Project Report

Regression on Page Relevancy

Amogh Antarkar

Person # 50134359

Introduction

The motivation for linear basis comes from the fact that if a data set has a very few features (2-3) it is possible to plot the points and their output in 2-D or 3-D and find the best curve going through them to obtain a generalization. But if the number of features increases it is very difficult to obtain a fitting curve in such a high dimension. This problem can be solved if we use a function that maps n features to m functions which can be easily solved using Linear Regression. Such functions are known as basis functions. These basis functions are calculated in a number of ways.

Project Overview

The Project is implemented in MATLAB and the data set used for the learning purpose is Microsoft LETOR. There are two techniques used for learning one is Maximum Likelihood closed form solution and another one is the Maximum Likelihood with Gradient Descent.

Objective

This project is to implement and evaluate several supervised machine learning approaches to the task of linear regression. The objective is to learn how to map an input vector x into a target value t using the model. The main objective of the system is to learn the parameters for tuning the regression function to minimize the errors. The error in this case is the difference between the estimated relevancy of a query and the given relevance score for the query.

Maximum Likelihood Closed Form Solution

The goal of the project was to predict the relevancy label for given query url pair using linear regression. For linear regression of more than one variable (in this case there are 46 variables) a basis function needs to be chosen. For such a high number of variables polynomial regression is not the correct option as the number of weights will be very high. Therefore, a basis function model from two possible options- Gaussian radial basis function and sigmoid basis function. Gaussian basis function gives better results in terms of error vs. Sigmoid.

This is the first learning technique implemented in this project. The objective is to learn w (weight vector) w = (w1,w2,w3….,wM) and the parameters to obtain the mapping of an input vector x to output y. Where M is the number of basis functions used. In this project the basis function $\Phi(x)$ is the Gaussian basis function. Suitable values of M, s, μ and λ are tuned to obtain minimum error. The Error is the difference between the observed output i.e. the relevancy score of a query-document pair and the given relevance score.

- A basis function model from two possible options- Gaussian radial basis function and sigmoid basis function. Gaussian basis function gives better results in terms of error vs. Sigmoid. To implement Gaussian Basis Function we need values of mu and sigma i.e. mean and standard deviation. I calculated mean and standard deviation from the data itself. And added some error to the mus and sigmas for M basis functions.

- A certain value of M (No. of basis functions used) is taken and a value s (standard deviation) is taken. The value of μ is taken randomly such that all values in a column are same and there are M columns and number of rows is equal to the number of features. So each vector x is subtracted M times with M different μ for M different

values of $\Phi(x)$ such that 46 features are converted to M basis functions. So for each 46-dimensional vector x we calculate M Gaussian functions. This is known as a design matrix.

- The Design matrix is then used to calculate wml which is the Maximum Likelihood for the closed form solution.
- The wml (Maximum Likelihood solution) is then used to calculate the error. The error denotes the difference between the observed relevance and the expected relevance of the training data.

The mentioned steps are run continuously for different values of M, s and λ and a best combination of all these parameters are obtained for which the Error is minimum. So it is important to know the relation between the parameters and get a combination with minimum error for the same this is the training phase.

The validation dataset also is run through all the possible values of parameters and the minimum error giving parameters are obtained. The Error for the parameters for which training error is minimum is calculated on validation dataset and compared to minimum training error. If the errors are comparable then these values of parameters are learnt and then used to obtain the relevancy of the test data.

Stochastic Gradient Descent

Stochastic Gradient Descent is an another way of finding the Maximum Likelihood solution for a given data set. It is a type of sequential learning algorithm where the data points are considered one at a time. This technique is used where there are huge data sets and learning from such huge data set is time consuming. The algorithm goes through each dataset tuning the value of w accordingly and converges at a point for which the error obtained is minimum. Following steps are involved in Stochastic Gradient Descent algorithm

The value of μ is taken randomly same as in Maximum Likelihood solution. The value of M, lambda and s are tuned accordingly to obtain minimum error.

Parameters Choice

The parameter μ is any random value taken from the data set.

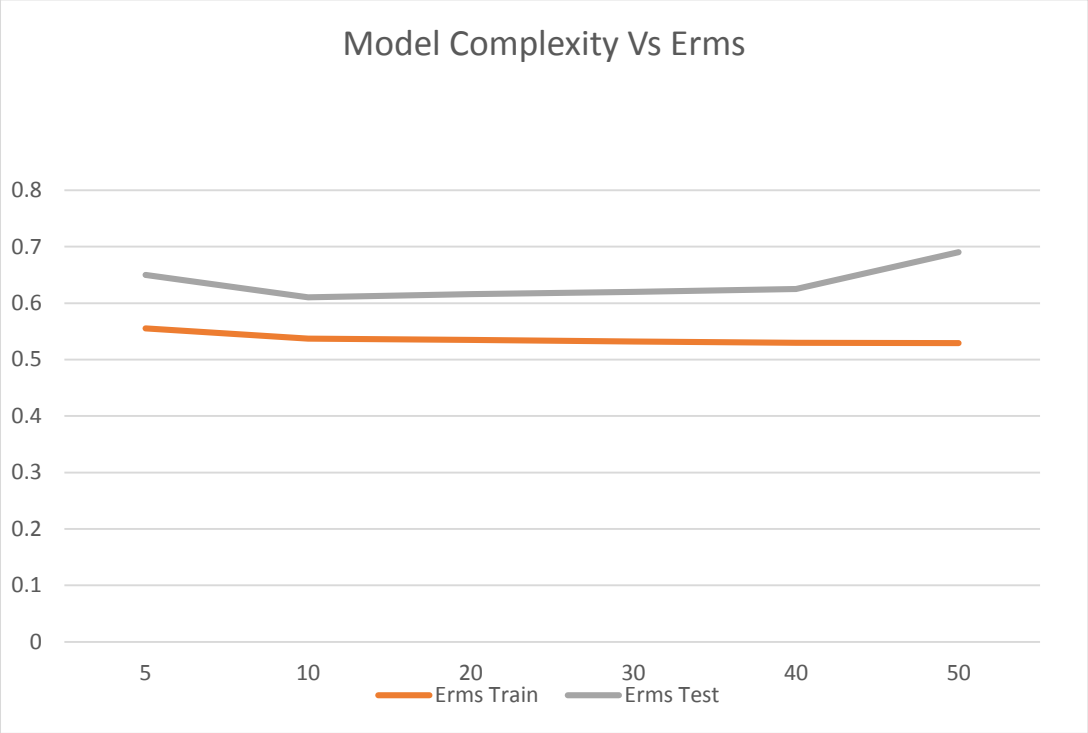
Obtain minimum error in both the techniques.

The optimal values are learned and used for the test data to predict the relevance of the test

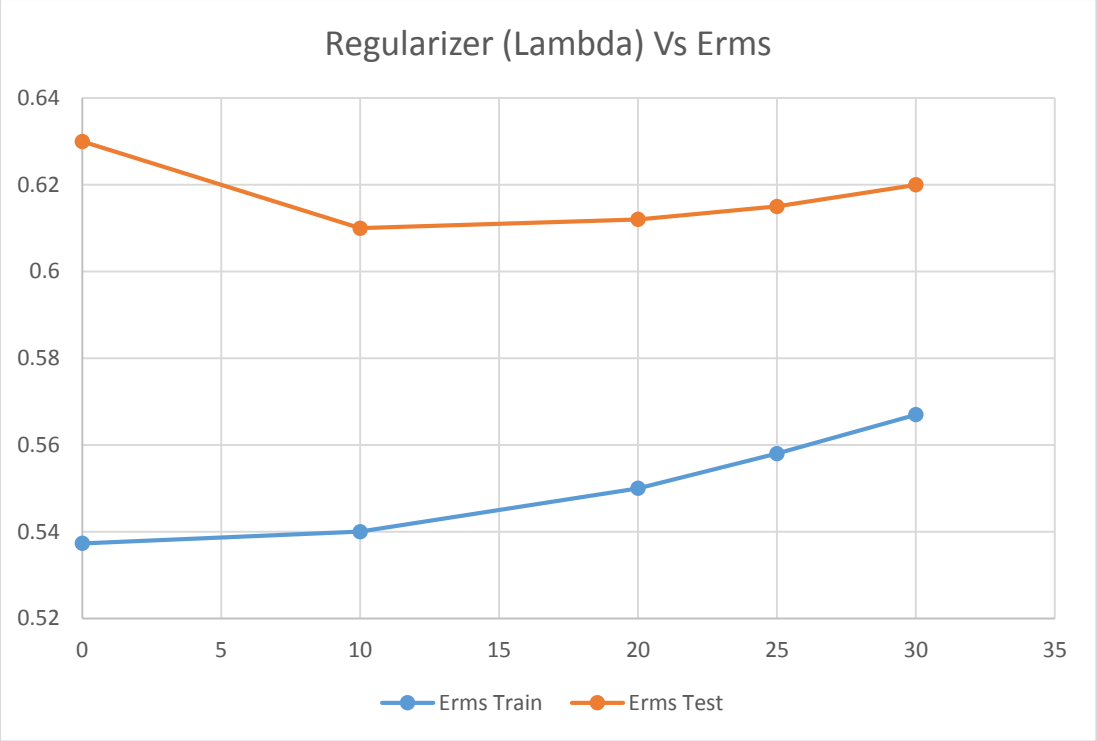
Hyper parameter Mu was created by taking the **mean across the input dataset** and further creating an error within an interval. This error created was added across the Mu. The Mu matrix was eventually replicated to adjust the dimensions using the function.

Parameters S was also created similarly by taking the variance across the input data. The variance squared was used further. Again an error addition technique was used. For the variance matrix the sizes of the data were replicated using the function. Again, the error was added to the Parameter dataset.

Evaluation of CFS and GD models



The graph shows that for validation and testing due to over fitting issue error goes on increasing after M>10.
The value goes on increasing significantly above M>40
So, the best value of M for validation and testing was M=10, I verified this value in Validation Process.

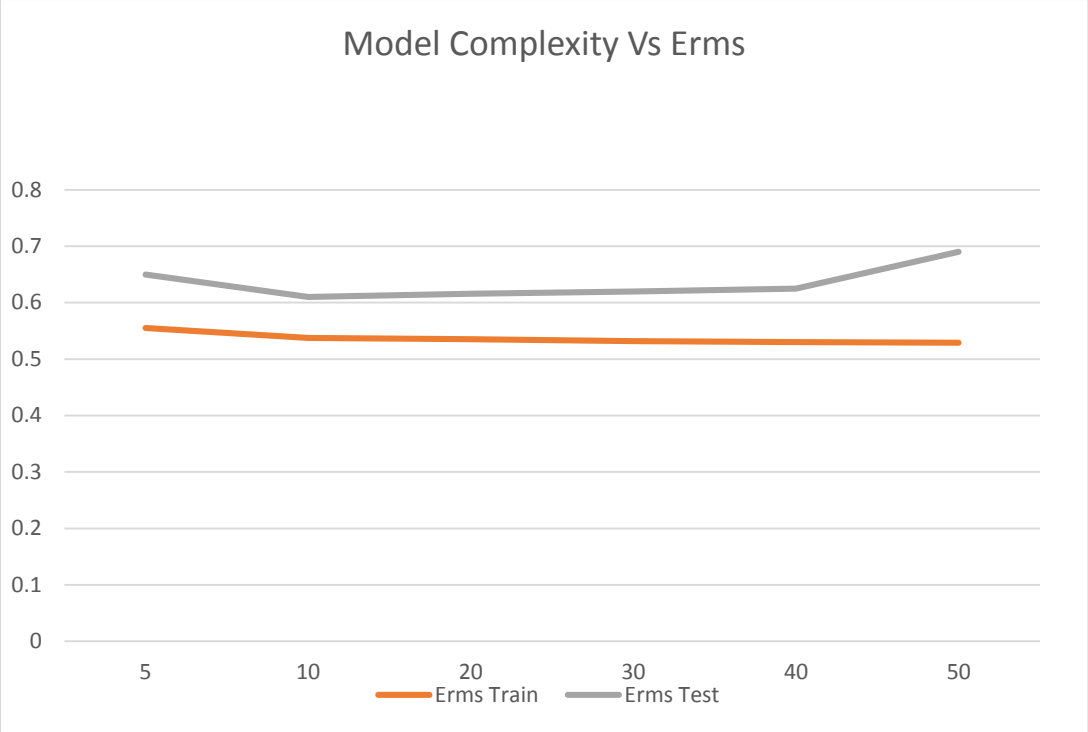


For M=10 the graph of Erms validation against lambda.

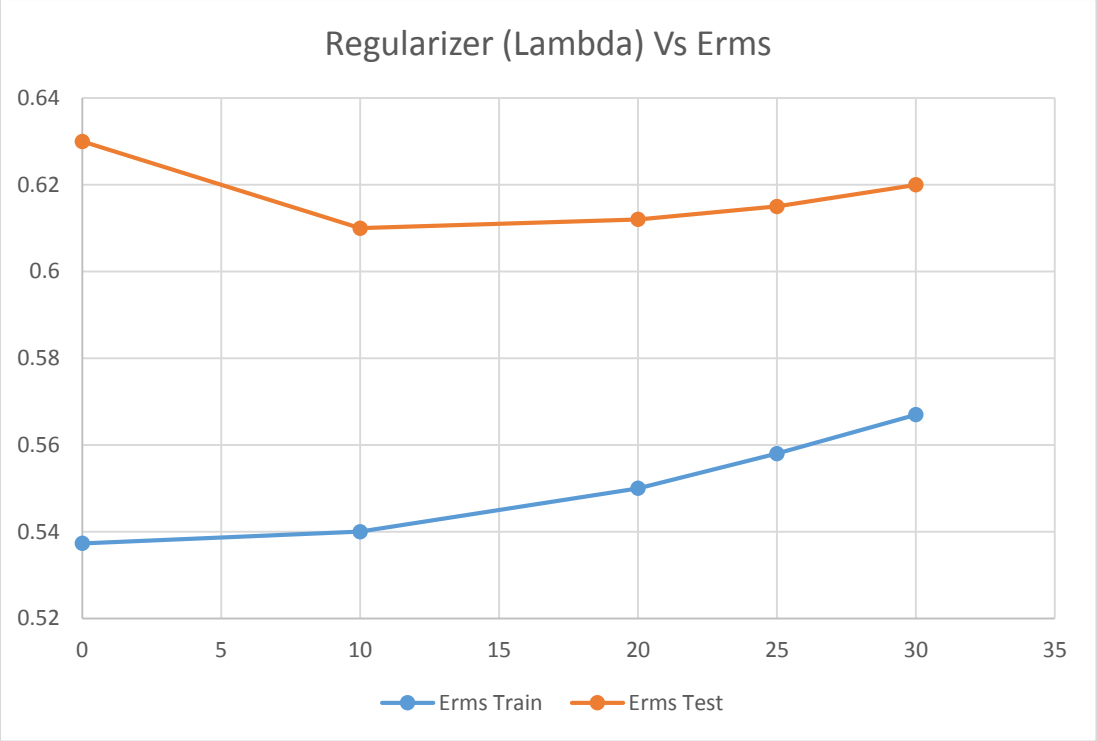
It's clear from the graph that lambda almost stops affecting the error after lambda=20.
So I chose lambda=20, as the error was least in that case.

Erms testing without regularization	Erms testing with regularization
0.63	0.61

Final Result:
The error on testing dataset is
Erms_ts= 0.61 for M=10, Lambda=20



Model Complexty	Erms Train	Erms Test
5	0.555	0.65
10	0.5373	0.61
20	0.535	0.616
30	0.532	0.62
40	0.53	0.625
50	0.529	0.69



regularization coefficient	Erms Train	Erms Test
0	0.5373	0.63
10	0.54	0.61
20	0.55	0.612
25	0.558	0.615
30	0.567	0.62