# Creating an ADABoosting Model from Scratch (problem 3)

Mohamad Lakkis

May, 2024

This PDF is to explain some of the coding part in the *Creating an ADABoosting Model from Scratch (problem 3*

## Algorithm Used

**Algorithm 1** AdaBoost.M1 using Trees

1. Initialize the observation weights $w_i = \frac{1}{N}$, $i = 1, 2, \ldots, N$.
2. For $m = 1$ to $M$:
   (a) Fit a tree classifier with max-depth = 1(i.e. a stump)$G_m(x)$ to the training data using weights $w_i$.
   (b) Compute
   $$\text{err}_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i}.$$
   (c) Compute $\alpha_m = \log\left(\frac{1-\text{err}_m}{\text{err}_m}\right)$.
   (d) Set $w_i \leftarrow w_i \cdot \exp(\alpha_m \cdot I(y_i \neq G_m(x_i)))$, $i = 1, 2, \ldots, N$.
3. Output $G(x) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m G_m(x)\right)$.

## Code Explanation

- So using Algorithm 1 we run this process for different $M$, and for each M we predict the test data and calculate the test error, and then calculate the training error.

- There is two versions of the code on that is parallelized and the other is not. The parallelized version is faster and more efficient when used in HPC.
- Now in order to get the accuracy of a 244 node tree, we varied the max leaf nodes parameter over a range of values, and then for each value calculate the number of nodes in the tree until we reach the exact value of 244, or something very close to it.

# Analysis of Figure 10.2

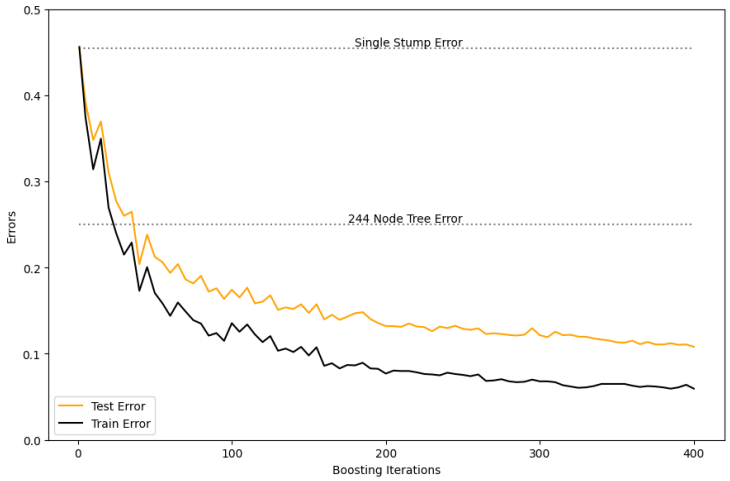


Figure 1: Fig 10.2: Errors vs Boosting Iterations

- We can see that both errors decreases up to a certain point, and then become somewhat constant. For the Testing Error it becomes somewhat constant around 0.13 after boosting iterations reaches 200 after which we don't see a significant impact on the error when we increase the boosting iterations. Similar to the train error but the train error is lower than the test error (as expected), where we see that if we keep on training the model the train error will keep on decreasing(in a slower rate than before but it will keep on decreasing). But as of the test error that is not always the case because as we increase the number of boosting iterations the model will start to overfit the data and the test error will start to increase. So now in order to explore this we need to run on more boosting iterations to see when it starts to increase the test error ( this is to answer C)(please refer to the second graph where we can only see the test error and the number of boosting iterations)

- Please note that we can see some small flunctuations in both error this is totally normal and it is due to the randomness of the data and the model.
- The decrease in error rates suggests that the ensemble model is able to generalize well from the training data to unseen test data, largely because the clear separation between classes allows each weak learner to effectively contribute to reducing overall misclassification.

## Analysis of Test Error vs Boosting Iterations

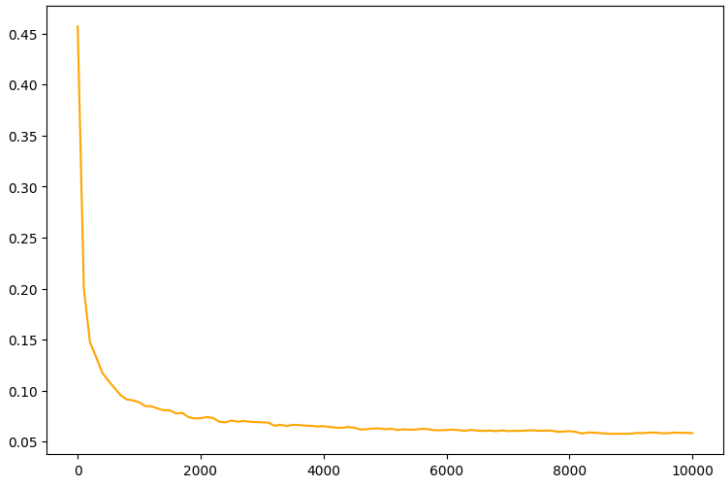


Figure 2: Test Error VS number of boosting iterations

- We should expect that the test error to start increasing especially after a certain point, but we can see that the test error is still decreasing. This is due to the fact that the classes are well separated and the model is able to generalize well from the training data to unseen test data. ( but as the question was phrased we know that the test error should start increasing but I don't know why this is not the case here )

## Case where classes have significant overlap in feature space(Part d)

- In this case, we can see that the test error is higher than the previous case where the classes were well separated. This is due to the fact that the model is not able to distinguish between the two classes because they are overlapping. So the model will keep on training and trying to minimize the error but it will not be able to do so because the classes are overlapping.

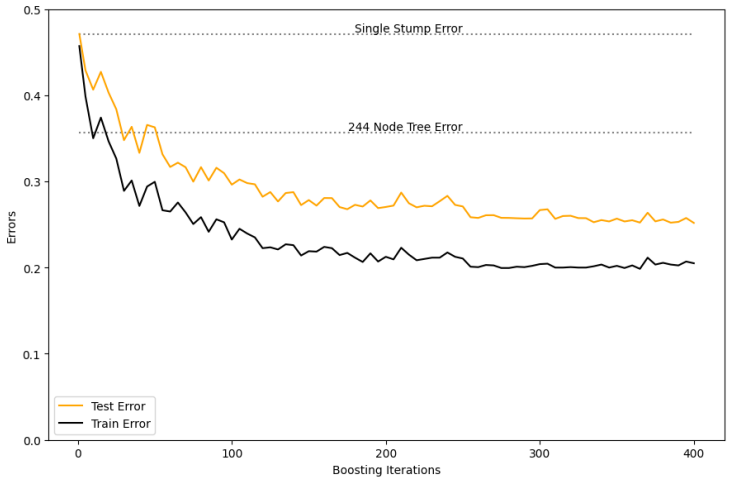


Figure 3: Fig 10.2 where classes have significant overlap in feature space

- The test error starts to flatten out and remains higher compared to the scenario without overlap. This indicates that the test error does not improve significantly beyond a certain point.
- We should expect as before that if we keep on training the model on more number of boosting iterations the test error will start to increase because the model will start to overfit the data on the training points, and since in this case we have more overlapping we expect this increase to be earlier than the one before, and more extreme(in a general sense).