MACHINE LEARNING PROJECT 2 REPORT

UBIT Name: SINDHURA UPPU

Person Number: 50206730

Data Partition:

- The project has a specific requirement in terms of dividing the Microsoft LETOR 4.0 data set and the synthetic data set into training, validation and testing set.
- The data set is supposed to be partitioned into a training set which takes around 80% of the total, a validation set that comprises of 10% of the total and a testing set that takes the rest.
- In view of this requirement, I have identified that there are 2 ways to partition this data set.
- Partition can be done randomly in which any rows that constitute 80% of the total 69623 rows, which is 55601 query-document pairs (rows) will be taken as the training set, 10% of the remaining 20%, which is 7011 rows will be taken as the validation set and the other 7011 (10% of the total) will be taken as the testing set.
- Another way to partition the data is to directly take the first 80% rows as training set, the next 10% rows as the validation set and the rest 10% as the testing set.
- I have partitioned the data using the second way and hence have the following values for the training, validation and testing set:
  - Training data – X and Y matrices that split the main data into 46 feature vectors and the resulting output matrix which contains relevance score. X and Y constitute 55601 rows, which is 80% of the data.
  - Validation data – Xvalid and Yvalid matrices contain next 7011 rows each and have 46 feature vectors and corresponding relevance score.
  - Testing data – Xtest and Ytest matrices contain the remaining 7011 rows each and have 46 feature vectors and the relevance score.

Hyper-parameter Tuning:

- Hyper parameters M, μj, Σj, λ, η(τ) need to be evaluated first in order to proceed further with the linear regression model training.
- Basis function M can be chosen using grid search.

- Alternatively, by assuming M as some integer, say 4 or 5, we can get different weight vectors and train linear regression model.
- Later on, validation data can be trained on the obtained weight vectors and the hyper parameters will then be adjusted if the expected result is ambiguous.
- It should be noted that the value of M should not be too small or too large.
- If it is small, then the model under fits the data.
- If it is too large, then the model over fits the data.
- For this project, I have chosen 4 as M value and trained the model parameter w on the training set initially.
- Weights and regularized weights calculated using both closed form solution and stochastic gradient function are as follows:
- Weight from closed form solution:

```
[[ 0.01588176],
[ 1.39858078],
[-0.87881206],
[-0.08195343]]
```

- Weight from stochastic gradient function:

```
[[ -2.85644991e-04],
[  9.99792989e-01],
[ -2.25282447e-04],
[ -2.32610177e-04]]
```

- Here, for regularized weights and error calculation, λ value is needed.
- λ can also be calculated using grid search. It should be in the interval of 0 and 1 and hence, a random floating point in the range (0,1) is taken as λ.
- Using regularized function λ, the weights are calculated to be as follows:
- Regularized weight using closed form solution:

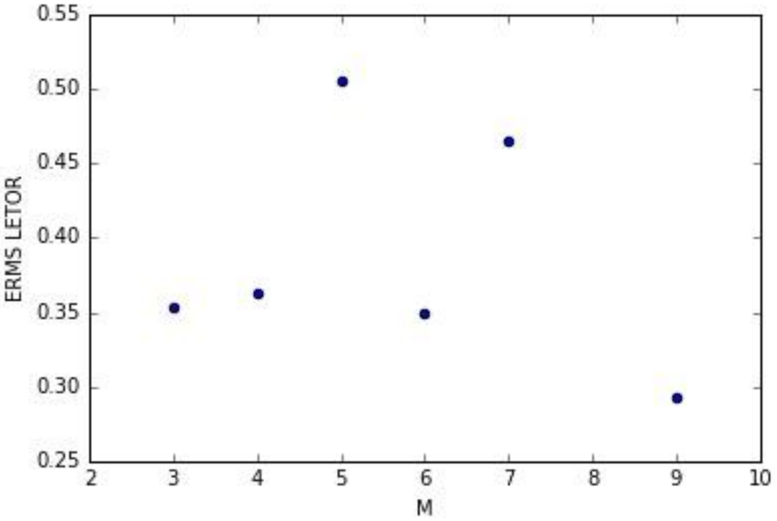
```
[[ 0.01696059],
[ 1.39168956],
[-0.87625263],
[-0.07928387]]
```

- Regularized weight using stochastic gradient descent function:

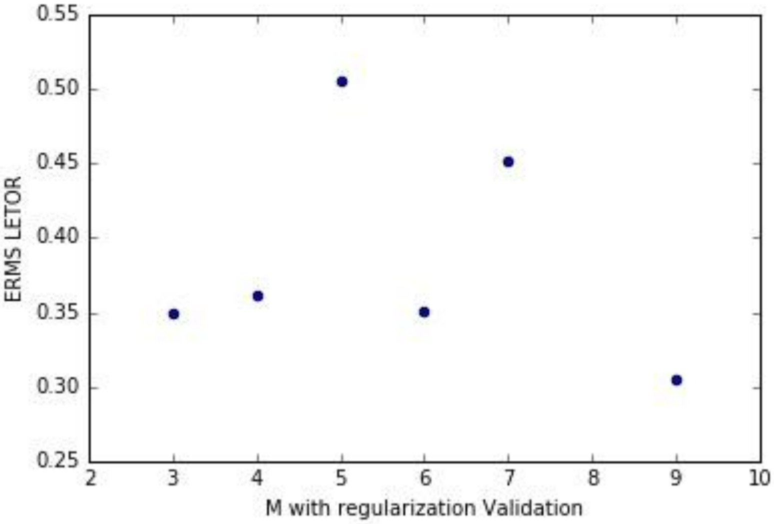
```
[[ -2.85511518e-04],
[  9.99433444e-01],
[ -2.25178493e-04],
[ -2.32501222e-04]]
```

- μj, Σj, η(τ) need to be calculated before the calculation of phi matrix too, the explanation of which is covered in the upcoming sections.
- After obtaining weight vector, the validation data is used to validate the accuracy of the model trained using the hyper parameters assumed in the initial stage.

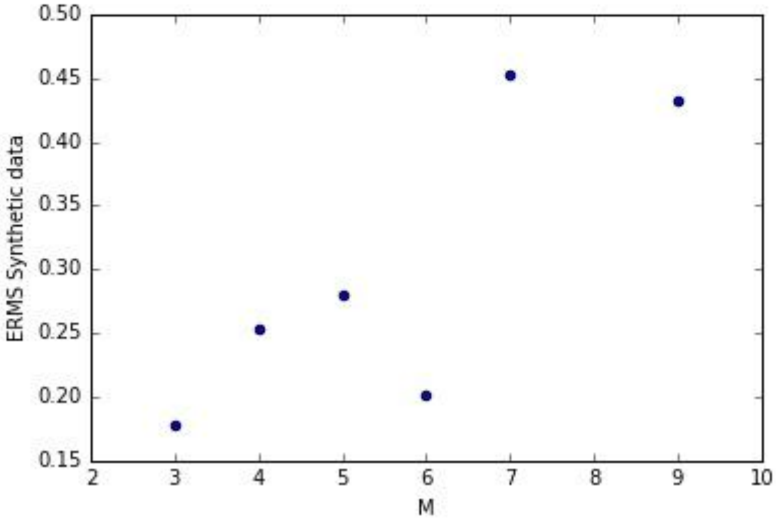
- By taking different values of M, we obtain different weights and Root mean square errors.
- The following graphs represent values of M, (3,4,5,6,7,9) and the corresponding Root Mean Square errors.



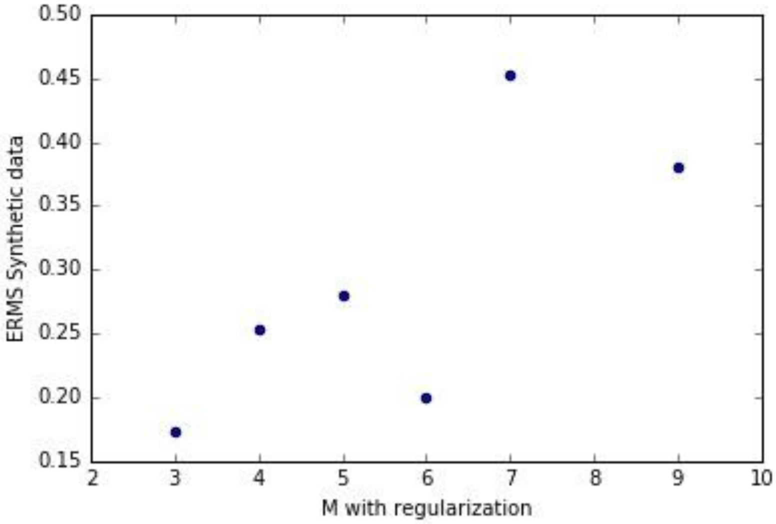
M vs ERMS Microsoft LETOR using closed form solution



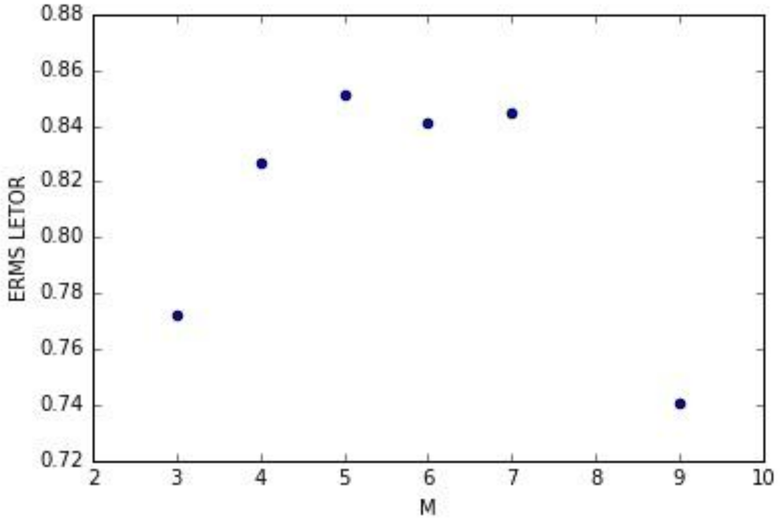
M vs ERMS for Microsoft LETOR using regularized closed form solution



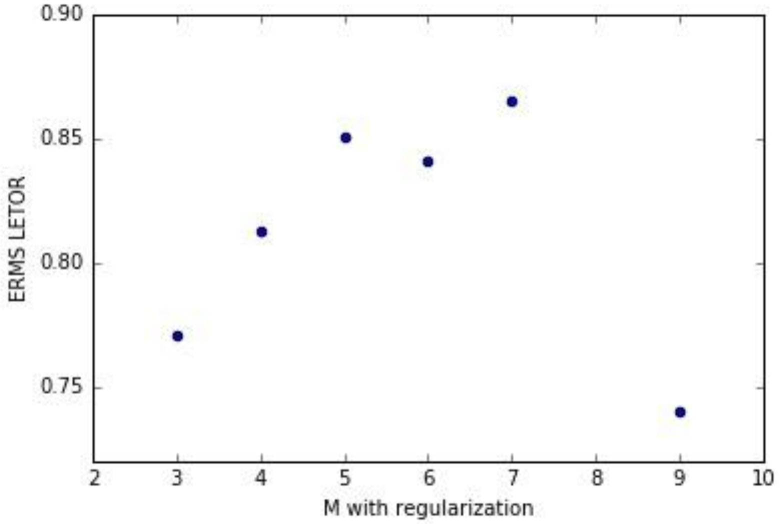
M vs ERMS for synthetic data using closed form



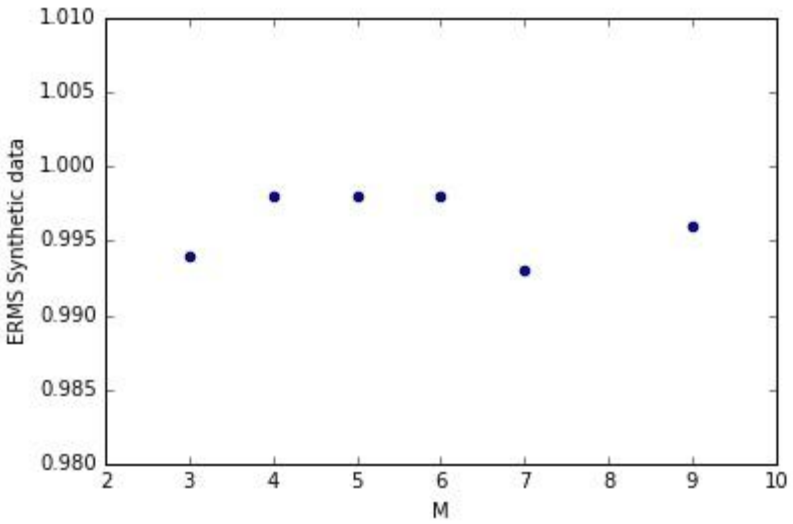
M vs ERMS for synthetic data using regularized closed form



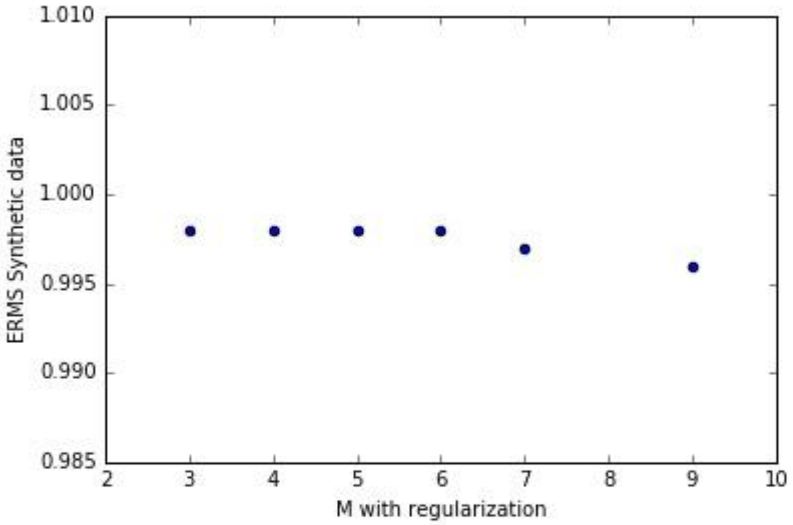
M vs ERMS for Microsoft LETOR using Stochastic Gradient Solution



M vs ERMS for Microsoft LETOR using regularized Stochastic Gradient solution



M vs ERMS for synthetic data using Stochastic Gradient solution



M vs ERMS for synthetic data using regularized Stochastic Gradient solution

- 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, µ, can be calculated using k-means clustering or simply, any M-1 rows from the data set can be considered as µ matrix.
- For the first column of phi matrix, we do not need to calculate the value of phi(x) as it 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:

```
[ 0.002609  0.        0.        0.66667   0.00301   0.        0.        0.
  0.        0.        0.82146   0.        0.        0.66667   0.82911
  0.008885  0.        0.875     0.64286   0.008731  0.7333    0.76702
  0.80045   0.75185   0.        0.        0.        0.        0.        0.
  0.        0.        0.90356   0.32356   0.        0.        0.75903
  0.79938   0.83132   0.7799    0.        0.        0.045249  0.66667
  0.74766   0.      ]
[  0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00
   0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00
   0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00
   0.00000000e+00   0.00000000e+00   0.00000000e+00   9.63000000e-04
   0.00000000e+00   0.00000000e+00   1.00000000e+00   0.00000000e+00
   1.72650000e-01   8.77020000e-01   1.26660000e-02   9.27960000e-01
   0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00
   1.00000000e+00   1.00000000e+00   1.00000000e+00   1.00000000e+00
   0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00
   2.02440000e-01   1.00000000e+00   1.74390000e-02   9.17460000e-01
   0.00000000e+00   0.00000000e+00   0.00000000e+00   1.00000000e+00
   8.00000000e-01   0.00000000e+00]
[ 0.25694   0.        0.33333   0.        0.25685   0.        0.        0.
  0.        0.        0.25095   0.        0.38826   0.        0.25093
  0.054732  0.        0.66667   0.36364   0.054955  0.85453   0.75632
  0.91316   0.7441    0.        0.        0.        0.        0.47003
  0.21246   0.44802   0.17072   0.        0.        0.        0.        0.83813
  0.74643   0.91406   0.75391   0.        0.        0.081356  0.66667
  0.25714   0.      ]
[ 0.16956   0.        0.        0.        0.16956   0.        0.        0.
  0.        0.        0.19307   0.        0.        0.        0.19503
  0.30788   0.047619  0.        0.5       0.30777   0.35746   0.74931
0.79274   0.679     0.        0.        0.        0.        0.        0.
0.        0.        0.        0.        0.        0.        0.36537
0.74518   0.79339   0.68249   0.003094  0.006993  0.        0.5       0.34483
0.      ]
```

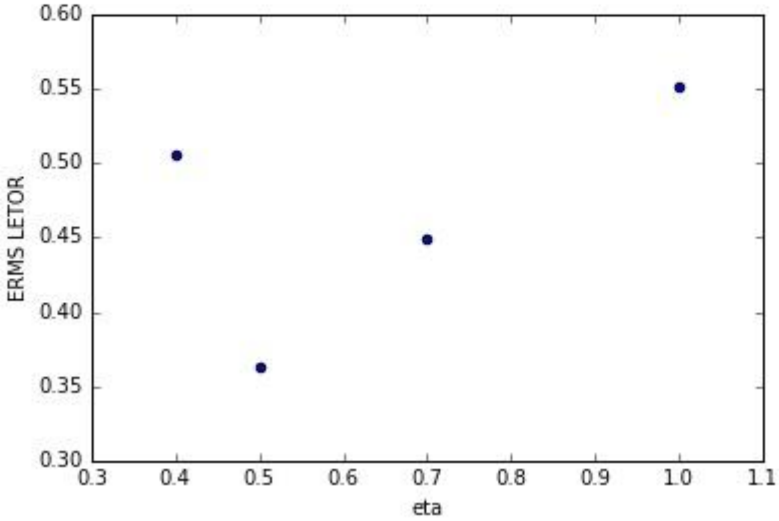
SIGMA Matrix:

- Basis function Σ is considered to be a diagonal matrix in which each diagonal element is chosen to be proportional to the ith dimension variance of the training data.
- Value of the diagonal Sigma matrix:

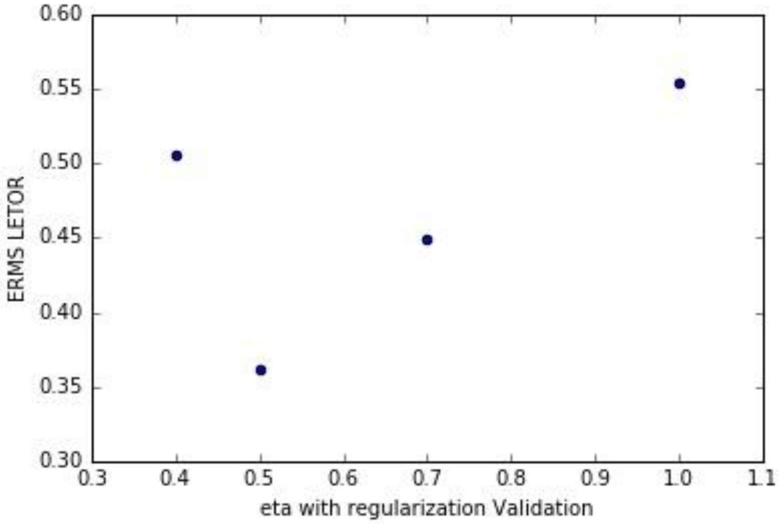
```
[[ 0.0538377 ,  0.        ,  0.        , ...,  0.        ,                 0.        ,  0.        ],

 [ 0.        ,  0.066607  ,  0.        , ...,  0.        ,          0.        ,  0.        ],

 [ 0.              ,  0.        ,  0.11956028, ...,  0.        ,                0.        ,  0.        ],

   ...,

 [ 0.        ,  0.        ,  0.        , ...,  0.06767358,          0.        ,  0.        ],

 [ 0.        ,  0.        ,  0.        , ...,  0.        ,            0.06476424,  0.        ],

[ 0.        ,  0.        ,  0.        , ...,  0.        ,            0.        ,  0.        ]]
```

ETA (η(τ)):

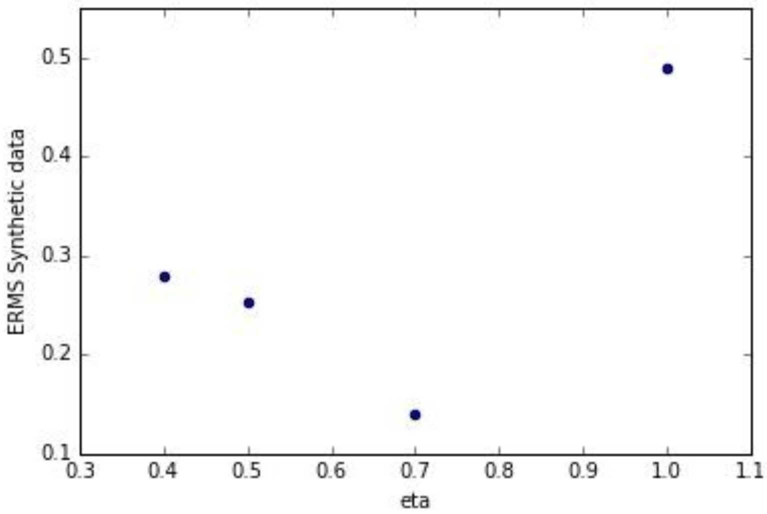
- Learning rate, η(τ) can be either fixed or variable.
- The learning rate should be in the range of 0 and 1, hence I have assumed it to be 0.5 (fixed) in the initial calculation.
- The weights that are obtained using this value of Eta are presented in the initial sections of this report.
- During validation, I have employed multiple values of Eta such as 1, 0.7, 0.4 for which the Root Mean Square error is obtained as shown in the graphs below:



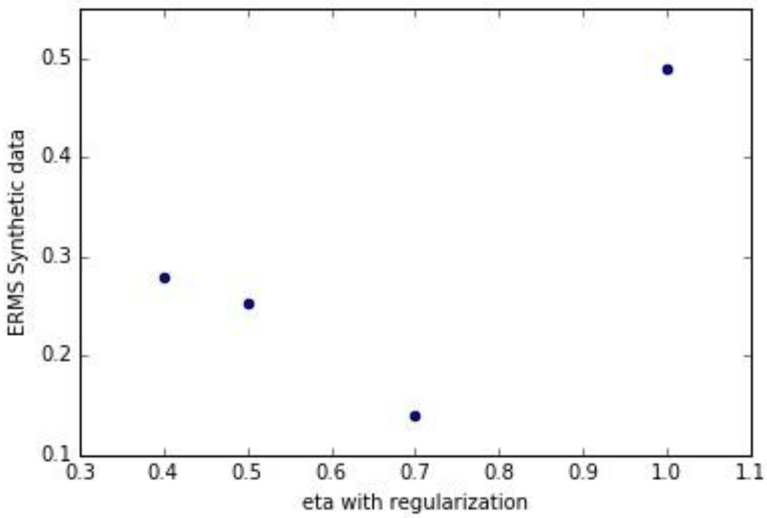
Eta vs ERMS for LETOR using Closed form solution



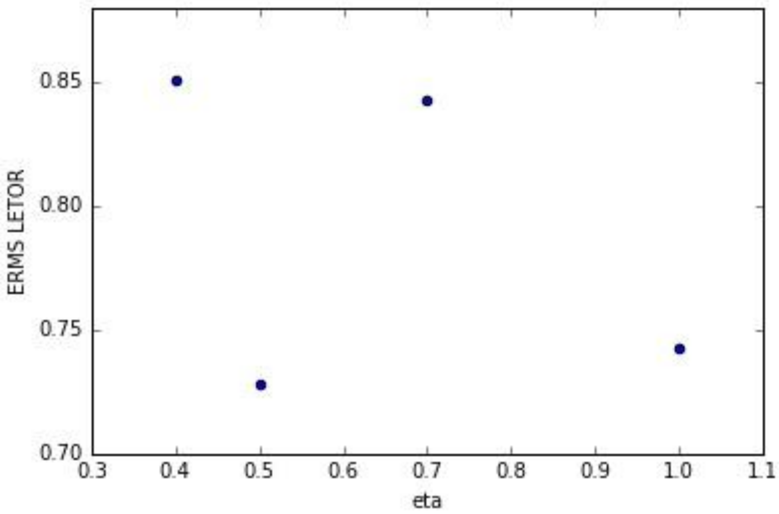
Eta vs ERMS for LETOR using regularized closed form solution



Eta vs ERMS for Synthetic Data set using closed form solution

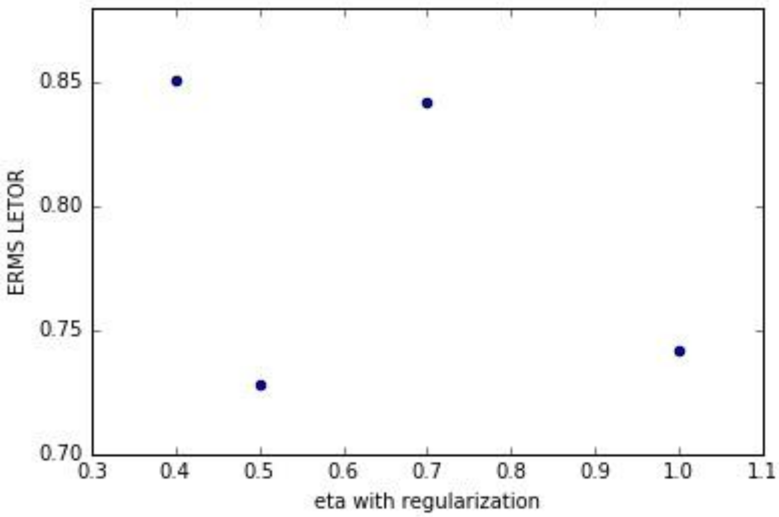


Eta vs ERMS for Synthetic Data set using regularized closed form solution

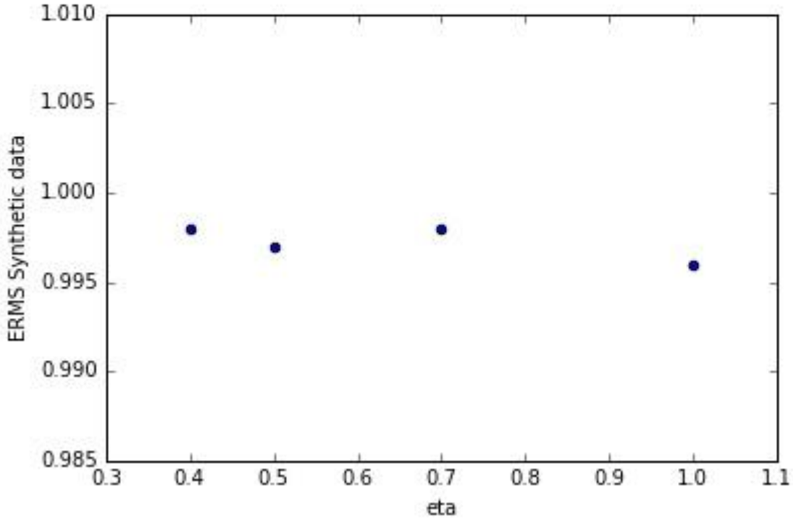


c

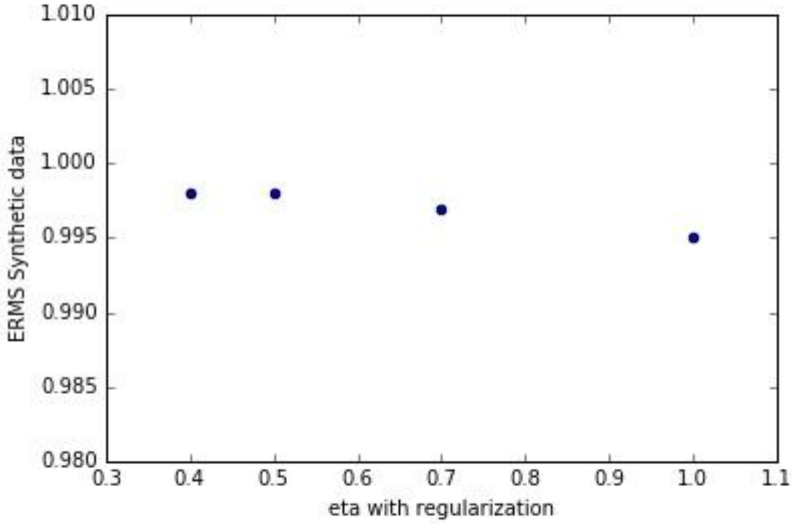
Eta vs ERMS for LETOR using Stochastic Gradient Solution



Eta vs ERMS for LETOR using regularized Stochastic Gradient Solution



Eta vs ERMS for Synthetic Data set using Stochastic Gradient Solution



Eta vs ERMS for Synthetic Data set using regularized Stochastic Gradient Solution

- It can be inferred that ERMS is optimal when Eta is 0.5 in all forms as when compared to other values of Eta.

Evaluation and Results:

- Root mean square error is calculated for both real and synthetic data sets, once each using closed form and stochastic gradient descent solution.
- In both the techniques, ERMS is calculated once each for normal and once for regularized training model.

ERMS values obtained for each data set is as follows:

Microsoft LETOR 4.0 Data Set:

- ERMS Closed form training set: 0.55846883620132348
- Regularized ERMS Closed form training set: 0.55846893302858491
- ERMS Closed form validation set: 0.46928333039226672
- Regularized ERMS Closed form validation set: 0.46885073135609101
- ERMS Closed form testing set: 0.29030615901370027
- Regularized ERMS Closed form testing set: 0.29003168214524172
- ERMS Stochastic form training set: 0.74455508171070861
- Regularized ERMS Stochastic from training set: 0.74436854188098256
- ERMS Stochastic form validation set: 0.74524333287708522
- Regularized ERMS Stochastic form validation set: 0.7449754013671861
- ERMS Stochastic form testing set: 0.70752334009354445
- Regularized ERMS Stochastic form testing set: 0.70726898716570252

Synthetic Data Set:

- ERMS Closed form training set: 0.56221005322597428
- Regularized ERMS Closed form training set: 0.56221006922533234
- ERMS Closed form validation set: 0.51587271052453054
- Regularized ERMS Closed form validation set: 0.51535635982708017
- ERMS Closed form testing set: 0.42107854333796224
- Regularized ERMS Closed form testing set: 0.42065934227146451
- ERMS Stochastic form training set: 0.68577744695566878
- Regularized ERMS Stochastic from training set: 0.68563281770423523
- ERMS Stochastic form validation set: 0.99904945136222634
- Regularized ERMS Stochastic form validation set: 0.99869025270085221
- ERMS Stochastic form testing set: 0.68062343362392663
- Regularized ERMS Stochastic form testing set: 0.68037875603226983