MACHINE LEARNING PROJECT2 REPORT

Ubit name: anmishar

Person #: 20508673

Machine Learning Project 2 Report

Data Partition:

The Data sets used for this project are LETOR 4.0 and the synthetically generated data set.

- The project requirement was to partition the data into 3 parts
- 80% -training data
- 10%-validation data
- 10%- testing data
- Each data set consists of 69623 rows of data, hence the data was partitioned into
- Training data = 80% of 69623 =55601
- Validation data =10% of 69623 =7011
- Testing data =10% of 69623 =7011
- The data sets hence were obtained by taking the first 55601 rows of data as training sets, the next 7011 rows as validation sets and the last 7011 rows as testing set.
- The requirement can also be satisfied by choosing 80% of the rows randomly from the data set as training and the validation and testing from the remaining 20% without repetition.
- For simplicity I used the first approach to make the data partition.
- The data sets I used are:
    - Training Data – The X matrix obtained contains 55601 rows and 46 columns. This 46 columns represent the feature vectors. The Y matrix is a single column matrix with 55601 rows. The y values consist only values 0, 1, 2 which are the relevance scores. This is the 80% of data.
    - Validation Data- The matrix Xvalid and Yvalid consist of the validation data set. They are similar to the matrices X and Y only smaller in size (i.e) 7011.
    - Testing Data – The testing data is Xtest and Ytest each of 7011 rows, having 46 feature vectors and 7011 relevance scores.

Hyper Tuning Parameters:

- In order to generate best output we need to select optimal values of parameters M, μj, η, Σj, λ in order to proceed further with the linear equations.
- The Basis function M can be chosen using grid search.
- With different values of M we get different weights and so the corresponding vectors, to train the linear regression.
- The validation set can be trained on the weight that has been obtained from the training data.
- The hyper parameters will be adjusted based on the results obtained by the validation data set. If the errors increase (i.e.) the values overfit or underfit the curve we need to change the values of the parameters in order to get the optimal result.
- The value of M should be in a way that does not promote the outcome to be either an underfit model or an overfit model.
- If the M is too small, the model underfits the data.
- If the M is large, the model overfits the data,
- The optimal value for M that I assumed for this project is 4, and I trained the model parameters for 'W' on the training data.
- Weights and regularized weights are calculated by two methods :
  - Stochastic Gradient Descent
    Weights from this method for synthetic data are:
    [-3.92699801e-04]
    [-3.04094302e-04]
    [9.99694235e-01]
    [-2.82435893e-04]]

    Weights for the MQ2007 data are:
    [-4.02775767e-04]
    [-3.35217301e-04]
    [9.99685597e-01]
    [-3.28714786e-04]]

- Gaussian Radial Distribution (Closed Form Solution)

  Weights for MQ2007:
  [[0.36830812]
  [1.39229279]
  [2.40399446]
  [-3.78103777]]
  Weights for synthetic data:
  [[-0.54304317]
  [0.26614899]
  [-0.07036017]
  [0.97811244]]

Lambda:

- For the calculation of regularized weights and errors we use the parameter λ.
- The value of λ can also be found by using grid search. The value should lie between the range of (0, 1). In this project I generated the value for λ using a random value generator in the range of 0 and 1.
- The value for lambda is ('lamda ', 0.4993466298908087)

- The weights generated using the regularization parameter λ are:

  For Synthetic data:

  Regularized Weights calculated from Stochastic Gradient Descent Method
  [[-3.92523462e-04]
  [-3.03957750e-04]
  [9.99245327e-01]
  [-2.82309066e-04]]

  Regularized Weights calculated from Radial Gaussian Distribution
  [[-0.5393479]
  [0.26446908]
  [-0.06950637]
  [0.97385678]]

For MQ2007:

Regularized Weights calculated from Stochastic Gradient Descent Method

[[-4.02594903e-04]
 [-3.35066774e-04]
 [9.99236694e-01]
 [-3.28567178e-04]]

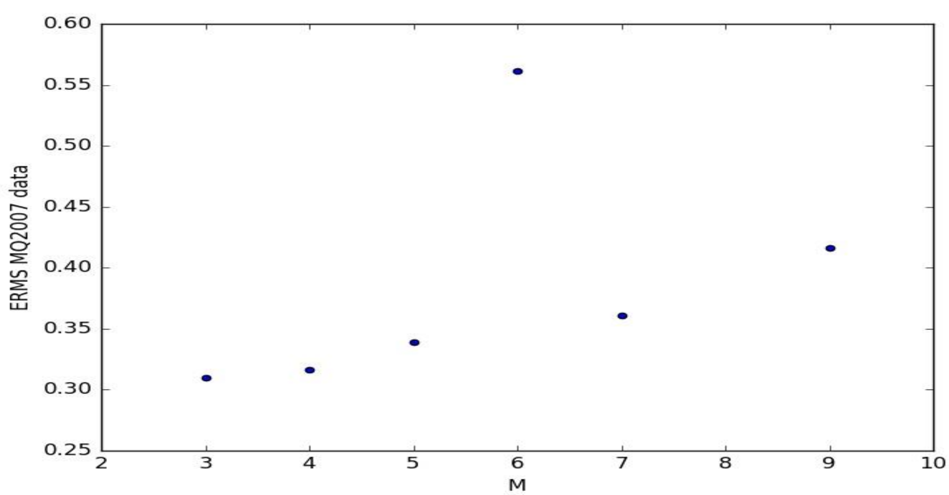
Regularized Weights calculated from Radial Gaussian Distribution

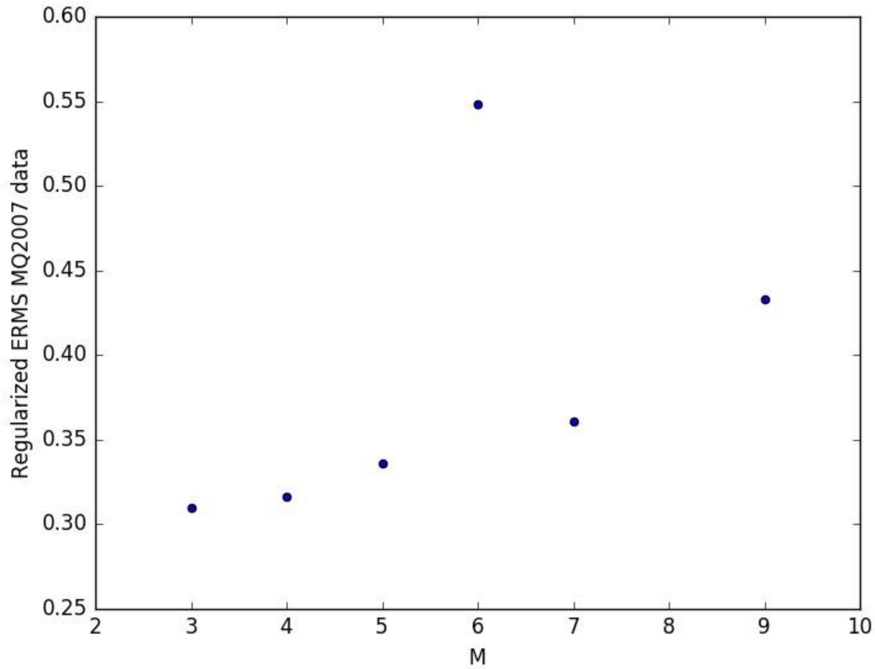
[[0.37129204]
[1.36902551]
[2.36945188]
[-3.72811484]]

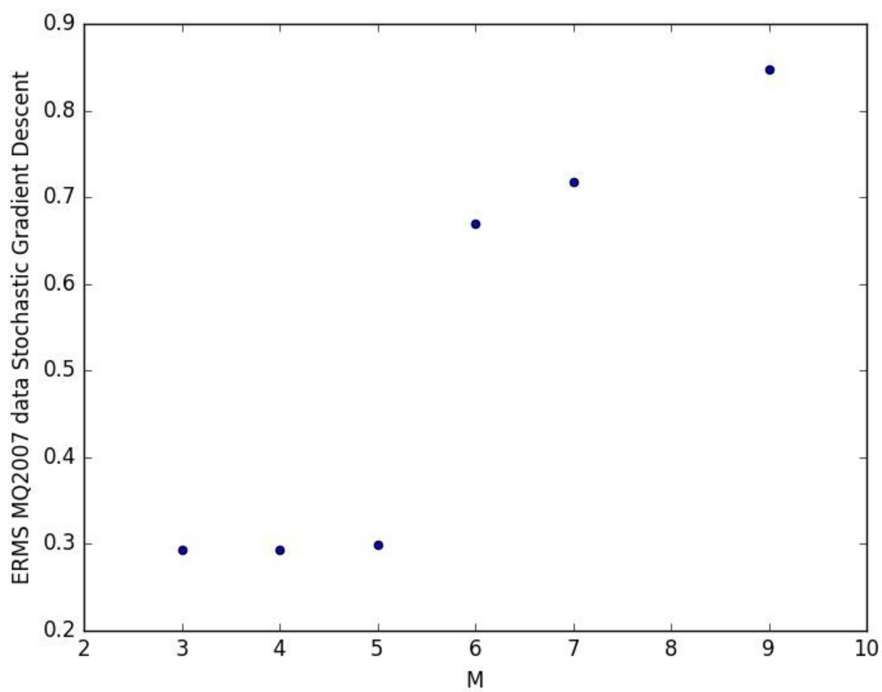
For generating the weights we need to find the μj, η, Σj and the 'phi' matrix, can be found in the report further.

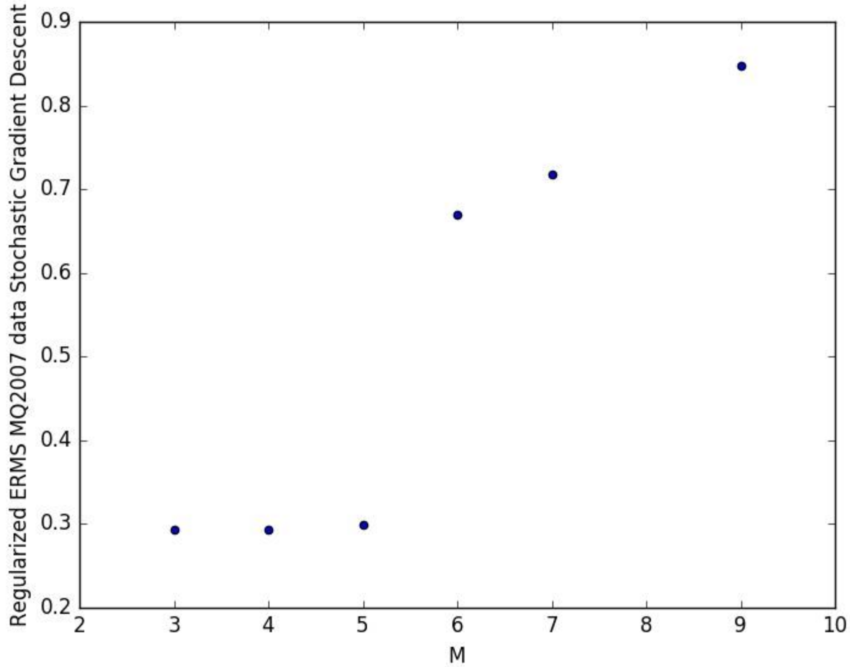
M:

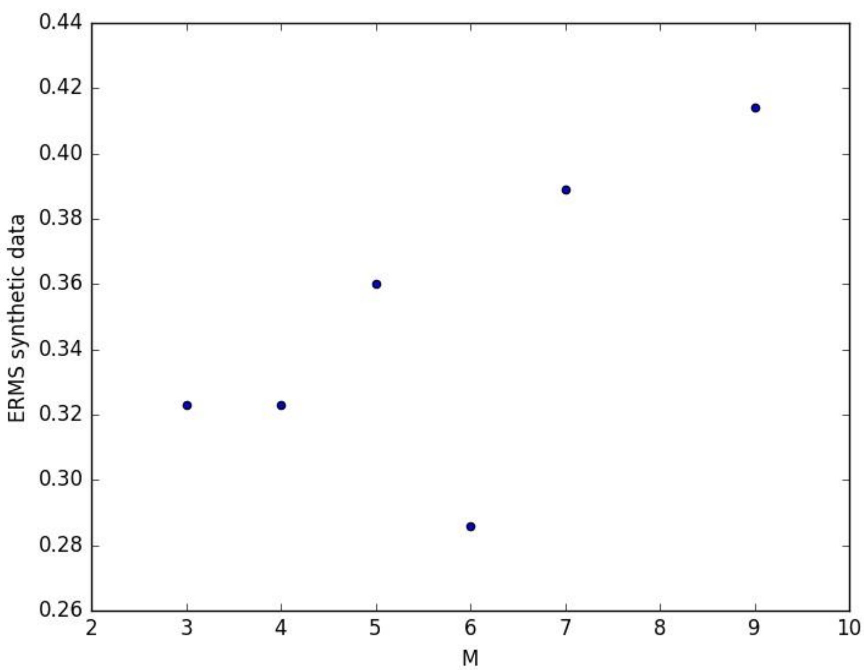
- The M value can be obtained by checking the output obtained after training the validation set.
- The values of M which I ci=considered are 3, 4, 5, 6, 7, 9
- The following graphs show how the value of error varies as M vary. I found 4 to be optimal.

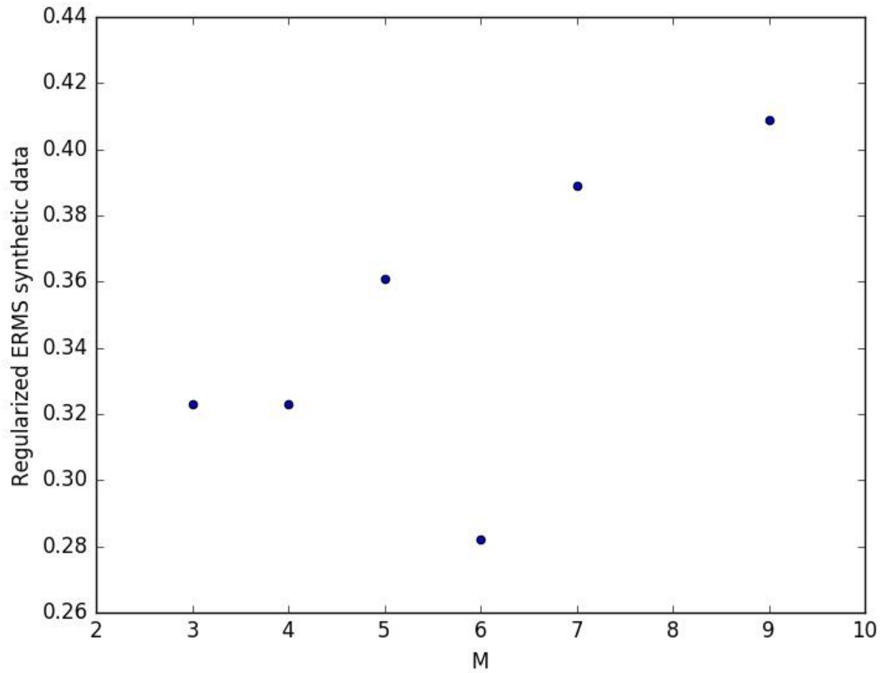


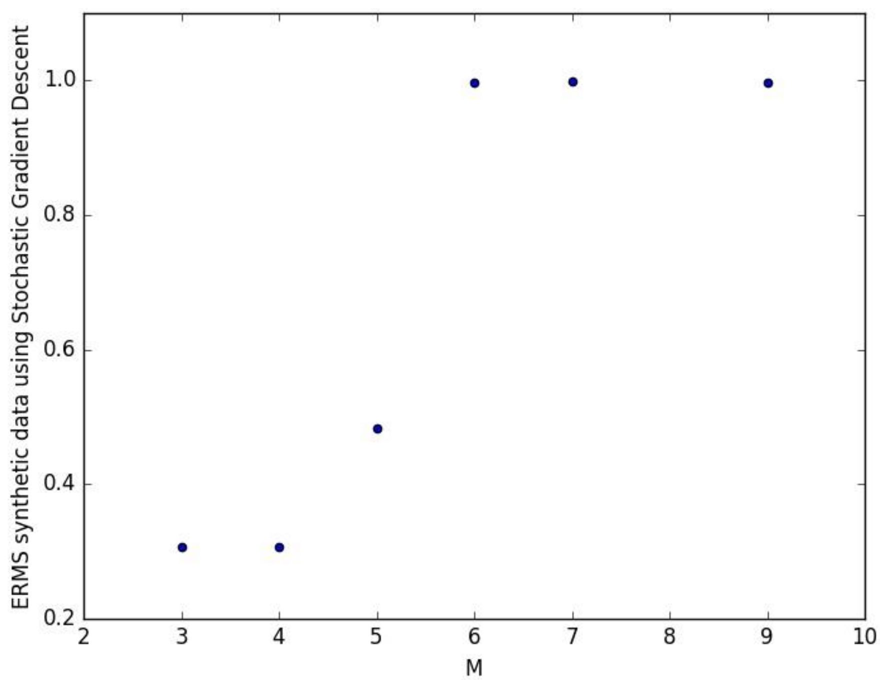

Regularized ERMS MQ2007 data
M
0.25
0.30
0.35
0.40
0.45
0.50
0.55
0.60
2
3
4
5
6
7
8
9
10

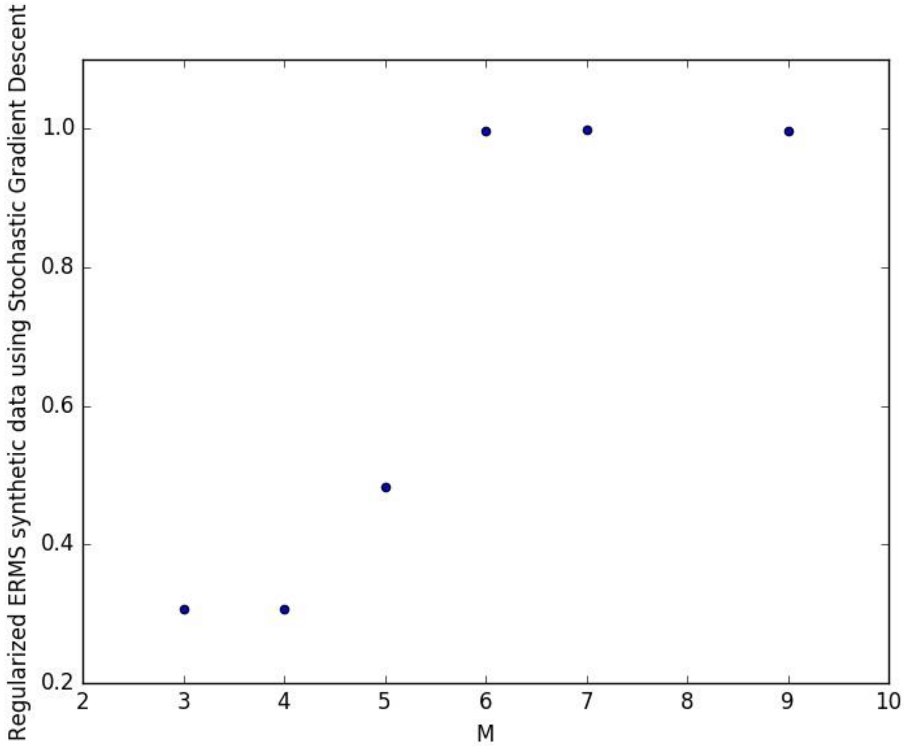

ERMS MQ2007 data Stochastic Gradient Descent
M
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
2
3
4
5
6
7
8
9
10


Regularized ERMS MQ2007 data Stochastic Gradient Descent
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
2
3
4
5
6
7
8
9
10
M


ERMS synthetic data
0.44
0.42
0.40
0.38
0.36
0.34
0.32
0.30
0.28
0.26
2
3
4
5
6
7
8
9
10
M


Regularized ERMS synthetic data
0.44
0.42
0.40
0.38
0.36
0.34
0.32
0.30
0.28
0.26
2
3
4
5
6
7
8
9
10
M


ERMS synthetic data using Stochastic Gradient Descent
1.0
0.8
0.6
0.4
0.2
2
3
4
5
6
7
8
9
10
M



- From the above graphs plotted, the values for M =3 and M=4, we can see that there is not much difference that can be seen when M=3 and M=4.

- As the value of M increases the error also can be found to be increasing.

- We know that as the value of M increases the training error decreases and the testing error increases.

- Hence, we select M =4, so that it helps in getting least errors and prevents over fitting and under fitting of the data.

SIGMA Matrix:

- The value for Σj is represented as a diagonal matrix of the dimension 46x46. The elements of this matrix are the variances of the each column in the matrix X. Each column has number of rows which are equal to the size of the data set being trained.
- The inverse of a diagonal matrix is the same as the matrix itself. Hence we use this matrix for the calculation of phi and weights.

$$\Sigma = \begin{pmatrix} \sigma_1^2 & & & \\ & \sigma_2^2 & & \\ & & \ddots & \\ & & & \sigma_D^2 \end{pmatrix}$$

('**Sigma array** ([[5.50747062e-02, 0.00000000e+00, 0.00000000e+00, ...,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
[0.00000000e+00, 6.51663990e-02, 0.00000000e+00, ...,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
[0.00000000e+00, 0.00000000e+00, 1.16757254e-01, ...,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
...,
[0.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
7.06564368e-02, 0.00000000e+00, 0.00000000e+00],
[0.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
0.00000000e+00, 6.18923598e-02, 0.00000000e+00],
[0.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
0.00000000e+00, 0.00000000e+00, 1.79849646e-05]]))

MEANS:

The values of μj (i.e.) the means can be calculated using 2 different ways. For the calculation of phi we need M-1 values of μ . This is done as follows.

- Using K-Means clustering for different values.
- Or randomly assuming M values from the X matrix as the centers.

I calculated the means by assuming random rows of X as the means and obtained M-1 means. M-1 as the value of first element of each row of the phi matrix is considered to be 1.

Hence we require to calculate only M-1 elements of the rows thus, needing only M-1 values of μ.

Mean values:

```
[ 0.136298  0.        0.666667  0.        0.139168  0.        0.        0.
  0.        0.        0.131676  0.        0.693636  0.        0.138177
  0.060366  0.214286  0.538462  0.4       0.060874  0.614932  0.282248
  0.310176  0.104167  0.        0.        0.        0.        0.64001
  0.782959  0.719272  0.838581  0.        0.        0.        0.        0.617542
  0.281239  0.270342  0.091169  0.066751  0.003497  0.02      0.5       0.383333
  0.      ]
```

```
[ 0.155385  0.        0.        0.        0.155385  0.        0.        0.
  0.        0.        0.237335  0.        0.        0.        0.234387
  0.058478  0.016667  0.178571  0.        0.057356  0.897114  0.427687
  0.610111  0.528197  0.        0.        0.        0.        0.        0.
  0.        0.        0.        0.        0.        0.        0.874725
  0.394307  0.581391  0.500456  0.005256  0.026251  0.266667  0.        0.133333
  0.      ]
```

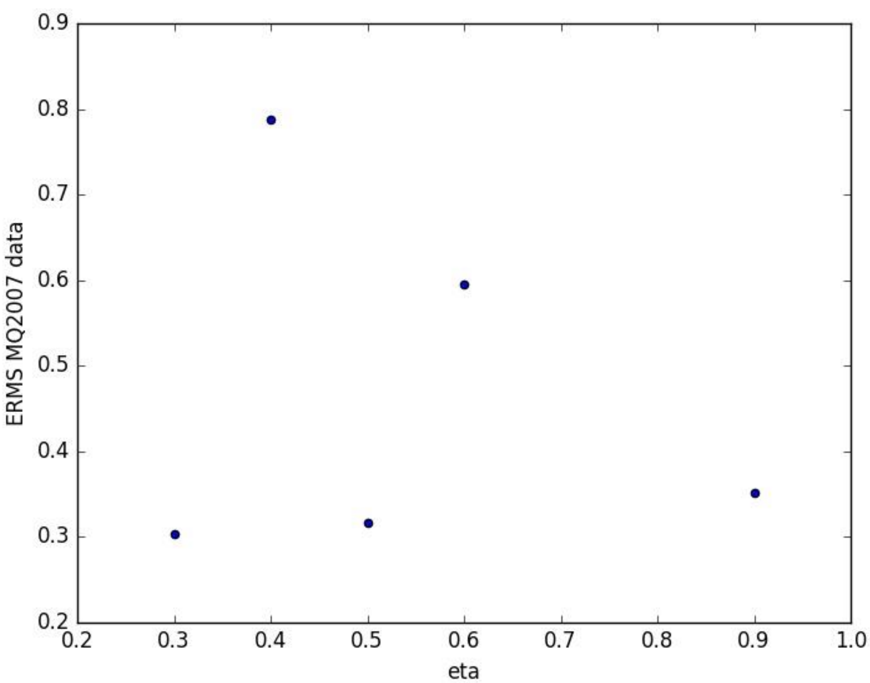
[0.053097 0.4   0.5   0.   0.060177 0.   0.   0.
  0.   0.   0.060861 0.392418 0.615168 0.   0.069793
  0.004749 0.2   0.2   0.138889 0.004954 0.810083 0.512855
  0.505315 0.432562 0.79665   0.688845 0.590998 0.54228   0.876717
  0.802161 0.902848 0.591754 0.   0.   0.   0.   0.793406
  0.494175 0.505217 0.438259 0.011583 0.03012   0.054054 0.6   0.102041
  0.   ]

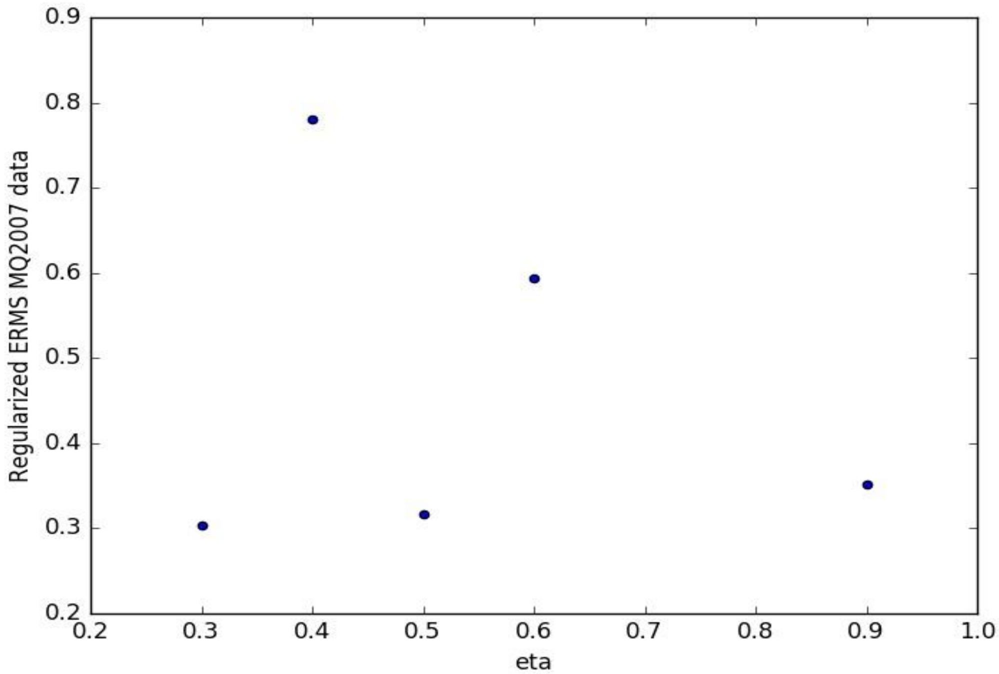
ETA, ƞ:

For concluding on what value of eta as the optimal eta, we train different data sets using the different values of eta, ƞ. The errors that are so obtained are checked and the value of ƞ for which the errors are normal is considered.

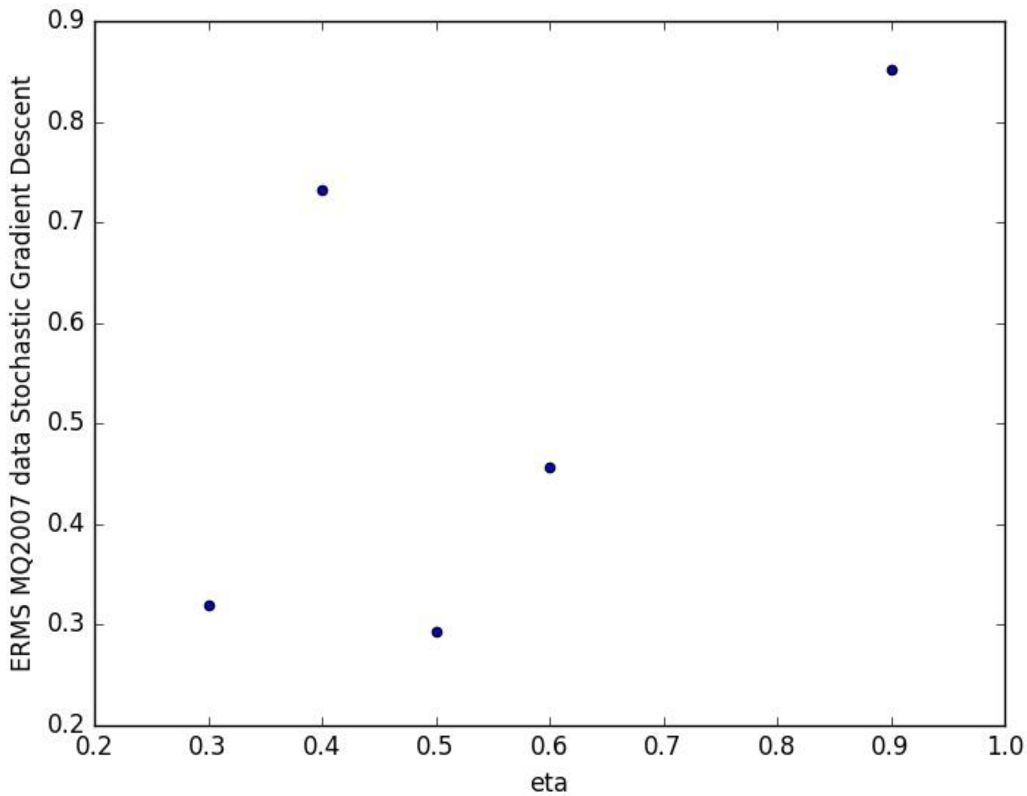
The different values of eta I considered are 0.3, 0.4, 0.5, 0.6, 0.9

The following graphs have been plotted for different values of ƞ and the errors are shown:

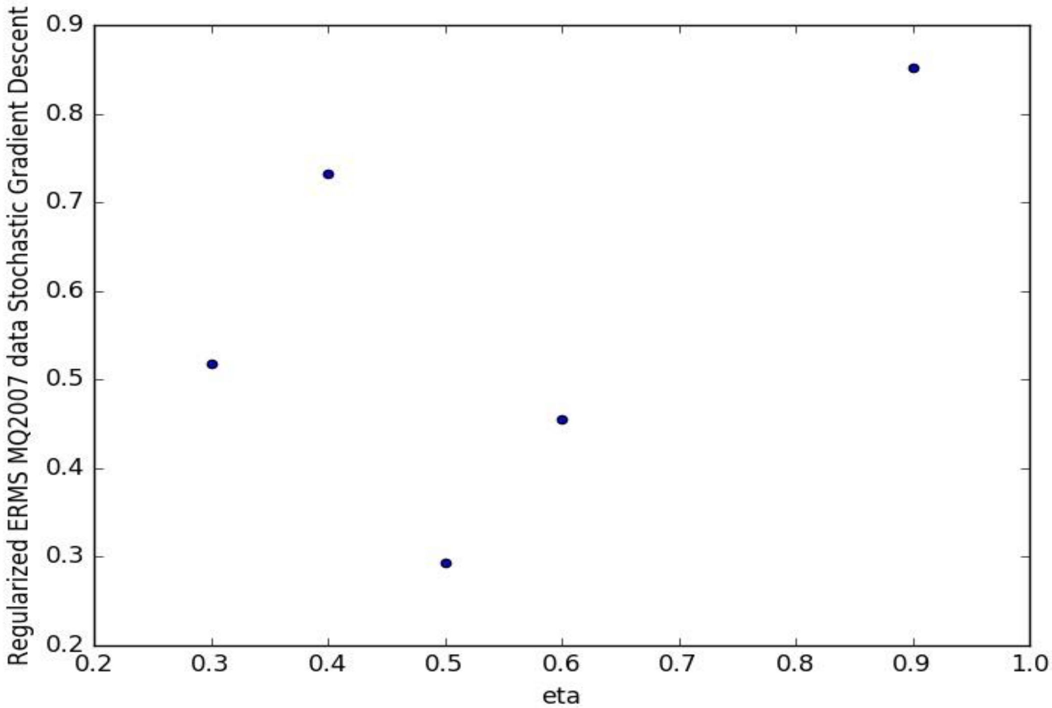


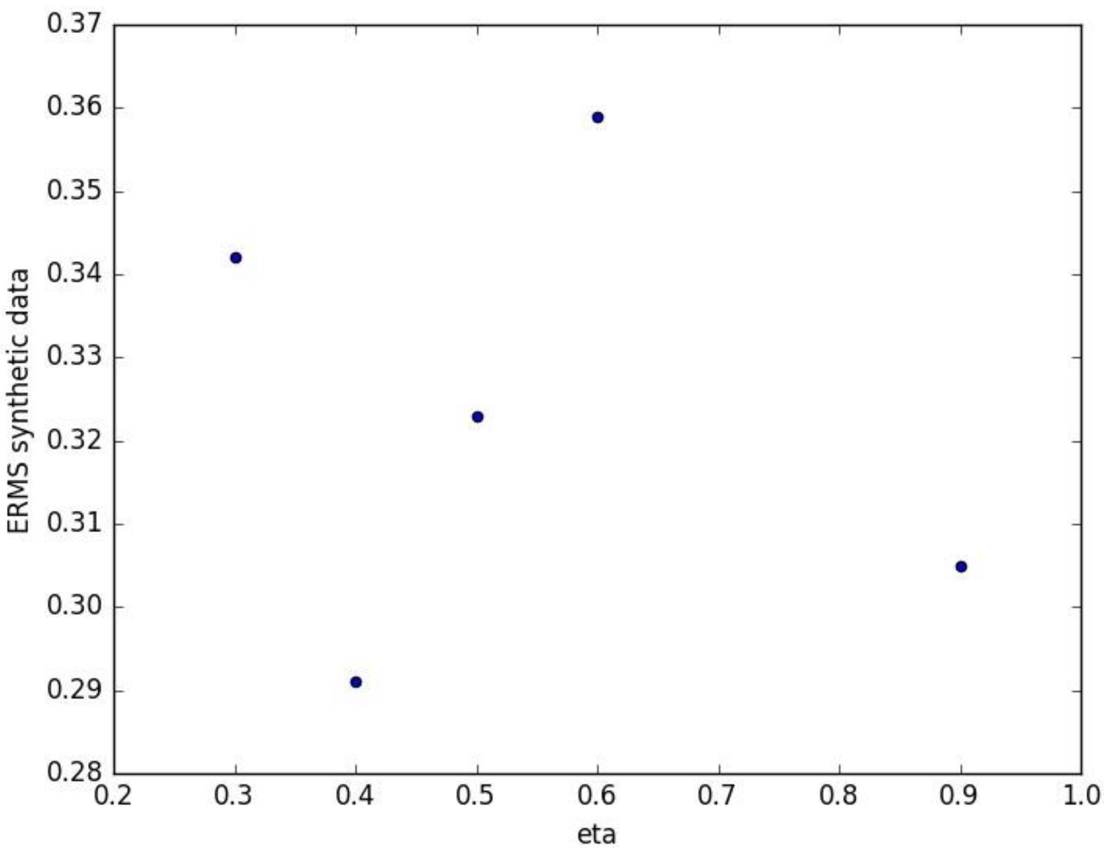


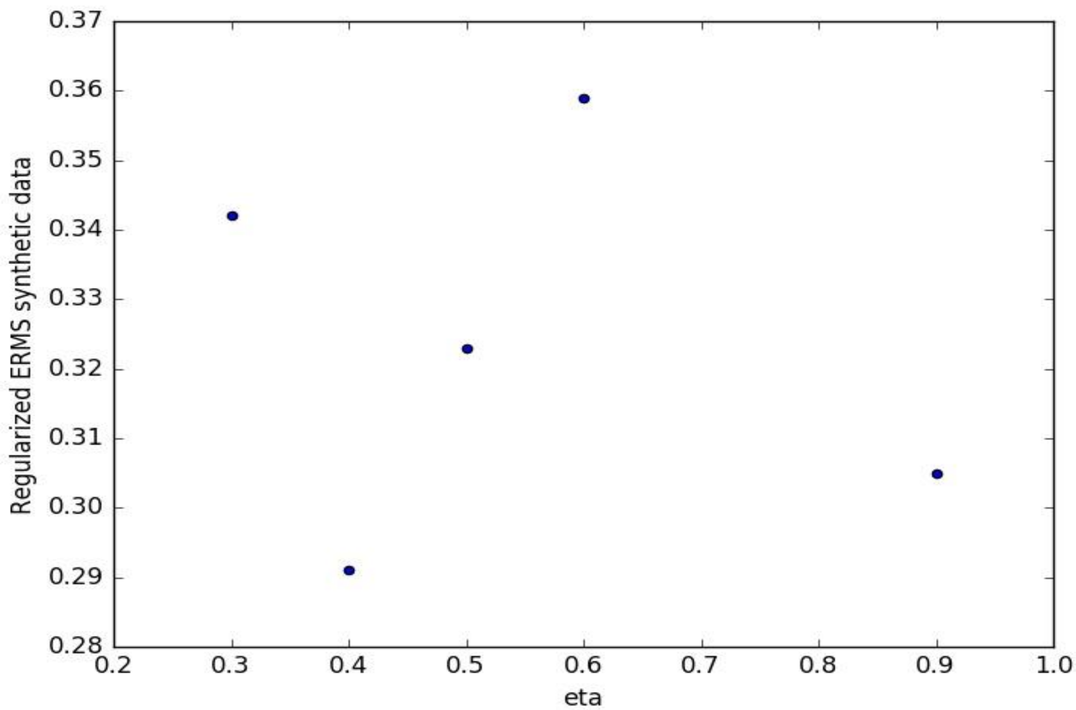
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
Regularized ERMS MQ2007 data
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
eta

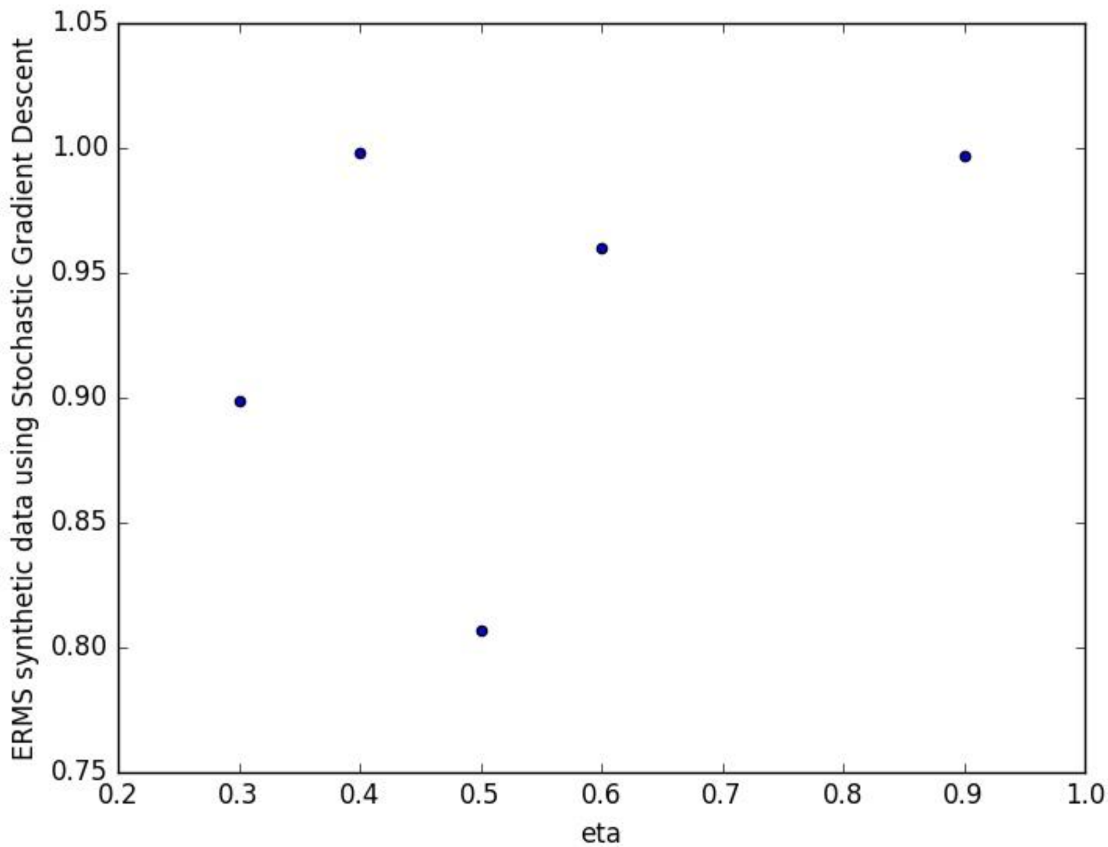


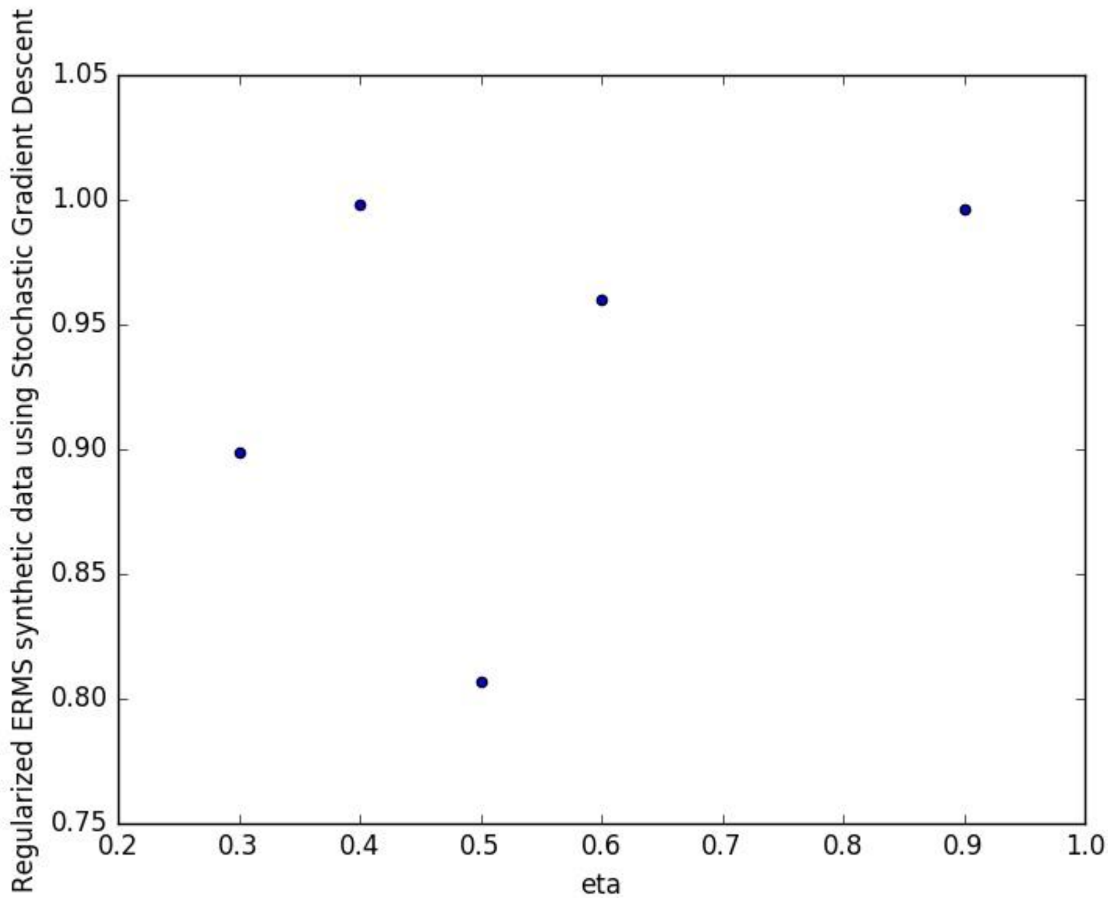
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
ERMS MQ2007 data Stochastic Gradient Descent
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
eta


Regularized ERMS MQ2007 data Stochastic Gradient Descent
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
eta


ERMS synthetic data
0.37
0.36
0.35
0.34
0.33
0.32
0.31
0.30
0.29
0.28
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
eta


Regularized ERMS synthetic data
eta
0.37
0.36
0.35
0.34
0.33
0.32
0.31
0.30
0.29
0.28
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0


ERMS synthetic data using Stochastic Gradient Descent
eta
1.05
1.00
0.95
0.90
0.85
0.80
0.75
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0



Considering all the cases, the value for eta, where the error is normal and not too large, I found 0.5 to be optimal.

The value of eta can either be varying or constant, the method in which the value of eta keeps varying, the values for errors were huge, and so I considered eta to be constant and the value to be 0.5.

Evaluation and Results:

- Root mean square errors are calculated for both the MQ2007 data and the synthetic data.

- Both the regularized and non-regularized models are used and values are calculated.

  For the MQ2007 data set, the ERMS values are as follows:

  RMS Error Calculated from Gaussian Radial Basis Functions
  0.554537098959
  Regularized RMS Error Calculated from Gaussian Radial Basis Functions
  0.55453736253
  RMS Error for training data from Stochastic Gradient Descent Method
  0.795813412558
  Regularized RMS Error for training data from Stochastic Gradient Descent Method
  0.721932813977
  RMS Error for validation data Gaussian Radial Basis Functions:
  0.508438425678
  Regularized RMS Error for validation data Gaussian Radial Basis Functions:
  0.505394831931
  RMS for validation data stochastic gradient descent:
  0.810410444421
  Regularized RMS for validation data stochastic gradient descent:
  0.810046533948
  RMS for testing data Gaussian Radial Basis Functions:
  0.52669102115
  Regularized ERMS for testing data Gaussian Radial Basis Functions:
  0.523134315633
  ERMS for testing data stochastic gradient descent:
  0.822983489506
  Regularized ERMS for testing data stochastic gradient descent:
  0.822613942792

The following values are the ERMS values obtained on the Synthetic data set:

RMS Error Calculated from Gaussian Radial Basis Functions
0.561599739701
Regularized RMS Error Calculated from Gaussian Radial Basis Functions
0.561599792801
RMS Error for training data from Stochastic Gradient Descent Method
0.725651877319
Regularized RMS Error for training data from Stochastic Gradient Descent Method
0.725436301838
RMS Error for validation data Gaussian Radial Basis Functions:
0.630858094051
Regularized RMS Error for validation data Gaussian Radial Basis Functions:
0.629471591535
RMS for validation data stochastic gradient descent:
0.998715004661
Regularized RMS for validation data stochastic gradient descent:
0.998266537033
RMS for Testing data Gaussian Radial Basis Functions:
0.414287813737
Regularized ERMS for Testing data Gaussian Radial Basis Functions:
0.413799415065
ERMS for Testing data stochastic gradient descent:
0.630768794236
Regularized ERMS for Testing data stochastic gradient descent:
0.630485590496