COMP4702/COMP7703/DATA7703 - Machine Learning
Homework 9 - Evaluating Models and Decision Trees
Solutions

Marcus Gallagher

Core Questions

1. I found a random page on the internet containing python code and when I ran it the following confusion matrix was produced (note that the correct classes run down the rows and the predicted classes across the columns as in the lecture and Alpaydin):

```
Confusion matrix
[[27  0  0  0  0  0  0  0  0  0]
 [ 0 37  0  0  0  0  0  0  0  0]
 [ 0  1 24  0  0  0  2  1  0  0]
 [ 0  0  0 28  0  5  0  1  0  1]
 [ 0  0  0  0 24  0  0  0  0  0]
 [ 0  0  0  0  0 32  0  0  0  2]
 [ 0  0  0  0  0  1 24  0  0  0]
 [ 0  0  0  0  1  3  0 31  0  0]
 [ 0  7  0  0  0  0  1  0 21  2]
 [ 0  0  0  0  1  2  0  0  0 21]]
```

   (a) What is the size of the training set used?
   Solution: sum of all elements = 300.

   (b) Which class had the highest number of errors?
   Solution: The 9th class (row) has the most (10) off-diagonal values.

   (c) What was the overall percentage correct?
   Solution: Ratio of diagonal v's total = $\frac{269}{300} = 89.67\%$

2. Decision trees partition the feature space into regions which are (hyper)rectangles. Consider the decision tree shown in Figure 9.5 of the Hastie et al. textbook. How many hyper-rectangles does the model define in the feature space for the `spam` class?

   Answer: 8. The number of hyper-rectangles will be equal to the number of leaf nodes in the decision tree (see Figure 9.2 of Hastie et al.). In Figure 9.5, there are 8 leaf nodes that predict `spam`.

3. Consider again the decision tree shown in Figure 9.5 of the Hastie et al. textbook. How many parameters does this model have?

Answer: 16. Each non-leaf node splits on a single feature. The value that it splits on is what is determined by training and would need to be stored. From this point of view, these values are the model parameters and there are 16 non-leaf nodes in the tree. One could possibly argue that the feature also needs to be stored at each non-leaf node, which would be 32 parameters.

Extension Questions

In my opinion, one of best experimental comparisons of classification techniques ever done can be found in this book[1] (pdf version is freely available and I have put it on the course blackboard site). You might like it as an alternative reference for some of the material that we have studied in the course. The book is now old enough (1994) to have some historical value - it does not include support vector machines, ensemble methods and deep learning because it pre-dates them!

Grab a coffee, sit down and have a relaxing read the start of Chapter 7 of this book (<3 pages, up to the end of 7.1.3 only). Do you think it is still correct and useful in 2020? Are there any statements that are out of date or irrelevant? Write a paragraph (approx. 250 words) as your answer.

Solution (outline): This part of the book is a discussion about assessing the performance of a classifier, in particular comparing cross-validation with bootstrapping. Overall I think it is correct and useful in 2020. Perhaps the sizes of datasets that might be considered "moderate", "large", etc. might have changed a bit. Computational power has obviously increased but so have the requirements of some models (e.g. neural networks). More precise things can be said about the properties of bootstrapping and cross-validation (see, e.g. the Hastie et al book) and more is known about them than was known in 1994.

There is some scope for student opinion variations of answers here.

References

[1] D. Michie et al. (editors), Machine Learning, Neural and Statistical Classification. Ellis Horwood, 1994.