Empirical Analysis of Boosting Error Bound

Arman Bolatov [*1] **Kaisar Dauletbek** [*2]

Abstract

This study empirically verified the error bound of the AdaBoost algorithm for both synthetic and real-world data. The results of this research show that the error bound holds up in practice, demonstrating its efficiency and importance to a variety of applications. Understanding the accuracy limits of the AdaBoost algorithm is essential for data scientists to properly measure performance so they can continually improve their models' predictive capabilities. The corresponding source code is available at github.com/armanbolatov/adaboost_error_bound.

1. Introduction

In this report, we aim to present an empirical verification of the AdaBoost (Schapire, 2013) algorithm. We are going to do so by first presenting the theoretical error bounds along with the necessary conditions, and initial derivations. Afterwards, we will describe an experimental setup and report on the findings. Finally, we will apply the designed experiments on both synthetic and real-world data to provide empirical verification. The theoretical part of this report is based on "Foundations of Machine Learning" book (Mohri et al., 2018).

1.1. AdaBoost Algorithm

AdaBoost is a special type of boosting algorithms, which are designed to construct a strong PAC-learnable algorithm by means of combining distinct weak PAC-learnable classifiers (base classifiers) (Mohri et al., 2018). The formal algorithm for the implementation of AdaBoost is presented in Algorithm 1.

A more intuitive interpretation of AdaBoost is that the algorithm aims to combine the base classifiers by assigning particular weights to each of them. Each weight is calculated in accordance with the number of misclassifications the base classifiers returns. That makes the final combined prediction of the ensemble model more robust.

Algorithm 1 AdaBoost

Input: data $S = ((x_1, y_1), ..., (x_m, y_m))$
for $i \leftarrow 1$ **to** m **do**
  $D_1(i) \leftarrow \frac{1}{m}$
end for
for $t \leftarrow 1$ **to** T **do**
  $h_t \leftarrow$ classifier in H with $\varepsilon_t = P_{i\sim D_t}[h_t(x_i) \neq y_i]$
  $\alpha_t \leftarrow \frac{1}{2}\log\frac{1-\varepsilon_t}{\varepsilon_t}$
  $Z_t \leftarrow 2[\varepsilon_t(1-\varepsilon_t)]^{\frac{1}{2}}$
  for $i \leftarrow 1$ **to** m **do**
    $D_{t+1}(i) \leftarrow \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$
  end for
end for
$f \leftarrow \sum_{t=1}^{T} \alpha_t h_t$
return f

1.2. Factors That Influence AdaBoost's Performance

The base learner is the individual model taken from certain family of functions $\mathscr{H}$ that is used to make predictions. For the experiment, we chose a $d-1$ dimensional perceptron via `sklearn.Perceptron` due to its VC-dimension being equal to d, as required by equation (1). In this paper, we will refer to the set of base learners as a vector $\mathbf{h}$.

The weight coefficients are the real numbers that represents the significance of each individual prediction of a base learner in the final ensemble. We will denote weights as a single vector α.

The number of iterations determines the number of base learners that are used. It has been generally observed that the more base learners are used, the better the performance of the model. However, a trade-off exists between the number of iterations and the computational cost. Surprisingly, the number of rounds of boosting (referred to as T), does not appear in the generalization bound.

The data set used for training also has a significant impact on the performance of AdaBoost, as boosting is particularly

[1]Department of Computer Science, Nazarbayev University, Astana, Kazakhstan [2]Department of Mathematics, Nazarbayev University, Astana, Kazakhstan. Correspondence to: Arman Bolatov <arman.bolatov@nu.edu.kz>, Kaisar Dauletbek <kaisar.dauletbek@nu.edu.kz>.

Preliminary work.

effective on datasets with a large number of features.

1.3. Geometric Margin Over a Dataset

The L_1-geometric margin ρ_f of a linear function $f = \sum_{t=1}^{T} \alpha_t h_t$ over a dataset $S = (x_1, \ldots, x_n)$, is defined as,

$$\rho_f = \min_{i \in [m]} \frac{|\alpha \cdot \mathbf{h}(x_i)|}{\|\alpha\|_1} = \min_{i \in [m]} \frac{\left|\sum_{t=1}^{T} \alpha_t h_t\right|}{\|\alpha\|_1}.$$

The margin serves an important role in error bound analysis, as it indicates the "separability" of classes. That is, the larger the margin, the more separable the Gaussian clusters in the dataset are for a function f, and the easier the classification task will be.

1.4. Ensemble VC-Dimension Margin Bound

In (Mohri et al., 2018) there is a following error bound:

Theorem. Let $\mathscr{H}$ be a family of functions taking values in $\{+1, -1\}$ with VC-dimension d. Select a sample set S with size m and fix L_1-geometric margin ρ. Then, for any $\delta > 0$, with probability at least $1 - \delta$, the following holds for all $h \in \text{conv}(\mathscr{H})$

$$R(h) \leq \hat{R}_{S,\rho}(h) + \frac{2}{\rho}\sqrt{\frac{2d \log \frac{em}{d}}{m}} + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}, \qquad (1)$$

where e is the Euler's constant, $\text{conv}(\mathscr{H})$ is the convex hull of $\mathscr{H}$, $R(h)$ and $\hat{R}_{S,\rho}$ are the errors (misclassification rates) on the testing and training set, respectively.

2. Methodology

The error of AdaBoost will be analyzed through experimental data, which will come from randomly generated datasets and the "Heart Disease Health Indicators" dataset with varied properties such as size and dimensionality. For synthetic data, the `sklearn.make_classification` with parameters `class_sep=0.5` and `flip_y=0.05` will be used to generate two Gaussian clusters for binary classification. Each dataset will be split into equal in size training and testing sets via `sklearn.train_test_split`. Train set will be used to fit the AdaBoost classifier, and the misclassification rates for both sets will be recorded. `Pandas`, `matplotlib`, and `seaborn` will also be used for data analysis and visualization.

We will conduct three experiments, investigating the influence of sample size of the train set m, VC-dimension of the base learner d, and the number of AdaBoost's iteration T on the difference of the training and testing errors, which will be denoted as $\Delta R(h) := R(h) - \hat{R}_{S,\rho}(h)$. Then, we will evaluate the theoretical error bound from the equation (1) and look at the relationship between ΔR and m, d, T.

In the following experiments we will fix the parameter δ to be equal to 0.05.

3. Experimental Results

3.1. Effect of the Number of Iterations

First we will test the influence of the number of base learners T on the error. We ran two experiments with different parameters for d and m, evaluated train/test errors of the classifier and averaged them by 100 iterations. The results are shown in Figure 1.

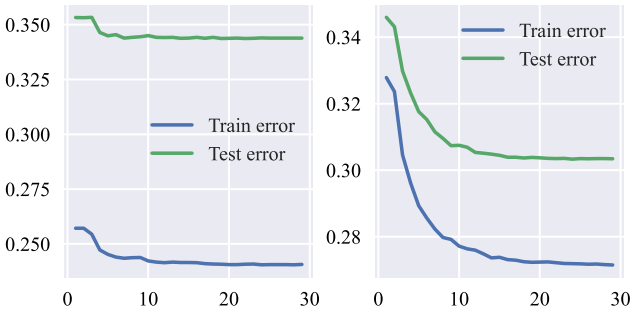

Figure 1. Results of the experiment 3.1 for $d = 50, m = 1000$ (left) and $d = 100, m = 500$ (right). The x-axis represents the number of iterations T, while the blue and green lines on the y-axis represent the errors on the training and testing sets, respectively.

As can be seen in the graph, the test error looks like the train error but shifted down by a constant amount. Hence the difference between errors is also approximately constant, meaning that ΔR is not affected T.

3.2. Effect of the Sample Size

The equation (1) can be rewritten as

$$\Delta R(h) \leq \frac{2}{\rho}\sqrt{\frac{2d \log \frac{em}{d}}{m}} + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}. \qquad (2)$$

Denote the right hand side as $\varepsilon_{\text{boost}}(\rho, d, m, \delta)$. The inequality above suggests that $\Delta R(h) \in O\left(\sqrt{\frac{\log m}{m}}\right)$ and the difference of errors will slowly decrease without exceeding the theoretical bound.

We will verify this hypothesis by the following steps:

1. Choose d to be equal 25, 50, 75, and 100.
2. Generate train and test sets with dimension $d - 1$ and varying sample size m from 10 to 10000 with step 10.
3. Calculate the L_1-margin ρ and the theoretical error bound $\varepsilon_{\text{boost}}(\rho, d, m, \delta)$.
4. Find the difference of error on train and test sets $\Delta R(h)$.
5. Scatter plot ΔR versus $\varepsilon_{\text{boost}}$.

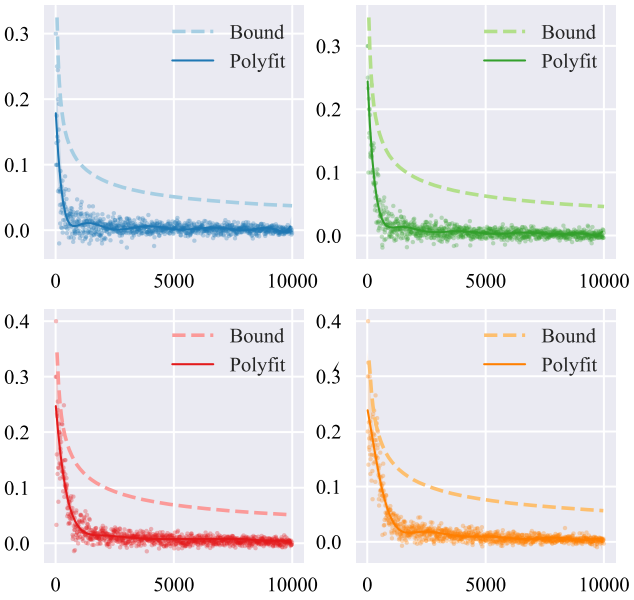

Figure 2. Results of the experiment 3.2 for different VC-dimensions d: the blue graph is $d = 25$, the green graph is $d = 50$, the red graph is $d = 75$, and the yellow graph is $d = 100$. The x-axis is the sample size m, the y-axis is the difference of the error on training and testing sets ΔR. Solid line represents polynomial fit for the training data. Dashed line represents the theoretical error bound.

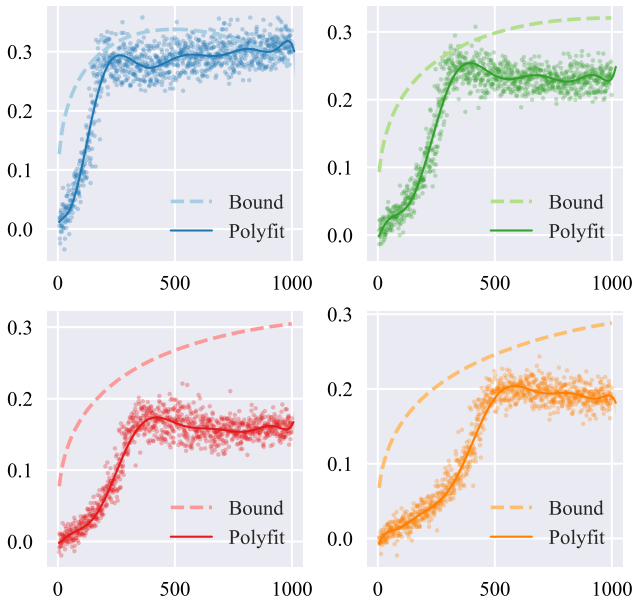

Figure 3. Results of the experiment 3.3 for different sample size m: the blue graph is $m = 500$, the green graph is $m = 1000$, the red graph is $m = 1500$, and the yellow graph is $m = 2000$. The x-axis is the VC-dimension d, the y-axis is the difference of the error on training and testing sets ΔR. Solid line represents polynomial fit for the training data. Dashed line represents the theoretical error bound.

The results can be seen in Figure 2. For clarity, we provided a polynomial fit of order 10. As expected, ΔR doesn't exceed the error bound and stays around 0 as we increase m.

3.3. Effect of the Base Learners' VC-dimension

Analogously, we can derive $\Delta R(h) \in O\left(\sqrt{Cd - d\log d}\right)$ (C is large enough number) from the equation (2). It suggests that the difference between errors will increase quickly up to a certain point, then decrease slowly after that. Also without exceeding the theoretical bound.

We will verify that by the similar steps:

1. Choose m to be equal 500, 1000, 1500, and 2000.
2. Generate train and test sets with sample size m and varying dimension d from 5 to 1000.
3. Calculate the L_1-margin ρ and the theoretical error bound $\varepsilon_{\text{boost}}(\rho, d, m, \delta)$.
4. Find the difference of error on train and test sets $\Delta R(h)$.
5. Scatter plot ΔR versus $\varepsilon_{\text{boost}}$.

The results are shown in Figure 3. Indeed, for $m = 1500$ and 2000, the difference of errors stays below the theoretical bound. However for $m = 500$ and 1000, some values of ΔR exceed the bound.

3.4. Evaluation of the Confidence Parameter

Denote $(1-\delta)\cdot 100\%$ as the confidence parameter. Recall that we set $\delta = 0.05$. It means that with a 95% chance the equation (1) will hold. Let the experimental confidence be the proportion of parameters (m, d) when the the equation held from the list of all selected parameters. These experimental confidences are provided in Table 1.

Exp. 3.2	Confidence	Exp. 3.3	Confidence
$d = 25$	100%	$m = 500$	82.5%
$d = 50$	99.9%	$m = 1000$	99.3%
$d = 75$	99.7%	$m = 1500$	100%
$d = 100$	98.9%	$m = 2000$	100%

Table 1. Experimental confidences for experiments 3.2 and 3.3 for all parameters m and d, respectively.

It is apparent that in approximately seven out of eight instances, the experimental confidence remains close to 99%, substantially higher than 95%. This may be due to the fact that the training data was generated from a normal distribution, thus rendering it very suitable. However, equation (1) does not specify what the initial distribution entailed.

3.5. Experiments on Real Data

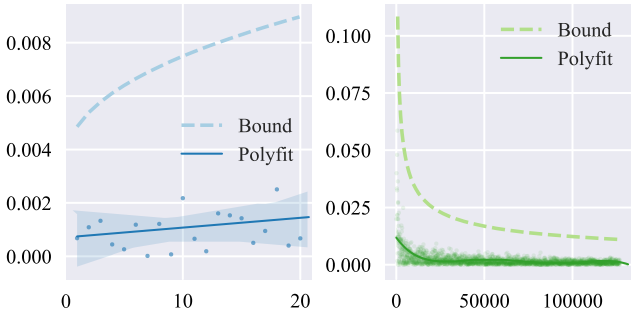

Figure 4. Results of the experiments 3.5. The left graph shows how error varies with respect to the VC-dimension of the base classifier; the x-axis is the VC-dimension d. The right graph shows the error when the sample size m varies; m is on the x-axis. The y-axis is the difference of the error on training and testing sets ΔR for both graphs. Solid line represents polynomial fit for the training data. Dashed line represents the theoretical error bound.

We chose "Heart Disease Health Indicators" dataset since it provides us with enough features and datapoints to run the proposed experiments, and is designed for binary classification task.

The total number of datapoints in the dataset is 253680 and the total number of features is 22. When running the experiments, we split the dataset into equal train and test splits, each with a total of 126840 datapoints to simplify the experimental procedures when calculating the theoretical error bound.

In order to analyze the effect of the sample size on the error of the AdaBoost algorithm, we set the VC-dimension of the base classifiers at 21, i.e. using all of the available features, and varying m (the training sample size) from 50 to 126840 with a step size of 50. Similarly, when assessing the effect of base classifier's VC-dimension, we vary the dimensionality of the inputs from 2 to 22, while fixing m at 126840. It is important to note that we do not vary d in a random manner, but feed in the features sorted by their importance. Their importance is calculated via scikit-learn's `feature_importances_` method. The results are summarized in Figure 4.

As we can see, the empirical error behaves as expected, and does not exceed the theoretical bound for both cases anywhere on the graph.

4. Conclusion

To conclude, in this work we have provided an empirical verification for the error bound of the AdaBoost algorithm. As the results show, we see that the bound holds for both the synthetic and real data, which was the initial purpose of this report.

5. Author Contributions

Theoretical analysis, A. B.; methodology A. B. and K. D.; synthetic data experiments, A. B.; real data experiments, K. D.; visualization, A. B.; editing, A. B. and K. D.; supervision, Zhenisbek Assylbekov.

References

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2 edition, 2018. ISBN 978-0-262-03940-6.

Schapire, R. E. Explaining adaboost. In *Empirical inference*, pp. 37–52. Springer, 2013.