Shortened Final Exam

Stat 435, Spring 2020
Due Wednesday, June 10, 2020, 11:59pm

Note: Whatever you hand in has to be **your own work**. "Group work" is not allowed.

Problem 1: (60 points) This problem motivates the use of the Gini index as a splitting criterion in CART classification. For background see ISLR Chapter 8.1.3. We will consider a two class classification problem with unit loss:

$$\begin{aligned} R_0 &= [0,3] \times [0,3] \\ R_1 &= [0,1] \times [0,1] \\ p(\mathbf{x} \,|\, Y=0) &= \text{Uniform}\,(R_0) \\ p(\mathbf{x} \,|\, Y=1) &= \text{Uniform}\,(R_1) \\ \pi_0 &= 7/8 \\ \pi_1 &= 1/8 \\ L(0,1) &= L(1,0) = 1 \end{aligned}$$

(a) (10 points) What is the minimum risk prediction rule for a tree consisting only of the root node? What is the corresponding risk?

(b) (10 points) Now consider a tree with leaves $N_l = [0,1] \times [0,3]$ and $N_r = (1,3] \times [0,3]$ What is the minimum risk prediction rule for this tree? What is the corresponding risk? Did splitting reduce the risk?

(c) (10 points) Is there a different split of R_0 into two axis parallel boxes that would the reduce the risk? Justify your answer.

Now consider the probabilistic classification rule that predicts $y = i$ with probability p_i (in contrast to the optimal rule that predicts $y = \text{argmax}_i(p_i)$).

(d) (10 points) What is the risk of the probabilistic rule for the tree that consists only of the root node?

(e) (10 points) Now consider a tree with leaves $N_l = [0,1] \times [0,3]$ and $N_r = (1,3] \times [0,3]$. What is the risk of the probabilistic rule for this tree? Did splitting reduce the risk?

(f) (10 points) Is there a different split of R_0 into two axis parallel boxes for which the probabilistic rule has a smaller risk than computed in (e)?

Problem 2: (60 points) In this problem you will do spam classification. Consider the email spam dataset (in “spam.data”). This consists of 4601 email messages, from which 57 features (or covariates) have been extracted. These are as follows:

- 48 features, in $[0, 100]$, giving the percentage of words in a given message which match a given word on the list. The list contains words such as “business”, ‘ ‘free”, “george”, etc.
- 6 features, in $[0, 100]$, giving the percentage of characters in the email that match a given character on the list. The characters are ; ([! $ #
- Feature 55: The average length of an uninterrupted sequence of capital letters
- Feature 56: The length of the longest uninterrupted sequence of capital letters
- Feature 57: The sum of the lengths of uninterrupted sequence of capital letters

More detail about the data can be found in the file “spam.info”.

Load the data from “spam.data”, in which Columns 1-57 are the features and Column 58 is our response variable, the indicator of spam emails (1 is spam and 0 is non-spam). Divide the dataset into a training set (of size 3065) and a test set (of size 1536) by the indicator in the file “spam.traintest” (1 is test set and 0 is training set).

(a) (20 points) Fit a classification tree to the training set. Plot the fitted un-pruned tree. Make binary predictions and report the error rate on the training and test sets.

(b) (40 points) Prune the tree and select the optimal tuning parameter λ by minimizing the 10-fold cross validation error with the one standard error rule (1-SE Rule). The 1-SE Rule means that we should choose a simpler model if its penalized RSS is less than 1 SE worse than the next complex model. Plot the fitted pruned tree. Make binary predictions and report the error rate on the training and test sets.

Hint: You may find the “rpart” package in R helpful. The examples in the introduction below are also helpful.
http://cran.r-project.org/web/packages/rpart/vignettes/longintro.pdf

Problem 4: (60 points) In this problem you will apply a k-nearest neighbor classifier to the handwritten digit data. The data are divided into a training set `zip-train.dat` and test set `zip-test.dat`. Both data sets have lines of length 257. The first entry in each line is the digit that was written $(0, \ldots 9)$ and the remaining 256 entries are grey levels pixels in a 16×16 bitmap.

(a) (30 points) Write an R function

`knn.classifier(X.train, y.train, X.test, k.try = 1, pi = rep(1/K, K), CV = F)`

where X.train, y.train, X.test have the obvious meanings; k.try is a vector of neighborhood sizes; pi is a vector of prior probabilities; CV = T if leave-on-out cross-validation is to be used.

CV = T only makes sense if X.train = X.test

The function should return a (n.test x length(k.try)) matrix of predicted class identities for the n.test test observations and the different values of k provided in k.try.

(b) (10 points) Run the function on the Iris data with k = 5 and with both choices for CV. Print the respective number of misclassifications.

(c) (10 points) Run the function on the hand-written digit training data with k.try = c(1, 3, 7, 11, 15, 21, 27, 35, 43) and CV = T. Use unweighted Euclidean distance as a dissimilarity measure. Choose the priors to reflect the class frequencies in the training data. What is the optimal choice of k and the corresponding training error rate?

(d) (10 points) Calculate the test set error rate.