UNIT-4

CLUSTER ANALYSIS

Classification by Back propagation:

- Back propagation is a neural network learning algorithm.
- A neural network is a set of connected input/output units in which each connection has a weight associated with it.
- During the learning phase, the network learns by adjusting the weights so as to be able to predict the correct class label of the input tuples.
- Neural network learning is also referred to as connectionist learning due to the connections between units.
- Neural networks involve long training times and are therefore more suitable for applications where this is feasible.

- Back propagation learns by iteratively processing a data set of training tuples, comparing the network's prediction for each tuple with the actual known target value.
- The target value may be the known class label of the training tuple (for classification problems) or a continuous value (for prediction).

- For each training tuple, the weights are modified so as to minimize the mean squared error between the network's prediction and the actual target value. These modifications are made in the ―backwards‖ direction, that is, from the output layer, through each hidden layer down to the first hidden layer hence the name is back propagation.
- Although it is not guaranteed, in general the weights will eventually converge, and the learning process stops.

Advantages:

1. It include their high tolerance of noisy data as well as their ability to classify patterns on which they have not been trained.
2. They can be used when you may have little knowledge of the relationships between attributes and classes.
3. They are well-suited for continuous-valued inputs and outputs, unlike most decision tree algorithms.
4. They have been successful on a wide array of real-world data, including handwritten character recognition, pathology and laboratory medicine, and training a computer to pronounce English text.
5. Neural network algorithms are inherently parallel; parallelization techniques can be used to speed up the computation process.

Process:

Initialize the weights:

The weights in the network are initialized to small random numbers

ranging from-1.0 to 1.0, or -0.5 to 0.5. Each unit has a *bias* associated with it. The biases are similarly initialized to small random numbers.

Each training tuple, ***X***, is processed by the following steps.

Propagate the inputs forward:

First, the training tuple is fed to the input layer of the network. The inputs pass through the input units, unchanged. That is, for an input unit *j*, its output, *Oj*, is equal to its input value, *Ij*.

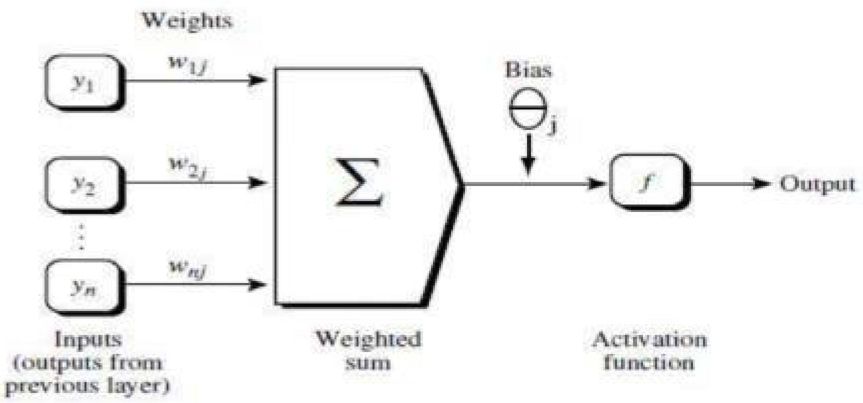
Next, the net input and output of each unit in the hidden and output layers are computed. The net input to a unit in the hidden or output layers is computed as a linear combination of its inputs.

Each such unit has a number of inputs to it that are, in fact, the outputs of the units connected to it in the previous layer.

Each connection has a weight. To compute the net input to the unit, each input connected to the unit is multiplied by its corresponding weight, and this is summed.

$$I_j = \sum_i w_{ij} O_i + \theta_j,$$

Where w I ,j is the weight of the connection from unit I in the previous layer to unit j; O i is the output of unit I from the previous layer ϴ j is the bias of the unit & it acts as a threshold in that it serves to vary the activity of the unit.
Each unit in the hidden and output layers takes its net input and then applies an activation function to it.



Back propagate the error:

The error is propagated backward by updating the weights and biases to reflect the error of the network's prediction. For a unit *j* in the output layer, the error *Err j* is computed by

$$Err_j = O_j(1-O_j)(T_j-O_j)$$

Where O j is the actual output of unit j, and T j is the known target value of the given training tuple.

The error of a hidden layer unit j is

$$Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$$

Where wjk is the weight of the connection from unit j to a unit k in the next higher layer, and Err k is the error of unit k.

Weights are updated by the following equations, where D *w I j* is the change in weight *w I j*:

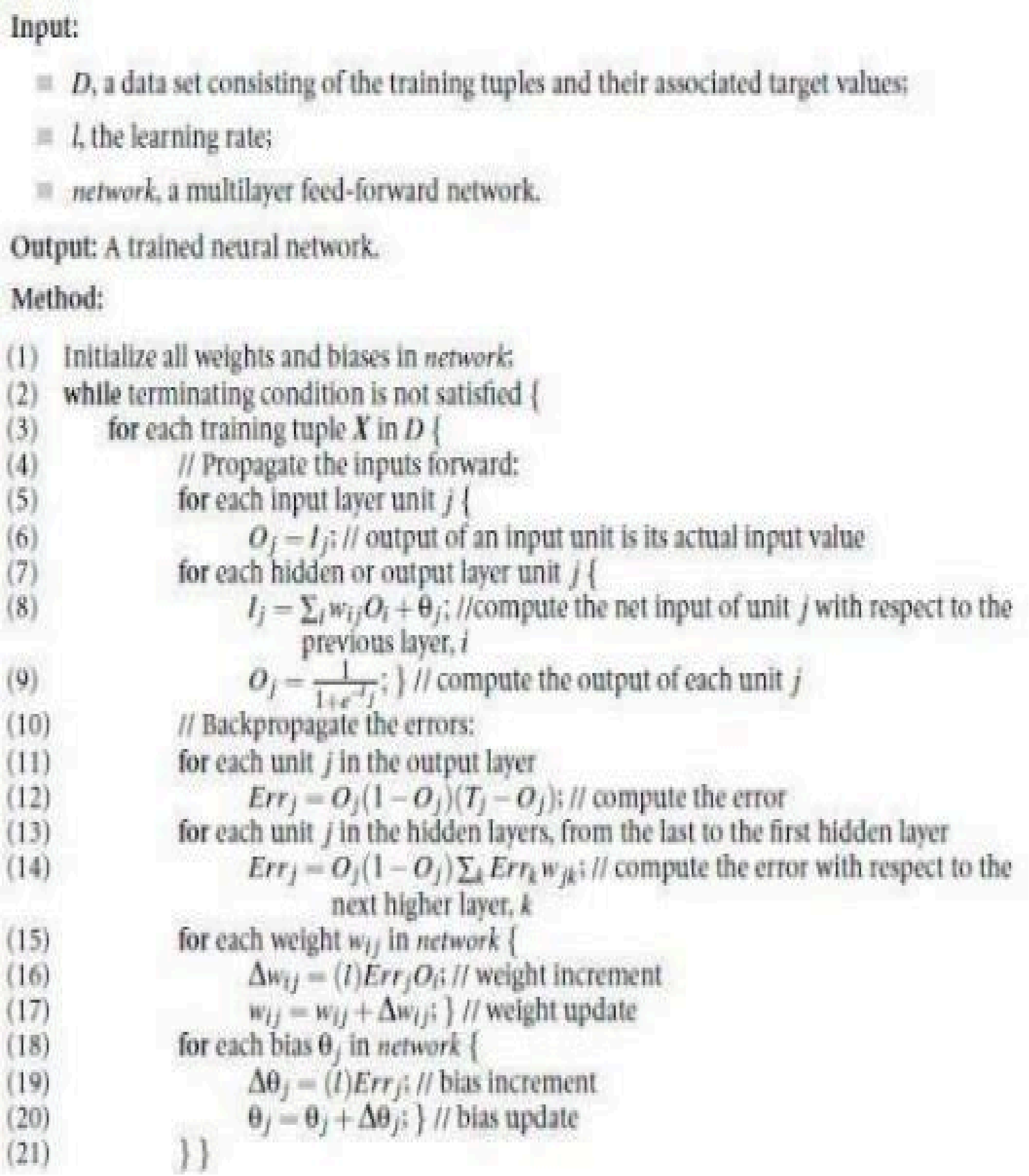
$$\Delta w_{ij} = (l)Err_j O_i$$
$$w_{ij} = w_{ij} + \Delta w_{ij}$$

Biases are updated by the following equations below

$$\Delta\theta_j = (l)Err_j$$

$$\theta_j = \theta_j + \Delta\theta_j$$

Algorithm:



Input:

- D, a data set consisting of the training tuples and their associated target values;
- l, the learning rate;
- *network*, a multilayer feed-forward network.

Output: A trained neural network.

Method:

```
Initialize all weights and biases in network;
while terminating condition is not satisfied {
    for each training tuple X in D {
        // Propagate the inputs forward:
        for each input layer unit j {
            $O_j = I_j$; // output of an input unit is its actual input value
        for each hidden or output layer unit j {
            $I_j = \sum_i w_{ij} O_i + \theta_j$; //compute the net input of unit j with respect to the
                previous layer, i
            $O_j = \frac{1}{1+e^{-I_j}}$; } // compute the output of each unit j
        // Backpropagate the errors:
        for each unit j in the output layer
            $Err_j = O_j(1-O_j)(T_j-O_j)$; // compute the error
        for each unit j in the hidden layers, from the last to the first hidden layer
            $Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$; // compute the error with respect to the
                next higher layer, k
        for each weight $w_{ij}$ in network {
            $\Delta w_{ij} = (l) Err_j O_i$; // weight increment
            $w_{ij} = w_{ij} + \Delta w_{ij}$; } // weight update
        for each bias $\theta_j$ in network {
            $\Delta \theta_j = (l) Err_j$; // bias increment
            $\theta_j = \theta_j + \Delta \theta_j$; } // bias update
        } }
```

k-Nearest-Neighbor Classifier:

- Nearest-neighbor classifiers are based on learning by analogy, that is, by comparing a given test tuple with training tuples that are similar to it.
- The training tuples are described by *n* attributes. Each tuple represents a point in an *n*-dimensional space. In this way, all of the training tuples are stored in an *n*-dimensional pattern space. When given an unknown tuple, a ***k***-nearest-neighbor classifier searches the pattern space for the *k* training tuples that are closest to

theunknown tuple. These k training tuples are the k nearest neighbors of the unknown tuple.

Closeness is defined in terms of a distance metric, such as Euclidean distance.

- The Euclidean distance between two points or tuples, say, $\boldsymbol{X}1 = (x11, x12, \ldots , x1n)$
- and $\boldsymbol{X}2 = (x21, x22, \ldots ,x2n)$, is

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}.$$

In other words, for each numeric attribute, we take the difference between the corresponding

values of that attribute in tuple ***X*1**and in tuple ***X*2**, square this difference, and accumulate it. The square root is taken of the total accumulated distance count.
Min-Max normalization can be used to transform a value v of a numeric attribute A to $v0$ in the range [0, 1] by computing

$$v' = \frac{v - min_A}{max_A - min_A},$$

Where $minA$ and $maxA$ are the minimum and maximum values of attribute A

- For k-nearest-neighbor classification, the unknown tuple is assigned the most common class among its k nearest neighbors.
- When $k = 1$, the unknown tuple is assigned the class of the training tuple that is closest to it in pattern space.
- Nearest neighbor classifiers can also be used for prediction, that is, to return a real-valued prediction for a given unknown tuple.
- In this case, the classifier returns the average value of the real-valued labels associated with the k nearest neighbors of the unknown tuple.

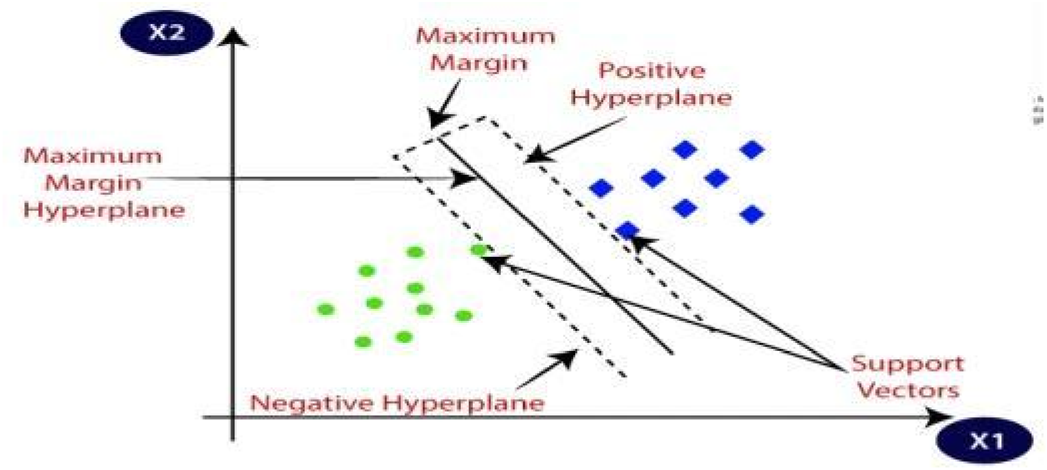
Support Vector Machines

support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector

Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



Example: SVM can be understood with the example that we have used in the KNN classifier. Suppose we see a strange cat that also has some features of dogs, so if we want a model that can accurately identify whether it is a cat or dog, so such a model can be created by using the SVM algorithm. We will first train our model with lots of images of cats and dogs so that it can learn about different features of cats and dogs, and then we test it with this strange creature. So as support vector creates a decision boundary between these two data (cat and dog) and choose extreme cases (support vectors), it will see the extreme case of cat and dog. On the basis of the support vectors, it will classify it as a cat. Consider the below diagram:

SVM algorithm can be used for **Face detection, image classification, text categorization,** etc.

Types of SVM

SVM can be of two types:

- **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Hyperplane and Support Vectors in the SVM algorithm:

Hyperplane: There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

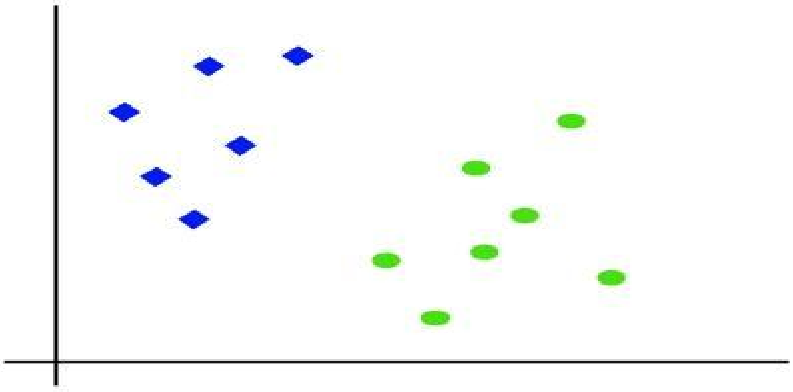
The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.

We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

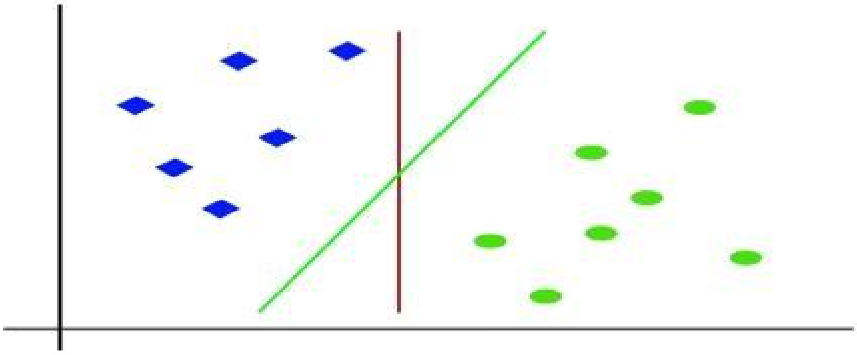
Support Vectors:

The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

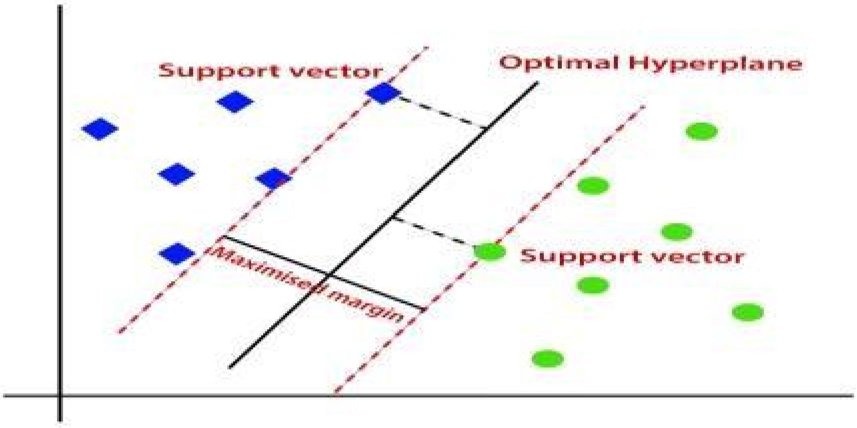
The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue. Consider the below image:



So as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes. Consider the below image:



Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a **hyperplane**. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called as **margin**. And the goal of SVM is to maximize this margin. The **hyperplane** with maximum margin is called the **optimal hyperplane**.

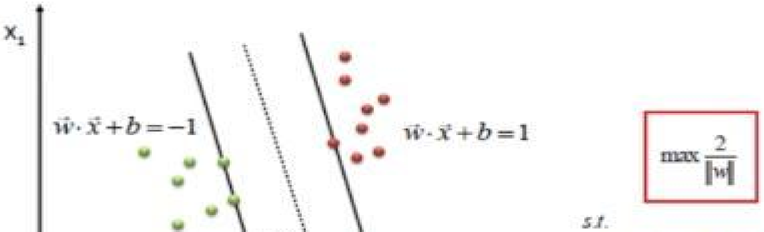


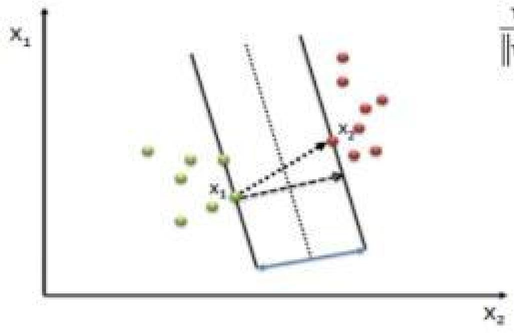
Algorithm:

1. Define an optimal hyperplane: maximize margin

2. Extend the above definition for non-linearly separable problems: have a penalty term for misclassifications.

3. Map data to high dimensional space where it is easier to classify with linear decision surfaces: reformulate problem so that data is mapped implicitly to this space.
To define an optimal hyperplane we need to maximize the width of the margin (w).





$$\frac{w}{\|w\|}\cdot(x_2-x_1)=\text{width}=\frac{2}{\|w\|}$$

$$w\cdot x_2+b=1$$

$$w\cdot x_1+b=-1$$

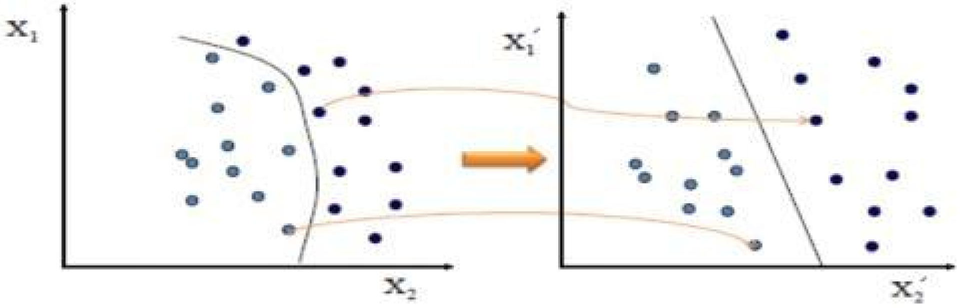
$$w\cdot x_2+b-w\cdot x_1-b=1-(-1)$$

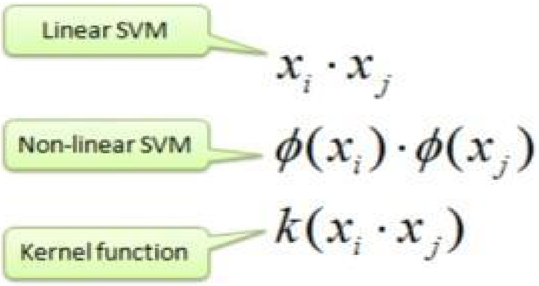
$$w\cdot x_2-w\cdot x_1=2$$

$$\frac{w}{\|w\|}(x_2-x_1)=\frac{2}{\|w\|}$$

SVM handles this by using a kernel function (nonlinear) to map the data into a different space where a hyperplane (linear) cannot be used to do the separation. It means a non-linear function is learned by a linear learning machine in a high-dimensional feature space while the capacity of the system is controlled by a parameter that does not depend on the dimensionality of the space.

This is called kernel trick which means the kernel function transform the data into a higher dimensional feature space to make it possible to perform the linear separation.





Map data into new space, then take the inner product of the new vectors. The image of the inner product of the data is the inner product of the images of the data. Two kernel functions are shown below.

Polynomial

$$k(\mathrm{x}_i, \mathrm{x}_j) = (\mathrm{x}_i . \mathrm{x}_j)^d$$

Gaussian Radial Basis function

$$k(\mathrm{x}_i, \mathrm{x}_j) = \exp\left(- \frac{\left\| \mathrm{x}_i - \mathrm{x}_j \right\|^2}{2\sigma^2} \right)$$

PREDICTION

When an input is provided, its ordered values can be predicted. This activity is called Numeric Prediction. A numeric prediction can be performed using the ‘Regression Technique’.

The Regression Technique is used to determine the relationship between any number of predictor variables and a single response variable.

The values of the predictor variable are already known and they represent a tuple, whereas, the values of the response variable is to be predicted. Regression analysis can be performed by using either of the following techniques.

- Linear regression
- Non-linear regression

i).Linear Regression

- Linear regression: involves a response variable y and a single predictor variable x.

  y = w0 + w1x

  where w0 (y-intercept) and w1 (slope) are regression coefficients

- Method of least squares: estimates the best-fitting straight line

- Multiple linear regression: involves more than one predictor variable
- Training data is of the form (X1, y1), (X2, y2),…, (X|D|, y|D|)
- Ex. For 2-D data, we may have: y = w0 + w1 x1+ w2 x2
- Solvable by extension of least square method or using SAS, S-Plus
- Many nonlinear functions can be transformed into the above.

ii).Nonlinear Regression

- Some nonlinear models can be modeled by a polynomial function
- A polynomial regression model can be transformed into linear regression model.
  - For example,
    - y = w0 + w1 x + w2 x2 + w3 x3
    - convertible to linear with new variables: x2 = x2, x3= x3
    - y = w0 + w1 x + w2 x2 + w3 x3
- Other functions, such as power function, can also be transformed to linear model
- Some models are intractable nonlinear (e.g., sum of exponential terms)
- Possible to obtain least square estimates through extensive calculation on more complex formulae.

CLUSTER ANALYSIS

Cluster is a group of objects that belongs to the same class. In other words, similar objects are grouped in one cluster and dissimilar objects are grouped in another cluster.

Definition: Clustering is the process of making a group of abstract objects into classes of similar objects.

- A cluster of data objects can be treated as one group.
- While doing cluster analysis, we first partition the set of data into groups based on data similarity and then assign the labels to the groups.
- The main advantage of clustering over classification is that, it is adaptable to changes and helps single out useful features that distinguish different groups.

Applications of Cluster Analysis

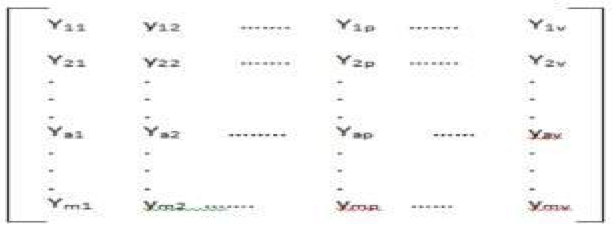
- Clustering analysis is broadly used in many applications such as market research, pattern recognition, data analysis, and image processing.
- Clustering can also help marketers discover distinct groups in their customer base. And they can characterize their customer groups based on the purchasing patterns.

- In the field of biology, it can be used to derive plant and animal taxonomies, categorize genes with similar functionalities and gain insight into structures inherent to populations.
- Clustering also helps in identification of areas of similar land use in an earth observation database. It also helps in the identification of groups of houses in a city according to house type, value, and geographic location.
- Clustering also helps in classifying documents on the web for information discovery.
- Clustering is also used in outlier detection applications such as detection of credit card fraud.
- As a data mining function, cluster analysis serves as a tool to gain insight into the distribution of data to observe characteristics of each cluster.

TYPES OF DATA IN CLUSTER ANALYSIS

The memory based clustering algorithm uses two kinds of data patterns to cluster the data objects. They are,

i)Two mode Data Matrix: Data Matrix also called ‘Object-by-variable structure’ when m objects with v variables are represented in the form of a relational table ie m-by-v matrix. The objects may be persons and the variables may be height, age, gender and weight. It is also called two mode matrix since rows and columns of the data matrix represent different entities as shown in below.



$$\begin{bmatrix} Y_{11} & Y_{12} & \text{-------} & Y_{1p} & \text{-------} & Y_{1v} \\ Y_{21} & Y_{22} & \text{-------} & Y_{2p} & \text{-------} & Y_{2v} \\ - & - & & - & & - \\ - & - & & - & & - \\ - & - & & - & & - \\ Y_{a1} & Y_{a2} & \text{--------} & Y_{ap} & \text{------} & Y_{av} \\ - & - & & - & & - \\ - & - & & - & & - \\ - & - & & - & & - \\ Y_{m1} & Y_{m2} & \text{--------} & Y_{mp} & \text{------} & Y_{mv} \end{bmatrix}$$

ii). One mode Dissimilarity Matrix: The Dissimilarity Matrix also called ‘Object-by-Object Structure’ which is represented as m- by-m matrix. It preserves a group of proximities that are helpful for all pairs of (m) objects. The entities denoted by the rows and columns are alike and hence it is also known as one-mode matrix. One mode matrix is operated by several clustering algorithms. The representation of the matrix is as shown below.

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ d(4,1) & d(4,2) & d(4,3) & 0 & \\ \cdot & \cdot & \cdot & \cdot & \\ \cdot & \cdot & \cdot & \cdot & \\ \cdot & \cdot & \cdot & \cdot & \\ d(m,1) & d(m,2) & \ldots\ldots & & \end{bmatrix}$$

Where

i). d(a,b) denotes the measured difference between objects a and b.

ii). This is a positive number which is close to 0 if the objects a and b are similar to each other and this number becomes larger when the objects differ.

iii). d(a,b) = d(b,a) and d(a,a) =0.

The object dissimilarity computed for objects is described by various types of data variables. They are:

- Interval-scaled variables:

- Binary variables:

- Categorical, Ordinal, and Ratio – Scaled variables:
- Mixed type Variables:

a). Interval-Scaled variables: Interval-Scaled variables or Numeric variables are continues measurements of a roughly linear scale. Longitude and Latitude coordinates, weather temperature, Height and weight are the examples of Interval-Scaled variables.

Dissimilarity between Object of Interval-Scaled variables: The most frequently used distance measures to determine the existence of dissimilarity between the data objects represented by interval-scaled variables are,

Euclidean Distance: Euclidean Distance measure is expressed in terms of two n-dimensional data objects given as,

a=(ya1,ya2,ya3, ……..yan) and b=(yb1,yb2,yb3,……..ybn)
the formula is,

$$d(a,b) = \sqrt{(y_{a1}-y_{b1})^2+(y_{a2}-y_{b2})^2+ \ldots\ldots\ldots\ldots \ldots\ldots\ldots +(y_{an}-y_{bn})^2}$$

b). Binary variables:A Binary variable is defined in terms of two states ie; 0 or 1. If the variable is present it is denoted by '1' and if it absent is dented by '0'.

Example: Let 'operation' be a variable describing a doctor's action, where the binary state '1' indicates that the operation done by the doctor is successful and '0' indicates that the operation done by the doctor is failed.

c).Categorical, Ordinal, and Ratio – Scaled variables:

Categorical Variables:A categorical variable is a generation of the binary variable in that it can take on more than two states. For examples, map-color is a categorical variable that may have, say five states Red, Yellow, Green, Pink and Blue.

Ordinal Variables: Ordinal Variables are very useful for registering subjective assessments of qualities that can't be measured objectively. For example, professional ranks are often enumerated in a sequential order, such as assistant, associate and full for professors.

There are two types of ordinal variables, they are:

- Continuous ordinal variables
- Discrete Ordinal Variables

Ratio – Scaled variables:A Ratio – Scaled variables makes a positive measurement on a linear scale, such as an exponential scale, approximately following the formula,

$$Ae^{Bt} \text{ or } Ae^{-Bt}$$

Where, A and B are positive Constants. 't' typically represents time.
Common examples include the growth of bacteria population or decay of a radioactive element.

d).Mixed type Variables:A data base may consists of objects described by various kinds of variables such as interval scaled , categorical, ratio scaled, ordinal, symmetrical binary and asymmetrical binary. That is, objects are mixture of various variables.

A CATEGORIZATION OF MAJOR CLUSTERING METHODS

A Categorization of Major Clustering Methods:

1. Partitioning algorithms: Construct various partitions and then evaluate them by some criterion

2. Hierarchy algorithms: Create a hierarchical decomposition of the set of data (or objects) using some criterion

3. Density-based: based on connectivity and density functions

4. Grid-based: based on a multiple-level granularity structure
5. Model-based: A model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other.

i).Partitioning Method: Suppose we are given a database of ‘n’ objects and the partitioning method constructs ‘k’ partition of data. Each partition will represent a cluster and k ≤ n. It means that it will classify the data into k groups, which satisfy the following requirements −

- Each group contains at least one object.
- Each object must belong to exactly one group.

For a given number of partitions (say k), the partitioning method will create an initial partitioning. Then it uses the iterative relocation technique to improve the partitioning by moving objects from one group to other.

ii).Hierarchical Methods: This method creates a hierarchical decomposition of the given set of data objects. We can classify hierarchical methods on the basis of how the hierarchical decomposition is formed. There are two approaches here −

- Agglomerative Approach
- Divisive Approach

Agglomerative Approach: This approach is also known as the bottom-up approach. In this, we start with each object forming a separate group. It keeps on merging the objects or groups that are close to one another. It keep on doing so until all of the groups are merged into one or until the termination condition holds.
Divisive Approach: This approach is also known as the top-down approach. In this, we start with all of the objects in the same cluster. In the continuous iteration, a cluster is split up into smaller clusters. It is down until each object in one cluster or the termination condition holds. This method is rigid, i.e., once a merging or splitting is done, it can never be undone.

iii).Density-based Method: This method is based on the notion of density. The basic idea is to continue growing the given cluster as long as the density in the neighbourhood exceeds some threshold, i.e., for each data point within a given cluster, the radius of a given cluster has to contain at least a minimum number of points.

iv).Grid-based Method: In this, the objects together form a grid. The object space is quantized into finite number of cells that form a grid structure.

Advantages

- The major advantage of this method is fast processing time.

- It is dependent only on the number of cells in each dimension in the quantized space.

v).Model-based methods: In this method, a model is hypothesized for each cluster to find the best fit of data for a given model. This method locates the clusters by clustering the density function. It reflects spatial distribution of the data points.

This method also provides a way to automatically determine the number of clusters based on standard statistics, taking outlier or noise into account. It therefore yields robust clustering methods.

Apart from Clustering methods, there are two types of Clustering tasks that require special attention. They are:

a).Clustering High Dimensional Data: Clustering High Dimensional Data is of crucial important because in many advanced applications data objects such as text documents and micro array data are high dimensional in nature.
b). Constraint-based Method: In this method, the clustering is performed by the incorporation of user or application-oriented constraints. A constraint refers to the user expectation or the properties of desired clustering results. Constraints provide us with an interactive way of communication with the clustering process. Constraints can be specified by the user or the application requirement.

PARTITIONING METHOD

Suppose we are given a database of ‘n’ objects and the partitioning method constructs ‘k’ partition of data. Each partition will represent a cluster and k ≤ n. It means that it will classify the data into k groups, which satisfy the following requirements −

- Each group contains at least one object.
- Each object must belong to exactly one group.

Points to remember −

- For a given number of partitions (say k), the partitioning method will create an initial partitioning.
- Then it uses the iterative relocation technique to improve the partitioning by moving objects from one group to other.

THE *K*-MEANS CLUSTERING ALGORITHM

The *k*-means clustering algorithm is used to cluster observations into groups of related observations without any prior knowledge of those relationships. By sampling, the algorithm attempts to show in which category, or cluster, the data belong to, with the number of clusters being defined by the value *k*.

The *k*-means algorithm is one of the simplest clustering techniques and it is commonly used in medical imaging, biometrics, and related fields. The advantage of *k*-means clustering is that it tells about your data (using its unsupervised form) rather than you having to instruct the algorithm about the data at the start (using the supervised form of the algorithm).

Working of K-Means Algorithm:

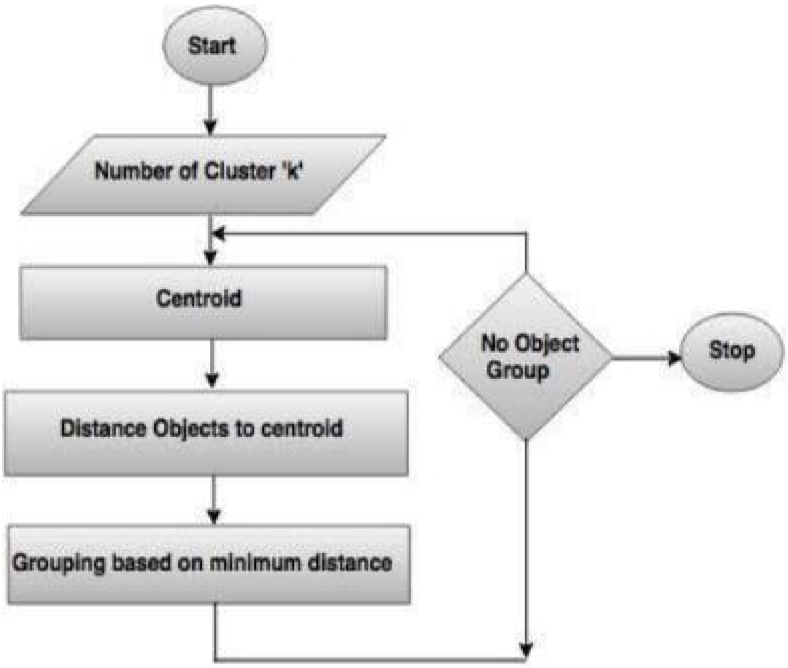
The *k*-means algorithm is an evolutionary algorithm that gains its name from its method of operation. The algorithm clusters observations into *k* groups, where *k* is provided as an input parameter. It then assigns each observation to clusters based upon the observation's proximity to the mean of the cluster. The cluster's mean is then recomputed and the process begins again. Here's how the algorithm works:

1. The algorithm arbitrarily selects k points as the initial cluster centers (the means).

2. Each point in the dataset is assigned to the closed cluster, based upon the Euclidean distance between each point and each cluster center.

3. Each cluster center is recomputed as the average of the points in that cluster.

4. Steps 2 and 3 repeat until the clusters converge. Convergence may be defined differently depending upon the implementation, but it normally means that either no observations change clusters when steps 2 and 3 are repeated, or that the changes do not make a material difference in the definition of the clusters.


Start
Number of Cluster 'k'
Centroid
Distance Objects to centroid
Grouping based on minimum distance
No Object Group
Stop

K- mean Algorithm

Advantages:

- With a large number of variables, k-means may be computationally faster than hierarchical clustering.(if k is small).
- K-means may produce tighter than hierarchical clustering, especially if the clusters are globular.

Disadvantages:

- Difficult in comparing the quality of the clusters produced
- Applicable only mean is defined
- Need to specify k, the no. of clusters in advance.
- Unable to handle noise data and outliers.

HIERARCHICAL METHODS

A Hierarchical clustering method works via grouping data into a tree of clusters. Hierarchical clustering begins by treating every data points as a separate cluster. Then, it repeatedly executes the subsequent steps:

1. Identify the 2 clusters which can be closest together, and

2. Merge the 2 maximum comparable clusters. We need to continue these steps until all the clusters are merged together.

In Hierarchical Clustering, the aim is to produce a hierarchical series of nested clusters. A diagram called Dendrogram (A Dendrogram is a tree-like diagram that statistics the sequences of merges or splits) graphically represents this hierarchy and is an inverted tree that describes the order in which factors are merged (bottom-up view) or cluster are break up (top-down view).

The basic method to generate hierarchical clustering are:

1. Agglomerative: Initially consider every data point as an individual Cluster and at every step, merge the nearest pairs of the cluster. (It is a bottom-up method). At first everydata set set is considered as individual entity or cluster. At every iteration, the clusters merge with different clusters until one cluster is formed.

Algorithm for Agglomerative Hierarchical Clustering is:

- Calculate the similarity of one cluster with all the other clusters (calculate proximity matrix)
- Consider every data point as a individual cluster
- Merge the clusters which are highly similar or close to each other.
- Recalculate the proximity matrix for each cluster

- Repeat Step 3 and 4 until only a single cluster remains.

Let’s see the graphical representation of this algorithm using a dendrogram.

Note: This is just a demonstration of how the actual algorithm works no calculation has been performed below all the proximity among the clusters are assumed.
Let’s say we have six data points A, B, C, D, E, F.

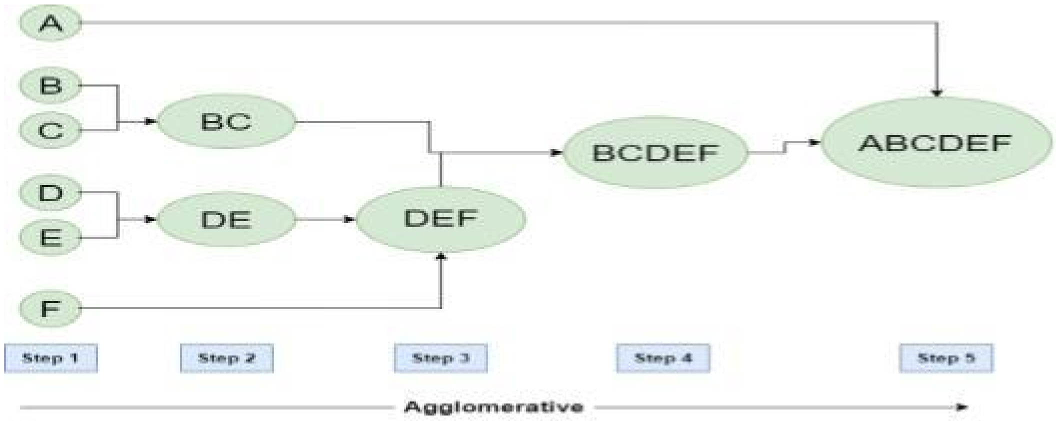


Figure – Agglomerative Hierarchical clustering

- Step-1: Consider each alphabet as a single cluster and calculate the distance of one cluster from all the other clusters.
- Step-2: In the second step comparable clusters are merged together to form a single cluster. Let’s say cluster (B) and cluster (C) are very similar to each other therefore we merge them in the second step similarly with cluster (D) and (E) and at last, we get the clusters [(A), (BC), (DE), (F)]
- Step-3: We recalculate the proximity according to the algorithm and merge the two nearest clusters([(DE), (F)]) together to form new clusters as [(A), (BC), (DEF)]
- Step-4: Repeating the same process; The clusters DEF and BC are comparable and merged together to form a new cluster. We’re now left with clusters [(A), (BCDEF)].
- Step-5: At last the two remaining clusters are merged together to form a single cluster [(ABCDEF)].

2. Divisive: We can say that the Divisive Hierarchical clustering is precisely the opposite of the Agglomerative Hierarchical clustering. In Divisive Hierarchical clustering, we take into account all of the data points as a single cluster and in every iteration, we separate the data points from the clusters which aren’t comparable. In the end, we are left with N clusters.

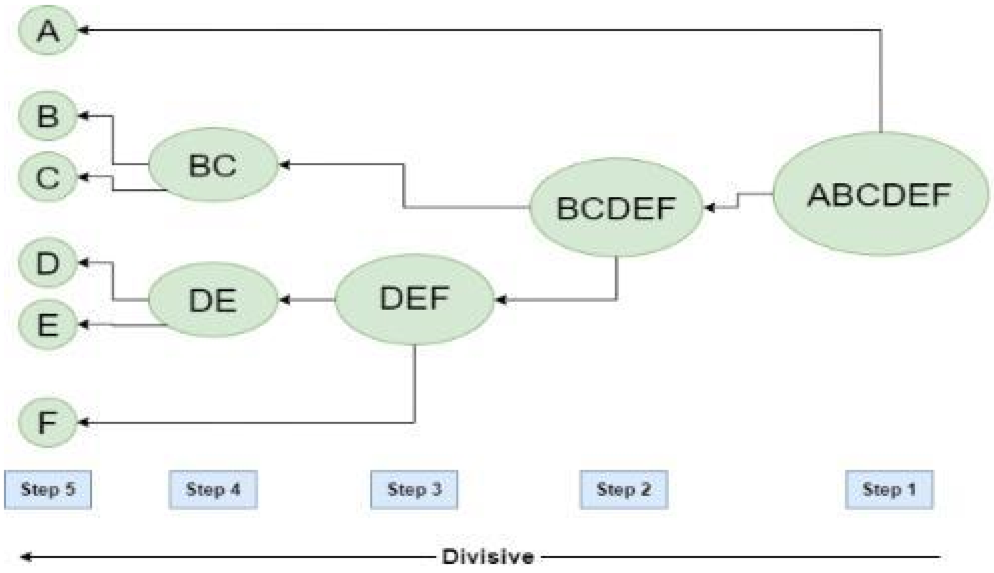


Figure – Divisive Hierarchical clustering

DENSITY-BASED METHODS

Density-based Method: This method is based on the notion of density. The basic idea is to continue growing the given cluster as long as the density in the neighborhood exceeds some threshold, i.e., for each data point within a given cluster, the radius of a given cluster has to contain at least a minimum number of points.

Density based clustering algorithm: Density based clustering algorithm has played a vital role in finding non linear shapes structure based on the density. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is most widely used density based algorithm. It uses the concept of density reachability and density connectivity.

Density Reachability - A point "p" is said to be density reachable from a point "q" if point "p" is within ε distance from point "q" and "q" has sufficient number of points in its neighbors which are within distance ε.

Density Connectivity - A point "p" and "q" are said to be density connected if there exist a point "r" which has sufficient number of points in its neighbors and both the points "p" and "q" are within the ε distance. This is chaining process. So, if "q" is neighbor of "r", "r" is neighbor of "s", "s" is neighbor of "t" which in turn is neighbor of "p" implies that "q" is neighbor of "p".

Algorithmic steps for DBSCAN clustering

Let X = {x1, x2, x3, ..., xn} be the set of data points. DBSCAN requires two parameters: ε (eps) and the minimum number of points required to form a cluster (minPts).

1) Start with an arbitrary starting point that has not been visited.

2) Extract the neighborhood of this point using ε (All points which are within the ε distance are neighborhood).

3) If there are sufficient neighborhood around this point then clustering process starts and point is marked as visited else this point is labeled as noise (Later this point can become the part of the cluster).

4) If a point is found to be a part of the cluster then its ε neighborhood is also the part of the cluster and the above procedure from step 2 is repeated for all ε neighborhood points. This is repeated until all points in the cluster is determined.

5) A new unvisited point is retrieved and processed, leading to the discovery of a further cluster or noise.
6) This process continues until all points are marked as visited.

Advantages

1) Does not require a-priori specification of number of clusters.

2) Able to identify noise data while clustering.

3) DBSCAN algorithm is able to find arbitrarily size and arbitrarily shaped clusters.

Disadvantages

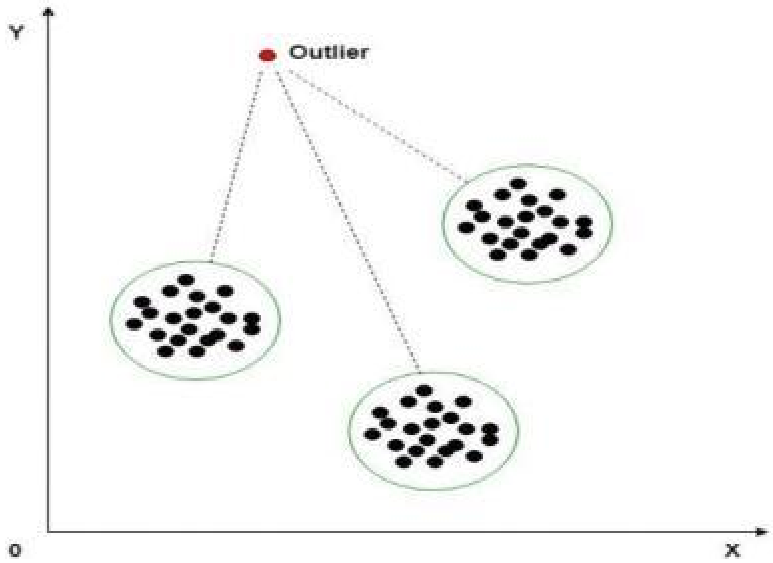
1) DBSCAN algorithm fails in case of varying density clusters.
2) Fails in case of neck type of dataset.

OUTLIER ANALYSIS

An **outlier** is an object that deviates significantly from the rest of the objects. They can be caused by measurement or execution error. The analysis of outlier data is referred to as outlier analysis or outlier mining.

Why outlier analysis?

Most data mining methods discard outliers noise or exceptions, however, in some applications such as fraud detection, the rare events can be more interesting than the more regularly occurring one and hence, the outlier analysis becomes important in such case.



Detecting Outlier:

Clustering based outlier detection using distance to the closest cluster: In the K-Means clustering technique, each cluster has a mean value. Objects belong to the cluster whose mean value is closest to it. In order to identify the Outlier, firstly we need to initialize the threshold value such that any distance of any data point greater than it from its nearest cluster identifies it as an outlier for our purpose. Then we need to find the distance of the test data to each cluster mean. Now, if the distance between the test data and the closest cluster to it is greater than the threshold value then we will classify the test data as an outlier.

Algorithm:

1. Calculate the mean of each cluster

2. Initialize the Threshold value

3. Calculate the distance of the test data from each cluster mean

4. Find the nearest cluster to the test data

5. If (Distance > Threshold) then, Outlier

ASSIGNMENT-4

SECTION A 1*5=5M

1. Define Cluster.
2. write a note on types of data in cluster analysis
3. Explain about outlier analysis

4. write a short note on Prediction

5. Briefly explain about a categorization of major clustering methods

SECTION-B 5*5=25M

1. Explain about Classification by Back propagationin detail.

2.Explain in depth about Support Vector Machine.

3. Explain in depth aboutPartitioning methods.

4. write about Hierarchical methods.
5. Explain about Density-based methods.