CS74/174, Winter 2013, Problem Set # 2

January 31, 2013

Due: February 14, 2013 @ 11:59pm

This problem set requires you to provide written answers to a few technical/theory questions. In addition, you will need to implement some learning algorithms in MATLAB and apply them to data sets enclosed with this homework. You can submit your answers to the technical questions either via Blackboard in electronic form (e.g. by typing your solutions in Latex or Word), or manually (handwrite your answers and drop them in the course mailbox located near the main entrance of Sudikoff). If you choose to hand in manually your answers and you submit late, please write the date and time of your submission on the homework. It is a violation of the honor code to report an incorrect submission date. The MATLAB code must be submitted via Blackboard: include all sources and the figures in a single folder, zip up the entire folder, and submit the compressed file. If you submit your technical answers electronically, include your document in the same folder containing the source code and the figures. Please use the following naming convention for the source code and figure files: the main script/function for question - say - 1(b) should be saved as file "q1b.m" and the corresponding output figure/plot as file "q1b.fig" (you can save .fig files in MATLAB using the command "saveas").

1. **[20 points]** Suppose we want to perform binary classification of real-valued vectors $x \in \mathbb{R}^n$ using a generative classifier.

   (a) [5 points] Show that the decision rule to classify a text example x can be written as:

   $$y = \begin{cases} 1 & \text{if } \frac{1}{1+\exp(-a(x))} > \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

   where $a(x) = \log\left[\frac{p(x|y=1)p(y=1)}{p(x|y=0)p(y=0)}\right]$.

   (b) [15 points] Assume that we decide to use as generative classifier a Linear Gaussian Discriminant Analysis with maximum likelihood parameters $\{\mu_0, \mu_1, \Sigma\}$ learned from a given training set. Prove that for this choice of classifier the resulting decision boundary is *linear* by showing that $a(x) = \theta^T x + b$. To prove this you must derive the analytical expressions of $\theta \in \mathbb{R}^n$ and $b \in \mathbb{R}$ in terms of the maximum likelihood parameters μ_0, μ_1, Σ.

2. **[30 points]** Consider a problem in which the class label $y \in \{0, 1\}$ and each training example x has two binary attributes, i.e., $x = [x_1\ x_2]^\top$ where $x_1, x_2 \in \{0, 1\}$. Let the class prior be

$p(y = 1) = 0.5$. Furthermore, let us assume that:

$$\begin{aligned} p(x_1 = 1|y = 1) &= 0.8 \\ p(x_2 = 1|y = 1) &= 0.5 \\ p(x_1 = 0|y = 0) &= 0.7 \\ p(x_2 = 0|y = 0) &= 0.9. \end{aligned}$$

(a) [5 points] Assume that x_1 and x_2 are truly independent given y. Write down the decision rule of Naive Bayes for this particular case, i.e., list the predicted label y^{pred} for the four possible values of vector x.

(b) [10 points] Show that if Naive Bayes uses both attributes, x_1 and x_2, the error rate is 0.235. The error rate is defined as the probability that the predicted label is incorrect, i.e., $p(y \neq y^{pred})$. To receive credit you must explain your derivation.

(c) [3 points] Calculate the error rate when using only the single attribute x_1 for prediction. What is the error rate when using only x_2? Are these errors higher or lower than when using both attributes? Can you explain why?

(d) [7 points] Now, suppose that we create a new attribute x_3, which is an exact copy of x_2. So, for every training example, attributes x_2 and x_3 have the same value, $x_2 = x_3$. What is the error rate of Naive Bayes now? (Hint: The true distribution has not changed.)

(e) [3 points] Explain what is happening with Naive Bayes in this last case.

(f) [2 points] Would Logistic Regression suffer from the same problem? Explain your answer.

3. **[30 points]** In this problem you will have a chance to develop your own spam filter. You will actually implement several versions, corresponding to the following classifiers: Naive Bayes, Decision Tree, and Random Forest. The data set that you will be using ("spamdata.mat") was derived from spam and non-spam emails collected a few years ago at Hewlett Packard Research Labs to test different spam detection algorithms. Each email is represented as a vector of 48 binary features indicating whether or not particular words occur in the email (the complete list of words can be found in the file "spambase_names.txt"). The label y takes on two values, one corresponding to "ham" (i.e., valid email) and the other to "spam". It should be easy for you to figure out whether "spam" is denoted by 0 or 1 (hint: look at the frequency of occurrence of certain cue words in examples of class 0 and class 1). There are 1500 training cases and 3101 test examples. Train the classifiers on the training set.

   (a) [10 points] Implement the Maximum Likelihood Naive Bayes classifier with Laplacian smoothing illustrated in class and train it on the training set. Report the classification error (i.e., the fraction of examples misclassified) obtained on the training set as well as on the test set. **Note:** Special care must be taken here to avoid underflow problems in the computation of $p(x|y)$. This probability, being the product of many numbers less than one, is in general a very small number. You will have to find a way to compute the class posterior probabilities without explicitly calculating the small $p(x|y)$ numbers (hint: use logarithms).

   (b) [4 points] Using the learned Naive Bayes model, compute $p(y = 0|x_j = 1)$ for all features $j = 1, \ldots, 48$. Based on these probabilities, list the 6 words that are most indicative of a

message being "spam", and the 6 words most indicative of a message being "ham" (the words associated to the binary features can be found in file "spambase_names.txt").

(c) [10 points] Write code to learn a spam filtering decision tree with binary decision tests of the form ($x_j = 1$?) at each node. You can use the code outline provided in `DecisionTree.m` and `DT_recursive.m` to implement the learning algorithm. Use the entropy as a measure of impurity. Choose at each node the test yielding the maximum decrease of impurity. Stop splitting at a node if the number of training examples assigned to its decision region is less than $C \cdot m$, where m is the training set size and $C \in \{0.005, 0.01, 0.05, 0.1\}$. Plot the training error and the test error as functions of C in a single figure.

(d) [4 points] Implement a random forest consisting of 11 independently learned random decision trees. At each node, consider a subset of 3 features randomly selected from all features yielding non-degenerate partitions (a partition is degenerate if one of the regions of the partition contains 0 training examples). Obviously, use all the features producing non-degenerate partitions if these are fewer than 3. Use the same impurity measure and the stopping criterion as above, but now with $C = 0.01$. Predict using the most voted class. Report the errors on the training set and the test set.

(e) [2 points] Now predict by choosing the class having largest average class posterior (i.e., for each test example, average the class posteriors of the 11 leaves reached for the input example and pick the class having largest average posterior value). What is test set error? And the training set error?

4. *Problem for graduate credit only: if you are enrolled in COSC74 (instead of COSC174) you do not need to solve this problem.*
[20 points]

(a) [12 points] Implement the k-Nearest Neighbors classifier (kNN) based on Euclidean distance and evaluate it on the enclosed Parkinsons dataset *parkinsons.mat*, which we obtained from the UCI Machine Learning Repository (the data is formatted as in the previous problem). Look at *parkinsons.names* for a description of the features. The objective is to recognize healthy people from those with Parkinson's disease using a series of biomedical voice measurements. Report the test error obtained with $k = 1$.

(b) [5 points] Plot the 5-fold cross-validation score (measured as misclassification rate) as a function of k, for $k \in \{1, 3, 5, 7, 9, 11, 13\}$.

(c) [1 point] Now try measuring the leave-one out cross validation error for the same values of k. Is the shape of this plot any different from the curve generated using 5-fold cross-validation?

(d) [2 points] Plot the test error for $k \in \{1, 3, 5, 7, 9, 11, 13\}$. Is this curve more similar to the 5-fold cross-validation score or to the leave-one out cross validation score? Why?