Project 2: Learning to Rank using Linear Regression

Baskar Adyar Krishnan
UBIT Name:BASKARAD Person no:50291475

October 11, 2018

Introduction

The goal of this project is to use machine learning to solve a problem that arises in Information Retrieval,one known as the Learning to Rank (LeToR) problem. We are using linear regression, to solve this problem, where we map an input vector x to a real-valued scalar target y(x, w). This is implemented in two ways:
1. Training a linear regression model on LeToR dataset using a closed-form solution.
2. Training a linear regression model on the LeToR dataset using stochastic gradient descent (SGD).

1 Linear Regression Model

Our linear regression function y(x,w) has the form:

$$y(x, w) = w^T(x)$$

where $w = (w_0, w_1, .., w_{M-1})$ is a weight vector to be learned from training samples and $\phi = (\phi_0, .., \phi_{M1})^T$ is a vector of M basis functions. The Gaussian radial basis functions are used:

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

2 Closed-form solution

Closed-form solution solves a given problem in terms of functions and mathematical operations.The closed-form solution without regularization, has the form

$$W = (\phi^T\phi)^{-1}\phi^T t$$

where t = {t1, .., tN } is the vector of outputs in the training data and ϕ is the design matrix

$$\phi(X) = \begin{bmatrix} \phi_0(x_1) & \phi_1(x_1) & \phi_2(x_1) & \dots & \phi_{M-1}(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & \phi_1(x_2) & \dots & \phi_{M-1}(x_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_0(x_N) & \phi_1(x_N) & \phi_1(x_N) & \dots & \phi_{M-1}(x_N) \end{bmatrix}$$

$\phi(X)$ is a 2D array with scalar values

2.1 Procedure for Closed form solution

1. Processing the data: Every model requires a specific type of input, in order to provide a understandable input to the model we need to process the data.In our problem the data has some features which are all zero across all the rows. They don't have any affect on the solution. Therefore those features are removed and not considered anymore. The raw input dataset and target values are divided into 3 parts:
   1. Training Data and target which we are going to use for training (80 percent)
   2. Validation Data and target: (10 percent) is used for validating the model during training to check how good the model predicts the unseen inputs.
   3. Testing Data and target is used to check the accuracy of the model after training is over. (10 percent)

2. The design matrix is calculated as given by equation of $\phi(X)$. Where Each element in the ϕ matrix is given by

   $$\phi_i(x) = exp^{-\frac{(x-\mu)^2}{\sigma^2}}$$

   The above equation can be rewritten as below. The advantage we get is that we can do the matrix multiplication easily if we use the equation below:

   $$\phi_i(x) = e^{-(x-\mu)\sum^{-1}(x-\mu)^T}$$

   Where μ is the mean and $\sum^{-1}$ is the inverse of variance matrix . $\phi_i(x)$ is a scalar value

3. $\sum^{-1}$ Matrix will only have variance of the each features in the diagonals. Thus we will get a square matrix of form <number of features> x <number of features> In our case, it will be 41 x 41 diagonal matrix given by

   $$\begin{bmatrix} \sigma_1^2 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & 0 & \dots & 0 \\ 0 & 0 & \sigma_3^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \sigma_{41}^2 \end{bmatrix}$$

   The variance is calculated for each features of the training data. Training data is used because the amount of data in the training set is more and therefore will be able to get better variance .

4. The entire data set is clustered as M clusters using K-means algorithm. M clusters we will have M number of basis function. The centroid of each cluster is the mean μ for that basis function. Therefore μ will be of M x <number of features> dimension.

5. In order to prevent over-fitting a regularization term is added to the weights. Therefore weight becomes

   $$W = (\lambda I + \phi^T\phi)^{-1}\phi^T t$$

6. Once the weigts are calculated using the above equations, it means the model training is complete. The next step is to test the model with testing dataset and target. For testing, the target values are predicted by W we just calculated and ϕ

7. Testing Error Mean Squared is estimated by comparing the actual target and predicted target we obtained.

2.2 Analysis of Closed form solution

The Code was studied with different values of hyper parameters namely, number of basis function and regularization term

1. Changing the number of Basis function:
   Accuracy and ERMS for various M (1,2,3,4,5,6,7,8,9,10) values are plotted below:

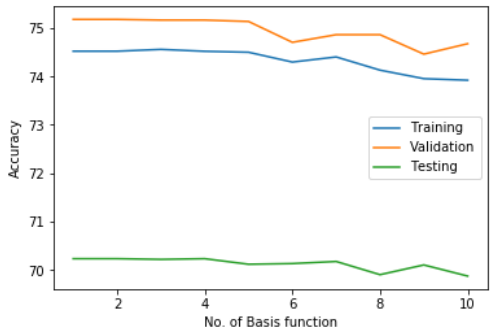

Figure 1: Accuracy

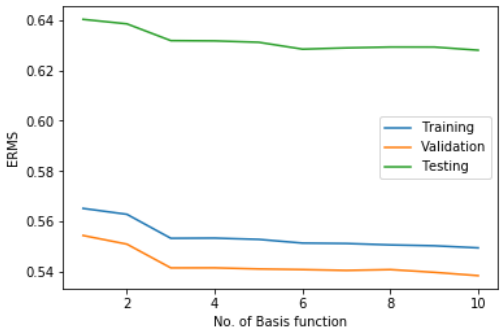

Figure 2: ERMS

It is observed from both the graphs above that as number of clusters increases the accuracy and ERMS does not change much. The data set is in such a way that one basis function is sufficient to train the model.
In other terms, the number of basis function is the number of clusters used to cluster the entire dataset.

2. Changing the value of regularization term
   Accuracy and ERMS for various λ (1,2,5,10,20,50,100,150,200,500) values are plotted below

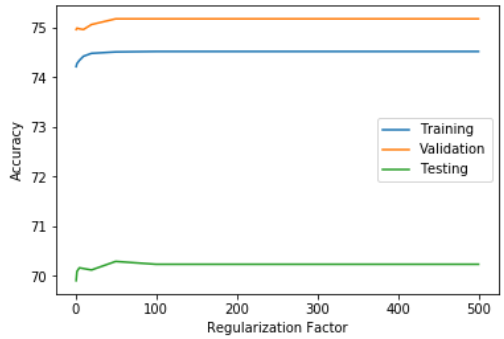

Figure 3: Accuracy

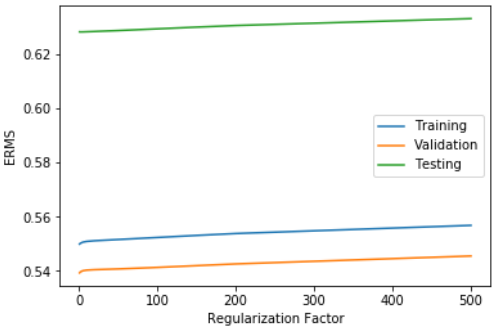

Figure 4: ERMS

From the above two graphs we can see that the ERMS and the accuracy doesn't seem to change for any regularization term much. Even on trying with Lamda value greater than 500 upto 10^5, ERMS and accuracy doesn't seem to change. Since there wasn't much change to plot the graph that is not included as a part of this report.

3 Stochastic Gradient Descent Solution for W

Gradient Descent is the process of minimizing error function by descending down the gradients. To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient (or approximate gradient) of the function at the current point.The stochastic gradient descent algorithm first takes a random initial value $w^{(0)}$. Then it updates the value of w adding delta(w).

$\eta^{(T)}$ is the learning rate, deciding how big each update step would be while descending the gradient and λ is the regularization term.

The target values can be represented by below equation:

$$w^T \phi(X) = w_1\phi(x) + w_2\phi(x)...w_M\phi(x)$$

Like the process we followed for closed form, once the weights for which the ERMS is low is determined using gradient descent algorithm, the training is complete. Using the obtained weights the model is tested using testing dataset and target.

3.1 Analysis of Gradient Descent Solution

From the below graph we can say that the ERMS gets saturated after few iterations and doesn't reduce further. We can say that,the model has learned in few iterations less than 100 for this data set.

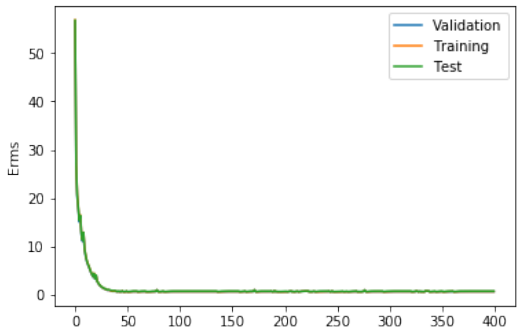

Figure 5: Number of iterations Vs ERMS

If possible we can implement a early stopping to avoid wastage of time running the insignificant iterations. In every iteration the gradient descent algorithm tries to minimize the error function by updating the weights. This is how the minimum ERMS is obtained.

From the graph it is seen that the erms for training, validation and testing are same for all the iterations. Since the model

1. Changing the regularization factor:
   Accuracy and ERMS for various λ (0.5,1,1.5,2,5,10,20,50,100) values are plotted below

2. Changing the learning rate:
   The model is testing by changing various learning rate, but the there was no change in the ERMS of the model on changing any learning rate. Thus we can conclude that the learning rate didn't affect for this model for this particular dataset.

   Since, there wasn't much change in the graph it is not included in this report.

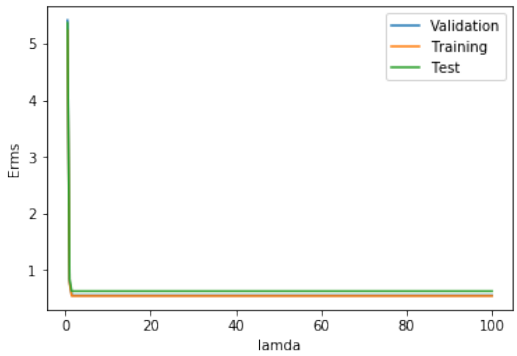

Figure 6: Lambda vs ERMS

3.2 Result

Linear Regression is learned by doing hands-on in this project. Because of the characteristics of this particular data set, new variations were noted on changed various hyper parameters.