(7th March, week 2)

Improving that accuracy function

We need to improve the accuracy classification function we introduced last lecture. So some problems are;

- for leaves there is no issue (if a final split is possible, then great)
- but for internal nodes, not great for identifying the best choice...

Hence we move to "Entropy"/"Information Gain" derivative functions.

Where the core idea here is that each level gets evaluate on merit of "predictable" (using lamen terms).

So even if the accuracy classification function isn't improved directly by a particular split, it becomes easier to split further down.

Roughly

In a rough formula fashion, we have the following idea;

$$information\_gain(y, y_{condition}) = entropy(y) - \frac{n_{true}}{n} entropy(y_{true}) - \frac{n_{false}}{n} entropy(y_{false})$$

(example in 35min)

Pruning

Now we know how to grow a decision tree... recall the recursive splitting algorithm. The recursive splitting algorithm will hotswap the classification accuracy function for entropy, although we must know for example, when to stop.

Any technique relevant to restricting the natural growth of this algorithm (either preventative so *pre-pruning* **or** *post-pruning* which happens after the algorithm has run).

Reduced Error Pruning

We iterate through the internal nodes (starting at the bottom), we compare the baseline mode classification model accuracy with the current decision outcome accuracy and if the mode is better than the current decision outcome, then replace the tree section with a majority class classification leaf.

Unsupervised Learning

So we covered decision trees so far in a supervised fashion... where the classification labels are provided (deemed for regression/classification). Unsupervised learning covers things such as; (x_i are features y_i is the associated class labels)

- identify outliers
- similarity (which examples look like x_i)
- association rules (which x_j occur together)
- extract further latent-factors from x_i
- what does the high dimensional X look like
- ranking, which are the most important x_j
- cluster, what *types* of x_i are there