Random forest notes

Justin

# Metrics for splitting the nodes

Algorithms for constructing decision trees usually work top-down, by choosing a variable at each step that best splits the set of items. Different algorithms use different metrics for measuring "best". These generally measure the **homogeneity** of the target variable within the subsets.

This [link](https://github.com/rasbt/python-machine-learning-book/blob/master/faq/decision-tree-binary.md) compares the advantages of different metrics.

Let be the proportion of the samples that belong to class at a particular node , then for this node:

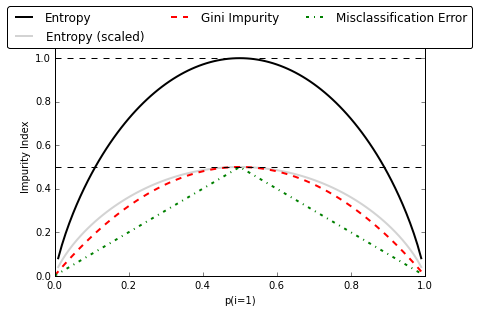


Figure 1. Comparison of various metrics for splitting a node in a decision tree

The classification error metric is not recommended for growing a tree as it’s less sensitive to changes in the class probabilities of the nodes.

As for Gini impurity vs. Entropy, the resulting trees are typically very similar in practice. Maybe an advantage of Gini would be that one needs not to compute the logarithm.

# When will a tree based model behave poorly

The tree models in general cannot extrapolate. In case predicating beyond the training dataset window is desired, a tree model will behave poorly.

# How does the random forest overcome the over-fit issue of a classic decision tree?

A classic decision tree tends to over-fit, which is the direct outcome of the curse of dimensionality. The random forest adopts the ensembling technique and “averages” a bunch of tree models. In addition, during the node splitting, only a fraction of features are randomly selected. This leads to i) variability in the grown trees (which helps with over-fit) and ii) reduced dimensionality for each individual tree (which helps dealing with the curse of the dimensionality)