Proseminar
Advanced topics in machine learning

Bagging, Boosting, and Ensemble Learning

Author

Florian Zager

Matrikelnummer: *2489018*

Date:

Karlsruhe, January 21, 2024

Abstract

Ensemble Learning is a very powerful machine learning method, which combines multiple smaller learning algorithms. Previous research has shown that ensembles often achieve higher predictive accuracy than an individual learning algorithm in the ensemble. In this report, we explain the Ensemble Learning methods Bagging, Boosting, Random Forests, and Gradient Boosting, and compare them against Decision Trees. Our evaluation, demonstrates a significant improvement in predictive accuracy with Ensemble Learning methods. The results indicate that methods like Bagging, Boosting, Random Forests, and Gradient Boosting consistently outperform Decision Trees. These findings underscore the potential of Ensemble Learning in advancing the field of machine learning, offering valuable insights for data scientists in model selection and application.

Contents

1 Introduction

Machine learning is becoming increasingly pivotal in a variety of critical applications. Its usage extends to healthcare tasks such as detecting cancer and predicting diabetes outbreaks, underscoring its importance in public health and safety. Additionally, machine learning plays a significant role in environmental sustainability, for example in forecasting energy usage, which is essential for efficient resource management. Beyond these critical areas, it also enhances everyday technology experiences, like search engine page rankings, demonstrating its broad impact across diverse aspects of daily life.

Therefore it is essential to be able to train accurate and robust models. With Ensemble Learning methods such as Bagging or Boosting it is possible to reduce the bias, variance and to create more resilient, robust, and generalized models. Thereby improving the overall accuracy and outperforming individual learning algorithms. Furthermore, Ensemble Learning methods are able to manage complex datasets characterized by noise, imbalance, and high dimensionality, as well as effectively hanlding missing data.

As a part of the proseminar "Advanced topics in machine learning", this report aims to provide a comprehensive overview of Bagging, Boosting, and Ensemble Learning, explaining how they work, why they are important in the field of machine learning, and comparing them on real world datasets.

2 Ensemble Learning

Ensemble learning is an advanced machine learning approach that combines the strengths of multiple smaller learning algorithms to improve predictive performance. The concept behind ensemble learning is analogous to the "wisdom of crowds". Which describes, that a crowd, on average, makes collectively better decisions, than any single member of it.

Just as a diverse group of people can provide a more accurate collective decision than an individual, in ensemble learning, a combination of learning algorithms often predicts more accurately than an individual learning algorithm. This approach is based on the principle that a diverse set of learning algorithms can capture different patterns or trends in the data, leading to more robust and accurate predictions.

To be more precise, ensemble methods use multiple smaller learning algorithms, which specialize in small aspects of the problem. However, by combining these algorithms, the ensemble often achieves better predictive performance than the used algorithm could achieve alone because they complement each others strengths and weaknesses.

So the goal of ensemble learning is to achieve a better predictive performance. Nevertheless, it comes at the cost of increased computational resources for training as well as prediction and storage.

Overall, there are many different ensemble methods, such as Bagging and Boosting, which we will go into more detail in this report. However, there are

many more like stacking and blending.

2.1 Bagging

Bootstrap Aggregating, commonly known as Bagging, is an ensemble learning method developed by Breiman (1996). The models for the ensemble get trained individually by using the bootstrapping technique. Bootstrapping involves creating random subsets of the original training dataset. The subsets are created by drawing random data points with replacement and have the same size as the original training dataset. This means that data points can be chosen more than once and that some data points might not be in the subset. The models can be trained in parallel, because of the individually created subsets.

Figure 1 shows how bootstrapping might work on an imaginary dataset. In subset 1 each class is equally distributed. Subset 2 has a focus on the red and orange bubbles. The subset N has a strong focus on blue, however it doesn't even have a single orange bubble. This will probably mean that the model trained on subset N will be very good at predicting blue, but not very accurate with orange.

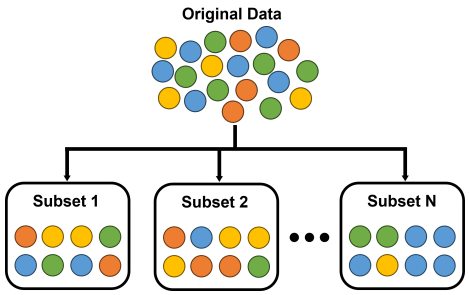

Figure 1: Creating subsets with bootstrapping.

In the Bagging ensemble, each model makes its prediction independently. Because of that the predictions of the individual models can be run in parallel, similar to the training. Once every model in the ensemble has made their prediction, they get aggregated to form a final ensemble prediction. The method of aggregation depends on the problem that is being solved by the Bagging ensemble. For classification problems, a common method is majority voting. Each model votes for a particular class. The class that receives the most votes is chosen for the final ensemble prediction. For regression problems, the predictions of the individual models are typically averaged.

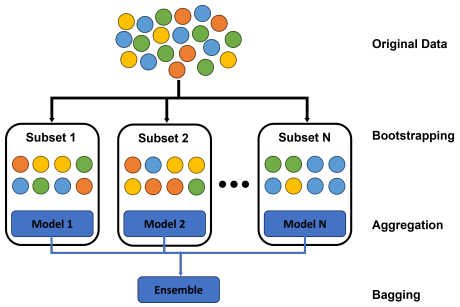

Figure 2: Bagging prediction example

Bagging can be particularly effective when using base learning algorithms that have high variance or tend to overfit quickly. By bootstrapping and averaging the predictions, the variance gets reduced and potential overfitting is avoided. The same goes for unstable learning algorithms, that produce significantly different results on small changes in the data. That's why Bagging is often used with Decision Trees as they tend to be unstable and to have high variance. Additionally, Bagging can be very helpful when dealing with noisy, imbalanced datasets or datasets with missing data. The bootstrapping helps to create diverse datasets with each class being adequately represented and also averaging out the noise in combination with aggregation.

All in all, Bagging can help to reduce variance, prevent overfitting, and to build a more resilient, robust, and generalized model.

2.2 Random Forest

Random Forest is an ensemble learning method, like Bagging, also developed by Breiman (2001). The difference between Bagging and Random Forests lies in the training of the models within the ensemble. First, the base learning algorithm is always a Decision Tree. Second, Random Forests utilize a method called Feature Randomness. When constructing each tree in a Random Forest, instead of considering all available features for splitting at each node, a random subset of features is selected. This randomness reduces the correlation between the individual trees, avoiding overfitting and increasing robustness and generalization.

2.3 Boosting

Boosting (Schapire, 1990) is another ensemble learning method like Bagging. However, unlike Bagging, Boosting trains the models sequentially, where each model learns from the mistakes of its predecessors. In the first step, the data subset for the first model gets created and a model is trained on it. Initially, the dataset is equally weighted, similar to the initial setup in Bagging: data points are drawn with replacement until the subset has the same size as the

original dataset. In the second step, the performance of the model is evaluated. The weight gets increased for incorrectly predicted samples and decreased for correctly predicted samples. In the third step, the next model is trained on the dataset with the updated weights. This weight adjustment makes the next model focus more on the data points that previous models predicted incorrectly. Steps two and three, the process of updating weights and training new models now gets repeated until all models are trained. Different from Bagging, Boosting can't be trained in parallel, because each model is trained based on the previous model's performance.

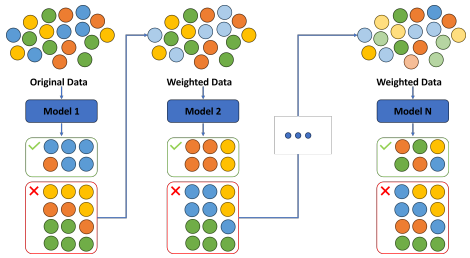

Figure 3: Boosting training example

In contrast to the training, each model makes its individual prediction in Boosting. This means the predictions can be done in parallel. Unlike Bagging where each model's prediction is given equal weight, Boosting uses a weighted average approach to combine the individual predictions. This means each model has its own weight based on the performance in the training phase. For classification problems, this means that the vote of each model can have a different influence on the final ensemble prediction. In the case of regression problems, the final prediction is the weighted average of the predictions from all models.

Boosting is ideal to use when the base learning algorithm suffers from high bias. This is combated through the way the ensemble is trained, as each model focuses on the weaknesses of the previous one. Additionally, Boosting is particularly effective with high-dimensional data, where the number of features is large. Other models might not be able to capture the complexity or more complex models might overfit. However, Boosting can strike a balance by incrementally building complexity and focusing on features that improve the predictive performance. Furthermore, it's important to note that Boosting works best on datasets with little to no outliers. The ensemble strongly focuses on correcting errors. Outliers can disproportionately influence the direction of the learning process, leading to overfitting.

To put it in a nutshell, Boosting can help to reduce bias, avoid underfitting, and can help to build overall precise models. Nevertheless, you have to be aware of the challenges with Boosting e.g. noisy data.

2.4 Gradient Boosting

Gradient Boosting (Breiman, 1997; Friedman, 2001, 2002) is an ensemble learning method based on Boosting. It is typically, but not solely used in combination with Decision Trees. Also called Gradient-Boosted Trees. Each Decision Tree is fit on the residuals (difference between target and predicted value) of the previous Decision Tree. As a start, the residuals are calculated by subtracting the mean. For the prediction, the trained trees are combined by multiplying their prediction with the learning rate and adding them all together with the mean. In difference to regular Boosting, all trees have the same weight, resulting in an equal contribution to the final prediction.

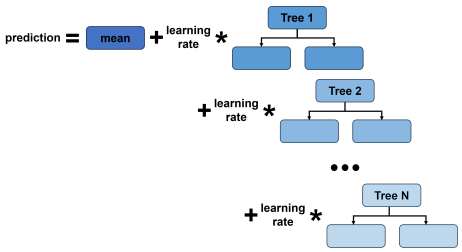

Figure 4: Gradient Boosting prediction example

3 Examples

In the examples, we aim to compare various ensemble learning algorithms - Bagging, Boosting (AdaBoost - Freund, Schapire, et al. (1996)), Random Forest, and Gradient Boosting - against Decision Trees across different datasets. First, we are looking for the optimal depth of the Decision Tree which maximizes the precision. This Decision Tree will be used as the base learning algorithm for the ensemble learning algorithms. In order to determine the best maximum depth of the Decision Tree, the datasets are split into a validation set (20%) and a training set (80%). The training set is used for a 10-fold cross-validation, where the overall accuracy is calculated by averaging the accuracy of all folds. With the 10-fold cross-validation of the training set, we are searching for the best maxmimum depth, trying all of them from 1 to 30. We stopped at 30 as the accuracy declined or stagnated for all examples. The Decision tree with the best accuracy is then used as the base learning algorithm for the ensembles.

The same method is used to find the best estimator count for Bagging, Boosting, Random Forest and Gradient Boosting. However, in contrast to the Decision tree, we are trying all estimator counts from 1 to 500. The learning rate is always 0.1 if applicable. Afterwards, the Decision Tree and the ensemble learning algorithms are tested on the validation set. This approach allows us to comprehensively assess each algorithm's effectiveness and draw meaningful comparisons.

The experiments were implemented in Python 3.10.1, using scikit-learn 1.3.2, and conducted on a system equipped with an AMD Ryzen 5 5600X CPU (3.7 GHz, 6 cores).

3.1 Example 1

In the first example we are using the Breast Cancer Wisconsin (Diagnostic) dataset by Wolberg, Mangasarian, Street, and Street (1995). The dataset originates from the University of Wisconsin Hospitals. It contains 30 features and has 569 data points. The goal of the dataset is to classify if breast cancer is benign or malignant (classification).

Criteria	DT	Bag	RF	Ada	GB
Worst fold	0.822	0.869	0.889	0.891	0.933
Best fold	0.978	1.000	1.000	1.000	1.000
Average fold	0.927	0.945	0.956	0.956	0.963
Validation	0.965	0.974	0.974	0.991	0.991
Training time	0.1s	2.9s	0.3s	2.5s	11.7s
Estimator count	-	61	21	33	174

Table 1: Subset accuracy on the Breast Cancer Wisconsin (Diagnostic) dataset using (1) a decision tree (DT) classifier; (2) a Bagging (Bag) classifier; (3) a Random forest (RF) classifier; (4) a AdaBoost (Ada) classifier; (5) a Gradient boosting (GB) classifier. All ensembles use a decision tree with maximum depth of four as the base learning algorithm.

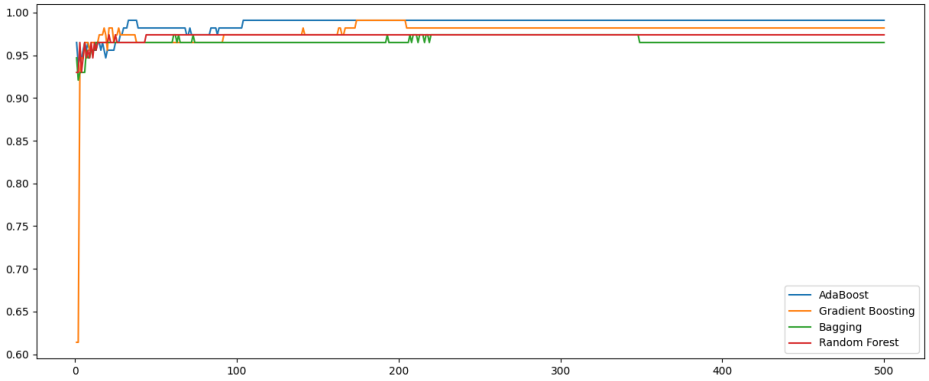

Figure 5: Accuracy comparison between the ensemble learning algorithms on the Breast Cancer Wisconsin (Diagnostic) dataset. The x-axis shows the amount of estimators and the y-axis the subset accuracy.

In the first example, all evaluated ensemble learning methods surpassed the decision tree in performance. Notably, Bagging and Random forest, as well as AdaBoost and Gradient boosting achieved identical accuracies on the validation

set. However, AdaBoost and Gradient boosting performed better on the validation set than the Bagging and Random forest ensemble. Overall, Gradient boosting performed best when also looking at the accuracy on the folds. Nevertheless, it also has the longest training time, and highest estimator count, which may be a trade-off to consider depending on the application. Also figure 5 shows that the performance of the ensemble learning methods mostly improve within the first 50 estimators, then it only changes slightly until around 220 estimators. After 220 estimators it nearly stagnates. Furthermore, as illustrated in Figure 5, the performance gains of the ensemble learning methods are most pronounced within the first 50 estimators and tend to plateau beyond 220 estimators. This indicates diminishing returns with the addition of further estimators.

3.2 Example 2

In the second example we are using the Heart Disease Cleveland dataset by Janosi, Steinbrunn, Pfisterer, and Detrano (1988). It contains 13 features and has 303 data points. The goal of the dataset is to determine if a patient has a heart disease (classification).

Criteria	DT	Bag	RF	Ada	GB
Worst fold	0.600	0.720	0.720	0.667	0.640
Best fold	0.917	0.875	0.917	0.833	0.875
Average fold	0.815	0.823	0.815	0.761	0.778
Validation	0.770	0.820	0.869	0.902	0.869
Training time	0.1s	0.6s	0.3s	0.2s	1.1s
Estimator count	-	48	48	15	160

Table 2: Subset accuracy on the Heart Disease Cleveland dataset using (1) a decision tree (DT) classifier; (2) a Bagging (Bag) classifier; (3) a Random forest (RF) classifier; (4) a AdaBoost (Ada) classifier; (5) a Gradient boosting (GB) classifier. All ensembles use a decision tree with maximum depth of three as the base learning algorithm.

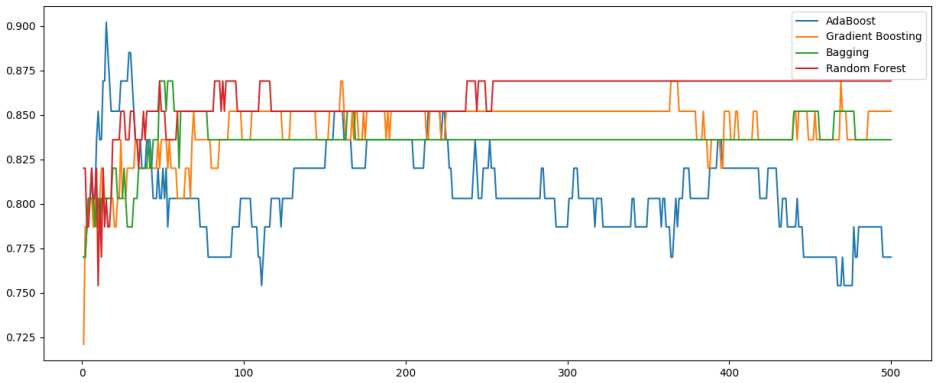

Figure 6: Accuracy comparison between the ensemble learning algorithms on the Heart Disease Cleveland dataset. The x-axis shows the amount of estimators and the y-axis the subset accuracy.

In the second example, all evaluated ensemble learning methods surpassed the decision tree in performance. There were significant performance enhancements, ranging from 5% up to 13%. Additionally, Table 2 shows that that random forest notably surpassed Bagging by nearly 4% in accuracy, eventhough

they have the same estimator count. Despite having the worst results on the folds, AdaBoost stood out with the highest accuracy on the validation set. This is particularly noteworthy considering that AdaBoost utilized the fewest estimators of all methods examined, yet managed to achieve the best performance. The results depicted in Figure 6 reveal a more volatile relationship between performance and estimator count compared to the first example. Consistent with the previous findings, the most substantial performance improvements occurred within the initial 50 estimators. A notable point of interest is the declining accuracy of AdaBoost as the number of estimators increase, suggesting a tendency for the model to overfit the training data. This underscores that more estimators can also have a negative effect on the generalizability of the model.

3.3 Example 3

In the third example we are using the day version of the Bike Sharing dataset by Fanaee-T (2013). It contains 13 features and has 731 data points. The goal of the dataset is to determine the amount of bikes that are going to be rented out on the given day (regression).

Criteria	DT	Bag	RF	Ada	GB
Worst fold	0.606	0.680	0.680	0.682	0.749
Best fold	0.906	0.925	0.925	0.943	0.945
Average fold	0.771	0.837	0.838	0.842	0.868
Validation	0.808	0.878	0.881	0.875	0.882
Training time	0.1s	0.6s	0.5s	8.3s	0.7s
Estimator count	-	30	30	498	38

Table 3: Coefficient of determination R^2 on the Bike Sharing dataset using (1) a decision tree (DT) regressor; (2) a Bagging (Bag) regressor; (3) a Random forest (RF) regressor; (4) a AdaBoost (Ada) regressor; (5) a Gradient boosting (GB) regressor. All ensembles use a decision tree with maximum depth of five as the base learning algorithm.

4 Summary and conclusion

Ensemble Learning is a powerful approach in machine learning that combines multiple models to produce a superior output compared to individual models. Within ensemble learning, two primary techniques are Bagging and Boosting, each with its unique methodology and advantages.

Bagging aims to reduce variance in predictions. In bagging, models are trained independently in parallel, using different subsets of the training data. The final prediction is typically made by aggregating the outputs of all models, using methods like majority voting (classification) or averaging (regression).

Boosting focuses on reducing bias. In this technique, models are trained sequentially, where each model attempts to correct the errors made by the previous one. Despite the sequential training, predictions can be made in parallel. The final prediction is made similar to bagging. However, the performance of each model in the training step determines how strong the influence is on the final prediction.

The evolution of these methods into more advanced versions like Random Forests and Gradient Boosting has further enhanced their capabilities. These advanced techniques can outperform standard individual models and even the basic forms of bagging and boosting.

The choice between the model should be guided by the specific requirements of the problem at hand, considering factors like data characteristics, computing resources, and the inherent trade-offs between bias and variance.

References

Breiman, L. (1996, Aug 01). Bagging predictors. *Machine Learning*, *24*(2), 123-140. Retrieved from https://doi.org/10.1007/BF00058655 doi: 10.1007/BF00058655

Breiman, L. (1997). *Arcing the edge* (Tech. Rep.). Citeseer.

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5–32.

Fanaee-T, H. (2013). *Bike sharing dataset.* UCI Machine Learning Repository. (DOI: https://doi.org/10.24432/C5W894)

Freund, Y., Schapire, R. E., et al. (1996). Experiments with a new boosting algorithm. In *icml* (Vol. 96, pp. 148–156).

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational statistics & data analysis*, *38*(4), 367–378.

Janosi, A., Steinbrunn, W., Pfisterer, M., & Detrano, R. (1988). *Heart disease.* UCI Machine Learning Repository. (DOI: https://doi.org/10.24432/C52P4X)

Schapire, R. E. (1990, Jun 01). The strength of weak learnability. *Machine Learning*, *5*(2), 197-227. Retrieved from https://doi.org/10.1007/BF00116037 doi: 10.1007/BF00116037

Wolberg, W., Mangasarian, O., Street, N., & Street, W. (1995). *Breast cancer wisconsin (diagnostic).* UCI Machine Learning Repository. (DOI: https://doi.org/10.24432/C5DW2B)