Learning To Rank (LeTor)

Name **Varsha Ganesh** UB ID **50288433** Email **vganesh2@buffalo.edu**

1. Objective

To solve the Learning to rank Problem(LeTor) problem using Machine Learning techniques. In this project, we will be attacking the Letor with two different Machine Learning approaches. First, one of them will Closed-Form Solution and the second is Stochastic gradient descent.

2. Introduction

We have LeToR training data that consists of input values x and target values t. The input values are derived from a query-document pair. They are real-valued vectors. The target values ((relevance labels) ->{0, 1, 2}) are scalars. Here, we will train our linear regression model on the LeTor dataset by using both Closed form solution and Stochastic Gradient Descent (SGD) Solution. While going through this project, I came across libraries like sklearn, CSV, math and all of them, have unique functionalities that made my life easy in implementing the project.

3. Approach

We approach the problem with two Methodologies.

1. Closed form solution
2. Stochastic Gradient Descent Solution

4. Model Description

4.1 Model

We created a linear regression model of the form:

$y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^\top \phi(\mathbf{x})$ where, $\mathbf{w} = (w_0, w_1, .., w_{M-1})$

is a weight vector will be gathered from training samples. $\phi = (\phi_0, .., \phi_{M-1})^\top$ is a vector of M basis functions. Each basis function $\phi_j(\mathbf{x})$ converts the input vector X into a scalar value. In addition, we will be utilizing Gaussian radial basis functions.

$\phi_j(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_j)^\top \Sigma_j^{-1}(\mathbf{x} - \boldsymbol{\mu}_j)\right)$ where, μ_j is the center of the basis function and Σ_j decides how broadly the basis function spreads.

4.2 Model Training

The LeTor dataset does not have any feature set. Hence, the model was first trained by Closed form and then by stochastic gradient descent. The Reason being that Kmeans algorithm can be used hextract features in group of clusters. In Addition by using Gaussian radial function, we are transforming a linear function to a non-linear function that will greatly simplify the analysis of this class of models.

The closed-form solution with least-square regularization is,

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

where,

$$\mathbf{\Phi} = \begin{bmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \phi_2(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{bmatrix}$$

After dividing the input dataset in to three sets, i.e, Training set, validation set and activation set, we run the model to obtain the following results.

Accuracy		Error (root mean square) [E_rms]	
Training	73.94432353	E_rms Training	0.548965
Validation	74.67983453	E_rms Validation	0.537845
Testing	68.78954452	E_rms Testing	0.6265345

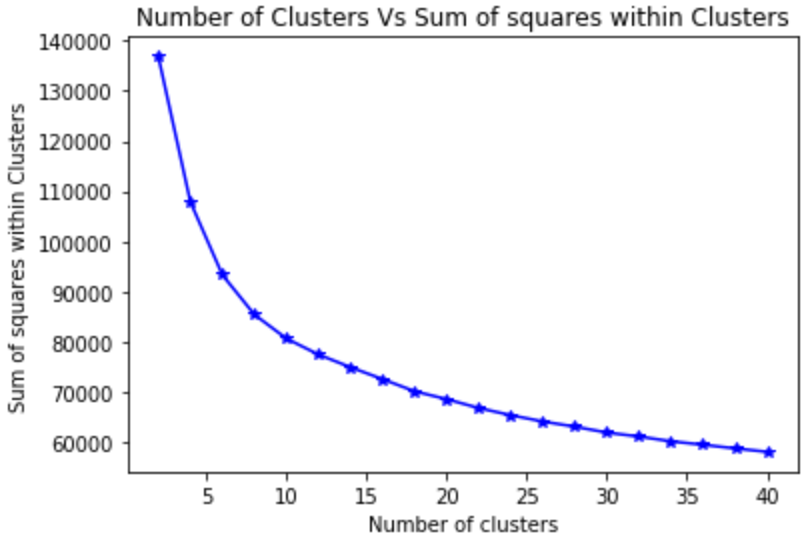
From above results, we can infer that the current model is either overfitted or that the dataset is skewed, i.e, we do not have sufficient data points under every category. There is a substantial difference in accuracies for the three different sets which is an indicator for the same.

5.Experiments

In order to achieve high accuracy, several experiments were conducted by tuning the hyper parameters like number of Clusters(K) and regularization parameter(C_Lambda), μ_j (center of the basis function) for the closed form solution. In addition experiments were also conducted on the Learning rate in Stochastic gradient Descent solution. The results of these experiments are tabulated below. The results were inconclusive, i.e, it didn't help us come up with a best combination of hyper-parameters that would increase the accuracy. To obtain the right set of hyper parameter, I then employed the elbow method and plotted graphs to identify Number of clusters to be 7 and Lambda = 0.09 for best results of Closed form Solution. Similarly I adopted the same strategy to identify Learning rate to be 0.03 for best results of Gradient Descent Solution.

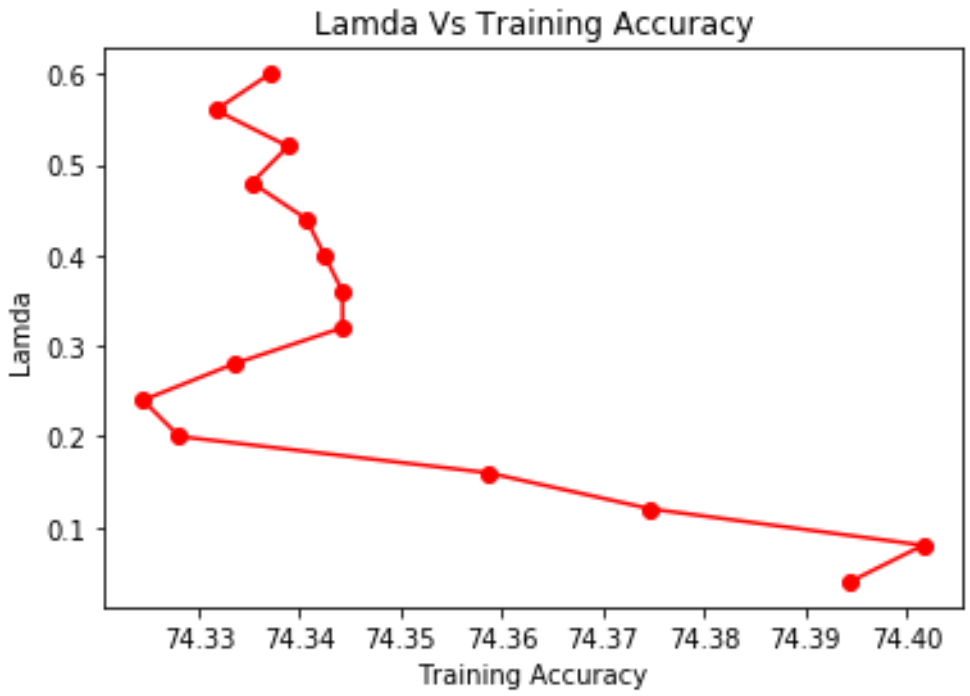
5.1 Number of clusters Vs E_rms

Number of clusters	Erms_Training	Erms_Validation	Erms_Testing
2	0.56123	0.5504	0.63876
3	0.55389	0.541897	0.63123
4	0.55389	0.54178	0.63012
6	0.55123	0.54012	0.628975
8	0.55012	0.6212	0.629012
10	0.549654	0.5381287	0.538874
13	0.547	0.537965	0.62712
15	0.546	0.537412	0.62712
16	0.537	0.537214	0.62712
18	0.538	0.53124	0.6271



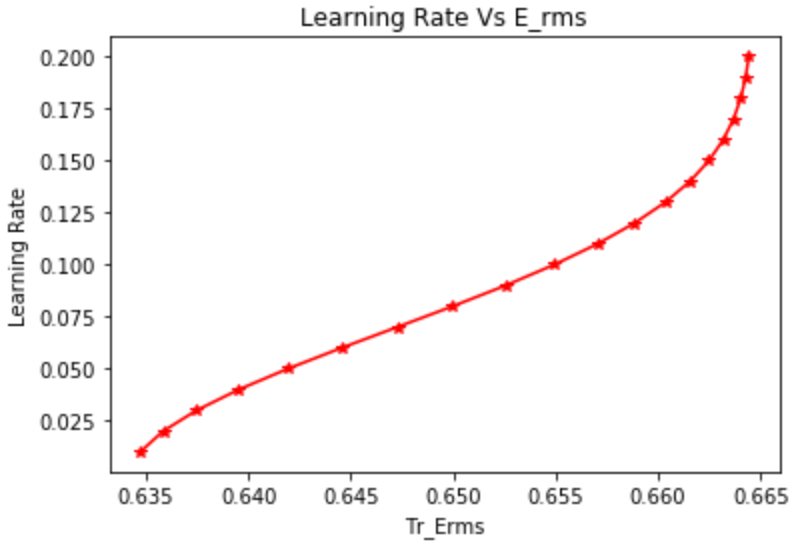
5.2 Lambda vs E_rms

Lambda	Erms_Training	Erms_Validation	Erms_Testing
0.03	0.56123	0.5504	0.63876
0.06	0.55123	0.54012	0.628975
0.09	0.55389	0.54178	0.63012
0.12	0.549654	0.5381287	0.538874



5.3 Learning Rate Vs E_rms

Learning Rate	Erms_Training	Erms_Validation	Erms_Testing
0.01	0.54964	0.53846	0.62372
0.02	0.55869	0.54741	0.63076
0.04	0.55315	0.5416	0.62556
0.06	0.55423	0.54425	0.62597
0.08	0.56451	0.55361	0.6357
0.1	0.58417	0.57105	0.64879



6. Conclusion

In the course of this project, I got a chance to learn of the two approaches to this problem. How they work and how different they are from one another. While Closed Form solution has its advantages, its performs deteriorates as the data grows substantially. On the other hand, Gradient Descent Solution works best with large amount of
data.