Data Mining Project 3: Adaboost using Decision Stumps as Weak Learners

Lalithanjana Kollipara (G01386376)
Mail ID: lkollipa@gmu.edu
Miner username: nocturnal

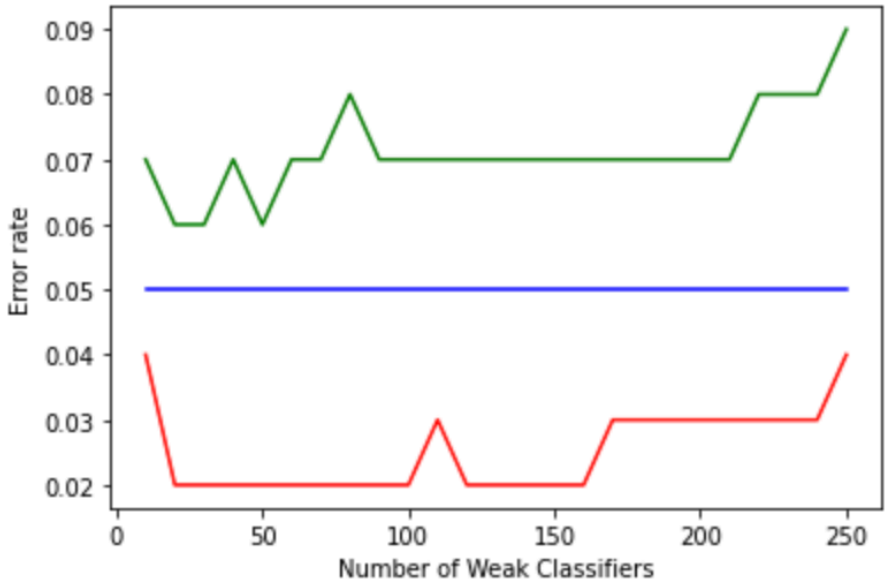
Aim:
Implement the Adaboost algorithm and a Gini-index based method for learning decision stumps. Better understand the relationship between the number of rounds of boosting, training error, and test error.

Observations:

Number of Weak Classifiers	Train Error	Estimated Test Error
10	0.04	0.07
50	0.02	0.06
100	0.02	0.07
150	0.02	0.07
200	0.03	0.07
250	0.04	0.09

- The table shows the training error and estimated test error for a boosting model with varying numbers of weak classifiers. Overfitting occurs when the model performs well on the training data but poorly on unseen test data.
- The boosting algorithm starts with 10 weak classifiers and gradually increases the number of weak classifiers up to 250.
- As the number of weak classifiers increases, the training error generally decreases, which indicates that the algorithm is becoming more accurate in classifying the training data.
- However, the estimated test error does not decrease monotonically with the number of weak classifiers, which suggests that overfitting may be occurring.
- The estimated test error reaches a minimum around 50 weak classifiers and then increases slightly as the number of weak classifiers continues to increase.
- This behaviour is consistent with overfitting, where the algorithm becomes too specialized to the training data and loses generalization ability to new, unseen data.
- The best trade-off between the training and test error is achieved with 50 weak classifiers, where the test error is estimated to be 0.06.
- The performance of the algorithm starts to degrade beyond 50 weak classifiers, with the test error reaching 0.09 at 250 weak classifiers.
- In conclusion, the table illustrates the importance of selecting an appropriate number of weak classifiers to balance training accuracy and generalization ability.
- We also observe that the train error of the model is approximately 0.05 when ran using the DecisionTreeClassifier from the sklearn library with the criterion as ‘gini’, random_state = 42 and a max_depth = 10. These attributes were chosen because when used, the model produced the highest accuracy.

Graph Analysis:



The graph shows the behaviour of a model regarding overfitting.

- The x-axis represents the number of weak classifiers used to train and test the model, the y-axis represents different error rates of the model.
- There are three curves on the graph, each representing a different error rate: training error, test error and the error when the model is trained using the DecisionTreeClassifier from the sklearn library.
- The training error (in red) starts high and decreases as the number of weak classifiers increases. This is because the model is fitting more and more to the training data, becoming more and more accurate in predicting the training data.
- The test error (in green) starts decreasing as well, but after some point, starts increasing again, indicating overfitting.
- The overfitting starts around 70 rounds, as the test error starts to increase while the training error keeps decreasing.
- The gap between the training error and test error is small at the beginning, indicating that the model is not overfitting. As the number of weak classifiers increases, the gap between the training error and test error starts to increase, indicating overfitting,
- Overfitting can be averted by stopping the training at a point where the test error is still decreasing.
- The blue line on the graph represents the training error of the model when it is trained using the DecisionTreeClassifier from the sklearn library.

Conclusion:

In conclusion, implementing AdaBoost using weak classifiers is an effective technique for improving classification accuracy. It combines the strengths of multiple weak classifiers to create a strong ensemble model. We can also conclude that we get the best accuracy (approximately 0.95) when the model is trained using the classifier from the sklearn library compared to that of our own.