COMP 135 – Machine Learning – Fall 2017

Homework Assignment 6

Due date: Monday, 12/11 (hardcopy in class)

1. ~~In this question we consider the Naive Bayes algorithm and active learning.~~ ~~Consider the dataset below where the first 6 examples are labeled and the last 4 are not.~~ Consider ~~active learning with uncertainty sampling~~ (where in the notation of the slides from lecture we pick the instance with the smallest p_1).

   TODO: Look back at this, p1

   ```
   000  +
   011  +
   011  -
   110  -
   101  +
   100  -
   010
   101
   110
   001
   ```

   (a) ~~Show the hypothesis of Naive Bayes after learning from the 6 labelled examples~~.

   (b) Which example will be picked by uncertainty sampling? explain your answer

2. ~~Recall that Bagging obtains multiple hypotheses by running the algorithm multiple times with bootstrap samples~~. ~~The algorithm then predicts the label of a test example to be the mode of the predicted labels (the label predicted the highest number of times by the learned hypotheses).~~

   (1) ~~What do you expect about the performance of Bagging when run with a stable base learner (which is not sensitive to small changes in the training set), for example, a support vector machine algorithm? explain why you expect this behavior.~~

   (2) Explain how Adaboost obtains multiple hypotheses and how it predictcs.

   (3) Explain why and how Adaboost is better at using stable base learners than Bagging.

3. (Variation of problem 7.7 from mitchell's text page 228)

   Consider the hypothesis class of H_{rd2} of "regular depth 2 decision trees" over n Boolean variables. A "regular depth 2 decision tree" is a full depth 2 decision tree (having 3 internal nodes and 4 leaves all at depth 2) and where the left and right children of the root split on the same variable test.

   (a) As a function of n, how many syntactically distinct trees are there in H_{rd2}?

   (b) Give an upper bound on the number of examples needed to PAC learn H_{rd2} with error ϵ and confidence δ.

   (c) Consider applying the Weighted Majority (WM) algorithm as follows. Each syntactically distinct tree is used as one advisor for the WM algorithm. In each step, advisors who predicted wrongly are penalized by having their weight multiplied by 0.5 (exactly as in the analysis in class).

   What is the run time of the algorithm for each prediction step in terms of n (please use $O()$ notation)? Provide an upper bound on the number of mistakes the WM will make in terms of n, and the best performance of any H_{rd2} on the same sequence.