DECISION TREES

One of the most widely used and effective learning method to approximate discrete valued target functions.

1 Algorithm

1. Use a heuristic to select an attribute for the root node.
2. Create branches for each possible value of the attribute.
3. Split the instances having a particular attribute value to the corresponding branch's node.
4. Repeat recursively for each branch and stop recursion for a branch if all instances have the same class.

2 Attribute Selection Heuristic

Prefer attribute which splits the data into purest successor nodes. Entropy is such a measure of disorder. For a set of items belonging to two classes positive and negative,

$$\mathbf{E}(p,n) = -\frac{p}{p+n} \times log_2(\frac{p}{p+n}) - \frac{n}{p+n} \times log_2(\frac{n}{p+n})$$

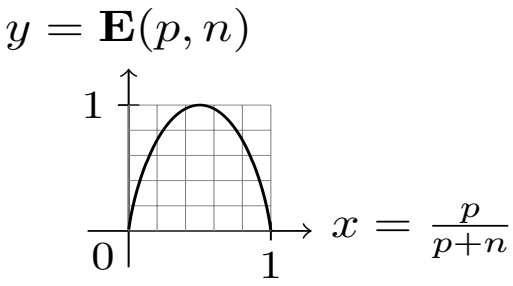


Figure 1: Entropy in two class distribution

Generally for n classes,

$$\mathbf{E}(S) = \mathbf{E}(p_1,\ p2,\ ...,\ p_n) = -\sum_{i=1}^{n} p_i \times log_2(p_i)$$

Entropy only calculates the quality of a single split sub-set. To calculate the quality of the entire split over the attribute A,

$$\mathbf{Gain}(S, A) = \mathbf{E}(S) - \sum_{i} \frac{|S_i|}{|S|} \times \mathbf{E}(S_i)$$

The attribute that maximizes the information gain is selected.

3 Example

Table 1: Training data

Sr.	**Outlook**	**Temperature**	**Humidity**	**Windy**	**PlayGolf**
1	Sunny	Hot	High	False	No
2	Sunny	Hot	High	True	No
3	Overcast	Hot	High	False	Yes
4	Rainy	Cool	Normal	False	Yes
5	Overcast	Cool	Normal	True	Yes
6	Sunny	Mild	High	False	No
7	Sunny	Cool	Normal	False	Yes
8	Rainy	Mild	Normal	False	Yes
9	Sunny	Mild	Normal	True	Yes
10	Overcast	Mild	High	True	Yes
11	Overcast	Hot	Normal	False	Yes
12	Rainy	Mild	High	True	No
13	Rainy	Cool	Normal	True	No
14	Rainy	Mild	High	False	Yes

All four attributes - **Outlook**, **Temperature**, **Humidity** and **Windy** are candidates for evaluation to choose the best splitting attribute.

$$\begin{aligned}\mathbf{E}(S) &= -\frac{9}{14} \times log2(\frac{9}{14}) - \frac{5}{14} \times log2(\frac{5}{14}) \\ &= 0.940 \textbf{ bits}\end{aligned}$$

Outlook Gain

Outlook	Yes	No	Split Entropy	Split Weight
Rainy	3	2	$-\frac{3}{5} \times log_2(\frac{3}{5}) - \frac{2}{5} \times log_2(\frac{2}{5}) = 0.971$	$\frac{5}{14}$
Overcast	4	0	$-\frac{4}{4} \times log_2(\frac{4}{4}) - \frac{0}{4} \times log_2(\frac{0}{4}) = 0$	$\frac{4}{14}$
Sunny	2	3	$-\frac{2}{5} \times log_2(\frac{2}{5}) - \frac{3}{5} \times log_2(\frac{3}{5}) = 0.971$	$\frac{5}{14}$

$$\begin{aligned}\textbf{Gain}(S, Outlook) &= \mathbf{E}(S) - (\frac{5}{14} \times 0.971 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.971) \\ &= 0.940 - 0.694 \\ &= 0.246 \textbf{ bits}\end{aligned}$$

Temperature Gain

Temperature	Yes	No	Split Entropy	Split Weight
Cool	3	1	$-\frac{3}{4} \times log_2(\frac{3}{4}) - \frac{1}{4} \times log_2(\frac{1}{4}) = 0.811$	$\frac{4}{14}$
Hot	2	2	$-\frac{2}{4} \times log_2(\frac{2}{4}) - \frac{2}{4} \times log_2(\frac{2}{4}) = 1$	$\frac{4}{14}$
Mild	4	2	$-\frac{4}{6} \times log_2(\frac{4}{6}) - \frac{2}{6} \times log_2(\frac{2}{6}) = 0.918$	$\frac{6}{14}$

$$\begin{aligned}\textbf{Gain}(S, Temperature) &= \mathbf{E}(S) - (\frac{4}{14} \times 0.811 + \frac{4}{14} \times 1 + \frac{6}{14} \times 0.918) \\ &= 0.940 - 0.911 \\ &= 0.029 \textbf{ bits}\end{aligned}$$

Humidity Gain

Humidity	Yes	No	Split Entropy	Split Weight
High	3	4	$-\frac{3}{7} \times log_2(\frac{3}{7}) - \frac{4}{7} \times log_2(\frac{4}{7}) = 0.985$	$\frac{7}{14}$
Normal	6	1	$-\frac{6}{7} \times log_2(\frac{6}{7}) - \frac{1}{7} \times log_2(\frac{1}{7}) = 0.592$	$\frac{7}{14}$

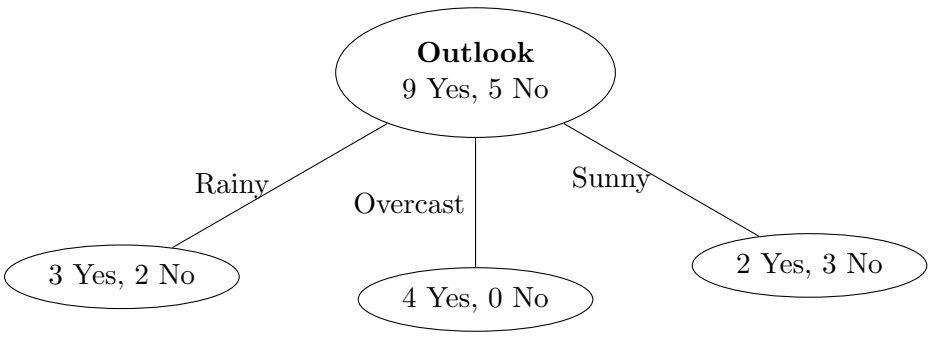
$$\begin{aligned}\textbf{Gain}(S, Humidity) &= \textbf{E}(S) - (\frac{7}{14} \times 0.985 + \frac{7}{14} \times 0.592) \\ &= 0.940 - 0.789 \\ &= 0.151 \textbf{ bits}\end{aligned}$$

Windy Gain

Windy	Yes	No	Split Entropy	Split Weight
False	6	2	$-\frac{6}{8} \times log_2(\frac{6}{8}) - \frac{2}{8} \times log_2(\frac{2}{8}) = 0.811$	$\frac{8}{14}$
True	3	3	$-\frac{3}{6} \times log_2(\frac{3}{6}) - \frac{3}{6} \times log_2(\frac{3}{6}) = 1$	$\frac{6}{14}$

$$\begin{aligned}\textbf{Gain}(S, Windy) &= \textbf{E}(S) - (\frac{8}{14} \times 0.811 + \frac{6}{14} \times 1) \\ &= 0.940 - 0.892 \\ &= 0.048 \textbf{ bits}\end{aligned}$$

Since $\textbf{Gain}(S, Outlook)$ represents the highest information gain, the root is split by **Outlook** as visualized below.



4 Gain Ratio

Information gain favours attributes with many values over those with few values. If an attribute is highly branching, the splits generally tend to be small and heterogeneous causing the split entropies to be close to zero and thus resulting in very high information gain.

To avoid this difficulty, information gain is divided by a quantity called intrinsic information that is sensitive to highly branching attribues to give the gain ratio which is maximized instead.

$$\textbf{IntrinsicInfo}(S, A) = \sum_i \frac{|S_i|}{|S|} \times log_2(\frac{|S_i|}{|S|})$$

$$\textbf{GainRatio}(S, A) = \frac{\textbf{Gain}(S, A)}{\textbf{IntrinsicInfo}(S, A)}$$

5 Refinements

5.1 Continuous Attributes

For a continuous attribute **A**, define a boolean attribute $\mathbf{A}_c$ which is $True$ if $\mathbf{A} < c$ and $False$ otherwise. To pick the optimal value of the threshold c that maximizes the information gain, values of **A** are sorted and candidate thresholds are generated midway of each consecutive pair.

Temperature	40	48	60	72	80	90
PlayGolf	No	No	Yes	Yes	Yes	No

Candidate thresholds and their information gain is computed as below. It is clear from the table below that **Temperature** = 54 is the optimal value for the threshold c.

Split	**Yes**	**No**	**Gain**
< 44	0	1	0.191
$>= 44$	3	2	
< 54	0	2	**0.459**
$>= 54$	3	1	
< 66	1	2	0.082
$>= 66$	2	1	
< 76	2	2	0
$>= 76$	1	1	
< 85	3	2	0.191
$>= 85$	0	1	

5.2 Overfitting

A hypothesis is said to overfit the training data if it performs very accurately on the training data but fares poorly on unseen test data. Two major approaches to avoid overfitting in decision tree learning are as follows:

- Stop growing the tree before it starts to overfit.
- Grow tree fully but post prune it.

5.2.1 Reduced Error Pruning

Data is divided into training and validation sets. The training set is used to form the decision tree. Now, each decision node is considered for pruning.

Pruning a decision tree consists of removing the subtree rooted at that node, making it a leaf node, and assigning it the most common classification of the training examples affiliated with that node. Nodes are removed only if the resulting tree performs no worse than the original tree over the validation set.

Nodes are pruned iteratively, always choosing the node whose removal most increases the accuracy over the validation set. Pruning continues until further removal of nodes is harmful.

In case of limited data, the prospects of this pruning scheme are relatively dim.

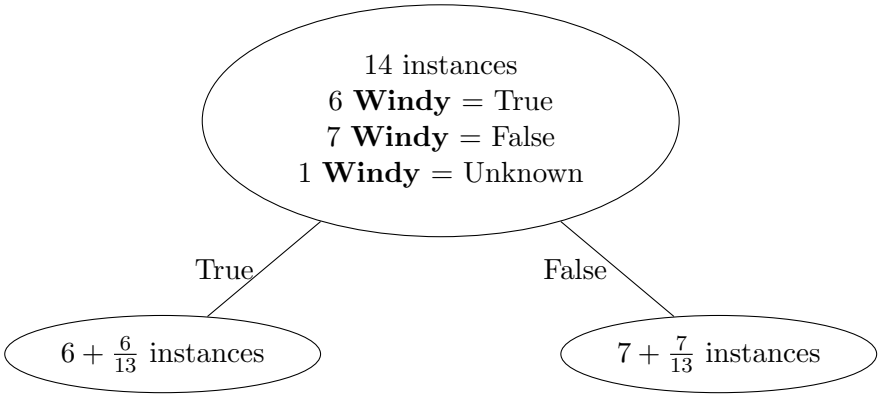
5.2.2 Rule Post Pruning

1. Infer the decision tree from the training set allowing overfitting.
2. Convert tree into equivalent set of rules by creating one rule per one path from root to a leaf node.
3. Prune each rule by removing any preconditions that results in improving its estimated accuracy on the validation set. If a rule is of the form

   $$IF\ (\textbf{Outlook} = Sunny) \wedge (\textbf{Humidity} = High)$$
   $$THEN\ \textbf{PlayGolf} = No$$

   then (**Outlook** $= Sunny$) and (**Humidity** $= High$) are preconditions.
4. Sort the pruned rules by their estimated accuracy, and consider them in this sequence when classifying subsequent instances.

5.3 Missing Attributes Values



5.4 Attribute With Differing Costs

Instead of maximizing information gain ratio, a quantity like $\frac{f(\mathbf{Gain}(S,A))}{g(\mathbf{Cost}(S,A))}$ is maximized where f and g are chosen on a case by case basis.

6 Comments

- Decision trees provide a practical method for concept learning and for learning other discrete-valued functions.
- A complete hyphothesis space is searched because the space of decision trees can represent any dicsrete-valued function defined over discrete valued instances. It therby avoids major difficulty associated with approaches that consider only restricted sets of hypothesis.