Machine Learning 2009
(Home work 2)

June 13, 2009

- Due on Monday, June 24, before 10 a.m.
- Late submissions will not be accepted.
- Submit hard copy of the results, plots and your workings
- Email the codes for problem 1 to debrup.otro@gmail.com with ml09hw2 as the subject.
- You may save time if you use MATLAB for the computations and plots.
- Please do not hesitate to contact me if you do not understand the problems.

1. **[30 points] Naive Bayes**

   In this example we shall do some experiments on classifying emails into spam and non spam. The data required and a starter code are provided in the web page [1].

   The data provided is preprocessed. So you do not have to extract words from real emails. The training data is provided in the file MATRIX.TRAIN. MATRIX.TRAIN contains a matrix where the rows correspond to the emails and the columns correspond to the words present in the email, the (i, j)-th entry of the matrix corresponds to the number of times the j-th word occurs in the i-th email. As you can guess, this matrix would be very sparse (i.e., contain a lot of zeros) as all words will not be present in all emails. Thus, the matrix is stored in the file MATRIX.TRAIN with a special representation for saving space. The code readMatrix.m is a matlab code which contains a function to read the matrix. You do not have to understand what readMatrix.m does, you can just directly call it from your program, this function will read the file MATRIX.TRAIN and load its contents in a 'normal' matrix. The file TOKENS.LIST contains the whole list of the words with their indices.

[1] This data and the starter code is due to Prof. Andrew Ng, obtained from his page at http://www.stanford.edu/class/cs229/ps/ps2/

Your task would be to train a Naive Bayes classifier using the multinomial events model along with Laplace smoothing for classifying spam and non spam. You can start with the code nb_train.m given in the web-page. Once you have trained the classifier (i.e., computed all the relevant probabilities), you need to test the classifier on the data provided in MATRIX.TEST. You can start with the code given in the file nb_test.m.

Finally, you have to report the number of misclassified emails.

2. **[20 +20 points] Non Parametric Methods**

(a) Recall weighted linear regression discussed in class. For a regression problem we want to weight the cost function for different query points. We use a linear function as the local model. Specifically we define a cost function as

$$J(\boldsymbol{\theta}) = \frac{1}{2}\sum_{i=1}^{n} w_i(\boldsymbol{\theta}^T\boldsymbol{x}^{(i)} - y^{(i)})$$

where

$$w_i = \exp - \left(\frac{||\boldsymbol{x} - \boldsymbol{x}^{(i)}||^2}{2\tau^2}\right).$$

Using this cost function implement a locally weighted regression method to fit a function to the data in wr.txt. (Find out the predicted values for the following x values: -5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10,11,12, and plot the function using these values only). Use $\tau = 0.1, 3.0, 8.0$ and 10.0 and draw four plots of the fitted function for the different values of τ. Comment briefly on what happens to the fit when τ is too small or too large.

(b) In the course web-site you will find four files named, band1.irs, band2.irs, band3.irs and band4.irs. Download these files. These files contain a satellite image of Kolkata in 4 different bands. Size of each image is 512×512. Using these files one can create a data of four dimension, where each four dimensional point will represent a pixel. Your task would be to classify these 512×512 pixels into two categories land and water. For this purpose you will have to use a training set given in the file rsTraining.txt. This file contain 200 points, each point is 4 dimensional and the last column is either a 0 or a 1, which signifies whether the pixel is land or water. Using this data as a training data classify the whole data using a nearest neighbor (1-NN) classifier. Show your result as a black and white figure showing the land pixels in black and the water pixels in white.