Survey of Boosting Algorithms for Big Data Applications

Anju¹, Nonita Sharma²
Department of Computer Science & Engineering
National Institute of Technology Delhi
New Delhi, India¹, ²

Abstract - This manuscript compares the state-of-the-art boosting algorithms for Big Data Analytics. In boosting technique, a number of weak learners are combined and give a strong learner with higher accuracy. Boosting is mainly used in handling missing values and avoiding the problem of overfitting. This research work compares XGBoost, Random forest and AdaBoost algorithms for accuracy. XGBoost is a scalable boosting algorithm. It gives importance of variable. It solves many real world problems with minimum amount of resources. The comparative analysis reveals that XGBoost demonstrates the best accuracy and hence is best suited for big data applications where computations are done in parallel.

Keywords—XGBoost, AdaBoost, Random Forest, Big Data, Boosting.

I. INTRODUCTION
In boosting system, various weak learners are consolidated and give a strong learner with higher precision. Weak learners are those predictors that give more accuracy than random guessing. However, strong learners are those classifiers that give maximum accuracy and hence coined as the base of machine learning. It has wide application area and applies on many classification techniques viz. feature selection, feature extraction, and multi-class categorization. Some of the most widely used applications of boosting are medical area, text classification, academic, and commercial etc.

Further, Boosting technique is a type of ensemble method, which is used when there is a collection of many weighted same or different type of predictors. However in this technique, a collection of several hypothesis is selected and eventually their prediction is combined. For example, if 50 decision trees are generated over same or different training data set then a new test dataset is created and voted for best classification.

II. SURVEY OF DIFFERENT TECHNIQUES
A. XGBoost
XGBoost stands for extreme gradient boosting, developed by Tianqi Chen[3]. It is an implementation over the gradient boosting. XGBoost is greedy in nature so it follows greedy approach. It has high performance and speed. Additionally, it has following advantages.

• Missing Values: XGBoost has built-in function that handles missing values.

• Speed: Due to parallel processing process it has faster performance than gradient boosting.

B. AdaBoost
Adaptive learning is shortly abbreviated as AdaBoost. It is most commonly used machine learning algorithm. Freund and Schapire[5] gave this algorithm. They also won gold prize for this in 2003. In it, base learner is chosen and improved it iteratively for the misclassified data. In short AdaBoost is,

• Assign equal weight to all training data.

• A base algorithm is chosen.

• At each step, increase the weight of misclassified data.

• Iterate it n times.

• Final model is made by weighted sum of n learners.

C. Random Forest
It is a machine learning algorithm and it is used in classification, regression and many more also. At training time, multiple decision trees are created and the output is the mean or average prediction of each tree. The algorithm is proposed by Tin Kam Ho [7].Random forest follows following steps:

• Using the bagging process sampling of training dataset takes place. It gives a no of trees.

• Nodes are split according to some splitting criteria.

• Due to splitting criteria, data is divided into each node.

• Classification takes place on leaf node.

• After trained for trees, test data is sampled. Each sample is given to all tress.

• At the leaf node classification is taking place.

• At last, the class of the test dataset is decides by majority voting or average process.

III. COMPARITIVE STUDY
In the given table a detailed comparative study is made amongst the three selected algorithms:
<table>
<thead>
<tr>
<th>Paper Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xgboost:eXtreme Gradient Boosting</td>
<td>XGBoost is an extension of gradient boosting. It consists linear model. It also contains tree learning method. It is faster than other because of parallel computation. Regression, classification, ranking and various objective functions are support by XGBoost. In this users can easily define their own objectives.</td>
</tr>
<tr>
<td>XGBoost: A Scalable Tree Boosting System</td>
<td>Sparsity-aware algorithm is works on sparse data. Approximate tree learning works on weighted quantile sketch. Cache access patterns, sharding and data compression are given to make XGBoost scalable. Regularized learning objective is also given for completeness.</td>
</tr>
<tr>
<td>XGBoost: Reliable Large-scale Tree Boosting System</td>
<td>XGBoost is fast parallel tree. Reason of its designed is fault tolerant of the distributed setting. XGBoost handles millions of sample on a single node.</td>
</tr>
<tr>
<td>ada: An R Package for Stochastic Boosting</td>
<td>In Stochastic gradient boosting a refined dataset is used in every iteration. It shows an increase in performance and speed in each step. ada implements three types of boosting. Plots are extent of the multi-class case. Data analytics used plots.</td>
</tr>
<tr>
<td>adabag: An R Package for Classification with Boosting and Bagging</td>
<td>Adabag is an implementation on AdaBoost. When the classifier trained then prediction of new data is possible. Cross validation estimation of the error also was done. margins() function is determine the margins of classifiers. Higher flexibility is acquired by rpart.control(). Rpart also used to handle missing values. errorevol() is used to shows the error of the ensembles. In the predict.bagging() and predict.boosting() , 'newmfinal' is used to pruned ensembles. In Version 3.1 variable of each tree give the gain of gini index and the weights of the trees. In this there are three new plots i.e. importanceplot(), plot.errorevol() and plot.margins(). Prediction on unlabeled data is also available in Version 4.1.</td>
</tr>
<tr>
<td>Random Forests</td>
<td>Overfitting problem is controlled by the law of large numbers. Its accuracy depends on the robustness of each classifier. Random selection and Random linear combinations is used for inputs. The results are compared with Adaboost.</td>
</tr>
</tbody>
</table>
IV. COMPARISON ON DIFFERENT PARAMETER

<table>
<thead>
<tr>
<th>Paper Title</th>
<th>Performance</th>
<th>Cross Validation of Error</th>
<th>Training and Testing Algorithm</th>
<th>Accuracy</th>
<th>Bias and Variance</th>
<th>Over fitting Problem Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>XGBoost: eXtreme Gradient Boosting</td>
<td>Faster than AdaBoost and Random Forest</td>
<td>Implemented</td>
<td>Any algorithm can be used</td>
<td>Maximum</td>
<td>high bias, low variance</td>
<td>Yes</td>
</tr>
<tr>
<td>XGBoost: A Scalable Tree Boosting System</td>
<td>Faster than AdaBoost and Random Forest</td>
<td>Implemented</td>
<td>Any algorithm can be used</td>
<td>Maximum</td>
<td>high bias, low variance</td>
<td>Yes</td>
</tr>
<tr>
<td>XGBoost: Reliable Large-scale Tree Boosting System</td>
<td>Faster than AdaBoost and Random Forest</td>
<td>Implemented</td>
<td>Any algorithm can be used</td>
<td>Maximum</td>
<td>high bias, low variance</td>
<td>Yes</td>
</tr>
<tr>
<td>ada: An R Package for Stochastic Boosting</td>
<td>Slower than Random Forest</td>
<td>Implemented</td>
<td>Any algorithm can be used</td>
<td>Lower than XGBoost</td>
<td>high bias, low variance</td>
<td>Less prone to Over fitting problem</td>
</tr>
<tr>
<td>adabag: An R Package for Classification with Boosting and Bagging</td>
<td>Slower than Random Forest</td>
<td>Implemented</td>
<td>Any algorithm can be used</td>
<td>Lower than XGBoost</td>
<td>high bias, low variance</td>
<td>Less prone to Overfitting problem</td>
</tr>
<tr>
<td>Random Forests</td>
<td>Faster than AdaBoost and slower than XGBoost</td>
<td>No need</td>
<td>Bootstrapping</td>
<td>Lowest among two</td>
<td>low bias, high variance</td>
<td>Avoid Overfitting</td>
</tr>
</tbody>
</table>

V. CONCLUSION

In this survey paper, we saw that boosting algorithm is very vast in itself and also it has many interpretations. AdaBoost is better than a random imagination and also we saw that XGBoost has a fast performance due to parallel computation while other boosting algorithm works on serial computations. Missing values is handled in these algorithms. Over fitting problem also be overcome by these algorithms.

REFERENCE