STATE UNIVERSITY OF NEW YORK AT BUFFALO

Project Report 2

Learning to Rank using Linear Regression

CS 574 Introduction To Machine Learning

Aditi Asthana (50169677)

11/01/2015

Introduction:

The goal of this project report is to describe the implementation of linear regression to solve regression problem and evaluate its performance. The objective is to learn how to map an input vector x into a scalar target t.

Data Set:

Two different type of data sets have been provided to work on. They were:

1. **Microsoft LETOR 4.0 Dataset** : real world dataset
   LETOR is a package of benchmark data sets for research on 'Learning To Rank' released by Microsoft Research Asia.
   This data set is divided into 3 parts:
   a. Training Data Set (80%)
   b. Validation Data Set (10%)
   c. Test Data Set (10%)
2. **Synthetic Dataset :** some data is generated using mathematical formula:

$$y = f(x) + €$$

   where € is some noise.

   This data set is divided into 3 parts:
   a. Training Data Set (80%)
   b. Validation Data Set (10%)
   c. Test Data Set (10%)

Linear Regression Model:

The linear Regression function y(x,w) can be described as:

$$y(x,w) = w^T \varphi(x)$$

where w: $(w_0,\ldots\ldots w_{M-1})$ is a weight vector which has to be learn from training data

ϕ(x): vector of M basis function

In this project, Gaussian radial basis functions have been used, which can be computed as:

$$\Phi j(x) = \exp((-1/2) * (x-\mu j)' * \text{inv}(\sum j) * (x-\mu j))$$

Where x: input vector x

μj: center of the basis function (mean- computed as random data points of data)

size: 1*M

∑j: describes how broadly the basis function spreads (standard deviation of data)

Size: 46*46*M

Where 46: number of features

To solve linear regression model optimal 'w' needs to be evaluated. It can be evaluated in below 2 ways:

1. Closed Form Solution

Using the above basis function, design matrix is computed as below:

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

The design matrix is of the order N*M where N is the number of training data and M is the number of basis functions.

Solution to maximum likelihood solution with quadratic regularization has the form:

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

After varying multiple lambda values, we determine optimal W for the data. We can do that by determining minimum error for the validation set i.e. choose lambda which yields minimum error for validation data set and thus compute optimal W for the same.

2. Stochastic Gradient Descent

In this method optimal W is calculated by the following formula:

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

And

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

Where DeltaW: weight update

Eta: learning rate, decides how big the each update step would be

The rest formulas are computed as follows:

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

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

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

After every update, we need to determine whether the error is increasing or decreasing. If its increasing we need to update the eta value to its half and then compute the modified W. The stopping criteria would be when the error update is very small. We need to iterate atleast N times (i.e. number of data set).

Error Evaluation:

The error is evaluated on the dataset using Root Mean Square (RMS) error, which is evaluated as:

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

where w* : optimal w

E(w*): squared error function

N: size of data set

Choosing Parameters and Hyper-parameters:

1. Number of Basis Functions M and Regularization Term λ:

For tuning M and λ, we need to do grid search i.e. start with the small values of M and λ, and gradually move to the bigger value until the error minimizes or performance increases.

The result obtained by following this process for **real data set** is as follows:

1. M1 = 30 , Lambda1 = 0.05, Validation Error = 0.5615, Training Error = 0.5721, Test Error = 0.5721
2. M1 = 30 , Lambda1 = 0.5, Validation Error = 0.5695, Training Error = 0.5727, Test Error = 0.5596
3. M1 = 20 , Lambda1 = 0.05, Validation Error = 0.5777, Training Error = 0.5701, Test Error = 0.6430
4. M1 = 10 , Lambda1 = 0.5, Validation Error = 0.5709, Training Error = 0.5712, Test Error = 0.5716
5. M1 = 5 , Lambda1 = 0.05, Validation Error = 0.5673, Training Error = 0.5715, Test Error = 0.5733
6. M1 = 5 , Lambda1 = 0.5, Validation Error = 0.5706, Training Error = 0.5716, Test Error = 0.5692
7. M1 = 4 , Lambda1 = 0.05, Validation Error = 0.5775, Training Error = 0.5702, Test Error = 0.5734

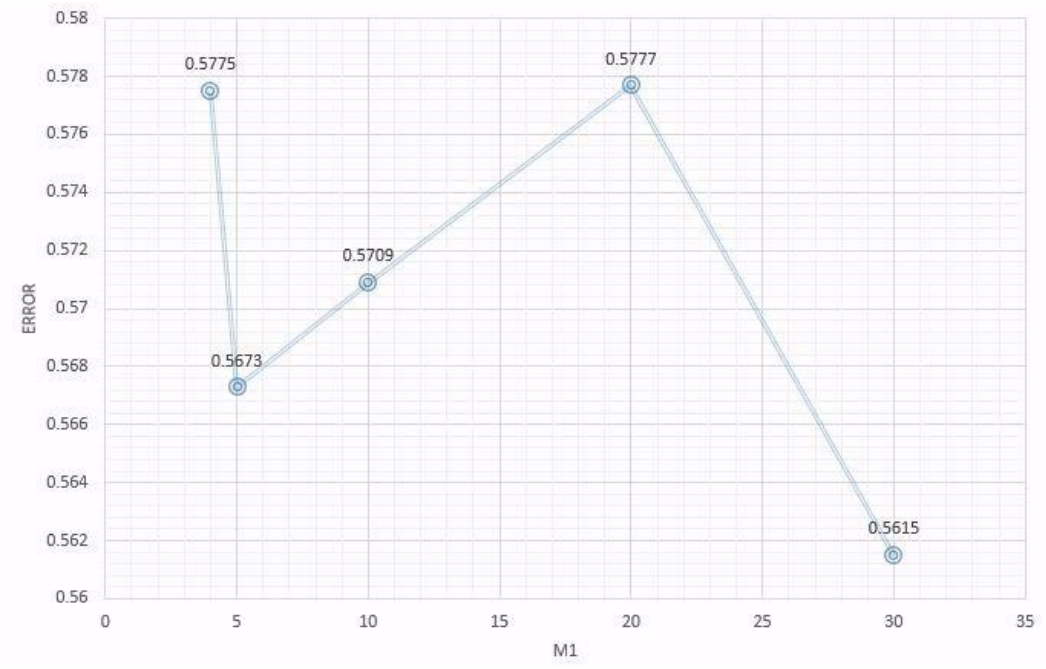

Figure 1: Validation Error Vs M1

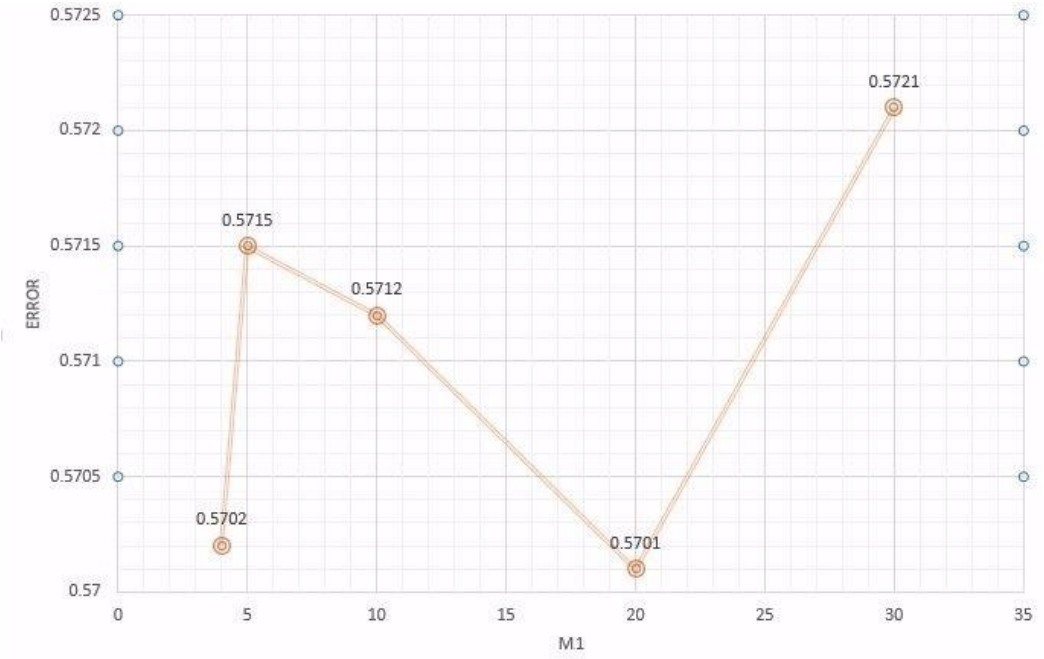

Figure 2: Training Error Vs M1

Optimal Solution chosen:

M1 = 30, Lambda1 = 0.5, Validation Error = 0.5695, Training Error = 0.5727, Test Error = 0.5596

The result obtained by following this process for **synthetic data set** is as follows:

1. M2 = 4 , Lambda2 = 1.0, Validation Error = 0.1528, Training Error = 0.1447, Test Error = 0.1596
2. M2 = 5 , Lambda2 = 0.1, Validation Error = 0.1493, Training Error = 0.1485, Test Error = 0.1334
3. M2 = 5 , Lambda2 = 1.0, Validation Error = 0.1547, Training Error = 0.1452, Test Error = 0.1555
4. M2 = 10 , Lambda2 = 0.1, Validation Error = 0.1403, Training Error = 0.1479, Test Error = 0.1419
5. M2 = 20 , Lambda2 = 0.1, Validation Error = 0.1415, Training Error = 0.1484, Test Error = 1.8640
6. M2 = 30, Lambda2 = 1.0, Validation Error = 0.1410, Training Error = 0.1468, Test Error = 0.1501

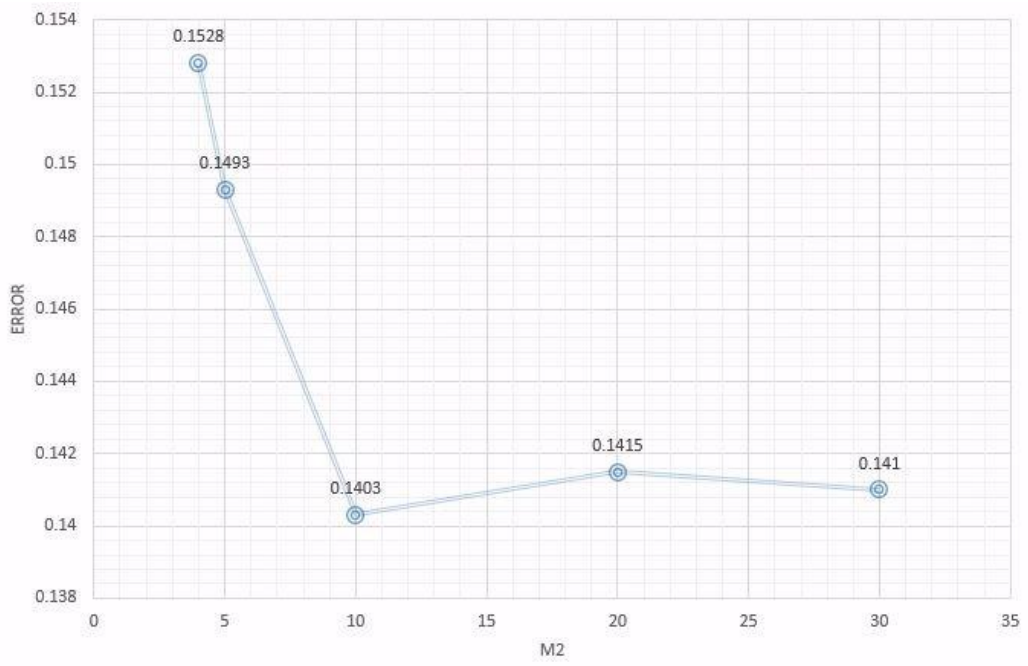

Figure 3: Validation Error Vs M2

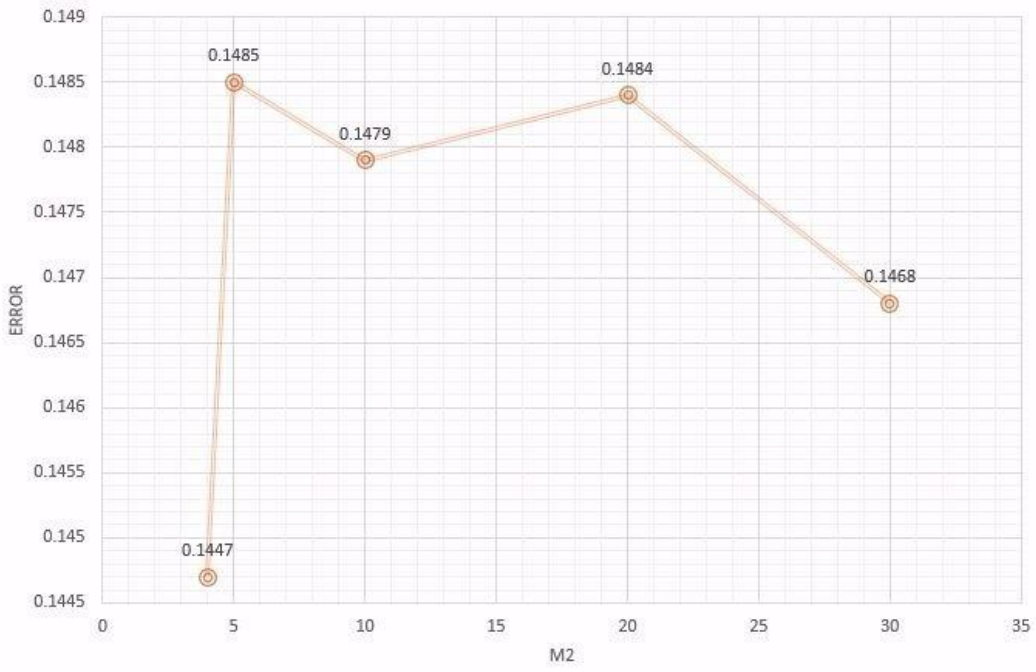

Figure 4: Training Error Vs M2

Optimal Solution chosen:

M2 = 5, Lambda2 = 0.1, Validation Error = 0.1493, Training Error = 0.1485, Test Error = 0.1334

2. Centers for Gaussian Radial basis Functions μ_j:

Random M data points from the training data set were picked to act as the centers.

3. Spread for Gaussian Radial basis Functions $\sum_j$:

Diagonal matrix of the variance is picked to serve the purpose.

4. Learning Rate η:

In Gradient Descent, starting learning rate is set to 1 and it decreases to 0.5 times if the error is increased in the subsequent iterations

Overfitting Avoidance:

The regularization parameter reduces overfitting, which reduces the variance of your estimated regression parameters; however, it does this at the expense of adding bias to your estimate. Increasing lambda results in less overfitting but also greater bias.

References:

1. https://en.wikipedia.org
2. Project Description of 'Project 2: Learning to Rank using Linear Regression'
   CSE 574 : Introduction to Machine Learning (Fall 2015)