CSE574 Introducion to Machine Learning

by Prof: Sargur Srihari

Project 1.2 LeToR Report.

Submitted by

Sai Varun Alapati
Ubit : saivarun
Personal Number :50290571

Table of Contents

Learning to Rank Using Linear Regression

Introduction :

The aim of the project to use machine learning to solve a problem that arises in Information Retrieval, one known as the Learning to Rank (LeToR) problem. In this project, we train a linear regression model onLeToR dataset using a closed-form solution and stochastic gradient descent (SGD).

Problem Definition :

In this project, we rank the document query models ranging based on features extracted. We canformulate this as a problem of linear regression where we map an input vector x to a real-valued scalar target y(x; w).Here, we use LeToR datasetconsists of 69,623 data samples, where each data sample has 46 feature vectors with target values 0,1,2.

Dataset Description and Pre Processing :

LeToR Dataset Descrition

In the LeToR dataset, the input vector is derived from a query-URL pair and the target value is human value assignment about how well the URL corresponds to the query. In this project, we use "QueryLevelNorm" version "Querylevelnorm.txt" among the e three versions for each dataset: "NULL", "MIN" and "QueryLevelNorm" . The entire dataset consists of 69623 query-document pairs(rows), each having 46 features. The sample data row is like

0 qid:10 1:0.000000 2:0.000000 3:0.000000 4:0.000000 5:0.000000 6:0.000000 7:0.000000 8:0.000000 9:0.000000 10:0.000000 11:0.000000 12:0.000000 13:0.000000 14:0.000000 15:0.000000 16:0.001348 17:0.000000 18:0.222222 19:0.000000 20:0.001282 21:0.000000 22:0.000000 23:0.000000 24:0.000000 25:0.000000 26:0.000000 27:0.000000 28:0.000000 29:0.000000 30:0.000000 31:0.000000 32:0.000000 33:0.000000 34:0.000000 35:0.000000 36:0.000000 37:0.000000 38:0.000000 39:0.000000 40:0.000000 41:0.000000 42:0.000000 43:0.017241 44:0.000000 45:0.000000 46:0.000000 #docid = GX000-00-0000000 inc = 1 prob = 0.0246906

The meaning of each column of above dataset are as follows :

1. The first column is the relevance label of the row. It takes one of the discrete values 0, 1 or 2. The larger the relevance label, the better is the match between query and document. Note that objective output y of our linear regression will give a continuous value rather than a discrete one– so as to give a fine-grained distinction.
2. The second column qid is the query id. It is only useful for indexing the dataset and not used in regression.
3. The following 46 columns are the features. They are the 46-dimensional input vector x for our linear regression model. All the features are normalized to fall in the interval of [0; 1].

Data Preprocessing steps :

Extracting Feature Values and lables form the data :

We processed the original text data file into a csv file which makes accessing easier. The csv file contains the feature vectors that contains the labels and the target values.

Target Values :

The first column of the csv file contains the target values ranging from 0,1,2where 2 is the value of data query having highest relevance, 1 being the middleand 0 being the lowest.

Feature vectors :

The entire dataset consists of 69623 query-document pairs(rows), each having 46 features. The feature vectors at 5,6,7,8,9 positions have complete zero values which leads to determinant of zero which is not eligible to calculate inverse of matric. So we remove the above columns from the dataset which concludes to 41 feature column vectors.

Data Partition :

We partitioned the data into a training set, a validation set and a testing set. The training set takes around 80% of the total. The validation set takes about 10% . The testing set takes the rest without overlapping with the above.

Training Linear Regression Model :

we train a linear regression model onLeToR dataset using the following methods :

$$y(x, w) = \omega^T \phi(x)$$

1. Linear Regression usingClosed-form solution
2. Linear Regression using Stochastic gradient descent (SGD).

Linear Regression using Closed-form solution :

In this model we perform Linear Regression using the closed form solution for non linear.

Linear Regression on Non-Linear Problems:

The idea is to apply a non-Linear transformation to the data matrix prior to the fitting step, which then enables a non-linear fit. This non-linear fit can be obtained by using

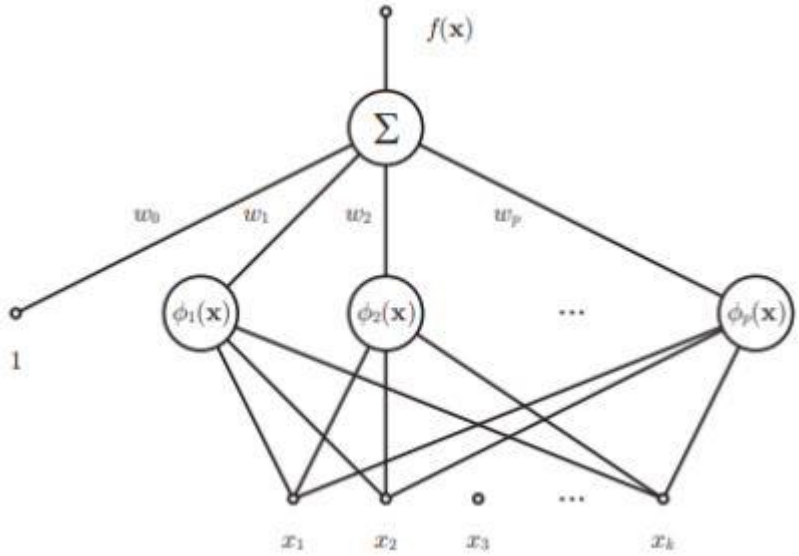
1. Polynomial curve fitting
2. Radial basis function

Radial basis function networks

The idea of radial basis function (RBF) networks is a natural generalization of the polynomial curve fitting. Given data set D = {(xi, yi)}, we start by picking p points to serve as the "centers" in the input space X . We denote those centers as c1, c2,..., cp. Usually, these can be selected from D or computed using some clustering technique (K-means). When the clusters are determined using a Gaussian mixture model, the basis functions can be selected as

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

The whole network can be written as



The closed-form solution of (4), i.e., sum-of-squares error without regularization, has the form

$$\mathbf{w}_{ML} = (\Phi^T\Phi)^{-1}\Phi^T\mathbf{t}$$

where t = {t1; ::; tN} g is the vector of outputs in the training data and Φ is the design matrix:

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

Experimental Values :

Hyperparameters Used :

The following are the hyper parameters used in this model :

- M - The number of basis functions used in the model.
- C_Lambda – This is the regularization factor which helps to reduce the overfitting of the model by adding some random weights to the desired weights.

Varying Number of Basis Functions and Regularization :

These are the number of basis funcitons used in the model are varied from 2 to 16 for different Lambda Values:

For M=3, the following values are observed from this model:

M	C_Lambda	Training Accuracy	Validation Accuracy	Testing Accuracy	Emrs Training	Erms Validaiton	Erms Testing
3	0.03	74.56	75.165	70.22	0.553	0.542	0.632
3	0.15	74.56	75.165	70.22	0.553	0.542	0.632
3	0.45	74.554	75.165	70.22	0.553	0.542	0.632
3	0.7	74.554	75.165	70.22	0.553	0.542	0.632
3	1	74.556	75.165	70.205	0.553	0.542	0.632
3	1.5	74.558	75.165	70.205	0.553	0.542	0.632
3	2.0	74.556	75.165	70.22	0.553	0.542	0.632
3	2.5	74.554	75.165	70.234	0.553	0.542	0.632
3	3	74.554	75.165	70.234	0.553	0.542	0.632

For M=7, the following values are observed from this model:

M	C_Lambda	Training Accuracy	Validation Accuracy	Testing Accuracy	Emrs Training	Erms Validaiton	Erms Testing
7	0.03	74.403	74.864	70.177	0.551	0.541	0.629
7	0.15	74.364	74.907	70.191	0.551	0.541	0.629
7	0.45	74.342	74.82	70.177	0.551	0.541	0.629
7	0.7	74.326	74.835	70.134	0.551	0.541	0.629
7	1	74.344	74.82	70.191	0.551	0.541	0.628
7	1.5	74.355	74.835	70.191	0.551	0.541	0.628
7	2.0	74.357	74.849	70.162	0.551	0.541	0.628
7	2.5	74.373	74.878	70.134	0.551	0.541	0.628
7	3	74.387	74.892	70.091	0.551	0.541	0.628

For M=9, the following values are observed from this model:

M	C_Lambda	Training Accuracy	Validation Accuracy	Testing Accuracy	Emrs Training	Erms Validaiton	Erms Testing
9	0.03	73.955	74.461	70.105	0.55	0.54	0.629
9	0.15	74.052	74.763	69.904	0.55	0.54	0.629
9	0.45	74.208	74.82	69.947	0.551	0.54	0.629
9	0.7	74.281	74.734	70.062	0.551	0.541	0.629
9	1	74.278	74.763	70.119	0.551	0.541	0.629
9	1.5	74.301	74.72	70.177	0.551	0.541	0.629
9	2.0	74.31	74.734	70.249	0.551	0.541	0.629
9	2.5	74.35	74.734	70.22	0.551	0.541	0.629
9	3	74.355	74.763	70.205	0.551	0.541	0.628

For M=10, the following values are observed from this model:

M	C_Lambda	Training Accuracy	Validation Accuracy	Testing Accuracy	Emrs Training	Erms Validaiton	Erms Testing
10	0.03	73.922	74.677	69.875	0.549	0.538	0.628
10	0.15	73.987	74.72	69.832	0.55	0.539	0.628
10	0.45	74.052	74.964	69.774	0.55	0.539	0.628
10	0.7	74.167	74.935	69.846	0.55	0.539	0.628
10	1	74.217	74.964	69.904	0.55	0.539	0.628
10	1.5	74.272	74.993	70.091	0.55	0.539	0.628
10	2.0	74.278	74.993	70.091	0.55	0.54	0.628
10	2.5	74.274	75.007	70.177	0.55	0.54	0.628
10	3	74.301	75.007	70.191	0.55	0.54	0.628

For M=11, the following values are observed from this model:

M	C_Lambda	Training Accuracy	Validation Accuracy	Testing Accuracy	Emrs Training	Erms Validaiton	Erms Testing
11	0.03	73.933	74.605	69.918	0.549	0.538	0.628
11	0.15	73.998	74.648	69.875	0.55	0.539	0.628
11	0.45	74.064	74.95	69.818	0.55	0.539	0.628
11	0.7	74.156	74.978	69.774	0.55	0.539	0.628
11	1	74.22	74.935	69.932	0.55	0.539	0.628
11	1.5	74.229	74.95	70.076	0.55	0.539	0.628
11	2.0	74.276	75.007	70.047	0.55	0.54	0.628
11	2.5	74.271	75.007	70.148	0.55	0.54	0.628
11	3	74.301	75.036	70.148	0.55	0.54	0.628

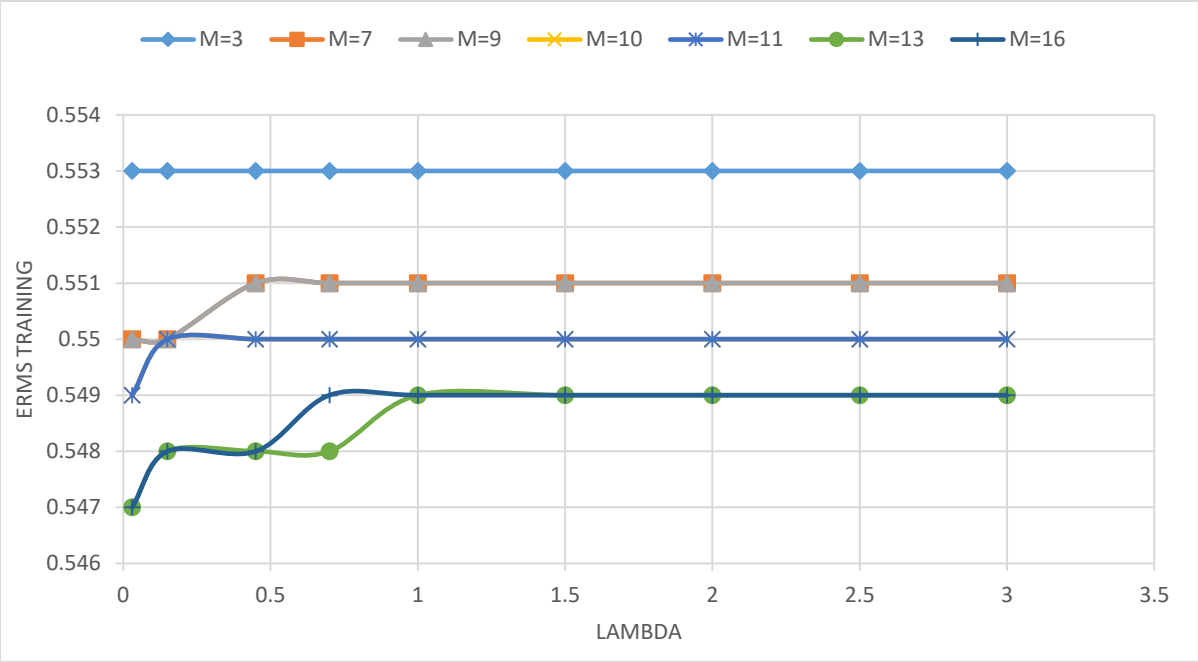
For M=13, the following values are observed from this model:

M	C_Lambda	Training Accuracy	Validation Accuracy	Testing Accuracy	Emrs Training	Erms Validaiton	Erms Testing
13	0.03	73.35	74.232	69.214	0.547	0.538	0.628
13	0.15	73.565	74.361	69.43	0.548	0.538	0.628
13	0.45	73.789	74.605	69.631	0.548	0.539	0.628
13	0.7	73.885	74.648	69.746	0.548	0.539	0.628
13	1	73.935	74.806	69.889	0.549	0.539	0.628
13	1.5	74.096	74.849	69.861	0.549	0.539	0.628
13	2.0	74.201	74.892	69.861	0.549	0.539	0.628
13	2.5	74.285	75.007	69.918	0.549	0.539	0.628
13	3	74.312	74.978	69.932	0.549	0.539	0.628

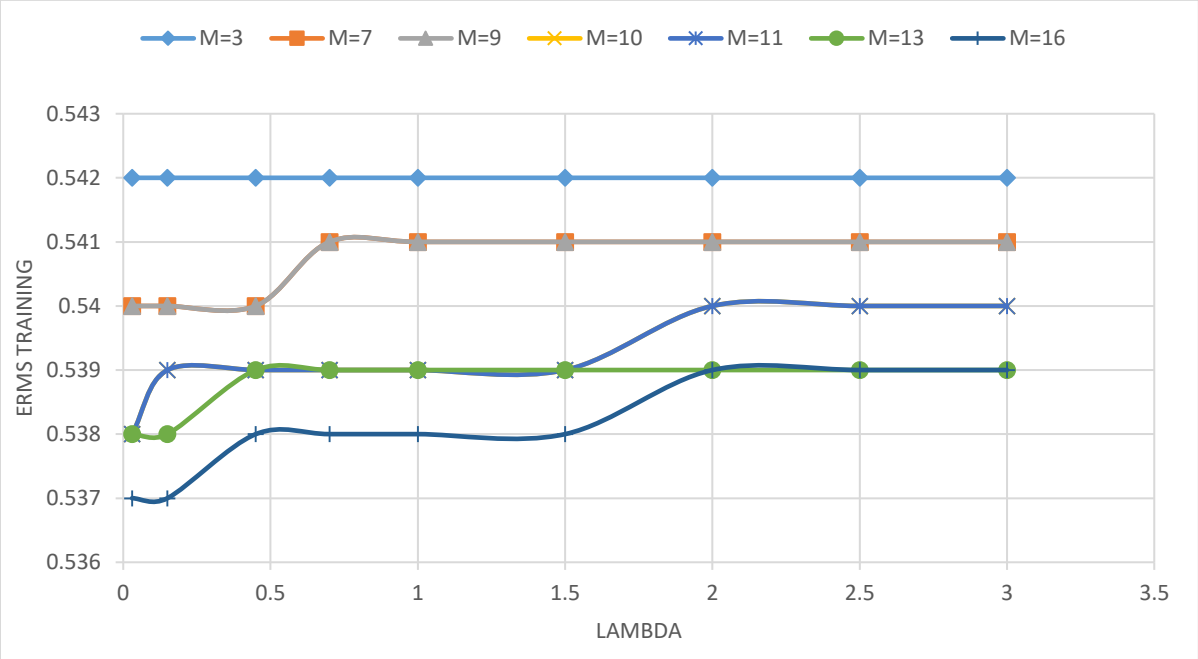
For M=16, the following values are observed from this model:

M	C_Lambda	Training Accuracy	Validation Accuracy	Testing Accuracy	Emrs Training	Erms Validaiton	Erms Testing
16	0.03	73.542	74.318	69.099	0.547	0.537	0.627
16	0.15	73.646	74.49	69.315	0.548	0.537	0.627
16	0.45	73.775	74.548	69.645	0.548	0.538	0.628
16	0.7	73.84	74.562	69.774	0.549	0.538	0.628
16	1	73.922	74.562	69.76	0.549	0.538	0.628
16	1.5	73.956	74.706	69.875	0.549	0.538	0.628
16	2.0	74.019	74.763	69.861	0.549	0.539	0.628
16	2.5	74.043	74.763	69.961	0.549	0.539	0.628
16	3	74.08	74.806	69.976	0.549	0.539	0.628

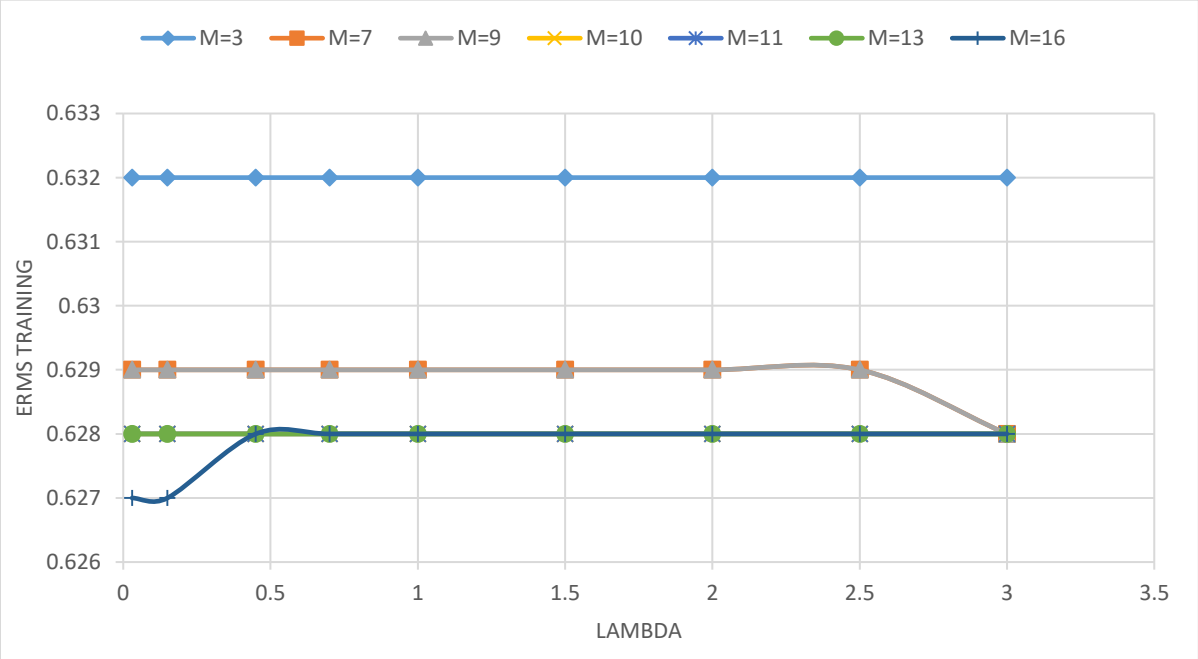
The below graph is the pictorial representation of Erms Training data obtained for different values of Lambda which is in X-axis and for different values of M



The below graph is the pictorial representation of Erms Training data obtained for different values of Learning rate which is in X-axis and for different values of M



The below graph is the pictorial representation of Erms Training data obtained for different values of Learning rate which is in X-axis and for different values of M



Observations :

Effect of Lambda on Model :

Lambda is the regularization factor which helps to reduce the overfitting of the model by adding some random weights to the desired weights.

- We observed that for testing data the increase in lambda slightly decreased the training Ermsbut it is not recommended to increase the lambda since it is the random weight we are adding might result in underfitting of model. If our lambda is high then the model learn wont learn enough to predict the real time data.
- If lambda value is low then our model might result in overfitting of the data. So it is recommended to adjust lambda to a lower value.

Effect of Number of Basis Functions M :

M is the The number of basis functions used in the model.

- We observed that the increase in the number of basis functions slightly decreased the Erms value but if we have too many basis functions we will have many clusters which is not recommended fora Ideal model.

Final Model :

From above experiments, I observed that the Emrs is minimum and Accuracy is maximum at M=10,11 and Lambda at 0.03.

Linear Regression using Stochastic gradient descent (SGD) :

- It is the process of mininmizing a function by following the gradients of the cost function.
- This involves knowing the form of the cost as well as the derivative so that from a given point you know the gradient and can move in that direction, e.g. downhill towards the minimum value.
- In Machine learning we can use a similar technique called stochastic gradient descent to minimize the error of a model on our training data.
- The stochastic gradient descent algorithm takes a random initial value. Then it updates the value of w using
  $W^{(\tau+1)}=W^{(\tau)}+\Delta W^{(\tau)}$

where $\Delta w(\tau) = -\eta^{(\tau)} \Delta E$ is called the weight updates. It goes along the opposite direction of the gradient of the error. $\eta(\tau)$ is the learning rate, deciding how big each update step would be. Because of the linearity of differentiation, we have

$$\nabla E = \nabla E_D + \lambda \nabla E_W$$

$$\nabla E_D = -(t_n - \mathbf{w}^{(\tau)T}\phi(\mathbf{x}_n))\phi(\mathbf{x}_n)$$

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

Experimental Values :

Hyperparameters Used :

The following are the hyper parameters used in this model :

- M - The number of basis functions used in the model.
- C_Lambda – This is the regularization factor which helps to reduce the overfitting of the model by adding some random weights to the desired weights.

Varying Number of Basis Functions and Regularization, Eta :

These are the number of basis funcitons used in the model are varied from 1 and 10 for different Lambda Values and eta values:

For M=1, the following values are observed from this model:

M	Lambda	Eta	Training Accuracy	Validation Accuracy	Testing Accuracy	Erms Training	Erms Valiaition	Erms Testing
1	0.01	0.01	1.993	1.652	2.787	2.296	2.305	2.241
1	0.01	0.03	74.522	75.18	70.234	0.599	0.586	0.692
1	0.01	0.15	74.522	75.18	70.234	0.642	0.627	0.74
1	0.03	0.01	5.018	4.41	7.801	2.116	2.124	2.063
1	0.03	0.03	74.522	75.18	70.234	0.601	0.587	0.694
1	0.03	0.15	74.522	75.18	70.234	0.642	0.628	0.74
1	1.5	0.01	74.522	75.18	70.234	0.622	0.608	0.718
1	1.5	0.03	74.522	75.18	70.234	0.64	0.625	0.738
1	1.5	0.15	74.522	75.18	70.234	0.643	0.628	0.741
1	2.0	0.01	74.522	75.18	70.234	0.63	0.616	0.727
1	2.0	0.03	74.522	75.18	70.234	0.641	0.627	0.739
1	2.0	0.15	74.522	75.18	70.234	0.643	0.628	0.741

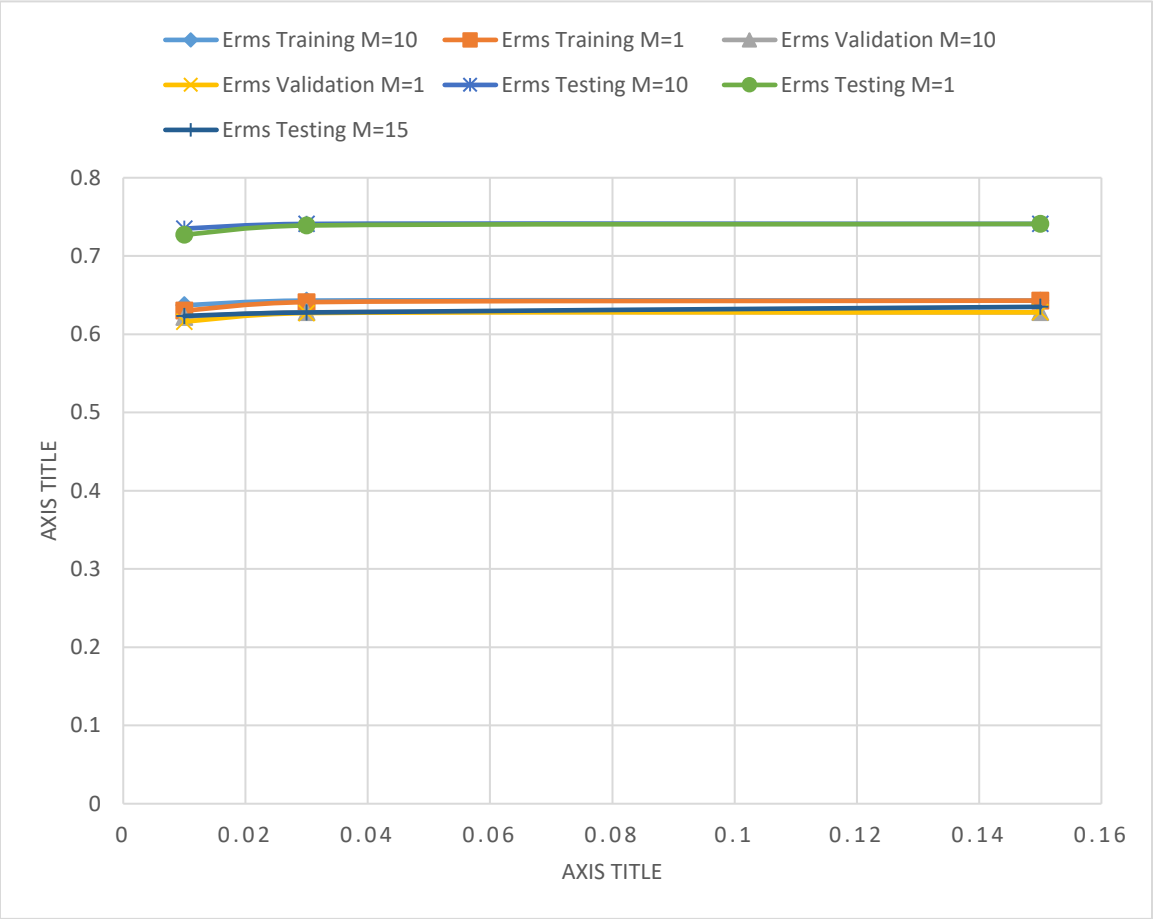
For M=10, the following values are observed from this model:

M	Lambda	Eta	Training Accuracy	Validation Accuracy	Testing Accuracy	ErmsTraining	Erms Valiaiton	Erms Testing
10	0.01	0.01	1.124	1.178	0.977	27.304	27.092	27.187

10	0.01	0.03	1.102	1.379	1.35	24.605	24.409	24.488
10	0.01	0.15	3.747	3.849	4.123	12.662	12.631	12.61
10	0.03	0.01	1.221	1.278	1.049	25.187	24.992	25.077
10	0.03	0.03	1.354	1.652	1.58	19.301	19.149	19.203
10	0.03	0.15	12.476	12.223	12.268	3.873	3.864	3.899
10	1.5	0.01	74.522	75.18	70.234	0.616	0.602	0.714
10	1.5	0.03	74.522	75.18	70.234	0.643	0.628	0.741
10	1.5	0.15	74.522	75.18	70.234	0.643	0.628	0.741
10	2.0	0.01	74.522	75.18	70.234	0.637	0.622	0.735
10	2.0	0.03	74.522	75.18	70.234	0.643	0.628	0.741
10	2.0	0.15	74.522	75.18	70.234	0.643	0.628	0.741

For M=15, the following values are observed from this model:

Learning Rate	E_rms Training	E_rms Validation	E_rms Testing
0.01	0.5497	0.5384	0.6234
0.001	0.54945	0.53843	0.62782
0.0001	0.55043	0.53976	0.62781
0.1	0.55044	0.53975	0.6241
0.3	0.5528	0.54196	0.62392



Observations :

Effect of Lambda :

- From above observations we noticed that for lower lambda values the error is huge. Since we are intaializing random weights in SGD this lambda factor plays an important role which adds random number of weights. So for higher rate the error is less which is recommended.

Effect of Learning rate:

- From above observations we noticed that , If the learning rate value is too high, your model will be simple, but you run the risk of under fitting your data. Your model won't learn enough about the training data to make useful predictions.
- If the learning rate value is too low, your model will be more complex, and you run the risk of overfitting your data. Your model will learn too much about the particularities of the training data, and won't be able to generalize to new data.

Effect of the number of Basis Functions:

- From above observations we noticed that, increase in the basis function increased the emrs for lower lambda values. So it is strictly recommended to keep lamda values high while increasing M.

Conclusion and Summary :

We observed that there isn't significant difference in the accuracies for any change in hyper parameters. The given dataset has the highest accuracies are obtained is 75.18. For the given dataset. The Gradient descent model we get higher accuracies when compared with closed form solution. But if you consider Emrs the value are little lower in Closed Form solution which is 0.627 compared to Gradient descent solution.