Classification Using Decision Trees

1. Introduction

Data mining term is mainly used for the specific set of six activities namely Classification, Estimation, Prediction, Affinity grouping or Association rules, Clustering, Description and Visualization. The first three tasks - classification, estimation and prediction are all examples of directed data mining or supervised learning. Decision Tree (DT) is one of the most popular choices for learning from feature based examples. It has undergone a number of alterations to deal with the language, memory requirements and efficiency considerations.

A DT is a classification scheme which generates a tree and a set of rules, representing the model of different classes, from a given dataset. As per Hans and Kamber [HK01], DT is a flow chart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test and leaf nodes represent the classes or class distributions. The top most node in a tree is the root node. Figure 1 refers to DT induced for dataset in Table 1. We can easily derive the rules corresponding to the tree by traversing each leaf of the tree starting from the node. It may be noted that many different leaves of the tree may refer to the same class labels, but each leaf refers to a different rule. DTs are attractive in DM as they represent rules which can readily be expressed in natural language. The major strength of the DT methods are the following:

1. DT are able to generate understandable rules.
2. They are able to handle both numerical and categorical attributes.
3. They provide a clear indication of which fields are most important for prediction or classification.

Some of the weaknesses of DT are:

1. Some DT can only deal with binary valued target classes, others are able to assign records to an arbitrary number of classes, but are error prone when the number of training examples per class gets small. This can happen rather quickly in a tree with many levels and many branches per node.
2. The process of growing a DT is computationally expensive. At each node, each candidate splitting field is examined before its best split can be found.

Many classification models have been proposed in the literature. Classification trees also called Decision Trees (DT) are especially attractive in a data mining environment for several reasons. First, due to their intuitive representation, the resulting classification model is easy to assimilate by humans [BFOS84, SAM96]. Second, decision trees do not require any parameter setting from the user and thus are especially suited for exploratory knowledge discovery. Third, DT can be constructed relatively fast and the accuracy of DT is comparable or superior to other classification models.
2. Decision Tree Induction

DT induction is a well defined technique in statistics and machine learning. A common basic principle of all DT induction algorithms is outlined below.

2.1 Basic Principle (Hunt’s method)

All DT induction algorithms follow the basic principle, known as CLS (Concept Learning system), given by Hunt [HH61, HMS66]. Ross Quinlan attributes his work on trees (ID3, C4.5) as furtherance of Hunt’s ideas of CLS. A CLS tries to mimic the human process of learning a concept, starting with examples from two classes and then inducing a rule to distinguish the two classes based on other attributes. Let the training dataset be T with class-labels \{C1, C2, ..., Ci\}. The decision tree is built by repeatedly partitioning the training data using some splitting criterion till all the records in a partition belong to the same class. The steps to be followed are:

1. If T contains no cases (T is trivial), the decision tree for T is a leaf, but the class to be associated with the leaf must be determined from information other than T.
2. If T contains cases all belonging to a single class Cj (homogeneous), corresponding tree is a leaf identifying class Cj.
3. If T is not homogeneous, a test is chosen, based on a single attribute, that has one or more mutually exclusive outcomes \{O1, O2, ..., On\}. T is partitioned into subsets T1, T2, T3, ..., Tn, where Ti contains all those cases in T that have the outcome Oi of the chosen test.

The decision tree for T consists of a decision node identifying the test, and one branch for each possible outcome. The same tree building method is applied recursively to each subset of training cases.

2.2 Measures of the Diversity

The diversity index is a well developed topic with different names corresponding the various fields. To statistical biologist, it is Simpson diversity index. To cryptographers, it is one minus the repeat rate. To econometricians, it is the Gini index that is also used by the developers of the CART algorithm. Quinlan [Qui87, Qui93] used entropy as devised by Shannon in the information theory [Sha48]. A high index of diversity indicates that the set contains an even distribution of classes whereas a low index means that members of a single class predominate. The best splitter is the one that decreases the diversity of the record sets by the greatest amount. The three common diversity functions are discussed here. Let there be a dataset S (training data) of C outcomes. Let P(I) denotes the proportion of S belonging to a class I where I varies from 1 to C for the classification problem with C classes.

\[
Simple Diversity index = \text{Min}(p(I))
\]  

(1)

Entropy provides an information–theoretic approach to measure the goodness of a split. It measures the amount of information in an attribute.

\[
Entropy(S) = \sum_{I=1}^{C} (-p(I)\log_2 p(I))
\]  

(2)
Gain(S, A), the information gain of the example set S on an attribute A, defined as

\[
Gain(S, A) = Entropy(S) - \sum \left( \frac{|S_V|}{|S|} \right) * Entropy(S_V)
\]  

(3)

where \(\sum\) is over each value \(V\) of all the possible values of the attribute A,

\(SV = \text{subset of } S \text{ for which attribute } A \text{ has value } V, |SV| = \text{number of elements in } SV, \text{ and } |S| = \text{number of elements in } S.\)

The above notion of gain tends to favour the attributes that have a larger number of values. To compensate for this, Quinlan [Qui87] suggests using the gain ratio instead of gain, as formulated below.

\[
Gain Ratio(S, A) = \frac{Gain(S, A)}{SplitInfo(S, A)}
\]  

(4)

where \(SplitInfo(S, A)\) is the information due to the split of \(S\) on the basis of the value of the categorical attribute A. Thus \(SplitInfo(S, A)\) is entropy due to the partition of \(S\) induced by the value of the attribute A.

Gini index measures the diversity of population using the formula

\[
Gini\ Index = 1 - \sum (p(I))^2
\]  

(5)

where \(p(I)\) is the proportion of \(S\) belonging to class \(I\) and \(\sum\) is over \(C.\)

In this thesis, entropy, information gain and gain ratio (equations (2), (3) and (4)) have been used to determine the best split.

All of the above functions attain a maximum where the probabilities of the classes are equal and evaluate to zero when the set contains only a single class. Between the two extremes all these functions have slightly different shapes. As a result, they produce slightly different rankings of the proposed splits.

2.3 ID3 Algorithms

Many DT induction algorithms have been developed in the past over the years. These algorithms more or less follow the basic principle discussed above. Discussion of all these algorithms is beyond the scope of the present chapter. Quinlan [Qui93] introduced the Iterative Dichotomizer 3 (ID3) for constructing the decision tree from data. In ID3, each node corresponds to a splitting attribute and each branch is a possible value of that attribute. At each node, the splitting attribute is selected to be the most informative among the attributes not yet considered in the path from the root. The algorithm uses the criterion of information gain to determine the goodness of a split. The attribute with the greatest information gain is taken as the splitting attribute, and the dataset is split for all distinct values of the attribute.
3. Illustration

3.1 Description of the Data Set

Table 1: Weather Dataset

<table>
<thead>
<tr>
<th>ID</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>sunny</td>
<td>hot</td>
<td>high</td>
<td>weak</td>
<td>no</td>
</tr>
<tr>
<td>X2</td>
<td>sunny</td>
<td>hot</td>
<td>high</td>
<td>strong</td>
<td>no</td>
</tr>
<tr>
<td>X3</td>
<td>overcast</td>
<td>hot</td>
<td>high</td>
<td>weak</td>
<td>yes</td>
</tr>
<tr>
<td>X4</td>
<td>rain</td>
<td>mild</td>
<td>high</td>
<td>weak</td>
<td>yes</td>
</tr>
<tr>
<td>X5</td>
<td>rain</td>
<td>cool</td>
<td>normal</td>
<td>weak</td>
<td>yes</td>
</tr>
<tr>
<td>X6</td>
<td>rain</td>
<td>cool</td>
<td>normal</td>
<td>strong</td>
<td>no</td>
</tr>
<tr>
<td>X7</td>
<td>overcast</td>
<td>cool</td>
<td>normal</td>
<td>strong</td>
<td>yes</td>
</tr>
<tr>
<td>X8</td>
<td>sunny</td>
<td>mild</td>
<td>high</td>
<td>weak</td>
<td>no</td>
</tr>
<tr>
<td>X9</td>
<td>sunny</td>
<td>cool</td>
<td>normal</td>
<td>weak</td>
<td>yes</td>
</tr>
<tr>
<td>X10</td>
<td>rain</td>
<td>mild</td>
<td>normal</td>
<td>weak</td>
<td>yes</td>
</tr>
<tr>
<td>X11</td>
<td>sunny</td>
<td>mild</td>
<td>normal</td>
<td>strong</td>
<td>yes</td>
</tr>
<tr>
<td>X12</td>
<td>overcast</td>
<td>mild</td>
<td>high</td>
<td>strong</td>
<td>yes</td>
</tr>
<tr>
<td>X13</td>
<td>overcast</td>
<td>hot</td>
<td>normal</td>
<td>weak</td>
<td>yes</td>
</tr>
<tr>
<td>X14</td>
<td>rain</td>
<td>mild</td>
<td>high</td>
<td>strong</td>
<td>no</td>
</tr>
</tbody>
</table>

The Weather dataset (Table 1) is a small dataset and is entirely fictitious. It supposedly concerns the conditions that are suitable for playing some unspecified game. The condition attributes of the dataset are in \{Outlook, Temperature, Humidity, Wind\} and the decision attribute is Play to denote whether or not to play. In its simplest form, all four attributes have categorical values. Values of the attributes Outlook, Temperature, Humidity and Wind are in \{sunny, overcast, rainy\}, \{hot, mild, cool\}, \{high, normal\} and \{weak, strong\} respectively.

The values of all the four attributes produce $3\times3\times2\times2 = 36$ possible combinations out of which 14 are present as input.

3.2 Example 1

To induce DT using ID3 algorithm for Table 1, note that $S$ is a collection of 14 examples with 9 yes and 5 no examples. Using equation (2),

$$\text{Entropy (S)} = -(9/14) \times \log_2(9/14) - (5/14) \times \log_2(5/14) = 0.940$$

There are 4 conditional attributes in this table. In order to find the attribute that can serve as the root node in the decision tree to be induced, the information gain is calculated corresponding to all the 4 attributes. For attribute Wind, there are 8 occurrences of Wind = weak and 6 occurrences of Wind = strong. For Wind = weak, 6 out of the 8 examples are
yes and the remaining 2 are no. For Wind = strong, 3 out of the 6 examples are yes and the remaining 3 are no. Therefore, by using equation (3)

\[
\text{Gain (S, Wind) = Entropy (S) - (8/14) * Entropy (Sweak) - (6/14) * Entropy (Sstrong)}
\]

\[
= 0.940 - (8/14) (0.811) - (6/14) * 1.0
\]

\[
= 0.048
\]

\[
\text{Entropy (Sweak) = - (6/8) * log}_2 (6/8) - 2/8 * log}_2 (2/8) = 0.811
\]

\[
\text{Entropy (Sstrong) = - (3/6) * log}_2 (3/6) - (3/6) * log}_2 (3/6) =1.0
\]

Similarly Gain (S, Outlook) = 0.246

Gain (S, Temperature) = 0.029

Gain (S, Humidity) = 0.151

Outlook has the highest gain, therefore it is used as the decision attribute in the root node. Since Outlook has three possible values, the root node has three branches (sunny, overcast, rain). The next question is “which of the remaining attributes should be tested at the branch node sunny?”

Ssunny = \{X1, X2, X8, X9, X11\} i.e. there are 5 examples from Table 1 with Outlook = sunny. Thus using equation (3),

Gain (Ssunny, Humidity) = 0.970

Gain (Ssunny, Temperature) = 0.570

Gain (Ssunny, Wind) = 0.019

The above calculations show that the attribute Humidity shows the highest gain, therefore, it should be used as the next decision node for the branch sunny. This process is repeated until all data are classified perfectly or no attribute is left for the child nodes further down the tree. The final decision tree as obtained using ID3 is shown later in Figure 1.

The corresponding rules are:

1. If outlook=sunny and humidity=high then play=no
2. If outlook=sunny and humidity=normal then play=yes
3. If outlook=overcast then play=yes
4. If outlook=overcast and wind=weak then play=yes
5. If outlook=overcast and wind=strong then play=no
4. Summary

Notions and algorithms for DT induction are discussed with special reference to ID3 algorithm. A small hypothetical data set is used to illustrate the concept of decision tree using ID3 algorithm.

References


[HK01] Han, J., Kamber, M. Data Mining Concepts and Techniques, Morgan Kaufmann Publisher, 2001


