PROJECT REPORT

CSE574
INTRO TO MACHINE LEARNING

PROJECT 1.2

BY
YASH AHUJA
50245092
yashahuj@buffalo.edu

1. Introduction

This is a machine learning class project. The goal of this project is to understand the concept and how to perform Linear Regression and also to understand the concept of LeToR. Hence, the given datasets are LeToR datasets, and we perform two parts of Linear Regression:

- Using a closed-form solution
- Using Stochastic Gradient Descent

2. Problem and model description

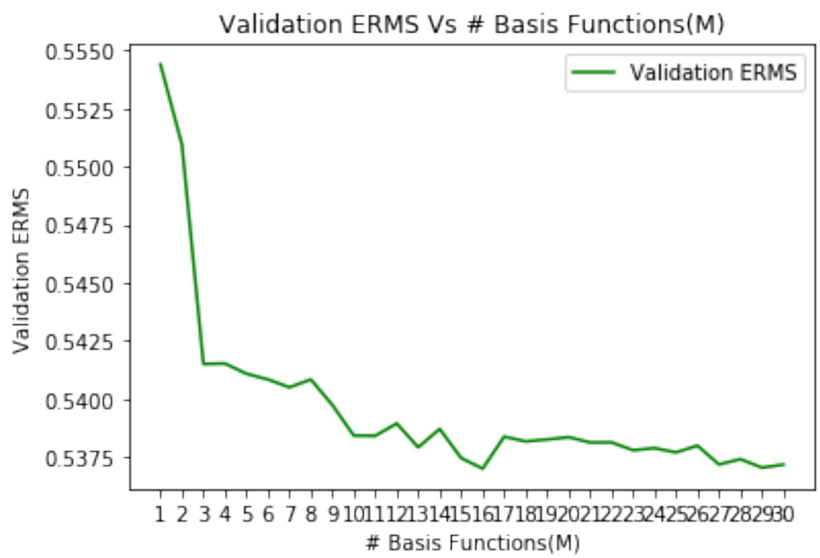
- The first part to this is to process and prepare the data, where the functions are defined and training, validation & testing sets are created from the raw data.
- In our case, 80% of the data is used as the training set, 10%, as the validation, and the last 10% as testing.
- The major hyper-parameters used are:
  - *M* – The number of radial basis functions. This parameter is used because we have chosen KMeans clustering, which involves other hyper-parameters like centroids(Mu).
  - *Mu* – This is the centroid of each cluster (used in KMeans), which is a feature variable itself.
  - *BigSigma* – This is the variance of the basis function
  - *Lambda* – Regularization term
  - *Eta* – Learning Rate
- We then tune these hyper-parameters on the validation set and then use the best values of the hyper-parameters to test it on the testing set.

3. Results

PLOTS

CLOSED-FORM SOLUTION

1. Varying M Values

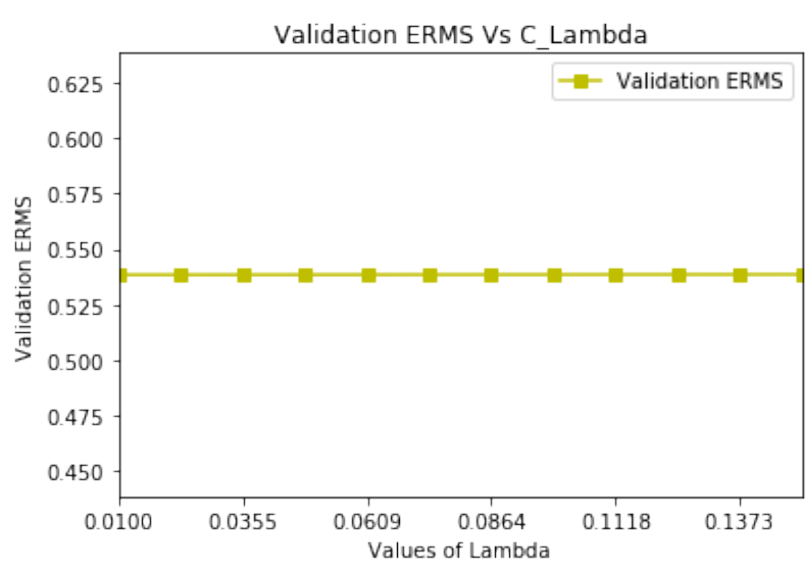


Note: The default value of hyper-parameter C_Lambda= 0.01 is taken in this case.

Comments:

As seen from the above graph, I have varied the hyper-parameter, M, and found the Validation Erms for each case. We can see that there is a gradual decrease in the ERMS as we increase the number of basis functions. The number of radial basis functions are used to fit the given function as perfectly as possible. Hence, higher the number of RBF, lesser the error, and better the accuracy. So, in this problem, the observed minimum error is with 16 basis functions. M = 16 will be taken as the best value while testing for the error at the end of the model.

2. Varying the C_Lambda values

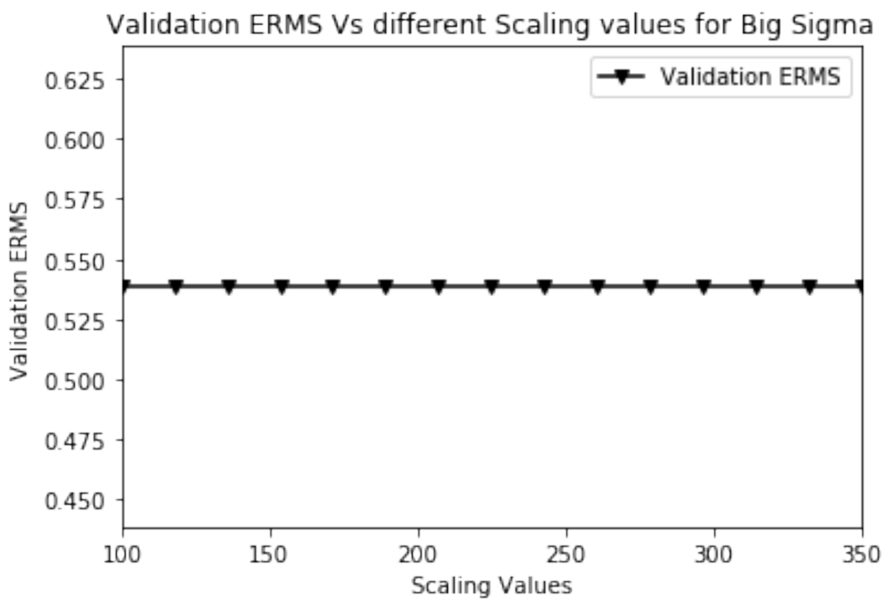


Note: The default value of hyper-parameter M=10 is taken in this case.

Comments:

As seen from the above graph, the Validation Erms value is constant as the regularization term(C_Lambda) increases. The regularization term is used to choose a model in a way so as to avoid overfilling it. If the value is very high, some data may be lost leading to a higher error with porr accuracy. But, it does not really matter what lambda value we choose in our case for testing the model as the graph is constant for many values. So I would stick with the default value of 0.01.

3. Varying the scaling values for Big Sigma



Note: The other hyper-parameters taken are default. As follows:

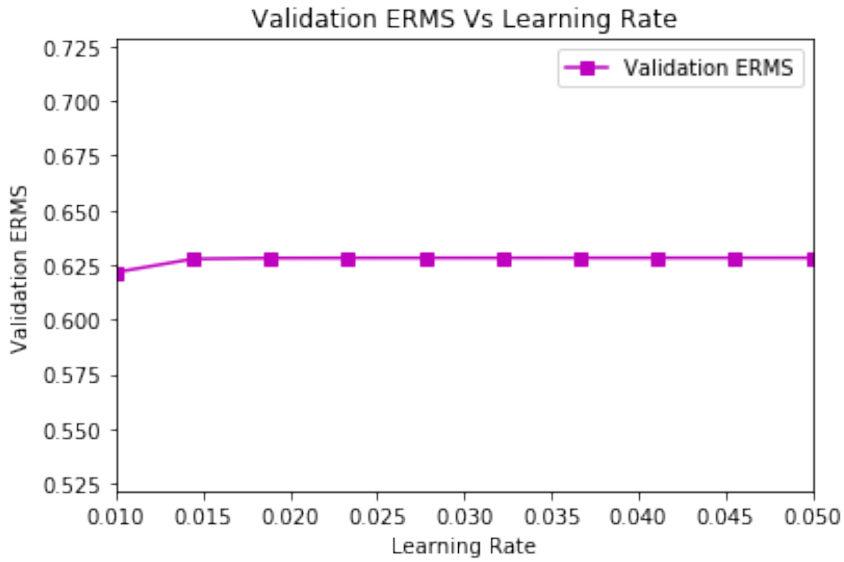
- M = 10, C_Lambda = 0.01, etc ,…

Comments:

As seen from the above graph, the ERMS remains constant as we increase the scale. This would obviously remain constant as the entire model will run on the same scale for every iteration. So, while testing the model at the end, we could put any scaling value, and it wouldn't change the result.

GRADIENT DESCENT

1. Varying the learning rate(eta)

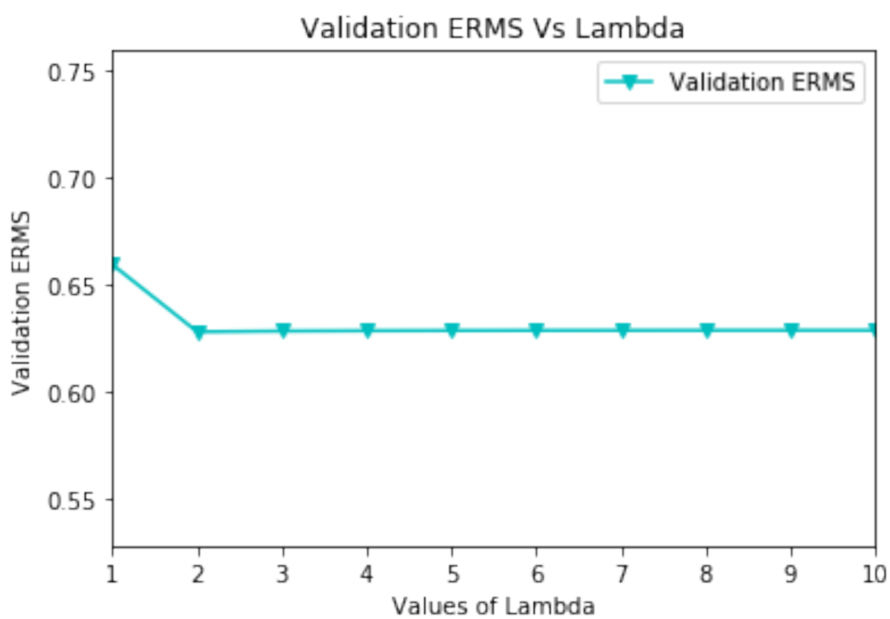


Note: The other hyper-parameter, Lambda = 2 , is taken at the default value.

Comments:

As seen from the above graph, the ERMS remains constant as we increase the learning rate. Learning rate is a hyperparameter controlling the weight adjustment of our model with respect to the loss gradient. If the learning rate is low, the number of steps taken to reach the local minima will increase, hence increasing the time taken. And if the learning rate is high, there is a risk of overshooting and never arriving at the local minima. Initially, at lambda value 0.01, the error is observed to be lower than normal. After that point, it increases till the learning rate value 0.015 and then remains constant. So, 0.01 can be selected as the best learning rate value.

2. Varying the Lambda (La)



Note: The other hyper-parameter, Learning Rate = 0.01, is taken at the default value.

Comments:

As seen from the above graph, the ERMS remains constant as we increase the Lambda value. Initially, with a very small value of Lambda, the error is higher than normal, but as lambda is increased the error decreases till it saturates approximately at 3. So, it can be concluded that the error obtained for the lambda value above 3 remains constant. It can be observed from the graph that the lambda value at the lowest point of error is 2.

OBSERVED TEST VALUES

	Closed-Form Solution	**Gradient Descent**
Test Value	0.6271127363319977	0.6229878691761276

The above solutions are the E-Rms values for the closed-form and gradient descent solutions, testing it with the best values of the hyper-parameters found from experimenting with the validation sets.
The best values of the hyper-parameters
Hence the error values in this model are very high ***62.7%*** and ***73.5%*** obtained from the closed-form and gradient descent methods respectively.

4. Conclusion

In this project, we majorly focused on performing linear regression on the LeToR dataset. Changing the hyper-parameters on the validation set to get best results on the testing set gives us better insights on analyzing data and using the basics of machine learning.