RGPVNOTES.IN

Please do not share these notes on apps like WhatsApp or Telegram.

The revenue we generate from the ads we show on our website and app funds our services. The generated revenue **helps us prepare new notes and improve the quality of existing study materials**, which are available on our website and mobile app.

If you don't use our website and app directly, it will hurt our revenue, and we might not be able to run the services and **have to close them**. So, it is a humble request for all to **stop sharing the study material** we provide on various apps. Please **share the website's URL instead.**

CS-703(B) Open Elective, Data Mining and Warehousing

UNIT-IV: Supervised Learning: Classification: Statistical-based algorithms, Distance-based algorithms, Decision tree-based algorithms, neural network-based algorithms, Rule-based algorithms, Probabilistic Classifiers

---

Supervised Learning

Supervised learning, also known as supervised machine learning, is a subcategory of machine learning and artificial intelligence. It is defined by its use of labeled datasets to train algorithms that to classify data or predict outcomes accurately. In supervised learning, algorithms learn from labeled data. After understanding the data, the algorithm determines which label should be given to new data by associating patterns to the unlabeled new data.

Supervised learning can be divided into two categories:

I. Classification: The model finds classes in which to place its inputs.

II. Regression: The model finds outputs that are real variables.

Basically supervised learning is a learning in which we train the machine using data which is well labeled that means some data is already tagged with the correct answer. After that, the machine is provided with a new set of examples (data) so that supervised learning algorithm analyses the training data (set of training examples) and produces a correct outcome from labeled data.

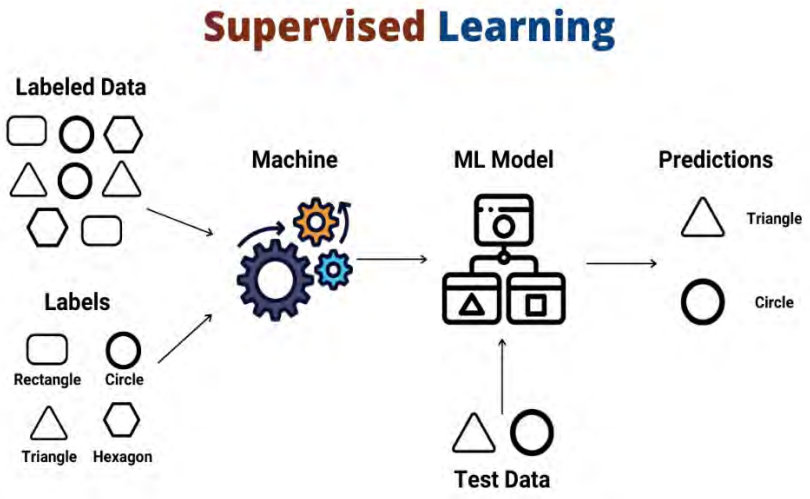

Figure 1: Supervised Learning

Classification

Classification is a technique for determining which class the dependent belongs to based on one or more independent variables. 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 −

i) Classification

ii) 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 region 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.

Statistical-based algorithms

Statistical Distribution-Based Outlier Detection: The statistical distribution-based approach to outlier detection assumes a distribution or probability model for the given data set (e.g., a Normal or Poisson distribution) and then identifies outliers with respect to the model using a discordancy test. Application of the test requires knowledge of the data set parameters knowledge of distribution parameters such as the mean and variance and the expected number of outliers.

A statistical discordancy test examines two hypotheses:

- A working hypothesis
- An alternative hypothesis

A working hypothesis, H, is a statement that the entire data set of n objects comes from an initial distribution model, F, that is,

$$H : o_i \in F, \quad \text{where } i = 1, 2, \ldots, n.$$

The hypothesis is retained if there is no statistically significant evidence supporting its rejection. A discordancy test verifies whether an object, o_i, is significantly large (or small) in relation to the distribution F. Different test statistics have been proposed for use as a discordancy test, depending on the available knowledge of the data.

Assuming that some statistic, T, has been chosen for discordancy testing, and the value of the statistic for object o_i is v_i, then the distribution of T is constructed. Significance probability, SP (v_i) = Prob ($T > v_i$), is evaluated. If SP (v_i) is sufficiently small, then o_i is discordant and the working hypothesis is rejected.

An alternative hypothesis, H, which states that o_i comes from another distribution model, G, is adopted. The result is very much dependent on which model F is chosen because o_i may be an outlier under one model and a perfectly valid value under another. The alternative distribution is very important in determining the power of the test, that is, the probability that the working hypothesis is rejected when o_i is really an outlier. There are different kinds of alternative distributions.

Inherent alternative distribution:

In this case, the working hypothesis that all the objects come from distribution F is rejected in favor of the alternative hypothesis that all of the objects arise from another distribution.

G: H:Pr (F>0) oi € G, where i = 1, 2…., n

F and G may be different distributions or differ only in parameters of the same distribution.

There are constraints on the form of the G distribution in that it must have potential to produce outliers. For example, it may have a different mean or dispersion.

Mixture alternative distribution:

The mixture alternative states that discordant values are not outliers in the F population, but contaminants from some other population, G. In this case, the alternative hypothesis is:

$$\overline{H} : o_i \in (1-\lambda)F + \lambda G, \quad \text{where } i = 1, 2, \ldots, n.$$

Slippage alternative distribution:

This alternative states that all of the objects (apart from some prescribed small number) arise independently from the initial model, F, with its given parameters, whereas the remaining objects are independent observations from a modified version of F in which the parameters have been shifted.

There are two basic types of procedures for detecting outliers:

Block procedures: In this case, either all of the suspect objects are treated as outliers or all of them are accepted as consistent.

Consecutive procedures: Its main idea is that the object that is least likely to be an outlier is tested first. If it is found to be an outlier, then more extreme values are also considered outliers; otherwise, the next most extreme object is tested, and so on. This procedure tends to be more effective than block procedures.

Distance-Based Outlier Detection:

The notion of distance-based outliers was introduced to counter the main limitations imposed by statistical methods. An object, o, in a data set, D, is a distance-based (DB) outlier with parameters pct and dmin, that is, a DB (pct; dmin)-outlier, if at least a fraction, pct, of the objects in D lie at a distance greater than dmin from o. In other words, rather than relying on statistical tests, we can think of distance-based outliers as those objects that do not have enough neighbors, where neighbors are defined based on distance from the given object. In comparison with statistical-based methods, distance based outlier detection generalizes the ideas behind discordancy testing for various standard distributions. Distance-based outlier detection avoids the excessive computation that can be associated with fitting the observed distribution into some standard distribution and in selecting discordancy tests.

For many discordancy tests, it can be shown that if an object, o, is an outlier according to the given test, then o is also a DB (pct, dmin)-outlier for some suitably defined pct and dmin. For example, if objects that lie three or more standard deviations from the mean are considered to be outliers, assuming a normal distribution, then this definition can be generalized by a DB (0.9988, 0.13s) outlier. Several efficient algorithms for mining distance-based outliers have been developed.

Index-based algorithm:

Given a data set, the index-based algorithm uses multi dimensional indexing structures, such as R-trees or k-d trees, to search for neighbors of each object o within radius dmin around that object. Let M be the maximum number of objects within the dmin-neighborhood of an outlier. Therefore, once M+1 neighbors of object o are found, it is clear that o is not an outlier. This algorithm has a worst-case complexity of O (n^2k), where n is the number of objects in the data set and k is the dimensionality. The index-based algorithm scales well as k increases. However, this complexity evaluation takes only the search time into account, even though the task of building an index in itself can be computationally intensive.

Nested-loop algorithm:

The nested-loop algorithm has the same computational complexity as the index-based algorithm but avoids index structure construction and tries to minimize the number of I/Os. It divides the memory buffer space into two halves and the data set into several logical blocks. By carefully choosing the order in which blocks are loaded into each half, I/O efficiency can be achieved.

Decision tree-based algorithms

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 2.

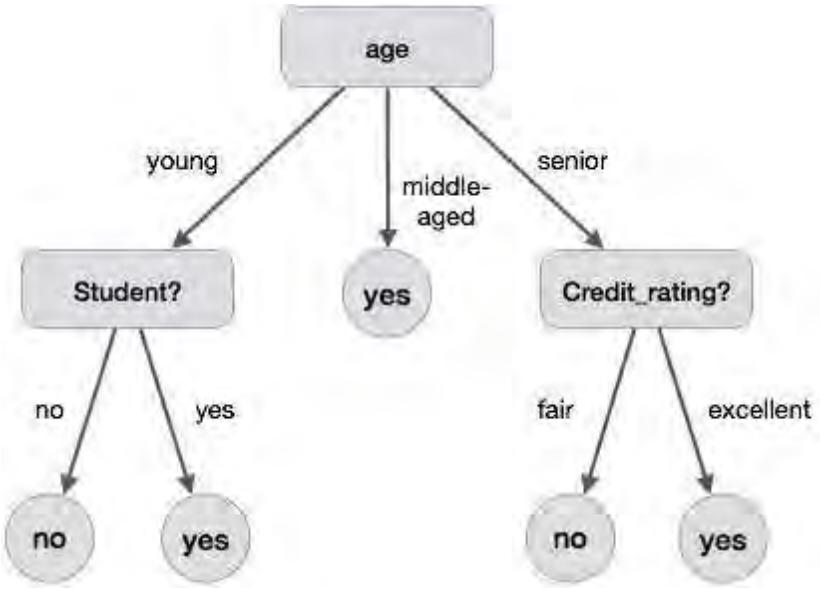

Figure 2: Decision Tree

The benefits of having a decision tree are as follows −

i. It does not require any domain knowledge.
ii. It is easy to comprehend.
iii. 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.

Neural network-based algorithms

- 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 time.
- 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, 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.

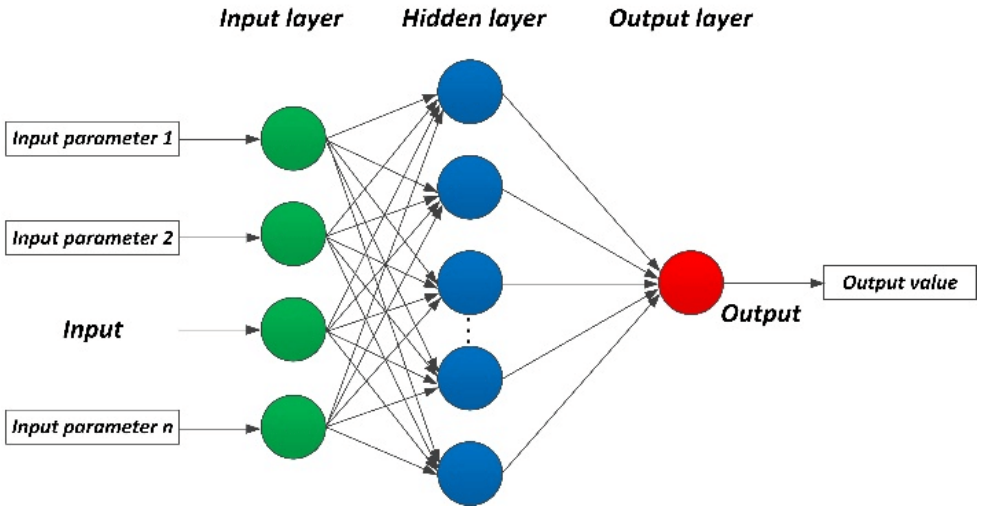

Figure 3: Neural Network

Advantages:

- It includes their high tolerance of noisy data as well as their ability to classify patterns on which they have not been trained.
- They can be used when user may have little knowledge of the relationships between attributes and classes.
- They are well-suited for continuous-valued inputs and outputs, unlike most decision tree algorithms.
- 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.
- Neural network algorithms are inherently parallel; parallelization techniques can be used to speed up the computation process.

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:

```
(1)  Initialize all weights and biases in network;
(2)  while terminating condition is not satisfied {
(3)      for each training tuple X in D {
(4)          // Propagate the inputs forward:
(5)          for each input layer unit j {
(6)              O_j = I_j; // output of an input unit is its actual input value
(7)          for each hidden or output layer unit j {
(8)              I_j = Σ_i w_ij O_i + θ_j; //compute the net input of unit j with respect to the
                     previous layer, i
(9)              O_j = 1/(1+e^{-I_j}); } // compute the output of each unit j
(10)         // Backpropagate the errors:
(11)         for each unit j in the output layer
(12)             Err_j = O_j(1 − O_j)(T_j − O_j); // compute the error
(13)         for each unit j in the hidden layers, from the last to the first hidden layer
(14)             Err_j = O_j(1 − O_j) Σ_k Err_k w_jk; // compute the error with respect to the
                     next higher layer, k
(15)         for each weight w_ij in network {
(16)             Δw_ij = (l)Err_j O_i; // weight increment
(17)             w_ij = w_ij + Δw_ij; } // weight update
(18)         for each bias θ_j in network {
(19)             Δθ_j = (l)Err_j; // bias increment
(20)             θ_j = θ_j + Δθ_j; } // bias update
(21)     } }
```

Rule-based algorithms:

IF-THEN Rules

Rule-based classifier makes use of a set of IF-THEN rules for classification.

Rule Format:

IF condition THEN conclusion

Let us consider a rule R1,

R1: IF age = youth AND student = yes

THEN buy_computer = yes

Points to remember −

- The IF part of the rule is called **rule antecedent** or **precondition**.
- The THEN part of the rule is called **rule consequent**.

- The antecedent part the condition consist of one or more attribute tests and these tests are logically ANDed.
- The consequent part consists of class prediction.

Note − We can also write rule R1 as follows −

R1: (age = youth) ^ (student = yes) (buys computer = yes)

If the condition holds true for a given tuple, then the antecedent is satisfied.

Rule Extraction:

Learn how to build a rule-based classifier by extracting IF-THEN rules from a decision tree.

Points to remember −

To extract a rule from a decision tree −

- One rule is created for each path from the root to the leaf node.
- To form a rule antecedent, each splitting criterion is logically ANDed.
- The leaf node holds the class prediction, forming the rule consequent.

Rule Induction Using Sequential Covering Algorithm:

Sequential Covering Algorithm can be used to extract IF-THEN rules form the training data. We do not require generating a decision tree first. In this algorithm, each rule for a given class covers many of the tuples of that class.

Some of the sequential Covering Algorithms are AQ, CN2, and RIPPER. As per the general strategy the rules are learned one at a time. For each time rules are learned, a tuple covered by the rule is removed and the process continues for the rest of the tuples. This is because the path to each leaf in a decision tree corresponds to a rule.

The following is the sequential learning algorithm where rules are learned for one class at a time. When learning a rule from a class Ci, rule to cover all the tuples from class C only and no tuple form any other class.

Algorithm: Sequential Covering

Input:

D, a data set class-labeled tuple,

Att_vals, the set of all attributes and their possible values.

Output: A Set of IF-THEN rules.

Method:

Rule_set={ }; // initial set of rules learned is empty

for each class c do

repeat

Rule = Learn_One_Rule(D, Att_valls, c);

Remove tuples covered by rule form D;

until termination condition;

Rule_set=Rule_set+Rule; // add a new rule to rule-set

end for

return Rule_Set;

Rule Pruning

The rule is pruned is due to the following reason −

- The Assessment of quality is made on the original set of training data.
- The rule may perform well on training data but less well on subsequent data.

FOIL is one of the simple and effective method for rule pruning. For a given rule R,

FOIL_Prune = pos - neg / pos + neg

Where pos and neg is the number of positive and negative tuples covered by R, respectively.

Probabilistic Classifiers:

A Bayes classifier is a probabilistic model that is used for supervised learning. A Bayes classifier is based on the idea that the role of a class is to predict the values of features for members of that class. Examples are grouped in classes because they have common values for some of the features. Such classes are often called natural kinds. The learning agent learns how the features depend on the class and uses that model to predict the classification of a new example.

The simplest case is the naive Bayes classifier, which makes the independence assumption that the input features are conditionally independent of each other given the classification. The independence of the naive Bayes classifier is embodied in a belief network where the features are the nodes, the target feature (the classification) has no parents, and the target feature is the only parent of each input feature. This belief network requires the probability distributions P(X) P(Y) for the target feature, or class, YY and P(Xi|Y) P(Xi|Y) for each input feature XiXi. For example, the prediction is computed by conditioning on observed values for the input features and querying the classification. Multiple target variables can be modeled and learned separately.

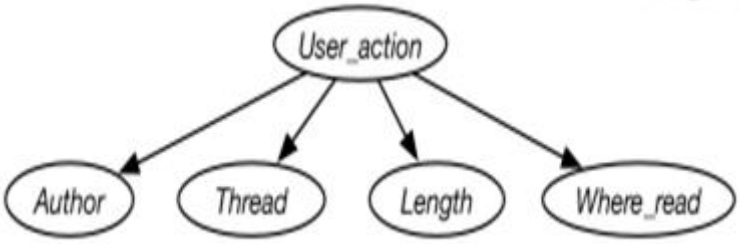

Figure 4: Belief network corresponding to a naive Bayes classifier

Learning a Bayes Classifier

To learn a classifier, the distributions of P(Y)P(Y) and P(Xi|Y)P(Xi|Y) for each input feature can be learned from the data. Each conditional probability distribution P(Xi|Y)P(Xi|Y) may be treated as a separate learning problem for each value of YY.

The simplest case is to use the maximum likelihood estimate (the empirical proportion in the training data as the probability), where P(Xi=xi|Y=y)P(Xi=xi|Y=y) is the number of cases where Xi=xi∧Y=yXi=xi∧Y=y divided by the number of cases where Y=yY=y.

----------***----------

RGPVNOTES.IN

Thank you for using our services. Please support us so that we can improve further and help more people.
https://www.rgpvnotes.in/support-us

If you have questions or doubts, contact us on
WhatsApp at +91-8989595022 or by email at hey@rgpvnotes.in.

For frequent updates, you can follow us on
Instagram: https://www.instagram.com/rgpvnotes.in/.