How Weak is a "Weak Learner?"

Neil Leonard nleonard5@wisc.edu

Yewon Lee ylee578@wisc.edu

George gliu84@wisc.edu

Abstract

In ensemble boosting methods, 'weak learners' are used to train each other and boost overall performance of an ensemble. The definition of what a 'weak learner' is is vague and ill-defined. In this paper, the authors set to quantify what this means, both for the overall performance of the ensemble and the effects on bias and variance. Using a Decision tree classifier as a base learner and the standard MNIST data set, these questions are explored. Along the way, a relationship between a models capacity and the effectiveness of ADA-Boosting is found. This was a fruitful exploratory mission, and leads to more questions to be asked.

1. Introduction

By using multiple learners, ensembles of machine learning models can become better that the individual. A practical application of democracy. To expand upon this, ensemble boosting methods will further increase the accuracy.

A popular method of ensemble boosting is ADA-Boosting. This method trains the first learner, and then by using weights, trains the subsequent models to pay more attention to training examples that the previous model missed. Instead of using the same model multiple times, ADA-Boosting takes advantage of the additional capacity that the multiple models could provide. The ensemble then takes a weighted majority vote to make a guess at provided input data. This process is in Figure 1. An additional part of the process is that if a model is trained to a training accuracy of less than random guessing, the training stops and does not use this individual learner.

We will be trying to quantify how the weakness of a learner effects the ADA-Boosting process. This seems to be an open question, so potential results could be novel and (hopefully) applicable. While in theory, we want our models to be perfectly accurate, in some real world applications this might not be possible. The amount of data in our world is exploding, and boosting even a weak algorithm could help us sift through it all.

Algorithm 2 *SAMME*

1. *Initialize the observation weights* $w_i = 1/n,\ i = 1, 2, \ldots, n.$
2. *For* $m = 1$ *to* M*:*
   (a) *Fit a classifier* $T^{(m)}(\boldsymbol{x})$ *to the training data using weights* w_i*.*
   (b) *Compute*
   $$err^{(m)} = \sum_{i=1}^{n} w_i \mathbb{I}\left(c_i \neq T^{(m)}(\boldsymbol{x}_i)\right) / \sum_{i=1}^{n} w_i.$$
   (c) *Compute*
   $$\alpha^{(m)} = \log \frac{1 - err^{(m)}}{err^{(m)}} + \log(K - 1).$$
   (d) *Set*
   $$w_i \leftarrow w_i \cdot \exp\left(\alpha^{(m)} \cdot \mathbb{I}\left(c_i \neq T^{(m)}(\boldsymbol{x}_i)\right)\right),\ i = 1, \ldots, n.$$
   (e) *Re-normalize* w_i*.*
3. *Output*
   $$C(\boldsymbol{x}) = \arg\max_k \sum_{m=1}^{M} \alpha^{(m)} \cdot \mathbb{I}(T^{(m)}(\boldsymbol{x}) = k).$$

Figure 1. The ADA-Boosting Algorithm as described by Multi-Class AdaBoost paper [3]

2. Related Work

When researching this paper, two main sources on ADA-Boosting were used: Professor Raschka's notes [4] and the paper MultiClass AdaBoost [3]. This project was picked because of its novelty, so there is not much related work that the authors are aware of.

3. Proposed Method

To quantify the 'weak learners' in Ada-boosting, we will sweep through hyper parameters of a Decision Tree Classifier. The hyper parameters we will be focusing on are ensemble size (the number of weak learners) and the depth of the Decision Tree. Using these hyper-parameters, analysis will be done on over-fitting, the breaking point of the ensemble method, and the over-all effectiveness of the Ada-boosting upon the algorithm.

The motivation for sweeping across the tree depth is to control the base accuracy of the individual models. For decision trees, controlling the depth was the hyper-parameter that worked to best to control the base accuracy when compared to training size, learning rate, etc. It is important to note that there is not a completely direct relationship between tree depth and accuracy rate, and could have some non-linear effects

Figure 2. samples from the MNIST data set [1]

4. Experiments

Using the above proposed method, we will be doing analysis on on these properties:
- Over-fitting/Variance of Ada-Boosting
- The breaking point of the ensemble
- Over-all effectiveness of Ada-Boosting with 'weak learners

To understand over-fitting, the difference between the ensembles training and test accuracies will be plotted against base accuracy and ensemble size. For the 'breaking point' of the ensemble, the number of models that it took the training to get below random guessing will be plotted against the base accuracy of the ensemble. Finally, for the over-all effectiveness of ADA-Boosting, the ensembles accuracy will be plotted against our two main hyper parameters: base accuracy and ensemble size

4.1. Dataset

Here we will be using a classic standard of machine learning: the MNIST[2] data set. When doing question based investigation, using a known data set is important to understanding the how generalizable the forthcoming analysis is. Additionally, the MNIST data set was the right level of complexity for our proposed method, as individual learners to not reach the capacity of the data set too quickly. When implementing the MNIST set, analysis must be modeled on 10 potential distinct answers as opposed to the more simple binary situation

4.2. Software

The Scikit-learn machine learning library was used in the implementation, as well as Numpy and Matplotlib for analysis.

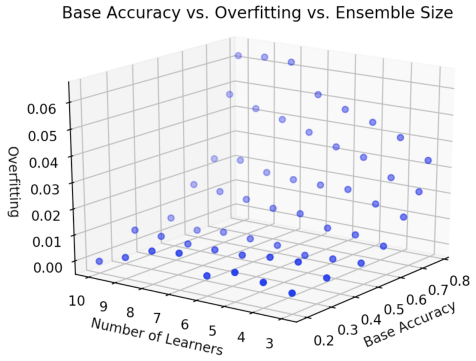

Figure 3. over-fitting vs Ensemble size and base learning rate

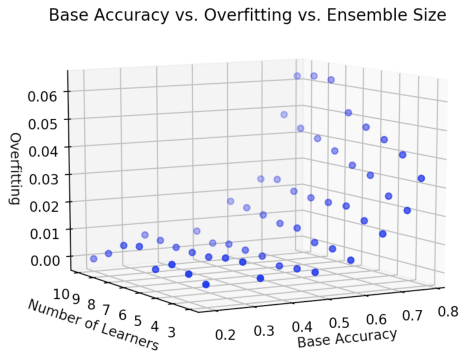

Figure 4. over-fitting vs Ensemble size and base learning rate, another view

5. Variance in Ada-Boosting

When selecting an algorithm for a machine learning task, it is vital to understand how the algorithm over-fits to training data, and thus under performs on testing data. In this pursuit, the difference in training and training accuracy versus the hyper-parameters (ensemble size and base accuracy) as described above

5.1. Variance vs base accuracy

When looking at over-fitting of Ada-Boosting, figures 3 and 4 show a clear trend of higher over-fitting of stronger learners. Once the base learning rate becomes low enough, we see no over-fitting happening in the boosting process. This may be counter intuitive, as the 'weak learner' does better in this case. This indicates that over-fitting can only happen if the model is relatively fit.

5.2. Variance vs ensemble size

While over-fitting may more strongly correlated to base accuracy, there is a clear trend towards higher variance in

larger ensembles. The highest variance is seen in the in the largest ensemble with the largest base accuracy, but if the base accuracy is lowered it trends towards zero. Indicating that base accuracy is the controlling factor

6. Breaking an Ensemble (So you think your Tough, huh?)

When applying the ADA-Boosting algorithm, it is stopped when an individual model goes below the threshold of a random learner, 10 percent in the case of MNIST data. To ascertain when our algorithm meets this threshold, we created a large ensemble and fit them until a learner went below ten percent. This is shown in figure 5.

With a higher base accuracy, it has a longer height to fall to get below random guessing. As figure 5 shows, with higher base accuracy it takes longer to break the ensemble. As such, it takes more iterations to break the ensemble. Our analysis only plots when weaker learners break, so there could be interesting effects at higher limits that we are not seeing. Unfortunately, without using parallel processing and investing in some serious server space, analyzing these higher limits is not really feasible in our implementation. As the base accuracy goes up, the size of the ensembles and number of iterations greatly increases, and so does the computation time. Luckily, any real world application of Ada-Boosting will be in the threshold that we have plotted, as ensemble sizes in the hundreds is unnecessary.

7. Ability of a 'Weak Learner'

Probably the most basic question to ask when quantifying a 'weak learner', is how does the 'weakness' of the learner affect the over all performance of the Ada-Boosting algorithm. As described in Section 4, we will be sweeping through the hyper-parameters ensemble size and base accuracy and seeing how this affects the performance of the boosted ensemble. But first, a baseline to compare this against must be established

7.1. Defining a metric

By taking multiple learners and using a majority vote for classification, the ensemble will be more accurate than the individual learner. Calculating this for a binary problem is a relatively simple combinatorics question, but because we are using MNIST, which has 10 potential answers, this becomes a bit trickier. Now it turns out that this is an interesting generalization of the famous Birthday problem, but we will not bother our math graduate student friends to solve this analytically. We are computer scientist, so we will use a simple Monte-Carlo simulation.

By using random data and simple correlation functions, the accuracy rate of ensembles for 10-fold potential answers was graphed against base accuracy of the learners (fig 6).

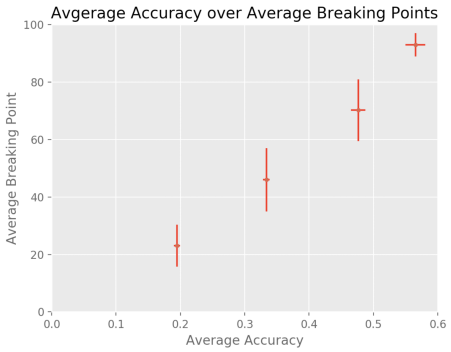

Figure 5. The number of ensembles it took to break the boosting algorithm vs base accuracy of the learners.

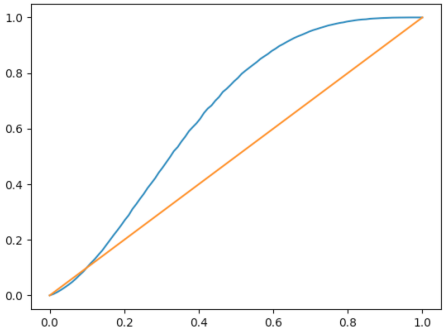

Figure 6. ensemble accuracy vs base learner accuracy from the Monte Carlo

By increasing the size of the random data set, the figure above was pushed close to a theoretical analytical limit. While the mathematics may be beyond the scope of this paper to prove this, the sample size was increased to a high enough value to give the authors confidence that this an extremely accurate approximation. This is bar that a boosted ensemble system should be above, if the learners are truly independent. But more about this subtly later.

7.2. Ada-Boosting in smaller ensembles

To demonstrate the effectiveness of Ada-Boosting on smaller ensembles, the over all accuracy of the ensemble plotted against the base accuracy of learners for ensembles sizes, as seen in figure. The result from the Monte-Carlo (Figure 6) as have added to compare against. Each point represents a different depth size, with the farthest left point being depth one, all the way up to depth of 12.

7.3. How weakness effects Ada-Boosting

For all ensembles, there is a clear conclusion to be drawn from the figures: weaker learners make better use of Ada-boosting. Regardless of ensemble size, the weaker

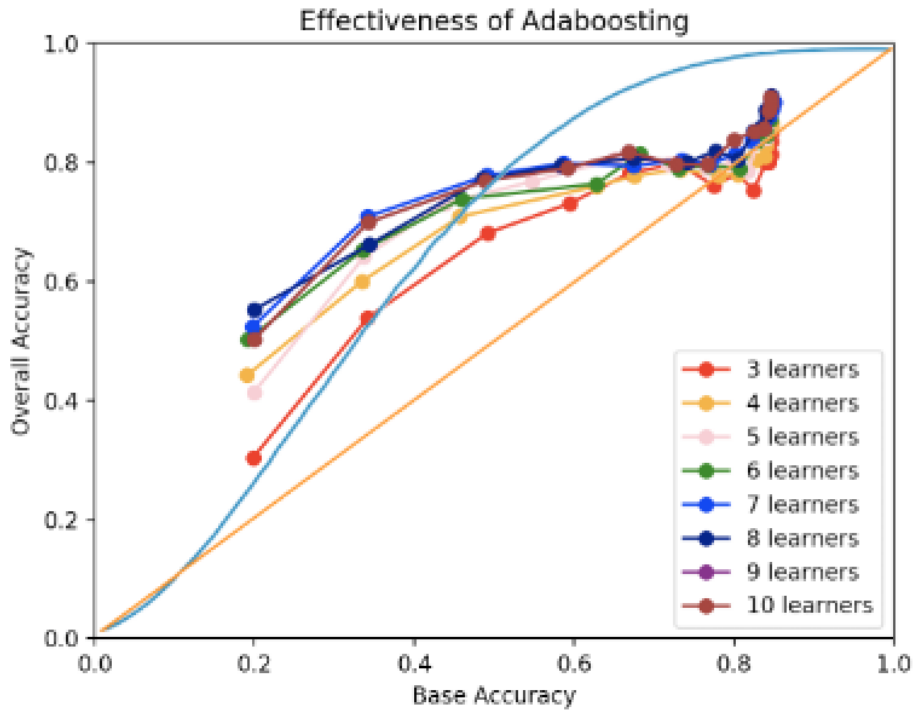

Figure 7. Overall accuracy vs base accuracy for smaller ensembles

learns all outperformed the Monte-Carlo, with the higher ensembles being more effective. As the ensembles come to about 50 percent base accuracy, they start to not outperform the Monte-Carlo. While this seems counter intuitive, this makes sense as our learners are not independent by design. This will be explored in the conclusion.

7.4. Ensemble size and Ada-Boosting Effectiveness

Similar to above, the greatest effect that ensemble size had on the overall effectiveness is at the lower base accuracy. The smaller ensembles do worse at lower values, but as they reach there max base accuracy (about .8 in our case) all the ensembles effectiveness converge, with only minor differences between them.

7.5. Ada-Boosting on larger ensembles

In addition the lower ensemble size, larger ensemble sizes were effects on ADA-Boosting was looked at. In Figure 8, we can see a similar trend to the lower ensemble numbers, but an even more dramatic increase to weak learners ensemble boosting. Additionally, the plateau of the 'moderate learners' is present, but an overall increase to these and the strongest learners. Between the two, 32 and 64 learners in the ensemble, there was little difference, indicating

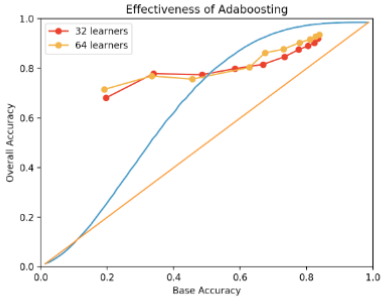

Figure 8. ensemble accuracy vs base learner accuracy from the Monte Carlo

that we had reach the threshold for where ensemble size increases help the overall algorithm

7.6. "The Tail" Effect

Possible the most interesting aspect of our figures, is the upswing in the ensembles accuracy at the right end of the graphs. As the depth of the Decision tree increase, the indi-

vidual accuracy stays about the same, while the ensembles accuracy goes up. This implies that while the increased capacity in the individual models does not effect their accuracy, the Ada-Boosting can take advantage of this and help the whole ensemble. This is not something we expected, and is a delightful surprise.

8. Conclusions

Asking 'how weak is a "Weak Learner"?' is an open ended question. It is a research based question, so this paper is more of an exploration, without a clear conclusion for the beginning. Luckily, we encountered some interesting phenomena, and these are the ones we found most important.

8.1. Independence and it's role in Ada-Boosting

One of the more nuanced, but vitally important, issues in Machine Learning is the independence of the data and models. When simulating the Monte-Carlo, the learners were completely independent, so that the effect of using an ensemble was maximized. In the real world application, this could never be the case. So even if a boosted ensemble performs less than this theoretical benchmark, it still can be boosting the ensemble.

While one usual tries to make maximize independence in a machine learning algorithm, Ada-Boosting actually presupposes relationships between these models. By subsequent models learning from the failures of previous models, there is an designed dependency between these individual learners. This is an inherent tension, and probably can not be resolved. Looking at it optimistically though, true independence is almost impossible, so using algorithms like Ada-Boosting can take advantage of this insoluble flaw.

8.2. Weak Learners with ADA-Boosting

In general, our results indicated that this boosting method works better for weaker learners. These learners saw a greater boost than just an ensemble would provide, while having a lower variance! These effects are encouraging to potential practitioners of machine learning, as if your particular problem has training issues, like a lack of training data, could use ADA-Boosting to great effect.

While we had great effects for weaker learners, moderate learners did not seem to gleam such an increase. Implying that applications that use base models with similar accuracies may not get as much from ADA-Boosting. Other boosting methods may work better.

For the stronger learners, the conclusion is less clear but more exciting. Overall, stronger learners saw some increase by using ADA-Boosting. Additionally it showed that in certain circumstances, ADA-Boosting can be more effective when increasing the capacity of the ensemble, even if this does not increase the base accuracy. This indicates that certain models and applications may be interested in using ADA-Boosting.

9. Further Questions/Thoughts

9.1. Weakness of learners and other machine learning models

The results that we gained from our experiments was strong and interesting. But before generalizing our results, it seems pertinent to try this on other models. To control base level accuracy, the depth of the Decision Tree Classifier was changed. Obviously this is not a parameter that we can extend to other models. Even in our analysis, we saw that there was a non-linear relationship between these two variables. How ADA-Boosting effects the overall gain in performance seems to be tied to how it takes advantage of the increased capacity of the ensemble. Each machine learning model has an idiosyncratic capacity, so it follows that the results may be different.

9.2. Should we move the breaking threshold

An interesting implication of of this interaction between Ada-Boosting and independence is that maybe we should ask the question again: where should the breaking threshold be? Being better than random guessing seems like the obvious answer, but this is only clearly true if the models are independent. Even if a model is a little below random guessing, could it be specialized enough to actually provide more capacity than being just under random guessing took away? This could be possible, but the minuscule size of the difference in capacity is likely so small that its not discernible. The sheer size of the ensemble at that point would wash out the effect from something like this

9.3. Perhaps a new base line

In this paper, we used a Monte Calro to provide a baseline of what a non-boosted ensemble would be, and expected boosting algorithms to out perform this. As was clearly shown, boosted algorithms under-performed from this benchmark at higher base accuracy rates. The reason for this, as discussed, was the lack of independence in the algorithm. Perhaps we should have expected this, but we did not. In hindsight and in future explorations, a better baseline would perhaps be non-boosted ensembles for our model. Then our results would be clearer and more generalizable.

10. Acknowledgements

Special Acknowledgements to Ibrahim Safa and Jeff Lazar, Physics graduate students who helped with the Monte-Carlo and general python skills!

11. Contributions

Neil Leonard: Implementation Design, writing, analysis Yewon Lee: Data processing, Graph creation, slideshow presentation

References

[1] https://www.researchgate.net/figure/ten-samples-for-each-digit-in-the-mnist-dataset$_fig5_322079459$.

[2] http://yann.lecun.com/exdb/mnist/. Mnist data set.

[3] H. Z. T. H. Ji Zhu, Saharon Rosset. Mutliclass adaboosting.

[4] S. Raschka. Stats479. 2019.