PAGE RELEVANCE OF QUERY-URL PAIRS

ABHISHEK KOLUGURI

INTRODUCTION:

Over the years, many algorithms based on machine learning are being proposed to rank the search results in a search engine. Query-URL relevance (QUR) is an important criterion to measure the quality of commercial search engines. Many models have been proposed to study query-url relevance. In our project, we analyze the correlation between query-url by training a regression model and then predict the page relevancy labels for new queries.

Supervised learning is a type of machine learning algorithm that makes predictions using a known dataset (training dataset). The training dataset includes both the input data and response values. Then, the supervised learning algorithm makes predictions of response values for new dataset by building a model. This model is then validated using a test dataset. Supervised learning includes two categories of learning:

1. Classification models
2. Regression models

If the task of the prediction is to classify or separate the data into a set of finite classes or labels, then such a model is called classification model. On the other hand, if the goal is to predict continuous response values or target variables, then such a model is called regression model.
The system is trained using two models:

1. Closed form
2. Gradient Descent

A closed form is a mathematical expression which can be evaluated in finite number of standard operations. Closed means only one solution which implies only one function can establish a relationship between the outcome and predictors.

When a function with a set of parameters are given, gradient descent algorithm starts with an initial set of parameters and then iteratively move towards the set of parameter values that minimizes the function. The aim of gradient descent algorithm is to minimize the value of the function.

Learning to rank algorithms require large volume of training data. Larger training datasets often yield models with higher predictive power that can generalize well for new datasets. The large-scale dataset used in this project is Microsoft LETOR 4.0(LEarning TO Rank).Given data is a package of benchmark data sets for research on Learning To Rank. In the project, a date set of "MQ2007" which is "Querylevelnorm.txt" is used which contains 69,623 urls/samples.

The dataset consists of queries and urls, which are represented by ID. It also consists of feature vectors extracted from query-url pairs. In the data files, each row corresponds to a query-url pair. The first column is relevance label of the pair, the second column is query id, and the following columns are features. The larger value the relevance label has, the more relevant the query-url pair is.

The aim of the project is to train a regression model and predict the page relevancy labels for new queries. It is based on query - url pair datasets. In intricate manner, a set of urls are fed into a regression model which takes a query as input and outputs the relevancy labels (say 1,2,3 and so on) corresponding to the query. Higher the value indicates higher relevance. The large scale data set used is Microsoft LETOR 4.0. Training, validation and testing is done on three parts of the data set. An error is calculated using the predicted and actual target values. Purpose of the project is to reduce the error and attain the optimum solution. A set of parameters like model complexity, variance, mean and lambda are tweaked to attain the best results.

TRAINING:

- The system is trained using the closed form and gradient descent models and trained on 80% of data.
- An input data of 69,623 samples are given. Initially the given data is parsed and the rich data or the features of every sample are retrieved.
- The retrieved values are stored in a designed matrix of order 69,623 by 46. After attaining the values a function "phi" is developed. In the project Gaussian function is considered as the "phi" or basis function.

- The formula is given as

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

- Set of means are generated randomly using the rand function in matlab and a variance of 5 is taken into consideration of trying different values. The variance of 5 has the best fit for my model.
- The features of each sample are considered as a vector. So total 69,623 vectors are present.
- Now each vector is substituted into the above basis function to attain a matrix as below.

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

- A new matrix is constructed using the given data which is the target matrix. It is denoted as ‘t’ and has an order of 69,623 by 1.
- After the generation of the basis function, we need to find the weights using the formula as below.

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

- In the project a weight matrix is constructed of length 8 by 1 by implementing the above formula.
- This above implementation is the closed form model
- The second model is the gradient model, which is also a form of regression model.

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{\tau} + \eta(t_n - \mathbf{w}^{(\tau)T}\phi_n)\phi_n$$

- The above formula is implemented using the target values, basis function and constant.
- The constant is varied to attain the best result. The motto of the formula is to attain a set of weights that provide the optimal solution.

- The basis function used is the Gaussian basis function and ‘t’ is the target value.
- The values of ‘w’ are found recursively.

VALIDATION:

- The process of validation is done to avoid over fitting and done over 10% of data.
- Over fitting can be avoided by specifically managing or tweaking or finding the best fit model complexity, hyper parameters and regularization constants.
- Without validation, if the values are learned the model may lead to over fitting which means the process is done without regularization.
- So to avoid over fitting regularization is used, which can be done using the below formula.

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

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

- The value of q in the above equation is considered as 2. After implementing this regularization, we could avoid the over fitting.
- For the lowest RMS error, the system attains the model complexity which is found to be 8.
- The regularization constant lambda is found to be 16.

TESTING:

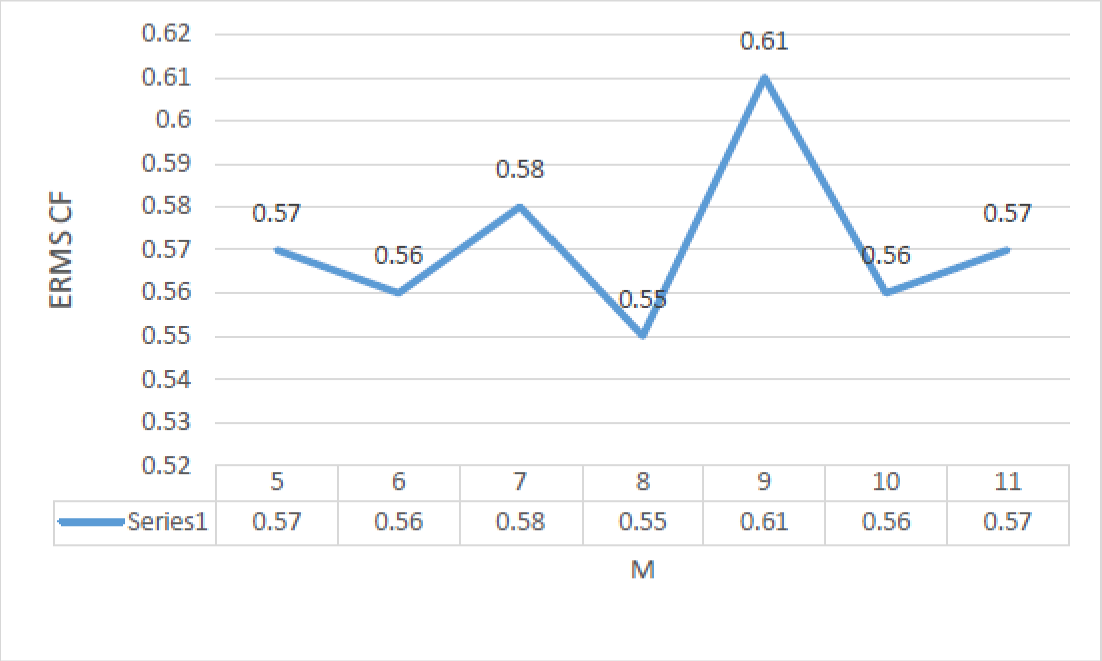
- Testing is the process of checking the correctness of the algorithm or model.
- The system tests the two regression models say closed form and gradient descent.
- Testing is done on 10% of the entire data.
- Testing in other terms can be used as the way of calculation the error. The system can use two types of error calculations but implemented only rms error using the below formula.

$$E_{RMS} = \sqrt{2E(\mathbf{w}*)/N_T}$$

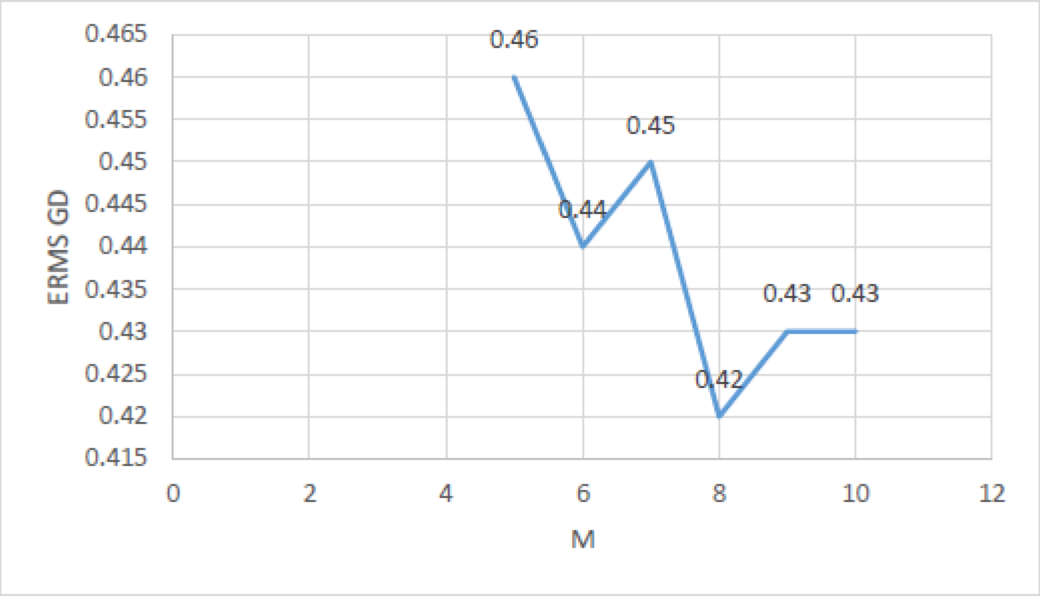
- The rms error for closed form was found to be 0.56.
- The rms error for gradient descent was found to be 0.6.

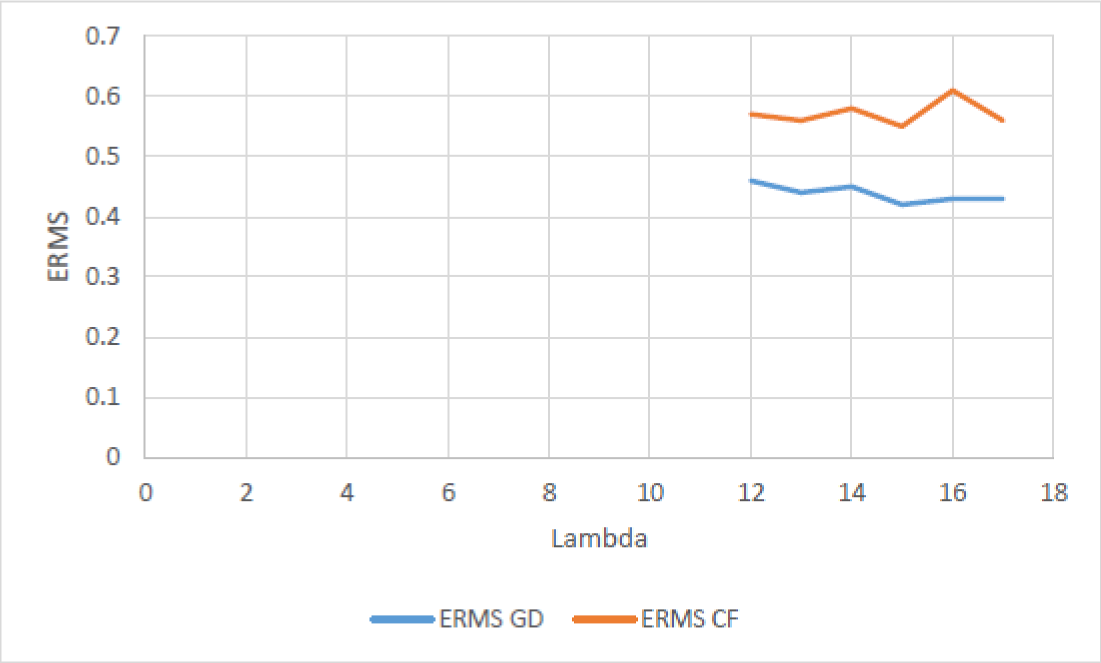
PARAMETERS AND RELATIONSHIPS:

- Various parameters used are mean, variance, regularization constant lambda, “q” in regularization equation, eta and model complexity.
- Variance is found to be 5. I tried for different values of variance but the taken value gave me the best results.
- Lambda is found to be 16.
- Eta is taken as 0.00001 and q is taken 2 as it gave the optimum results.
- Similarly, model complexity for closed form is 8.
- Model complexity for gradient descent is 8.
- Mean is calculated using the rand function in mat lab.



	5	6	7	8	9	10	11
Series1	0.57	0.56	0.58	0.55	0.61	0.56	0.57


0.465
0.46
0.455
0.45
0.445
0.44
0.435
0.43
0.425
0.42
0.415
ERMS GD
0.46
0.44
0.45
0.42
0.43
0.43
0
2
4
6
8
10
12
M


0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
ERMS
0
2
4
6
8
10
12
14
16
18
Lambda
ERMS GD
ERMS CF

EVALUATION:

The developed models are evaluated and the errors are found using the root mean square error.

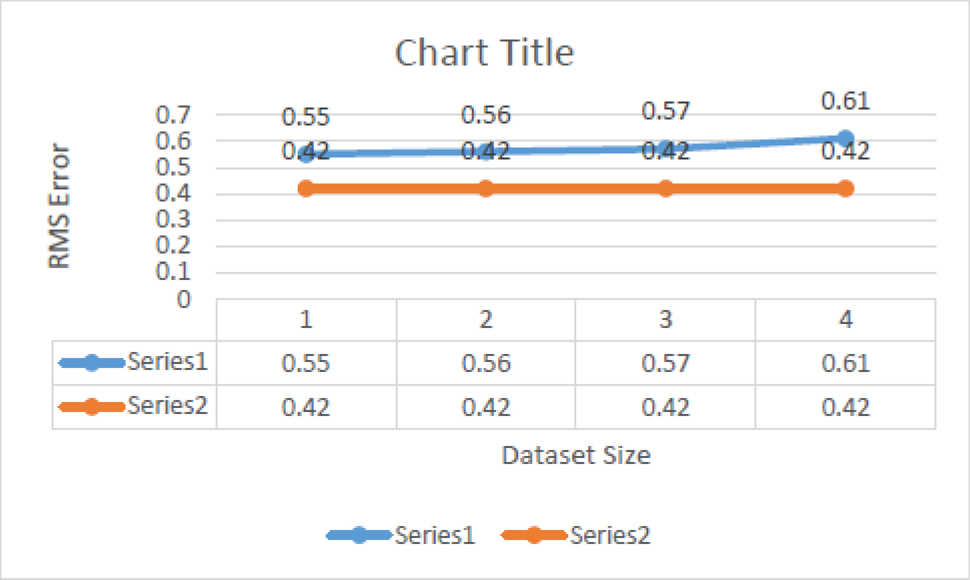
- RMS error for closed form is 0.55.
- RMS error for gradient descent is 0.42
- Formula used is

$$E_{RMS} = \sqrt{2E(\mathbf{w}*)/N_T}$$

- The models are calculated over the developed target values and the given target values.
- Weight matrices are 8 by 1.
- Basis function matrix are 69623 by 8.
- Given data matrix is 69623 by 46.
- Given target matrix is 69623 by 1.

COMPARISON

The system trained and tested on closed form and gradient descent models. The performance comparison using rms error is as follows.



	1	2	3	4
Series1	0.55	0.56	0.57	0.61
Series2	0.42	0.42	0.42	0.42

- Series 1 is closed form and series 2 is gradient descent performance.
- The gradient descent model has the least error for all the above cases.

- According to me gradient descent model performed better over the closed form.

When we use normal equation to compute the weight function, we need to calculate the inverse of the matrix. If the order of the matrix is low, one can take normal equations to be the better option for calculation. But if the order of the matrix increases then it takes long time to finish. To solve the problem of time complexity one can use Gradient Descent method as it is easy to implement and much faster.

CONCLUSION

The LETOR data set is used with two regression models and the performance is compared. A system which finds the relevancy in the web pages to a given query is developed. The hyper parameters are tweaked for the best result and errors are reduced to the maximum against the target values.

REFERENCES:

- **http://stats.stackexchange.com/questions/70848/what-does-a-closed-form-solution-mean**
- **http://research.microsoft.com/en-us/um/beijing/projects/letor//**
- **http://en.wikipedia.org/wiki/Gaussian_function**
- **http://www.holehouse.org/mlclass/01_02_Introduction_regression_analysis_and_gr.html**
- **http://statweb.stanford.edu/~susan/courses/s60/split/node60.html**
- **http://www.math.binghamton.edu/jbrennan/home/S13MAT148/Chapter11.pdf**
- **http://www.cs.berkeley.edu/~russell/classes/cs194/f11/lectures/CS194%20Fall%202011%20Lecture%2004.pdf**