COEN 281, Homework 2 - Linear Classifiers
Due: Tuesday, November 4

Please turn in a paper copy in class, or hand-deliver it to Apryl Roberts and have her time-stamp it. Email should only be used as a last resource. Work turned in d days late is graded and the grade is multiplied by (1 – d/10) if d ≤ 5, and 0 otherwise.

Work is to be done in groups of 2. Partner will be assigned randomly for each project. You must submit a confidential 1-to-5 (1=Poor; 5=Good) rating of your partner's contribution to the project. This rating will make 15% of the project's grade. Students with an average rating below 3 at the end of the quarter will have to submit himself/herself to a final exam. Please send an email to the instructor with the subject "HW2 – Group #," and the name of your partner and grade in the message body.

1. The file "az-5000.txt" contains 5000 lowercase character samples that have been preprocessed as in HW1 – i.e., raw images were scaled to fit in a 128x128 box and re-sampled to keep 9 coordinate pairs only. Additionally, the coordinate's values were further normalized to lie between 0.0 and 1.0. Each character is thus represented by 18 real values – i.e., $\mathbf{x}^t = [x_1, y_1, x_2, y_2, \ldots, x_9, y_9]$ and $x_i, y_i \in [0,1]$. The class labels are given by the first value in each row.

a. Use the *read.table* command to load this data into R. Make sure you set the '*header*' option. Check by printing the first and last rows… it should look like this:

```
      char      x1      y1          x2          y2      x3       y3          x4          y4
1        n 0.1875 0.140625 0.09375 0.515625  0 0.8828125 0.1796875 0.5078125
             x5          y5       x6          y6       x7       y7          x8          y8
0.4609375 0.140625 0.640625 0.109375 0.515625 0.5390625 0.3828125 0.8828125
             x9          y9
0.671875 0.9765625

      char         x1          y1       x2          y2          x3          y3       x4      y4
5000     e  0.2265625 0.125 0.4921875 0.2890625 0.84375 0.2109375 0.84375  0
             x5          y5          x6       y6       x7       y7       x8          y8
0.390625 0.0234375 0.15625 0.203125  0 0.5859375 0.1328125 0.953125
             x9          y9
0.5234375 0.9921875
```

b. Use the *sample* command to randomly select 80% of the data for training.

c. Use the *table* command to show the number of cases per class in the training data.

2. Linear Discriminant Analysis.

a. Use the *c()* command to create a vector of prior probabilities equal to 1/26 for each class.

b. Use the *lda* command to run linear discriminant analysis on the training data with the equal priors above. You may need to load the "MASS" package. In R, the syntax "*char* ~."

indicates the formula for our functional model – i.e., that we are trying to predict char (column one in the data) as a function of all the other variables.

c. Combine the functions *table* and *predict* to print a "confusion" matrix on the test data. This is a 26x26 matrix with diagonal elements equal to correct classifications and off-diagonal elements equal to mistakes. Which character had the best/worst performance?

d. What was the total accuracy on the test and train sets?

3. Logistic Regression. The file "credit_data.txt" contains information about the financial characteristics of 885 firms which applied for a bank loan. Use the sample command to randomly select 80% of the data for training. Use the table command to show the number of cases per class in the training and test data.

(a) Use the *glm* (with *family=binomial*) command to fit a logistic regression to predict which firms will go bankrupt. Report the table of coefficients from R with their p-values. What are the 4 most important predictor variables?

(b) Do their signs appear to be what you'd expect?

(c) Suppose that we predict a firm will go bankrupt if the predicted probability $P(Y = 1 \mid X = \mathbf{x})$ of bankruptcy is 0.5 or greater. Find the confusion matrix for such predictions on the test data.

4. Regularized Logistic Regression. The R package *glmnet* fits penalized logistic regression models using the Lasso penalty. We want to compare the regularized vs. the unregularized fit to the credit data.

(a) Use the *cv.glmnet* (with *family=binomial*) command to fit a regularized logistic regression to the same training data used in 3a (you may need a cast from data.frame to matrix and map y from 0/1 to -1/1). Plot the cross-validation curve. Explain the plot.

(b) The object returned by *cv.glmnet()* contains the value of the best lambda. Pass this value of lambda to the *coef()* function to retrieve the corresponding coefficient vector. Print the coefficients. Compare to your answer in 3a.

(c) Use the predict function with the same value of lambda to predict on the test data. Show the confusion matrix. Compare the accuracy with 3c.

5. Curse of Dimensionality. Ch.4. Problem 4.7.4.