# CS 181 Assignment 1: Decision Trees

*Mark VanMiddlesworth & Shuang Wu*

2/11/11

Problem 1 (Decision Trees and ID3).

(a) We calculate

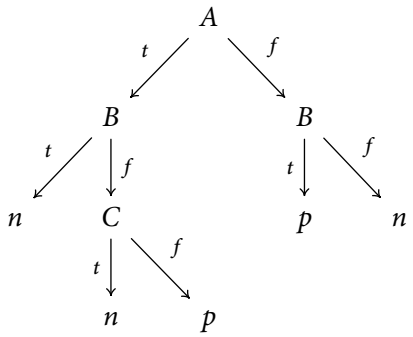
$$\text{Entropy}(3:4) = -\frac{3}{7}\cdot\log_2\left(\frac{3}{7}\right) - \frac{4}{7}\cdot\log_2\left(\frac{4}{7}\right) = 0.985.$$

We then have the following information gain:

| Feature | Induced Data Partition | | | | Remainder | Gain |
|---|---|---|---|---|---|---|
| | Value | Negative | Positive | Entropy | | |
| $A$ | true | 2 | 2 | 1 | $\frac{4}{7}\cdot 1 + \frac{3}{7}\cdot 0.918 = 0.965$ | 0.020 |
| | false | 1 | 2 | 0.918 | | |
| | Value | Negative | Positive | Entropy | | |
| $B$ | true | 1 | 1 | 1 | $\frac{2}{7}\cdot 1 + \frac{5}{7}\cdot 0.971 = 0.979$ | 0.006 |
| | false | 2 | 3 | 0.971 | | |

Hence ID3 will choose to split on feature $A$. While splitting on feature $A$ leads to a higher information gain due to the fact that there is lower entropy for the false values, feature $B$ might be more useful since there are less data points with entropy 1 (that is, perfectly mixed) after the split. This example shows how ID3 is biased towards splits that generate a few partitions that are very extreme at the cost of mixture in the remaining data.

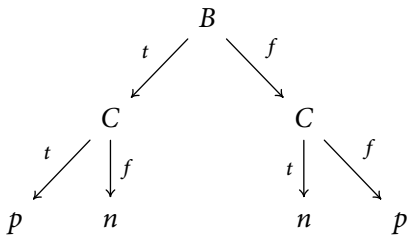
(b) A possible tree generated by ID3 is given here:




We split first on $A$ since splitting on $C$ and $D$ yield the same ratios in their induced data partitions as $B$, and from the previous part, $A$ had a higher information gain. For $A$ true, we see that splitting by $D$ yields two perfectly mixed partitions, while splitting by $B$ or $C$ leads to a perfectly classified true partition, so we pick one of those two to split on (here we chose $B$). On the other hand, for $A$ false, we see that $B$ and $C$ again induce a perfectly classified true partition, while $D$ does not split the data, so we again arbitrarily

pick one of $B$ or $C$ (we again chose $B$). On both sides of the tree, since $B$ perfectly classifies the true partition, we hit create a leaf. On the left hand side, for $B$ false, we see that $C$ perfectly splits the remaining data, while $D$ does not, so $C$ is the next split leading to their respective leaves. on the right hand side, for $B$ false, neither $C$ nor $D$ yield any information gain (neither splits the remaining two data points), so we arbitrarily choose a value (negative here) since there is exactly one negative and one positive instance.

(c) One possible tree with the same training error (that is, one error) is as follows:




We see from this example that the ID3 algorithm does not necessarily generate the simplest tree for which the training error is minimized, even though it will always find a tree that minimizes such error. ID3's greediness when it comes to choosing a split based on highest information gain performs poorly when features have low gain independently but are co-dependent; we constructed the graph above by noticing that $B \oplus \neg C$ (in other words, true if and only if $B = C$) is a pretty good hypothesis.

Problem 2 (ID3 with Pruning).

(c) ID3 likely suffers from overfitting on this data set. Because pruning improves performance in both the clean and noisy cross-validation, it is likely that the pruned nodes contributed to overfitting but did not contribute to test-set or real-world accuracy.

(d) i. Our ID3 implementation uses weighted entropy to calculate information gain. If it used instance count, rather than weight, than AdaBoost's instance weighting (designed to improve performance on "harder" data points) would be ineffective.

ii. True weight: $W_t = 0.5 \times 1 = 0.5$; false weight: $W_f = \frac{0.5}{n-1} \cdot (n-1) = 0.5$. Hence entropy is 1.

i. As predicted, boosting performs poorly on noisy data. This is because boosting weights "difficult" data, i.e. data that many hypotheses classify incorrectly, more heavily than "easy" data. This can lead to weighting noise heavily, and thus giving more weight to classifiers that correctly classify noisy (incorrect) training data. On the other hand, pruned decision trees blindly simplify classifiers up to a certain threshold, so they are less susceptible to noise in compensating for overfitting.

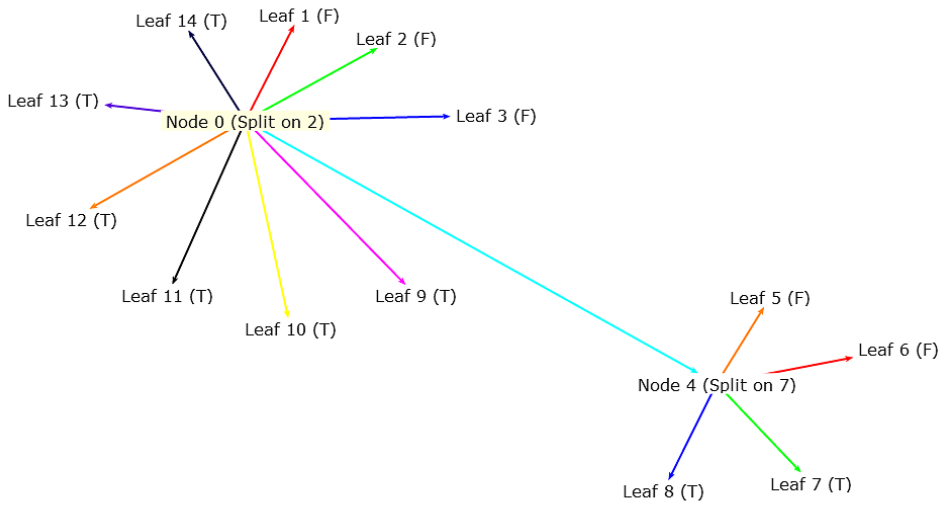
ii. Depth 1 performs better than depth 2 on non-noisy training data, but the performance is roughly equal on noisy training data. A possible explanation is that the depth 2 trees overfit the non-noisy training data, and that the overfitting hypotheses (i.e. those that correctly classify the training data) are heavily weighted by AdaBoost. This is perhaps an example of the base learner actually being too strong at depth 2 on the clean data. On noisy training data, the same hypothesis (that overfit the non-noisy data) would receive a lesser weight because it would be unable to correctly classify all of the training instances.

iii. 30 rounds performs better than 10 rounds of boosting, even though all training data is correctly

classified after 10 rounds. This might be surprising if we didn't know that boosting produces a maximum-margin classifier because we might naïvely assume that there is nothing to be learned after all training data is correctly classified.

iv. After 6 rounds, all training data is correctly classified. However, because boosting produces the maximum margin classifier, performance continues to improve until the 8th round, and does not stabilize until the 11th. The 11th round does not perform as well as the 8th; this could arguably be due to overfitting, although the difference of 1% may not be statistically significant.

Problem 3 (Tree Analysis). Based on the cross-validated performance on the BCW data, we chose the pruned decision tree as a relatively effective decision tree. Although the boosted result does much better on the clean data than any of the other trees, it also suffers the greatest (and is indeed the worst performing) for the noisy data. Given the nature of the data, it seems appropriate to assume some degree of noise in the data, so we choose the second best performing tree overall, the pruned tree. The tree is given here:




This particular tree was generated from the first 90 out of 100 entries in the clean BCW data, and post-pruned using a validation set consisting of the last 10 data points. The leaves are arranged in increasing order starting from the 12 o'clock position for each node and progressing clockwise, and the root node is the one highlighted in yellow. The root node corresponds to a split on the attribute "uniformity of cell shape" and the child node corresponds to a split on "normal nucleoli". We see that as the "uniformity of cell shape" increases past a certain threshold, the tumors are classified as malignant, while at the margin, "normal nucleoli" is used as a secondary classifier which also moves from benign to malignant past a threshold level.