Does Size Matter?

1st Ben van Duivenbooden (23709235)
dept. of Computer Science
University of Stellenbosch
Stellenbosch, South Africa
23709235@sun.ac.za

***Abstract*—**

I. Introduction

Ensemble learning (EL) is a machine learning (ML) paradigm whereby multiple base models are trained, after which the resulting predictions are combined [1], [2]. EL techniques obtain results that outperform, and have better generalization abilities than that of the individual base learners [3]. The principle driving EL is the recognition that ML models can make errors and have certain limitations. Subsequently, EL aims to improve classification and generalization performance by employing multiple base models. Limitations include low predictive accuracy, high bias, and high variance [4], [5]. By harnessing the strengths of multiple models, EL approaches generally achieve greater overall accuracy than single ML algorithms [6]. Furthermore, EL methods can reduce bias and variance using methods such as bagging and boosting.

One of the most commonly used EL approaches the the Random Forest (RF) algorithm. RF is a supervised learning algorithm, and can be used for both regression and classification tasks. The algorithm uses a combination of decsion trees (DTs) as base learners, each of which is trained on a subset of the data. In addition, RF is considered a homogeneous ensemble, since all the individual learners that make up the ensemble employ the same ML model.

As with any ML algorithm, selecting the appropriate hyperparameters forms a critical part of optimizing the performance RF. The most commen parameters inclyde, number of trees, number of features and samples for each DT, and maximum depth of trees. Several studies have explored the effect that the number of trees has on the overall performance of RF [7]–[9]. Furthermore, sensitivity analysis of the various parameters have also been conducted [10]–[12].

The main objective of this study is to further explore the effect of maximum tree depth, as well as the impact of the number of randomly selected features when deciding on a node split. The RF algorithm is applied to five classification problems of various complexity. The aim is to investigate the the performance of the RF where individual ensemble members underfit on the training data, and moving to a RF where the members each overfit the data. The relationship between the maximum depth and the number of trees used is also analyzed

The remainder of the paper is structured as follows: Section II gives and overview of ensemble learning, DTs, and RFs. Section III describes the RF implementation employed in this study. Next, Section IV describes the experimental procedure and statistical analysis conducted. The results are presented in Section V, and final remarks and future prospects is given in Section VI.

II. Backround

A. Ensemble Learning

B. Decion Trees

A decision tree (DT) is a tree structure which can be recursively defined and consists of both leaf nodes and internal nodes (decision nodes). Leaf nodes (terminal nodes) contain the predicted outcomes and each internal node denotes a test on a feature, with branches to lower nodes (child nodes) that represent the outcomes of the test [13], [14]. When the target variable is nominal, the DT is referred to as a classification tree, and when the target variable is numerical, the DT is referred to a regression tree [15], [16].

DT induction is a supervised machine learning approach that can be used for classification as well as regression problems. Learning in the context of DTs refers to the induction algorithm used to construct the DT from a set of observations. The first regression tree algorithm, qutomatic interaction detection (AID), was published in 1963 [17]. DTs became notably prominent in the 1980s when several induction algorithms were developed [15]. Some popular algorithms include CHAID [18], CART [19], ID3 [20], and C4.5 [21].

The ID3 [20] builds the tree recursively, starting at the root node. Next, the feature that best seperates the data is selected. The feature is selected by computing the information gain (IG) of all the features. The feature that results in the largest IG is then used as the test to seperate the data. This process of finding the best split for the data at each given internal node continuous untill all the instances in a data partition have the same target label. When this condition is met, a leaf node is created.

The ID3 algorithm utilizes entropy to calculate the IG. Entropy is defined as

$$Entropy(D) = -\sum_{i=1}^{N} p_i \log_2(p_i) \tag{1}$$

where p_i is the probability of an instance belonging to class c_i at the current node, for the current data partition. The information gain is calcualted as

$$\text{IG}(A) = \text{Entropy}(D) - \sum_{j=1}^{d} \frac{|D_i|}{|D|} \text{Entropy}(D_i) \tag{2}$$

where D is the data set, and A a specific attribute. If attribute A has d different outcomes $\{a_1, a_2, ..., a_d\}$, then D_j represents the subset of data that have the outcome a_j.

The main issue of the ID3 algorithm, is that it only works with categorical features. The C4.5 [21] and CART [19] algorithms on the other hand are capable of handling both categorical and continuous features.

The C4.5 is the extention of the ID3 algorithm, and uses the gain ratio as the metric for determining optimal splits. The IG has a bias towords features with many unique values, and the gain ratio addresses this problem by taking the sizes and number of braches into account [22]. The gain ratio is defined as

$$\text{Gain Ratio}(A) = \frac{\text{IG}(A)}{\text{SI}(A)} \tag{3}$$

where SI, the split information, is given by

$$\text{SI}(A) = -\sum_{i=j}^{d} \frac{|D_j|}{|S|} \log_2 \left(\frac{|D_j|}{|D|} \right) \tag{4}$$

The CART algorithm [19] on the other hand uses the gini index to determine the optimal split. In addition the CART algorithm constructs only binary decision trees, i.e., each internal node only has two children nodes. The Gini index is given by

$$\text{Gini Index}(D) = 1 - \sum_{i=1}^{N} P_i^2 \tag{5}$$

where $P_i = \frac{|C_i|}{|D|}$. The number of instances in D is given by $|D|$, and C_i is the number if instances relative to class C_i.

Decsion trees are often prone to overfitting, especiallywhen the training data contains noise. One technique to prevent overfitting is tree pruning, whereby subtrees that cause the model to overfit are removed. The easiest way to prune a tree is to use pre-pruning strategies, also known as early stopping. Examples of early stopping criteria include a predefined tree depth, and minimum number of instances to form a leaf node. More complex pre-pruning approached, for exmaple χ^2 pruning uses statistical tests to determine the importance of subtrees. The problem with pre-runing strategies is that that they often miss potentially good splits due to the nature of early stopping [22].

An alternative approach is to use post-pruning, which allows the tree to grow to completion. Each branch in the tree is subsequently analysed. Branches that likely cause overfitting are removed. A common method is cost-complexity-pruning [23], which assigns a cost to each subtree, and removing those with the lowest cost. The cost is calcualted as

$$C_\alpha(T) = R(T) + \alpha|T| \tag{6}$$

where $C_\alpha(T)$ and $R(T)$ is the cost-complexity and misclassification rate of subtree T. The number of leaf nodes of T is given by $|T|$. The complexity parameter α determines how much the tree is penalized. An increase in α results in more of the tree being pruned. Another popular approach is reduced error pruning, proposed by Quinlan [21] in 1993, which prunes based on error rates.

C. Random Forests

RFs became notably prominent after the work done by Breiman in 2001 [24], in which he proposed proposed the RF algorithm. Breiman was highly influenced by the work of Amit and Geman [25], who propsed the "randomized trees", method, as well as "random decision forests" introduced by Ho in 1995. The original RF algorithm [24] utilizes the CART algorithm for each DT, and the datasets for each DT is constructed using bootstrap samples. The Gini index is used as the criteria for determining optimal node splits, and at each split, only a subset of randomly selected features are considered. When applied to classification tasks, the most common procedure for combining the results of the DTs is majority voting [26].

III. Implementation

This study utilizes the scikit-learn [27] random forest classifier

IV. Empirical Procedure

A. Datasets

- **Iris**: Possibly the most commonly used benchmark dataset, the Iris dataset contains 150 samples, described by four continuous features. The target variable is comprised of three classes, which represent the type of iris plant, namely Setosa, Versicolour, and Virginica. The distribution of the classes is uniform, i.e., each class occurs 50 times.
- **Breast Cancer**: The Breast Cancer dataset is a popular dataset in the field of machine learning and biomedical research. It contains data collected from 569 breast cancer patients, each represented by 30 features that describe various characteristics of cell nuclei extracted from breast mass biopsies. The dataset is a binary classification problem, with classes 0 and 1 representing benign (non-cancerous) and malignant (cancerous) respectively. Of the 569 samples, 357 instances are malignant, and 212 benign.
- **Wine Recognition Data**: The Wine recognition dataset consists of 178 samples, characterized by 13 continuous features that represent the chemical constituents of wines derived from three different cultivars grown in the same region of Italy. The dataset is designed for classification tasks, where the target variable represents three different cultivars. The features include measurements such as alcohol, malic acid, ash, and total phenols. No missing values are present in the dataset.
- **Ionosphere**: The Ionosphere dataset is a well-known dataset for classification tasks in the field of radar signal

analysis. It contains 351 instances, each represented by 34 continuous features. The data was collected using a phased array of 16 high-frequency antennas, stationed in Goose Bay, Labrador, to study radar returns from the ionosphere. The features describe the autocorrelation function of signals transmitted and received by the antennas, with each pulse number providing two complex-valued attributes. The target variable is binary, with classes labeled as "Good" or "Bad." A "Good" radar return indicates evidence of structure in the ionosphere, while a "Bad" return signifies signals that passed through the ionosphere without detection of such structure.

- **Optical Recognition of Handwritten Digits**: The Optical Recognition of Handwritten Digits dataset is a widely used benchmark in the field of machine learning, particularly for digit recognition tasks. It consists of 1,797 instances, each represented by 64 attributes. Each attribute corresponds to an 8x8 pixel image of a handwritten digit, where pixel values are integers in the range of 0 to 16. This dataset is a subset of the UCI ML hand-written digits datasets, derived from the National Institute of Standards and Technology (NIST) preprocessing programs, which extracted normalized bitmaps of handwritten digits from preprinted forms. The digits belong to 10 classes, each representing a digit from 0 to 9.

B. Performance Analysis

The initial analysis explores the effect of the maximum tree depth hyper-parameter on the overall performance of the RF algorithm for the respective datasets. All other parameters, namely the number of decsion trees in the RF, number of features used for determining optimal split, and bag size, are fixed. For each value of maximum tree depth explored, a total of 30 experiments are run, after which the average out of bag score is used to evaluate each model. The minimum tree depth is simply set to one, which is the smallest possible tree. However, selecting the maximum tree depth is more challenging, since it has no upper bound. For each dataset, a simple early stopping condition is used. If the performance does not increase for five consecutive increments of maximum depth, then stop. The value obtained from the stopping criteria is simply rounded up to the nearest five. Lastly, the number of features for deciding on the optimal split is set to the square root of the number of features present in the respective datasets [28].

The next analysis looks at the relationship between the maximum tree depth and the number of DTs in the RF. The same range for maximum tree depth is used as described for the initial analysis. For each value of maximum depth, different values for the number of DT are explored, starting at 20, and incrementing with step sizes of 20 up to 200.

C. Statistical Analysis

The following statistical analysis procedure was conducted to analyse the overall performance for each of the final expert models obtained, as well as the standard GPDT and BGP algorithm For each of the algorithms the Friedman test, with significance level (α) of 0.05, was conducted to determine if any significant dfferences existed between the final mean test accuracies. In the case that the null hypothesis is rejected, the Holm and Nemenyi post-hoc procedures are used to determine which algorithms are significantly different from one another. The Iman-Davenport correction is also aplied to the Friedman test to address overly conservative behaviour.

V. Results

VI. Conclusion

References

[1] I. D. Mienye and Y. Sun, "A survey of ensemble learning: Concepts, algorithms, applications, and prospects," *IEEE Access*, vol. 10, pp. 99129–99149, 2022.

[2] B. Naderalvojoud and T. Hernandez-Boussard, "Improving machine learning with ensemble learning on observational healthcare data," *AMIA Annu Symp Proc*, vol. 2023, pp. 521–529, Jan 11 2024.

[3] J. Zhou, Z. Jiang, F. L. Chung, and S. Wang, "Formulating ensemble learning of svms into a single svm formulation by negative agreement learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, pp. 6015–6028, Oct 2021.

[4] S. Mishra, K. Shaw, D. Mishra, S. Patil, K. Kotecha, S. Kumar, and S. Bajaj, "Improving the accuracy of ensemble machine learning classification models using a novel bit-fusion algorithm for healthcare ai systems," *Frontiers in Public Health*, vol. 10, p. 858282, 2022.

[5] Y. Sun, Z. Li, X. Li, and J. Zhang, "Classifier selection and ensemble model for multi-class imbalance learning in education grants prediction," *Applied Artificial Intelligence*, vol. 35, no. 4, pp. 290–303, 2021.

[6] G. Brown, "Ensemble learning," in *Encyclopedia of Machine Learning* (C. Sammut and G. I. Webb, eds.), Boston, MA: Springer, 2011.

[7] R. Banfeld, K. Bowyer, W. Kegelmeyer, and L. Hall, "A comparison of decision tree ensemble creation techniques," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 173–180, 2007.

[8] D. Hernandez-Lobato, G. Martinez-Munoz, and A. Suarez, "How large should ensembles of classifiers be?," *Pattern Recognition*, vol. 46, pp. 1323–1336, 2013.

[9] T. Oshiro, P. Perez, and J. Baranauskas, "How many trees in a random forest?," in *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, (Berlin), pp. 154–168, Springer, 2012.

[10] P. Probst and A.-L. Boulesteix, "To tune or not to tune the number of trees in a random forest?," *Journal of Machine Learning Research*, vol. 18, pp. 1–18, 2018.

[11] E. A. Freeman, G. G. Moisen, J. W. Coulston, and B. T. Wilson, "Random forests and stochastic gradient boosting for predicting tree canopy cover: comparing tuning processes and model performance," *Canadian Journal of Forest Research*, vol. 46, no. 3, pp. 323–339, 2015.

[12] B. F. F. Huang and C. B. Paul, "The parameter sensitivity of random forests," *BMC Bioinformatics*, vol. 17, p. 331, 2016.

[13] L. Rokach and O. Maimon, *Classification Trees*, pp. 149–174. Boston, MA: Springer US, 2010.

[14] G. Potgieter, "Mining continuous classes using evolutionary computing," Master's thesis, University of Pretoria, 2006.

[15] S. B. Kotsiantis, "Decision trees: a recent overview," *Artificial Intelligence Review*, vol. 39, pp. 261–283, 2013.

[16] H. Blockeel, L. Devos, B. Frénay, G. Nanfack, and S. Nijssen, "Decision trees: From efficient prediction to responsible ai," *Frontiers in Artificial Intelligence*, 2023.

[17] J. N. Morgan and J. A. Sonquist, "Problems in the analysis of survey data, and a proposal," *Journal of the American Statistical Association*, vol. 58, pp. 415–434, 1963.

[18] G. V. Kass, "An exploratory technique for investigating large quantities of categorical data," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 29, no. 2, pp. 119–127, 1980.

[19] L. Breiman, J. Friedman, C. Stone, and R. Olshen, *Classification and Regression Trees*. Taylor & Francis, 1984.

[20] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, 1986.

[21] J. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann series in machine learning, Elsevier Science, 1993.

[22] J. D. Kelleher, B. MacNamee, and A. D'Arcy, *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies*. Cambridge, Massachusetts: The MIT Press, 2015. Print.
[23] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Wadsworth International Group, 1984.
[24] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
[25] Y. Amit and D. Geman, "Shape quantization and recognition with randomized trees," *Neural Computation*, vol. 9, pp. 1545–1588, 1997.
[26] Z. Sun, G. Wang, P. Li, H. Wang, M. Zhang, and X. Liang, "An improved random forest based on the classification accuracy and correlation measurement of decision trees," *Expert Systems with Applications*, vol. 237, p. 121549, 2024.
[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[28] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer Series in Statistics, New York, NY, USA: Springer New York Inc., 2001.