CSE 574 – Introduction to Machine Learning

Project 2.0 – Learning to Rank using Linear Regression

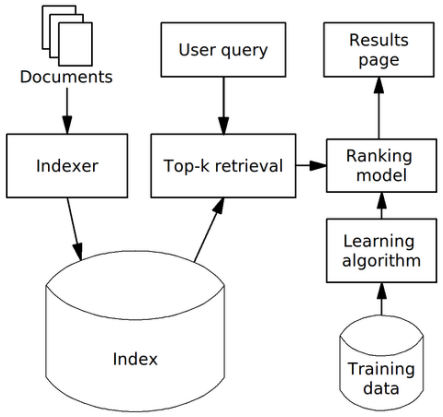
Asish Kakumanu
UB Person No :50288695
asishkak@buffalo.edu

Abstract

The Abstract of the project is to solve an Information Retrieval problem called Learning to Rank "**LeToR**" by applying Supervised Learning approaches to Linear Regression. In this Project, we train a linear regression model on the given Microsoft LeToR 4.0 dataset by using closed-form solution and Stochastic gradient descent and to compare both of solutions.

1 Dataset

Learning to Rank is the application of Machine Learning, in the construction of Information Retrieval systems. In Learning to Rank, All the documents are indexed and when a query is made for the relevant information, **Ranking model** ranks the relevant pages with the highest **Relevancy Index** / **Relevancy label**, thereby showing the relevant documents at the top and irrelevant documents at the bottom.



In this Project, we are using Microsoft LeToR - Supervised Ranking Dataset to perform both the tasks.

LeToR Dataset has pair of Input values ***x*** and target values ***t***

1. ***Input*** values are ***Vectors (Features)*** derived from a query pair document. It has 46 Features totally in each query-document pair.
2. ***Output*** target Value is a Scalar which is also a Relevance Label - ***1,2,3.*** Highest being the ***best fit/ better match*** between ***Document & Query.***

The dataset consists of 69624 rows of data where **each row is a query-document pair** and 46 Dimensions of feature vectors. Therefore, the dataset has a matrix dimension of **69624*46**. The First Column is **relevance label** of this pair, the second column is **query id**, the following are features, and the end of the row is comment about the pair which includes **document id**.

1.1 Rank Aggregation

Each query is associated with a set of input ranked lists. The task of rank aggregation is to output a better final ranked list by aggregating the multiple input lists. A row in the data indicates a query-document pair.

0 qid:10002 1:1 2:30 3:48 4:133 5:NULL ... 25:NULL #docid = GX008-86- 4444840....

In the above, query-document pair.

- First Column **0** represents the **relevance label** of this pair.
- Second Column is Query id.
- Following columns are ranks of the documents in input ranked lists
  - **2:30** means that the **rank** of the document is **30** in the **second** input list.
  - **Larger Rank** means **top positions** in Input Ranked list.
  - **NULL** means document doesn't appear in Ranked list.
- The end of the row is comment about the pair including document id.
- Larger the Relevance label, more relevant is the query-document pair.
- There are 46 Features which are required are useful for Learning to Rank.

1.2 Data partition

Now, the dataset is divided into parts.

- Training Set (80% of Dataset)
- Validation Set (10% of the Dataset)
- Testing Set (10% of the Dataset)

2 Linear Model

Our Linear Model function $y(x, w)$ has the form:

$$y(x, w) = w\hat{}T\phi(x)$$

W is the weight Vector learnt from training samples
ϕ is vector of M basis functions.

We consider $\phi_{0(x)} = 1$ to become a bias in the system. This parameter counterweighs for the difference in the avg. values of target vector in the training data with avg. of the basis function values.

3 Closed Form Solution Approach

The Closed form solution for linear regression is carried out using Gaussian basis Function which is given by:

$$\phi_j(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_j)^\top \Sigma_j^{-1}(\mathbf{x} - \boldsymbol{\mu}_j)\right)$$

Where μ_j is the center of basis function and $\sum_j -$ decides the spread of the basis function.

The design Matrix ϕ is of dimension 46*46 and M is the Basic Function.

$$\mathbf{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}.$$

Now, by using the design matrix ϕ we now calculate the closed form solution to the linear regression using gaussian bias function in the following form. Which is closed form solution aka. Sum of Squared Errors without Regularization. This quantity is known as Moore-Penrose pseudo-inverse of the matrix ϕ

$$w_{ML} = (\Phi^T\Phi)^{-1}\Phi^T t$$

Now, the above solution is not regularized. So, there is a lot of chance that solution might be over fitted. So, to prevent over fitting we add a regularization term to the function. The weight vector is learnt from training samples and is given by

$$\mathbf{w}^* = (\lambda\mathbf{I} + \mathbf{\Phi}^\top\mathbf{\Phi})^{-1}\mathbf{\Phi}^\top\mathbf{t}$$

This is the closed form solution which is regularized. I is the Identity matrix.

With regularized weight obtained from the above equation we can calculate the sum of squared errors, defined as

$$E_{rms} = \sqrt{\frac{2E(w*)}{N_v}}$$

Where w^* is the solution and N_V is the size of the test data.

4 Stochastic Gradient Descent Solution

We use gradient descent method to compute the weights to minimize the lowest mean error. Here, η is carefully chosen such that we don't observe any variation in convergence. Instead we can reduce the risk in the local minima by selecting larger learning rate in the beginning and reducing proportional to the time. Now, weight is calculated using the equation below:

$$w^{\tau+1} = w^{\tau} + \eta\left(t_n - w^{(\tau)(T_{\phi_n})}\right)\phi_n$$

With regularized weight obtained from the above equation we can calculate the sum of squared errors, defined as

$$E_{rms} = \sqrt{\frac{2E(w*)}{N_v}}$$

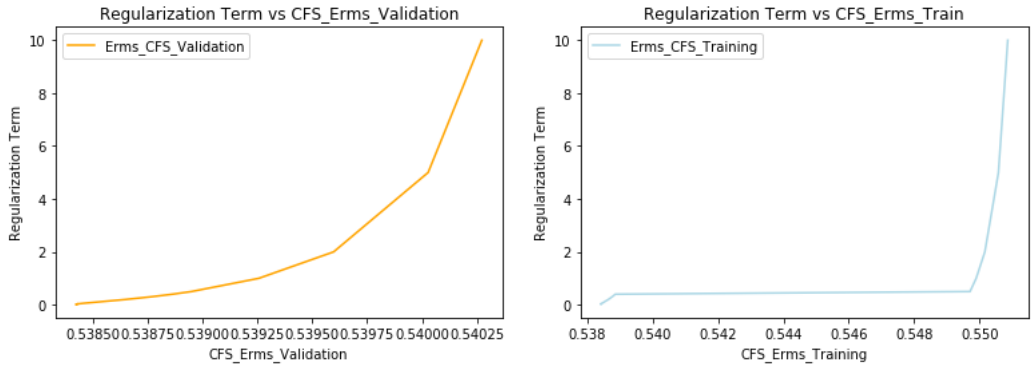
4 Evaluation

4.1 Closed Form Solution

Keeping Basis Functions **M** and Learning Rate η and changing just the regularization term. We get the below results for **Closed Form Solution**.

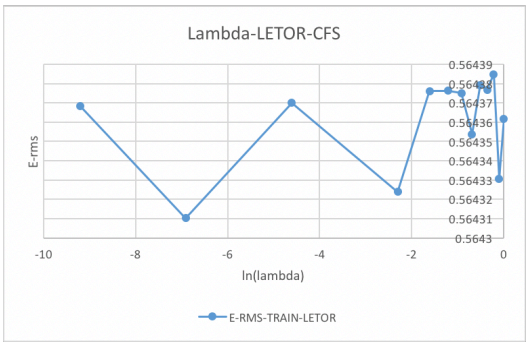
λ (Regularization Term)	Closed Form
0.03	E_rms Training = 0.5494694067137873 E_rms Validation = 0.5384281741391367 E_rms Testing = 0.6279788453842321
0.01	E_rms Training = 0.5494573662542529 E_rms Validation = 0.5384211743833384 E_rms Testing = 0.6278937255787389
0.02	E_rms Training = 0.549462918989187 E_rms Validation = 0.5384219372832513 E_rms Testing = 0.6279412477132231

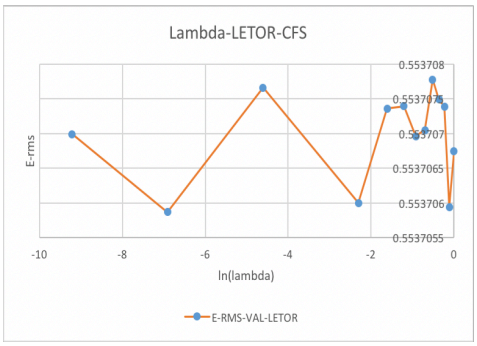
0.04	E_rms Training = 0.5494761182960914 E_rms Validation = 0.5384375274863877 E_rms Testing = 0.6280093314669583
0.05	E_rms Training = 0.5494827834801572 E_rms Validation = 0.538448736802394 E_rms Testing = 0.6280345564774603
0.2	E_rms Training = 0.5495684048344858 E_rms Validation = 0.5386478517407141 E_rms Testing = 0.6281846747720651
0.3	E_rms Training = 0.5496184028685688 E_rms Validation = 0.5387635628826097 E_rms Testing = 0.6282134745009941
0.4	E_rms Training = 0.5496652683214674 E_rms Validation = 0.5388629310083863 E_rms Testing = 0.6282266300246935
0.5	E_rms Training = 0.5497093690844128 E_rms Validation = 0.5389490530962294 E_rms Testing = 0.6282321603011393
1	E_rms Training = 0.5498964777153236 E_rms Validation = 0.5392560625823308 E_rms Testing = 0.6282247341561223
5	E_rms Training = 0.5505777468673206 E_rms Validation = 0.5400281440210178 E_rms Testing = 0.6282112851327002
10	E_rms Training = 0.5508634175169813 E_rms Validation = 0.5402730659126659 E_rms Testing = 0.628264930645502



Now the graph is plotted for Training and validation Erms when λ = 0.01. At the value of M= 7 , the training set and validation set errors are reasonably low and the difference between these two errors are also observed to be relatively low. At higher values of M such as 50 , the training and validation set errors may decrease but the time to compute the weights and deriving the regression function and model is becoming higher.

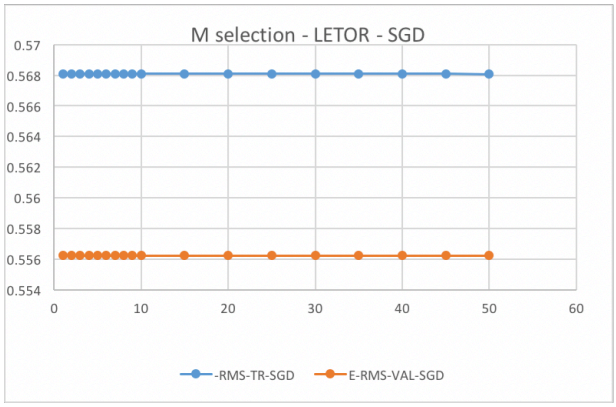
In order to choose the l value, the plot between RMS value of error and ln(l) is graphed.



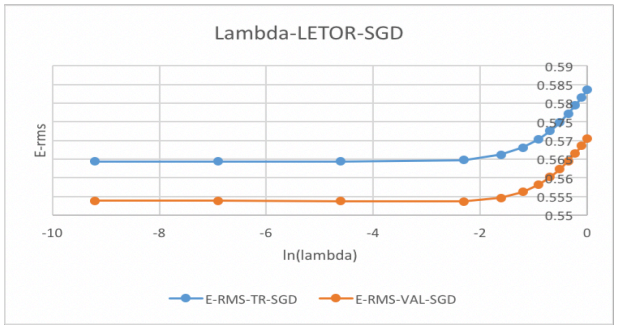


4.2 Stochastic gradient descent Solution

Now, Changing the hyper parameters, which are used in the constructions of the design matrix, are chosen as follows:



Now, Changing the λ, we get



5 Result

Hence, various hyper parameters are tested and optimal weights for linear regression are found with the following parameters.

M = 8
λ = 1
Learning Rate η = 0.001