Principles of Data Mining (420)
Spring 2022, Homework 05

Mohammed Mehboob
(mm2260@rit.edu)

April 3, 2022

1. What cost function did you design for your splitting criterion?
   Did you design it?
   What was your objective function?
   What part was your regularization, if any?

Answer: The cost function for the splitting criterion was the ***mixed entropy*** of the resulting data partitions after the split. The attribute and its corresponding threshold value to split our data at was greedily calculated by $\underset{\texttt{thresh,attr}}{\arg\min}\,(\ \textit{mixed-entropy}\ )$, and the decision tree was constructed in a top-down approach.

The objective function was to maximize the information gain, which is equivalent to minimizing the mixed entropy at each split.

Regularization was not directly implemented as part of the cost function in our decision-tree generation process, since regularization in terms of the decision trees is to control how the tree grows. The prevention of overfitting during the generation of our decision tree is done through controlling hyperparameters such as the minimum leaf node size, maximum tree depth, and maximum percent representation of either class within a node.

2. Describe the decisions of your final trained classifier program (the resulting classifier).
   Copy the top few decisions here.
   Inspecting it, what does it tell you about the relative importance of the attributes

Answer:

Looking at the first few top-level decisions, it is clear that the `EarLobes` attribute cuts down the most entropy. This does not come as a surprise, since it is a categorical attribute, and making this the top-level decision greatly cuts down on the entropy. After right after that, in the true branch, comes `bangln`. We can also see that the `height` attribute pop up twice.

```
if ( earlobes < 1 ):
    if ( bangln <= 6 ):
        if ( hairln <= 10 ):
            if ( ht <= 136 ):
                if ( tailln <= 6 ):
                    if ( ht <= 132 ):
                        prediction = 1
                    else:
                        if ( reach <= 140 ):
                            if ( age <= 46 ):
                                prediction = −1
                            else:
                                if ( age <= 56 ):
                                    prediction = 1
                                else:
                                    prediction = −1

                                ⋮
```

Taking a cursory glance at the decision-tree else-if ladder points towards `earlobes`, `bangln`, `hairln` and so on attributes being of fairly high amount of importance in terms of how much information gain they bring to the table, having been chosen by our algorithm as the first few decision stumps.

3. Generate a confusion matrix for the given training data

Answer:

	True (Actual)	False (Actual)
True (Predicted)	2360	97
False (Predicted)	140	2403

4. What was the accuracy of your resulting classifier, on the training data?

Answer:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + TP} = 0.9526$$

Therefore, we have an accuracy of 95.26%.

5. Did anything go wrong?

Answer: The decision tree we generated may have been overfitted to our training data. It gives really good results for our specific training set, however, there is no guarantee that this decision tree is generalized and that it will produce satisfactory results for a dataset that it has not seen before.

6. Discussion:
What would happen if you changed the number 9 to 23?
What if it was 5?
What does this value control?

Answer: When we choose our minimum leaf size hyperparameter to 23, we end up with a similar accuracy of 94.54% in addition to the code length of our classifier going down from about 300 lines to close to 200. And so, according to Occam's razor, given that two solutions are similar in accuracy, we can choose the one that is similar. And since the tree isn't as big, we can also say that there is a relatively lower of a chance of our decision tree being overfitted.

	True (Actual)	False (Actual)
True (Predicted)	2361	134
False (Predicted)	139	2366

Confusion Matrix: Min Leaf Size = 23

Changing it to 5 also yields favorable results, with an accuracy of 95.44%, however, this is at a classifier code length of roughly 350 lines.

Conclusion:

The first question was very interesting, in such that it threw me for a loop, since it was instructed to us to used the mixed entropy as our cost function in the homework writeup, yet the question asked for what we did. It got me thinking about the relationship between cost functions, loss functions, objective functions and the like. It was also intriguing to make the connection between the minimization of mixed entropy and the maximization of information gain to be equivalent. It seems trivial in retrospect, yet it was not as such on my way to getting there. Regarding regularization, I conjecture it to be sort of a trick question, perhaps?? In my search to figure out regularization for cost functions for desicison trees, all I could find was that we simply try to maximize the information gain. Regularization comes in through how we control the growth of our decision tree. That, and we can use approaches such as post-pruning to counter overfitting.

It also related back to the no-free-lunch-theorem, in the way that we have to try different hyperparameters – different values for our minimum leaf size, or the percent representation of either class in a node, or perhaps the maximum depth – to figure out exactly what model will serve our particular needs best. Also that accuracy is not everything: we also have to consider the performance of our model, and whether or not our model is generalized enough to work with a variety of datasets. That also brings in the fact that accuracy is not everything, we may also want a model where, for example, we want our false positive rate to be very low (like in our discussion about ROC curves).