CSL465/603– Machine Learning
Lab 2

Due on 9/9/2016 11.55pm

Instructions: Upload to your moodle account one zip file containing the following. Please do not submit hardcopy of your solutions. In case moodle is not accessible email the zip file to the instructor at ckn@iitrpr.ac.in. Late submission is not allowed without prior approval of the instructor. You are expected to follow the honor code of the course while doing this homework.

1. **You are allowed to work in pairs for this lab. However, this does not mean only one team member works on the lab, while the other only gets to add his/her name. Both the team members should know the inner workings of the code. I and the TAs reserve the right to question you if required. If you are found unable to explain the code, you will receive no points for the lab.**

2. This lab must be implemented in Matlab or Python.

3. A neatly formatted PDF document with your answers for each of the questions in the homework. You can use latex, MS word or any other software to create the PDF.

4. Include a separate folder named as 'code' containing the scripts for the homework along with the necessary data files. Ensure the code is documented properly.

5. Include a README file explaining how to execute the scripts.

6. Name the ZIP file using the following convention rollnumberhwnumber.zip

---

Naïve Bayes Classifiers

We will be using a subset of 2005 TREC Public Spam Corpus [1] containing 5000 training examples and 1000 test examples. This dataset is available in the zip folder of this lab (*nbctrain* and *nbctest*). Each line in the train/test files represents a single email in the following format.

email-id class-label word count word count …

The *email-id* is of the format /xxx/yyy. The *class-label* is either *spam* or *ham* (non-spam). *word count* pair indicate the number of occurrences of the word in the email. The data has been preprocessed to remove non-word characters such as punctuation marks.

1. Compute the prior probabilities $P(spam)$ and $P(ham)$ using the training data

2. Determine the vocabulary and compute the conditional probabilities $P(x_i/spam)$ and $P(x_i/ham)$ using the m-estimate, where $m = |vocabulary|$ and $p = 1/|vocabulary|$. Which are the 5 most frequently words indicative of a spam and a ham email?
3. Use these probabilities to classify the test data and report the accuracy. What difficulty do you face when computing the posterior probabilities? What do you propose to overcome this problem?
4. Vary the m parameter and study the changes in the accuracies as a function of m. What assumptions are we making when the value of m is very large compared to it being very small?
5. Now that you have built a classifier that can automatically detect spam emails, how would you modify your emails to beat the classifiers?

[courtesy: Pedro Domingos]

Linear Ridge Regression

We will be implementing Linear Regression to predict the age of Abalone (is a type of snail). The data set is made available as part of the zip folder (linregdata). You can read more about the dataset at the UCI repository [2]. We are primarily interested in predicting the last column of the data that corresponds to the age of the abalone using all the other attributes.

1. The first column in the data encodes the attribute that encodes-female, infant and male as 0, 1 and 2 respectively. The numbers used to represent these values are symbols and therefore are not ordered. Transform this attribute into a three column binary representation. For example, represent female as (1, 0, 0), infant as (0, 1, 0) and male as (0, 0, 1).
2. Before performing linear regression, we must first standardize the independent variables, which includes everything except the last attribute (target attribute) - the number of rings. Standardizing means subtracting each attribute by its mean and dividing by its standard deviation. Standardization will transform the attributes to possess zero mean and unit standard deviation. You can use this fact to verify the correctness of your code.
3. Implement the function named mylinridgereg(X, Y, lambda) that calculates the linear least squares solution with the ridge regression penalty parameter lambda (λ) and returns the regression weights. Implement the function mylinridgeregeval(X, weights) that returns a prediction of the target variable given the input variables and regression weights.
4. Before applying these functions to the dataset, randomly partition the data into a training and test set. Refer to the partition fraction as frac. If we want to use a 20%/80% training/testing split, then the value of frac will be 0.2. Now use your mylinridgereg with a variety of λ values to fit the penalized linear model to the training data and predict the target variable for the training and also for the testing data using two calls to your mylinridgeregeval function.
5. Implement the function meansquarederr(T, Tdash) that computes the mean squared error between the predicted and actual target values.
6. Pick a value for λ and examine the weights of the ridge regression model. Which are the most significant attributes? Try removing two or three of the least significant attributes and observe how the mean squared errors change.
7. Let us now try to answer two questions
    a. Does the effect of λ on error change for different partitions of the data into training and testing sets?

b. How do we know if we have learned a good model?

To answer these questions, modify your code to perform the following steps.

a. For different training set fractions, repeat 100 times
    I. Randomly divide data into training and testing partitions.
    II. Standardize the training input variables.
    III. Standardize the testing input variables using the means and standard deviations from the training set.
    IV. For different values of lambda
        i. Fit a linear model to the training data for the given lambda
        ii. Use it to predict the number of rings in the training data and calculate the mean squared error (MSE)
        iii. Do this again, using the same linear model applied to the testing data.
b. Calculate the average mean squared error over the 100 repetitions for each combination of training set fraction and lambda value

8. To see if the training set fraction affects the effect of lambda on error, plot the effect in multiple graphs, one for each training set fraction, by building the following figure. Make one figure of multiple graphs, one for each training set fraction, each graph being a plot of the average mean squared training error versus λ values and a plot of the average mean squared testing error versus λ. To enable the comparison across graphs, force each graph to have the same error (y axis) limits. You will find subplot, plot, hold on and ylim Matlab functions useful for plotting these graphs.
9. The figures provide some insight, but is not very clear right? So let us draw two more graphs. In the first graph plot the minimum average mean squared testing error versus the training set fraction values. In the second graph, plot the λ value that produced the minimum average mean squared testing error versus the training set fraction.
10. So far we have been looking at only the mean squared error. We might also be interested in understanding the contribution of each prediction towards the error. Maybe the error is due to a few samples with huge errors and all others have tiny errors. One way to visualize this information is to a plot of predicted versus actual values. Use the best choice for the training fraction and λ, make two graphs corresponding to the training and testing set. The X and Y axis in these graphs will correspond to the predicted and actual target values respectively. If the model is good, then all the points will be close to a 45-degree line through the plot.
11. Include all the plots and your observations in the report.

An important aspect of machine learning is reproducibility of the results presented in a paper/report. Therefore, we will run your code to see if the results are closely matching with what you have presented in the report. Any deviation beyond a reasonable threshold will be considered as fudging of results and will invite severe penalty.

Reference

[1] http://plg.uwaterloo.ca/~gvcormac/treccorpus/about.html
[2] http://archive.ics.uci.edu/ml/datasets/Abalone