RGPVNOTES.IN

Program : **B.E**

Subject Name: **Data Mining**

Subject Code: **CS-8003**

Semester: **8th**

Unit-V Classification:-Introduction, Decision Tree, The Tree Induction Algorithm, Split Algorithms Based on Information Theory, Split Algorithm Based on the Gini Index, Overfitting and Pruning, Decision Trees Rules, Naïve Bayes Method. Cluster Analysis:- Introduction, Desired Features of Cluster Analysis, Types of Cluster Analysis Methods:- Partitional Methods, Hierarchical Methods, Density- Based Methods, Dealing with Large Databases. Quality and Validity of Cluster Analysis Methods.

Classification:

Introduction:

Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data. For example, a classification model could be used to identify loan applicants as low, medium, or high credit risks.

There are various applications of classification algorithms as

1. Medical Diagnosis
2. Image and pattern recognition
3. Fault detection
4. Financial market position etc.

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.

There are three main approaches to classify problem:

1. The first approach divides the space defined by data points into regions and each reagion correspond to a given class.
2. The second approach is to find the probability of an example belonging to each class.
3. The third approach is to find the probability of a class containing that example.

Decision Tree:

A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node.

The following decision tree is for the concept buy_computer that indicates whether a customer at a company is likely to buy a computer or not. Each internal node represents a test on an attribute. Each leaf node represents a class shown in figure 1.

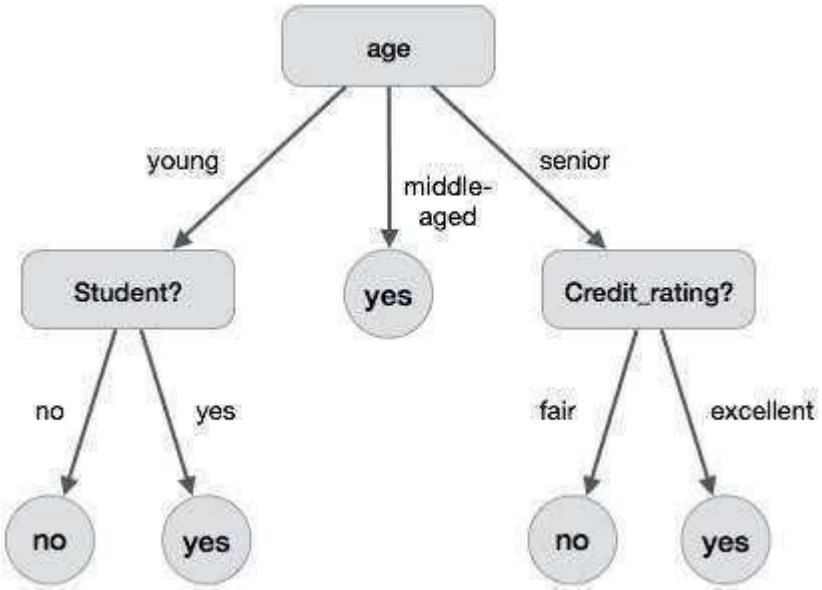


Figure 1: Decision Tree

The benefits of having a decision tree are as follows −

- It does not require any domain knowledge.
- It is easy to comprehend.
- The learning and classification steps of a decision tree are simple and fast.

The Tree Induction Algorithm:

Decision Tree Induction Algorithm

A machine researcher named J. Ross Quinlan in 1980 developed a decision tree algorithm known as ID3 (Iterative Dichotomiser). Later, he presented C4.5, which was the successor of ID3. ID3 and C4.5 adopt a greedy approach. In this algorithm, there is no backtracking; the trees are constructed in a top-down recursive divide-and-conquer manner.

The algorithm needs three parameters : D(Data Partition), Attribute List, and Attribute Selection Method. Initially, D is the entire set of training tuples and associated class labels. Attribute list is

a list of attributes describing the tuples. Attribute selection method specifies a heuristic procedure for selecting the attribute that "best" discriminates the given tuples according to class.

Generating a decision tree form training tuples of data partition D
Algorithm : Generate_decision_tree

Input:
Data partition, D, which is a set of training tuples
and their associated class labels.
attribute_list, the set of candidate attributes.
Attribute selection method, a procedure to determine the
splitting criterion that best partitions that the data
tuples into individual classes. This criterion includes a
splitting_attribute and either a splitting point or splitting subset.

Output:
A Decision Tree

Method
create a node N;

if tuples in D are all of the same class, C then
  return N as leaf node labeled with class C;

if attribute_list is empty then
  return N as leaf node with labeled
  with majority class in D;|| majority voting

apply attribute_selection_method(D, attribute_list)
to find the best splitting_criterion;
label node N with splitting_criterion;

if splitting_attribute is discrete-valued and
  multiway splits allowed then // no restricted to binary trees

```
attribute_list = splitting attribute; // remove splitting attribute
for each outcome j of splitting criterion

  // partition the tuples and grow subtrees for each partition
  let Dj be the set of data tuples in D satisfying outcome j; // a partition

  if Dj is empty then
    attach a leaf labeled with the majority
    class in D to node N;
  else
    attach the node returned by Generate
    decision tree(Dj, attribute list) to node N;
  end for
return N;
```

Split Algorithms Based on Information Theory:

Splitting Criterion

- Work out entropy based on distribution of classes.
- Trying splitting on each attribute.
- Work out expected information gain for each attribute.
- Choose best attribute.

Split Algorithm Based on the Gini Index:

The Gini Index is calculated by subtracting the sum of the squared probabilities of each class from one. It favors larger partitions. Information Gain multiplies the probability of the class times the log (base=2) of that class probability. Information Gain favors smaller partitions with many distinct values. Ultimately, you have to experiment with your data and the splitting criterion.

The Gini Index measure of goodness is based on the measure of diversity. The best splitter is one that decreases the diversity of the record sets by the greatest amount. In other words we want to maximize.

Algorithm / Split Criterion	Description	Tree Type
Gini Split / Gini Index	Favors larger partitions. Very simple to implement.	CART
Information Gain / Entropy	Favors partitions that have small counts but many distinct values.	ID3 / C4.5

Diversity (before split) – diversity (left child)+diversity (right child)

Using Gini Split / Gini Index

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

Where pi is the relative frequency of class i in Gini.

- Favors larger partitions.
- Uses squared proportion of classes.
- Perfectly classified, Gini Index would be zero.
- Evenly distributed would be 1 – (1/# Classes).
- You want a variable split that has a low Gini Index.
- The algorithm works as 1 – (P(class1)^2 + P(class2)^2 + … + P(classN)^2)

The Gini index is used in the classic CART algorithm and is very easy to calculate.

Gini Index:

for each branch in split:

Calculate percent branch represents #Used for weighting

for each class in branch:

Calculate probability of class in the given branch.

Square the class probability.

Sum the squared class probabilities.

Subtract the sum from 1. #This is the Ginin Index for branch

Weight each branch based on the baseline probability.

Sum the weighted gini index for each split.

Overfitting and Pruning:

Overfitting is a significant practical difficulty for decision tree models and many other predictive models. Overfitting happens when the learning algorithm continues to develop hypotheses that reduce training set error at the cost of an increased test set error. There are several approaches to avoiding overfitting in building decision trees.

- Pre-pruning that stop growing the tree earlier, before it perfectly classifies the training set.
- Post-pruning that allows the tree to perfectly classify the training set, and then post prune the tree.
- These models are incorrect
- They require collection of unnecessary features
- Also they are more difficult to comprehend.

Practically, the second approach of post-pruning overfit trees is more successful because it is not easy to precisely estimate when to stop growing the tree.

Decision Trees Rules:

Once a decision tree has been constructed, it is a simple matter to convert it into an equivalent set of rules shown in figure 2.

Converting a decision tree to rules before pruning has three main advantages:

1. Converting to rules allows distinguishing among the different contexts in which a decision node is used.

    - Each distinct path through the tree produces a distinct rule.
    - Therefore, a single path can be pruned, rather than an entire decision node.
    - If the tree itself were pruned, the only possible actions would be to remove an entire node, or leave it in its original form.

2. Unlike the tree, the rules do not maintain a distinction between attribute tests that occur near the root of the tree and those that occur near the leaves.

    - This allows pruning to occur without having to consider how to re-build the tree if root nodes are removed.

3. Rules are easier for people to read and understand.

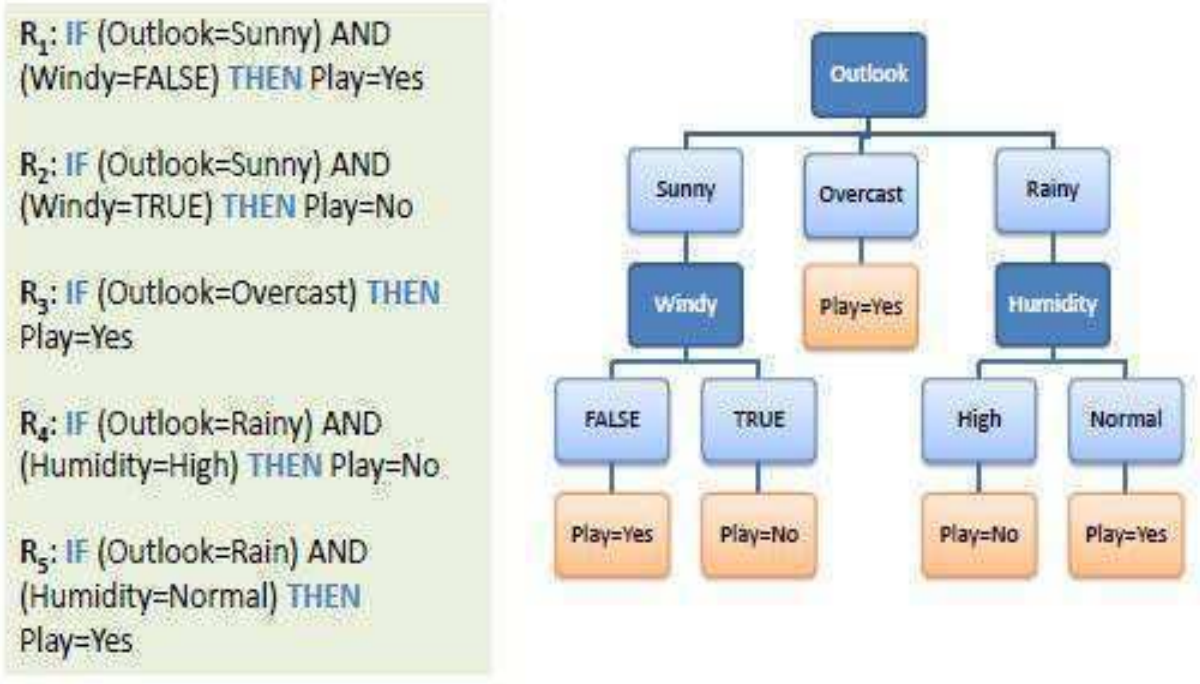


Figure 2: Decision Tree Rules

Naïve Bayes Method:

The Naive Bayesian classifier is based on Bayes' theorem with independence assumptions between predictors. A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets. Despite its simplicity, the Naive Bayesian classifier often does surprisingly well and is widely used because it often outperforms more sophisticated classification methods.

Advantages of Naïve Bayes Classifier

- It is easy to implement.

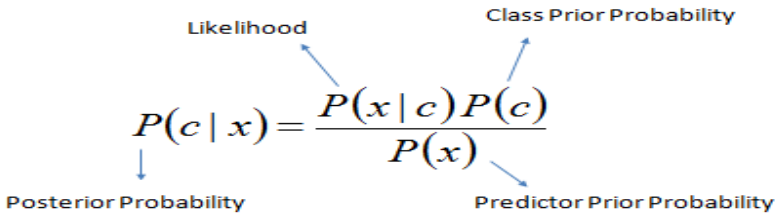
- In most of the cases we can obtain optimal results.

Disadvantages of Naïve Bayes Classifier

- There is an assumption of class conditional independence, therefore it may lead to loss of accuracy.
- We can deal with dependencies using Bayesian belief Networks.

Algorithm

Bayes theorem provides a way of calculating the posterior probability, P(c|x), from P(c), P(x), and P(x|c). Naive Bayes classifier assumes that the effect of the value of a predictor (x) on a given class (c) is independent of the values of other predictors. This assumption is called class conditional independence.



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

- P(c|x) is the posterior probability of class (target) given predictor (attribute).
- P(c) is the prior probability of class.
- P(x|c) is the likelihood which is the probability of predictor given class.
- P(x) is the prior probability of predictor.

Cluster Analysis

Introduction

Cluster analysis is the process of finding groups of objects in such a way that the objects of a group are similar (or related) to one another and different from (or unrelated) to the objects of the other groups.

Cluster analysis is a multivariate method which aims to classify a sample of subjects (or objects) on the basis of a set of measured variables into a number of different groups such that similar subjects are placed in the same group. An example where this might be used is in the field of psychiatry, where the characterization of patients on the basis of clusters of symptoms can be useful in the identification of an appropriate form of therapy. In marketing, it may be useful to identify distinct groups of potential customers so that, for example, advertising can be appropriately targeted.

The idea of cluster analysis is that we have a set of observations, on which we have available several measurements. Using these measurements, we want to find out if the observations naturally group together in some predictable way. For example, we may have recorded physical measurements on many animals, and we want to know if there's a natural grouping (based, perhaps on species) that distinquishes the animals from another. (This use of cluster analysis is sometimes called "numerical taxonomy").

Desired Features of 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.

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 (Characteristics of clustering techniques)

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.
- High Dimensionality
- Constraint based clustering

Types of Cluster Analysis Methods:-

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.

Advantage

- 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.

Dealing with Large Databases:

For details, please refer following link.
http://www.irma-international.org/viewtitle/31522/
https://arxiv.org/ftp/arxiv/papers/1307/1307.5437.pdf

Quality and Validity of Cluster Analysis Methods

The term cluster validation is used to design the procedure of evaluating the goodness of clustering algorithm results. This is important to avoid finding patterns in a random data, as well as, in the situation where you want to compare two clustering algorithms.
Generally, clustering validation statistics can be categorized into 3 classes (Charrad et al. 2014, Brock et al. (2008), Theodoridis and Koutroumbas (2008)):

1. Internal cluster validation, which uses the internal information of the clustering process to evaluate the goodness of a clustering structure without reference to external information. It can be also used for estimating the number of clusters and the appropriate clustering algorithm without any external data.
2. External cluster validation, which consists in comparing the results of a cluster analysis to an externally known result, such as externally provided class labels. It measures the extent to which cluster labels match externally supplied class labels. Since we know the “true” cluster number in advance, this approach is mainly used for selecting the right clustering algorithm for a specific data set.
3. Relative cluster validation, which evaluates the clustering structure by varying different parameter values for the same algorithm (e.g.,: varying the number of clusters k). It’s generally used for determining the optimal number of clusters.

RGPVNOTES.IN

We hope you find these notes useful.

You can get previous year question papers at https://qp.rgpvnotes.in .

If you have any queries or you want to submit your study notes please write us at rgpvnotes.in@gmail.com

LIKE & FOLLOW US ON FACEBOOK
facebook.com/rgpvnotes.in