Patrick Austin  
CS 491: Data Mining

Homework # 2

4/10/2017

2-1 a. The entropy of this training set is - ( .4 \* log(.4) ) - ( .6 \* log(.6) )

= .529 + .442 = .9708

b.   
 Body temperature:

The entropy of the children is

.5 ( - ( .2 \* log(.2) ) - ( .8 \* log(.8) ) ) + .5 ( - ( 1 \* log(1) ) - ( 0 \* log(0) ) )

= .5 ( .4643 + .2575) + 0 = .3609

And the gain is .9708 - .3609 = .6099

Give birth:

The entropy of the children is

.5 ( - ( .2 \* log(.2) ) - ( .8 \* log(.8) ) ) + .5 ( - ( 1 \* log(1) ) - ( 0 \* log(0) ) )

= .5 ( .4643 + .2575) + 0 = .3609

And the gain is .9708 - .3609 = .6099

c.

Parent value:

1 - max (.4, .6) = .6

Give birth:

The classification error of the children is

.5 ( 1 - max(.2, .8) ) + .5 ( 1 - max(1,0) ) = .1

And the gain is .6 - .1 = .5

Four legged:

The classification error of the children is

.4 ( 1 - max(.5, .5) ) + .6 ( 1 - max( ⅓, ⅔ ) ) = .4\*.5 + .6\*⅓ = .4

And the gain is .6 - .4 = .2

Since we choose the split that maximizes the gain, splitting by “Give birth” is better.

d.

Parent value:

1 - (.5)2 - (.5)2 = .5

Give birth:

The GINI index of the children is

.5 ( 1 - (.8)2 - (.2)2 ) + .5 ( 1 - (1)2 - (0)2) = .5\*.32 = .16

And the gain is .5 - .16 = .34

Four legged:

The GINI index of the children is

.4 ( 1 - (.5)2 - (.5)2 ) + .6 ( 1 - (⅓)2 -(⅔)2 ) = .4\*.5 + .6\*.444 = .4666

And the gain is .5 - .4666 = .0334

Since we choose the split that maximizes the gain, splitting by “Give birth” is better.

e. Attached.

2-2 a. Entropy is a measure of the heterogeneity of a node. Zero entropy, which is the minimum possible value for entropy, signifies that a node has no heterogeneity. That is to say, zero entropy for a node means that the node is homogeneous, i.e. all the records in that node belong to the same class. In terms of information gain, zero entropy implies the maximum possible information gain. Zero entropy nodes are thus the most desirable to split into, and should not be split further once reached.

b. Pre-pruning and post-pruning are both approaches aimed at addressing decision tree overfitting, which is the problem of generating a tree so well-tailored to the training dataset that the tree actually loses predictive accuracy when used to classify test data.

The pre-pruning approach attempts to address this problem by adding additional conditions that can take place during tree generation that will terminate the process of adding nodes to the tree. The intent is that these conditions will prevent the tree from becoming overly complex and will thus prevent overfitting. Some of these conditions might include a user-specified maximum number of nodes the tree is allowed to generate, or adding tests that will stop tree generation if class distribution is independent of available features, or if additional splits do not improve impurity measures in the leaf nodes. This approach has a significant downside in that seemingly bad splits can lead to good splits further down the tree, and thus pre-pruning can stop the generation process too early.

The post-pruning approach attempts to address this problem by allowing the complete decision tree to be generated, then trimming sub-trees by replacing them with leaf nodes if their removal improves generalization error. This avoids the problem of stopping too early faced by a pre-pruning approach.

c. Entropy has trouble with real attributes where the number of possible values for the attribute equals or approaches the number of records. For example, a user ID attribute where each user has a unique ID will create very favorable splits when using entropy, since a split on user ID will create zero entropy nodes for all the records. However, such a split is of little practical use and is very likely to create an overfitted tree. Similar but less severe problems may occur if the number of possible values merely approaches the number of records. Consider, for example, an address attribute in a record set consisting of individuals: most records have different addresses, but a few people may have the same address. A similar problem with entropy measures would arise.

A preprocessing approach to address this problem can compare the mapping of the set of values for an attribute, x, to the set of n records. If there is a one-to-one map between x and n (or the mapping approaches one-to-one to some degree that could be specified by the user) then the attribute should be excluded from consideration by the tree generation algorithm.

2-3 a. Optimistic generalization error for T1 = 15/73 = .2055

Optimistic generalization error for T2 = 20/73 = .2740

b. Pessimistic generalization error with penalty 0.5 for T1 = ( 15 + .5\*13 ) / 73 = .2945

Pessimistic generalization error with penalty 0.5 for T2 = ( 20 + .5\*6 ) / 73 = .3151

Pessimistic generalization error with penalty 0.75 for T1 = ( 15 + .75\*13 ) / 73 = .3390

Pessimistic generalization error with penalty 0.75 for T2 = ( 20 + .75\*6 ) / 73 = .3356

Pessimistic generalization error with penalty 1 for T1 = ( 15 + 13 ) / 73 = .3836

Pessimistic generalization error with penalty 1 for T2 = ( 20 + 6 ) / 73 = .3562

c. T1 is preferable from the optimistic point of view, or when using penalty term 0.5 in a pessimistic approach, since it has a lower measure for error than T2. If the penalty term is .75 or 1 and the pessimistic approach is used, then the additional error added by T1’s larger number of nodes compared to T2 begin to weigh more and more heavily, making T2 preferable in terms of error value. So the question of which tree is preferable depends on the approach used, and the weight applied if using the pessimistic approach.

d. Occam’s Razor says that given two models with similar generalization error, the simpler model (i.e. the one with fewer nodes) should be preferred. We could certainly quibble about how to define the word ‘similar’: is a tree with an error of .2055 similar to a tree with an error value of .2740? What about a tree with an error value of .3836 versus one with an error value of .3562? What is clear is that there are various ways to measure error as shown in a. and b., and the choice of measure will influence how similar the error values for the trees are.

In the absence of further guidance about the numerical meaning of similarity, I would say in my judgment T1 should be preferred in the optimistic case, since its error is significantly lower than that of T2, even though it has more nodes. When using the pessimistic approach the difference in error between T1 and T2 decreases considerably, to an extent that I think T2 should be preferred for being a simpler model with fewer nodes, as Occam’s Razor specifies.