Review of Machine Learning Ensemble Method

Yoong Pei Xian 100334770,
Research Techniques,
University of East Anglia

December 6, 2021

Abstract

Despite the significant success achieved in the machine learning field, individual algorithms still struggle to maximize their performances due to the complexity of data. Complex data which present high skewness, dimensionality, or even noise can challenge algorithms to effectively capture the shape of the trends. For decades, researchers have been exploring different ways to maximize the performances of machine learning to create efficient models and advance the process of data mining and knowledge discovery. Ensemble methods, the backbone of machine learning techniques are deemed to be the most powerful and easiest to use of predictive algorithms to resolve the limitations faced by individual algorithms. As the term 'ensemble' indicates, ensemble learning refers to grouping several algorithms, combining the average of their votes on a particular matter to yield better performance. Several of the most popular ensemble learning methods include bagging and boosting, which are later modified and improved by researchers to adapt to the different characteristics of data.

In this paper, the theoretical backgrounds of various ensemble learning methods are explained along with the justification of adopting them in machine learning. A particular focus is given to bagging, boosting, and stacking that is widely evaluated in the existing studies. In addition, challenges and limitations presented in these studies are evaluated. Three areas of ensemble learning methods are of great interest in this paper: 1) performance of bagging and boosting in comparison to the individual algorithm, 2) evaluation of bagging, boosting, and stacking for the class imbalance problems as it is put into question by many researchers regarding the extent it applies to real-life cases, 3) potential research gap discovered when conducting the literature review.

1 Introduction

Machine learning allows computers to learn complicated rules through the different algorithms to analyze and interpret vast amounts of data. Google prediction algorithm, Siri speech recognition, and Netflix recommendation system are all examples of machine learning. Compared to the traditional predictive model, machine learning can improve the accuracy of the prediction by consuming high-dimensional data and calculating them mathematically. Continued improvement is supported, meaning that the machine can better adjust their output based on the input given (Wyman 2018).

Machine learning can be broken down into three types, supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, the model is trained using labeled data and predicts the outcome based on the pattern of the labeled data. The algorithms find the most ideal hypothesis by searching through the hypothesis space to make a prediction, but it is always challenging to determine which one is the most ideal hypothesis. This can be assisted by using ensemble learning methods, which combine multiple hypotheses to select a better hypothesis.

Ensemble learning techniques work by combining several base models to enhance the robustness of the algorithm and achieve an optimal result. Two questions are present in the context of ensemble learning: the generation of a pool of accurate base algorithms and the combination of different base algorithms to maximize the accuracy (Schwenker 2013). It emphasizes the strengths and reduces the limitations presented in each of the individual algorithms (Guzman et al. 2015). The individual algorithm is trained independently using different parameters and their average outputs are taken to be the final prediction (Nti et al. 2020). This technique permits higher predictive performance compared to an individual model. It also enhances accuracy, consistency by avoiding overfitting and reduces bias and variances errors.

Ensemble learning is divided into two categories: sequential ensemble techniques and parallel ensemble techniques (CFI n.d.). Sequential ensemble techniques generate the base learners in sequence and allow them to promote dependence within each other. One of the examples is AdaBoost. The overall performance can be boosted as higher weights are

assigned to previously misrepresented learners. In contrast, parallel ensemble techniques generate the base learners in parallel, for instance, random forest. It can effectively reduce the error by encouraging the independence of the base learners.

2 Aims and Objectives

- To evaluate the performance of ensemble learning algorithms in comparison to individual algorithm.
- To review the performance of ensemble learning algorithms for the class imbalance problems..
- To identify the potential research gaps in the ensemble learning fields.

3 Literature Review

A detailed account of current related work is given below. First, the justification of ensemble learning and the different ensemble learning methods are reviewed followed by their strengths and weaknesses as discovered in the research papers. Second, a discussion on the performances of different techniques using existing studies to draw comparison and critiques on these studies are formed. Then, the performance of different ensemble learning methods on the imbalance dataset is evaluated.

3.1 Justification of Ensemble Learning

The rising popularity of ensemble learning methods can be justified with the following advantages, which address the challenges faced by most of the individual algorithms:

- Computational, statistical and representation problems

The algorithms usually search for the most optimal hypothesis among other hypotheses within the search space. The computational problem arises when the algorithm is unable to find the best hypothesis, meaning that it is stuck in the local minima and this issue is

particularly obvious in the decision tree and neural network algorithms (Dietterich 2000). This can be resolved with ensemble learning algorithms that perform a local search via various starting points (Ganaie et al. 2021).

The statistical problem happens when the training data size is smaller than the size of hypothesis space (Dietterich 2000). Thus, hypotheses that produce similar performance are identified. In this case, ensemble learning algorithms can combine the different results to reduce the risk of selecting a less ideal local optimal and enhance the approximation of the unknown function.

The representation problem occurs when none of the hypotheses within the hypothesis space is a good representation of the unknown function. Ensemble learning algorithms allow the weight of the vote of each hypothesis to be taken to expand the representable function space, forming a more accurate approximation of the unknown function (Ganaie et al. 2021). Algorithms that suffer from statistical problems are considered to have high variance while those with representation issues are considered to have a high bias. The bias and variance issues are discussed in the following points.

- Bias variance decomposition

Bias error means that the model shows a high inclination towards an outcome of a problem it seeks to solve while variance error means overfitting, where the model fits the training data too well while performing badly on the new dataset (Ayuya 2018). Various approaches such as bagging, and boosting are widely integrated to reduce the bias and variance of the individual algorithms and they are proven to have the ability to reduce squared errors when compared to individual algorithms (Krogh & Vedelsby 1995, Brown et al. 2005).

- Diversity

Ensemble learning methods allow the base of the algorithms to be diverse, with the integration of multiple approaches such as bagging, boosting, random forest to improve the performance of the model. Even though the definition of diversity measure is still unclear on how it can be used to improve the accuracy of algorithms (Minku et al. 2010)

multiple studies present a positive correlation between the diversity and the accuracy of models (Kuncheva & Whitaker 2003, Dietterich 2004). Kuncheva & Whitaker (2003) have conducted an experiment using 10 different diversity measures for the classifiers and they are all positively correlated to each other, despite the exhibition of different behaviors within each other (Ganaie et al. 2021).

3.2 Ensemble Learning Methods

3.2.1 Bagging

Bagging, also known as bootstrapping aggregation, generates results based on the output of a lot of models. In terms of the variances, bagging can effectively reduce the variance by aggregating individual models without increasing the bias, where Breiman (1994) considers it as a variance reduction technique for algorithms that perform attributes selection to fit in a linear model. This argument has been supported by Bühlmann (2012), claiming that bagging has been effective in reducing the mean squared error of the model where it serves as a smoothing operation to enhance the model performance. Due to the reason that bagging is always associated with prohibitive cost, sub-bagging, where subsampling is used rather than the aggregation bootstrap, is introduced to reduce the computational cost (Bühlmann & Yu 2001).

Bagging is particularly useful for high variance models like decision trees (Ayuya 2018). Taking the individual decision tree for instance as it is often used in practice in conjunction with bagging, it presents weaknesses of instability, meaning that minor change in the dataset can lead to a substantial change in the structure of the optimal decision tree, which can be effectively addressed by using bagging (Bühlmann 2012). In Zareapoor & Shamsolmoali (2015)'s research on detecting credit card fraud using the bagging technique, they discover that bagging shows a stable performance on both training and testing evaluation, which yields better performance on the prediction accuracy while not increasing the false alarm rate, unlike the individual decision trees. The study also finds that bagging can handle the imbalance dataset particularly well, meaning that it is suitable to be applied for most real-life cases. Real-life cases frequently face class imbalance issues where the

data distribution is highly skewed, and the minority occupied 10% of even lesser of the entire dataset. This can cause a strong bias in the prediction of the supervised classifier to favor the majority class.

Despite many studies showing that bagging is effective in improving the performance of decision trees, researchers argue that its smoothing effect is only limited when it comes to models that do not involve in making hard decisions (Bühlmann 2012). Smoothing is a process of trends detection in the presence of noisy data in cases the trend shape is unknown. The decision tree is a highly unstable algorithm due to its nature where hard decisions are involved (Bühlmann 2012). For the more stable algorithms such as KNN Classifier, bagging usually does not have a significant impact on the performance. To address this, random subspace methodology can be introduced to integrate with stable algorithms. Random subspace methodology uses only a portion of the training data, else only a certain number of attributes are being selected to avoid overfitting and produce variation among these models. This can be achieved by picking a subset of attributes randomly by adjusting the max features parameter in the classifier. According to Dietterich (2000), the combination of bagging and random subspace methodology produces an excellent accuracy, and it can perform relatively well even when there is noise in the dataset.

While bagging is useful for high variance models, it does not produce high accuracy for the high bias models. To enhance the performance of the high bias models, boosting is more commonly adopted.

3.2.2 Boosting

Boosting increases the complexity of the model as it trains new models based on the error that occurred in the previous model to improve the performance. It is particularly suitable for models with the tendency of underfitting, meaning that the model poorly fits the training dataset. These models usually consist of high bias and have low variance.

Weight is assigned to each model for the next model to select a model with a higher misclassification rate and train on them. This allows every new model to emphasize its

effort on the data that cannot fit correctly. The iteration allows the data to fit better every time and thus, reducing the errors (Ayuya 2018). One of the famous successful cases of boosting is the Viola Jone Face Detection algorithm using the decision tree as the base model.

There are distinct types of boosting algorithms that can be adopted based on different requirements, for instance, AdaBoost (adaptive boosting) and gradient boosting.

AdaBoost is often called the "best out-of-the-box classifier" due to the reason that it can be applied on top of any algorithms to learn from the weakness and propose a more accurate model (Kurama 2018). AdaBoost was the first realization of boosting that saw great success in application. For example, research conducted by Bauer & Kohavi (1999) shows that AdaBoost can yield a 27% of improvement in the misclassification error compared to the individual decision tree. The research also testifies that boosting can significantly reduce the bias error, supported by Breiman (1998). The overfitting behavior of the AdaBoost has been a long-lasting discussion as it usually stops very late in the iteration, causing an increase of the complexity to favor overfitting. However, studies show that it can be quite resistant to the overfitting Grove & Schuurmans (2004), Rätsch et al. (2004), which opposes the Occam's Razor principle stating that a less complex algorithm will normally outperform a more complex one (Freund & Schapire 1996). In the meantime, instead of only binary classification, the usage of AdaBoost has been extended to text and image classification. As AdaBoost is highly sensitive to outliers and noise, it is essential to eliminate them to obtain an accurate result.

The generalization of AdaBoost is called Gradient boosting, where it uses gradient instead of the weight to assign to every model, a model developed by Friedman (Brownlee 2020*a*). A weak learner is added at a time to minimize the loss of the model with the process of gradient descent.

3.2.3 Stacking

Stacking is a mechanism that involves training two or more base models, often indicated as base models and meta-model to forecast the target variable while also learning to use

the predictions of each base model to predict the value of the target variable (Brownlee 2020*b*). The meta-model involves utilizing the forecasts derived by the base models on out-sample data. Data that are not used to teach the base models are utilized in estimating outputs, which inform the dataset used to fit the final meta-model. The goal of stacking is to minimize variances in data and predictive force.

While bagging and boosting only consider homogeneous weak learners, where they take the average and weighted median respectively, they can be integrated with stacking, which takes heterogeneous weak learners into account and identifies the best way of combining them (Drucker 1997).

Stacking is not an algorithm, but a generic name used to describe the process of learning from the weak learners in parallel and identify the best way to combine them by training a meta-classifier or a meta-regressor to produce a prediction-based on the different weak models (Rocca 2020).

4 Analysis and Discussion

4.1 Comparison and Contrast

The performance of different ensemble learning methods is always a popular research topic where boosting is usually considered to be more powerful than bagging and stacking in terms of the predictive power (Breiman 1998). However, there is still considerable disagreement regarding this finding. A summary of the research papers used in this paper has been presented in the following table.

Authors	Titles	Methods	Imbalance Dataset
Breiman (1994)	Bagging Predictors	Boosting	No
Drucker (1997)	Improving Regressors Using Boosting Technique	Boosting with pruning, Bagging, stacking	No
Alazzam et al. (2017)	Software fault proneness prediction: a comparative study between bagging, boosting, and stacking ensemble and base learner methods	Bagging, boosting, stacking	No
Hido & Kashima (2008)	Roughly Balanced Bagging for Imbalanced Data	Roughly Balance Bagging (RB)	Yes
Khoshgoftaar et al. (2011)	Comparing Boosting and Bagging Techniques With Noisy and Imbalanced Data	SMOTEBoost, RUSBoost, EBBag, and RBBag	Yes
Zareapoor & Shamsolmoali (2015)	Application of Credit Card Fraud Detection: Based on Bagging Ensemble Classifier	Bagging	Yes

4.2 Evaluation

Several of the earliest studies conducted on the ensemble learning methods (Breiman 1994, Freund & Schapire 1996) have shown superiority on the boosting over bagging, supported by many researchers (Drucker 1997, Alazzam et al. 2017).

Drucker (1997) has asserted that boosting statistically outperformed bagging where it improved the model significantly by reducing the prediction error to a greater extent when compared to bagging. His study was a replication of Breiman's study, with the incorporation of pruning modification that yielded better performance. The metrics selected in Drucker (1997)'s study was the modeling and prediction error rate, where Zareapoor & Shamsolmoali (2015) criticized to be biased metrics and not recommended to be adopted in the imbalance class problems.

The shortcomings of Drucker (1997)'s methodology was recognized that the size of the dataset used was relatively small, which might be a limitation on the computational resources, and he neglected to consider that most of the real-life cases are imbalance, where the algorithms tend to favor the majority class. This limits how far the findings can be taken and applied to real-life cases. Besides, a more systematic and theoretical analysis providing rationale behind each step of experiment design is also required for the findings to be convincing.

Hido & Kashima (2008) have also reimplemented the same study conducted by Breiman and they realized that boosting never outperformed bagging, supported by Zareapoor & Shamsolmoali (2015) in his study on the detection of credit card fraud where they also found that bagging takes a relatively short time, which serves as an important parameter for fraud detection. The study (Hido & Kashima 2008) has also revealed that boosting resulted in a strong bias towards the majority class (Hido & Kashima 2008). Hido & Kashima (2008) strongly criticized that Breiman's study involved the duplication of the minority class, which can potentially result in overfitting due to the multiple replications. To resolve this, SMOOTBoost, which generates synthetic minority class, which is favored by many researchers Hido & Kashima (2008), (Khoshgoftaar et al. 2011) was adopted in their study. In Syarif et al. (2012)'s study on the application of bagging, boosting, and stacking to intrusion detection, they agreed that boosting never outperformed boosting, and, the use of bagging, boosting, and stacking is unable to significantly improve the accuracy.

To handle imbalance class problems, Khoshgoftaar et al. (2011) ran the experiment using the four different datasets with various percentages of the minority class, and they have converted the original multi-class datasets into binary-class datasets. In contrast to the previous studies (Breiman 1998, Drucker 1997), Khoshgoftaar et al. (2011) found that there was no significant difference in the performance of boosting and bagging when the dataset is clean but bagging outperformed boosting when the dataset is imbalanced and noisy. They have selected seven different performance metrics including area under the receiver operating characteristic curve (AROC), area under the precision-recall curve (APRC), Kolmogorov–Smirnov statistic (KS), F-measure (FM), geometric mean (GM), best FM (BFM), and best GM (BGM) that have not been adopted in the prior studies to present a different perspective. This could raise concerns about the internal threats of validity as there is a considerable amount of uncertainty in the performance of these metrics. Despite this, Khoshgoftaar et al. (2011)'s study seems to be well-grounded with the presentation of a well-designed research structure that clearly explains the rationale behind the dataset selection, specifies the bagging and boosting techniques, and sets up

a comprehensive experimental design. It is worth taking note is that their study entails tedious procedures and high computational resources to handle the considerably large dataset used, which might be less feasible for cost-constraint experiments.

In the meantime, Hido & Kashima (2008) has also proposed the Roughly Balanced Boosting (RB) method to tackle the imbalance dataset, presenting a positive finding that RB outperformed the normal bagging and all the ensemble learning algorithms being tested in the experiments, including the most robust AdaBoost, and it is promising to be applied to real-life cases.

The research conducted by Nti et al. (2020) on the stock market prediction using various ensemble learning algorithms discovers that stacking produces a higher accuracy compared to bagging and boosting but it compromises on the higher computational cost due to the high training and testing time. This finding has been supported by Syarif et al. (2012), claiming that stacking demonstrates the longest execution time among other ensemble learning algorithms and so it is inefficient despite its ability to reduce the false positive rate significantly, but it does not present improvement in the accuracy, as opposed to the research of Abubacker et al. (2020) which concluded that stacking proves higher accuracy in mammogram classification. Drucker (1997) has also shown that stacking shows an inconsistent performance on bagging and boosting, and it can potentially make the performance worse than using standalone bagging or boosting algorithm. The significant differences in the results across studies can be justified by the varied algorithms stacked. For instance, combinations of naïve Bayes, iBK, J48, and JRip were selected in Syarif et al. (2012)'s study but none of these algorithms were presented in Abubacker et al. (2020)'s studies. It is almost impossible to draw a comparison when the algorithms, datasets, and methodology adopted were entirely different. Nonetheless, there are similarities presented in the findings of these studies, stating that stacking involved high computational cost and long execution time could still be observed.

5 Conclusion

This literature review has summarized the existing studies on ensemble learning to justify its rising popularity, also the strengths and limitations of the different ensemble learning methods, ranging from bagging, boosting, and stacking, to provide readers a preliminary understanding of these methods. The key findings of this review are summarized as follows:

- Boosting outperforms bagging when the dataset consists of a low noise level, but it is extremely sensitive and detrimental to the noise
- Roughly Balance Bagging yields better performance than usual bagging when handling an imbalanced dataset. It also outperforms the powerful AdaBoost.
- Stacking is proven to be the most effective compared to bagging and boosting, but it comes with a high computational cost. Should the accuracy of the result be the most vital, stacking is highly encouraged to be adopted.

A closer look at the literature on ensemble learning, however, reveals several gaps and shortcomings. Most of the studies in this field failed to recognize the importance of applying ensemble learning to the imbalance dataset to explore the performance of most of the real-life settings (Breiman 1998, Drucker 1997). Even researchers recognized that most of the real-life datasets are skewed, they have not treated them in many details (Alazzam et al. 2017). To the best of our knowledge, there is only a limited number of studies conducted to evaluate the performance of various algorithms on the imbalance dataset. Additional studies to understand more completely the key tenets of the performance comparison using a different set of real-world datasets and benchmarks with varied class ratios, sample sizes and research methodologies are required.

References

Abubacker, N., Hashem, I. & Hui, L. (2020), 'Mammographic classification using stacked ensemble learning with bagging and boosting techniques', *Journal of Medical and Biological Engineering* **40**, 908–916.

Alazzam, I., Alsmadi, I. & Akour, M. (2017), 'Software fault proneness prediction: a comparative study between bagging, boosting, and stacking ensemble and base learner methods', *International Journal of Data Analysis Techniques and Strategies* **9**, 1.

Ayuya, C. (2018), 'Ensemble learning on bias and variance'. Available at `https://www.section.io/engineering-education/ensemble-bias-var/` [Accessed on November 29, 2021].

Bauer, E. & Kohavi, R. (1999), 'An empirical comparison of voting classification algorithms: Bagging, boosting, and variants', *Machine Learning* **36**, 105–139.

Breiman, L. (1994), 'Bagging predictors'.

Breiman, L. (1998), Arcing classifiers.

Brown, G., Wyatt, J., Harris, R. & Yao, X. (2005), 'Diversity creation methods: A survey and categorisation', *Information Fusion* **6**, 5–20.

Brownlee, J. (2020*a*), 'A gentle introduction to the gradient boosting algorithm for machine learning'. Available at `https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/` [Accessed on November 29, 2021].

Brownlee, J. (2020*b*), 'Stacking ensemble machine learning with python'. Available at `https://machinelearningmastery.com/stacking-ensemble-machine-learning-with-python/` [Accessed on November 29, 2021].

Bühlmann, P. (2012), 'Bagging, boosting and ensemble methods', *Handbook of Computational Statistics* .

Bühlmann, P. & Yu, B. (2001), Analyzing bagging.

CFI (n.d.), 'Ensemble methods'. Available at `https://corporatefinanceinstitute.com/resources/knowledge/other/ensemble-methods/` [Accessed on November 29, 2021].

Dietterich, T. G. (2000), Ensemble methods in machine learning, *in* 'Multiple Classifier System, LBCS-1857', Springer, pp. 1–15.

Dietterich, T. G. (2004), 'An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization', *Machine Learning* **40**, 139–157.

Drucker, H. (1997), 'Improving regressors using boosting techniques', *Proceedings of the 14th International Conference on Machine Learning* .

Freund, Y. & Schapire, R. E. (1996), Experiments with a new boosting algorithm, *in* 'In Proceedings of the Thirteen International Conference of Machine Learning', Morgan Kaufmann, pp. 148–156.

Ganaie, M., Hu, M., Tanveer, M. & Suganthan, P. (2021), 'Ensemble deep learning: A review'.

Grove, A. & Schuurmans, D. (2004), 'Boosting in the limit: Maximizing the margin of learned ensembles', *Proceedings of the National Conference on Artificial Intelligence* .

Guzman, E., El-Haliby, M. & Bruegge, B. (2015), Ensemble methods for app review classification: An approach for software evolution (n), *in* '2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)', pp. 771–776.

Hido, S. & Kashima, H. (2008), Roughly balanced bagging for imbalanced data, *in* 'SDM'.

Khoshgoftaar, T. M., Van Hulse, J. & Napolitano, A. (2011), 'Comparing boosting and bagging techniques with noisy and imbalanced data', *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* **41**(3), 552–568.

Krogh, A. & Vedelsby, J. (1995), Neural network ensembles, cross validation, and active learning, *in* 'Advances in Neural Information Processing Systems', MIT Press, pp. 231–238.

Kuncheva, L. & Whitaker, C. (2003), 'Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy', *Machine Learning* **51**, 181–207.

Kurama, V. (2018), 'A guide to adaboost: Boosting to save the day'. Available at `https://blog.paperspace.com/adaboost-optimizer/` [Accessed on November 29, 2021].

Minku, L. L., White, A. P. & Yao, X. (2010), 'The impact of diversity on online ensemble learning in the presence of concept drift', *IEEE Transactions on Knowledge and Data Engineering* **22**(5), 730–742.

Nti, I. K., Adekoya, A. F. & Weyori, B. A. (2020), 'A comprehensive evaluation of ensemble learning for stock-market prediction', *Journal of Big Data* **7**(1).

Rocca, J. (2020), 'Ensemble methods: bagging, boosting and stacking'. Available at `https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205` [Accessed on November 29, 2021].

Rätsch, G., Onoda, T. & Müller, K.-R. (2004), 'Soft margins for adaboost', *Machine Learning* **42**, 287–320.

Schwenker, F. (2013), 'Ensemble methods: Foundations and algorithms [book review]', *IEEE Computational Intelligence Magazine* **8**(1), 77–79.

Syarif, I., Zaluska, E., Adam I. Syarif, E. P.-B. & Wills, G. B. (2012), Application of bagging, boosting and stacking to intrusion detection, *in* 'MLDM'.

Wyman, O. (2018), 'Why walls street needs to make investing a machine learning a higher priority'. Available at https://www.oliverwyman.com/content/dam/oliver-wyman/v2/publications/2018/may/Machine-Learning-on-Wall-Street.pdf [Accessed on November 29, 2021].

Zareapoor, M. & Shamsolmoali, P. (2015), 'Application of credit card fraud detection: Based on bagging ensemble classifier', *Procedia Computer Science* **48**, 679–686.