CS 217: Artificial Intelligence and Machine Learning **Jan-Apr 2024**

Lecture 10: Decision Trees

Lecturer: Swaprava Nath *Scribe(s): Tanmay Gejapati; Rijul Bhat; Yashwanth VVS; MVS Mukhesh*

Disclaimer: *These notes aggregate content from several texts and have not been subjected to the usual scrutiny deserved by formal publications. If you find errors, please bring to the notice of the Instructor.*

10.1 Example 1

Exam Result	Online Courses	Background	Mock Tests
P	Y	Math	N
F	N	M	Y
F	N	M	Y
P	Y	CS	N
⋮	⋮	⋮	⋮

Goal: Create a classifier on whether a given student will pass or not depending on the data

By splitting along all features at first, splitting with respect to background seems most intuitive as some backgrounds have all passing or all failing students. Similarly after splitting on background we can prefer splitting on mock tests over online courses as online courses are not providing any new info.

However, we need a more systematic way of splitting in order to make effective decision trees .

10.2 Example 2

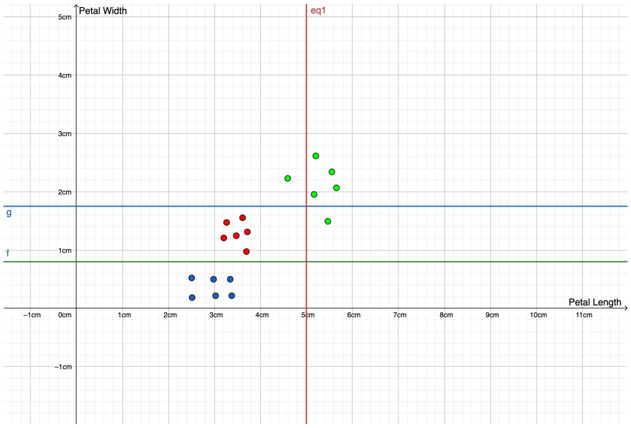
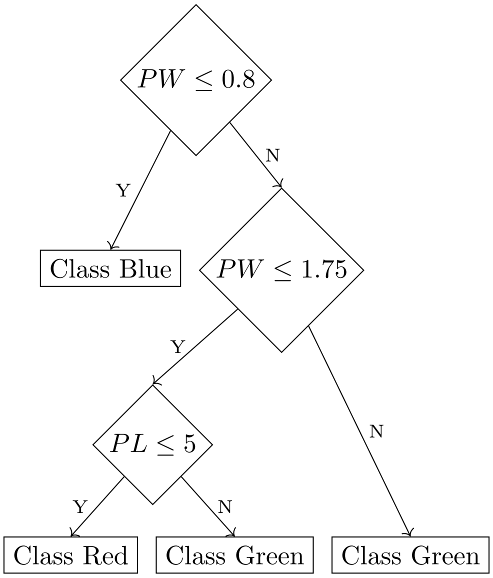


Figure 10.1: Iris Dataset

We can observe graphically that the data is regional,

- $PW < 0.8$ will classify all the points of blue colour
- $0.8 < PW < 1.75$ and $PL < 5$ will classify all the points of red colour
- Rest of the points will get classified as green

Hence, in this case it is easy to build the decision tree based on their coordinates.



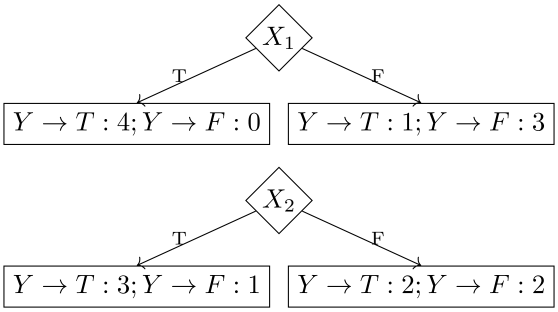
Now we will try to generalise the construction of a decision tree by subseqeuntly answering the following questions:

Q1 How to build the tree?
Q2 Where to stop?

10.3 Example 3

Consider this example where X_1, X_2 and Y are binary random variables and this is an observation table:

X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F
F	T	F
F	F	F


X_1
T
F
$Y \to T : 4; Y \to F : 0$
$Y \to T : 1; Y \to F : 3$
X_2
T
F
$Y \to T : 3; Y \to F : 1$
$Y \to T : 2; Y \to F : 2$

If we divide with respect to X_1 or X_2, what can we say about the classification and with what "certainty"? This measure of certainty is linked to the concept of **Entropy**.

10.4 Entropy

It's the measure of randomness of a random variable.
Let X be a categorical random variable and let $p(x)$ be a short-hand for $P(X = x) \; \forall \, x \in X$
Then entropy of X is defined by:

$$H(X) = -\sum_{x \in X} p(x) \cdot \log_{|X|} (p(x))$$

For a binary random variable, $X = \{0, 1\}$, the entropy is:

$$H(X) = -p(0) \cdot \log_2 p(0) - p(1) \cdot \log_2 p(1)$$

10.4.1 Observations of Entropy Function

1. $H(X) \geq 0$
   Since $0 \leq p(x) \leq 1$, the log part of each term is always negative, hence entropy is always positive.
   The equality is attained for random variables which are certain, so probability for each category except the one which is certain (say x^*) is 0, so all those terms cancel out (assuming $0 \cdot \log 0 = 0$) and $\log p(x^*) = 0$. Hence total entropy is 0.
2. $H(X) \leq 1$
   This can be proved using Jensen's inequality which states that, for any function $f(x)$ which is convex in R_X, and $\mathbb{E}[f(X)]$ and $f(\mathbb{E}[X])$ are finite, then

   $$\mathbb{E}[f(X)] = f(\mathbb{E}[X])$$

   Notice that $H(X)$ is a concave function, so the inequality just reverses.
   Proving this is left as an exercise to the reader.

10.4.2 Conditional Entropy

Conditional entropy is almost like saying if I observe a random variable Y, and it's a proxy for another random variable X, then before observing X we already know something about it.
Formally, it's defined as:

$$H(X|Y) = -\sum_{y} \sum_{x} p(x, y) \cdot \log p(x|y)$$

where, $p(x|y) = P(X = x|Y = y)$

Some observations:

- If $X \perp\!\!\!\perp Y$ (notation for denoting X and Y are independent random variables), then $H(X|Y)$ is just $H(X)$ which is intuitive because knowing about Y doesn't provide us any information about X.
- For a specific y,

$$H(X|Y = y) = -\sum_x p(x|y) \cdot \log p(x|y)$$

10.4.3 Mutual Information

It's the measure of how much information is gained by observing X given that you've already observed Y. Formally, it's written as:

$$\mathrm{I}(X;Y) = H(X) - H(X|Y)$$

Note that mutual information is symmetric, that is:

$$\begin{aligned} \mathrm{I}(X;Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \end{aligned} \tag{10.1}$$

10.5 Algorithm for decison tree building

Keep finding the feature that yields the maximum information gain (minimum conditional entropy) until stopping criteria is met.

Remark: **Gini index** can be used as another metric for building the decision tree.
Gini index: probability for a random instance being misclassified when chosen randomly

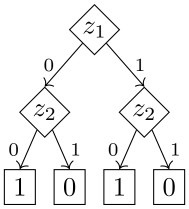
Where to stop?

- **Base case 1:** Reaching nodes with atomic distributions i.e., $H(Y|node) = 0$.
  After reaching a node, if there is no randomness and we know that all data points reaching this node belong to a unique class then it makes sense to stop there as the data has been classified.
- **Base case 2:** Information gain is same for all remaining variables.
  It is not always a good base case as it can lead to no splitting at all. This is illustrated in example 4.

10.6 Example 4

Z_1	Z_2	Y
0	0	0
0	1	1
1	0	1
0	0	0

Observe that,
$H(Y) = 1$
$H(Y|Z_1) = 1$
$H(Y|Z_2) = 1$
$I(Y; Z_1) = H(Y) - H(Y|Z_1) = 0$
$I(Y; Z_2) = H(Y) - H(Y|Z_2) = 0$
Following base case 2, there will be no splitting as both Z_1 and Z_2 are providing equal information gain.
Following base case 1, we get the following decision tree



10.7 Dealing with Over-fitting in Decision Trees

A deep decision tree suffers from getting too specific to the training data and learns the noise as well, which makes it bad at generalizing the training data, in other words, it's over-fitting.
These are some of the methods used to reduce over-fitting:

- **Pre-pruning or Early stopping**: Error over a validation dataset is maintained and the model increases its depth until the error over validation set is decreasing, once it starts increasing, we stop increasing its depth.

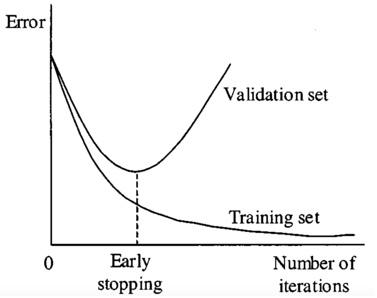


Figure 10.2: Early Stopping

- **Post-pruning**: The tree is allowed to increase in complexity and after training, the depth is reduced.
- **Ensemble method**: Multiple decision trees are prepared where each of them is slightly different and a mean of the output of all these trees is used as a final output.