Ensemble Learning

Students Adam Vinestock, Idan Cohen

Introduction

In this report, we present our application of Ensemble Learning methods - Gradient Boosting Regression Trees (GBRT) and AdaBoost. The report is organized such that in each assignment part we will explain our approach and findings.

Part 0: Generate datasets

In this section, we generated two distinct datasets, each possessing unique characteristics. Both datasets consist of 1000 samples, with binary balanced labels. Notably, these datasets exhibit a dependency between the features and the labels, albeit with the presence of introduced noise. Consequently, the datasets are inherently non-linearly separable within their original feature space, thereby posing challenges for linear classifiers. Dataset1 is characterized by the inclusion of a redundant feature, namely feature 3, which demonstrates a high correlation with feature 1. This redundancy introduces intricacies in the dataset, necessitating careful consideration of the interplay between the features when establishing relationships with the labels. On the other hand, Dataset2 encompasses two features without any redundancy. The labels assigned are influenced by the distances of these samples from the origin. Notably, the labels exhibit a discernible pattern based on the proximity of the samples to specified inner and outer negative radii. As such, Dataset2 enables the exploration of classification techniques in a setting where the relationship between the features and labels is characterized by geometric considerations. By studying these datasets, we can gain insights into the challenges posed by non-linearity and the impact of feature redundancy on classification performance.

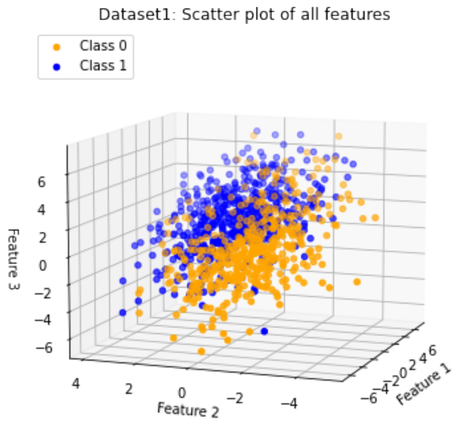


Figure 1: Dataset1

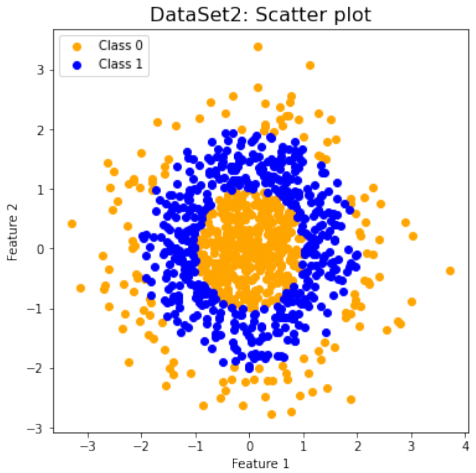


Figure 2: Dataset2

Part 1: Gradient Boosting Regression Trees (GBRT

We presented accuracy over both datasets and conducted experiments over Dataset2 using various settings to examine the effects of the following parameters: regression tree max depth (max_depth), number of weak estimators ($n\_estimators$), and the learning rate parameter ($alpha$).

Overall, the worst results were obtained with ($max\_depth = 1, n\_estimators = 100, alpha = 0.01$) with an accuracy of 0.66. On the other hand, the best results were achieved with the parameters $max\_depth = 3$, $n\_estimators = 100$, and $alpha = 0.1$. Notably, this parameter combination aligns with the default settings in the scikit-learn implementation of this algorithm.

Here are a few insights from this study:

1. Using decision stumps (weak learners with a maximum depth of 1) with $alpha = 0.01, n\_estimators = 100$ yielded

the lowest accuracy of 0.66 on the test set. However, when we increased *alpha* to 0.1 the accuracy significantly improved to 0.9. This suggests that decision stumps are able to perform well on this task, but a low learning rate slows down the model's learning process.

2. Our experiments with deep trees, specifically $max\_depth = 5$ and $max\_depth = 6$ achieved relatively high performance, with accuracy of 0.92 and 0.93 respectively. It is often assumed that deep trees lead to overfitting. However, in this fabricated data-set, which doesn't much complexities, we observed these impressive results.

3. The experiment with $max\_depth = 6$, $n\_estimators = 1$ and $alpha = 0.01$ achieved a test accuracy of 0.91. This serves as additional evidence for our insight from section 2, that this dataset is relatively simple, without many interactions and complexities. Thus, classical decision tree algorithms can perform well on this specific dataset, without the need to combine multiple trees into an ensemble model.

4. Examining the experiments with $max\_depth = 2$ reveals that with a single estimator, i.e $n\_estimators = 1$, we obtained an accuracy of 0.71. Then, as we increased the number of estimators up to 200, the accuracy also increased, reaching 0.926 (there is also an experiment with $n\_estimators = 500$, but without an accuracy improvement). Thus, the complexities that a single tree with $max\_depth = 6$ was able to understand, also the ensemble was able to understand, but it needs more than 100 estimators for that.

5. Based on the previous insights from sections 3 and 4, we propose a research question: Is there a benefit in constructing an ensemble consisting of both deep and shallow trees? This investigation could be motivated by considerations of accuracy or running time. Specifically, we are interested in examining whether certain complexities, which the ensemble encountered during the construction of H^*, can be better addressed by including a deep tree in the ensemble compared to an ensemble of shallow trees. It is important to note that our suggestion refers to a hybrid-depth ensemble model, where both deep and shallow trees are combined, rather than favoring single deep trees over ensembles[1].

Part 2: AdaBoost

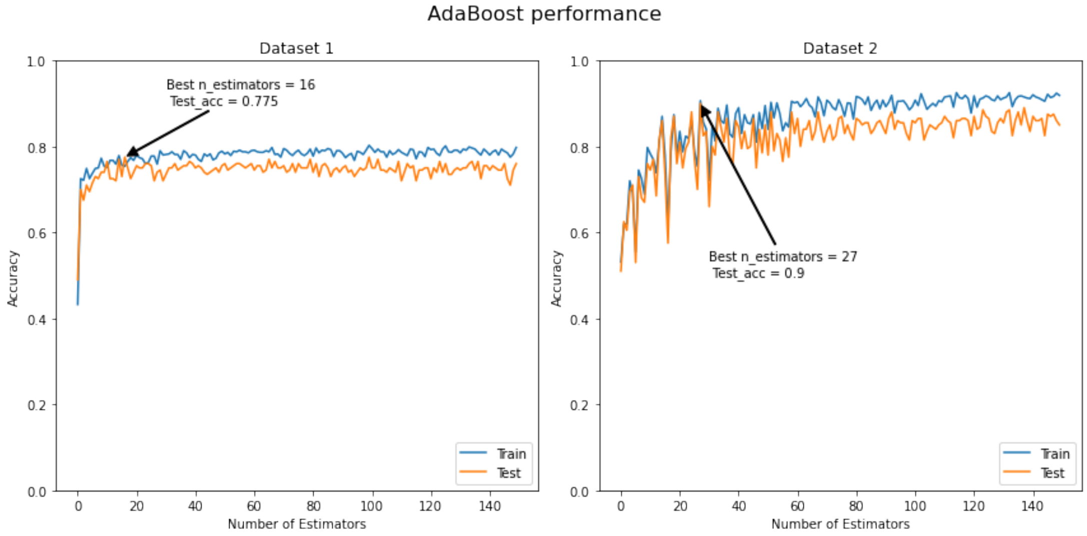
The AdaBoost classifier was applied to both datasets and yielded results that were in line with our expectations. Despite the use of weak classifiers (single depth decision stumps), the performance exhibited considerable effectiveness. It is worth noting that both datasets are characterized by noise and lack linear separability in their original feature space, posing challenges for linear classifiers.

For Dataset 1, the labels are solely dependent on features 1 and 2. By applying two simple constraints, a test accuracy of 0.685 was achieved:

$$\left\{ \begin{array}{ll} 1 & \forall i \in [0, \text{Samplespace size}],\ \text{if } X[i,0] > 0 \vee X[i,1] > 0 \\ -1 & \text{otherwise} \end{array} \right\}$$

Based on our expectations, an increase in the number of estimators beyond 2 does not significantly improve performance. This observation is visually evident in the provided plot, where the accuracy plateaus after reaching a number of estimators greater than 2. The best accuracy falling in the range of $[2, 150]$ estimators is subject to stochastic noise.

For Dataset 2, the labels exhibit a more complex dependence on both features. As the number of estimators increases, performance improves up to a threshold of approximately 30. Beyond this threshold, the improvement in performance becomes marginal. This indicates that indiscriminately increasing the number of estimators does not always lead to a significant accuracy gain.



[1] From a brief literature review, we couldn't find relevant works which address this topic