

Chapter 16

Decision Trees

Given a training dataset \mathcal{D} , with n points, in a d -dimensional space which may be composed of categorical and numeric attributes, the goal of classification is to build a model that associates the correct class c_i with each point \mathbf{x}_i . Decision trees are linear models that use axis parallel “splits” to recursively partition the labeled points, so that each region is as “pure” as possible in terms of the labels. A decision tree consists of internal nodes that represent decisions or splits, and leaf nodes that are labeled with a class.

Point	Age	Car	Risk
\mathbf{x}_1	25	Sports	L
\mathbf{x}_2	20	Vintage	H
\mathbf{x}_3	25	Sports	L
\mathbf{x}_4	45	SUV	H
\mathbf{x}_5	20	Sports	H
\mathbf{x}_6	25	SUV	H

Figure 16.1: Classification Example: Age is numeric and Car is categorical. Risk gives the class label for each point: high (H) or low (L).

Depending on the type of attribute, the decisions at the internal nodes are of different types. For numeric data, we consider decisions of the kind $A \leq v$, where A is the attribute and v is some value in the domain of A . For categorical data, we use a decision of the kind $A \in V$, where V is some subset of the values of the attribute. For example, consider the example dataset shown in Table 16.1. For Age, we might have a decision of the kind $Age \leq 22.5$, whereas for Car, we may have a decision of the kind $Car \in \{sports, SUV\}$.

Algorithm

The decision tree algorithm, given in Algorithm 16.1, is a recursive method that takes a given subset of the data \mathcal{D} , and evaluates all possible splits (or decisions), considering both numeric and categorical attributes. The best decision/split is chosen to partition the data into two subsets, on which the method is called recursively. The method stops when certain stopping conditions are met.

Stopping Criteria

A number of stopping conditions can be used to stop the recursive process:

```

DECISIONTREE ( $\mathcal{D}$ ):
1 if (stop condition met) then
2   └─ Choose majority class in  $\mathcal{D}$  as the leaf label
3 foreach (attribute A in  $\mathcal{D}$ ) do
4   └─ if (A is numeric) then
5     └─ Try all possible distinct splits:  $A \leq v$ 
6   └─ if (A is categorical) then
7     └─ Try all possible distinct splits:  $A \in V$ 
8 Choose the best split based on highest information gain
9 Partition  $\mathcal{D}$  into  $\mathcal{D}_L$  and  $\mathcal{D}_R$  based on the best split
10 DecisionTree( $\mathcal{D}_L$ )
11 DecisionTree( $\mathcal{D}_R$ )

```

Algorithm 16.1: Decision Tree Algorithm

1. If all the points in \mathcal{D} have the same label, then stop, since in this case the sample \mathcal{D} is already “pure” in terms of labels.
2. Stop if most of the points are already of the same class. This is a generalization of the first approach, with some error threshold (for better generalizability). For example, we may stop if at least a given fraction (say 90%) of the points in \mathcal{D} share the same label.
3. If $|\mathcal{D}| \leq s$, where s is some minimum leaf size threshold, then stop. This condition prevents over-fitting the model to the training set, since we avoid trying to model very small subsets of the data.

Selecting Splits

We now need an objective criteria for judging how good a split is. Intuitively, we want to select a split that gives the best separation or discrimination between the different labels.

One measure of the purity of a sample is measured using the concept of Entropy. Entropy, in general, measures the amount of disorder or uncertainty in a system. In the classification setting, higher entropy (i.e., more disorder) corresponds to a sample that has a mixed collection of labels. Lower entropy corresponds to a case where we have mostly pure partitions. In information theory, the entropy of a sample \mathcal{D} is defined as follows:

$$H(\mathcal{D}) = - \sum_{i=1}^k P(c_i|\mathcal{D}) \log_2 P(c_i|\mathcal{D}) \quad (16.1)$$

where $P(c_i|\mathcal{D})$ is the probability of a data point in \mathcal{D} being labeled with class c_i , and k is the number of classes. $P(c_i|\mathcal{D})$ can be estimated directly from the data as follows:

$$P(c_i|\mathcal{D}) = \frac{|\{\mathbf{x}_j \in \mathcal{D} \mid \mathbf{x}_j \text{ has label } y_j = c_i\}|}{|\mathcal{D}|} \quad (16.2)$$

We can also define the weighted entropy of a decision/split as follows:

$$H(\mathcal{D}_L, \mathcal{D}_R) = \frac{|\mathcal{D}_L|}{|\mathcal{D}|} H(\mathcal{D}_L) + \frac{|\mathcal{D}_R|}{|\mathcal{D}|} H(\mathcal{D}_R) \quad (16.3)$$

where D has been partitioned into \mathcal{D}_L and \mathcal{D}_R due to some split decision. Finally, we can define the *information gain* for a given split as

$$\text{Gain}(\mathcal{D}, \mathcal{D}_L, \mathcal{D}_R) = H(\mathcal{D}) - H(\mathcal{D}_L, \mathcal{D}_R) \quad (16.4)$$

The higher the information gain, the more the reduction in entropy due to the split. We score the different splits based on their information gain and choose the one that gives the highest gain.

Instead of entropy we can also use other measures that give an indication of how pure or mixed a given sample is. For example, the *Gini Index* is defined as follows:

$$G(\mathcal{D}) = 1 - \sum_{i=1}^k P(c_i|\mathcal{D})^2 \quad (16.5)$$

As before, we can compute the weighted gini index of a split as follows:

$$G(\mathcal{D}_L, \mathcal{D}_R) = \frac{|\mathcal{D}_L|}{|\mathcal{D}|} G(\mathcal{D}_L) + \frac{|\mathcal{D}_R|}{|\mathcal{D}|} G(\mathcal{D}_R) \quad (16.6)$$

Further, we can compute the information gain in terms of the Gini index:

$$\text{Gain}(\mathcal{D}, \mathcal{D}_L, \mathcal{D}_R) = G(\mathcal{D}) - G(\mathcal{D}_L, \mathcal{D}_R) \quad (16.7)$$

Other measures can also be used instead of information gain to evaluate the splits. For example, the Classification And Regression Trees (CART) measure is given as:

$$\text{CART}(\mathcal{D}_L, \mathcal{D}_R) = 2 \left(\frac{|\mathcal{D}_L|}{|\mathcal{D}|} \right) \left(\frac{|\mathcal{D}_R|}{|\mathcal{D}|} \right) \sum_{i=1}^k |P(c_i|\mathcal{D}_L) - P(c_i|\mathcal{D}_R)| \quad (16.8)$$

The CART measure prefers a split that tries to maximize the difference between the class probability distribution function in the two partitions.

Example

Let us consider how a complete decision tree is induced for our example dataset in Table 16.1. In the complete dataset we have $P(H|\mathcal{D}) = P_H = \frac{2}{3}$ and $P(L|\mathcal{D}) = P_L = \frac{1}{3}$. Thus the entropy of \mathcal{D} is

$$H(\mathcal{D}) = - \left(\frac{2}{3} \log_2 \frac{2}{3} + \frac{1}{3} \log_2 \frac{1}{3} \right) = -(-0.387 - 0.528) = 0.915$$

At the root of the decision tree, we consider all possible splits on Age and Car. For Age, the possible distinct splits are to consider are $\text{Age} \leq 22.5$ and $\text{Age} \leq 35$, which were chosen to be the mid-points between the distinct values, namely, 20, 25, and 45, that we observe for Age.

1. For $\text{Age} \leq 22.5$, \mathcal{D}_L is comprised of points \mathbf{x}_2 and \mathbf{x}_5 , whereas \mathcal{D}_R is comprised of the remaining points: \mathbf{x}_1 , \mathbf{x}_3 , \mathbf{x}_4 , and \mathbf{x}_6 . For \mathcal{D}_L , this yields $P_L = 0$ and $P_H = 1$, whereas for \mathcal{D}_R we have $P_L = \frac{2}{4}$ and $P_H = \frac{2}{4}$. The weighted entropy is then

$$\begin{aligned} H(\mathcal{D}_L, \mathcal{D}_R) &= \frac{2}{6} H(\mathcal{D}_L) + \frac{4}{6} H(\mathcal{D}_R) \\ &= -\frac{2}{6}(0) - \frac{4}{6} \left(\frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{2} \log_2 \frac{1}{2} \right) \\ &= -\frac{2}{3} \log_2 \frac{1}{2} = 0.67 \end{aligned}$$

This yields an information gain of $0.915 - 0.67 = 0.245$.

2. In a similar manner we can compute the weighted entropy for $Age \leq 35$. For $D_R = \{\mathbf{x}_4\}$ and D_L has the remaining points, so that $H(D_L) = \frac{2}{5} \log_2 \frac{2}{5} + \frac{3}{5} \log_2 \frac{3}{5} = 0.971$ and $H(D_R) = 0$. The split entropy is then $H(D_L, D_R) = \frac{5}{6}(0.971) = 0.809$, the information gain is: $0.915 - 0.809 = 0.106$, which is not as high as for $Age \leq 22.5$.

Next, we evaluate all possible splits for Car. Note that categorical data, in general, yields $\frac{v-1}{2}$ possible splits, where v is the set of possible values for the attribute. This can be reduced to $O(v)$ by using a greedy split selection approach. For Car the possible values are $\{Sports(S), Vintage(V), SUV(U)\}$, which yields the following three distinct splits:

$Car \in$	$Car \notin$
$\{S\}$	$\{V, U\}$
$\{V\}$	$\{S, U\}$
$\{U\}$	$\{S, V\}$

Note that the split $Car \in \{V, U\}$ is essentially the same as the split $Car \in \{S\}$, the only difference being that the decision has been “reversed”. It is therefore not a distinct split, and we do not consider such splits. Next we evaluate the three categorical splits as follows:

1. For the split $Car \in \{S\}$, $\mathcal{D}_L = \{\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_5\}$, and $\mathcal{D}_R = \{\mathbf{x}_2, \mathbf{x}_4, \mathbf{x}_6\}$. For \mathcal{D}_L , this yields $P_L = \frac{2}{3}$ and $P_H = \frac{1}{3}$, and for \mathcal{D}_R , $P_L = 0$ and $P_H = 1$. The weighted entropy of the split is then

$$\begin{aligned}
 H(\mathcal{D}_L, \mathcal{D}_R) &= \frac{3}{6}H(\mathcal{D}_L) + \frac{3}{6}H(\mathcal{D}_R) \\
 &= -\frac{3}{6} \left(\frac{1}{3} \log_2 \frac{1}{3} + \frac{2}{3} \log_2 \frac{2}{3} \right) - \frac{3}{6}(0) \\
 &= 0.194
 \end{aligned}$$

This yields an information gain of $0.915 - 0.194 = 0.721$.

2. For $Car \in \{V\}$, we get the same information gain as for $Age \leq 35$, i.e., 0.106, and for $Car \in \{U\}$, the gain is the same as for $Age \leq 22.5$, i.e., 0.245.

Among all the possible split points for both Age and Car, the one with the highest information gain is $Car \in \{S\}$, which is chosen as the best split decision at the root of the decision tree, as shown in Figure 16.2. We therefore make this split and recursively call the decision tree algorithm on each new subset: $\mathcal{D}_L = \{\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_5\}$ and $\mathcal{D}_R = \{\mathbf{x}_2, \mathbf{x}_4, \mathbf{x}_6\}$.

Notice that for \mathcal{D}_R all points are already labeled as high risk (H). Since the partition is already pure, we make it a leaf node, labeled as H . On the other hand, \mathcal{D}_L is not completely pure, so we consider partitioning it further. Since all points in \mathcal{D}_L have $Car \in \{S\}$, we cannot use Car to further distinguish the points. Further, for Age, $Age \leq 20$ is the only possible split to consider. Note that the entropy of D_L is given as

$$H(D_L) = - \left(\frac{2}{3} \log_2 \frac{2}{3} + \frac{1}{3} \log_2 \frac{1}{3} \right) = 0.387$$

For $Age \leq 22.5$, $\mathcal{D}_{LL} = \{\mathbf{x}_1, \mathbf{x}_3\}$, whereas $\mathcal{D}_{LR} = \{\mathbf{x}_5\}$. For \mathcal{D}_{LL} , we get $P_L = 1$ and $P_H = 0$, and for \mathcal{D}_{LR} , we get $P_L = 0$ and $P_H = 1$. The weighted entropy is then

$$H(\mathcal{D}_{LL} = \{1, 3\}, \mathcal{D}_{LR} = \{5\}) = \frac{2}{3}H(\mathcal{D}_{LL}) + \frac{1}{3}H(\mathcal{D}_{LR}) = -\frac{2}{3}(0) - \frac{1}{3}(0) = 0$$

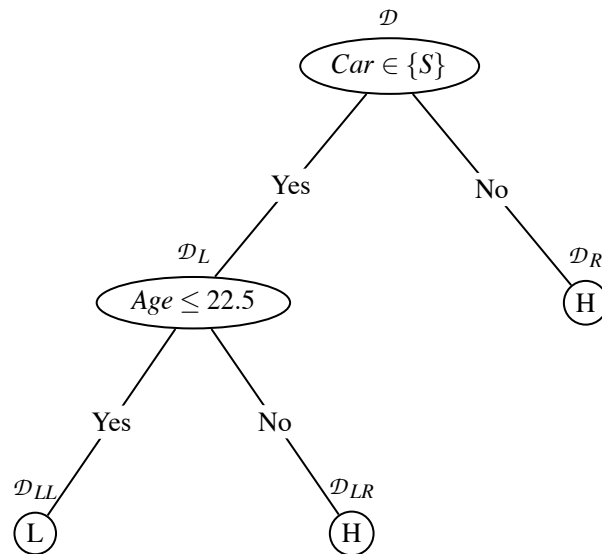


Figure 16.2: Induced Decision Tree

Thus the information gain is $0.387 - 0 = 0.387$. In this example, this is the only possible split decision. After \mathcal{D}_L is split, we obtain the two new leaves \mathcal{D}_{LL} , which is labeled as low-risk (L), and \mathcal{D}_{LR} , which is labeled as high-risk (H). The full decision tree is shown in Figure 16.2.

One of the advantages of decision trees, is that each path from the root to a leaf can be written as a rule. For our example tree above, we obtain the following three rules:

1. R_1 : if $Car \in \{S\}$ and $Age \leq 22.5$, then $Risk = L$
2. R_2 : if $Car \in \{S\}$ and $Age > 22.5$, then $Risk = H$
3. R_3 : if $Car \notin \{S\}$, then $Risk = H$

This is one of the strengths of decision trees, namely the ability to aid understanding of the model via simple rules presented to the user.

Once a decision tree model has been built, it can be used to classify new points. For example, if a new point has $Age = 27$, and $Car = Vintage$, then we can classify the point by applying a set of decisions starting at the root. First we check whether $Car \in \{S\}$. Since this test will be false, we go to the right branch, and since it is a leaf, we predict the class to be H . Similarly if another point has $Age = 20$ and $Car \in \{S\}$. Since we now have $Car \in \{S\}$, we follow the left branch, and test for $Age \leq 22.5$. Since this test is also true, we go to the left leaf and predict the class to be L .