Split Effect in Ensemble

Dongjun Chung[1], Hyunjoong Kim[2]

Abstract

Classification tree is one of the most suitable base learners for ensemble. For past decade, it was found that bagging gives the most accurate prediction when used with unpruned tree and boosting with stump. Researchers have tried to understand the relationship between the size of trees and the accuracy of ensemble. With experiment, it is found that large trees make boosting overfit the dataset and stumps help avoid it. It means that the accuracy of each classifier needs to be sacrificed for better weighting at each iteration. Hence, split effect in boosting can be explained with the trade-off between the accuracy of each classifier and better weighting on the misclassified points. In bagging, combining larger trees give more accurate prediction because bagging does not have such trade-off, thus it is advisable to make each classifier as accurate as possible.

KEY WORDS: Ensemble; Classification tree; Split; Bagging; Boosting

1. Introduction

Ensemble has been known as one of the most powerful classification methods that improve the prediction accuracy dramatically. Breiman (1998) referred ensemble as the "perturb and combine" strategy. Ensemble method generates multiple versions of predictions by perturbing the training dataset and combines these multiple versions of predictions into a single predictor. Bagging (Breiman, 1996) and boosting (AdaBoost.M1; Freund and Schapire, 1996) are most famous and successful ones among those. Bagging perturbs the dataset by using bootstrap then combines the predictions from the perturbed dataset with unweighted majority voting. Boosting increases weight on the misclassified observations through iterations and predicts using the updated weights. It combines these predictions with weighted majority voting by giving more weights on more accurate predictions.

Ensemble method improves the original predictions especially when unstable classifiers are combined such as classification trees. For this reason, many researchers have developed new ensemble methods based on classification trees. The number of splits (the size of tree) is an important parameter to control the accuracy of trees. A single tree is most accurate when it is pruned. However, this is not true in ensemble. Researchers found that combining unpruned trees gives the most accurate predictions in bagging and stump (2-terminal-node

1) M.A. Candidate, Department of Applied Statistics, Yonsei University, 134 Sinchon-dong, Seodaemun-gu, Seoul 120-749, Korea. e-mail: soonceagain@yonsei.ac.kr
2) Assistant Professor, Department of Applied Statistics, Yonsei University, 134 Sinchon-dong, Seodaemun-gu, Seoul 120-749, Korea. e-mail: hkim@yonsei.ac.kr

tree) in boosting.

2. Explanation

2.1. Literature

The relationship between the number of splits and the accuracy of ensemble has been studied especially on boosting. Drucker (2000) claimed that combining unpruned trees gives the most accurate predictions in boosting when using "Z-criterion" to build the trees. Friedman (2001) explained the number of splits as the order of interactions considered in the model. In most practical cases, the main effect and lower interactions usually dominate the model. Hence, he claimed smaller trees perform better in boosting because the model is not influenced by the higher interactions that are not essentially needed in the model. He suggested to use trees with 4-10 splits.

2.2. New Explanation

The success of stump in boosting can be explained by the view of better weighting. As already mentioned, boosting gives more weight on the misclassified observations. The training error rate at each iteration decreases as the number of splits increases (or trees become larger). It means that boosting gives more weight on the smaller set of observations at the next iteration. In later iterations, boosting gives much weight on a few but 'hard to predict' observations. When large trees are used in boosting, this phenomenon becomes more apparent. As a result, a few outliers might dominate the weight used in constructing classification trees. This results in poor performance in each iteration of boosting procedure. On the other hand, the stump would not suffer the above problem.

3. Experiment

To verify our explanation, we experimented on the simulation data. Simulation was iterated 100 times with 500 samples generated for each class. The training dataset and the test dataset were generated independently from the following distribution:

$$Logit(p) = \sin(3x_1 + 5x_2) - x_3 - 2x_4 + 4x_5 + \varepsilon, \quad \varepsilon \sim N(0,5)$$
$$(x_1, x_2, x_3, x_4, x_5)' \sim N_5((0,0,0,0,0)', diag(1,4,7,2,5))$$
$$Y = \begin{cases} 1 & , p \geq 0.5 \\ 0 & , p < 0.5 \end{cases}$$

CART (Breiman et al., 1984) was used for the classification tree algorithm. The number of bootstrap samples in bagging and the number of iterations in boosting were fixed as 50. For implementation, R 2.1.1 was used. CART algorithm was implemented using the function *rpart()* in the library *rpart*. Bagging was programmed based on Breiman (1996)

and boosting based on the algorithm of AdaBoost.M1 in Hastie et al. (2001).

3.1. Measure

Two measures were used: sum of normalized weights on each observation (only for boosting) and test error rate after voting (both for bagging and boosting). Measures are defined as following:

$$\text{test error rate} = \#(\text{misclassification}) \,/\, \text{size of test dataset}$$

$$\begin{matrix}\text{sum of normalized weights} \\ \text{on } k^{th} \text{ observation}\end{matrix} = \sum_{i=1}^{B} \frac{w_{ik}}{S_i},$$

where w_{ij} are the weights on each observation, $S_i = \sum_{j=1}^{N} w_{ij}$, B the number of iterations and $k = 1,2,...,N$. Sum of normalized weights on each observation indicates how many points are overweighted by boosting. Test error rate after voting indicates whether ensemble overfits the dataset or not.

3.2. Split Effect in Bagging

To understand split effect in bagging, three different number of splits were tried: stump, 3-split tree, 5-split tree. Figure 1 indicates that the test error rate after voting decreases as the number of splits increases. We can conclude that combining better classifiers gives better accuracy for bagging.

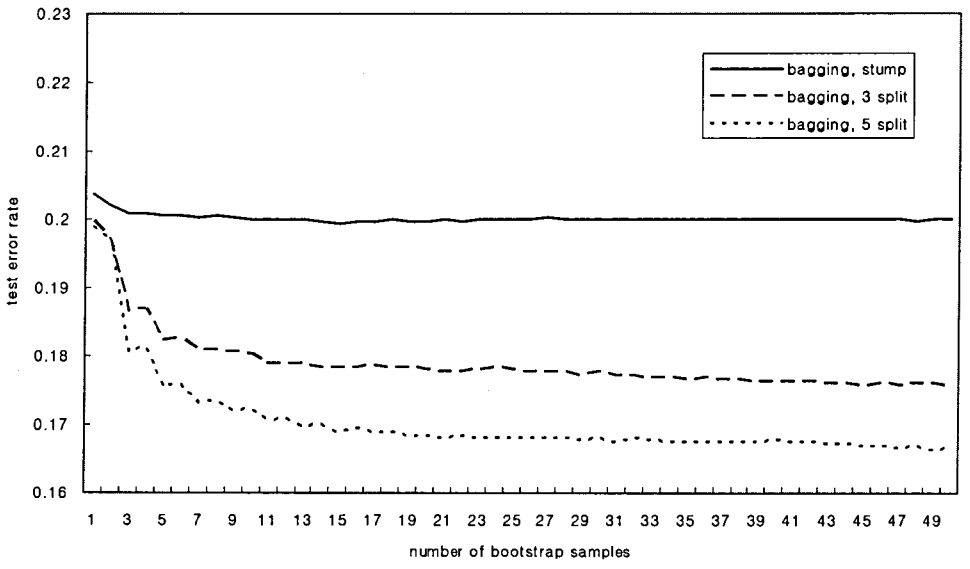

Figure 1. Test error rate of bagging, after combining k bootstrap samples

3.3. Split Effect in Boosting

Three different number of splits are tried: stump, 3-split tree and 5-split tree. The result of figure 2 was obtained from the experiment with slightly different setting from above because observations need to be fixed over iterations. 100 samples were generated for each class, and their weights are updated as the iterations continue.

Figure 2 shows sum of normalized weights of the observations sorted in ascending order. The figure indicates that boosting with 5-split tree gives much more weight on the smaller set of observations than boosting with stump or 3-split tree. Figure 3 shows that the test error rate after voting increases as trees become larger. This is more apparent as the number of iterations becomes larger. These figures indicate that the accuracy suffers when boosting combines large trees. Hence, it can be concluded that avoiding such concentration on a few observations make stump to perform better in boosting.

4. Conclusion

We examined the relationship between the size of trees and the accuracy of ensemble methods. We found that the relationship was positive for bagging, but negative for boosting. Since bagging uses equal weights on the predictions of each iteration, it would be better to predict well on each iteration. However, since boosting updates weights on each observation, it overweights some observations if a large-sized tree is used. Due to the overweight problem, it was found that the stump performed better for boosting.

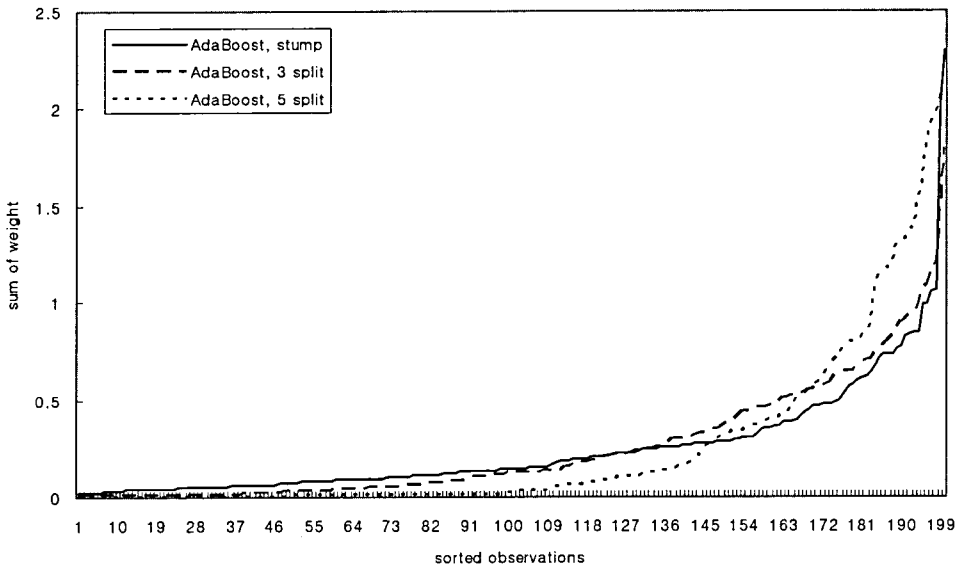

Figure 2. Sum of normalized weights on each observation in boosting

References

Bauer, E. and Kohavi, R. (1999), An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants, Machine Learning, Vol. 36, No. 1-2, pp. 105-139

Breiman, L. (1996), Bagging predictors, Machine Learning, Vol. 26, No. 2, pp. 123-140

Breiman, L. (1998), Arcing classifiers, Annals of Statistics, Vol. 26, pp. 801-824

Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J. (1984), Classification and Regression Trees (CART), Chapman & Hall/CRC, New-York

Drucker, H. (2000), Effect of pruning and early stopping stopping on performance of a boosted ensemble. In Proceedings of the International Meeting on Nonlinear Methods and Data Mining, Rome, Italy, 2000, pp. 26-40

Freund, Y. and Schapire, R. (1996), Experiments with a new boosting algorithm, Machine Learning: Proceedings of the Thirteenth International Conference, pp. 148-156

Friedman, J.H. (2001), Greedy Function Approximation: A Gradient Boosting Machine, Annals of Statistics, Vol. 29, No. 5, pp. 1189-1232

Hastie, T., Tibshirani, R. and Friedman, J.H. (2001), The Element of Statistical Learning - Data Mining, Inference, and Prediction, Springer-Verlag, New-York

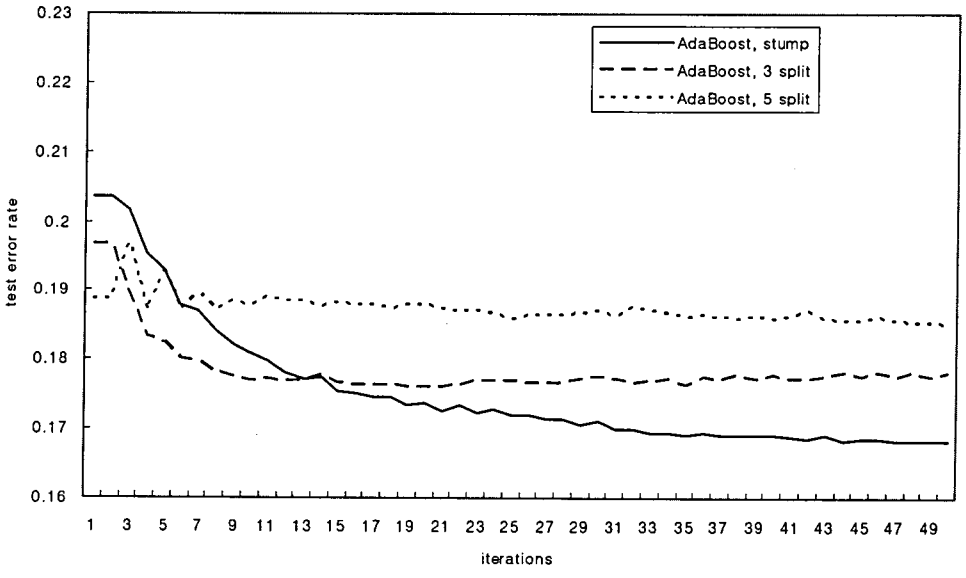

Figure 3. Test error rate of boosting, after combining k iterations