1 Introduction

In this assignment we review several concepts that were discussed in class. Please submit paper solutions in class.

1. Recall that in decision tree learning we had a choice of splitting criterion. Different criteria were based on different notions of impurity. In particular, consider a 2 class case and assume that the current node has \( n \) examples of which a proportion of \( p \) are positive example and \((1 - p)\) are negative examples. We discussed impurity in terms of the following three measures. (1) error rate: \( \min\{p, 1 - p\} \). (2) entropy: \( p \log \frac{1}{p} + (1 - p) \log \frac{1}{1-p} \). (3) the gini criterion: \( 2p(1 - p) \).

Construct hypothetical situations (specifying number of examples, numbers of each class at root node, and same for children after a split on different attributes) where (1) and (2) will make different choices, and where (2) and (3) will make different choices.

2. Complete the details of the version space example discussed in class (and available as slide copy on the course web page). In particular specify the \( G, S \) sets after examples 1-3 have been seen and after each of examples 4,5,6,7 and their labels are seen. (so in total you need to provide five \( S \) and \( G \) sets. For every transition explain how the sets \( S, G \) are updated.

3. Professor Cubic proposed to use neural networks to approximate real valued functions by using a concrete polynomial \( \sigma(s) = 3x^2 - 5x + 2 \) as the activation function. In other words a node with inputs \( \vec{x} = (x_1, \ldots, x_n) \) and corresponding weights \( \vec{w} = (w_1, \ldots, w_n) \) will output \( o = \sigma(\vec{w} \cdot \vec{x}) = \sigma(\sum w_i x_i) \).

Develop a gradient descent algorithm for a single node of this type. In particular, given a training set \((\vec{x}^1, L^1), (\vec{x}^2, L^2), (\vec{x}^3, L^3), \ldots \) where superscripts index examples and their labels use the error function \( E = \frac{1}{2} \sum_k [L^k - \sigma(\vec{w} \cdot \vec{x}^k)]^2 \) to derive the gradient descent update for \( \vec{w} \).

4. A learning algorithm produces a hypothesis that ranks examples in the test set so that their labels are ordered as follows, where the list from left to right denotes ranking from highest to lowest score:

\[
+ - + + + - + - - + - - - - + - - - 
\]

Draw a ROC curve, and a Precision-Recall curve for this ranking using a threshold after each example to generate the points.

5. You are given a hypothesis for a particular domain, test it on 100 independent examples and observe 93% accuracy. Calculate a \( N = 0.99 \) confidence interval for the hypothesis. Use the two solutions given in class to the problem of unknown variance to compare the effect on the size of the intervals.

6. Imagine you ran your favorite algorithm on a new dataset using a 10 fold cross validation scheme and got the following accuracies in the folds: 0.88 0.75 0.79 0.80 0.91 0.83 0.77 0.79 0.82 0.85. Use the formulation of the \( T \) confidence interval to calculate two intervals for the average accuracy, using confidence of 0.99 and 0.9 respectively.