Impact of Noise in Boosting Methods

Qi Feng, Xiaonan Zhao

***Abstract*—THIS IS THE abstract.**

I. Introduction

The fundamental idea of ensemble methods is to construct a combination of weak base classifiers that are diverse and result in a high accuracy. Multiple ensemble methods, including boosting[], bagging[], and decision tree ensemble[], are being introduced in the past 20 years. Boosting algorithms took a significant place in ensemble methods. Adaboost[1] and the recently introduced Deepboost[2] are typical boosting algorithms with a good experimental result without overfitting the training set. They both have good theoretical learning bound and benefit directly from minimizing the learning bound.

Boosting algorithms maintains a set of weights over the original training set S, and adjust these weights each iteration. They utilize the base classifiers and create a combination of these classifiers with a complex classifier that typically has a good performance. Boosting increases weight of samples that are mislabeled by the base classifier and decreases weight of samples that are correctly labeled during each iteration. Therefore, the algorithm will keep focus on the misclassified samples. As we shall discuss later, noise is typically distributed densely near the misclassified samples. Adaboost has been shown to be very effective in practical[3]. Since Adaboost is a special case of Deepboost by setting $\lambda = 0$ and $\beta = 0$, Deepboost will always out performs Adaboost. Therefore, both of these boosting algorithm will have a good performance in practical. However, the experimental robustness of these algorithms have not been tested before.

Our work is to test the robustness of boosting algorithms with experiments by introducing realistic noise into the training dataset. Finally, an explanation of the results is given based on the theoretical learning bound from both algorithms.

II. Realistic Noise

The impact of noise in the performance of an algorithm is regarded as the robustness. In Dietterich's work[4], multiple levels of noise are added to sample dataset to test the robustness of Adaboost. However, the noise introduced was by reverting labels in training data randomly without replacement with a fraction r. This makes the noise in the training dataset unrealistic. Many other publications have used different procedure to add noise into the training data The procedure is to set each training sample's label to a random class with probability r. Both of these procedures are adding noise with a uniform distribution over the entire training data.

A. Distribution of Realistic Noise

However, in actual datasets noise are not distributed with a uniform fashion across the entire dataset. In Xingquan et al.'s work[5], a general method to eliminate noise from training data is introduced. In this algorithm, noise identification is based on the majority and non-objection schemes, which is founded on the assumptions that noise is distributed according to the distribution of empirical errors. The denser the classification errors, the denser the noise in training set. More generally, realistic noise distribution should not be uniform over the entire training dataset, but with a relation to the classification error.

Following the definition of Adaboost, the distribution $D_t(i)$ updated during each iteration of Adaboost is a perfect simulation of the training error distribution after round t. Since $D_t(i)$ is updated with according to the loss function, the distribution would be updated with a higher level where empirical errors are denser. Therefore, the distribution from Adaboost after T rounds(finished) would be suitable for introducing realistic noise.

B. Margin

Following the margin theory in Adaboost, we have the margin defined as

$$\rho(x) = y\frac{\alpha \cdot \mathbf{h}(x)}{\|\alpha\|_2}.$$

The distribution maintained by Adaboost is

$$D_{t+1}(i) = \frac{\exp(-y_i\alpha_t \cdot \mathbf{h}_t(x_i))}{m\prod_{s=1}^{t} Z_s}.$$

It it trivial to see that $D_{t+1}(i)$ has a negative correlation with margin ρ. Therefore, since we are introducing noise with the distribution after T iterations of Adaboost with some stopping criteria, the samples whose margin are relatively small, i.e. that are closer to the threshold hyperplane in our experiment later and are difficult to classify by boosting, will have a higher distribution of noise. This makes the noise more realistic than the uniform distribution over the sample in a sense that noise are typically distributed at the samples that has a small margin and are difficult to classify by boosting.

III. Methods

We tested Adaboost and Deepboost on the UCI dataset, ionosphere[6]. We randomly split the data to two parts; 80% for the training data and 20% for testing.

A. Adaboost

B. Deepboost

Following the work from Cortes et al. [2], we use the H_1^{stumps} as the base classifier for Deepboost. The Rademacher complexity of H_1^{stumps} can be bounded by its growth function. It is trivial to see that

$$\Pi_{H_1^{stumps}}(m) \leq 2md,$$

since there are $2m$ distinct threshold functions for each dimension with m points. Therefore,

$$\mathfrak{R}_m(H_1^{stumps}) \leq \sqrt{\frac{2\log(2md)}{m}}.$$

By now, we have the notation from Deepboost

$$\Lambda_j = \lambda \cdot \mathfrak{R}_m(H_1^{stumps}) + \beta.$$

where we conducted experiments for $\lambda \in \{10^{-i} : i = 3, \cdots, 7\}$ and $\beta \in \{10^{-i} : i = 3, \cdots, 7\}$ as well, and optimize the training error on these experiments.

We optimize the parameter by minimizing the 10-fold cross validation, and then measure the error by testing data.

In all of our experiments, the number of iterations was set to 50. We also test the result for 100 rounds, but the test error remains basically the same. As we shall see, in some experiments the training error decreases vastly and reaches to zero after 40 iterations.

IV. Results

Observe that with the exponential loss, Deepboost has a smaller test error than Adaboost, which is in accordance with Cortes et al.'s work[2]. Noise (realistic), in general, does not affect the boosting algorithms harshly, which is different from Dietterich's work[4]. As is reported from Cortes et al. it is difficult to obtain statistically significant results for small datasets. Therefore, the level of the testing error at approximately less than 10% is acceptable.

Figure 10 shows the training error and test error of Adaboost of the original ionosphere dataset, while figure 2 to 4 shows the training error and test error with the introduced noise of corresponding level of 5%, 10% and 20%. On the other hand,

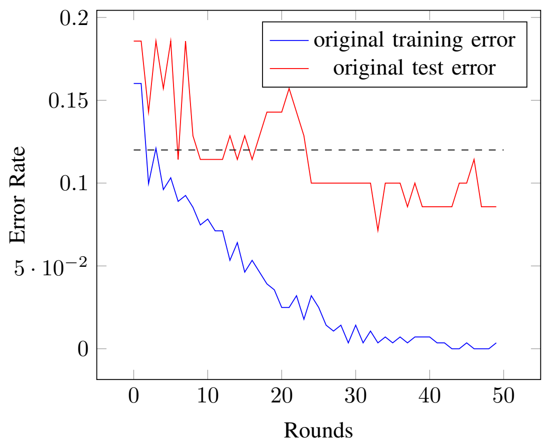

Fig. 1. Adaboost running on Ionosphere.

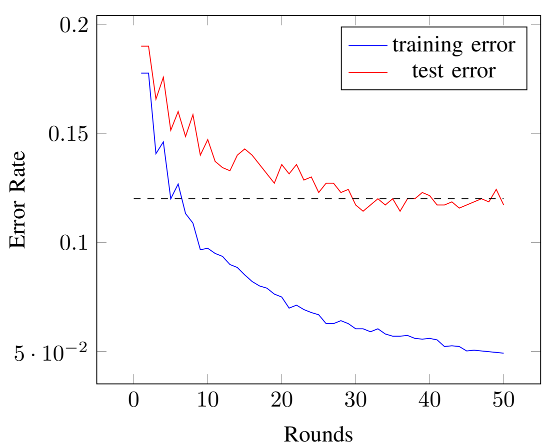

Fig. 2. Adaboost test on Ionosphere. 5% noise

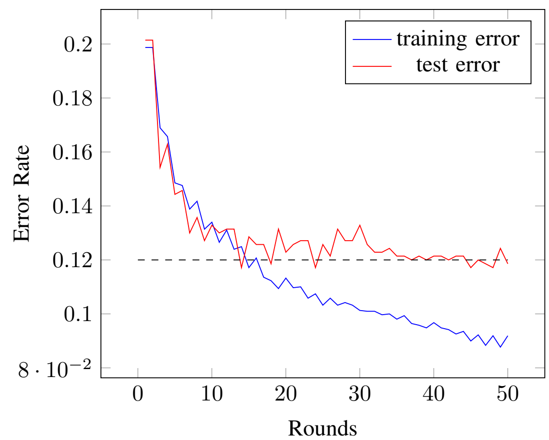

Fig. 3. Adaboost test on Ionosphere. 10% noise

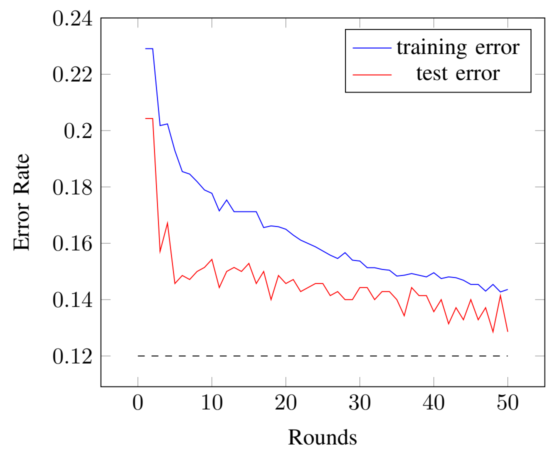

Fig. 4. Adaboost test on Ionosphere. 20% noise

figure 5 to 8 shows the training and test error of Deepboost on those datasets with noise.

In figure 2 to 4 and figure 5 to 8, with the increase of the noise, the training error is increased. An intuitive explanation would be that the training error contains many of the noise that is being introduced. These noise are distributed near the sample whose margin are relatively small. Therefore, these noise would be intuitively difficult to classify correctly by boosting.

V. Conclusion

We presented a practical analysis of the robustness of two boosting algorithm, Adaboost and Deepboost. We argued that uniformly introduced noise based on the distribution of sample does not reflect the realistic distribution. We introduced a new realistic way of simulating noise in training data, and utilized it to introduce noise into our experiments against these algorithms. We also reported that both Adaboost and Deepboost has a good performance with the realistic noise we intro-

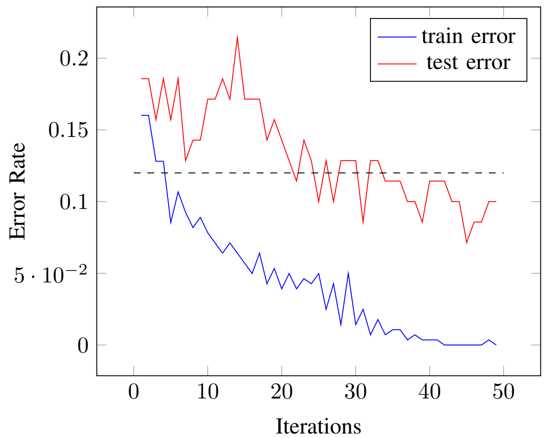

Fig. 5. Deepboost running on Ionosphere with original data.

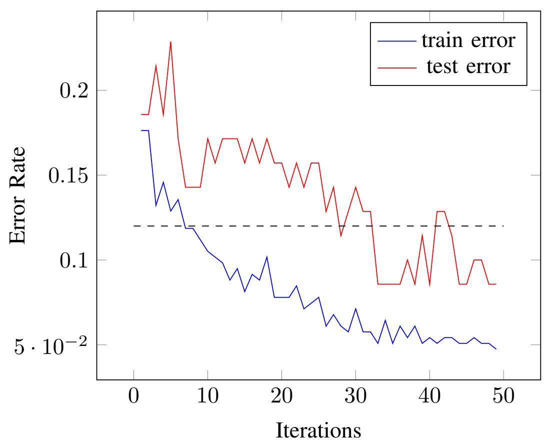

Fig. 6. Deepboost running on Ionosphere with 5% noise.

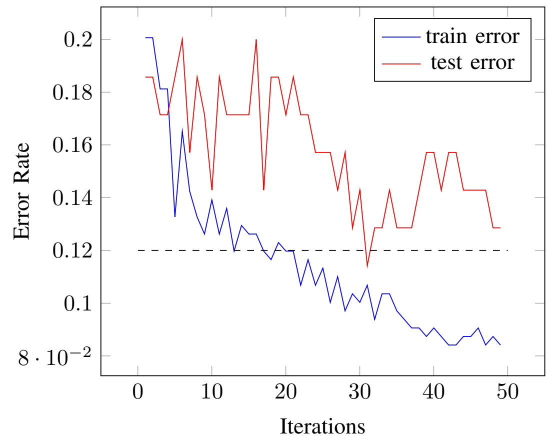

Fig. 7. Deepboost running on Ionosphere with 10% noise.

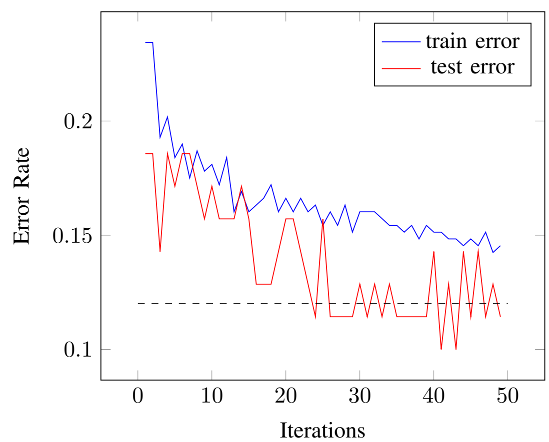

Fig. 8. Deepboost running on Ionosphere with 20% noise.

duced. This is different from Diettrich's work[4], in which noise is added according to the distribution of sample. Our work coinincides with multiple reports that Adaboost has a good performance in general practice.

Our experimental result also shed some new light on analysing the robustness of other algorithms.

VI. Acknowledgments

References

[1] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[2] C. Cortes, M. Mohri, and U. Syed, "Deep boosting," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1179–1187.

[3] J. R. Quinlan, "Bagging, boosting, and c4. 5," in *AAAI/IAAI, Vol. 1*, 1996, pp. 725–730.

[4] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Machine learning*, vol. 40, no. 2, pp. 139–157, 2000.

[5] X. Zhu, X. Wu, and Q. Chen, "Eliminating class noise in large datasets," in *ICML*, vol. 3, 2003, pp. 920–927.

[6] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

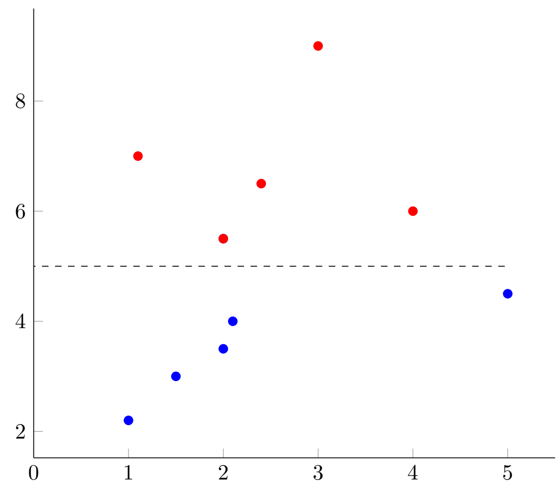

Fig. 9. Drawing noise according to uniform distribution

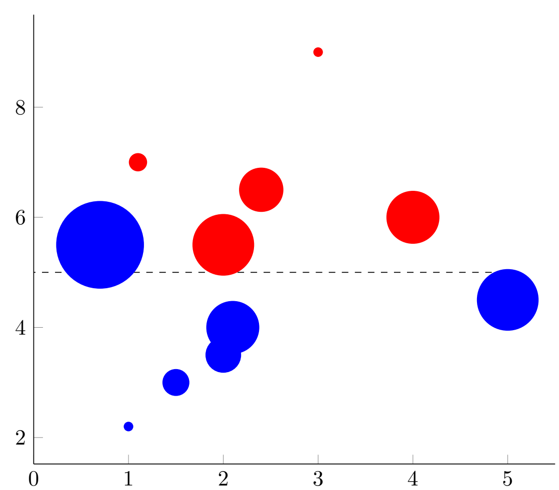

Fig. 10. Drawing noise according to margin distribution