Project 2

# Learning to Rank using Linear Regression

*CSE 574*
*Introduction to Machine Learning*

Anirudh Reddy Nalamada
UB No.: 5016-9240
UBIT Name: aniredn

## 1. Introduction

The main aim of this project is to train and test a linear regression model on two sets of data using the Closed Form Maximum Likelihood method (Batch method) and the Stochastic Gradient Descent method.

The first set of data was a real world dataset, Microsoft LETOR 4.0 Dataset. LETOR is a package of benchmark data sets for research on Learning to Rank released by Microsoft Research Asia. The real dataset consists of 69623 entries and each entry has 46 features.

The second set of data is a synthetically generated dataset consisting of 2000 entries and each having 10 features.

## 2. Formulae Used

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

$$\mathbf{y(x,w) = w^T \phi(x)}$$

**Closed Form Maximum Likelihood Solution**

1. Basis Function (Gaussian Radial Basis Function)

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

2. Design Matrix $\phi$ has the form as below

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

3. The maximum likelihood is calculated as below

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

**Stochastic Gradient Descent Solution for w**

1. The weights are updated as below

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \Delta \mathbf{w}^{(\tau)}$$

2. Weight updates are calculated using the formula below

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

$$\Delta \mathbf{w}^{(\tau)} = -\eta^{(\tau)} \nabla E$$

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

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

3. Calculation of Root Mean Square Error, where λ is the regularization term.

$$E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^*)/N_{\mathrm{V}}}$$

$$E(\mathbf{w}) = E_D(\mathbf{w}) + \lambda E_W(\mathbf{w})$$

$$E_D(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{t_n - \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_n)\}^2$$

$$E_W(\mathbf{w}) = \frac{1}{2} \sum_{j=0}^{M-1} |w_j|^2$$

## 3. Work Done

**Data Partitioning**

The real world data set was downloaded and imported using the Import Data App on MATLAB.

The real world data set consisting of 69623 entries was split into 3 parts

1. Training Set -80% - 55700 entries – rN1
2. Validation Set -10% - 6960 entries – rN2
3. Testing Set - 10% - 6963 entries (Not Used)

The Synthetic data consisting for 2000 entries was loaded into MATLAB and split into 2 parts.

1. Training Set – 80% - 1600 entries – sN1
2. Validation Set – 20% - 400 entries – sN2

**Closed Form Maximum Likelihood method for real and Synthetic data sets**

The variables were chosen in the following way for the batch method.
**mu1 and mu2** was chosen randomly from the training data set using the randi() function. Hence each time the program is executed the value of mu1 is different.
**Sigma1 and Sigma2** was calculated using the cov() function and all the variables except the diagonal were set to zero. The elements lying on the diagonal whose value was very close to zero or zero were set to a significant non-zero value so that the determinant is not zero.

The values for the regularization terms **lambda1** and **lambda2** and the number of basis functions **M1** and **M2** are discussed in the Optimization section below.

The Design matrix was computed on using the training data and it was used along with the value of target data to compute the value of maximum likelihood solution ($\mathbf{w_{ML}}$). $\boldsymbol{\lambda}$ was included in the calculation of the maximum likelihood solution for w to decrease the problem of over-fitting. A Design matrix was also computed in a similar way using the validation data.

**Computation of Root Mean Square Error**

Root mean square error was calculated for both the training set and validation set and used to evaluate the performance of the chosen hyper parameters. The regularization term was not included in the calculation of the root mean square error.

**Stochastic Gradient Descent solution**

The initial value of weight vector $w^{\tau}$ was chosen randomly using the rand() function.

The value of $\boldsymbol{\eta}$ was set to 1 initially and updated accordingly during the iterations. If the value of root mean square error calculated using the weights $w^{\tau+1}$ increases compared to the $w^{\tau}$ then we decrease the value of $\boldsymbol{\eta}$ and if the root mean square $w^{\tau+1}$ decreases compared to the $w^{\tau}$ then the value of $\boldsymbol{\eta}$ is increased.

The number of iterations was set equal to the number of entries in the training set so that we iterate over all the entries in the training set before arriving at a solution.

## 4. Optimization of Hyper Parameters

**Choosing M and Lambda:**

The number of basis functions (M) and the value of regularization term (lambda) were chosen based on a grid search method. The values of **mu1** and **mu2** were chosen randomly and kept constant and values of lambda and M were changed. For each combination of M and lambda the root mean square error (validPer1/validPer2) was

calculated using the validation data. The lowest value of **validPer1** and **validPer2** should correspond to the optimal value of lambda and number of basis functions.

**Table 1: ERMS values for Real Data (validPer1)**

| Lambda1\M 1 | 6 | 8 | 10 | 12 | 14 |
|---|---|---|---|---|---|
| 0.4 | 0.553701451 | 0.553704093 | **0.55369918** | 0.55369931 | 0.553701918 |
| 0.5 | 0.553701926 | 0.553704395 | 0.553699556 | 0.553699678 | 0.553702111 |
| 0.75 | 0.553702936 | 0.553705058 | 0.553700394 | 0.5537005 | 0.553702583 |
| 1 | 0.55370375 | 0.553705611 | 0.55370111 | 0.553701203 | 0.553703024 |
| 2 | 0.55370586 | 0.553707106 | 0.553703157 | 0.55370322 | 0.55370443 |

The value of **M1** from the above data should be 10 and **lambda1** 0.4.

**Table 2: ERMS values for Synthetic Data (validPer2)**

| Lambda2\M2 | 4 | 5 | 7 |
|---|---|---|---|
| 0.3 | 0.155105073 | 0.155056071 | 0.155057902 |
| 0.5 | 0.155113909 | 0.155071241 | 0.155072785 |
| 0.7 | 0.155120518 | 0.155082715 | 0.155084049 |
| 1 | 0.155127719 | 0.155102096 | 0.155096518 |

The value of **M2** from the above data should be 5 and **lambda2** 0.3.

## Choosing η

For real and synthetic data the value of eta was initialized to 1and 0.8 respectively and updated as below.

- If the error value increases in the successive iteration then the value of eta was decreased by 50%.
- If the error value decreases in the successive iteration then the value of eta was increased by 10%
- If the error value remained constant then the eta was kept constant.

## Choosing $w^{\tau}$

The value of $w^{\tau}$ (w01 and w02) was chosen randomly using the rand() function.

## 4. Results and Discussion:

### Closed Form Maximum Likelihood Solution

Real Data

To compute the solution for the batch method on real data, M1 was chosen as 10, lambda1 was tuned to be 0.4.
ERMS of training set = 0.5643
ERMS of validation set = 0.5537

Synthetic Data

To compute the solution for the batch method on synthetic data, M2 was chosen as 5, lambda2 was tuned to be 0.3.
ERMS of training set = 0.1450
ERMS of validation set = 0.1552

### Stochastic Gradient Descent Method

Real Data

To compute the weight using the stochastic gradient descent method, 55700 iterations were done on the training data.
ERMS of training set = 0.5813
ERMS of validation set = 0.5734

Synthetic Data

To compute the weight using the stochastic gradient descent method, 1600 iterations were done on the training data.
ERMS of training set = 0.2065
ERMS of validation set = 0.2117

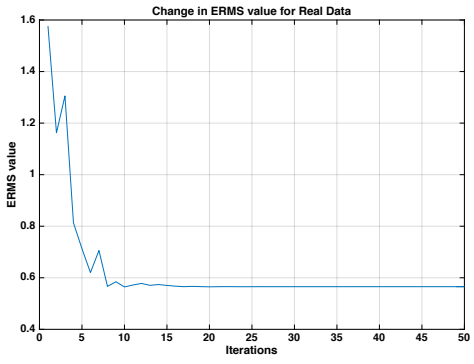


Figure 1: ERMS value for Real Data

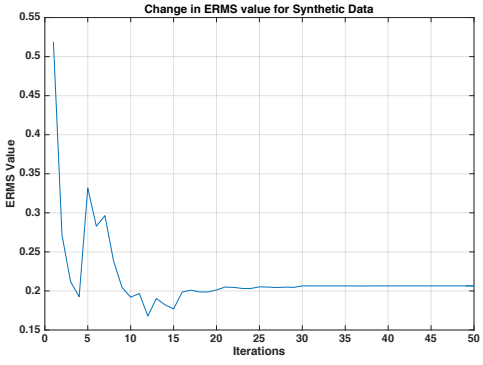


Figure 2: ERMS value for Synthetic Data

From the above data we can conclude that the Closed Form Maximum Likelihood Solution has given us a lesser error when compared to the Stochastic Gradient Descent Solution.

## 6. References


- Daphne Koller and Nir Friedman, "Probabilistic graphical models: principles and techniques", MIT Press 2009
- Christopher M. Bishop, “Pattern Recognition and Machine Learning”, Springer 2006
- https://en.wikipedia.org/wiki/Gradient descent.
- https://en.wikipedia.org/wiki/Hyperparameter optimization#Grid search.
- Prof. Srihari’s Lecture Slides