Abhinav Sharma, 2021, 9:2
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

Performance Enhancing Algorithms in Machine Learning

Abhinav Sharma, **Jayant Dhingra**
abhi199854@gmail.com, , jayant.dhingra6@gmail.com
Guru Tegh Bahadur Institute of Technology, jayant.dhingra6@gmail.com

Abstract- Machine learning is predominantly an area of Artificial Intelligence which has been a key component of providing solutions for the modern problem of digital arena. In this paper we present some of the unorthodox yet robust techniques in Machine Learning which assists in improving the efficacy of various machine learning models. Our use of these methods aims at classification techniques to increase their performance. Classification algorithms are very hefty in the terminology of performance but there are some methods that can help us to further make classification algorithms more efficient.

Keywords: - Machine Learning, Boosting, Bagging, Performance Enhancers.

I. INTRODUCTION

The appropriate way to start this paper is to discuss in brief about fundamentals of Machine Learning. Machine Learning is basically an application of artificial intelligence (AI) which imparts systems the capacity to instinctively learn from experience without being directly programmed and its performance will be measured by the count of correct predictions and detections.

The machine takes decisions and does predictions/ forecasting based on the data. Taking the example of a program that learns to detect/ forecast cancer from the medical reports of a patient.

This model will improve its performance as it gathers more experience by analyzing medical reports of wider population of patients.

Machine Learning is helpful in wide variety of field namely: traffic prediction, pattern recognition, natural language processing, computer games, robotics, medical diagnosis, and online fraud prediction, search engine result refining, E-mail spam filtering and so on.

The performance of any model depends on many aspects, for example on the selection of suitable algorithm for a given task, on the selection of a training set and so on but the performance of the model can further be improved using some algorithms like Bagging, Boosting, and Gradient Descent.

II. BAGGING ALGORITHM

Bagging is a procedure to ameliorate results of classification algorithms. The method of bagging was introduced by Leo Breiman and it was named after the phrase bootstrap aggregating. [1]

Leo Breiman prompted bagging as a technique of reducing the variance for a given base procedure, such as decision trees. Inevitably, the bagging technique has fascinated a lot of people in the machine learning community, most probably because of its lucid implementation and the popularity of the bootstrapping.

A chain of classifiers Hm, m=1, 2, 3..., M is created by the bagging algorithm in order to alter the training set. This chain of classifiers is then concatenated into a compound classifier. These compound classifiers are assigned weights according to:

$$H(d_i, c_j) = sign\left(\sum_{m=1}^{M} \alpha_m H_m(d_i, c_j)\right)$$ [3]

At the time of its invention, only heuristic arguments were presented why bagging would work. Subsequently, it has been shown that the method of bagging is a leveling operation which is hugely beneficial when trying to enhance the predictive performance of regression or classification trees. [2]

For the bagging algorithm to work we first do the initialization of the training set D for m=1,2,3 ..., M, then upon the Creation of a new set Dm of the same size D by random selection of training examples from the set D (some of examples can be selected repeatedly and some may not be selected at all).

After that learning of a particular classifier Hm: Dm →R by a given machine learning algorithm based on the actual training set Dm following which the compound classifier H is created as the aggregation of particular classifier Hm: m=1, ..., M and an example di is classified to the class cj in accordance with the number of votes obtained from particular classifiers Hm according to the formula [1].

III. BOOSTING

In Machine Learning, Boosting refers to a breed of algorithm which converts weak learners to strong learners due to which the accuracy of the model is increased.

This method of boosting works by generating new weak learners which are then sequentially combined and their predictions improve the overall performance of the model. If there are any incorrect predictions, then they are assigned to the misclassified samples and lower weights are assigned to the samples which are correctly classified.

In the final ensemble model the weak learners which perform better have higher weights. The boosting method only rectifies the next predictor by learning from mistakes. Some helpful boosting techniques are discussed below:

1. Gradient Boosting:
Gradient boosting is a particular case of boosting algorithm where errors are cut down by a gradient descent algorithm and construct a model in the form of infirm prediction models like decision trees. The key difference between the regular boosting and gradient boosting is in assigning of the weights when they encounter a wrong prediction. Gradient boosting repetitively enhances the loss of the model by updating weights.

Gradient boosting uses Additive Modeling in which a new decision tree is added one at a time to a model that minimizes the loss using gradient descent. Existing trees in the model remain untouched and thus slow down the rate of over fitting. The output of the new tree is combined with the output of existing trees until the loss is minimized below a threshold or specified limit of trees is reached.

2. XG Boosting:
Extreme Gradient Boosting (XG Boost) is an ascendable and improved version of the gradient boosting algorithm designed for efficacy, computational speed and model performance. It is an open-source library and a part of the Distributed Machine Learning Community. XG Boost is a perfect combination of software and hardware capabilities designed to enhance existing boosting techniques with accuracy in the shortest amount of time.

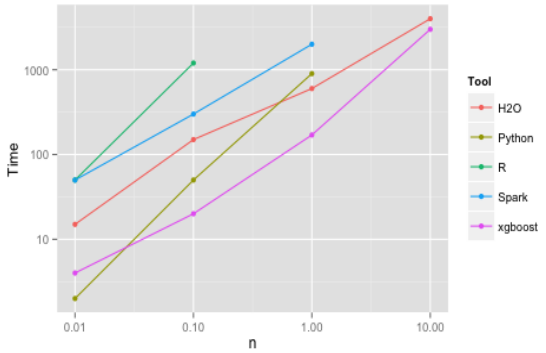

Fig.1. Benchmark performance of XG Boost. [3]

XG Boost has a lot of advantages over other boosting techniques and is a robust implementation of Gradient Boosting.

It is a robust model because of various advantages, some of which are stated below:

2.1 Tree Pruning: Pruning is a machine learning technique to reduce the size of regression trees by replacing nodes that don't contribute to improving classification on leaves. XG Boost creates nodes (also called splits) up to max depth specified and starts pruning from backward until the loss is below a threshold. Consider a split that has -3 loss and the subsequent node has +7 loss, XGBoost will not remove the split just by looking at one of the

International Journal of Science, Engineering and Technology

An Open Access Journal

negative loss. It will compute the total loss (-3 + 7 = +4) and if it turns out to be positive it keeps both.

2.2 Built-in Cross-validation: Cross validation is a statistical method to evaluate machine learning models on unseen data. It comes in handy when the dataset is limited and prevents over fitting by not taking an independent sample (holdout) from training data for validation. By reducing the size of training data, we are compromising with the features and patterns hidden in the data which can further induce errors in our model. XG Boost uses built-in cross validation function CV ().

3. ADA Boosting:

The ADA Boost algorithm, introduced in 1995 by Freund and Schapire [5], solved many of the practical difficulties of the earlier boosting algorithms. It focuses on classification problems and aims to convert a set of weak classifiers into a strong one.

The final equation for classification can be represented as:

$$F(x) = sign(\sum_{m=1}^{M} \theta_m f_m(x)),$$

[4]

Where; f_m stands for the m^{th} weak classifier and θ_m is the corresponding weight. It is exactly the weighted combination of M weak classifiers. It is worth noting that Breiman noticed the connection between random forests and ADA Boost as well, although his notion of a random forest was more general, including other types of large ensembles of randomly grown trees (Breiman, 2001)[6].

For the ADA Boosting to work we first do the initialize the observation weights as wi=1/N, i= 1,2,3.....,N. after which we fit a classifier Gm(x) to the training data using weights wi for m=1 to M after which we compute the errm by:

$$\frac{\sum_{i=1}^{N} w_i I\left(y_i \neq G_t\left(\mathbf{x}_i\right)\right)}{\sum_{i=1}^{N} w_i}.$$

[5]

after which we compute am by :

$$a_m = \log\left(\frac{1-\mathrm{err}_t}{\mathrm{err}_t}\right).$$

[6]

Following which the weight is adjusted as:

wi ← wi • exp (at • I (yi ≠ Gt (xi))). Subsequently set fi (x) = $\sum_{m=1}^{M}$ amGm(x) and finally Output F(x) = sign (fM(x)).

IV. GRADIENT DESCENT

Gradient Descent is an optimization method in which the objective is to reduce a cost function and increase the accuracy of the model. A cost function is a measure of how wrong the model is in terms of its ability to estimate the relationship between X and Y. The cost function can be estimated by iteratively running the Gradient Descent to compare estimated prediction against the known values of Y.

Gradient Descent works iteratively according to the learning rate. There are three primary types of gradient descent used in modern Machine Learning and in deep learning algorithms: "Stochastic Gradient Descent" (SGD), "Batch Gradient Descent" (BGD) and "Mini Batch Gradient" (MBGD). The main reason for these variations is computational efficiency.

BGD is considered to be computationally efficient and produces a stable error gradient and a stable convergence. It calculates the error for each example within the training set. After it evaluates all training examples, it updates the model parameters. However, the algorithm has the disadvantage is that the stable error gradient can sometimes result in a state of convergence that is not the best the model can achieve.

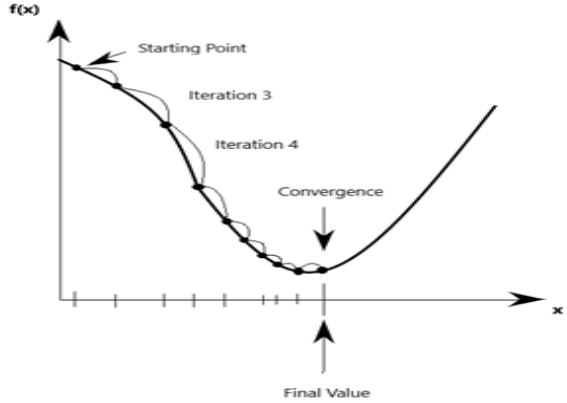

Fig 2. Gradient Descent graph plot.

In SGD, it updates the parameters according to the gradient of the error with respect to a single training

example. This can make SGD faster than BGD depending on the problem. However, it's disadvantage is that the frequency updates are more computationally expensive than BGD.

MBGD is the combination of SGD and BGD and is considered to be the most preferred method. It separates the training set into small batches and performs an update for each of these batches. It creates a balance between the efficiency of BGD and the robustness of SGD. MBGD is commonly used for deep learning problems.

V. CASCADING

One of many ways that people try to escalate the accuracy of a model in classification is by merging different predictions from various algorithms [7]. It has been observed time and again that this amalgamation of various predictions is likely to have superior accuracy if it is compared to a regular sole prediction form one algorithm.

A cascading technique is mostly used to detect fraudulent credit card transactions, or maybe when you want to be absolutely sure that you don't have cancer. A typical scenario when a cascading algorithm is used is when the cost of making a mistake is very high.

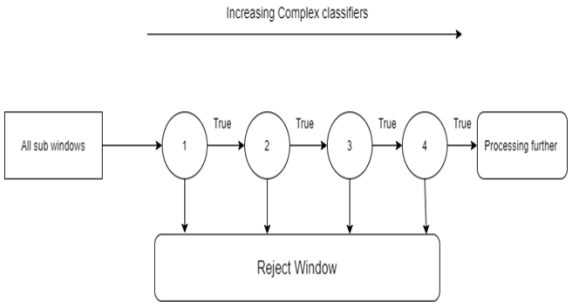

Fig 3. Cascading Algorithm.

A very common area where the cascading algorithms are used extensively is in the area of neural network. Neural network ensembles [8] are turning a lot of heads in the recent neural network research, due to their engrossing and precise feature event will occur or not (in terms of 0 and 1) based on values of input variables. For example, predicting if a tumor is malignant or benign or an e-mail is classified as spam or not are the instances which can be considered as binomial outcome of Logistic Regression.

There can be multinomial outcome of Logistic Regression as well e.g. prediction of type of cuisine preferred: Chinese, Italian, Mexican etc. There can be ordinal outcome as well like: product rating 1 to 5 etc. So Logistic Regression deals with prediction of target variable which is categorical. Whereas Linear Regression deals with prediction of values of continuous variable eg. Prediction of real estate price over a span of 3 years.

Logistic Regression has the following advantages: simplicity of implementation, computational efficiency, efficiency from training perspective, ease of regularization. No scaling is required for input features. This algorithm is predominantly used to solve problems of industry scale.

As the output of Logistic Regression is a probability score so to apply it for solving business problem it is required to specify customized performance metrics so as to obtain a cutoff which can be used to do the classification of the target. Also logistic regression is not affected by small noise in the data and multi- co linearity.

Logistic Regression has the following disadvantages: inability to solve non-linear problem as its decision surface is linear, prone to over fitting, will not work out well unless all independent variables are identified.

Some examples of practical application of Logistic Regression are: predicting the risk of developing a given disease, cancer diagnosis, predicting mortality of injured patients and in engineering for predicting probability of failure of a given process, system or product.

VI. BACK PROPAGATION ALGORITHM

This algorithm provides a very simple and efficient way to compute the gradient in a neural network and one can use it in conjunction with stochastic gradient descent which is also quite simple. There are more complex "quasi-Newton" techniques which make a better estimate of the gradient direction and step size, but they don't perform better than back prop and SGD.
Back Propagation Algorithm is used in deep learning. Neural Network (NN) has its specific applications in different industry segments and it has its merits and

demerits. Scenarios where there are no well defined criteria or rules to find an answer then NN is useful. It gives the solution but it becomes difficult to explain how the solution is arrived at and so it is like a black box. NN finds its application in classification of credit rating and in forecasting market dynamics in financial sector.

Here are some of the applications of NN in marketing segment: in product classification, in classification of customer segments i.e. which customers will like and purchase which products, finding new market for specific product category, in associating relationship between customer and company. NN becomes instrumental in increasing revenue of a business house, in increasing the percentage of response to direct marketing. NN finds its application in Post offices for sorting the letters/parcels based on area zip code / postal code.

Following are the merits of NN for which it is widely used in industry segments as mentioned above: easy adaption to new scenarios, fault tolerant, and ability to handle noisy data.

Short coming of NN are the following: training time of NN is very long and for training the NN efficiently, the sample sets derivative of the activation function is very small and it results in Network Paralysis. ANN with multilayer needs many repeated presentations of the input patterns, in which we need to adjust the weights so that an optimal solution is achieved with the network settling down.

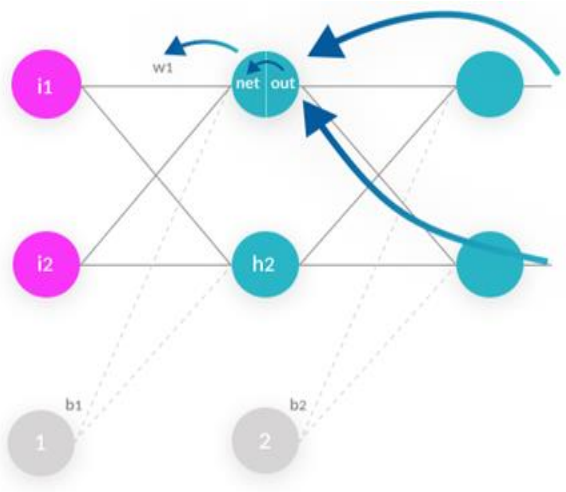

Fig 4. Back Propagation Algorithm.

VII. CONCLUSION

Some very efficient and lesser known techniques were discussed in this paper which would definitely help any machine learning algorithm designer. These techniques are underdogs of the ML community and are best friends of every algorithm designer.

REFERENCES

[1] Breiman, L.: Bagging predictors. Technical Report 421, Department of Statistics,University of California at Berkeley, 1994.

[2] https://towardsdatascience.com/boosting-algorithms-explained-d38f56ef3f30 (accessed on 15/02/2021)

[3] http://datascience.la/benchmarking-random-forest-implementations/ (accessed on 17/02/2021)

[4] https://towardsdatascience.com/boosting-algorithm-adaboost-b6737a9ee60c (accessed on 18/02/2021)

[5] Yoav Freund and Robert E. Schapire. A decisiontheoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1):119{139, August 1997.

[6] TU Chengsheng , LIU Huacheng , XU Bing: AdaBoost typical Algorithm and its application research, MATEC Web of Conferences, January 2017.

[7] A A Aziz, Indahwati and B Sartono: Improving prediction accuracy of classification model using cascading ensemble classifiers, IOP Conference Series Earth and Environmental Science, July 2019

[8] M. P. Perrone and L. N. Cooper. When networks disagree: Ensemble methods for hybrid neural networks. In R. J. Mammone, editor, Neural Networks for Speechand Image Processing, pages 126–142. Chapman – Hall, 1993