Project 1.2: Learning to Rank(LeTor) using Machine Learning

Srivenkata Krishnan Sesharamanujam
CSE574 Introduction to Machine Learning
UB Person no: 50288730
srivenka@buffalo.edu

1 Introduction

The following report is for the Learning to Rank (LeTor) problem solved using machine learning approach. Here, the problem is formulated as a linear regression model and idea is to train the model in two ways: One, closed-form solution; Two: Stochastic gradient descent (SGD). The objective of the project is to implement the machine learning algorithms using these two ways and compare the respective outputs by changing the hyper parameters and other factors. The rest of the report is organized as follows: Section 2 outlines the theoretical information needed to understand the concepts involved in implementation of this project while section 3 focuses on how the linear regression model is implemented on a data set. Section 4 analyzes the performance of implemented models along various parameters and finally, conclude based on the results in the final section.

2 Understanding the Concepts involved in the problem

2.1 What is Learning to Rank (LeTor) problem ?

Learning to Rank solves a ranking problem on a list of items. The aim of LeTor is to come up with optimal ordering of those items. For example, if you run an e-commerce website, a classical problem is to rank your product offering in the search page in a way that maximizes the probability of your items being sold. This is an ideal example of LeTor problem. LeTor is useful in many applications in information retrieval, Natural language processing and Data mining.

The goal of the project is to implement LeTor using machine learning techniques. We use linear regression model using two solutions to the problem. In the coming sections, we will understand what is linear regression and its types.

2.2 Linear Regression Model

2.2.1 Regression

Regression is a method of modelling a target value based on the independent predictors. The method is mostly used for forecasting and finding out cause and effect relationship between variables. Regression techniques mostly differ based on the no of independent variables and type of relationship between independent and dependant variables.

2.2.2 Linear Regression

Linear Regression is a a type of regression analysis where the number of independent variable is one and there is a linear relationship between the independent and dependent variable.

Linear Regression is a technique where a straight line is used to model a relationship between input and output values. In more than two dimensions, the straight line may be thought of as plane or hyper plane.

As given in the project description, the linear regression function is given as,

$$y(x, w) = w^T \phi(x) \tag{1}$$

where,

$w = w_0, w_1, ..., w_{M-1}$ is the weight vector to be learnt from training samples

$\phi = ((\phi_0, \phi_1, ..., \phi_{M-1})^T)$ is a vector of M basis functions.

$\phi(x)$ is the radial basis function. In our problem, we are using the gaussian radial basis function

2.2.3 Radial Basis Function

A radial basis function is a real valued function ϕ of whose value depends only on the distance from the origin, so that $\phi(x)$ = $\phi(||x||)$ or alternatively some other point c, called a center, so that $\phi(x, c) = \phi(|x - c|)$. Any function that satisfies the property $\phi(x)$ = $\phi(||x||)$ is a radial function.

Radial functions are special class of function . Their characteristic feature is that their response decreases or increases monotonically with distance from the central point.

Each basis function converts input vector x into a scalar value. The type of radial function we are using is the Gaussian.

2.2.4 Gaussian Radial Basis Function

Gaussian radial basis function is a well known bell curve. The bell curve can be narrower or wider. The width is determined by σ. The larger the σ, narrower the bell shape and other way around.

A gaussian radial function gives a significant response only in the neighbourhood near the center. And it monotonically decreases with distance from the center.

As given in the project description, a gaussian radial function is defined.

Now, we have the understanding of the basis function and now lets go into its application in the closed form and stochastic gradient descent linear regression models.

2.2.5 Linear Regression using Closed Form Solution

As given in the project description, closed form solution is given as,

$$W_{ML} = (\phi^T \phi)^{-1} \phi^T t \tag{2}$$

where,

$t - - - - - > t_1, t_2, ...$ vector of outputs in training data

$\phi - - >$ Design matrix

In the closed form solution, we will minimize j by explicitly taking its derivatives with respect to ϕ_j and setting them to zero. This allows us to find optimum without iteration.

We also have to see that $(\phi^T \phi)^{-1}$ is invertible. The cases where it may be non- invertible are,

- Redundant features, where two features are closely related (aka linearly dependant)
- Too many features.

The ideal scenario where closed form solution is preferred is for smaller data sets. It is a $O(n^3)$ algorithm. So, if the data sets is large, then computation requires more time.

For larger data sets, where the inverse of $\phi\phi^T$ may not exist, we go for Gradient descent or stochastic gradient descent.

2.2.6 Linear Regression using Stochastic Gradient Descent

The most common optimization algorithm used in machine learning is the stochastic gradient descent. Gradient descent is the process of minimizing a function by following the gradients of a cost function.

The way this optimization algorithm works is that each training instance is shown to model one at a time. The model makes a prediction for a training instance, the error is calculated and the model is updated in order to reduce the error for the next prediction. This process is repeated for a fixed no of iterations.

In Stochastic Gradient descent, we use one example or one training sample at each iteration instead of using whole data set to sum all for every steps.

As given in the project description, the stochastic gradient descent solution,

$$w^{T+1} = w^T + \triangle w^T \tag{3}$$

where,

$\triangle w^T = -\eta^T \nabla E.$

η^T is the learning rate.

$\nabla E = \nabla E + \lambda \nabla E_w$

Stochastic Gradient Descent is widely used for larger data sets. It is computationally faster and can be trained parallely. The order of algorithm is $O(kn^2)$

2.2.7 K-Means Clustering

We have the data set given, we have to transform the data in some useful form, so that we can train data. To achieve this , we use K-Means algorithm.

K-means clustering is a type of unsupervised learning, which is used when you have unlabeled data. The goal of the algorithm is to find groups in the data.

The output of the K-means clustering algorithm are,

- The centroids of K clusters, which can be used to label data.
- Labels for the training data.

Examining the centroid feature weights can be used to qualitatively interpret what kind of group each cluster represents.

The steps of the algorithm are,

- The algorithm inputs are the no of clusters K and the data set. The data set is a collection of features for each data point.
- The algorithm start with initial estimates for the K centroids, which can either be ram=randomly generated or randomly selected from the data set.
- The algorithm then iterates between two steps.
  - Data Assignment step: Each centroid defines one of the clusters. In this step, each data point is assigned to its nearest centroid, based on the squared euclidean distance.
  - Centroid update step: In this step, centroids are re computed. This is done by taking mean of all data points assigned to the centroid's cluster.
- The algorithm iterates between steps the above steps until no data points change clusters(If sum of the distances is minimized or a specific no of iterations is reached).

3 Implementation

The Learning to Rank (LeTor) solution is implemented in this step by step process.

- Extract feature values and data from data set.
- Apply K-means clustering algorithm.
- Apply Linear regression in closed form.
- Check for the accuracy of the model using root mean squares.
- Apply linear regression in Stochastic Gradient descent form.
- Check for the accuracy of the SGD model using root mean squares.

3.1 Extract Feature values and data from data set

As given in the project description, we have to split the given data set into three sets. One, Training set (80%). Two, validation set (10%) and finally testing set (10%).

3.2 Apply K-means clustering algorithm

After the splitting of data, we apply the k-means clustering algorithm and find groups in the data.

3.3 Apply Linear Regression in closed Form

From the equation 2, we apply the data to the equation and obtain the relationship between input and output variables.

3.4 Check for the accuracy of the model using root mean squares

Applying the root mean squares , we get the accuracy of the model on the given data.

3.5 Apply linear regression in Stochastic Gradient descent form

From the equation 3, we apply the data to the equation and obtain the output values.

3.6 Check for the accuracy of the SGD model using root mean squares

Applying the root mean squares , we get the accuracy of the model on the given data.

3.7 Tuning the parameters

Having applied the model to the given data set. Now we have a lot of factors to be considered to get a better model. Two models are different in their form and each model depends on a lot of parameters for its performance.

When you look at the SGD model equation, we can say that it depends on the learning rate.

By considering the parameters the equations depend upon, the following section will consolidate all the inferences based on the parameters.

4 Inference

Every estimator has its advantages and drawbacks. Its generalization error can be decomposed in terms of bias, variance and noise. The bias of an estimator is its average error for different training sets. The variance of an estimator indicates how sensitive it is to varying training sets. Noise is a property of the data.

Bias and variance are inherent properties of estimators and we usually have to select learning algorithms and hyper parameters so that both bias and variance are as low as possible . Another way to

reduce the variance of a model is to use more training data. However, you should only collect more training data if the true function is too complex to be approximated by an estimator with a lower variance.

In the simple one-dimensional problem it is easy to see whether the estimator suffers from bias or variance. However, in high-dimensional spaces, models can become very difficult to visualize. For this reason, it is often helpful to use the curves described below.

- Validation curve
- Learning curve

4.1 Validation Curve

To validate a model we need a function, for example accuracy for classifiers. The proper way of choosing multiple hyperparameters of an estimator are of course grid search or similar methods that select the hyperparameter with the maximum score on a validation set or multiple validation sets.

However, it is sometimes helpful to plot the influence of a single hyperparameter on the training score and the validation score to find out whether the estimator is overfitting or under fitting for some hyperparameter values.

If the training score and the validation score are both low, the estimator will be underfitting. If the training score is high and the validation score is low, the estimator is overfitting and otherwise it is working very well. A low training score and a high validation score is usually not possible.

4.2 Learning Curve

Learning curves constitute a great tool to diagnose bias and variance in any supervised learning algorithm. A learning curve shows the validation and training score of an estimator for varying numbers of training samples. It is a tool to find out how much we benefit from adding more training data and whether the estimator suffers more from a variance error or a bias error.

If both the validation score and the training score converge to a value that is too low with increasing size of the training set, we will not benefit much from more training data.

If the training score is much greater than the validation score for the maximum number of training samples, adding more training samples will most likely increase generalization.

Types of Learning Curves

- Bad Learning Curve: High Bias
- Bad Learning Curve: High Variance
- Ideal Learning curve

4.3 Learning curve for Closed form solution

Figure 1: Learning Rate of CLOSED FORM SOLUTION