CSL603– Machine Learning
Lab 2

Due on 18/9/2018 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. You are expected to follow the honor code of the course while doing this homework.

1. This lab has to be attempted individually.

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 rollnumber1_hwnumber.zip

---

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 [1]. 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, validation and test set. Let us fix the test set to be 20% of the training set. Randomly sample 20% of the instances into the test set. For the remaining data we will create training/validation set partitions of varying size. Refer to the partition fraction as frac. If we want to use a 20%/80% training/validation 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. Let us now try to answer two questions
    a. Does the effect of λ on error change for size of the training set?
    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 validation 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 (and standard deviation) over the 100 repetitions for each combination of training set fraction and lambda value

    Note: Start with a value of frac that results in only 100 instances in the training set. We will not be using the validation set for these experiments.

7. 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.
8. 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.

9. 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.
10. Include all the plots and your observations in the report.

Regularized Logistic Regression

In this exercise, you will experiment with regularized logistic regression and linear discriminants to predict whether credit card can be issued to an individual. As the research manager of the bank you have characterized each individual using two attributes x1 and x2. From these attributes, you would like to determine whether the credit card application of an individual should be accepted or rejected. To learn the models, you have a dataset of past credit card applications made by individuals and their outcomes. This is available as credit.txt in the zip file.

1. Plot the dataset using different colors for the two classes.
2. Implement regularized logistic regression that uses Gradient Descent and Newton-Raphson method as the optimization method. Choose the initial values of w in the range [-0.1, 0.1]. For a fixed set of iterations, comment on the performance of both the optimization routines.
3. Is the data linearly separable?
4. Logistic regression models only linear decision boundaries and therefore will not perform well on this dataset. One way to fit data better is to create more features for each data point. Implement the function featuretransform(X, degree) that takes the data and highest degree of polynomial terms of the input attributes x1 and x2 to create higher order polynomials of the input attributes. For example, if degree = 4, then the transformed data point will contain 15 attributes. We hope that this type of transformation helps to model the data better. Identify an appropriate degree of the transformation that results in the optimal performance. You can use either of the two optimization routines that you have implemented for this part of the assignment. Describe the processes (along with the evidence) that made you decide on the appropriate degree of the transformation.
5. Plot the non-linear decision boundary that separates the two classes learned by the classifier in the previous step.
6. Vary the value of the regularization parameter λ, and observe changes in the decision boundary. Include in the report one figure each depicting under fitting and over fitting along with the corresponding value of λ.

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://archive.ics.uci.edu/ml/datasets/Abalone