CHAPTER-17

Decision Tree Induction

17.1 Introduction

17.2 Attribute selection measure

17.3 Tree Pruning

17.4 Extracting Classification Rules from Decision Trees

17.5 Bayesian Classification

17.6 Bayes Theorem

17.7 Naïve Bayesian Classification

17.8 Bayesian Belief Networks

17.9 Review Questions

17.10 References

17.Decision Tree Induction

17.1 Introduction

The basic algorithm for decision tree induction is a greedy algorithm that constructs decision trees in a top-down recursive divide-and-conquer manner. The basicstrategy is as follows.

The tree starts as a single node representing the training samples(step 1). If the samples are all of the same class, then the node becomes a leaf and is labeled with that class (steps 2 and 3). Otherwise, the algorithm uses an entropy-based measure known as information gain as a heuristic for selecting the attribute that will best separate the samples into individual classes (step 6). This attribute becomes the "test" or "decision" attribute at the node (step 7). In this version of the algorithm, all attributes are categorical, that is, discrete-valued. Continuous-valued attributes must be discretized.

A branch is created for each known value of the test attribute, and the samples are partitioned accordingly(steps 8-10).

The algorithm uses the same process recursively to form a decision tree for the samples at each partition. Once an attribute has occurred at a node, it need not be considered in any of the node's descendents.

The recursive partitioning stops only when any one of the following conditions is true;

(a) All samples for a given node belong to the same class (steps 2 and steps 3), or

(b)There are no remaining attributes on which the samples may be further partitioned (step 4). In this case, majority voting is employed (step 5). This involves converting the given node into a leaf and labeling it with the class in majority among samples. Alternatively, the class distribution of the node samples may be stored.

(c) There are no samples for the branch test-attribute=a; (step 11). In this case, a leaf is created with the majority class in samples(step 12).

17.2 Attribute selection measure

The information gain measure is used to select the test attribute at each node in the tree. Such a measure is referred to as attribute selection measure or a measure of the goodness of fit. The attribute with the highest information gain (or greatest entropy reduction) is chosen as the test attribute for the current node. This attribute minimizes the information needed to classify the samples in the resulting partitions and reflects the least randomness or “impurity” in these partitions. Such an information-theoritic approach minimizes the expected number of tests needed to classify an object and guarantees that a simple (but not necessarily the simplest) tree is found.

Let S be a set consisting of data samples. Suppose the class label attribute has m

distinct values defining m distinct classes , C_i . The expected information needed to classify a given sample is

given by

$$I(S_1,S_{2.....}S_m)= \sum P_i \log(P_i)$$

Where P_i is the probability that an arbitrary sample belongs to class C, and is estimated By S_i/S. Note that a log function to the base 2 is used since the information is encoded in bits.

Let attribute A have v distinct values, ($a_{1,} a_{2 .....}a_{.v).}$ Attribute A can be used to partition S into v subsets, (S_1, S_2,….Sv), where S_j contains those samples. Classification by that have value a_j , of A. If A were selected as the test attribute(i.e., the best attribute for splitting), then these subsets would correspond to the branches grown from the node containing the set S. Let S_j be the number of samples of class C, in

a subset S j . The entropy, or expected information based on the partitioning into subsets by A, is given by

$$E(A)=\sum (S_{ij} + S_{mj} /S)I(S ij,.....,S_{mj})$$

The encoding information that would be gained by branching on A is

$$\text{Gain}(A) = I(S_1, S_2, \ldots S_m) - E(A)$$

In other words, Gain (A) is the expected reduction in entropy caused by knowing the value of attribute A.

The algorithm computes the information gain of each attribute. The attribute with the highest information gain is chosen as the test attribute for this given set S. A node is created and labeled with the attribute, branches are created for each value of the attribute, and the samples are partitioned accordingly.

In summary, decision tree induction algorithms have been used for classification in a wide range of application domains. Such systems do not use domain knowledge. The learning and classification steps of decision tree induction are generally fast.

17.3 Tree Pruning

When a decision tree is built, many of the branches will reflect anomalies in the training data due to noise or outliers. Tree pruning methods address this problem of over fitting the data. Such methods typically use statistical measures to remove the least reliable branches, generally resulting in faster classification and an improvement in the

ability of the tree to correctly classify independent test data.

There are two common approaches to tree pruning. In the pre-pruning approach, a tree is “pruned” by halting its construction early (e.g., by deciding not to further split or partition the subset of training samples at a given node). Upon halting,the node becomes a leaf. The leaf may hold the most frequent class among the subset samples or the probability distribution of those examples.

When constructing a tree, measures such as statistical significance, information gain, and so on, can be used to assess the goodness of a split. If partitioning the samples at a node would result in a split that falls below a pre-specified threshold, then further partitioning of the given subset is halted. There are difficulties, however, in choosing an appropriate threshold. High thresholds could result in oversimplified trees, while low thresholds could result in very little simplification

The second approach, post-pruning, removes branches from a “fully grown” tree, a tree node is pruned by removing its branches. The cost complexity-pruning algorithm is an example of the post pruning approach. The lowest un-pruned node becomes a leaf and is labeled by the most frequent class among its former branches. For each non-leaf node in the tree, the algorithm calculates the expected error rate that would occur if the sub-tree at that node were pruned . Next, the expected error rate occurring if the node were not pruned is calculated using the error rates for each branch. If pruning the

node leads to a greater expected error rate, then the sub-tree is kept. Otherwise, it is pruned. After generating a set of progressively pruned trees, an independent test set is used to estimate the accuracy of each tree. The decision tree that minimizes the expected error rate is preferred.

Rather than pruning trees based on expected error rates, we can prune trees based on the number of bits required to encode them. The “best pruned tree” is the one that minimizes the number of encoding bits. This method adopts the Minimum Description length(MDL)principle, which follows the notion that the simplest solution is preferred. Unlike cost complexity pruning, it does not require an independent set of samples. Alternatively, pre-pruning and post-pruning may be interleaved for a combined approach. Post-pruning requires more computation than pre-pruning, yet generally leads to a more reliable tree.

17.4 Extracting Classification Rules from Decision Trees

“ Can I get classification rules out of my decision tree? If so, how?” The knowledge represented in decision trees can be extracted and represented in the form of classification IF-THEN rules. One rule is created for each path from the root to a leaf node. Each attribute-value pair along a given path forms a conjunction in the rule antecedent (“IF” part). The leaf node holds the class prediction, forming the rule

consequent (“THEN” part). The IF-THEN rules may be easier for humans to understand, particularly if the given tree is very large.

The rules extracted are C4.5, a later version of ID3 algorithm, uses the training samples to estimate the accuracy of each rule . Since this would result in an optimistic estimate of rule accuracy, C4.5 employs a pessimistic estimate . to compensate for the bias, Alternatively, a set of test samples independent from the training set can be used to estimate rule accuracy.

A rule can be “pruned” by removing any conditioning in its antecedent that does not improve the estimated accuracy of the rule. For each class, rules within a class may then be ranked according to their estimated accuracy. Since it is possible that a given test sample will not satisfy any rule antecedent, a default rule assigning the majority class is typically added to the resulting rule set.

17.5 Bayesian Classification

Bayesian classifiers are statistical classifiers. They can predict class membership probabilities, such as the probability that a given sample belongs to a particular class.

Bayesian classification is based on Bayes Theorem, described below. Studies comparing classification algorithms have found a simple Bayesian classifier known as the naïve Bayesian classifier to be comparable in performance with decision tree and

neural network classifiers. Bayesian classifiers have also exhibited high accuracy and speed when applied to large databases.

Naïve Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence . It is made to simplify the computations involved and , in this sense, is considered “naïve”. Bayesian belief networks are graphical models, which unlike naïve Bayesian classifiers, allow the representation of dependencies among subsets of attributes. Bayesian belief networks can also be used for classification.

The section below reviews basic probability notation and Bayes theorem. You will then learn naïve Bayesian classification later. Bayesian belief networks are described after that.

17.6 Bayes Theorem

Let X be a data sample whose class label is unknown. Let H be some hypothesis, such as that the data simple X belongs to a specified class C. For classification problems, we want to determine P(H/X), the probability that the hypothesis H holds given the observed data sample X.

P(H/X) is the posterior probability, or a posteriori probability, of H conditioned on

X. For example, suppose the world of data samples consists of fruits, described by their

color and shape. Suppose that X is red and round, and that H is the hypothesis that X is an apple. Then P(H/X) reflects our confidence that X is an apple given that we have seen that X is red and round. In contrast, P(H) is the prior probability, or a priori, probability, of H. For our example, this is the probability that any given data sample is An apple, regardless of how the data sample looks. The posterior probability, P(H/X), is based on more information (such as background knowledge)than the prior probability, P(H), which is independent of X

Similarly, P(X/H) is the posterior probability of X conditioned on H. That is ,it is

the probability that X is red and round given that we know that it is true X is an

apple. P(X) is the prior probability of X. Using for example, it is the probability that a

data sample from our set of fruits is red and round.

"How are these probabilities estimated ?" P(X), P(H), and P(X/H)may be estimated

from the given data, as we shall see below. Bayes theorem is useful in that it provides a

way of calculating the posterior probability, P(H/X), from P(H), P(X), and P(X/H).

Bayes theorem is

$$P(H/X)=(P(X/H)P(H))/P(X)$$

In the next section, you will learn how Bayes theorem is used in the naïve

Bayesian classifier.

17.7 Naïve Bayesian Classification

The naïve Bayesian classifier, or simple Bayesian classifier, works as follows:

1)Each data sample is represented by an n-dimensional feature vector, $X==x_1,x_2,.....x_n$, depicting n measurements made on the sample from n attributes, respectively, $A_1,A_2,....A_n$

2)Suppose that there are m classes, $C_1,C_2,.....C_m$. Given an unknown data sample, X(i.e., having no class label), the classifier will predict that X belongs to the class having the highest posterior probability, conditioned on X. That is , the naïve Bayesian classifier assigns an unknown sample X to the class C_i, if and only if we maximize $P(C_i/X)$. The class C, for which $P(C_i/X)$ is maximized is called the maximum posteriori hypothesis.

3) As P(X) is constant for all classes, only $P(X/C_j)/P(Cj)$ need be maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely, that is, $P(C_1)==P(C_2)==..=P(C_n)$, and we would therefore maximize P(X/C,). Otherwise, we maximize $P(X/C_i)P(C_i)$. Note that the class prior probabilities may be estimated by $P(C_i)==s_i/s$ is the number of training samples.

17.8 Bayesian Belief Networks

The naïve Bayesian classifier makes the assumption of class conditional independence, that is , given the class label of a sample, the values of the attributes are conditionally independent of the other. This assumption simplifies computation. When the assumption holds true, then the naïve Bayesian Classifier is the most accurate in comparison with all other classifiers. In practice, however, dependencies can exist between variables. Bayesian belief networks specify joint, conditional probability distributions. They allow class conditional independencies to be defined between subsets of variables. They provide a graphical model of causal relationships, on which learning can be performed. These networks are also known as belief networks, Bayesian networks, and probabilistic networks. For brevity, we will refer to them as belief networks.

A belief network is defined by two components. The first is a directed acyclic graph, where each node represents a random variable and each arc represents a probabilistic dependence. If an arc is drawn from a node Y to a node Z, then Y is a; parent or immediate predecessor of Z, and Z is a descendent of Y. Each variable is conditionally independent of its non-descendents in the graph, given its parents. The variables may be discrete or continuous-valued. They may correspond to actual attributes given in the data or to “hidden variables” believed to form a relationship (such as medical syndromes in the case of medical data).

17.9 Review Questions

.1 Explian about Attribute selection measure

2. Explain about Tree Pruning

3 how can we Extracting Classification Rules from Decision Trees

4 Expalin about Bayesian Classification

5 Explian about Bayes Theorem

17.10 References

[1]. Data Mining Techniques, Arun k pujari 1st Edition

[2] .Data warehousung,Data Mining and OLAP, Alex Berson ,smith.j. Stephen

[3].Data Mining Concepts and Techniques ,Jiawei Han and MichelineKamber

[4]Data Mining Introductory and Advanced topics, Margaret H Dunham PEA

[5] The Data Warehouse lifecycle toolkit , Ralph Kimball Wiley student Edition