CS534 — Written Homework Assignment 3 — Due Nov 17th 23:59PM, 2018

Please submit electronically via TEACH in a single pdf file.

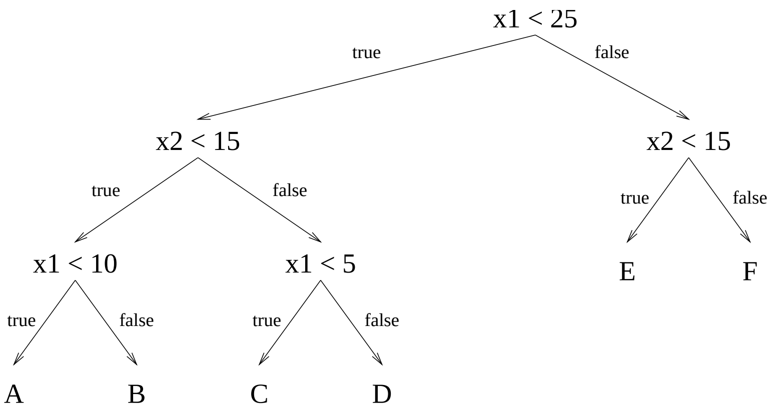
1. Neural network expressiveness
   In class, we have discussed that neural network can express any arbitrary boolean functions. Please answer the following question about neural networks. You can use a step function for the activation function.

   a. It is impossible to implement a XOR function $y = x_1 \oplus x_2$ using a single unit (neuron). However, you can do it with a neural net. Use the smallest network you can. Draw your network and show all the weights.

   b. Explain how can we construct a neural network to implement a Naive Bayes Classifier with Boolean features.

   c. Explain how can we construct a neural network to implement a decision tree classifier with boolean features.

2. Consider the following decision tree:



   (a) Draw the decision boundaries defined by this tree. Each leaf of the tree is labeled with a letter. Write this letter in the corresponding region of input space.

   (b) Give another decision tree that is syntactically different but defines the same decision boundaries. This demonstrates that the space of decision trees is syntactically redundant. How does this redundancy influence learning (does it make it easier or harder to find an accurate tree)?

3. In the basic decision tree algorithm (assuming we always create binary splits), we choose the feature/value pair with the maximum information gain as the test to use at each internal node of the decision tree. Suppose we modified the algorithm to choose at random from among those feature/value combinations that had non-zero mutual information, and we kept all other parts of the algorithm unchanged.

   (a) What is the maximum number of leaf nodes that such a decision tree could contain if it were trained on m training examples?

(b) What is the maximum number of leaf nodes that a decision tree could contain if it were trained on m training examples using the original maximum mutual information version of the algorithm? Is it bigger, smaller, or the same as your answer to (b)?

(c) How do you think this change (using random splits vs. maximum information mutual information splits) would affect the accuracy of the decision trees produced on average? Why?

4. Consider the following training set:

A	B	C	Y
0	1	1	0
1	1	1	0
0	0	0	0
1	1	0	1
0	1	0	1
1	0	1	1

Learn a decision tree from the training set shown above using the information gain criterion.

5. Please show that in each iteration of Adaboost, the weighted error of h_i on the updated weights D_{i+1} is exactly 50%. In other words, $\sum_{j=1}^{N} D_{i+1}(j)I(h_i(X_j) \neq y_j) = 50\%$.

6. In class we showed that Adaboost can be viewed as learning an additive model via functional gradient descent to optimize the following exponential loss function:

$$\sum_{i=1}^{N} \exp(-y_i \sum_{l=1}^{L} \alpha_l h_l(x_i))$$

Our derivation showed that in each iteration l, to minimize this objective we should seek an h_l that minimizes the weighted training error, where the weight of each example $w_l^i = \exp(-y_i \sum_{t=1}^{l-1} \alpha_t h_t(x_i))$ prior to normalization. Show how this definition of w_l^i is proportional to the $D_l(i)$ defined in Adaboost.