B555: Homework 6

1. Create 10 random points $(x_1, y_1), \ldots, (x_{10}, y_{10})$ by sampling taking the x's uniformly over the interval (0,10) and the y's according to $y_i = x_i + e_i$ where $e_i \sim N(0, 5)$.

   (a) Fit a 5th order polynomial to the data and compute the average squared error over your data set.

   (b) Generate 100 additional points according to the same process and compute your average squared error using the same polynomial model from the previous part.

   (c) The previous part estimates the generalization error, which is what we usually care about in ML problems. Use cross-validation to estimate the generalization error using only the original 10 points.

2. Jensen's inequality says that for a convex function, c, $E(c(X)) \geq c(E(X))$. Using the fact that $-\log$ is convex, it follows that

   $$E(\log(X)) \leq \log(E(X))$$

   (a) Use this inequality to show that the average entropy caused by a CART split is no greater than the original entropy. That is, if q_l and q_r are the proportions going to the left and right nodes and p, p_l, p_r are the class distributions at the original, left, and right nodes, then

   $$q_l H(p_l) + q_r H(p_r) \leq H(p)$$

   (b) Let Y be the class of an example and T be the leaf node of the tree for that example, regarded both as random variables. Define the conditional entropy of the class given the tree, $H(Y|T)$, to be $\sum_t p_t H(Y|T = t)$ where p_t is the probability of reaching leaf node t and $H(Y|T = t)$ is the entropy of the class distribution at leaf node t. Show that each split reduces $H(Y|T)$. It is fine to think of all probabilities in this case as proportions.

   (c) The joint entropy of the pair (Y, T) is defined to be $-\sum_{t,y} p_{t,y} \log p_{t,y}$. Show that

   $$H(T, Y) = H(T) + H(Y|T)$$

   This is a general fact about entropy or "information," not depending on the particular example of classification trees.

3. Using the data in "strange_binary.csv," build a classification tree that distinguishes the "good" examples from the "bad" ones using no more than 3 splits.

   (a) Report the classification error rate on this training set. Is it reasonable to assume that your classification accuracy would be similar on test data from the same model?

   (b) Introduce an additional feature that allows you to significantly decrease the error rate, still using only 3 splits. Report the training error rate for this new classifier. It should be possible to get about 80% correct on the training.

4. Create 1000 random points $x = (x_1, x_2)$ with both coordinates uniform in the unit interval $(0, 1)$. Let $t(x)$ be the class label of x in $\{0, 1\}$ and suppose

   $$P(t = 1|x) = \frac{x_1^2 + x_2^2}{2}$$

   Generate random labels according to this probabilistic model. Train a neural network using backpropagation with the two inputs, (x_1, x_2), 5 hidden nodes, and one output corresponding to the probability of class 1. The initial assignment of the 15 weights should be random. Your algorithm should take a small step in the gradient direction at each iteration. Verify that the log likelihood increases at each iteration. After having trained your algorithm plot both the true probability of class 1, as well as the probability according to your neural network for all points (x_1, x_2) such that $x_1 = x_2$. Thus your plots are functions of a single variable.