Home Assignment 2 – Ensemble Learning

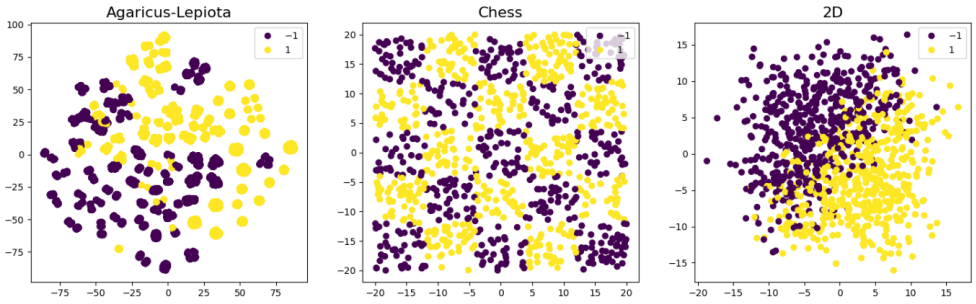
May 14, 2023

1 Generate Datasets

Three datasets were generated: **Agaricus-Lepiota:** A known binary dataset used to differentiate between two type of mushrooms. It comprised 8124 samples, each with 22 discrete features. We used label encoder to transform the samples values into numerical values. **Chess:** Using the sin function we generated a chess-like dataset consist of 1000 labeled samples, each sample has two features. **2D:** This dataset was populated with 1000 random 2D samples from a uniform distribution. Noise was added to disrupt the linear separation.

All the datasets were split into training and test sets with a 4:1 split ratio. The labels are 1, −1. The datasets are shown in figure 1. In order to visualise the Agaricus-Lepiota dataset, we used T-SNE to reduce dimensionality.

Figure 1: Datasets



2 Gradient Boosting Regression Trees (GBRT)

In this part we implemented Gradient Boosting algorithm using Regression Trees. During the learning phase, we tracked the training and test loss and error. We used 100 tree estimators, with max_depth of 3 and $\alpha = 0.1$. A summary table of the accuracy results is shown on figure 2.

Figure 2: GBRT - Accuracy Results

GBRT		
	Train	Test
Agaricus-Lepiota	100.00%	100.00%
Agaricus-Lepiota - 2D	99.08%	98.15%
Chess	98.60%	87.50%
2D	96.00%	80.50%

2.1 Experiments

Accuracy as a function of Max depth: We run experiments with different Max_depth values, with Max_depth $\in \{1..15\}$. The results show that, for all dataset, from a certain value of max_depth the accuracy doesn't improve and in same cases decreases. The accuracy was optimal when Max depth=4.
Accuracy as a function of n_trees: We run experiments with different number of estimators, with n_Trees $\in \{1..100\}$. The results show that, the ideal number of trees needed to achieve the highest test accuracy was around 70, 25, 5 for the Agaricus-Lepiota, Chess and 2D datasets respectively.
Convergence as a function of Alpha: We run experiments with different α values, with $\alpha \in \{0.001, 0.01, 0.1, 0.5\}$. As expected, the highest the value of α the quicker the model converged. However, with $\alpha = 0.1$ the model produced good results with less over-fitting.

3 AdaBoost

In this part we implemented AdaBoost algorithm using decision Tree classifiers. During the learning phase, we tracked the training and test loss and error. We used 100 tree estimators and a summary table of the accuracy results is shown on figure 3. Also, We plotted the training error against the loss curve to visually see that the training error is upper bounded by the loss.

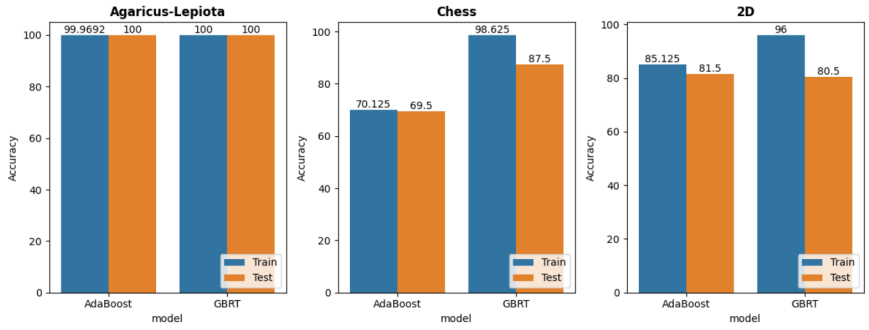
Figure 3: AdaBoost - Accuracy Results

AdaBoost		
	Train	**Test**
Agaricus-Lepiota	99.97%	100.00%
Agaricus-Lepiota - 2D	82.24%	81.66%
Chess	70.12%	69.50%
2D	85.12%	81.50%

3.1 Experiments

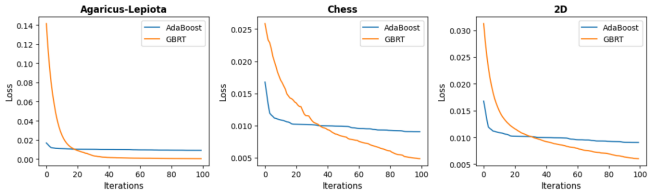
Accuracy as a function of n_trees: We run experiments with different number of estimators, with n_Trees $\in \{1..100\}$. The results show that, the ideal number of trees needed to achieve the highest test accuracy was around 70, 30, 10 for the Agaricus-Lepiota, Chess and 2D datasets respectively.
AdaBoost VS GBRT: In this part we compared the results achieved by the two models. We compared the training and test accuracy across all datasets, which are shown in figure 4.

Figure 4: AdaBoost VS GBRT - Accuracy



We also compared the Convergence rate across all datasets, The loss values were normalized, it can be noticed that AdaBoost converges faster than GBRT. the results are shown in figure 5.

Figure 5: AdaBoost VS GBRT - Convergence



4 Summary

In the assignment we implemented, experimented and compared two gradient boosting algorithms, GBRT and AdaBoost. We noticed that although GBRT produces better training accuracy results, it accompanies with high over-fitting compare to AdaBoost. AdaBoost adaptively allocates different weight to each estimator. At the beginning of learning phase the weights are larger while gradually getting smaller for newly added estimators. As a results, the algorithm manages to avert high variance and able to converge quicker than GBRT.