Introduction to prediction: linear and multiple regression, Clustering: types of data in cluster analysis: interval scaled variables, Binary variables, Nominal, ordinal, and Ratio-scaled variables; Major Clustering Methods: Partitioning Methods: K-Mean and K-Mediods, Hierarichal methods: Agglomerative, Density based methods: DBSCAN

There are two forms of data analysis that can be used for extracting models describing important classes or to predict future data trends. These two forms are as follows −

- Classification
- Prediction

Classification models predict categorical class labels; and prediction models predict continuous valued functions. For example, we can build a classification model to categorize bank loan applications as either safe or risky, or a prediction model to predict the expenditures in dollars of potential customers on computer equipment given their income and occupation.

What is classification?

Following are the examples of cases where the data analysis task is Classification −

- A bank loan officer wants to analyze the data in order to know which customer (loan applicant) are risky or which are safe.
- A marketing manager at a company needs to analyze a customer with a given profile, who will buy a new computer.

In both of the above examples, a model or classifier is constructed to predict the categorical labels. These labels are risky or safe for loan application data and yes or no for marketing data.

What is prediction?

Following are the examples of cases where the data analysis task is Prediction −

Suppose the marketing manager needs to predict how much a given customer will spend during a sale at his company. In this example we are bothered to predict a numeric value. Therefore the data analysis task is an example of numeric prediction. In this case, a model or a predictor will be constructed that predicts a continuous-valued-function or ordered value.

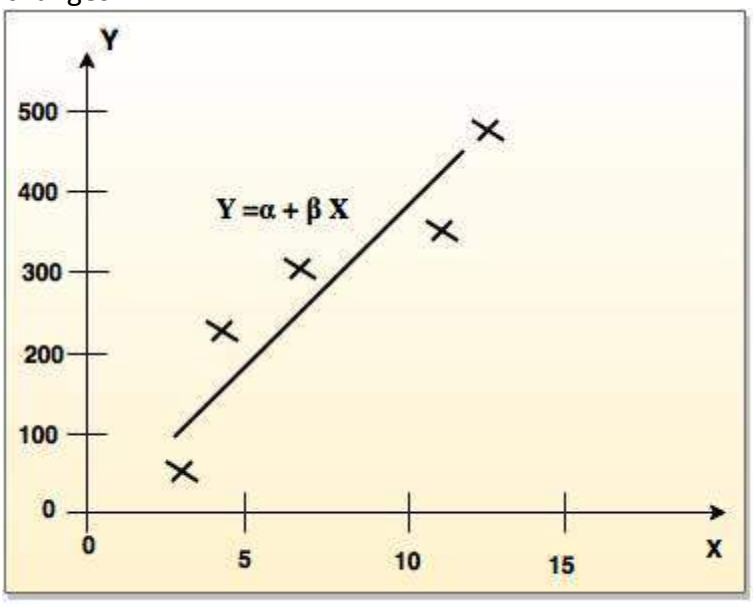
Regression involves **predictor variable** (the values which are known) and **response variable** (values to be predicted).

The two basic types of regression are:

1. Linear regression

- It is simplest form of regression. Linear regression attempts to model the relationship between two variables by fitting a linear equation to observe the data.
- Linear regression attempts to find the mathematical relationship between variables.
- If outcome is straight line then it is considered as linear model and if it is curved line, then it is a non linear model.
- The relationship between dependent variable is given by straight line and it has only one independent variable.

  $Y = \alpha + B X$
- Model **'Y'**, is a linear function of **'X'**.
- The value of 'Y' increases or decreases in linear manner according to which the value of 'X' also changes.



Linear Regression

2. Multiple regression model

- Multiple linear regression is an extension of linear regression analysis.
- It uses two or more independent variables to predict an outcome and a single continuous dependent variable.

  $Y = a_0 + a_1 X_1 + a_2 X_2 + \ldots\ldots\ldots + a_k X_k + e$

where,

'Y' is the response variable.

$X_1 + X_2 + X_k$ are the independent predictors.

'e' is random error.

a_0, a_1, a_2, a_k are the regression coefficients.

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.

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

Points to Remember

- 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.

Requirements of Clustering in Data Mining

The following points throw light on why clustering is required in data mining −

- **Scalability** − We need highly scalable clustering algorithms to deal with large databases.
- **Ability to deal with different kinds of attributes** − Algorithms should be capable to be applied on any kind of data such as interval-based (numerical) data, categorical, and binary data.
- **Discovery of clusters with attribute shape** − The clustering algorithm should be capable of detecting clusters of arbitrary shape. They should not be bounded to only distance measures that tend to find spherical cluster of small sizes.
- **High dimensionality** − The clustering algorithm should not only be able to handle low-dimensional data but also the high dimensional space.
- **Ability to deal with noisy data** − Databases contain noisy, missing or erroneous data. Some algorithms are sensitive to such data and may lead to poor quality clusters.
- **Interpretability** − The clustering results should be interpretable, comprehensible, and usable.

Types Of Data Used In Cluster Analysis Are:

- Interval-Scaled variables
- Binary variables
- Nominal, Ordinal, and Ratio variables
- Variables of mixed types

Interval-Scaled Variables

Interval-scaled variables are continuous measurements of a roughly linear scale.

Typical examples include weight and height, latitude and longitude coordinates (e.g., when clustering houses), and weather temperature. The measurement unit used can affect the clustering analysis. For example, changing measurement units from meters to inches for height, or from kilograms to pounds for weight, may lead to a very different clustering structure.

In general, expressing a variable in smaller units will lead to a larger range for that variable, and thus a larger effect on the resulting clustering structure. To help avoid dependence on the choice of measurement units, the data should be standardized. Standardizing measurements attempts to give all variables an equal weight.This is especially useful when given no prior knowledge of the data. However, in some applications, users may intentionally want to give

more weight to a certain set of variables than to others. For example, when clustering basketball player candidates, we may prefer to give more weight to the variable height.

Binary Variables

A binary variable is a variable that can take only 2 values. For example, generally gender variables can take 2 variables male and female.

Contingency Table For Binary Data

Let us consider binary values 0 and 1
Let p=a+b+c+d

Simple matching coefficient (invariant, if the binary variable is symmetric):
Jaccard coefficient (noninvariant if the binary variable is asymmetric):

Nominal or Categorical Variables

A generalization of the binary variable in that it can take more than 2 states, e.g., red, yellow, blue, green.

Method 1: Simple matching

The dissimilarity between two objects i and j can be computed based on the simple matching.

m: Let m be no of matches (i.e., the number of variables for which i and j are in the same state).

p: Let p be total no of variables.

Method 2: use a large number of binary variables

Creating a new binary variable for each of the M nominal states.

Ordinal Variables

An ordinal variable can be discrete or continuous. In this order is important, e.g., rank.

It can be treated like interval-scaled

By replacing xif by their rank,

By mapping the range of each variable onto [0, 1] by replacing i-th object in the f-th variable by,

Then compute the dissimilarity using methods for interval-scaled variables.

Ratio-Scaled Intervals

Ratio-scaled variable: It is a positive measurement on a nonlinear scale, approximately at an exponential scale, such as Ae^Bt or A^e-Bt.

Methods:

- First, treat them like interval-scaled variables — not a good choice! (why?)
- Then apply logarithmic transformation i.e.y = log(x)
    - Finally, treat them as continuous ordinal data treat their rank as interval-scaled.

Variables Of Mixed Type

A database may contain all the six types of variables
symmetric binary, asymmetric binary, nominal, ordinal, interval, and ratio.
And those combined called as mixed-type variables.

Clustering Methods

Clustering methods can be classified into the following categories −

- Partitioning Method
- Hierarchical Method
- Density-based Method
- Grid-Based Method
- Model-Based Method
- Constraint-based Method

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.

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.

Approaches to Improve Quality of Hierarchical Clustering

Here are the two approaches that are used to improve the quality of hierarchical clustering −

- Perform careful analysis of object linkages at each hierarchical partitioning.
- Integrate hierarchical agglomeration by first using a hierarchical agglomerative algorithm to group objects into micro-clusters, and then performing macro-clustering on the micro-clusters.

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.

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.

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.

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.

K-means clustering in Data Mining

- K-means clustering is simple unsupervised learning algorithm developed by **J. MacQueen in 1967 and then J.A Hartigan and M.A Wong in 1975.**
- In this approach, the data objects ('n') are classified into 'k' number of clusters in which each observation belongs to the cluster with nearest mean.
- It defines 'k' sets (the point may be considered as the center of a one or two dimensional figure), one for each cluster $k \leq n$. The clusters are placed far away from each other.
- Then , it organizes the data in appropriate data set and associates to the nearest set. If there is no data pending, first step is complicated to perform, in this case an early grouping is done. It is necessary to re-calculate 'k' new set as barycenters of the clusters from previous step.
- After having these 'k' new sets, the same data set points and the nearest new sets are bound together.
- Finally, a loop is generated. As a result of this loop, the 'k' sets change their location step by step until no more changes are made.

Finally, this algorithm aims at minimizing an objective function as:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

where,

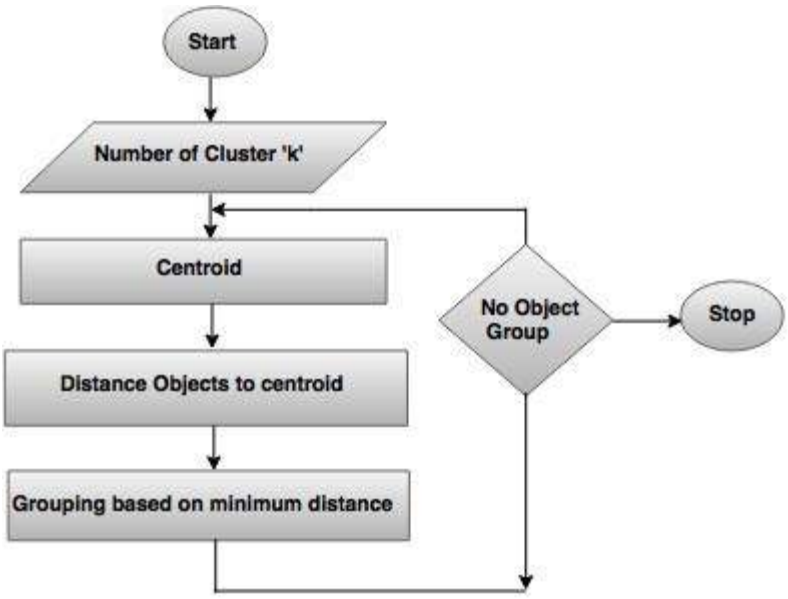
$x_i^{(j)}$ = data point

c_j = cluster center

n = Number of data points

k = Number of cluster

$||x_i^{(j)} - c_j||^2$ = distance between a data point $x_i^{(j)}$ and cluster centre c_j.



K- mean Algorithm

Solved examples of K-means:

Method 1:

Using K-means clustering, cluster the following data into two clusters and show each step.

{2, 4, 10, 12, 3, 20, 30, 11, 25}

Solution:

Given: {2, 4, 10, 12, 3, 20, 30, 11, 25}

Step 1: Assign alternate value to each cluster randomly.

Step 2: k_1= {2, 10, 3, 30, 25}, Mean value= 14

k_2= {4, 12, 20, 11, 25}, Mean value = 11.75

Step 3: Again assign the values,

k_1 = {20, 30, 25}, Mean value = 25

k_2 = {2, 4, 10, 12, 3, 11}, Mean value = 7

Step 4: Again assign the values,

k_1 = {20, 30, 25}, Mean value = 25

k_2 = {2, 4, 10, 12, 3, 11}, Mean value = 7

Method 2:

Step 1: Randomly assign the means: $m_1 = 3$, $m_2 = 4$

Step 2: Group the numbers close to mean $m_1 = 3$ are grouped into cluster k_1 and $m_2 = 4$ are grouped into cluster k_2

Step 3: k_1 = {2, 3}, k_2 = {4, 10, 12, 20, 30, 11, 25}, m_1= 2.5, m_2 = 16

Step 4: k_2 = {2, 3, 4}, k_2 = {10, 12, 20, 30, 11, 25}, m_1= 3 m_2 = 18

Step 5: k_1 = {2, 3, 4, 10}, k_2= {12, 20, 30, 11, 25}, m_1= 4.75, m_2 = 19.6

Step 6: k_1 = {2, 3, 4, 10, 11, 12}, k_2 = {20, 30, 25}, $m_1 = 7$, $m_2 = 25$

Step 7: k_1 = {2, 3, 4, 10, 11, 12}, k_2 = {20, 30, 25}, $m_1 = 7$, $m_2 = 25$

Step 8: Stop. The clusters in step 6 and 7 are same.

Final answer: k_1 = {2, 3, 4, 10, 11, 12} and k_2 = {20, 30, 25}

K-Medoids

- Instead of taking the mean value of the object in a cluster as a reference point, medoid can be used. It is most centrally located object. It is also called as **Partitioning Around Medoids (PAM).**

- In a single partition of data into 'K' clusters, where each cluster consists of point, this is centrally located point of the cluster based on some distance measure. These representative points are called as **medoids.**

Hierarchical Clustering in Data Mining

Hierarchy is more informative structure rather than the unstructured set of clusters returned by non hierarchical clustering.

Basic algorithm:

- Compute the proximity (similarity) matrix.
- Let each data point be cluster.
- Merge the two closest clusters.
- Update the proximity matrix until only one cluster remains.

Different approaches to define the cluster between the clusters.

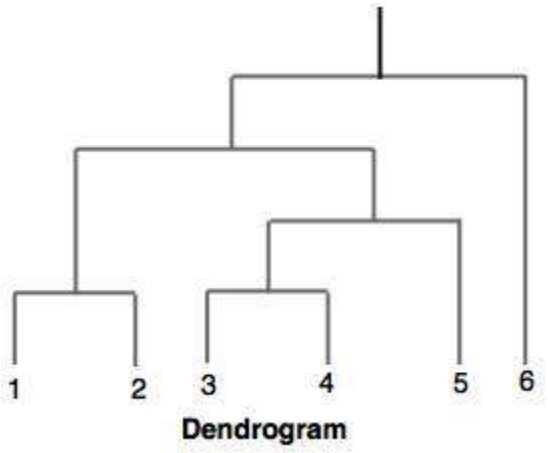
1. Single linkage
In this algorithm, the pair of clusters having shortest distance is considered, if there exists the similarity between two clusters.

2. Complete linkage
In this method, the distance between one cluster and another cluster should be equal to the greatest distance from any member of one cluster to any member of the other cluster.

3. Average linkage
In this method, the distance between one cluster and another cluster should be equal to average distance from any member of one cluster to any member of the other cluster.

Dendrogram

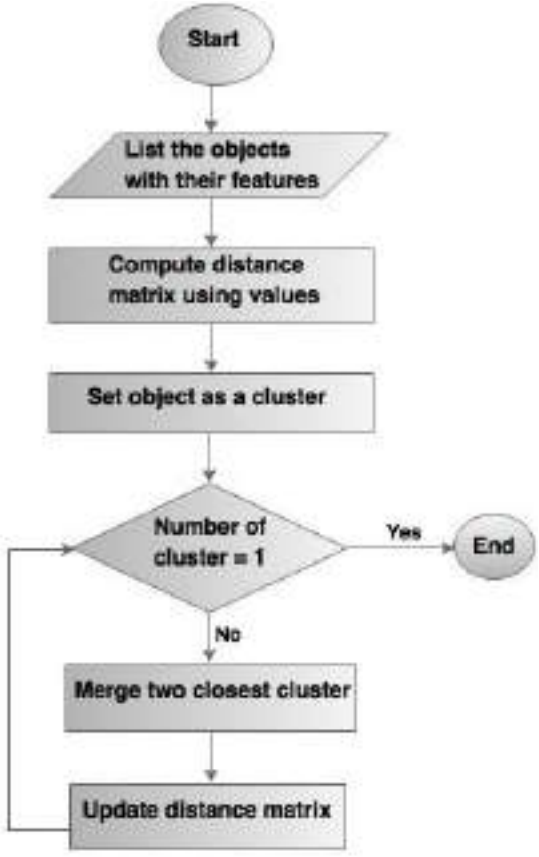
It is a tree structure diagram which illustrates hierarchical clustering techniques. Each level shows clusters for that level.



Dendrogram

Agglomerative hierarchical clustering

- It is a bottom-up approach, in which clusters have sub-clusters.
- The process is explained in the following flowchart.

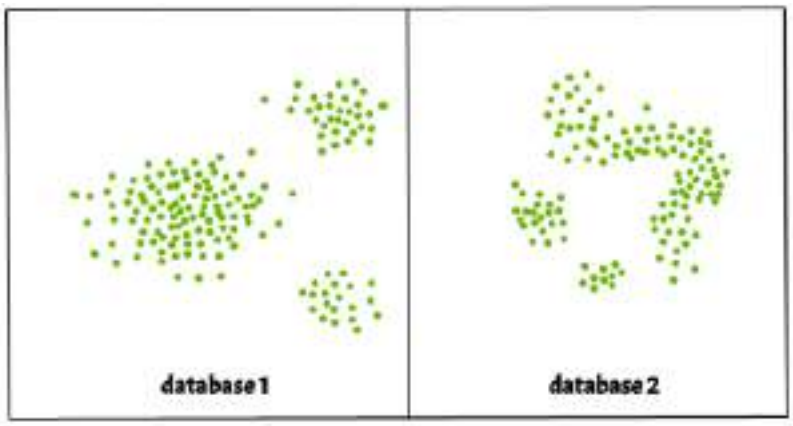


Agglomerative hierarchical clustering flowchart

Density based clustering

Clustering analysis or simply Clustering is basically an Unsupervised learning method that divides the data points into a number of specific batches or groups, such that the data points in the same groups have similar properties and data points in different groups have different properties in some sense. It comprises of many different methods based on different evolution. E.g. K-Means (distance between points), Affinity propagation (graph distance), Mean-shift (distance between points), DBSCAN (distance between nearest points), Gaussian mixtures (Mahalanobis distance to centers), Spectral clustering (graph distance) etc.

Fundamentally, all clustering methods use the same approach i.e. first we calculate similarities and then we use it to cluster the data points into groups or batches. Here we will focus on **Density-based spatial clustering of applications with noise** (DBSCAN) clustering method.
Clusters are dense regions in the data space, separated by regions of the lower density of points. The ***DBSCAN algorithm*** is based on this intuitive notion of "clusters" and "noise". The key idea is that for each point of a cluster, the neighborhood of a given radius has to contain at least a minimum number of points.



Why DBSCAN ?

Partitioning methods (K-means, PAM clustering) and hierarchical clustering work for finding spherical-shaped clusters or convex clusters. In other words, they are suitable only for compact and well-separated clusters. Moreover, they are also severely affected by the presence of noise and outliers in the data.
Real life data may contain irregularities, like –
i) Clusters can be of arbitrary shape such as those shown in the figure below.
ii) Data may contain noise.

DBSCAN algorithm requires two parameters –

1. **eps** : It defines the neighborhood around a data point i.e. if the distance between two points is lower or equal to 'eps' then they are considered as neighbors. If the eps value is chosen too small then large part of the data will be considered as outliers. If it is chosen

very large then the clusters will merge and majority of the data points will be in the same clusters. One way to find the eps value is based on the ***k-distance graph***.

2. **MinPts**: Minimum number of neighbors (data points) within eps radius. Larger the dataset, the larger value of MinPts must be chosen. As a general rule, the minimum MinPts can be derived from the number of dimensions D in the dataset as, MinPts >= D+1. The minimum value of MinPts must be chosen at least 3.

In this algorithm, we have 3 types of data points.
Core Point*: A point is a core point if it has more than MinPts points within eps.*
Border Point*: A point which has fewer than MinPts within eps but it is in the neighborhood of a core point.*
Noise or outlier*: A point which is not a core point or border point.*

DBSCAN algorithm can be abstracted in the following steps –

1. Find all the neighbor points within eps and identify the core points or visited with more than MinPts neighbors.
2. For each core point if it is not already assigned to a cluster, create a new cluster.
3. Find recursively all its density connected points and assign them to the same cluster as the core point.
   A point *a* and *b* are said to be density connected if there exist a point *c* which has a sufficient number of points in its neighbors and both the points *a* and *b* are within the *eps distance*. This is a chaining process. So, if *b* is neighbor of *c*, *c* is neighbor of *d*, *d* is neighbor of *e*, which in turn is neighbor of *a* implies that *b* is neighbor of *a*.
4. Iterate through the remaining unvisited points in the dataset. Those points that do not belong to any cluster are noise.