who eat healthly and exercise consistently having less probability of occurrence of liver disease, they may still do so due to other factors. For example, due to consumption of the high-calorie street foods and alcohol abuse. Determining whether an individual's eating routine is healthy or the workout efficiency is sufficient is also subject to analysis, which in turn may introduce vulnerabilities into the leaning issue. Bayesian classification uses Bayes theorem to predict the occurrence of any event. Bayesian classifiers are the statistical classifiers with the Bayesian probability understandings. The theory expresses how a level of belief, expressed as a probability.	[10]	CO3	L3

Bayes theorem came into existence after Thomas Bayes, who first utilized conditional probability to provide an algorithm that uses evidence to calculate limits on an unknown parameter.

Bayes's theorem is expressed mathematically by the following equation that is given below.

$$P(X/Y) = \frac{P(Y/X)P(X)}{P(Y)}$$

Where X and Y are the events and P (Y) ≠ 0

P(X/Y) is a **conditional probability** that describes the occurrence of event **X** is given that **Y** is true.

P(Y/X) is a **conditional probability** that describes the occurrence of event **Y** is given that **X** is true.

P(X) and P(Y) are the probabilities of observing X and Y independently of each other. This is known as the **marginal probability**.

Bayesian interpretation:

In the Bayesian interpretation, probability determines a "**degree of belief**." Bayes theorem connects the degree of belief in a hypothesis before and after accounting for evidence. For example, Lets us consider an example of the coin. If we toss a coin, then we get either heads or tails, and the percent of occurrence of either heads and tails is 50%. If the coin is flipped numbers of times, and the outcomes are observed, the degree of belief may rise, fall, or remain the same depending on the outcomes.

For proposition X and evidence Y,

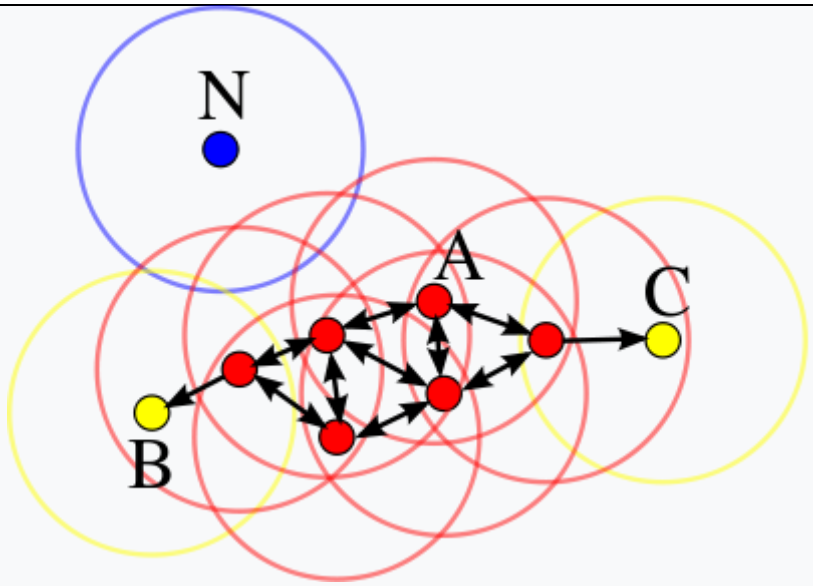
- P(X), the prior, is the primary degree of belief in X
- P(X/Y), the posterior is the degree of belief having accounted for Y.
- The quotient $\frac{P(Y/X)}{P(Y)}$ represents the supports Y provides for X.

Bayes theorem can be derived from the conditional probability:

$$P(X/Y) = \frac{P(X \cap Y)}{P(Y)}, \text{ if } P(Y) \neq 0$$

$$P(Y/X) = \frac{P(Y \cap X)}{P(X)}, \text{ if } P(X) \neq 0$$

	Where P (X∩Y) is the **joint probability** of both X and Y being true, because P (Y∩X) = P (X∩Y) or, P (X∩Y) = P(X/Y)P(Y) = P(Y/X)P(X) or, $P(X/Y) = \frac{P(Y/X)P(X)}{P(Y)}$, if P(Y) ≠ 0			
4	Explain various measures for selecting the best split in construction of Decision Tree.	[10]	CO3	L3
5	Explain DBSCAN algorithm with example. **Density-based spatial clustering of applications with noise** (**DBSCAN**) is a data clustering algorithm proposed by Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu in 1996.[1] It is a density-based clustering non-parametric algorithm: given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors), marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away). DBSCAN is one of the most common clustering algorithms and also most cited in scientific literature.[2] In 2014, the algorithm was awarded the test of time award (an award given to algorithms which have received substantial attention in theory and practice) at the leading data mining conference, ACM SIGKDD.[3] As of July 2020, the follow-up paper "DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN"[4] appears in the list of the 8 most downloaded articles of the prestigious ACM Transactions on Database Systems (TODS) journal.[5] Consider a set of points in some space to be clustered. Let ε be a parameter specifying the radius of a neighborhood with respect to some point. For the purpose of DBSCAN clustering, the points are classified as *core points*, (*density-*) *reachable points* and *outliers*, as follows: • A point p is a *core point* if at least minPts points are within distance ε of it (including p). • A point q is *directly reachable* from p if point q is within distance ε from core point p. Points are only said to be directly reachable from core points. • A point q is *reachable* from p if there is a path $p_1, \ldots, p_n$ with $p_1 = p$ and $p_n = q$, where each p_{i+1} is directly reachable from p_i. Note that this implies that the initial point and all points on the path must be core points, with the possible exception of q. • All points not reachable from any other point are *outliers* or *noise points*. Now if p is a core point, then it forms a *cluster* together with all points (core or non-core) that are reachable from it. Each cluster contains at least one core point; non-core points can be part of a cluster, but they form its "edge", since they cannot be used to reach more points.	[10]	CO3	L3



In this diagram, $\mathrm{minPts} = 4$. Point A and the other red points are core points, because the area surrounding these points in an ε radius contain at least 4 points (including the point itself). Because they are all reachable from one another, they form a single cluster. Points B and C are not core points, but are reachable from A (via other core points) and thus belong to the cluster as well. Point N is a noise point that is neither a core point nor directly-reachable.

Reachability is not a symmetric relation: by definition, only core points can reach non-core points. The opposite is not true, so a non-core point may be reachable, but nothing can be reached from it. Therefore, a further notion of *connectedness* is needed to formally define the extent of the clusters found by DBSCAN. Two points p and q are density-connected if there is a point o such that both p and q are reachable from o. Density-connectedness *is* symmetric.

A cluster then satisfies two properties:

1. All points within the cluster are mutually density-connected.
2. If a point is density-reachable from some point of the cluster, it is part of the cluster as well.

Algorithm[edit]

Original query-based algorithm[edit]

DBSCAN requires two parameters: ε (eps) and the minimum number of points required to form a dense region[a] (minPts). It starts with an arbitrary starting point that has not been visited. This point's ε-neighborhood is retrieved, and if it contains sufficiently many points, a cluster is started. Otherwise, the point is labeled as noise. Note that this point might later be found in a sufficiently sized ε-environment of a different point and hence be made part of a cluster.

If a point is found to be a dense part of a cluster, its ε-neighborhood is also part of that cluster. Hence, all points that are found within the ε-neighborhood are added, as is their own ε-neighborhood when they are also dense. This process continues until the density-connected cluster is completely found. Then, a new unvisited point is retrieved and processed, leading to the discovery of a further cluster or noise.

DBSCAN will also require a distance function[1][4] (as well as similarity functions or other predicates).[7] The distance function (dist) can therefore be seen as an additional parameter.

The algorithm can be expressed in pseudocode as follows:[4]

```
DBSCAN(DB, distFunc, eps, minPts) {
```

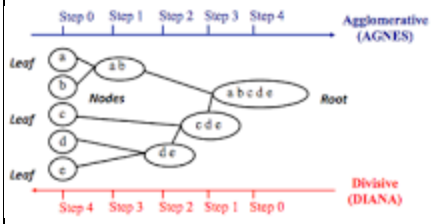
```
    C := 0                                                  /* Cluster
counter */
    for each point P in database DB {
        if label(P) ≠ undefined then continue               /* Previously
processed in inner loop */
        Neighbors N := RangeQuery(DB, distFunc, P, eps)     /* Find
neighbors */
        if |N| < minPts then {                              /* Density
check */
            label(P) := Noise                               /* Label as
Noise */
            continue
        }
        C := C + 1                                          /* next
cluster label */
        label(P) := C                                       /* Label
initial point */
        SeedSet S := N \ {P}                                /* Neighbors
to expand */
        for each point Q in S {                             /* Process
every seed point Q */
            if label(Q) = Noise then label(Q) := C          /* Change
Noise to border point */
            if label(Q) ≠ undefined then continue           /* Previously
processed (e.g., border point) */
            label(Q) := C                                   /* Label
neighbor */
            Neighbors N := RangeQuery(DB, distFunc, Q, eps) /* Find
neighbors */
            if |N| ≥ minPts then {                          /* Density
check (if Q is a core point) */
                S := S ∪ N                                  /* Add new
neighbors to seed set */
            }
        }
    }
}
```

where RangeQuery can be implemented using a database index for better performance, or using a slow linear scan:

```
RangeQuery(DB, distFunc, Q, eps) {
    Neighbors N := empty list
    for each point P in database DB {                       /* Scan all
points in the database */
        if distFunc(Q, P) ≤ eps then {                      /* Compute
distance and check epsilon */
            N := N ∪ {P}                                    /* Add to
result */
        }
    }
    return N
}
```

6	Explain the following: Agglomerative Hierarchical Clustering and Divisive hierarchical method	[10]	CO3	L3

Agglomerative Hierarchical Clustering (AHC) is a **clustering (or classification) method** which has the following advantages: It works from the dissimilarities between the objects to be grouped together. A type of dissimilarity can be suited to the subject studied and the nature of the data.



The agglomerative clustering is **the most common type of hierarchical clustering used to group objects in clusters based on their similarity**. It's also known as AGNES (Agglomerative Nesting). ... Next, pairs of clusters are successively merged until all clusters have been merged into one big cluster containing all objects.

Divisive Clustering:

The divisive clustering algorithm is a **top-down clustering approach**, initially, all the points in the dataset belong to one cluster and split is performed recursively as one moves down the hierarchy.

Divisive clustering

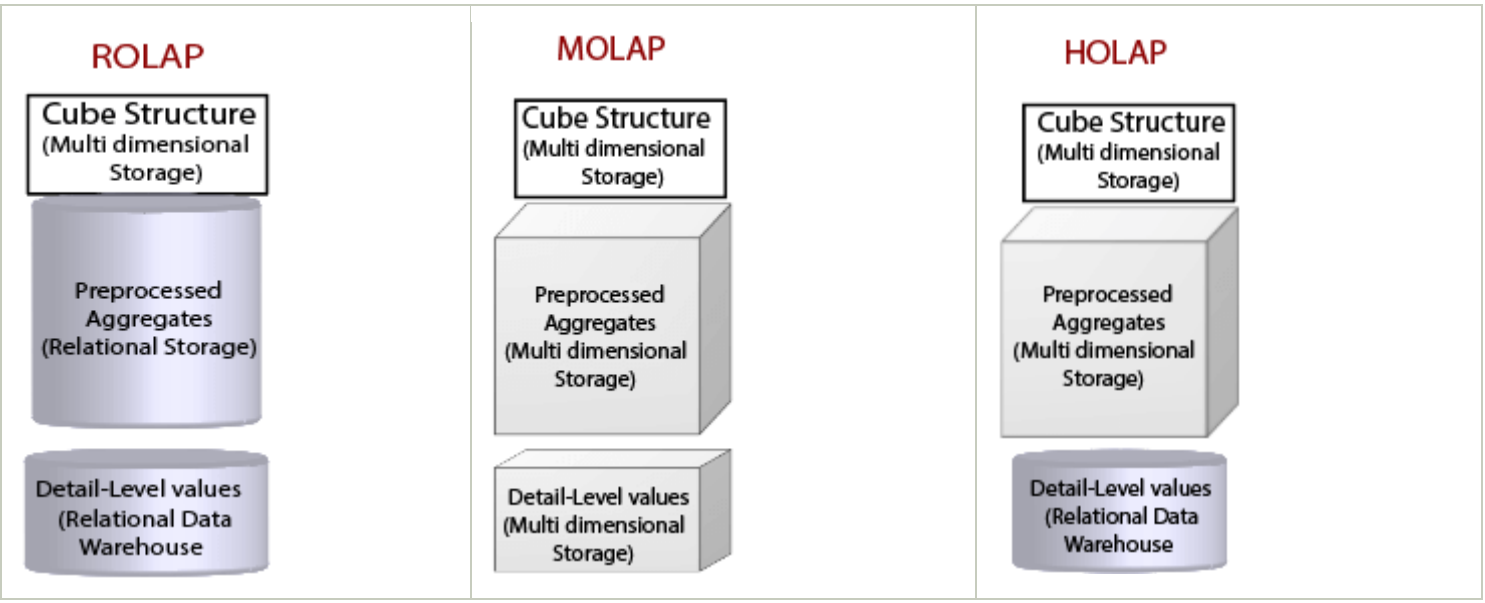
So far we have only looked at agglomerative clustering, but a cluster hierarchy can also be generated top-down. This variant of hierarchical clustering is called *top-down clustering* or *divisive clustering* . We start at the top with all documents in one cluster. The cluster is split using a flat clustering algorithm. This procedure is applied recursively until each document is in its own singleton cluster.

Top-down clustering is conceptually more complex than bottom-up clustering since we need a second, flat clustering algorithm as a ``subroutine". It has the advantage of being more efficient if we do not generate a complete hierarchy all the way down to individual document leaves. For a fixed number of top levels, using an efficient flat algorithm like K-means, top-down algorithms are linear in the number of documents and clusters. So they run much faster than HAC algorithms, which are at least quadratic.

There is evidence that divisive algorithms produce more accurate hierarchies than bottom-up algorithms in some circumstances. See the references on bisecting K-means in Section 17.9 . Bottom-up methods make clustering decisions based on local patterns without initially taking into account the global distribution. These early decisions cannot be undone. Top-down clustering benefits from complete information about the global distribution when making top-level partitioning decisions.

7	What is Cluster Analysis? Explain the different requirements of cluster analysis and issues in cluster evaluation. **Cluster analysis** or **clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar (in some sense) to each other than to those in other groups (clusters). It is a main task of exploratory data analysis, and a common technique for statistical data analysis, used in many fields, including pattern recognition, image analysis, information retrieval, bioinformatics, data compression, computer graphics and machine learning. Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their understanding of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with small distances between cluster members, dense areas of the data space, intervals or particular statistical distributions. Clustering can therefore be formulated as a multi-objective optimization problem. The appropriate clustering algorithm and parameter settings (including parameters such as the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and intended use of the results. Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimization that involves trial and failure. It is often necessary to modify data preprocessing and model parameters until the result achieves the desired properties. Besides the term *clustering*, there are a number of terms with similar meanings, including *automatic classification*, *numerical taxonomy*, *botryology* (from Greek βότρυς "grape"), *typological analysis*, and *community detection*. The subtle differences are often in the use of the results: while in data mining, the resulting groups are the matter of interest, in automatic classification the resulting discriminative power is of interest. **The main requirements that a clustering algorithm should satisfy are:** • scalability; • dealing with different types of attributes; • discovering clusters with arbitrary shape; • minimal requirements for domain knowledge to determine input parameters; • ability to deal with noise and outliers; **Current Challenges in Clustering** • Data Distribution. Large number of samples. The number of samples to be processed is very high. Algorithms have to be very conscious of scaling issues. ... • Application context. Legacy clusterings. Previous cluster analysis results are often available.	[10]	CO3	L2

ROLAP	MOLAP	HOLAP
ROLAP stands for Relational Online Analytical Processing.	MOLAP stands for Multidimensional Online Analytical Processing.	HOLAP stands for Hybrid Online Analytical Processing.
The ROLAP storage mode causes the aggregation of the division to be stored in indexed views in the relational database that was specified in the partition's data source.	The MOLAP storage mode principle the aggregations of the division and a copy of its source information to be saved in a multidimensional operation in analysis services when the separation is processed.	The HOLAP storage mode connects attributes of both MOLAP and ROLAP. Like MOLAP, HOLAP causes the aggregation of the division to be stored in a multidimensional operation in an SQL Server analysis services instance.
ROLAP does not because a copy of the source information to be stored in the Analysis services data folders. Instead, when the outcome cannot be derived from the query cache, the indexed views in the record source are accessed to answer queries.	This MOLAP operation is highly optimize to maximize query performance. The storage area can be on the computer where the partition is described or on another computer running Analysis services. Because a copy of the source information resides in the multidimensional operation, queries can be resolved without accessing the partition's source record.	HOLAP does not causes a copy of the source information to be stored. For queries that access the only summary record in the aggregations of a division, HOLAP is the equivalent of MOLAP.
Query response is frequently slower with ROLAP storage than with the MOLAP or HOLAP storage mode. Processing time is also frequently slower with ROLAP.	Query response times can be reduced substantially by using aggregations. The record in the partition's MOLAP operation is only as current as of the most recent processing of the separation.	Queries that access source record for example, if we want to drill down to an atomic cube cell for which there is no aggregation information must retrieve data from the relational database and will not be as fast as they would be if the source information were stored in the MOLAP architecture.


ROLAP
Cube Structure
(Multi dimensional Storage)
Preprocessed Aggregates (Relational Storage)
Detail-Level values (Relational Data Warehouse
MOLAP
Cube Structure
(Multi dimensional Storage)
Preprocessed Aggregates (Multi dimensional Storage)
Detail-Level values (Multi dimensional Storage)
HOLAP
Cube Structure
(Multi dimensional Storage)
Preprocessed Aggregates (Multi dimensional Storage)
Detail-Level values (Relational Data Warehouse