Learning to Rank using Linear Regression

Parth Shah
SEAS, University at Buffalo
Buffalo, NY, 41214
Person Number: 50291125
parthnay@buffalo.edu

Abstract

The project aims to rank the search engine web pages based on the features of each web page, the method is known as Learning to Rank (LeToR). We train a linear regression model using a closed-form solution and stochastic gradient descent method.

1 Introduction

The given problem is to use Machine Learning to solve LEarning TO Rank (LETOR) problem in Information Retrieval. We use linear regression as we map the input vector x to real valued labels or real-valued scalar targets y(x,w). We use two methods to perform linear regression on the dataset provided.

1. Closed-Form Solution
2. Stochastic Gradient Descent (SCD)

2 DataSet and Data Splitting

The given DataSet is LeToR. LeToR is a package of benchmark data sets for research on LEarning TO Rank, which contains standard features, relevance judgments, data partitioning, evaluation tools, and several baselines. Version 1.0 was released in April 2007. This version, 4.0, was released in July 2009.

The DataSet thus provided was partitioned as follows. 80% for the data was used for training. 10% of the data was used for validating our model and the remaining 10% was used to test our model for the performance.

3 Linear Regression

The given problem was solved using Linear Regression. Linear Regression is solved by the equation of the form $y(x,w) = w^T\Phi(x)$. where w is the weight vector to be learnt from regression and Φ is a basis function which is used to convert input vector x into a scalar value. We have used the radial gaussian bias function to calculate Φ. It can be written as

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

The Gaussian radial function takes in the input of one row of x, the mean μ^j and Σ^{-1} the precision and returns a scalar output which is one entry into the design matrix. The design matrix is computed for all rows and we get NxM dimension design matrix as shown in the project description.We used regularization to avoid overfitting and obtain better results. This can be handled by the Lambda Value.

We deploy 2 form of models to solve the problem, closed-form solution and stochastic gradient descent

4 Closed-Form Solution

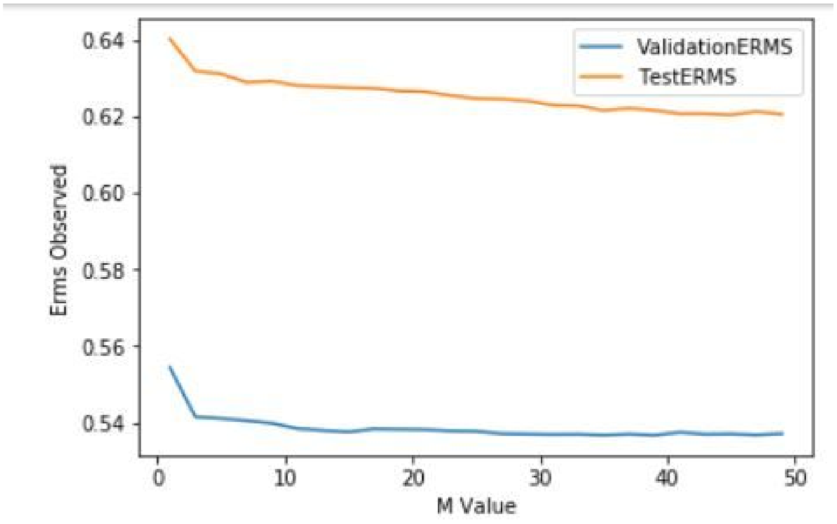
The closed form is one of the easiest types of regressions. We can control the solution by varying values of M and Lambda. Here, I have varied the value of M from 1 to 50 with a step of 2 and the value of lambda from 0.1 to 1.0 with a step of 0.04.

The calculation of weights using the closed form solution has the form $\mathbf{w}_{\mathbf{ML}} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\mathbf{t}$

The Gaussian radial function takes in the input of one row of x, the mean μj and Σ-1 the precision and returns a scalar output which is one entry into the design matrix. The design matrix is computed for all rows and we get NxM dimension design matrix. Where N is the number of observations.

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

A graph is plotted for the comparison of ValidationERMS and TestERMS obtained by the various M and Lambda variations. It can be seen below



```
Best TrainingERMS =  0.5410855742370273
Best ValidationERMS =  0.5367186897442348
Best TestERMS =  0.6203480316948987
```

As we can see the value of ERMS for validation and test varies very less with change in M and Lambda, we see significant decrease in the ERMS in initial changes of M from 1 to 10 but after that the ERMS decrease is menial compared to the computation time increase.
Thus we get the best values for Validation and Testing ERMS at M = 35 and Lambda = 0
They are
Best TrainingERMS = 0.5410855742370273
Best ValidationERMS = 0.5367186897442348
Best TestERMS = 0.620348031694898

5 Stochastic Gradient Descent Solution

Gradient descent is a first-order iterative optimization algorithm for finding the minimum of a function. To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient (or approximate gradient) of the function at the current point. If instead one takes steps proportional to the positive of the gradient, one approaches a local maximum of that function; the procedure is then known as gradient ascent.[2]

The stochastic gradient descent algorithm first takes a random initial value w(0). Then it updates the value of w_0 using

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

Where τ denotes the number of iterations and Δw denotes the weight update. Where,

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

is the learning rate and the update is given by

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

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

The hyperparameters varied for SCD is the learning rate and the number of iterations.
We see that the change in learning rate affects the ERMS but compared to the increase in computation complexity and time it is not significant.
Thus we fix the learning rate to 0.01 and 400 iterations and obtain the following results

```
E_rms Training   = 0.54123
E_rms Validation = 0.53774
E_rms Testing    = 0.61673
```

4 Conclusion

We used LeToR Dataset to perform gradient Descent using two methods viz- Closed-Form Solution and Stochastic Gradient Descent. The dataset was divided into 80-10-10 and used for Training-Validation-Test respectively. The results obtained are

For Closed Form Solution varying the complexity M (bias Functions) and the Regularization parameter Lambda does not reflect in much change on the value of ERMS compared to the computation complexity that it adds.Thus fixing on a median value of M and a small Lambda suffices for best solution.

For Stochastic Gradient Descent Solution the same anomaly was observed with the learning rate and the value of ERMS.
The performance of closed form solution is better than the stochastic gradient descent.

However, since closed form solution is slower than the Stochastic gradient descent, the latter is preferred in cases when the data set is very huge. Also training error for both the models decreases with increasing M. The validation error on the other hand first decreases and then sharply rises with increasing M. We also consider the computation time while considering the value of M. This is because selecting a big M can reduce the error but can be painfully slow. So in cases like that we prefer a smaller M. We have chosen a smaller M in this project to reduce the computation time.

Acknowledgments

This report was prepared in accordance to the Project1.2.pdf provided, LeToR.pdf and Linear Regression documentation available on Wikipedia

References

[1] Pattern Recognition and Machine Learning: Christopher Bishop

[2] Gradient descent: https://en.wikipedia.org/wiki/Gradient_descent

[3]For data: http://research.microsoft.com/en-us/um/beijing/projects/letor/letor4dataset.aspx