Learning to Rank using Linear Regression

Yashankit Shikhar
UBID: yshikhar

October 10, 2018

Abstract

The purpose of this paper is to propose a Machine Learning method to solve a problem which commonly arises in the Information Retrieval domain, known as Learning to Rank. Several hyper-parameters, such as the learning rate, number of basis function, regularization term, etc. are analyzed for their effect on the accuracy of the program. The performance of program for different values of the hyper-parameters is plotted with help of Jupyter Notebook and Spyder. Based on the analysis, the program is executed with obtained values of hyper-parameters.

1 Introduction

In this project, two approach to train a linear regression model, closed-form solution and stochastic gradient descent (SGD) is discussed. For this project the benchmark LeToR 4.0 supervised ranking data set has been used which contains 46 feature for learning to rank. In this data set, each row is a query-document pair. The first column is relevance label of this pair. The larger the relevance label, the better is the match between query and the document. A training data set of first 80% of data set is created and a testing data set of last 10% of data set is created. The program is tested on the testing data set.

2 Experiment

The aim of this project is to train a regression model based on query-doc pair data set, and predict the page relevancy labels for new docs. The following hyper-parameters are analyzed and their effect on the accuracy of the model is observed:

1. **Regularization Term** λ
2. **Number of Basis Functions M**
   The parameters dependent on the number of basis functions are:
   (a) **Centers for Gaussian radial basis functions** μ_j
   (b) **Spread for Gaussian radial basis functions** $\sum_j$
3. **Learning Rate** η^τ

3 Linear Regression Model

A linear regression function $\mathbf{y}(\mathbf{x}, \mathbf{w})$ is defined as,

$$y(x, w) = w^\top \phi(x) \tag{1}$$

where $\phi_j(x)$ is Gaussian radial basis function given by

$$\phi_j(x) = exp(-1/2(x - \mu_j)^\top (\sum_j)^{-1}(x - \mu_j)) \tag{2}$$

where $\mathbf{w} = (\mathbf{w}_0, \mathbf{w}_1, .., \mathbf{w}_{m-1})$ is a weight vector to be learned from training samples and $\phi = (\phi_0, \phi_1, .., \phi_{m-1})^\top$ is a vector of M basis functions. μ_j is the center of the basis function and $\sum_j$

decides how broadly the basis function spreads.

The objective is to minimize the sum-of-squares error,

$$E_D(w) = 1/2 \sum_{n=1}^{N} (t_n - w^\top \phi(x_n))^2 \tag{3}$$

where $\mathbf{t} = (\mathbf{t}_0, \mathbf{t}_1, .., \mathbf{t}_n)$ is the vector of outputs in the training data and ϕ is the design matrix.

3.1 Closed Form Solution

In closed form solution, derivative of objective function is set to zero and required parameters are obtained without resorting to iterative algorithm [1]. Closed form solution is calculated by,

$$w_{ML} = \lambda \mathbf{I} + (\phi^\top \phi)^{-1} \phi^\top t \tag{4}$$

The RMS Error is calculated for training, validation, and test data set using the following formula:

$$E_{RMS} = \sqrt{E(w*)/N_V} \tag{5}$$

3.2 Stochastic Gradient Descent

Gradient Descent is the process of minimizing a function by following the gradients of the cost function [2]. Stochastic Gradient Descent works on a single data point at a time. We start with a single random value of solution. We modify the solution for each data sample using:

$$w^{\tau+1} = w^\tau + \Delta w^\tau \tag{6}$$

where,

$$\Delta w^\tau = -\eta^\tau \nabla E \tag{7}$$

is called weight updates. η^τ is the Learning Rate.

Due to linearity of differentiation, we have

$$\nabla E = \nabla E_D + \lambda \nabla E_W \tag{8}$$

in which,

$$\nabla E_D = -(t_n - w^{(\tau)\top} \phi(x_n)) \phi(x_n) \tag{9}$$

$$\nabla E_W = w^{(\tau)} \tag{10}$$

4 Hyper-parameters

In machine learning, a hyper-parameter is a parameter whose value is set before the learning process begins. Given these hyper-parameters, the training algorithm learns the other parameters from the data [3].

4.1 Regularization Term λ

Regularization refers to the act of modifying a learning algorithm to favor “simpler” prediction rules to avoid over-fitting. Most commonly, regularization refers to modifying the loss function to penalize certain values of the weights being learned. λ is a hyper-parameter that adjusts the trade-off between having low training loss and having low weights [4].

4.2 Number of Basis Functions M

The number of basis function defines the number of clusters in k-means clustering. The centroid of these clusters form the μ_j matrix.

4.3 Learning Rate η^{τ}

Learning rate is defined in the context of optimization, and minimizing the loss function of a neural network. For this optimization problem, stochastic gradient descent is used where the model parameters are updated in a way to decrease the cost function [5].

5 Analysis

For analysis of the effect of various hyper-parameter, the model is fitted on the testing data to predict labels and compare it with original labels of testing data to determine the accuracy and RMS error. The effect of hyper-parameter on the accuracy and RMS error is shown with the help of graphs. The experiment is performed in two parts, first for closed form solution and then for stochastic gradient descent.

5.1 Analysis of Closed Form Solution

5.1.1 Regularization Term λ

The program is executed for different values of λ ranging from 0.001 to 5 and the accuracy and error of the program on the testing data is plotted for each value of λ (Fig.1).

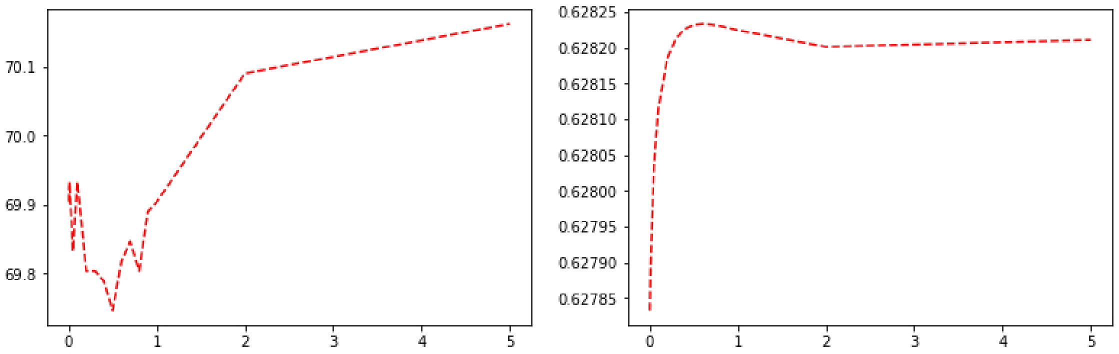


Figure 1: graph of accuracy percentage and error (y axis) vs λ (x axis) respectively

5.1.2 Number of Basis Functions M

The program is executed for different values of M ranging from 1 to 100 and the accuracy and error of the program on the testing data is plotted for each value of M (Fig.2).

5.2 Analysis of Stochastic Gradient Descent Solution

5.2.1 Regularization Term λ

The program is executed for different values of λ ranging from 0.001 to 5 and the accuracy and error of the program on the testing data is plotted for each value of λ (Fig.3).

5.2.2 Learning Rate η^{τ}

The program is executed for different values of η ranging from 0.001 to 0.2 and the accuracy and error of the program on the testing data is plotted for each value of η (Fig.4).

6 Results

In this experiment, a linear regression model was implemented using two methods, closed form solution and stochastic gradient solution. The accuracy and the error of both models are shown in Figure 5. Also, the effect of changing the following parameters on the accuracy of model were analyzed and following observation were made:

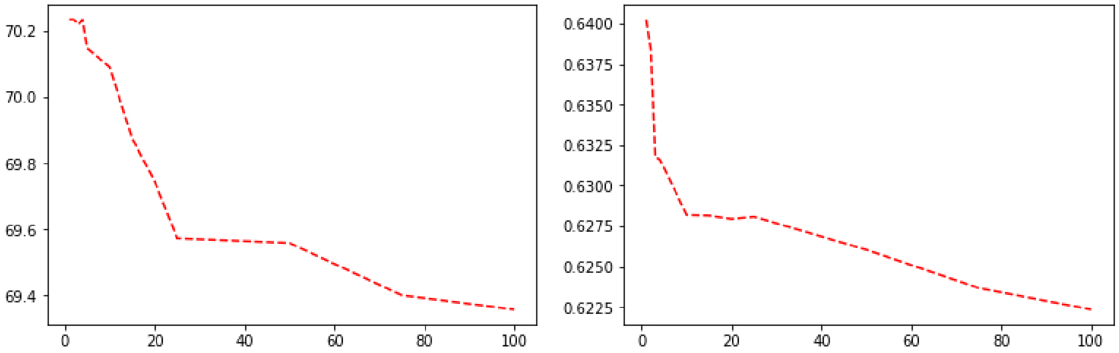


Figure 2: graph of accuracy percentage and error (y axis) vs M (x axis) respectively

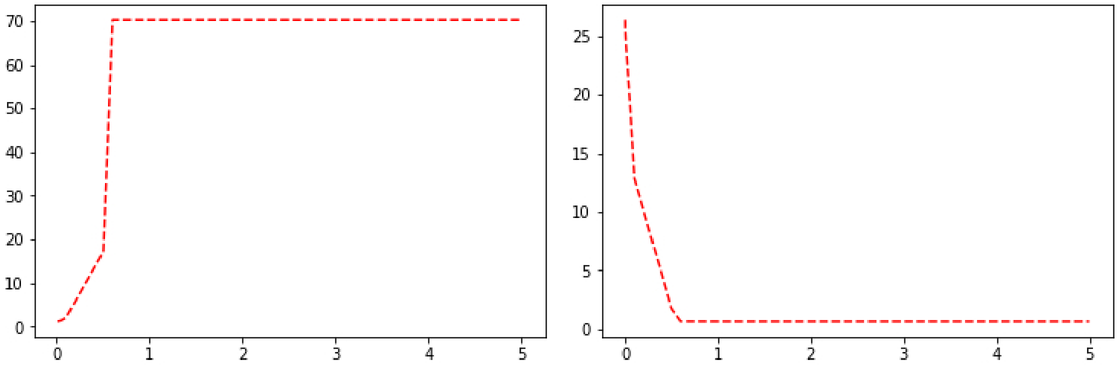


Figure 3: graph of accuracy percentage and error (y axis) vs λ (x axis) respectively

1. **Regularization Term λ:** It is observed that the for lower value of λ, the accuracy is low but it increases steadily and becomes constant after some values for both closed form solution and stochastic gradient descent. (Fig.1 & Fig.3) This can be fairly justified by the fact that when λ is 0, it results in over-fitting of the model due to which the error is high and accuracy is low. As λ increases the accuracy increases and stabilizes.

2. **Number of Basis Functions M:** It is observed that with increase in the value of M, the accuracy and error both decrease for the model. (Fig.2) The more the number of clusters, the lesser would be the variance in the clusters. Therefore, Big sigma would be lower, which means the basis function value would be higher.

3. **Learning Rate η^{τ}:** The accuracy steadily increased as the learning rate was increased but became constant after some values. On the other hand, error decreased and became constant

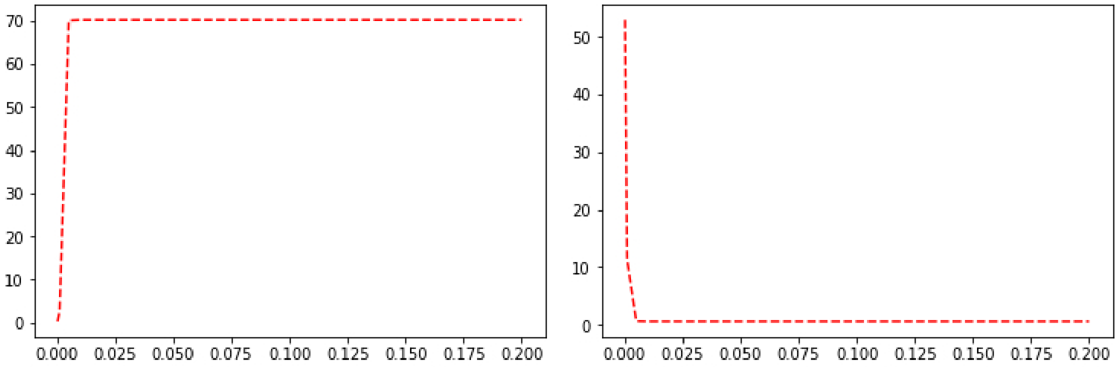
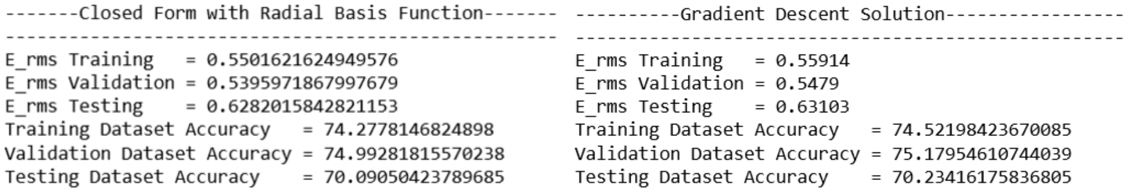


Figure 4: graph of accuracy percentage and error (y axis) vs η (x axis) respectively



```
-------Closed Form with Radial Basis Function-------  ----------Gradient Descent Solution----------------
----------------------------------------------------  -------------------------------------------------
E_rms Training   = 0.5501621624949576                 E_rms Training   = 0.55914
E_rms Validation = 0.5395971867997679                 E_rms Validation = 0.5479
E_rms Testing    = 0.6282015842821153                 E_rms Testing    = 0.63103
Training Dataset Accuracy   = 74.2778146824898        Training Dataset Accuracy   = 74.52198423670085
Validation Dataset Accuracy = 74.99281815570238       Validation Dataset Accuracy = 75.17954610744039
Testing Dataset Accuracy    = 70.09050423789685       Testing Dataset Accuracy    = 70.23416175836805
```

Figure 5: Accuracy percentage and Error for Closed Form Solution and Stochastic Gradient Descent respectively

after some values. (Fig.4) Ideally if learning rate continues to increase, the error will start to increase [6].

7 Conclusion

In this paper, two Machine Learning method is proposed to solve Learning to Rank problem, Closed Form Solution and Stochastic Gradient Solution. Several hyper-parameters, such as the learning rate, number of basis function, regularization term, etc. are analyzed for their effect on the accuracy and the root mean square error of the program and the performance of the program is observed. When the above conditions were met, accuracy of 70.1% & 70.2% was achieved on testing data set for Closed Form Solution and Stochastic Gradient Descent Solution respectively.

References

[1] Lecture 1: Linear Regression
`http://home.iitk.ac.in/m̃ithlesh/cs300/5.pdf`

[2] Stochastic Gradient Descent
`https://machinelearningmastery.com/implement-linear-regression-stochastic-gradient-descent-scratch-python/`

[3] Wikipedia:Hyperparameter (machine learning)
`https://en.wikipedia.org/wiki/Hyperparameter_(machine_learning)`

[4] Regularization
`http://cmci.colorado.edu/classes/INFO-4604/files/slides-6_regularization.pdf`

[5] What is the learning rate in neural networks?
`https://www.quora.com/What-is-the-learning-rate-in-neural-networks`

[6] Understanding Learning Rates and How It Improves Performance in Deep Learning
`https://towardsdatascience.com/understanding-learning-rates-and-how-it-improves-performance-in-deep-learning-d0d4059c1c10`