Table 13.1: Differences between Classification and Clustering

S.No.	Clustering	Classification
1.	Unsupervised learning and cluster formation are done by trial and error as there is no supervisor	Supervised learning with the presence of a supervisor to provide training and testing data
2.	Unlabelled data	Labelled data
3.	No prior knowledge in clustering	Knowledge of the domain is must to label the samples of the dataset
4.	Cluster results are dynamic	Once a label is assigned, it does not change

Applications of Clustering

1. Grouping based on customer buying patterns
2. Profiling of customers based on lifestyle
3. In information retrieval applications (like retrieval of a document from a collection of documents)
4. Identifying the groups of genes that influence a disease
5. Identification of organs that are similar in physiology functions
6. Taxonomy of animals, plants in Biology
7. Clustering based on purchasing behaviour and demography
8. Document indexing
9. Data compression by grouping similar objects and finding duplicate objects

Challenges of Clustering Algorithms

A huge collection of data with higher dimensions (i.e., features or attributes) can pose a problem for clustering algorithms. With the arrival of the internet, billions of data are available for clustering algorithms. This is a difficult task, as scaling is always an issue with clustering algorithms. Scaling is an issue where some algorithms work with lower dimension data but do not perform well for higher dimension data. Also, units of data can post a problem, like some weights in kg and some in pounds can pose a problem in clustering. Designing a proximity measure is also a big challenge.

The advantages and disadvantages of the cluster analysis algorithms are given in Table 13.2.

Table 13.2: Advantages and Disadvantages of Clustering Algorithms

S.No.	Advantages	Disadvantages
1.	Cluster analysis algorithms can handle missing data and outliers.	Cluster analysis algorithms are sensitive to initialization and order of the input data.
2.	Can help classifiers in labelling the unlabelled data. Semi-supervised algorithms use cluster analysis algorithms to label the unlabelled data and then use classifiers to classify them.	Often, the number of clusters present in the data have to be specified by the user.

(Continued)

Scan for 'Additional Examples'

13.4 PARTITIONAL CLUSTERING ALGORITHM

'*k*-means' algorithm is a straightforward iterative partitional algorithm. Here, *k* stands for the user specified requested clusters as users are not aware of the clusters that are present in the dataset. The *k*-means algorithm assumes that the clusters do not overlap. Therefore, a sample or data point can belong to only one cluster in the end. Also, this algorithm can detect clusters of shapes like circular or spherical.

Initially, the algorithm needs to be initialized. The algorithm can select *k* data points randomly or use the prior knowledge of the data. In most cases, in *k*-means algorithm setup, prior knowledge is absent. The composition of the cluster is based on the initial condition, therefore, initialization is an important task. The sample or data points need to be normalized for better performance. The concepts of normalization are covered in Chapter 3.

The core process of the *k*-mean algorithm is assigning a sample to a cluster, that is, assigning each sample or data point to the *k* cluster centers based on its distance and the centroid of the clusters. This distance should be minimum. As a new sample is added, new computation of mean vectors of the points for that cluster to which sample is assigned is required. Therefore, this iterative process is continued until no change of instances to clusters is noticed. This algorithm then terminates and the termination is guaranteed.

Algorithm 13.3: *k*-means Algorithm

Step 1: Determine the number of clusters before the algorithm is started. This is called *k*.

Step 2: Choose *k* instances randomly. These are initial cluster centers.

Step 3: Compute the mean of the initial clusters and assign the remaining sample to the closest cluster based on Euclidean distance or any other distance measure between the instances and the centroid of the clusters.

Step 4: Compute new centroid again considering the newly added samples.

Step 5: Perform the steps 3–4 till the algorithm becomes stable with no more changes in assignment of instances and clusters.

k-means can also be viewed as greedy algorithm as it involves partitioning *n* samples to *k* clusters to minimize Sum of Squared Error (SSE). SSE is a metric that is a measure of error that gives the sum of the squared Euclidean distances of each data to its closest centroid. It is given as:

$$SSE = \sum_{i=1}^{k} dist(c_i, x)^2 \tag{13.14}$$

Here, c_i is the centroid of the i^{th} cluster, x is the sample or data point and dist is the Euclidean distance. The aim of the *k*-means algorithm is to minimize SSE.

Advantages

1. Simple
2. Easy to implement

Disadvantages

1. It is sensitive to initialization process as change of initial points leads to different clusters.
2. If the samples are large, then the algorithm takes a lot of time.

How to Choose the Value of k?

It is obvious that k is the user specified value specifying the number of clusters that are present. Obviously, there are no standard rules available to pick the value of k. Normally, the k-means algorithm is run with multiple values of k and within group variance (sum of squares of samples with its centroid) and plotted as a line graph. This plot is called Elbow curve. The optimal or best value of k can be determined from the graph. The optimal value of k is identified by the flat or horizontal part of the Elbow curve.

Complexity

The complexity of k-means algorithm is dependent on the parameters like n, the number of samples, k, the number of clusters, $\Theta(nkId)$. I is the number of iterations and d is the number of attributes. The complexity of k-means algorithm is $O(n^2)$.

Example 13.5: Consider the following set of data given in Table 13.9. Cluster it using k-means algorithm with the initial value of objects 2 and 5 with the coordinate values (4, 6) and (12, 4) as initial seeds.

Table 13.9: Sample Data

Objects	X-coordinate	Y-coordinate
1	2	4
2	4	6
3	6	8
4	10	4
5	12	4

Solution: As per the problem, choose the objects 2 and 5 with the coordinate values. Hereafter, the objects' id is not important. The samples or data points (4, 6) and (12, 4) are started as two clusters as shown in Table 13.10.

Initially, centroid and data points are same as only one sample is involved.

Table 13.10: Initial Cluster Table

Cluster 1	Cluster 2
(4, 6)	(12, 4)
Centroid 1 (4, 6)	Centroid 2 (12, 4)

Iteration 1: Compare all the data points or samples with the centroid and assign to the nearest sample. Take the sample object 1 (2, 4) from Table 13.9 and compare with the centroid of

the clusters in Table 13.10. The distance is 0. Therefore, it remains in the same cluster. Similarly, consider the remaining samples. For the object 1 (2, 4), the Euclidean distance between it and the centroid is given as:

Dist (1, centroid 1) = $\sqrt{(2-4)^2 + (4-6)^2} = \sqrt{8}$

Dist (1, centroid 2) = $\sqrt{(2-12)^2 + (4-4)^2} = \sqrt{100} = 10$

Object 1 is closer to the centroid of cluster 1 and hence assign it to cluster 1. This is shown in Table 13.11. Object 2 is taken as centroid point.

For the object 3 (6, 8), the Euclidean distance between it and the centroid points is given as:

Dist (3, centroid 1) = $\sqrt{(6-4)^2 + (8-6)^2} = \sqrt{8}$

Dist (3, centroid 2) = $\sqrt{(6-12)^2 + (8-4)^2} = \sqrt{52}$

Object 3 is closer to the centroid of cluster 1 and hence remains in the same cluster 1.

Proceed with the next point object 4(10, 4) and again compare it with the centroids in Table 13.10.

Dist (4, centroid 1) = $\sqrt{(10-4)^2 + (4-6)^2} = \sqrt{40}$

Dist (4, centroid 2) = $\sqrt{(10-12)^2 + (4-4)^2} = \sqrt{4} = 2$

Object 4 is closer to the centroid of cluster 2 and hence assign it to the cluster table. Object 4 is in the same cluster. The final cluster table is shown in Table 13.11.

Obviously, Object 5 is in Cluster 3. Recompute the new centroids of cluster 1 and cluster 2. They are (4, 6) and (11, 4), respectively.

Table 13.11: Cluster Table After Iteration 1

Cluster 1	Cluster 2
(4, 6)	(10, 4)
(2, 4)	(12, 4)
(6, 8)	
Centroid 1 (4, 6)	Centroid 2 (11, 4)

The second iteration is started again with the Table 13.11.

Obviously, the point (4, 6) remains in cluster 1, as the distance of it with itself is 0. The remaining objects can be checked. Take the sample object 1 (2, 4) and compare with the centroid of the clusters in Table 13.12.

Dist (1, centroid 1) = $\sqrt{(2-4)^2 + (4-6)^2} = \sqrt{8}$

Dist (1, centroid 2) = $\sqrt{(2-11)^2 + (4-4)^2} = \sqrt{81} = 9$

Object 1 is closer to centroid of cluster 1 and hence remains in the same cluster. Take the sample object 3 (6, 8) and compare with the centroid values of clusters 1 (4, 6) and cluster 2 (11, 4) of the Table 13.12.

Dist (3, centroid 1) = $\sqrt{(6-4)^2 + (8-6)^2} = \sqrt{8}$

Dist (3, centroid 2) = $\sqrt{(6-11)^2 + (8-4)^2} = \sqrt{41}$

Object 3 is closer to centroid of cluster 1 and hence remains in the same cluster. Take the sample object 4 (10, 4) and compare with the centroid values of clusters 1 (4, 6) and cluster 2 (11, 4) of the Table 13.12:

Dist (4, centroid 1) = $\sqrt{(10-4)^2 + (4-6)^2} = \sqrt{40}$

Dist (3, centroid 2) = $\sqrt{(10-11)^2 + (4-4)^2} = \sqrt{1} = 1$

Object 3 is closer to centroid of cluster 2 and hence remains in the same cluster. Obviously, the sample (12, 4) is closer to its centroid as shown below:

Dist (5, centroid 1) = $\sqrt{(12-4)^2 + (4-6)^2} = \sqrt{68}$

Dist (5, centroid 2) = $\sqrt{(12-11)^2 + (4-4)^2} = \sqrt{1} = 1$. Therefore, it remains in the same cluster. Object 5 is taken as centroid point.

The final cluster Table 13.12 is given below:

Table 13.12: Cluster Table After Iteration 2

Cluster 1	Cluster 2
(4, 6)	(10, 4)
(2, 4)	(12, 4)
(6, 8)	
Centroid (4, 6)	Centroid (11, 4)

There is no change in the cluster Table 13.12. It is exactly the same; therefore, the *k*-means algorithm terminates with two clusters with data points as shown in the Table 13.12.

Scan for *'Additional Examples'*