Online Boosting Algorithm for Regression with Additive and Multiplicative Updates

Ali H. Mirza
Department of Electrical and Electronics Engineering
Bilkent University, Ankara 06800, Turkey
mirza@ee.bilkent.edu.tr

***Abstract*—In this paper, we propose a boosted regression algorithm in an online framework. We have a linear combination of the estimated output for each weak learner and weigh each of the estimated output differently by introducing ensemble coefficients. We then update the ensemble weight coefficients using both additive and multiplicative updates along with the stochastic gradient updates of the regression weight coefficients. We make the proposed algorithm robust by introducing two critical factors; significance and penalty factor. These two factors play a crucial role in the gradient updates of the regression weight coefficients and in increasing the regression performance. The proposed algorithm is guaranteed to converge in terms of exponentially decaying regret bound in terms of number of weak learners. We then demonstrate the performance of our proposed algorithm on both synthetic as well as real-life data sets.**

Keywords. Boosting, regression, ensemble learning, boosted regression, multiplicative updates

I. Introduction

A. Preliminaries and Related Work

Boosting algorithms are ensemble methods that work on the class of base functions with weak predictive or estimating power and convert them to highly efficient learning algorithms that have strong predictive capability [1], [2]. As an ensemble learning method [3], boosting combines several parallel running weakly performing algorithms to build a final strongly performing algorithm [1], [2], [4]. This is done by searching suitable a linear combination of weak learners in order to enhance the accuracy measure or minimizing the loss function [5]-[6]. Boosting methods are commonly applied to various dilemmas in the machine learning literature including classification [1], regression [6], and prediction [7]. But there is a very little literature in boosting for online regression. Mostly the boosting is done on the data in a batch setting which is not desirable in many fields where we have a huge corpus of data in an online framework. Online boosting is of vital importance and is widely done for classification purposes. In [8] theoretical bounds for the online boosting for classification are developed.

AdaBoost and Gradient Boosting are the most commonly used boosting methods in a wide arena of applications [9]. But the problem with these methods is that they operate in a batch setting which is not desirable for online classification framework applications. Moreover, another disadvantage of batch setting is that for big data applications, the memory is not sufficient enough to perform the boosting for classification using batch setting [1]. Chen [8] first introduced the idea of online boosting for classification. Later, in [10], the authors formulated an optimal online boosting algorithm.

Most of the literature on boosting is for classification while there is very less literature about boosting for regression. Usually, the boosting for regression is taken in terms of greedy stepwise models [11], [12]. Most of the work on the boosting for regression does not talk about any guarantee on the convergence of the algorithm [13]. In [13], [9] bounds on the speed of convergence and convergence proofs are presented. In [8] the boosting for regression is done by first converting the problem to classification task and then perform boosting. Mostly the boosting for regression is done in the batch setting. Such a framework is not desirable where we have to deal with huge amount of data in an online manner.

B. Contributions

Our main contributions are as follows:

- We developed a boosted regression algorithm in an online setting with a guaranty on the convergence of the algorithm with an exponentially decaying regret in terms of number of weak learners.We have excluded the theorem and proof of the convergence of this algorithm considering the restriction on the number of pages.
- We introduced two critical factors; significance and penalty factor; that helps in enhancing the overall regression performance of the algorithm.

II. Problem Description

In this paper, all vectors are column vectors and denoted by boldface lower case letters. Matrices are represented by boldface upper case letters. For a vector $\boldsymbol{u}$, $|\boldsymbol{u}|$ is the $\ell 1$-norm and $\boldsymbol{u}^T$ is the ordinary transpose.

In our problem setting, we sequentially receive regression vectors $\{\boldsymbol{x}_t\}_{t=1}^n, \boldsymbol{x}_t \in \mathbb{R}^p$, where n can be fixed or on-going. We also receive desired output $\{d_t\}_{t=1}^n, d_t \in \mathbb{R}$. For a given online learning algorithm, i.e., $f_t(\cdot)$, we estimate the desired output as $\hat{d}_t = f_t(\boldsymbol{x}_t)$. After estimating the desired output $\hat{d}_t$, we get desired output d_t and then calculate the mean square error, i.e., $e(t) = (d_t - \hat{d}_t)^2$. We then update the parameters of the weak learners based on $e(t)$. Mean squared error is most

978-1-5386-1501-0/18/$31.00

commonly used since it belongs to the class of smooth loss functions.

For the given online learning algorithm, we may use linear or non-linear modelling for estimating desired output. Commonly, linear modelling is preferred over non-linear modelling. We use linear modeling to estimate the desired output as $\hat{d}_t = \boldsymbol{w}_t^T \boldsymbol{x}_t$, where $\boldsymbol{w}_t \in \mathbb{R}^p$ is the linear algorithm coefficient. Based on the error measure, i.e., $e(t)$, we update the $\boldsymbol{w}_t$ coefficient vector. In short, we want to minimize the following

$$\boldsymbol{w}_t = \arg\min_{\boldsymbol{w}} \sum_{i=1}^{t-1} \big(d_i - \boldsymbol{w}_i^T \boldsymbol{x}_i\big)^2, \tag{1}$$

where the solution to the above mentioned minimization problem (1) is as follows:

$$\boldsymbol{w}_t^* = \big(\sum_{i=1}^{t} \boldsymbol{x}_i \boldsymbol{x}_i^T\big)^{-1} \big(\sum_{i=1}^{t-1} \boldsymbol{x}_i d_i\big). \tag{2}$$

We know from the literature of the Follow The Leader (FTL) approach [14], the upper bound on the convergence can be as follows:

$$\sum_{i=1}^{t} (e_i^2 - e_i^{*2}), \tag{3}$$

where $e_i = d_i - \boldsymbol{w}_i^T \boldsymbol{x}_i$ and $e_i^* = d_i - \boldsymbol{w}_i^{*T} \boldsymbol{x}_i$.

For the boosted regression framework in an online setting, we have q weak learners each one of them have their own estimates, i.e., $\hat{d}_t^{(i)}, i = 1, \ldots, q$, We ensemble the estimates of all the weak learners via linear combination, i.e., by weighting each weak learner's output differently. We use $\boldsymbol{v}_t^T$ for weighing weak learner's output and obtain a final estimate of the desired output as follows:

$$\hat{d}_t = \boldsymbol{v}_t^T \boldsymbol{\kappa}_t, \tag{4}$$

where $\boldsymbol{\kappa}_t = [\hat{d}_t^{(1)} \hat{d}_t^{(2)} \ldots \hat{d}_t^{(q)}]^T$ and $\hat{d}_t^{(i)} = \boldsymbol{w}_t^{(i)T} \boldsymbol{x}_t$. For each weak learner, we assign an significance factor, i.e., $\psi_t^{(i)}, \forall i$, that plays a critical role in the updates of the parameters of each weak learner and helps in sustaining the desired MSE of the system. We use similar assignment of significance factor as mentioned in [8] as follows:

$$\psi_t^{(i)} = \min\big\{1, (\theta^2)^{0.5\zeta_t^i}\big\}, \tag{5}$$

where θ^2 is the desired MSE and ζ_t^i is the penalty factor transferred to i^{th} weak learner from $(i-1)^{th}$ weak learner. The penalty factor for each weak learner is calculated as follows:

$$\zeta_t^i = \theta^2 - (e_t^{(i)})^2, \tag{6}$$

where $e_t^{(i)} = d_t - \hat{d}_t^{(i)}$.

Remark 1: *The penalty factor is also of utmost importance because it helps the overall system to keep track of the performance record. For example, if the $(i-1)^{th}$ weak learner does not work well on the data instance $\boldsymbol{x}_t$, then a higher penalty factor is transferred to i^{th} weak learner that compels the i^{th} weak learner to perform well on the incoming data instance.*

Based on significance and penalty factor, the parameter $\boldsymbol{w}_t^{(i)}$ of weak learners is updated based on stochastic gradient descent (SGD) update as follows:

$$\boldsymbol{w}_t^{(i)} = \boldsymbol{w}_t^{(i-1)} + \eta \psi_t^{(i-1)} \boldsymbol{x}_t \big(d_t - \boldsymbol{x}^T \boldsymbol{w}_t^{(i-1)}\big). \tag{7}$$

After updating the parameters of all the weak learners, we update the ensemble vector weight, i.e., $\boldsymbol{v}_t$ as follows:

$$\boldsymbol{v}_t = \boldsymbol{v}_{t-1} + \mu e_t \frac{\boldsymbol{\kappa}_t}{||\boldsymbol{\kappa}_t||^2}. \tag{8}$$

Remark 2: *The significance factor plays a critical role in the parameter update of each weak learner. Greater the value of significance factor, greater the amount of change in the parameter update and vice versa.*

The detailed schematic diagram of the proposed boosted regression algorithm is shown in Fig. 1 and Algorithm 1, gives the overall steps involved schematically in the process.

Algorithm 1 Boosted Regression Algorithm with Significance and Penalty Factor

Input: Receive $(\boldsymbol{x}_t, d_t)$ regression vector and desired output. Initialize the number of weak learners, ensemble coefficients $\boldsymbol{v}_t = [1, 1, \ldots, 1]^T$, significance factor $\psi_t^{(i)} = 1$ and weight coefficients $\boldsymbol{w}_1^{(i)}$ for each weak learner
for $t = 1$ to T
  Receive $\boldsymbol{x}_t$
  Compute $\boldsymbol{\kappa}_t = [\hat{d}_t^{(1)} \hat{d}_t^{(2)} \ldots \hat{d}_t^{(q)}]$
  Predict the desired output $\hat{d}_t = \boldsymbol{v}_t \boldsymbol{\kappa}_t$
  Receive d_t and initialize $\psi_t^{(1)} = 1, \zeta_t^{(1)} = 0$
  for $i = 1$ to q
    $\psi_t^{(i)} = \min\big\{1, (\theta^2)^{0.5\zeta_t^i}\big\}$
    $\boldsymbol{w}_t^{(i)} = \boldsymbol{w}_t^{(i-1)} + \eta \psi_t^{(i-1)} \boldsymbol{x}_t \big(d_t - \boldsymbol{x}^T \boldsymbol{w}_t^{(i-1)}\big)$
    $e_t^{(i)} = d_t - \hat{d}_t^{(i)}$
    $\zeta_t^{(i+1)} = \zeta_t^{(i)} + \big(\theta^2 - (e_t^{(i)})^2\big)$
  end for
  $\boldsymbol{v}_t = \boldsymbol{v}_{t-1} + \mu e_t \frac{\boldsymbol{\kappa}_t}{||\boldsymbol{\kappa}_t||^2}$
end for

III. Experiments

In this section, we validate the performance of our proposed boosted regression algorithm on synthetic and real-life data sets. We use various real-life data sets like Kinematics and Alcoa Corporation Stock Price data set.

A. Synthetic Data Set

We generate a stationary environment that generates 3-dimensional regression vectors, i.e., $\boldsymbol{x}_t = [x(1), x(2), 1]^T$ in an affine manner. Regression vectors are jointly gaussian and are in the range $[0\ 1]^2$. The desired output is calculated as $d_t = \boldsymbol{w}_t^T \boldsymbol{x}_t + \boldsymbol{\nu}_t$, where $\boldsymbol{\nu}_t$ belongs to normal gaussian distribution with zero mean and 0.01 variance. We use $q = 10$

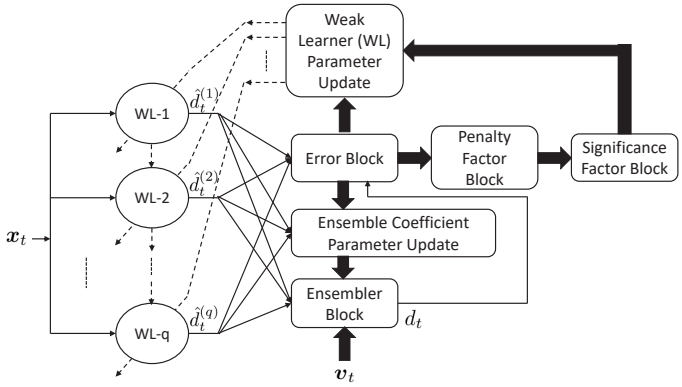

Fig. 1. Detailed schematic diagram of Boosted Regression Algorithm with Significance and Penalty Factor. Dotted lines shows the updates to be done on the parameters of each weak learner (WL). Here, $\boldsymbol{x}_t$ is the regression vector and $\boldsymbol{v}_t$ is the weak learner's weight output vector.

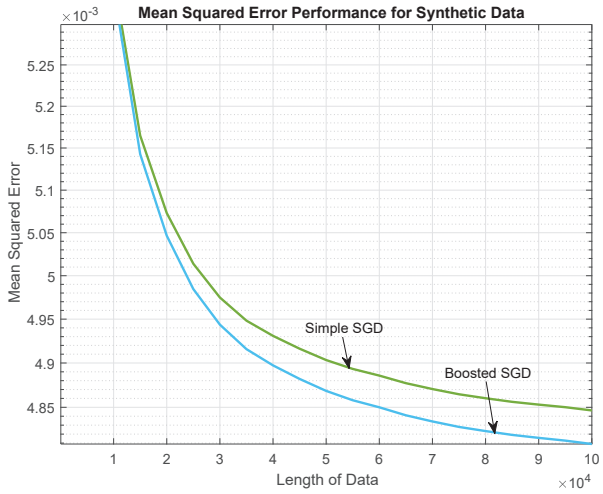

Fig. 2. MSE performance of the Boosted SGD regression algorithm with 10 weak learners compared with simple SGD regression algorithm with single learner. The MSE curves shown are averaged for 500 trials to show a smooth trend.

weak learners, $\eta = 0.01$ learning rate for each weak learner and θ^2 is the desired MSE. In Fig. 2, we observe that the weak learners gradually learn and reduces the total error. The decaying rate of the error is strongly dependent on the learning rate, number of weak learners and desired MSE.

B. Real Life Data Sets

In this subsection, we demonstrate the performance of our proposed boosted regression algorithm on various real-life data sets, i.e., Kinematics and Alcoa Corporation Stock Price data set. Table. I, shows the mean squared error performance of the real-life data sets for additive and multiplicative updates of ensemble coefficients respectively. In order to provide a fair experimental setup, we selected learning parameter $\eta = 0.01$ based on cross validation for all the experiments.

1) Effect of Number of Weak Learners: We carry out the experiment with learning rate $\eta = 0.01$ for each weak learner,

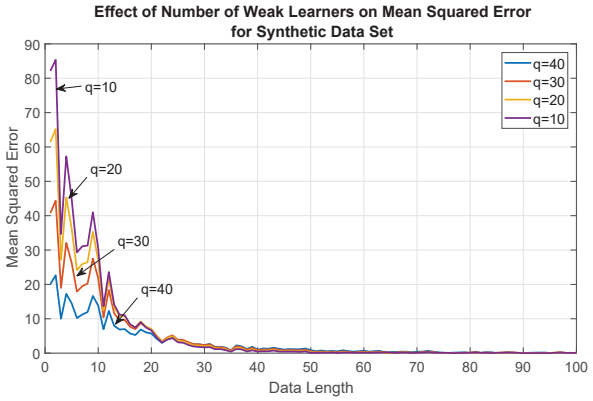

Fig. 3. MSE curve trend for various values of number of weak learners. We observe that as the number of weak learners increases, the predicting power capability of the whole system increases resulting in decrease of MSE.

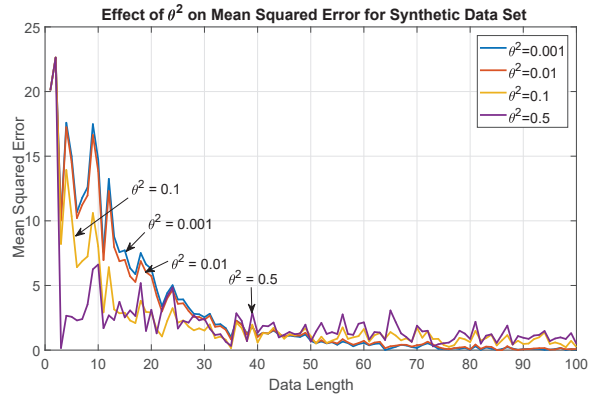

Fig. 4. MSE trend for various values of desired MSE of the overall system to be achieved with 10 weak learners. As the desired MSE level increases, the boosted algorithm drastically reduces the MSE as shown in the curve above.

desired MSE level $\theta^2 = 0.01$ and various values of number of weak learners $q = 10, 20, 30$ and 40 over the synthetic data set as shown in Fig. 3. We observe that, as we increase the number of weak learners, there is a decrease in the MSE as shown in Fig. 3. But this decrease in MSE is up to certain level of the length of the data. After some value, the MSE is almost the same. Hence, we must select an appropriate value for the number of weak learners in order to reduce the computational complexity and still have a desired final MSE value.

2) Effect of varying θ^2: We perform the experiment using 10 weak learners each having a learning rate of $\eta = 0.01$. We see from Fig. 4, that as we increase the value of θ^2 there is a significant and fast decrease in the MSE value of the algorithm. The value of θ^2 also has an effect on the regression coefficient weight updates. For very large values of θ^2, the weight updates are more drastic. This is not suitable always as this huge change in the regression coefficient weight updates may cause the system to oscillate. The value of θ^2 should be chosen upon cross-validation that results in a quick decrease in MSE along with producing no oscillation and blowing up of weights and MSE.

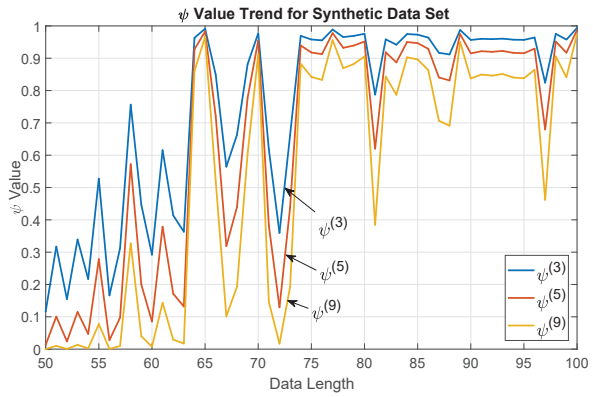

Fig. 5. Significance value trend for third, fifth and ninth weak learner.

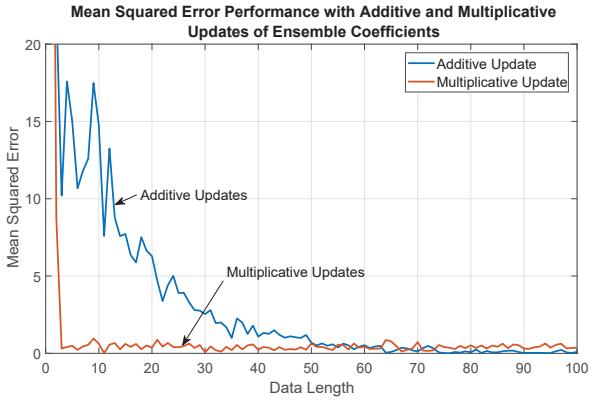

Fig. 6. MSE trend for ensemble coefficient with both additive and multiplicative updates. The effect of multiplicative updates on the ensemble coefficients is very drastic as compared to the one with additive updates.

3) Significance Factor Trend: In Fig. 5, we display the significance factor value trend for various weak learners for the synthetic data set. We show the trend for third, fifth and ninth weak learner. We observe that the starting weak learner plays more part in improving the regression performance. This is evident from the Fig. 5, that third weak learner significance factor is greater than others throughout the data length. One more important point to notice is that if one of the starting weak learner's significance factor $\psi^{(i)}$ reduces, then rest of the $\psi^{(j)}, j > i, j \neq i$, also decreases and vice versa.

Remark 3:*We also implemented the proposed algorithm with multiplicative updates. As shown in Fig. 6, we observe that the effect of multiplicative updates is drastic as compared to the additive updates in the start. After some time, the performance of the proposed algorithm with both additive and multiplicative updates is almost the same.*

IV. Conclusion

We proposed a boosted regression algorithm with SGD updates to improve the overall MSE performance. We introduced two critical factors namely: significance factor and penalty factor, to improve the regression performance. Each weak learner evaluates its own error and then a penalty factor is generated that is passed to the next weak learner. This penalty factor compels the next weak learner to perform according to the performance of the previous weak learner. The penalty factor is then used in evaluating the significance factor that plays a critical role in the gradient updates of the regression weight coefficients. The significance factor helps in sustaining the desired MSE level and helps in the convergence of the regret bound of the algorithm. We demonstrate the performance of our proposed boosted regression algorithm on synthetic as well as real-life data sets. We observe that the proposed algorithm performs better than the simple regression algorithm.

TABLE I
MSE PERFORMANCE FOR REAL LIFE AND SYNTHETIC DATA SETS FOR SIMPLE AND BOOSTED REGRESSION ALGORITHMS WITH ADDITIVE AND MULTIPLICATIVE UPDATES.

Data Sets/ Algorithms	Kinematics	Alcoa	Elevators
SGD (Add. Upd.)	0.2710	0.0128	0.004846
Boosted SGD (Add. Upd.)	**0.2687**	0.0111	**0.004809**
SGD (Mult. Upd.)	0.2702	0.0128	0.004888
Boosted SGD (Mult. Upd.)	0.2684	**0.0109**	0.004829

References

[1] R. E. Schapire and Y. Freund, "Boosting: Foundations and algorithms, adaptive computation and machine learning series," 2012.

[2] A. Beygelzimer, S. Kale, and H. Luo, "Optimal and adaptive algorithms for online boosting," in *International Conference on Machine Learning*, pp. 2323–2331, 2015.

[3] T. G. Dietterich, "Ensemble learning," *The handbook of brain theory and neural networks*, vol. 2, pp. 110–125, 2002.

[4] L. Mason, J. Baxter, P. L. Bartlett, and M. R. Frean, "Boosting algorithms as gradient descent," in *Advances in neural information processing systems*, pp. 512–518, 2000.

[5] T. Zhang, B. Yu, *et al.*, "Boosting with early stopping: Convergence and consistency," *The Annals of Statistics*, vol. 33, no. 4, pp. 1538–1579, 2005.

[6] N. Duffy and D. Helmbold, "Boosting methods for regression," *Machine Learning*, vol. 47, no. 2-3, pp. 153–200, 2002.

[7] S. B. Taieb and R. J. Hyndman, "A gradient boosting approach to the kaggle load forecasting competition," *International journal of forecasting*, vol. 30, no. 2, pp. 382–394, 2014.

[8] S.-T. Chen, H.-T. Lin, and C.-J. Lu, "An online boosting algorithm with theoretical justifications," *arXiv preprint arXiv:1206.6422*, 2012.

[9] M. Collins, R. E. Schapire, and Y. Singer, "Logistic regression, adaboost and bregman distances," *Machine Learning*, vol. 48, no. 1-3, pp. 253–285, 2002.

[10] A. Beygelzimer, S. Kale, and H. Luo, "Optimal and adaptive algorithms for online boosting," in *International Conference on Machine Learning*, pp. 2323–2331, 2015.

[11] T. J. Hastie and R. J. Tibshirani, "Generalized additive models, volume 43 of monographs on statistics and applied probability," 1990.

[12] T. Hastie, R. Tibshirani, and J. Friedman, "The elements of statistical learnine," 2001.

[13] L. Mason, J. Baxter, P. L. Bartlett, and M. R. Frean, "Boosting algorithms as gradient descent," in *Advances in neural information processing systems*, pp. 512–518, 2000.

[14] S. Shalev-Shwartz *et al.*, "Online learning and online convex optimization," *Foundations and Trends® in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2012.