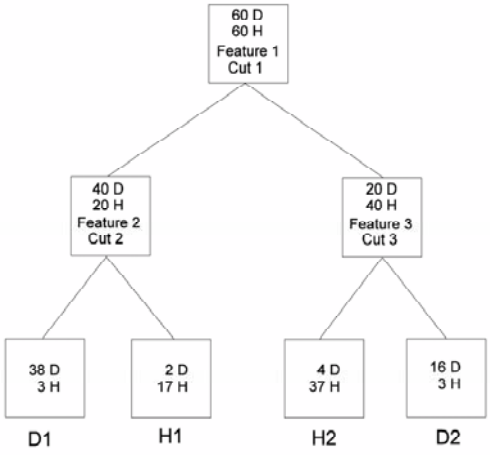
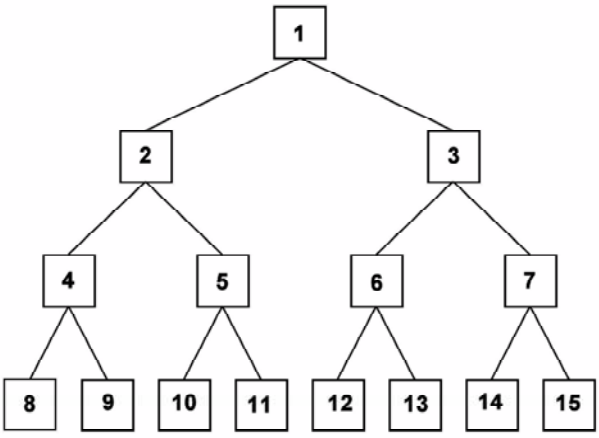
# Decision Trees

Brian T. Luke (lukeb@ncifcrf.gov)

The simplest example of fingerprinting is a single decision tree(DT) [Ho-06, Liu-05, Yan-05, Yu-05], like the one shown at the right which attempts to distinguish diseased from healthy individuals. All samples are initially placed in the root node at the top and a feature within the dataset us used to divide the samples into two daughter nodes. In practice, each feature in the dataset is selected and all possible cut points are examined. This is done by ordering the feature intensities from lowest to highest and assigning the possible cut points to be the midpoint of consecutive intensities. For each cut point, the samples are divided into two daughter nodes depending upon whether their intensity is below or above the cut point. Once all samples are placed into the daughter nodes, the quality of this feature and cut point combination is determined, usually using a metric like the Gini Index or Information Gain. The feature and cut point with the highest quality is retained. If either daughter node has a sufficient number of samples from each category (or two or more categories if there are more than two in the dataset) it becomes a new decision node and the search for an optimum feature and cut point continues. If the number of samples from all categories but one is sufficiently small, it becomes a terminal node and the classification of this node is set to the category with the largest number of samples. The only problem with this procedure is that effectively the same question is asked at each decision node: "Do you have the disease?"




Decision Support also uses decision trees, but an independent question is asked at each level in the tree. For example, Node 1 may be used to separate the individuals by gender, race, or other genetic difference, and then different features may be used to separate samples obtained from diseased and healthy patients at a given level of stratification. Since the stratifying variables are not known ahead of time, there is no way to know the proper metric that should initially separate the training set. Therefore, the procedure used here is to construct unconstrained decision trees that best classify the training individuals. This search uses a symmetric decision tree with seven decision nodes, like the one shown at the right. A modified Evolutionary Programming (mEP) procedure is used to construct these trees. Each putative decision tree classifier is represented by two 7-element arrays; the first contains the feature used at each node and the second contains the cut values. Both arrays assumed the node ordering



shown in this figure. The only caveats are that all seven features must be different and that this ordered septet of features cannot be the same as any other putative solution in either the parent or offspring populations. When a new putative decision tree is formed, a local search is used to find optimum cut points for this septet of features. The decision tree is constructed for each set of cut points and the classification of the terminal nodes is used to determine the overall sensitivity and specificity if this putative decision tree. The quality is set as the sum of the sensitivity and specificity, and the set of cut points with the highest quality is retained.

At this point, all of the decision nodes are examined. If any decision node has a small enough fraction of samples from all categories but one, it is converted into a terminal node. For example, if Node 4 contains a small enough fraction of samples from all categories but one, it is converted into a terminal node and Nodes 8 and 9 are removed from the tree; forming a decision tree with only six decision nodes. The quality of this tree is then re-determined by classifying the remaining seven terminal nodes. If this decision tree is retained by the mEP algorithm and is used to create a new decision tree, Node 4 must also be a terminal node and only the remaining decision nodes can be assigned a new feature to generate a new decision tree.

As described more completely in the section dealing with coverage, an *a priori* division of the samples into a training set and a testing set may lead to poor decision trees since it is possible to severely deplete the samples from one of the terminal nodes, which effectively changes that section of the decision tree. In addition, if less samples are present during the construction of the decision tree, the location of the optimal cut points may be incorrect. This may accidentally yield a decision tree that does not classify the testing set nearly as well as the training set and a modification of the cut points may produce more consistent results.

It should be stressed that the procedure used here to construct unconstrained decision trees will yield sub-optimal trees without using a prohibitive amount of computer time. Finding the optimum decision tree requires not only selecting the correct features but placing them in the proper order and finding the best cut points. Only a limited search of the best set/order of features is performed, and for each putative set of features the location of the optimum set of cut points is limited. Therefore, all results from this procedure should be taken as lower bounds to the accuracy that is attainable for any given dataset.

(Last updated 4/27/07)