Project-2 Report

Puneethp
50206906

Goal:

In this we solve the problem that occurs in Information Retrieval known as the Learning-To-Rank Problem using machine learning technique known as Linear Regression.

Given:

Here we are given a data that consists of pairs of input values x and target values t. The input values are real-valued vectors. The target values are scalars that take one of three values 0, 1, 2: the larger the relevance label, the better is the match between query and document.

Step-1:

Data Partition: In data Partition we are supposed to split data into 3 different parts: The training dataset, the test data set and the validation data set. The training data set should contain 80% of the data and test set -10% and the validation set must contain the remaining 10%.

This splitting data can be done in two different ways: In a random fashion or taking the first 80% as training set and the following 20% as the test and validation sets of 10% each.

Now, when we split the data of total 69623 rows. We get 80% as 55601 rows and 10% as 7011 rows. So the training data set contains 55601 rows and test and validation data sets contains 7011 rows.

Here in the input data for each row (output vector) we have 46 feature values of input and some noise. We eliminate the unnecessary noise by using an iterator and store only the required feature values in a matrix. So we store the training, test and validation sets in three different matrices.

Training dataset:

Matrix XT (X Training): This matrix contains 55601 rows and 46 columns.

Matrix YT (Y training): This matrix contains 55601 rows and 1 column (80% of target variable) since this is a column matrix of target variable either 0,1,2.

Test Dataset:

Matrix XTST (X test): This matrix contains 7011 rows and 46 columns since this is matrix of input variables.

Matrix YTST (Y Test): This matrix contains 7011 rows and 1 column since this is a column matrix of target variable either 0,1,2.

Validation Dataset:

Matrix XV (X Validation): This matrix contains 7011 rows and 46 columns since this is a validation matrix and should contain only 10% of data

Matrix YV (Y Validation): This matrix contains 7011 rows and 1 column since this is a column matrix of target variable either 0,1,2.

Step-2:

Hyper-paramater tuning:

First we have to find the values of M, μj, Σj, λ, η(τ) and then apply linear regression and train the data. Grid Search can be used to choose basis function M Or we can assume M as any random integer and find different corresponding weight vectors.

After finding these values validation set can be used to adjust the values of these hyper parameters in case of any result that is ambiguous.

We should be very cautious while choosing the value of M. It shouldn't be too small or very large. An ideal value should be chosen. If we choose very small value of M then it underfits the data. If the value of M is too large it overfits the data. Weights and regularized weights are calculated using closed form as well as stochastic gradient function.

Weights from closed form solution

Weight of Closed form sol: ',

```
array([[ 0.5208668 ],
       [ 0.48365411],
       [-2.08028611],
       [ 1.33753738]]))
```

Weight using stochastic gradient function

```
array([[ -3.27893182e-04],
       [  9.99735670e-01],
       [ -2.80508821e-04],
       [ -2.83763013e-04]]))
```

Now we have to calculate regularized weights. For calculating regularized weights and calculation of error the value of λ is needed. This can also be calculated using grid search. It should be in the interval of 0 and 1. We choose any random value in between 0 and 1 as λ.

Now based on this value of λ the regularized weights are:

Regularized weights from closed form solution

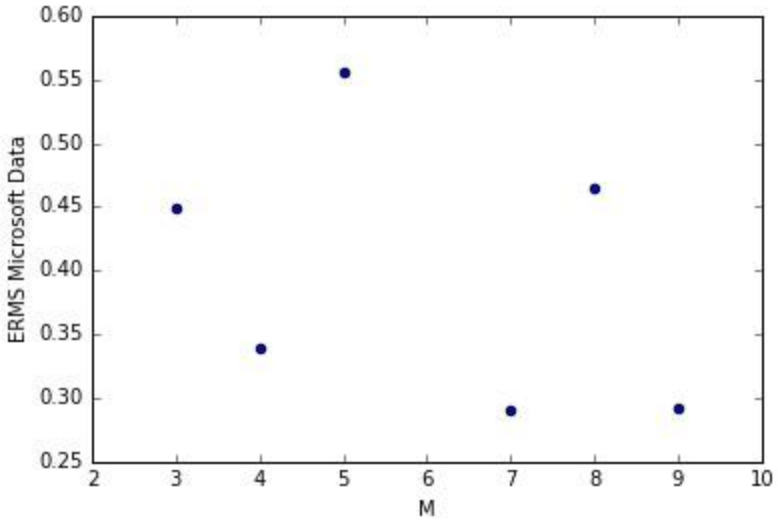
array([[0.52083081],
[0.49548529],
[-2.06763996],
[1.31411644]]))

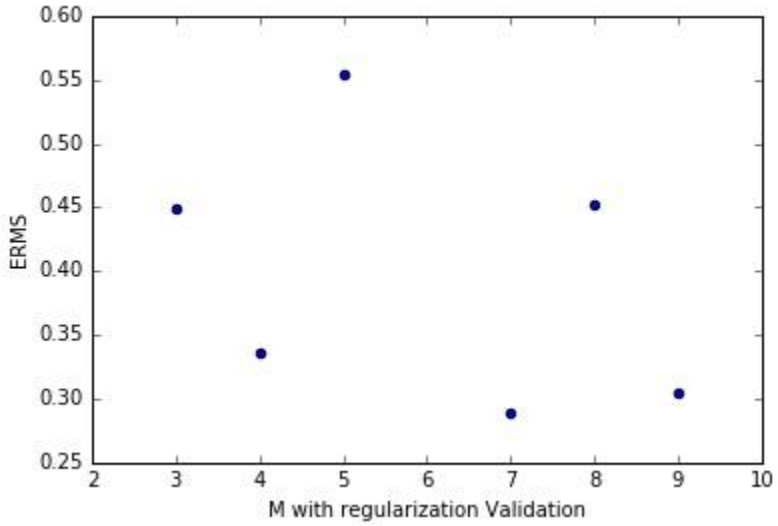
Regularized weights from stochastic gradient descent

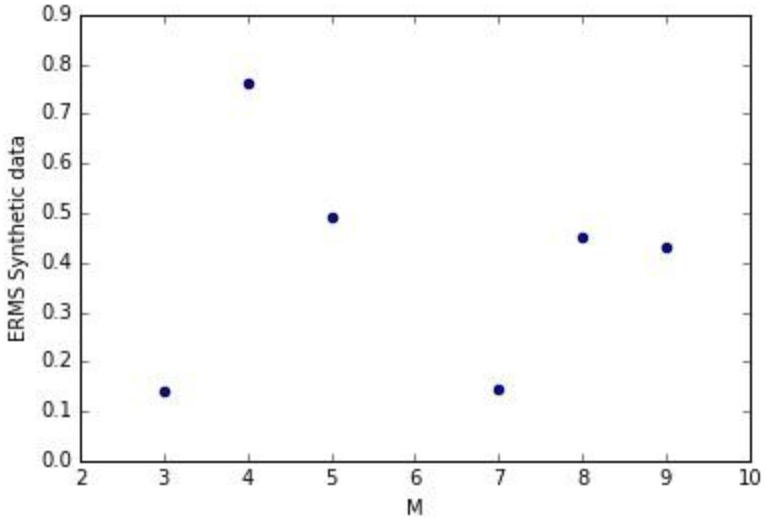
array([[-3.27744669e-04],
[9.99376148e-01],
[-2.80383697e-04],
[-2.83635240e-04]]))

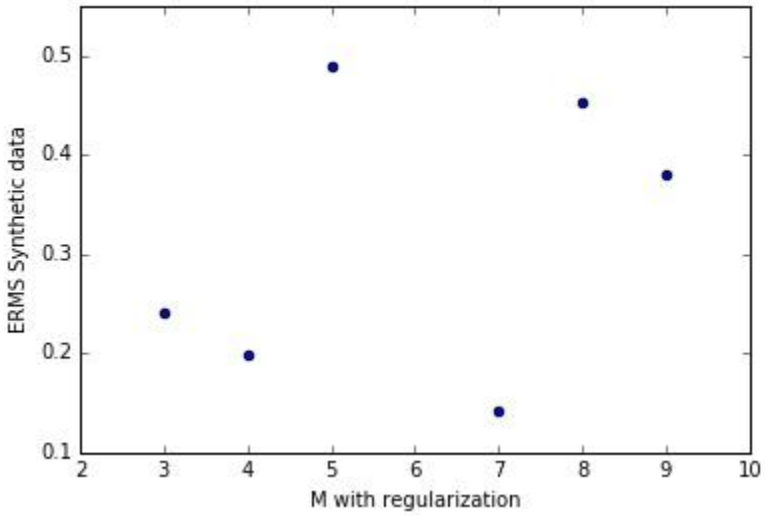
Now after getting the weights, we have to verify the accuracy of the model by using validation set with those of the assumed hyper parameters. For different values of M we get different values for weights and Root Mean square errors.

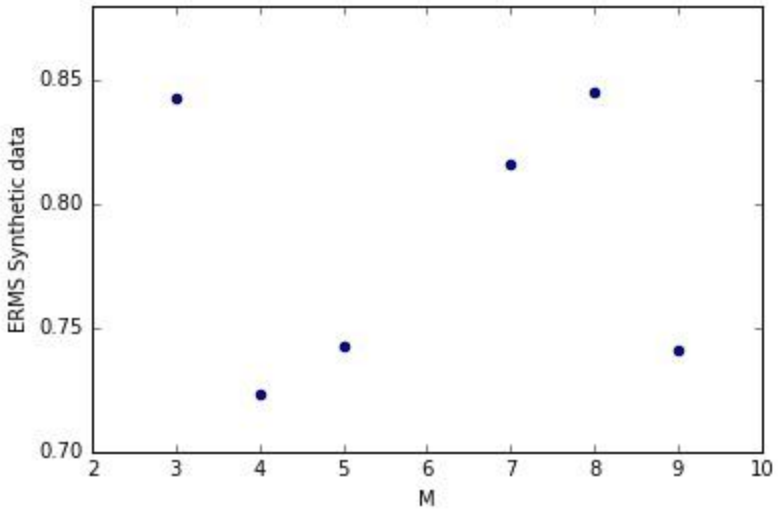
Here the values of M are taken as 3,4,5,7,8

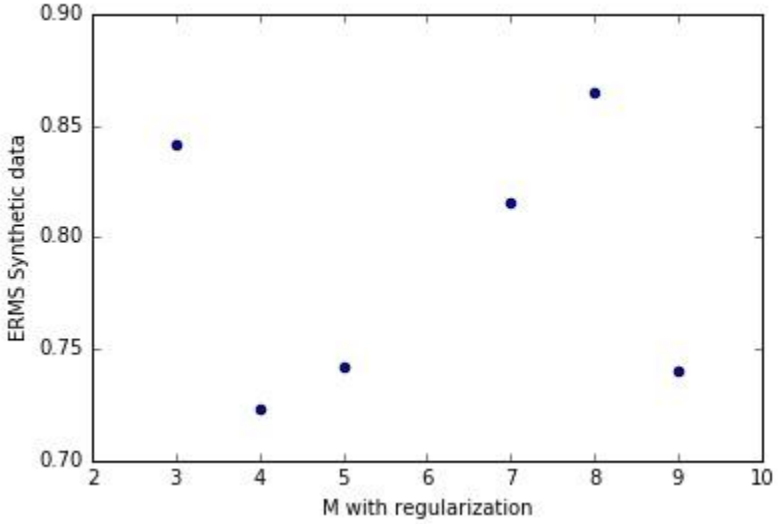

0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
ERMS Microsoft Data
2
3
4
5
6
7
8
9
10
M

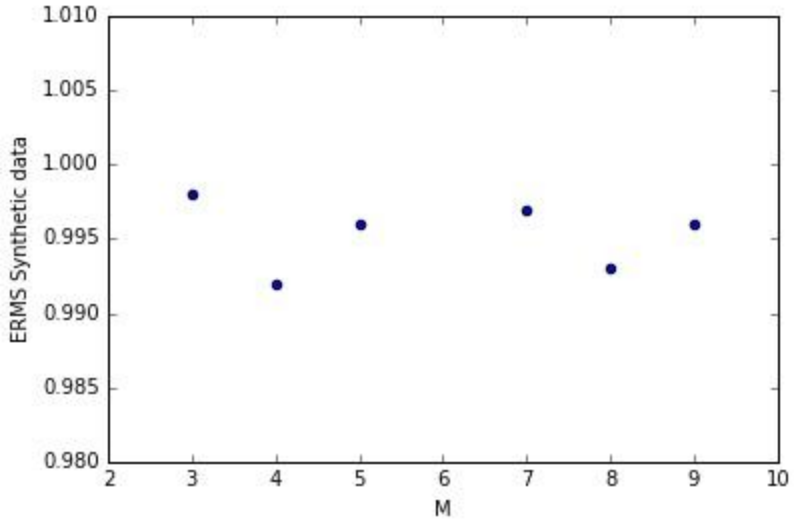

0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
ERMS
2
3
4
5
6
7
8
9
10
M with regularization Validation

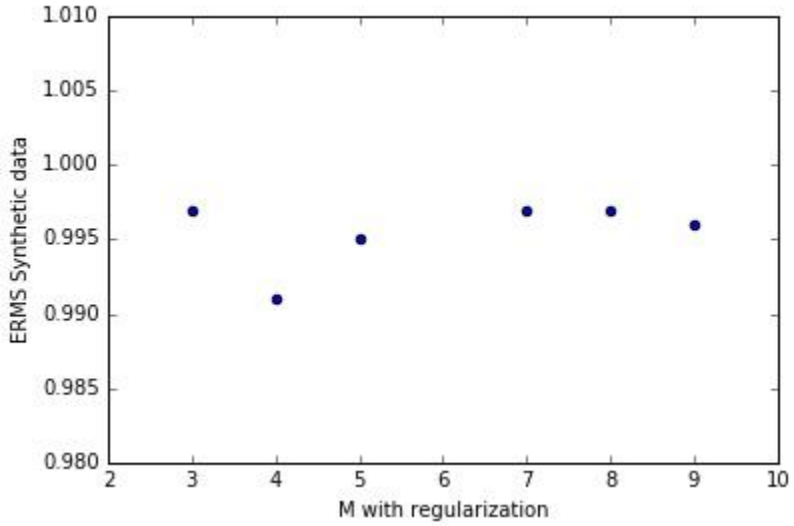

ERMS Synthetic data
M
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
2
3
4
5
6
7
8
9
10


ERMS Synthetic data
M with regularization
0.1
0.2
0.3
0.4
0.5
2
3
4
5
6
7
8
9
10


0.85
0.80
0.75
0.70
ERMS Synthetic data
2
3
4
5
6
7
8
9
10
M


0.90
0.85
0.80
0.75
0.70
ERMS Synthetic data
2
3
4
5
6
7
8
9
10
M with regularization





It can be inferred from the graphs that when M is 4, the ERMS is of optimal value.For any other value of M, the deviation in the range of ERMS is too huge, and hence they may result in under fit or over fit of data while training. Thus, the value of M can be fixed as 4.

Mean:

- Mean can be calculated simply by taking any M-1 rows from the data set can be considered as μ matrix.
- For the first column of phi matrix, phi(x) is a constant 1 in all cases.
- Hence, M-1 values of Mean matrix will be required to calculate phi(x) values for each x.
- Value of mean calculated using random M-1 rows in dataset:

```
[ 3.82780000e-02  0.00000000e+00  4.00000000e-01  1.00000000e+00
  4.28570000e-02  0.00000000e+00  0.00000000e+00  0.00000000e+00
  0.00000000e+00  0.00000000e+00  3.74650000e-02  0.00000000e+00
  3.95405000e-01  8.23954000e-01  4.17550000e-02  1.04420000e-02
  1.25000000e-01  2.10526000e-01  3.07692000e-01  1.03130000e-02
  9.90647000e-01  8.69042000e-01  7.56415000e-01  7.88567000e-01
  0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
  7.53434000e-01  8.52790000e-01  6.46241000e-01  8.88726000e-01
  7.55246000e-01  5.40530000e-02  7.03000000e-04  5.40670000e-02
  9.91501000e-01  8.58723000e-01  7.64378000e-01  8.53643000e-01
  7.78600000e-03  8.47500000e-03  9.71429000e-01  4.00000000e-01
  2.90323000e-01  0.00000000e+00]
[ 0.817786  0.      0.      0.      0.817786  0.      0.      0.
  0.      0.      0.861692  0.      0.      0.      0.86216
  0.722421  0.042254  0.12    0.294118  0.722369  0.507891  0.801346
  0.838883  0.832281  0.      0.      0.      0.      0.      0.
  0.      0.      0.      0.      0.      0.      0.506851
  0.803171  0.837448  0.834308  0.004577  0.027397  0.019727  0.6     0.166667
  0.    ]
[ 0.310976  0.      0.      0.      0.310976  0.      0.      0.
  0.      0.      0.304407  0.      0.      0.      0.304023
  0.19487   0.009709  0.      0.714286  0.194195  0.531698  0.324672
  0.400082  0.313792  0.      0.      0.      0.      0.      0.
  0.      0.      0.      0.      0.      0.      0.538882
  0.156837  0.206278  0.135905  0.001081  0.011111  0.      0.833333
  0.767857  0.    ]
[ 0.141654  0.      0.      0.      0.141654  0.      0.      0.
  0.      0.      0.580909  0.      0.      0.      0.585664
  0.123274  0.12    0.      0.363636  0.123484  0.668261  0.641147
  0.869359  0.673982  0.      0.      0.      0.      0.      0.
  0.      0.      0.      0.      0.      0.      0.684841
  0.616294  0.869464  0.674779  0.00164   0.006993  0.      0.666667
  0.324786  0.    ]
```

Sigma: Basis function Σ is considered to be a diagonal matrix in which each diagonal element is chosen to be proportional to the i^{th} dimension variance of the training data.

The obtained Sigma array is:

```
array([[ 5.50747091e-02,   0.00000000e+00,   0.00000000e+00, ...,
         0.00000000e+00,   0.00000000e+00,   0.00000000e+00],
       [ 0.00000000e+00,   6.51664069e-02,   0.00000000e+00, ...,
         0.00000000e+00,   0.00000000e+00,   0.00000000e+00],
       [ 0.00000000e+00,   0.00000000e+00,   1.16757161e-01, ...,
         0.00000000e+00,   0.00000000e+00,   0.00000000e+00],
       ...,
       [ 0.00000000e+00,   0.00000000e+00,   0.00000000e+00, ...,
         7.06563445e-02,   0.00000000e+00,   0.00000000e+00],
       [ 0.00000000e+00,   0.00000000e+00,   0.00000000e+00, ...,
         0.00000000e+00,   6.18923592e-02,   0.00000000e+00],
       [ 0.00000000e+00,   0.00000000e+00,   0.00000000e+00, ...,
         0.00000000e+00,   0.00000000e+00,   1.79849646e-05]]))
```

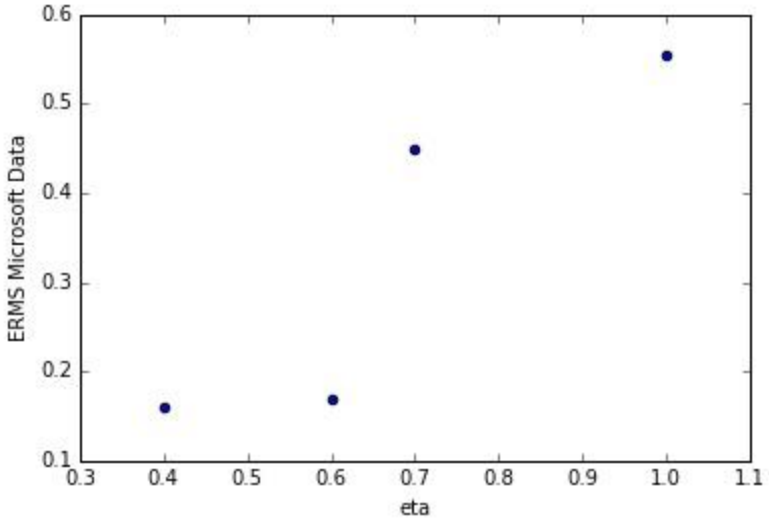
Eta:

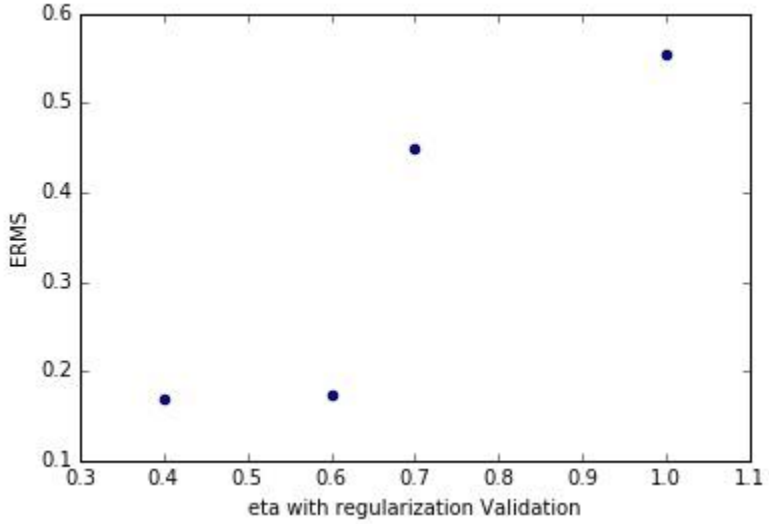
Learning rate, η(τ) can be either fixed or variable.

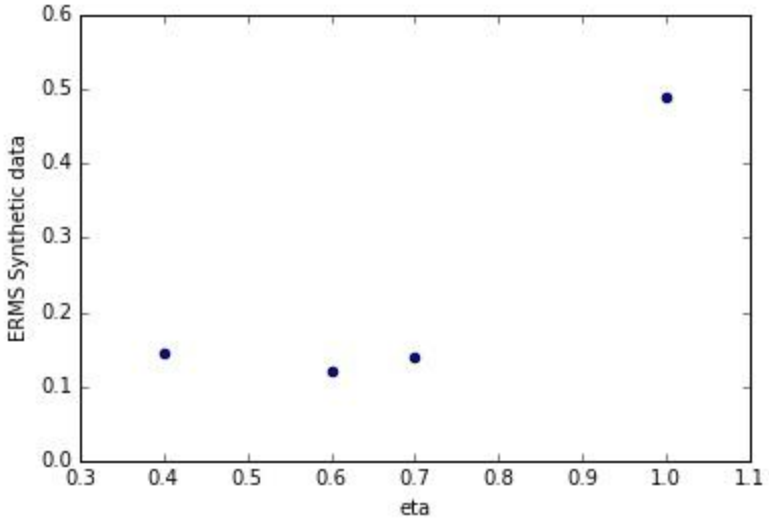
The learning rate should be in the range of 0 and 1, hence I have assumed it to be 0.5 (fixed) in the initial calculation.

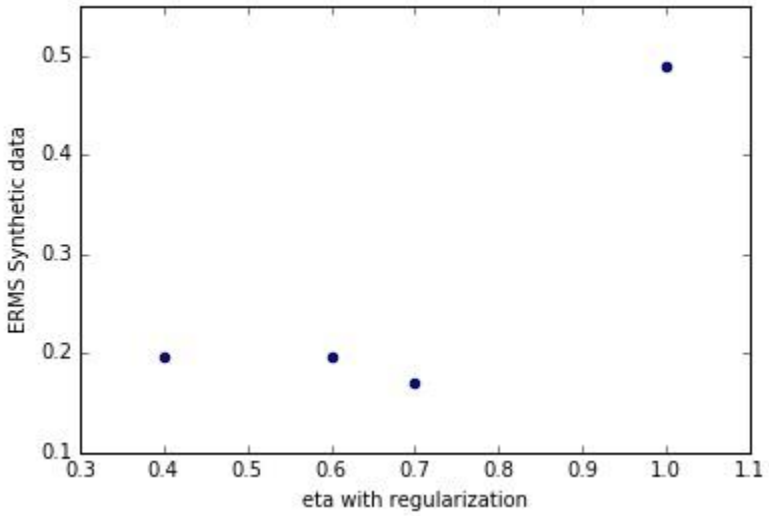
The weights that are obtained using this value of Eta are presented in the initial sections of this report.

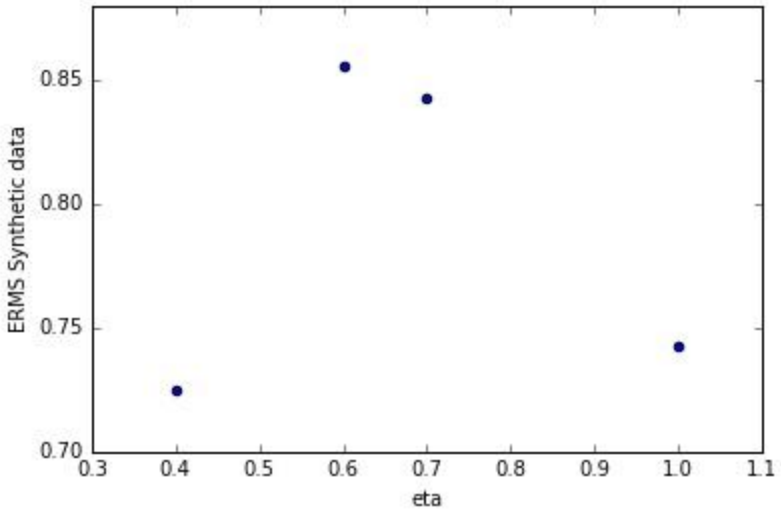
During validation, I have takem multiple values of Eta such as 1, 0.7, 0.4 for which the Root Mean Square error is obtained as follows:

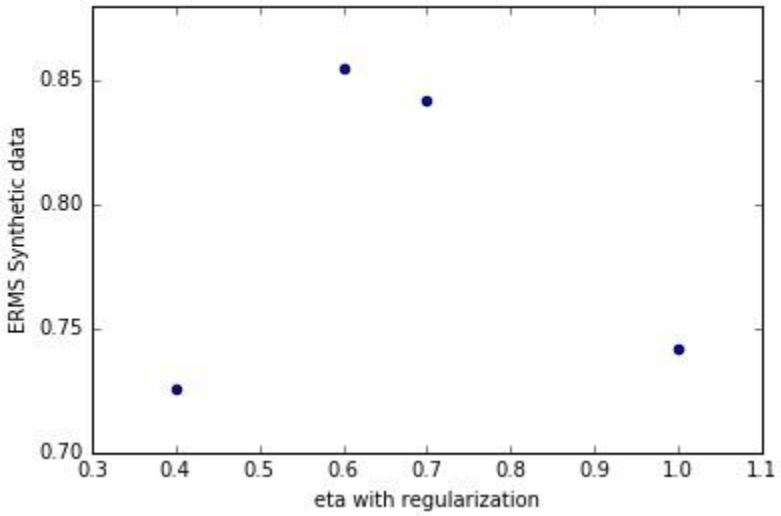

0.6
0.5
0.4
0.3
0.2
0.1
ERMS Microsoft Data
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1
eta

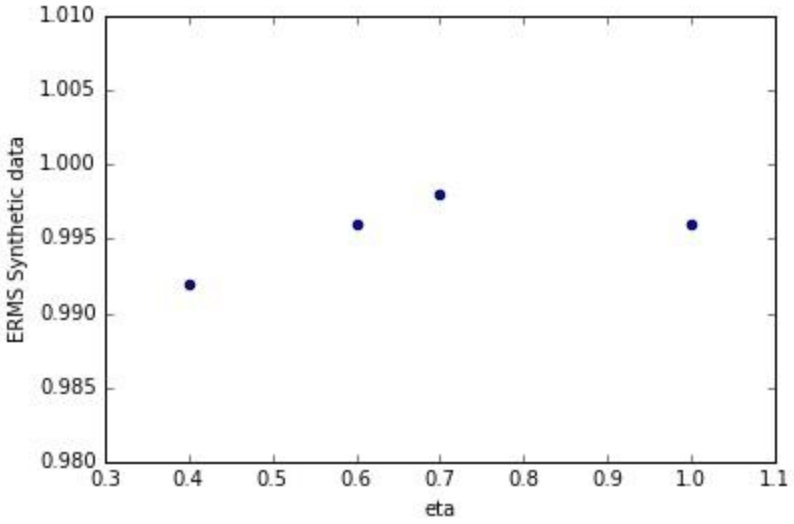

0.6
0.5
0.4
0.3
0.2
0.1
ERMS
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1
eta with regularization Validation

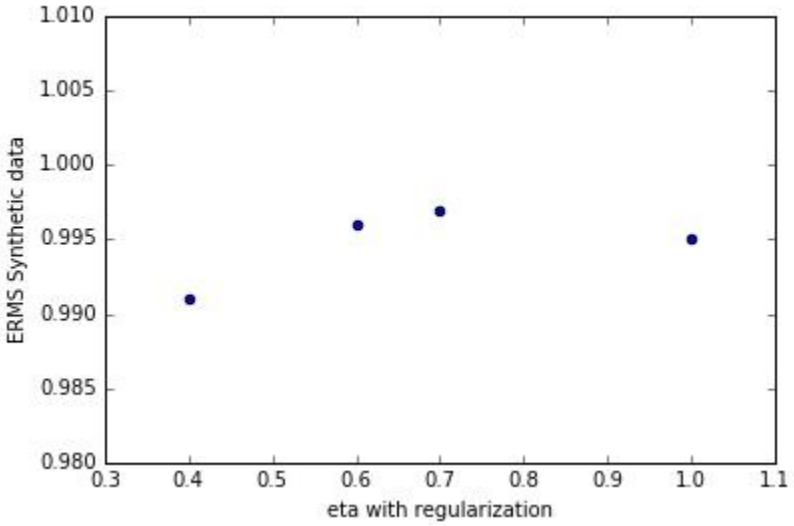

0.6
0.5
0.4
0.3
0.2
0.1
0.0
ERMS Synthetic data
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1
eta


0.5
0.4
0.3
0.2
0.1
ERMS Synthetic data
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1
eta with regularization


ERMS Synthetic data
0.85
0.80
0.75
0.70
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1
eta


ERMS Synthetic data
0.85
0.80
0.75
0.70
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1
eta with regularization





The Final results obtained are:

Microsoft LETOR 4.0 Data Set:

- ERMS Closed form training set: ', 0.55288382139423375)
- ('Regularized ERMS Closed form training set: ', 0.55288423681869858)
- 'ERMS Closed form testing set: ', 0.37718154515259927)
- ('Regularized ERMS Closed form testing set: ', 0.37693817381687422)
- ERMS Closed form validation set: ', 0.47288853905662659)
- ('Regularized ERMS Closed form validation set: ', 0.
- ERMS Stochastic form validation set: ', 0.78713171052996533)
- ('Regularized ERMS Stochastic form validation set: ', 0.78684871346922192)

- ERMS Stochastic form training set: ', 0.74640519110256698)
- ('Regularized ERMS Stochastic from training set: ', 0.74621996683504299
- ERMS Stochastic form testing set: ', 0.81977424907895202)
- ('Regularized ERMS Stochastic form testing set: ', 0.81947951744320635)

Synthetic Data Set:

- 'ERMS Closed form training set: ', 0.55795192232583146)
- ('Regularized ERMS Closed form training set: ', 0.55795216401299175)
- ERMS Closed form testing set: ', 0.24490141145811836)
- ('Regularized ERMS Closed form testing set: ', 0.24480463405053668
- 'ERMS Closed form validation set: ', 0.089530818244763527)
- ('Regularized ERMS Closed form validation set: ', 0.091694827181430766)
- 'ERMS Stochastic form training set: ', 0.74074512927156999)
- ('Regularized ERMS Stochastic from training set: ', 0.74056261135582058
- 'ERMS Stochastic form testing set: ', 0.77254609179036426)
- ('Regularized ERMS Stochastic form testing set: ', 0.77226834708918024)
- 'ERMS Stochastic form validation set: ', 0.99885726868236413)
- ('Regularized ERMS Stochastic form validation set: ', 0.99849814020597949)