CSE474/574: Introduction to Machine Learning (Fall 2013)

Report - Project 1: Regression

Introduction:

The project trains a regression model for the given data set from Microsoft LETOR 4.0 to predict the relevancy labels for query-url pairs from the feature values. The total data set provided is divided into 3 sets, one for training, the second for validation and the last for testing. The regression model is first developed and trained, followed by which the complexity, regularization constant and hyper parameters are varied to obtain an acceptable validation error. After finding the ideal values of complexity, regularization and the hyper parameters, the model is evaluated against the test data to obtain the test error. The purpose is to reduce this error as much as possible. The following sections explain in detail what has been done under every part of the project.

Training:

Training process starts with the construction of the design matrix. The regression model developed is a linear model with the Gaussian basis function.

$$\varphi_j(x) = \exp\{ - (x – \mu_j)^2 / 2s^2 \}$$

The hyper parameters for this Gaussian basis function are computed from the given training data set. The total training dataset is divided into M-1 x D sets (M is assumed complexity and D is number of features). Then the mean and standard deviations are computed for all these sets. These are used as the hyper parameters to train the model.

$$\begin{bmatrix} \varphi 0(x1) & \varphi 1(x1) \cdots & \varphi M-1(x1) \\ \vdots & \ddots & \vdots \\ \varphi 0(xN) & \varphi 1(xN) \cdots & \varphi M-1(xN) \end{bmatrix}$$

Once the design matrix has been constructed using the Gaussian functions that take the computed hyper parameters, the weights are computed.

$$w^* = (\varphi^T \varphi)^{-1} \varphi^T t$$

These are the weights for the particular assumed complexity. Then the error value is computed which is followed by the computation of the RMS error.

$$E(y,t) = (\varphi w - t)T (\varphi w – t)$$

Once the RMS error is obtained, it can be concluded that the regression model is complete and now the complexity parameter is varied through a set of intuitive values and their RMS errors are computed. A graph is plotted with these RMS errors against the model complexity (Figure 1). This verifies the fact that increasing complexity decreases the error for the training data.

The training is done using the train function (train.m script). This function takes in the training data, the size of the training data, assumed model complexity and regularization constant as input parameters and gives the hyper parameters means, standard deviations and also the normal and regularized weights.

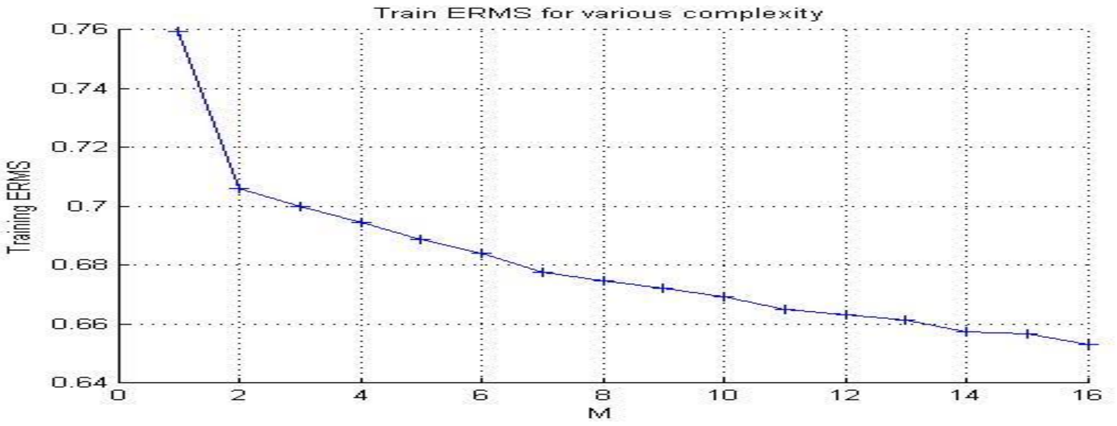

Figure 1

Validation:

Validation is the process of choosing the right model complexity, regularization constant and hyper parameters that will avoid over fitting and produce an acceptable RMS error for the validation data set. First the model is trained for various values of complexity without applying any regularization. This will result in over fitting (Figure 2) for the complexity parameters which had earlier given a small error for the training data set.

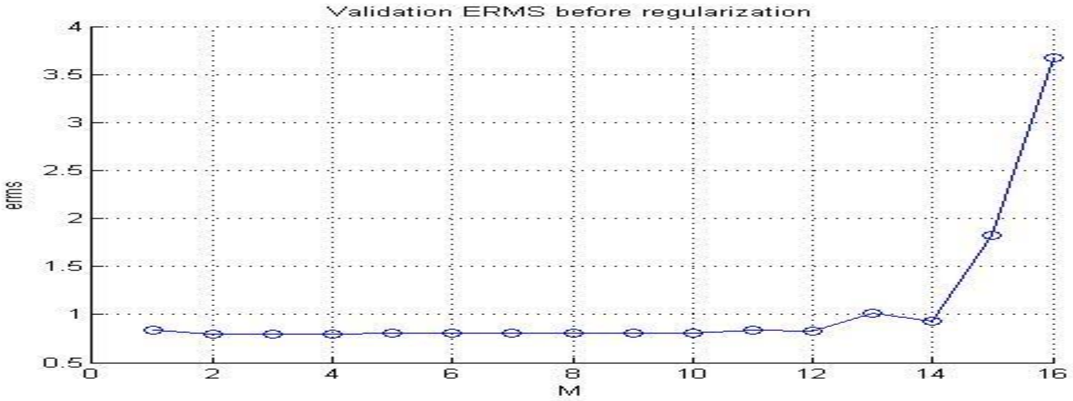

Figure 2

So, in order to avoid over fitting, a regularization factor is added to the computation of weights and also in the error value computation.

$$w^{*} = (\varphi^{T}\varphi + \lambda I)^{-1} \varphi^{T} t$$

$$E(y,t) = 0.5((\varphi w - t)T (\varphi w - t) + \lambda w^{T}w)$$

This regularization reveals the complexity for which the value of RMS error is the lowest (Figure 3) for the given model. **Therefore, the complexity M is fixed at 5**.

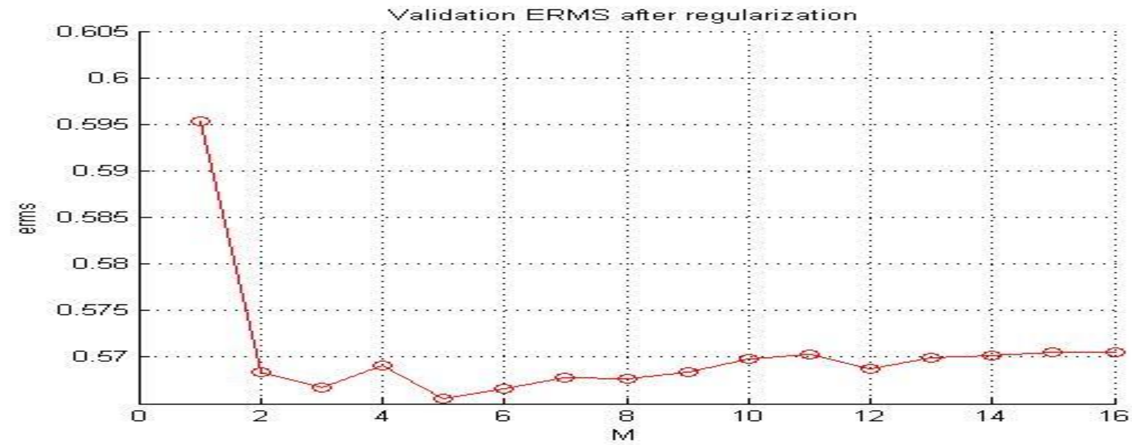

Figure 3

Now there is a need to fix the value of regularization to avoid any form of over regularized model. This is done by fixing the value of complexity and plotting the RMS error for several possible regularization constants (Figure 4). This gives the following graph from which it is **concluded that the regularization constant lambda is taken as 14.**

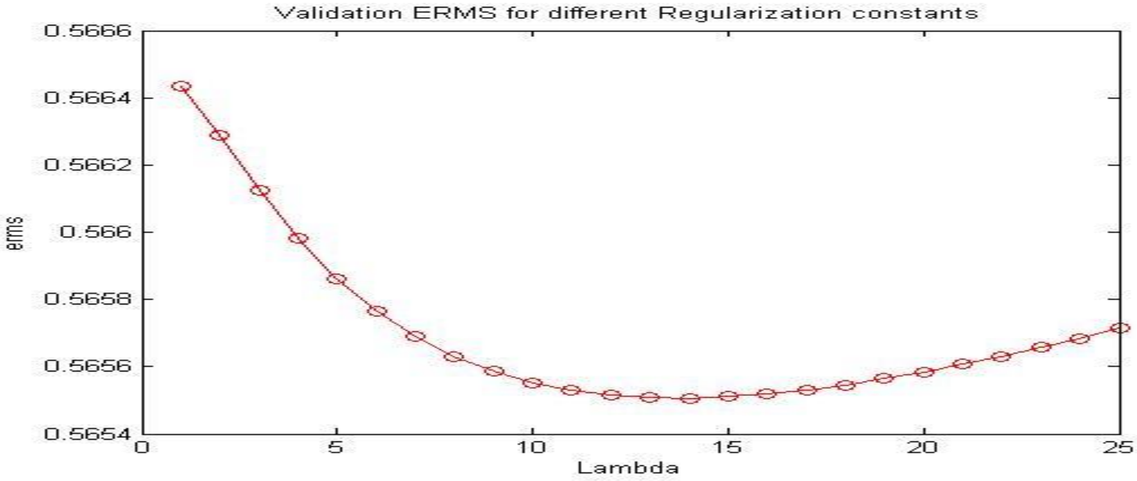

Figure 4

Validation is done in the predict.m script. The process of validation first fixes a value of model complexity and trains the model for the given regularization constant and obtains back the hyper parameters and the regularized weights. With these the design matrix for the validation data set is computed and the error function followed by the RMS error is calculated. This process fixes the M, lambda, means and standard deviations.

Testing:

The process of testing is to take in the complexity(M), regularization constant(lambda), means and standard deviations and compute the RMS error for the test data set. It essentially evaluates the model for its correctness. **The test RMS error is found to be 0.525181.**

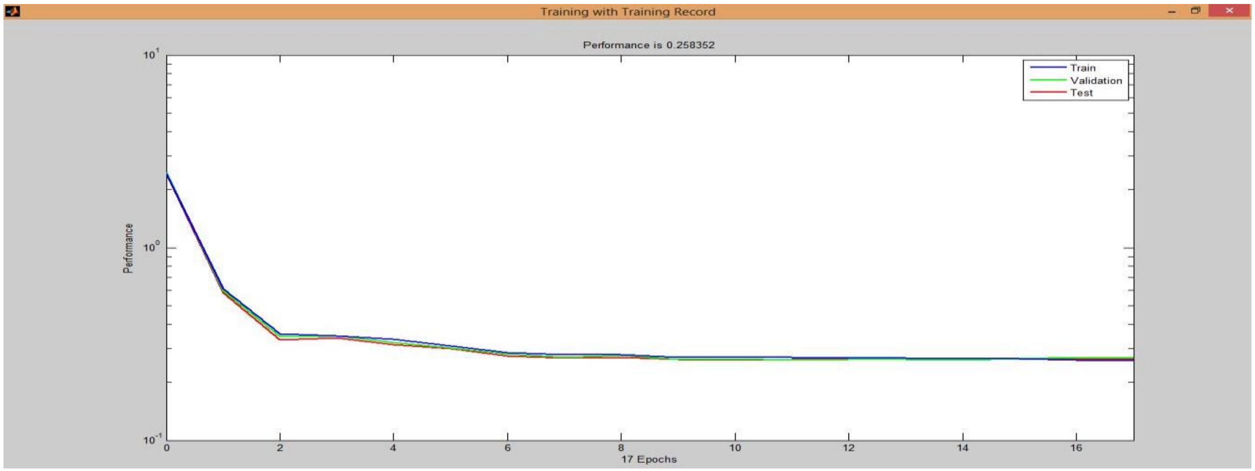

Figure 5

Predicted target labels are thus evaluated by finding the above error. Since the model that is developed is a linear model which is very simplistic, the error is quite large. **The model is thus verified using the Neural Networks toolbox available for MATLAB. And the toolbox gives a test error comparable with the error obtained by the linear model** (Figure 5)**.** [This value varies for different runs of the nn_model]

Execution Steps:

[Project files: project1_data.mat, train.m, predict.m, nn_model.m]

train and nn_model are functions.(See last paragraph of every phase which explains the input and output parameters to the functions)

predict.m is a script and project1_data.mat is the data needed to perform regression.

1. Run the predict.m script. (This will load the data, train the model, fix the parameters to get best results and test results against the standard neural networks toolbox)
2. If the graphs need to be verified
   a. Uncomment the graphs at the end of predict.m script and also at the end of the train.m script.
   b. Assign the limit value to 16 to see over fitting and regularization of the over fitted model.

Conclusion:

The project implemented a simple linear regression model and tested the model against the Microsoft LETOR 4.0 data to predict the relevancy values given the set of features for a query-url pair. Some of the key statistics obtained from the project:

Model complexity M = 5; Regularization constant lambda = 14; Test error rms_lr = 0.523181.