Machine Learning Project 1.2

Dhanashree Solanke
Department of Computer Science
University at Buffalo
Buffalo, NY 14214
ddsolank@buffalo.edu

Abstract

Learning to rank is the application of machine learning, where the system is supposed to rank the information retrieval models. For example, on a scale of 0 to 2, how is the IR model responding to the queries of the user. The model is typically supervised, semi-supervised or reinforcement learning system. We take up the LeToR dataset and apply linear regression with closed form solution to generate our model.

1 LeToR Dataset

The LeToR training data consists of pairs of input values **x** and target values *t*. The input values are real-valued vectors (features derived from a query-document pair). The target values are scalars (relevance labels) that take one of three values 0, 1, 2: the larger the relevance label, the better is the match between query and document.

2 Linear Regression

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is an explanatory variable, and the other is a dependent variable. For example, a modeler might want to relate the weights of individuals to their heights using a linear regression model.

A linear regression line has an equation of the form $\boldsymbol{Y = a + bX}$, where $\boldsymbol{X}$ is the explanatory variable and $\boldsymbol{Y}$ is the dependent variable. The slope of the line is $\boldsymbol{b}$, and $\boldsymbol{a}$ is the intercept (the value of $\boldsymbol{y}$ when $\boldsymbol{x} = 0$).

Multivariate Linear Regression

Multivariate is like the basic/simple linear regression but with multiple independent variables contributing to the dependent variable and hence multiple coefficients to determine.

$$Y_i=\alpha+\beta_1 x_i^{(1)}+\beta_2 x_i^{(2)}+....+\beta_n x_i^{(n)}$$

Y_i is the estimate of i^{th} component of dependent variable y, where we have **n** independent variables and x_{ij} denotes the ith component of the j^{th} independent variable/feature.

3 Radial Basis Functions

An RBF is a real valued function whose values calculated according to the distance from the origin $\Phi(x) = \Phi(||x||)$ or alternatively from some other point called center $\Phi(x,c) = \Phi(||x-c||)$. Any function that satisfies the above equation is a radial basis function.
In linear regression, f(x) as a combination of radial basis functions, one per data point.

$$f(x) = \Sigma^{n}_{i=1} Wn\ h(||x - x_n||)$$

4 K-Means Clustering

K-Means starts by randomly defining *k* centroids. It works in iterative steps to perform following tasks:

1. Assign each data point to the closest corresponding centroid, using the standard Euclidean distance - the straight-line distance between the data point and the centroid.
2. For each centroid, calculate the mean of the values of all the points belonging to it. The mean value becomes the new value of the centroid.

Then, all the centroids have new values that correspond to the means of all their corresponding points. These new points are put through above steps producing yet another set of centroid values. This process is repeated over and over until there is no change in the centroid values, meaning that they have been accurately grouped. Or, the process can be stopped when a previously determined maximum number of steps has been met.

5 Regularization

Regularization is a form of regression, that constrains or shrinks the coefficient estimates β towards zero. This technique discourages learning a more complex to avoid the risk of overfitting.

$$Y_i=\alpha+\beta_1 x_i^{(1)}+\beta_2 x_i^{(2)}+....+\beta_n x_i^{(n)}$$

Where Y is the linear regression equation.
We, basically, tune the overall impact of the regularization term by multiplying its value by a scalar known as **lambda** (also called the **regularization rate**).

When choosing a lambda value,

- If lambda value is too high, the model will be simple, but risk of *underfitting* data is high. The model won't learn enough about the training data to make correct predictions.
- If lambda value is too low, the model will be more complex, and the risk of *overfitting* data is high. The model will learn too much about the particularities of the training data and won't be able to generalize to new data.

6 Closed Form Solution

Wolfram Math world states that, an equation is said to be a closed-form solution if it solves a given problem in terms of functions and mathematical operations from a given generally-accepted set.

Closed form solution in linear regression would be the least square equation

$$\beta\hat{}=(X^T X)^{-1} X^T y$$

7 Stochastic Gradient Descent

SGD is an optimization technique used to update the parameters of a model. The way this optimization algorithm works is that each training instance is shown to the model one at a time. The model makes a prediction for a training instance, the error is calculated, and the model is updated to reduce the error for the next prediction. This process is repeated for number of iterations.

This procedure can be used to find the set of coefficients in a model that result in the smallest error for the model on the training data. Each iteration, the coefficients k is updated using the equation:

K = K – learning rate*error*x

Where K is the coefficient or weight being optimized, learning rate is a learning rate that needs to be configured, error is the prediction error for the model on the training data attributed to the weight, and x is the input value. The learning rate used should not be too high or too low here.

8 Root Mean Square Error

The regression line predicts the average y value associated with a given x value. RMS is used to get a measure of the spread of the y values around that average.
To construct the RMS error, we determine the residuals.

Residuals are the difference between the actual values and the predicted values.
$Y_j - Y_i$ where Y_i is the observed value for the i^{th} observation and Y_j is the predicted value. They can be positive or negative as the predicted value under or over estimates the actual value. Squaring the residuals, averaging the squares, and taking the square root gives the RMS error.

$$RMSErrors = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$$

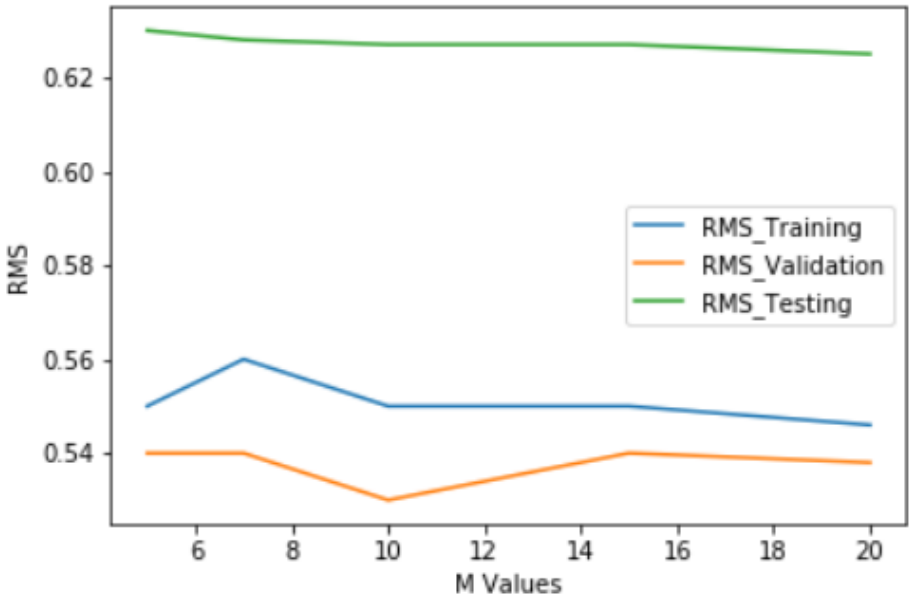
9 Model

We take up the problem of Learning to Rank to solve using to solve using Linear Regression using two approaches, first closed form solution and second stochastic gradient descent. We divide the raw data into training data(80%), validation data (10%), testing data (10%). For closed form solution, we generate the weight matrix using Moore-Penrose pseudo matrix inverse and generate rms for training, validation, and testing. In Gradient Descent, with learning rate we again test our model on rms for training, validation and testing.

9.1 Cases

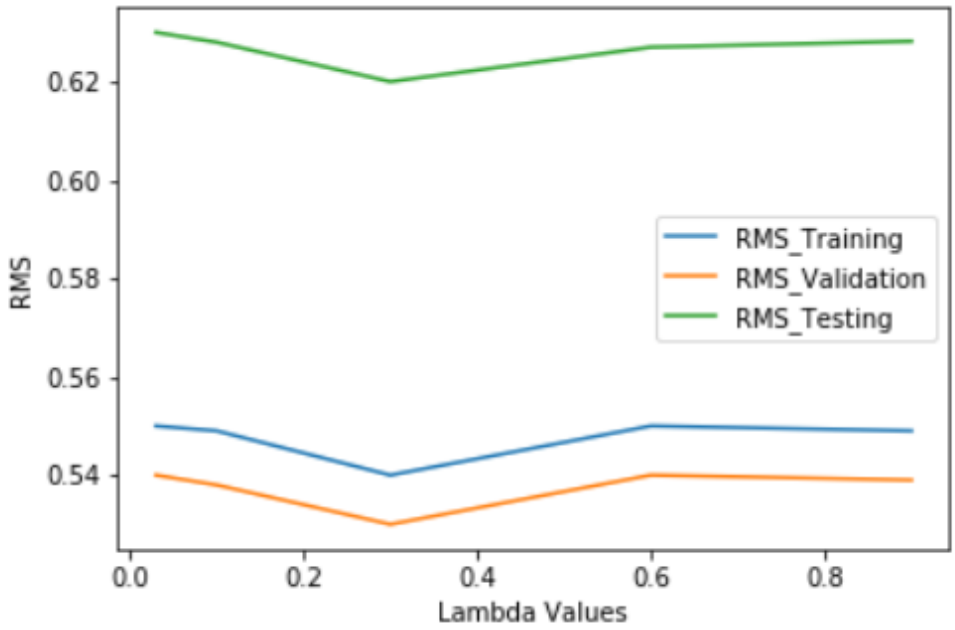
Changing M values

I tried changing M values for closed form solution.
M [5,7,10,15,20]
Below is the graph I observed for rms_validation, rms_testing, rms_training.



Changing Lambda values

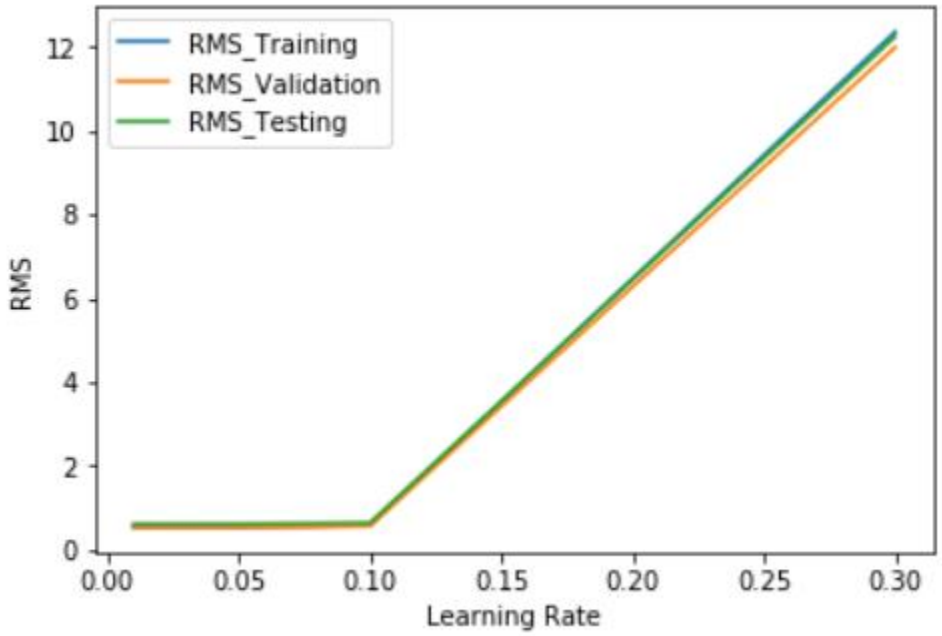
I tried changing Lambda values for closed form solution.
Lambda = [0.03,0.1,0.3,0.6,0.9]
Below is the graph I observed for rms_validation, rms_testing, rms_training.



Changing Learning Rate

I tried changing Learning rate for gradient descent.
Learning Rate = [0.01,0.05,0.08,0.1,0.3]

Below is the graph I observed for rms_validation, rms_testing, rms_training



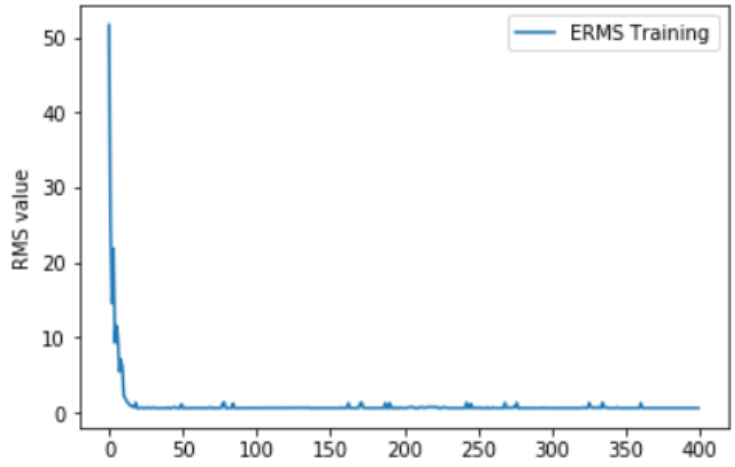
10 Conclusion

Based on the experiments performed on hyper-parameters, we can see the model best fits when M = 10, Lambda = 0.03, and Learning Rate = 0.01.

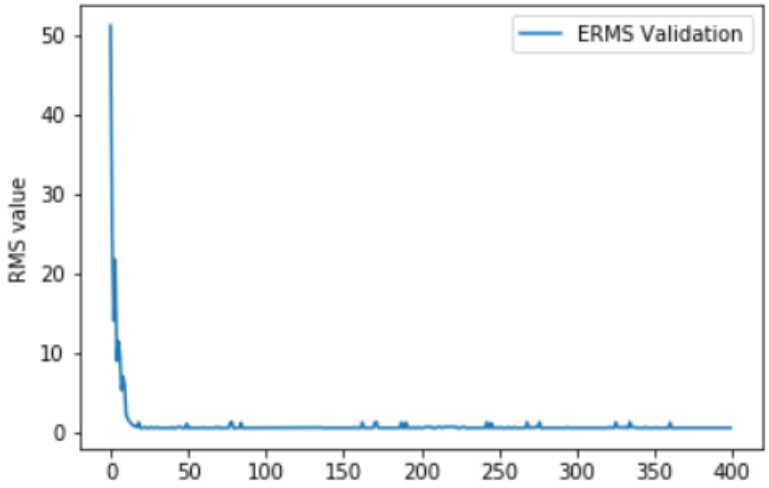
Output

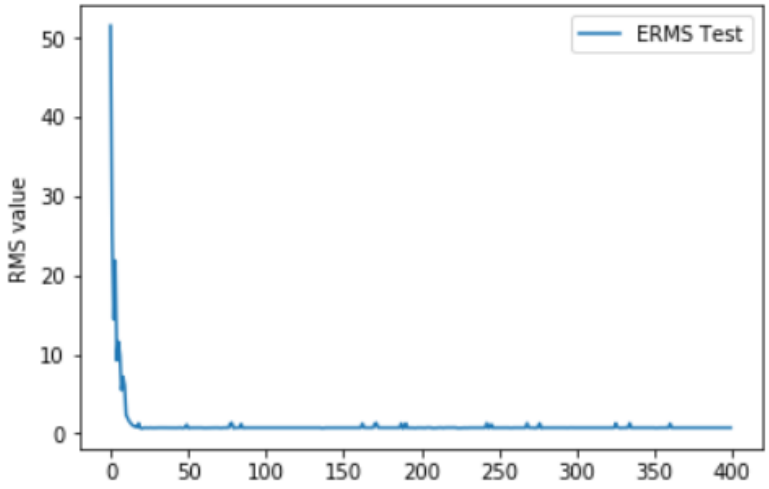
```
UBITname      = ddsolank
Person Number = 50290940
--------------------------------------------------
------------------LeToR Data----------------------
--------------------------------------------------
-------Closed Form with Radial Basis Function-------
--------------------------------------------------
E_rms Training   = 0.5494694067137847
E_rms Validation = 0.5384281741390526
E_rms Testing    = 0.6279788453847844
```

```
----------Gradient Descent Solution--------------------
E_rms Training   = 0.54964
E_rms Validation = 0.53846
E_rms Testing    = 0.62372
```


ERMS Training
RMS value
0
10
20
30
40
50
0
50
100
150
200
250
300
350
400

172
173


ERMS Validation
RMS value
0
10
20
30
40
50
0
50
100
150
200
250
300
350
400


ERMS Test
RMS value
0
10
20
30
40
50
0
50
100
150
200
250
300
350
400

174