ARIN - Statistical Machine Learning
ID3 Decision Tree Learning - Provisional Notes

shi at ebornet dot com

May 9, 2016

1 Problem Scenarios

For a decision tree problem, there are two possible scenarios:

1. If there is a 'conclusion column' in the table, then that column's data will be our starting point – we do not need to choose a random variable with the highest initial entropy.
2. If there is no 'conclusion column', we must calculate the entropy of each random variable, and choose the variable with the highest entropy to establish the baseline.

For convenience, this note will assume a binary sample space (true or false), it is trivial to extend this summary of method to problems with non-binary sample space by calculating more branches for each node.

2 Compute Root Entropy

In Scenario 1, we only need to compute for the 'conclusion column' as stated by the question; while in Scenario 2, we need to compute for all possible random variables, and choose the random variable with the highest entropy for root entropy.

To compute the root entropy:

$P(S_{initial}) = (\frac{true\_count}{total\_count}, \frac{false\_count}{total\_count})$

Entropy $H(S_{initial}) = -\frac{true\_count}{total\_count} log_2 \frac{true\_count}{total\_count} - \frac{false\_count}{total\_count} log_2 \frac{false\_count}{total\_count}$

where $true\_count$ is the number of true results and $false\_count$ is the number of false results for the random variable, out of $total\_count$ which is the total number of samples.

3 Compute A Child Node

For each parent node (which may be the root node, depends on position in the tree), we need to compute entropy and information gain for each of its child (other random variables), until we have no more random variables.

For each child of a parent node:

$$Q_{child\_true} = \frac{child\_true\_count}{parent\_total}$$
$$Q_{child\_false} = \frac{child\_false\_count}{parent\_total}$$

where $child\_true\_count$ is the total number occurrences when the child random variable is true, regardless of the true or false value of its branch from the parent node. The same applies to $child\_false\_count$ when when the child random variable is false.

$$parent\_total = child\_true\_count + child\_false\_count$$

Then we have vectors $P(child\_true)$ and $P(child\_false)$ for both possibilities:

$$P(S_{child\_true}) = (\frac{child\_true\_parent\_true}{child\_true\_count}, \frac{child\_true\_parent\_false}{child\_true\_count})$$
$$P(S_{child\_false}) = (\frac{child\_false\_parent\_true}{child\_false\_count}, \frac{child\_false\_parent\_false}{child\_false\_count})$$

For convenience, we shall abbreviate the above terms as:

$$P(S_{child\_true}) = (\frac{CTPT}{CTC}, \frac{CTPF}{CTC})$$
$$P(S_{child\_false}) = (\frac{CFPT}{CFC}, \frac{CFPF}{CFC})$$

Now we can calculate the entropy for both possibilities of the child random variable:

$$\text{Entropy } H(S_{child\_true}) = -\frac{CTPT}{CTC} log_2 \frac{CTPT}{CTC} - \frac{CTPF}{CTC} log_2 \frac{CTPF}{CTC}$$
$$\text{Entropy } H(S_{child\_false}) = -\frac{CFPT}{CTC} log_2 \frac{CFPT}{CTC} - \frac{CFPF}{CTC} log_2 \frac{CFPF}{CTC}$$

With entropy for both possibilities, we are able to calculate the split:

$$H(Split) = \frac{child\_true\_count}{parent\_total} \times H(S_{child\_true}) + \frac{child\_false\_count}{parent\_total} \times H(S_{child\_false})$$

Finally, the information gain of the child node can be calculated from its parent's entropy and its split:

$$InformationGain = H(S_{parent}) - H(Split)$$

This process is repeated for all child nodes (remaining random variables) of the parent node. We choose the random variable with the highest $InformationGain$ as the next best node in decision making. The next best node becomes the new parent node and this process continues, until exhaustion (as stated at the beginning of this section) or when both entropy values are 0 – the decision can then be made based on traversing the tree from the root node to that child node, and no more information is necessary.