Pruning a Random Forest by Learning a Learning Algorithm

Kumar Dheenadayalan, G. Srinivasaraghavan, and V.N. Muralidhara

International Institute of Information Technology, Bangalore, India
d.kumar@iiitb.org, {gsr,murali}@iiitb.ac.in

Abstract. Ensemble Learning is a popular learning paradigm and finds its application in many diverse fields. Random Forest, a decision tree based ensemble learning algorithm has received constant attention in the research community due to its ability to learn complex rules and generalize well for unknown data. Identifying the number of base classifiers (trees) required for a particular dataset is one of the key questions addressed in this paper. Statistical analyses of individual base classifiers are carried out to prune the ensemble model without compromising the classification accuracy of the model. Learning the learned model, i.e., learning the statistics of the forest in its entirety along with the information available in the dataset can reveal the optimal thresholds that should be used to prune an ensemble model. Experimental results reveal that on an average 78% of the trees were pruned on 26 different datasets obtained from the UCI repository. The impact of pruning was positive with 22 out of 26 datasets showing equal or better classification accuracy in comparison with the Classical Random Forest algorithm.

Keywords: Ensemble learning · Random forest · Pruning · Matthews correlation coefficient · Meta-learning

1 Introduction

Ensemble Learning is a class of supervised learning algorithms where a single base learning algorithm is used to train multiple hypotheses (classifiers) for learning the same task [5,14]. Given a dataset $\mathcal{D}$ belonging to the input space $\mathcal{X}$ and output space $\mathcal{Y}$, a supervised learning algorithm will try to approximate an unknown target function f from a possible hypothesis set $\mathcal{H}$ to output $h : \mathcal{X} \rightarrow \mathcal{Y}$. Typically, an ensemble learning algorithm consists of growing and combining phases. Growing phase involves generation of k different classifiers. Techniques like Bagging or Boosting are using to grow individual trees as these techniques have shown to improve the accuracy of a learning algorithm in both theoretic and empirical sense [3,7]. Both these techniques re-sample the given dataset $\mathcal{D}$ to generate different training instances for individual classifiers. In the combining phase the outputs of the individual classifiers are combined using a function $e : (h_1(X), \ldots h_k(X))$. Majority prediction or weighted averaging is a popular function used in the combining phase in machine learning literature.

Though there are many popular ensemble learning algorithms, the two algorithms that have set the benchmark in ensemble learning are Adaboost (adaptive boosting) and Random Forest algorithms. Both work on the principle of achieving a strong classifier by combining the outputs of many weak classifiers [4,6]. There have been several attempts in the past to improve upon the classic version of Random Forest. Attempts to improve the accuracy of classification can be broadly divided into two categories:

- Pruning of individual trees in the forest [11,12,20,21]
- Weighing individual trees [18]

We attempt to improve the accuracy of a Random Forest through selective pruning based on the statistical measures of individual trees.

The motivation for pruning of Random Forest arises from our past work on classifying the response state of a storage system where close to 100,000 features are available. Observations on the prediction accuracy of individual trees in the forest have revealed that a number of trees did not generalize well in the live testing environment. A statistical measure of Matthews Correlation Coefficient (MCC) was evaluated for individual trees. As MCC considers the True Positives and True Negatives of the classifier in evaluating the coefficient, it can be used effectively to identify the trees with maximum accuracy.

A p^{th} percentile of MCC was used as the threshold to prune the forest. Trees not satisfying the threshold were pruned to come up with a subset of base classifiers. The selective pruning has shown marked improvement in the classification accuracy on relatively large test sets. The extent of pruning ranges from 33% to 95% on various standard datasets that have been traditionally used in machine learning literature. There are significant portions of the ensemble learner, which can be eliminated to obtain similar or better classification accuracy without compromising with the intra-correlation or strength of individual trees.

The value of p was initially calculated iteratively. To explore the possibility of guiding the user towards near optimal value of p, we collected statistical data related to the un-pruned Random Forest. Along with these, statistics related to the dataset were also collected. A total of 39 statistics were collected for each of the 9 different percentile measures ranging from 55 to 95 to form a dataset $\mathcal{D}'$ for learning a learning algorithm.

We built a new Random Forest ($\mathcal{RF}^g$), which we will refer to as the guiding Random Forest with 10 trees by considering the percentile value p as the class attribute. The guiding Random Forest $\mathcal{RF}^g$ built using $\mathcal{D}'$, was able to suggest the near optimal percentile value of MCC that should be used for pruning a Random Forest built on any new dataset. Hence, we learn the statistics of a number of Random Forests to understand the behavior of the forest for a particular dataset and try to identify the percentile value of MCC that would provide the best pruned forest.

In section 2 we cover the past literature on pruning followed by an overview and analysis of the proposed pruning and learning of learning algorithm in Section 3. Evaluation of the proposed method along with implementation details are discussed in Section 4 followed by the conclusion in Section 5.

2 Literature Survey

A Random Forest classifier uses bagging or bootstrap aggregation to construct k different tree based classifiers $\{h_1(x,\Theta_1),...,h_k(x,\Theta_k)\}$. Bagging involves sampling-with-replacement of $|\Theta_i|$ random vectors that are independently and identically distributed from the given dataset $\mathcal{D}$. Each tree in the Random Forest casts a vote to one of the classes with some probability, which is aggregated to predict the class. The final ensemble model generated can be more accurate than its individual components if the necessary and sufficient conditions, namely high strength and diversity of individual classifiers are guaranteed [5,8].

Strength of the classifiers is defined as the expected value of the margin function,

$$s = E_{X,Y}(mr(X,Y)) \tag{1}$$

where the margin function is estimated as shown in Equation 2

$$mr(X,Y) = P_\Theta(h(X,\Theta) = Y) - max_{j \neq Y} P_\Theta(h(X,\Theta) = j) \tag{2}$$

The margin measures, the difference between votes acquired for the right class and the maximum vote acquired for any other class. The expected value of all the classifiers indicate the strength of the classifier. Large values of strength indicates higher confidence in the predicted class.

The diversity of trees within the forest is essential for generalization as each tree will able to model diverse areas of the input space. Classifiers are diverse if their predictions (or errors) vary on individual data points. The diversity is measured by evaluating the correlation (ρ) of predictions by individual trees in the forest. Lower the correlation, higher is the diversity. Breiman suggests that the ratio of correlation to squared strength gives a good estimation of the generalization error. Hence, each tree evaluated should have low correlation and high strength.

Reordering of classifiers and aggregating a sub-ensemble was proposed in [11] with 15% to 30% of the initial ensemble size being used in final aggregation. Similar approach is proposed in [12] with boosting technique being used for ordering the classifier. Both use a fixed percentage of trees in the sub-ensemble. It seems unreasonable however to assume a fixed fraction of the original ensemble to form a pruned forest. The percentages used in the works referred to above seem arbitrary, and their use for all datasets does not seem justified. Ideally, the best sub-ensemble size should be a function of the dataset. Our work focuses on identifying the best set of trees that can be aggregated to generalize well to the dataset irrespective of the initial size of the ensemble.

An attempt to prune the forest based on the margin function has been suggested in [19]. The pruning strategy is based on the margin function, which is part of the internal estimates evaluated during the construction of the Random Forest. The list of trees is ranked based on different margin metrics after which the least important tree is eliminated iteratively till the number of trees reduces to 20. The drawback with this approach is that the pruning target is pre-decided, which may or may not be the ideal size of sub-ensemble of trees.

Our evaluation on multiple datasets showed that the number of trees ideal for each dataset depends on multiple factors related to data and the forest constructed. The dataset has a direct influence on the statistics of the forest, and this varies with different datasets. This, in turn, influences the choice of the best subset of trees required for optimal class accuracy. We effectively address this issue in this paper by automatically determining the forest size and the collection of trees in the pruned forest based on the dataset.

Authors in [20] explore a correlation based pruning by considering both similarity and prediction accuracy of trees. They suggest that pruning of the forest based on the accuracy of individual trees performs slightly better in identifying the sub-ensemble forest compared to the similarity of trees. Other approaches, for example in [16], include mixing of different metrics in the construction of the Random Forest namely, Gain ratio, Gini index, minimum description length and many more. The results for the forest with several metrics shows a marginal increase in the prediction accuracy but utilize the complete forest by incorporating different types of diversity.

The experiments in past literature do not consider the two key aspects that need to be preserved for effective classification, namely the strength and the correlation. Another important aspect that has to be considered is the imbalance of classes in data. Imbalanced datasets have a tendency to mis-classify instances that belong to the class with low class probability in the training set. Pruning should not aggravate this tendency of misclassification of imbalanced data.

Collecting statistics related to the model are termed as meta-data of the model. There is research related to meta-learning that can help choose the best algorithm suited for a given problem. Meta-learning involves learning of performance of a base learning algorithm for an application [2,17], which is in contrast to the learning of meta-data to choose the best parameters for a base learner. The latter is being considered in this paper with the meta-data being the statistics of Random Forest for different applications. The near optimal percentile value of MCC to be used for pruning the forest has to be learned by using the meta-data. We achieve this by learning the meta-data using the base learning algorithm itself.

2.1 Matthews Correlation Coefficient

Commonly used evaluation measures including Recall, Precision, F-Measure are biased regarding label and population prevalence as argued in [9]. Experiments reveal that using such evaluation measures can appear to perform better for certain datasets but performs worse in the objective sense of Informedness or Markedness.

Informedness is a measure to quantify how informed a predictor is for the specified condition and specifies the probability that a prediction is informed in relation to the condition (versus chance) [9]. Markedness quantifies how marked a condition is for the specified predictor and specifies the probability that a condition is marked by the predictor (versus chance) [9].

The dependence of Matthews Correlation Coefficient with Informedness and Markedness that forms an unbiased accuracy measure was established in [15]. For our analysis in a Classification setting, Matthews' correlation Coefficient can be evaluated from a contingency matrix using the following formula

$$MCC = \frac{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}{TP \times TN - FP \times FN} \tag{3}$$

where TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative

MCC [13] provides an unbiased measure of accuracy that is widely used in bioinformatics [1]. The same measure is used in our analysis to prune out trees of the Classical Random Forest. Systematical pruning of the weak classifiers based on MCC measure can help in enhancing the classification accuracy without compromising with the generalization error.

WEKA (Waikato Environment for Knowledge Analysis) is a popular open source machine learning workbench targeted towards domain specialists who can directly apply the existing Machine Learning techniques to real world problems in various domains. The workbench though a couple of decades old is actively being updated with the latest state-of-the-art Machine Learning techniques. We use the WEKA's implementation of Random Forest with bagging as the base implementation for Classical Random Forest algorithm and compare it with the proposed statistically pruned Random Forest.

3 Pruning a Random Forest

3.1 Modeling

Given a dataset $\mathcal{D}$, Random Forest is built in a classical way by choosing a random subset of the training set to build a tree. This process is repeated k times to build a Random Forest of k trees. As part of the internal estimates to guide the generalization measures, a part of the training set is kept aside for each tree to estimate the Out Of Bag (OOB) error rate.

$$\vartheta_k = \Theta \setminus \Theta_k \tag{4}$$

This OOB data instances (ϑ_k) are utilized to evaluate the weighted MCC for each tree(k) individually.

The prediction or vote for OOB instances by each tree are evaluated and compared against the true class value.

$$\forall_{i \leq k} \forall_{x_j \in \vartheta_k} \{T_i(x_j)\} \tag{5}$$

The statistics for each individual tree like TP, TN, FP, FN are calculated, and the weighted MCC is noted. $classW_c$ in (6) is the class probability of each class c in the Test set.

$$\forall_{i \leq k} wMCC_i = \sum_{c \in C} MCC_c \times classW_c \tag{6}$$

Iterative pruning of the Random Forest is carried out based on various percentiles of weighted MCC of trees in the Classical Random Forest (CRF). All the trees with weighted MCC below the reference percentile MCC score will be pruned off.

$$PRF = \begin{cases} PRF \cup T_i & \text{if } wMCC_i > wMCC(p) \\ \text{prune } T_i & \text{otherwise} \end{cases} \tag{7}$$

The Pruned Random Forest (PRF) is tested with a separate test set, and its test accuracy along with the Area Under the Curve (AUC) for Receiver Operating Characteristic (ROC) is noted. The value of percentile is iterated to see its impact on the number of trees and the test accuracy until optimal test set accuracy is obtained.

Identifying the optimal number of trees for a dataset is possible with the process mentioned above. We now examine the two key conditions of maintaining higher strength for the classifiers and lower correlation among the individual classifiers. The strength of the classifier defined by (1) measures the expected maximum margin (true class predictions) of the classifier over a dataset. wMCC is used in the proposed pruning model, and this can be used as a weaker replacement of strength. This is true because wMCC is directly proportional to the difference in true predictions and false predictions. The numerator of MCC measure is similar but a weaker representation of the margin function shown in (2). Hence, the trees with higher strength will be the once that will be retained. This is experimentally verified, and a high correlation value of +0.62 was observed between Optimal wMCC and strength of classifiers. MCC being an unbiased measure of estimating the accuracy of a model gives another advantage to shortlist trees that are best in the forest in an unbiased probabilistic sense. We experimentally show that this pruning actually identifies trees, which have lower correlation for a number of different datasets especially binary classification problems.

Figure 1 shows the process of varying the percentile and the corresponding number of trees retained after pruning for autos dataset. As the percentile varies, the number of trees pruned also varies. The accuracy of the test set during this variation is shown in Figure 2. The best accuracy of 83.61 was obtained for a 55^{th} percentile score of wMCC with 45 trees. Classical Random Forest with 100 trees could achieve only 78% classification accuracy giving an indication of the effect of pruning based on a wMCC.

3.2 Learning a Learning Algorithm

There are at least 5 iterations that can be carried out to identify the percentile score for wMCC, which facilitates the retention of an optimal number of trees. We try to eliminate this by learning the statistics of a learning algorithm, i.e. Classical Random Forest in the current scenario. If we can learn the properties of the dataset and statistics of the Classical Random Forest that is constructed from the dataset, identifying the percentile for which wMCC value can generate an optimal number of trees can be determined.

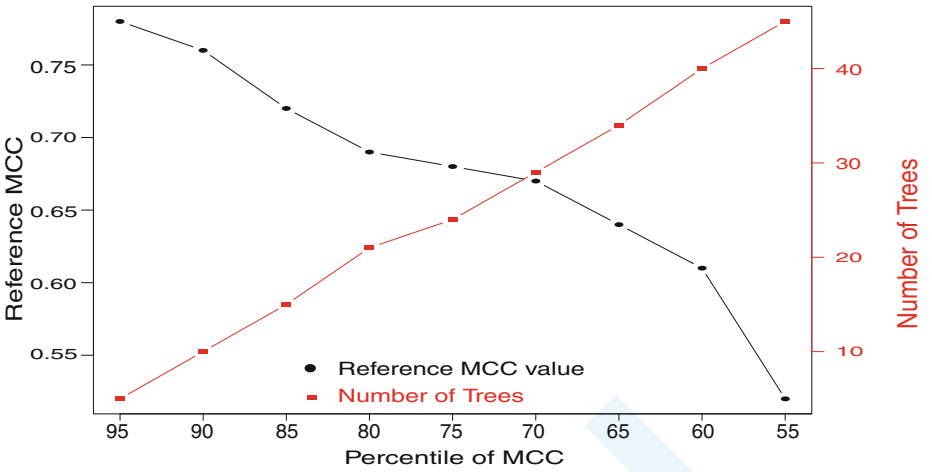

Fig. 1. Variation in the number of trees with varying Matthews Correlation Coefficient.

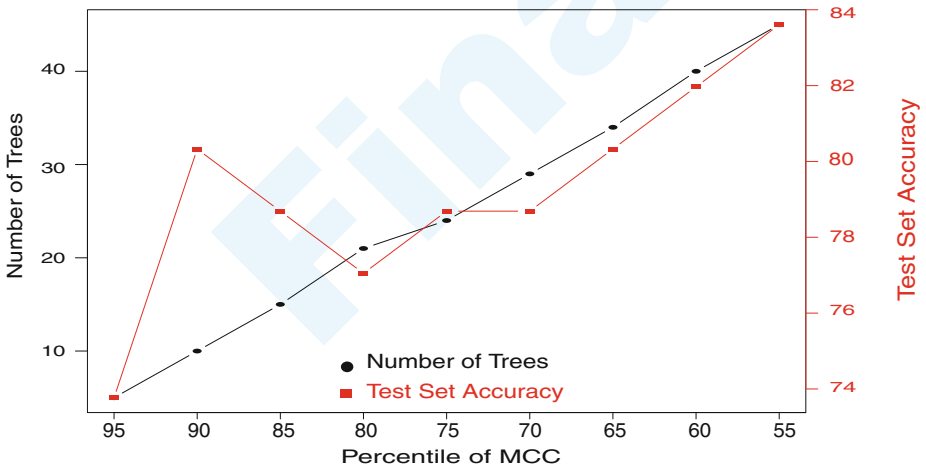

Fig. 2. Variance in the Test Set Classification Accuracy with varying number of trees in the Pruned Forest.

During the iterative process of varying the percentiles of wMCC, a number of statistics related to the CRF and dataset were collected. The table of statistics collected are listed in Table 1. WEKA's Random Forest has a convenient way of extracting these statistics in each iteration through the Evaluation class implementation. Statistics listed in Table 1 are collected for every dataset during each iteration of percentile value. The optimal percentile score for each dataset is used as the class variable in the dataset to learn a learning algorithm.

Some key statistics that are of interest are class complexity stats, Kononenko & Bratko Information [10], Kappa Statistics . . . , which gives an insight into the

Table 1. Statistics collected for each Classical Random Forest

CorrectClassification_Count	Correct Classification_Percentage	IncorrectClassification_Count
IncorrectClassification_Percent	Kappa_statistic	Total Cost
Average_Cost	Relative_Info_Score	Info_Score_bits
Info_Score_bits_per_instance	Correlation_coefficient	Class_complexity_order_0_bits
Class_complexity_order_0_bits_per_instance	Class_complexity_scheme_bits	Class_complexity_scheme_bits_per_instance
Complexity_improvement_Sf_bits	Complexity_improvement_Sf_bits_per_instance	Mean_absolute_error
Root_mean_squared_error	Relative_absolute_error	Root_relative_squared_error
Coverage_0.95level	Mean_rel_region_size_0.95level	UnClassified_Instances
Total Number_of_Instances	Ignored_Class_Unknown_Instances	Total Number_Of_Classes
Total_Number_Of_Attributes	Training_AUCROC	Training_Accuracy
Weighted_Precision	Weighted_Recall	Weighted_TruePositive
Weighted_FalsePositive	Weighted_FMeasure	Weighted_MCC
Weighted_PRCArea	OOB_Error_Rate	Optimal_Percentile(Class Variable)

class distribution, the information available in the dataset and to what extent were they used in building the Random Forest. These statistics by itself have a lot of information that can help in analyzing the number of trees required for good classification accuracy. We use the data collected in Table 1 as our dataset $\mathcal{D}'$ and learn the learning algorithm through another Random Forest ($\mathcal{RF}^g$). This can turn out to be a useful way of guiding the pruning algorithm by predicting the optimal percentile for wMCC score, which should be used as a thresholding measure. It can eliminate multiple iterations of pruning and evaluating the test set. UCI repository has a number of datasets for which statistics can be collected, and $\mathcal{RF}^g$ can be built. The entire iterative process of building $\mathcal{D}'$ from a dataset repository is explained in Algorithm 1.

4 Results

WEKA workbench was used to implement the proposed pruning method. The bagging class was modified and the pruning methodology was embedded during the bagging process. WEKA's current implementation of Random Forest, which we call the Classical Random Forest (CRF) is used as the reference that will be compared with our Pruned Random Forest (PRF). WEKA package from SourceForge contains datasets obtained from the UCI repository. These datasets have been used widely in the original paper on Random Forest [4] and other subsequent work on pruning.

The experimental setup for PRF consists of 26 datasets. For a particular percentile, PRF was built for each dataset for 10 iterations. The percentile values were varied from 95 to 55 giving a total of 9 (percentile values) $\times$ 10 (iterations) $=$ 90 instances for each dataset. The results reported here are the average values over ten iterations for each dataset. Results for CRF are obtained for 26 datasets, over 10 iterations and the results are averaged over these ten iterations. Both CRF and PRF are executed with a 70:30 split of training and testing samples from the dataset. If the dataset has an explicit test set (example: spect), then the test set is used instead of splitting the dataset into training and testing set. During the process of data collection, statistics from Table 1 is collected for each dataset while running PRF to build $\mathcal{D}'$.

Algorithm 1. Learning a learning algorithm

procedure PRUNE($trainIndex, testIndex$)
  for all $D \in repository$ **do**
    $trainData \leftarrow \mathcal{D}[trainIndex,]$
    $testData \leftarrow \mathcal{D}[testIndex,]$
    for all $p \in [55, 60, 65, 70, 75, 80, 85, 90, 95]$ **do**
      $i \leftarrow 10$
      $\mathcal{PRF}_{best} \leftarrow NULL$
      while $i > 0$ **do**
        Build $\mathcal{RF}$
        for all $T_i \in \mathcal{RF}$ **do**
          Evaluate $wMCC(p)_i$
          Update PRF using Equation (7)
        end for
        $\mathcal{PRF}_{err} \leftarrow test(\mathcal{PRF}_{err}, testData)$
        extract $\mathcal{D}'$
        if $\mathcal{PRF}_{best} > \mathcal{PRF}_{err}$ **then**
          $\mathcal{PRF}_{best} = \mathcal{PRF}$
        end if
        $i \leftarrow i - 1$
      end while
    end for
  end for
  Build $\mathcal{RF}^{\mathcal{G}}$
end procedure

Table 2 shows the results for PRF and CRF along with the details of the dataset. The dataset size, number of classes and number of attributes is extracted from the dataset input file. Test accuracy for PRF is the average test classification accuracy (of ten iterations for a single percentile) for the best of 9 different percentile values. Test accuracy for CRF was the average test classification accuracy out of the 10 iterations. Optimal wMCC indicates the value of wMCC that was used to prune and obtain the test accuracy reported in PRF Test (%). PRF Size represents the size of pruned forest. The row number in the table is used as the dataset index in the subsequent figures.

Results for 26 datasets presented in Table 2 shows that PRF has equal or better classification accuracy when compared to CRF for 22 out of the 26 datasets. CRF was better than PRF by more than 1% for only two datasets. Rest of the four datasets had better accuracy with very low statistical significance. The number of trees required to achieve these results ranges from a minimum of 2 trees to a maximum of 67 trees and 21 trees on an average. The optimal percentile varies from 95 for a majority of datasets to 55 for very few datasets. Optimal wMCC values that gave the best test accuracy were also high with an average optimal wMCC value of 0.81 for 26 datasets. Pearson's Correlation Coefficient of optimal wMCC with the strength of the PRF was observed at +0.62, which indicates that pruning based wMCC actually retains trees with high strength.

Table 2. Success Rate Comparison of Pruning with Classical Random Forest

Dataset Index	Dataset Name	Classes	Attributes	Total Instances	Optimal Percentile	Optimal wMCC	PRF Test(%)	CRF Test(%)	PRF Size
1	anneal.ORIG	6	39	898	95	0.94	**94.42**	93.68	5
2	autos	7	26	205	55	0.52	**83.61**	78.68	45
3	balance-scale	3	5	625	90	0.82	77.01	**79.14**	9
4	breast-cancer	2	10	286	70	0.68	**71.00**	69.76	29
5	breast-w	2	10	699	85	0.96	**96.19**	95.71	23
6	colic	2	23	368	95	0.88	**85.45**	83.63	4
7	contact-lenses	3	5	24	95	0.83	**42.86**	42.85	2
8	credit-a	2	16	690	90	0.79	85.80	**85.99**	8
9	diabetes	2	9	768	70	0.75	**77.48**	76.52	24
10	glass	7	10	214	60	0.88	77.19	**78.12**	4
11	hayes-roth	4	5	160	85	0.87	85.71	85.71	20
12	heart-c	5	14	303	80	0.74	**85.71**	84.61	27
13	heart-h	5	14	294	90	0.75	78.41	**79.54**	7
14	heart-statlog	2	14	270	70	0.86	**80.62**	79.01	30
15	hepatitis	2	20	155	75	0.85	**86.96**	86.95	24
16	hypothyroid	4	30	3772	95	0.99	**99.20**	99.11	5
17	ionosphere	2	35	351	60	0.89	**91.43**	90.47	44
18	iris	3	5	150	95	0.81	**95.56**	95.55	5
19	labor	2	17	57	90	0.83	**94.12**	94.11	6
20	lymph	2	19	148	90	0.7	**92.95**	88.63	10
21	mushroom	2	23	8124	95	1	100.00	100.00	67
22	sonar	2	61	208	70	0.9	**84.84**	82.25	27
23	spect	2	23	267	85	0.65	**71.12**	68.44	15
24	splice	3	62	3190	60	0.73	**94.25**	91.22	42
25	vowel	11	14	990	55	0.8	**94.28**	93.60	45
26	zoo	7	18	101	75	0.81	**93.33**	80.00	25

To see if PRF actually generalizes as well if not better than CRF, we evaluated the strength and mean correlation for each dataset. Any good Random Forest model requires the forest to have high strength and low mean correlation. Plot of mean correlation and strength for CRF is shown in Figure 3. We observe that 18 out of 26 datasets have strength higher than correlation but for a number of datasets, the strength and correlation are very close to each other. Figure 4 shows strength and correlation for PRF measured on all the 26 datasets. Even for PRF, 18 out of 26 have higher strength than correlation with wider gap compared to CRF, which is a desirable property for effective generalization.

Finally, we compare the $\frac{c}{s^2}$ ratio of CRF and PRF for each dataset in Figure 5. c is the mean correlation of trees in the fores and s^2 is the squared strength of the forest. The desired property of lower $\frac{c}{s^2}$ is better achieved through PRF than CRF

for 23 out of 26 datasets. Average improvement in $\frac{c}{s^2}$ ratio for 26 datasets is 1.82. The upper bound on the generalization error as derived in [4] is given by (8) where ρ is the mean correlation of the trees within the forest.

$$\frac{\rho \times (1 - s^2)}{s^2} \tag{8}$$

The plot for the upper bound on the generalization error of 26 datasets for PRF and CRF is shown in Figure 6 which is a scaled version Figure 5.

The value of optimal wMCC obtained by the optimal percentile has a big effect on the classification accuracy for a number of datasets. Many datasets are sensitive to the value of optimal wMCC and the number of trees in PRF, but some datasets have shown minimum to no variation in the classification accuracy with a change in the percentile value. This can be observed in Figure 7 where high variance in test accuracy is observed for 9 different percentile values.

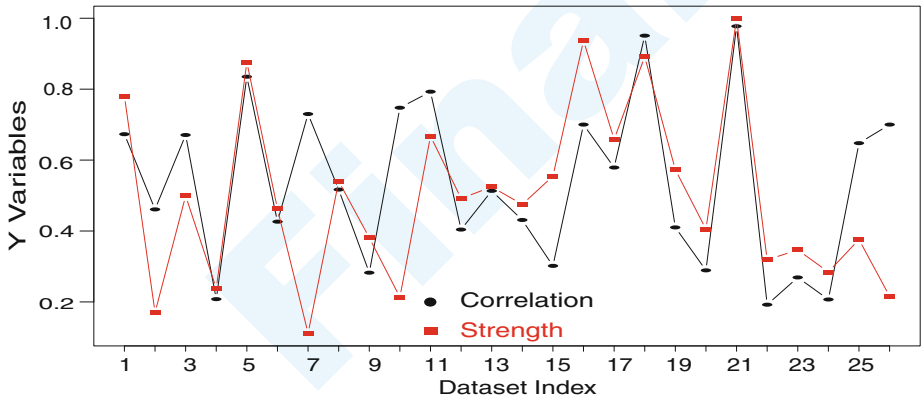

Fig. 3. Correlation and Strength of the Classical Random Forest.

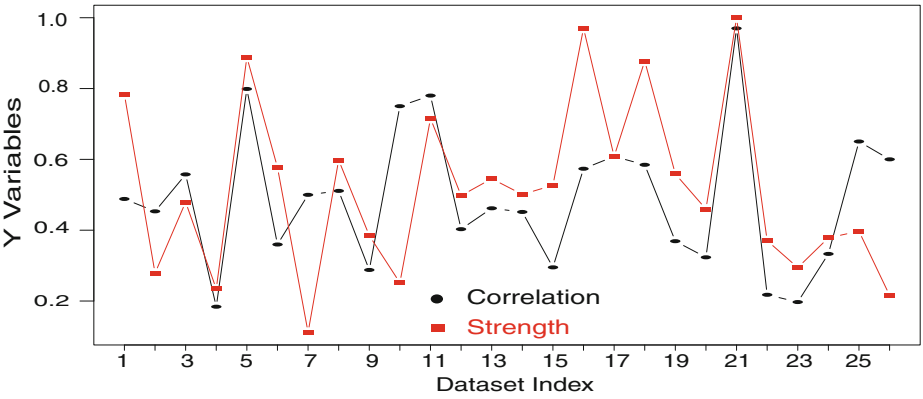

Fig. 4. Correlation and Strength of the Pruned Random Forest.

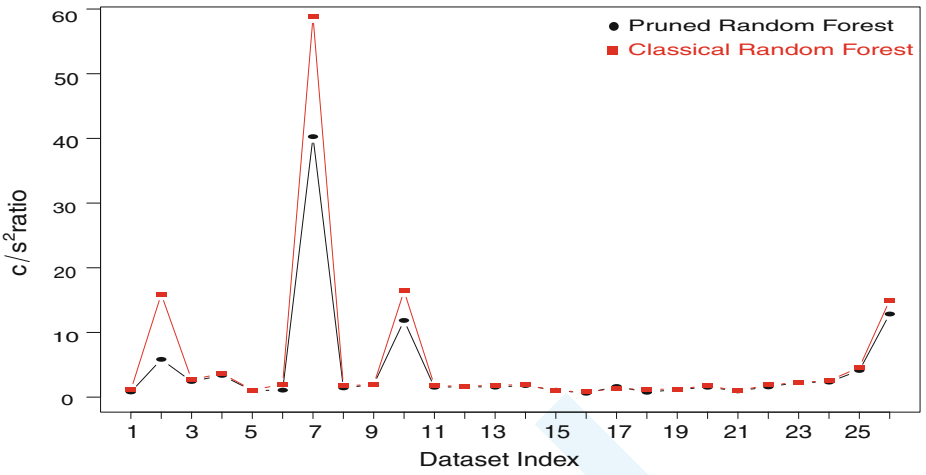

Fig. 5. $\frac{c}{s^2}$ ratio for Classical Random Forest and Pruned Random Forest.

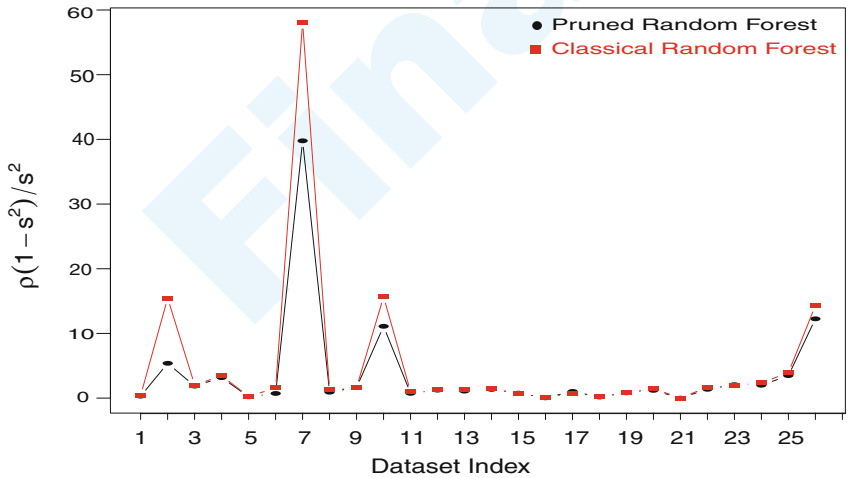

Fig. 6. $\frac{\bar{\rho}(1-s^2)}{s^2}$ ratio for Classical Random Forest and Pruned Random Forest.

Hence, it becomes increasingly important to identify the optimal percentile as achieved by learning a learning algorithm. Dataset $\mathcal{D}'$ containing 1800 instances obtained from 20 datasets with the optimal percentile value as the class variable is used to train a guiding Random Forest. $\mathcal{RF}^g$ was used to predict the optimal percentile value for the remaining 6 datasets. $\mathcal{RF}^g$ was able to guide the optimal percentile value for 4 datasets, while percentile value for 2 datasets was off by 5 percentile value. The effect of this was the classification accuracy of the test set was 0% and 1.2%.

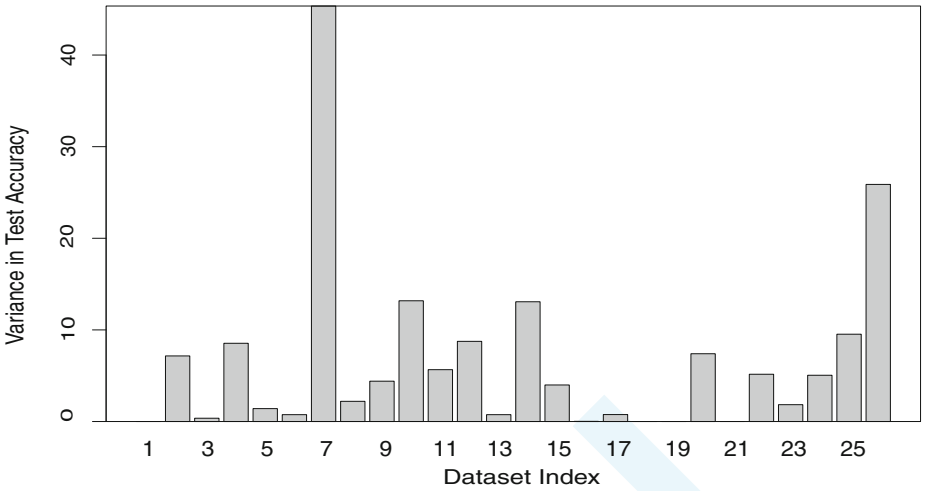

Fig. 7. Variance in Testset accuracy for Pruned Random Forest.

5 Conclusion

Pruning of an ensemble classifier like Random Forest without compromising on the generalization goals is presented in this paper. MCC was used to have an unbiased measure of accuracy for individual classifiers, which has effectively retained trees with relatively high strength. A high percentage of trees were pruned off indicating that a lot of redundant trees were part of CRF. When used on a large scale datasets or in image processing, pruned forests will help in reducing the test time with a possible increase in the classification accuracy.

By learning a learning algorithm, we were able to uncover a possible guiding mechanism for pruning. The guiding Random Forest was able to learn the statistics accurately indicating that there are statistics present within the dataset and un-pruned Random Forest that can decide on the optimal number of trees. A more theoretical process of uncovering the relation among the features of $\mathcal{D}'$ will be the next step of our work.

References

1. Baldi, P., Brunak, S., Chauvin, Y., Andersen, C.A.F., Nielsen, H.: Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics **16**(5), 412–424 (2000)
2. Brazdil, P., Giraud-Carrier, C., Soares, C., Vilalta, R.: Metalearning: Applications to Data Mining, 1st edn. Springer Publishing Company, Incorporated (2008)
3. Breiman, L.: Bagging predictors. Machine Learning **24**(2), 123–140
4. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
5. Dietterich, T.G.: Ensemble methods in machine learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000)

6. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. **55**(1), 119–139 (1997)
7. Freund, Y., Schapire, R.E.: A Short Introduction to Boosting. Journal of Japanese Society for Artificial Intelligence **14**(5), 771–780 (1999)
8. Hansen, L., Salamon, P.: Neural network ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence **12**(10), 993–1001 (1990)
9. Hu, B.G., He, R., Yuan, X.T.: Information-theoretic measures for objective evaluation of classifications. Acta Automatica Sinica **38**(7), 1169–1182 (2012)
10. Kononenko, I., Bratko, I.: Information-based evaluation criterion for classifier's performance. Mach. Learn. **6**(1), 67–80 (1991)
11. Martínez-Muñoz, G., Suárez, A.: Pruning in ordered bagging ensembles. In: Proceedings of the 23rd International Conference on Machine Learning, ICML 2006, pp. 609–616. ACM, New York (2006)
12. Martínez-Muñoz, G., Suárez, A.: Using boosting to prune bagging ensembles. Pattern Recogn. Lett. **28**(1), 156–165 (2007).
13. Matthews, B.W.: Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochim. Biophys. Acta **405**, 442–451 (1975)
14. Opitz, D., Maclin, R.: Popular ensemble methods: An empirical study. Journal of Artificial Intelligence Research **11**, 169–198 (1999)
15. Powers, D.M.W.: Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation. Journal of Machine Learning Technologies **2**(1), 37–63 (2011)
16. Robnik-Šikonja, M.: Improving random forests. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 359–370. Springer, Heidelberg (2004)
17. Vilalta, R., Giraud-Carrier, C., Brazdil, P.: Meta-learning - concepts and techniques. In: Data Mining and Knowledge Discovery Handbook, pp. 717–731. Springer, Boston (2010).
18. Winham, S.J., Freimuth, R.R., Biernacka, J.M.: A weighted random forests approach to improve predictive performance. Statistical Analysis and Data Mining **6**(6), 496–505 (2013)
19. Yang, F., Hang Lu, W., Kai Luo, L., Li, T.: Margin optimization based pruningfor random forest. Neurocomputing **94**, 54–63 (2012)
20. Zhang, H., Wang, W.: Search for the Smallest Random Forest, pp. 381–388 (2009)
21. Zhou, Z.H., Tang, W.: Selective ensemble of decision trees. In: Wang, G., Liu, Q., Yao, Y., Skowron, A. (eds.) Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing. LNCS, vol. 2639, pp. 476–483. Springer, Heidelberg (2003)