Maximizing Tree Diversity by Building Complete-Random Decision Trees

Fei Tony Liu[1], Kai Ming Ting[1], and Wei Fan[2]

[1] School of Computing and Information Technology,
Monash University Churchill, Victoria 3842 Australia
{Tony.Liu, KaiMing.Ting}@infotech.monash.edu.au
[2] IBM T.J. Waston Research, Hawthorne, NY 10532
weifan@us.ibm.com

Abstract. One of the ways to lower generalization error of decision tree ensemble is to maximize tree diversity. Building complete-random trees forgoes strength obtained from a test selection criterion. However, it achieves higher tree diversity. We provide a taxonomy of different randomization methods and find that complete-random test selection produces diverse trees and other randomization methods such as bootstrap sampling may impair tree growth and limit tree diversity. The well accepted practice in constructing decision trees is to apply bootstrap sampling and voting. To challenge this practice, we explore eight variants of complete-random trees using three parameters: ensemble methods, tree height restriction and sample randomization. Surprisingly, the most accurate variant is very simple and performs comparably to *Bagging* and *Random Forests*. It achieves good results by maximizing tree diversity and is called ***Max-diverse Ensemble***.

1 Introduction

Random tree ensembles introduce different random elements to construct diversified decision trees. For classification problems, results from these trees are combined by an ensemble method to produce the final prediction. *Random Decision Trees* [8] is one that is constructed without conventional test selection criteria, which questions the utility of these heuristics that are widely employed in many decision tree learning algorithms. *The underlying argument is that they are effective to compute accurate single trees but there is no guarantee on the final accuracy of a tree ensemble.*

As it stands, there is no creditable report known to us that extensively analyses and compares complete-random trees with other decision tree ensembles. This paper aims to explore complete-random trees and compare them with *Bagging* [3] and *Random Forests* [5] which are widely accepted and use techniques such as randomized feature selection, bootstrap sampling and voting. The fundamental objective of randomization in tree construction is to create diversity. After all, there is no point in combining a forest of identical trees. Section 2 of

T.B. Ho, D. Cheung, and H. Liu (Eds.): PAKDD 2005, LNAI 3518, pp. 605–610, 2005.

this paper discusses how increasing tree diversity can lower the generalization error of tree ensembles. Since there are many randomization methods, a systematic framework to characterize each method is necessary to guide the research in this area. A taxonomy is provided in section 3 and the focus of our study is *sample randomization* and *complete-random test selection*. Section 4 describes the random tree learning process, section 5 provides the experimental settings and results, and follows by conclusions in the last section.

2 Tree Diversity

In Breiman's analysis [5] on strength and correlation, he gives an upper bound on the generalization error $PE^* \leq \overline{\rho}\frac{(1-s^2)}{s^2}$, where s is the strength of the set of trees and $\overline{\rho}$ is the mean correlation. The implication of this upper bound is that no ensemble can do better than the boundary given its strength and correlation. Generally, this upper bound is applicable to classifier based ensemble, including complete-random trees the subject of this paper. Lowering PE^* can be achieved by either minimizing $\overline{\rho}$ or increasing s. Building complete-random trees forgoes strength obtained from a test selection criterion. However, it helps to achieve higher tree diversity.

3 Different Categories of Randomization

The taxonomy of tree randomizations is summarized as follows:

1. **Randomization before model induction**
   (a) Sample randomization
   e.g. *Bootstrap sampling* [3]
   (b) Feature randomization
   e.g. *Randomized Trees* [1] and *Random Subspace* [9]
   (c) Data perturbation
   e.g. *Output Smearing* and *Output Flipping* [4]
2. **Randomization during model induction**
   (a) Partial-random test selection
   e.g. *Tree Randomization* [6] and *Random Forests* [5]
   (b) Complete-random test selection
   e.g. *Random Decision Trees* [8]

This paper focuses on sub-categories (1a) *sample randomization* and (2b) *complete-random test selection* to investigate whether complete-random test selection produces good results.

4 Random Tree Ensemble

For the experiment, our implementation is based on *C4.5* release 8 [10] with modifications to cater for bootstrap sampling, multiple trees, complete-random

test selection, tree height restriction, random split point selection for continuous features and ensemble methods. The *tree height restriction* is originated from [8]. Let k be the total number of features, setting tree height to $\frac{k}{2}$ is called half height tree. Alternatively, unrestricted tree growth is called full height tree. Consider a rule or a branch in a tree, when selecting i features from k features, there are $C_i^k = \frac{k!}{i!\cdot(k-i)!}$ unique feature combinations. To use only a single value of i, $i = \frac{k}{2}$ produces the largest number of combinations. Fan et. al. [8] uses this argument as the basis to choose the tree height limit of $\frac{k}{2}$, but allowing any value of i is more desirable as it gives the maximum choice or diversity. Thus, the total number of possible unique combinations to include any value of i is $T(i) = \sum_i C_i^k$. Since $T(k) > T(\frac{k}{2}) > C_{\frac{k}{2}}^k$, setting tree height to k produces maximum diversity. For *continuous feature split point selection*, random split point is determined by randomly selecting two different sample values and assigning it as the mid point between the two. This increases the possible split points from $l-1$ to $\sum_{i=0}^{l-1} i$, l is the number of distinct feature values. Hence, it increases diversity. *Missing values* for probability averaging are handled by: 1. growing missing value branches; 2. classifying them with reduced weight $w = w_p \frac{n_{missing}}{n_{total}}$, where w_p is the classification weight from the parent node, $n_{missing}$ is the number of missing value samples and n_{total} is node size. This avoids disruption of the usual weight disseminating routine in handling missing values.

At classification phase, posterior probability estimation or class label is generated using these counts. To predict a class given a test case z, the predicted class c_p is obtained by:

1. Probability averaging, $c_p = arg\max_c(\frac{1}{N}\sum_{i=1}^{N}(w\frac{n_{h_i,c}}{n_{h_i}}))$
2. Voting, $c_p = arg\max_c(\frac{1}{N}\sum_{i=1}^{N} I(\frac{n_{h_i c}}{n_{h_i}}))$.

where N is the number of trees, $I()$ is an indicator function. Relevant to z, $n_{h_i,c}$ is the count of class c for tree h_i and n_{h_i} is the leaf size for h_i. Probability averaging are reported to cause overfitting [7]. However, it is worth noting that none of the empirical evaluations are conducted in the context of complete-random trees.

5 Experiments

There are three parameters in the experiments. The followings are the abbreviations used in the experiments:

Ensemble Methods	*Tree height restriction*	*Sample randomization*
Probability averaging	**F**ull height	**O**riginal training samples
Voting	**H**alf height	**B**ootstrap training samples

In total, there are eight possible variants from these three parameters. Each variant is represented by three letters, for example "VFO" refers to a random trees ensemble with parameters **V**oting, **F**ull height tree and **O**riginal training samples.

Table 1. The average error results are listed with asterisk(s)* indicating best error rate(s) among different methods

Data set	*size*	**PFO**	**PFB**	**PHO**	**PHB**	**VFO**	**VFB**	**VHO**	**VHB**	*Bagging*	*Random Forests*
abalone	4177	29.8	29.9	31.7	31.7	29.9	29.7	31.7	31.7	*29.1	*29.1
anneal	898	*0.9	1.7	2.6	2.9	1.0	1.8	3.7	3.9	3.2	14.8
audiology	226	19.5	*18.5	21.7	19.0	22.6	22.1	26.2	23.8	20.8	37.5
autos	690	24.3	22.9	24.4	22.4	26.7	22.0	25.3	22.9	*16.2	20.5
balance	205	13.8	13.6	13.9	13.8	13.6	*13.4	18.2	14.6	15.5	15.5
breast-w	625	*2.4	2.7	2.7	2.7	3.0	3.2	2.9	3.2	3.4	3.1
breast-y	699	25.5	25.5	*23.4	25.5	24.8	27.6	25.1	27.0	26.6	26.9
chess	286	*1.6	1.9	2.5	2.8	2.7	4.1	4.9	5.1	4.8	4.9
cleveland	20000	*41.3	41.9	43.9	44.3	42.6	43.6	43.6	44.2	43.6	42.2
coding	3196	*16.3	*16.3	*16.3	*16.3	18.7	23.9	19.2	23.8	33.5	27.5
credit-a	303	*12.3	12.6	12.9	13.5	13.3	15.1	14.1	14.2	13.2	13.6
credit-g	3186	25.3	25.7	27.1	27.3	27.3	29.2	29.0	29.3	*24.7	26.9
DNA	131	28.8	28.9	28.8	28.8	16.1	14.5	16.1	14.4	*7.1	12.9
echo	1066	32.9	34.3	*32.1	32.8	33.7	36.7	34.5	35.1	35.2	35.9
flare	1000	18.5	19.0	17.5	17.4	18.2	18.3	17.5	*17.1	17.5	17.6
glass	214	26.2	26.2	34.6	33.1	28.0	30.0	37.9	36.0	35.2	*21.9
hayes-roth	160	44.4	40.0	48.1	47.5	53.8	46.9	58.1	47.5	17.5	*17.5
hepatitis	155	15.3	15.7	16.0	15.7	17.3	*15.0	16.0	15.7	24.7	16.3
horse-colic	368	18.8	16.9	19.1	17.2	19.9	17.2	20.4	17.4	*14.2	16.3
hypothyroid	3163	2.2	2.4	4.7	4.7	2.3	2.4	4.7	4.7	*0.9	1.3
ionosphere	351	9.4	9.7	9.4	9.7	11.7	15.9	11.7	15.3	6.8	*5.7
iris	150	4.7	6.0	7.3	7.3	*4.0	5.3	11.3	11.3	6.7	6.0
led24	3200	*28.4	28.8	*28.4	29.0	36.8	37.4	36.5	36.5	28.5	29.8
liver	345	30.7	31.8	38.3	37.1	29.5	31.6	38.5	37.7	*28.1	29.2
lymph	148	15.5	15.4	15.5	*14.7	18.2	15.4	18.9	15.4	21.6	17.4
nursery	12960	*2.0	2.3	5.4	5.5	2.2	1.9	7.1	5.2	6.4	4.7
pima	768	24.7	23.6	28.6	28.1	24.3	25.4	29.4	29.5	24.7	*23.4
primary	339	56.1	55.2	*53.1	*53.1	54.6	55.8	55.5	55.2	55.5	*53.1
segment	2310	2.9	3.1	5.1	4.9	3.3	3.7	6.6	6.4	2.4	*2.3
sick	3163	7.6	7.8	9.3	9.3	8.1	8.0	9.3	9.3	2.1	3.7
solar	323	30.0	29.7	27.2	27.5	29.6	30.6	29.3	28.8	27.2	*24.7
sonar	208	*13.4	16.8	*13.4	16.8	23.5	26.9	23.5	26.9	20.1	19.6
soybean	683	6.0	5.7	5.7	5.6	5.6	6.0	6.0	5.7	6.2	*5.4
threeOf9	512	*0.2	0.8	11.3	11.1	8.2	2.7	13.1	12.9	3.3	2.2
tic-tac-toe	958	*9.4	10.4	24.6	23.8	18.8	27.5	27.5	26.2	29.4	26.4
vehicle	846	27.1	29.2	27.4	29.3	27.9	27.1	29.3	28.4	24.9	*25.2
vote	435	5.3	5.3	5.3	5.3	5.1	6.2	4.8	6.0	*4.6	4.8
waveform	5000	*14.1	14.2	14.8	14.9	14.3	14.6	14.2	14.1	16.3	14.7
wine	178	2.3	1.7	2.3	1.7	2.3	2.8	2.3	2.3	5.6	*1.1
zoo	101	*2.0	4.0	*2.0	3.0	3.0	3.0	3.0	3.9	6.9	7.9
Mean		*17.3*	*17.5*	*19.0*	*18.9*	*18.7*	*19.1*	*20.7*	*20.2*	*17.4*	*17.8*

Table 2. Summary of pairwise comparison*(wins, losses, draws)* reading from top to left. The number of significant wins and losses is bold faced, based on a sign test of 95% confidence level

	Max-diverse Ensemble PFO	PFB	PHO	PHB	VFO	VFB	VHO	VHB	*Bagging*
Random Forests	21,19,0	20,19,1	17,23,0	17,23,0	14,**26**,0	11,**29**,0	11,**29**,0	11,**29**,0	19,20,1
Bagging	23,17,0	22,18,0	20,20,0	18,22,0	18,22,0	15,25,0	12,**28**,0	13,**27**,0	
VHB	**30**,8,2	**29**,6,5	25,10,5	**29**,6,5	**29**,10,1	22,15,3	14,20,6		
VHO	**32**,6,2	**33**,6,1	**30**,4,6	**30**,14,5	**27**,8,5	21,15,4			
VFB	**28**,10,2	**27**,11,2	19,21,0	19,19,2	**26**,13,1				
VFO	**28**,11,1	23,15,2	21,18,1	16,21,3					
PHB	25,10,5	23,9,8	15,16,9						
PHO	25,6,9	25,11,4							
PFB	25,11,4								

In this experiment, the main aim are to investigate : 1. the main contributing factors among the eight possible variants; 2. if complete-random trees overfits. All variants will be compared with the benchmarking *Bagging* and *Random Forests*. The results are assessed by a sign test using 95% confidence level to determine whether the wins are statistically significant. Forty data sets are selected from the UCI repository [2]. Their data sizes range from one hundred to twenty thousand. This experiment uses ten thousand trees for each ensemble to see if any variants overfit. *Tenfold cross-validation* is conducted for each data set and the average error rate is reported.

5.1 Results

The average error is shown in table 1 and a pairwise comparison summary is presented in table 2. Comparing variants with the benchmark classifiers, we summarize the results as follows:

- PFO, PFB and PHO perform comparable to *Bagging*.
- PFO and PFB perform comparable to *Random Forests*.
- PFO has the most wins against the two benchmark classifiers having 23 and 21 out of 40.
- PFO has thirteen data sets with the best error rates as marked with asterisks in table 1 *Random Forests* has elevens and *Bagging* has eights.

For each of the three parameters, we summarize the results as follows:

- all probability averaging variants are significantly better than their voting counterparts according to the *sign test*.
- full height tree performs better than half height tree.
- bootstrap sampling impairs accuracy as suggested earlier on in section 3.

The results above suggest that the most accurate variant PFO is comparable to the benchmark classifiers and the probability averaging is the main contributing factor to complete-random decision trees. We call PFO "***Max-diverse Ensemble***". *Max-diverse Ensemble* performs better against all variants and has the lowest mean-error rate 17.3% as shown in table 1. Regarding overfitting, none of the data sets suffers from overfitting in general. It dispels the concern of using probability averaging with complete-random trees causes overfitting.

6 Conclusions

In this paper, we first discuss that maximizing tree diversity is a way to lower generalization error. Then, we provide a taxonomy on tree randomization as a systematic framework to characterize existing tree randomization methods. We find that complete-random test selection produces diverse trees. Finally, we thoroughly investigate the complete-random decision trees by exploring eight possible variants. The most accurate variant *Max-diverse Ensemble* has the maximum diversity according to our analysis and uses only simple probability averaging without any feature selection criterion or other random elements. For future work, it would be valuable to determine situations where *Max-diverse Ensemble* would perform better than other methods and vice versa.

References

1. Yali Amit and Donald Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 9(7):1545–1588, 1997.
2. C.L. Blake and C.J. Merz. Uci repository of machine learning databases, 1998.
3. Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
4. Leo Breiman. Randomizing outputs to increase prediction accuracy. *Machine Learning*, 40(3):229–242, 2000.
5. Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
6. Thomas G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, 2000.
7. Pedro Domingos. Bayesian averaging of classifiers and the overfitting problem. In *Proc. 17th International Conf. on Machine Learning*, pages 223–230. Morgan Kaufmann, San Francisco, CA, 2000.
8. Wei Fan, Haixun Wang, Philip S. Yu, and Sheng Ma. Is random model better? on its accuracy and efficiency. *Third IEEE International Conference on Data Mining*, pages 51–58, 2003.
9. Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.
10. J. R. Quinlan. *C4.5 : programs for machine learning.* Morgan Kaufmann Publishers, San Mateo, Calif., 1993.