Assignment 3: Decision Trees and Boosting

CS486/686 – Fall 2008

Out: October 28, 2008
Due: November 13, 2008

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

Text categorization

Text categorization is an important task in natural language processing and information retrieval. For instance, news articles, emails or blogs are often classified by topics. In this assignment, you will implement (in the language of your choice) a decision tree algorithm and a boosting algorithm to learn a classifier that can assign a newsgroup topic to any article. On the course webpage, a training set and test set of articles with their correct newsgroup label will be posted within 5 days. To simplify your implementation, these articles have been pre-processed and converted to the *bag of words* model. More precisely, each article is converted to a vector of binary values such that each entry indicates whether the document contains a specific word or not.

1. **[45 pts]** Decision Tree Learning

   Implement a decision tree learning algorithm. Here, each decision node corresponds to a word feature. Grow the tree one node at a time, up to 100 nodes, by training with the training set only. Use the information gain metric to decide with word feature to add at each step. Report the training and testing accuracy (i.e., percentage of correctly classified articles) of each tree (from 1 to 100 nodes) by producing a graph with two curves (one curve for training accuracy and one curve for testing accuracy).

   What to hand in:

   - A printout of your code.
   - A printout (or hand drawing) showing the decision tree with the first 10 word features selected and their corresponding information gain.
   - A graph showing the training and testing accuracy as the number of nodes increases.
   - Does overfitting occur? If yes, after how many nodes does overfitting start?
   - A brief discussion regarding the word features selected by the decision tree learning algorithm. In your opinion, did all the word features selected make sense? If not, how would you explain the word features that do not make sense?

2. **[55 pts]** Boosting

   Implement the AdaBoost algorithm. More precisely, consider decision stumps corresponding to single-node decision trees as weak learners. Boost up to 100 decision stumps by training with the training set only. Report the training and testing accuracy (i.e., percentage of correctly classified articles) as a function of the number of decision stumps by producing a graph with two curves (one curve for training accuracy and one curve for testing accuracy).

   What to hand in:

   - A printout of your code.
   - A printout listing the first 10 decision stumps with their weights (as found by AdaBoost).
   - A graph showing the training and testing accuracy as the number of decision stumps increases.
   - Does overfitting occur? If yes, after how many decision stumps does overfitting start?
   - 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.
   - A brief discussion regarding the word features selected by Adaboost. In your opinion, did all the word features selected make sense? If not, how would you explain the word features that do not make sense?
   - A brief discussion of decision tree learning versus boosting. Which algorithm do you prefer for text categorization? Explain your answer by discussing any relevant issue such as accuracy, overfitting, efficiency or choice of word features.