Assignment 4: Decision Trees and Boosting

CS486/686 – Spring 2006

Out: July 6, 2006
Due: July 25, 2006

Be sure to include your name and student number with your assignment.

Equine Colic Diagnosis

Equine colic is one of the leading causes of death in adult horses. However, if diagnosed early enough, it is usually surgically curable. In this assignment, you will implement (in the language of your choice) a decision tree algorithm and a boosting algorithm to learn to diagnose whether a horse is healthy or has colic. On the course website, the file **horse.train** contains a database of horse records to train your algorithms. A second file, **horse.test**, contains another set of horse records to test your algorithms. Each record has 16 numeric attributes (features) and a classification (healthy or colic), all separated by commas. The attributes correspond to the following measurements made from each horse at admission to the clinic:

1. K
2. Na
3. CL
4. HCO_3
5. Endotoxin
6. Aniongap
7. PLA2
8. SDH
9. GLDH
10. TPP
11. Breath rate
12. PCV
13. Pulse rate
14. Fibrinogen
15. Dimer
16. FibPerDim

Feel free to reformat the files **horse.train** and **horse.test** in any way you want so as to make reading easier for your programs.

1. **[50 pts]** Decision Tree Learning

   Implement a decision tree learning algorithm. In the decision tree, use only binary tests, i.e., each node should test whether a particular attribute has a value greater or smaller than a threshold. In deciding which attribute to test at any point, use the information gain metric. More precisely, at a given node, try setting the threshold for each potential attribute to any value that is halfway between successive attribute values. For instance, suppose that attribute K takes only values 4, 4.5, 4.6 and 4.7, then possible thresholds for K would be 4.25, 4.55 and 4.65. Use the threshold that yields the highest infomation gain. Allow the same attribute to be tested several times (but with different thresholds) in the same branch. Learn a decision tree of at most 10 decision nodes (fewer nodes may be sufficient if all the training records are correctly classified) with the file **horse.train**. Then test your decision tree with the file **horse.test**.

   What to hand in:

   - A printout of your code.
   - A printout showing the attribute selected and the corresponding information gain at each split when running your decision tree learning algorithm.
   - A picture of the decision tree learned (hand drawn is fine).
   - The number of train records correctly classified.
   - The number of test records correctly classified.
   - Does your decision tree generalize well on the horse data set (i.e., does it avoid overfitting)?

2. **[50 pts]** Boosting

   Implement the AdaBoost algorithm. More precisely, consider decision stumps corresponding to single-node decision trees as weak learners. Again, for each attribute consider thresholds that are halfway between successive attribute values. Boost at most 10 decision stumps (fewer decision stumps may be sufficient if all the training records are correctly classified) with AdaBoost using the file **horse.train**. Then test the resulting weighted majority of decision stumps with the file **horse.test**.

   What to hand in:

   - A printout of your code.
   - A picture showing the list of decision stumps, with their weight, found by AdaBoost (hand drawn is fine).
   - The number of train records correctly classified.
   - The number of test records correctly classified.
   - Does AdaBoost generalize well on the horse data set (i.e., does it avoid overfitting)?
   - Decision stumps can be boosted as long as their accuracy is at least as good as a random classifier. Are decision stumps guaranteed to have an accuracy at least as good as a random classifier? Explain briefly.