

Comparing Pure Parallel Ensemble Creation Techniques Against Bagging

Lawrence O. Hall, Kevin W. Bowyer¹, Robert E. Banfield, Divya Bhadoria
W. Philip Kegelmeyer² and Steven Eschrich
Computer Science & Engineering, University of South Florida, Tampa, Florida 33620-5399
{hall, rbanfiel, dbhadori}@csee.usf.edu

¹ Computer Science & Engineering, 384 Fitzpatrick Hall, Notre Dame, IN 46556
kwb@cse.nd.edu

² Sandia National Labs, Biosystems Research Dept, POB 969, MS 9951, Livermore, CA 94551-0969
wpk@ca.sandia.gov

Abstract

We experimentally evaluate randomization-based approaches to creating an ensemble of decision-tree classifiers. Unlike methods related to boosting, all of the eight approaches considered here create each classifier in an ensemble independently of the other classifiers. Experiments were performed on 28 publicly available datasets, using C4.5 release 8 as the base classifier. While each of the other seven approaches has some strengths, we find that none of them is consistently more accurate than standard bagging when tested for statistical significance.

1 Introduction

This paper compares eight methods of creating an ensemble without incrementally focusing on misclassified examples as in boosting [8, 14]. Bagging [4], three variations of random forests [5], three variations of randomized C4.5 [9] (which we will call by the more general name “random trees”), and random subspaces [10] are compared. Their classification accuracy is evaluated through a series of 10-fold stratified cross validation experiments on 28 data sets. The base classifier is a modification of the C4.5 release 8 [13] that we call USFC4.5. USFC4.5 produces identical output to C4.5 release 8 with default settings, but has significant added functionality. Each ensemble creation approach compared here can be distributed in a simple way across a set of processors. This makes them suitable for learning from very large data sets [11, 2, 7].

The experimental results show that the random tree approaches and random forests methods gave a statistically significant, though small, increase in accuracy over building

a single decision tree. However, in head-to-head comparisons with bagging, none of the ensemble building methods was generally significantly more accurate than bagging.

2 Ensemble Creation Techniques Evaluated

Ho’s random subspace method of creating a decision forest utilizes the random selection of attributes or features in creating each decision tree. Ho used a randomly chosen 50% of the attributes to create each decision tree in an ensemble. The ensemble size was 100 trees. Ho found the random subspace approach was better than bagging and boosting for a single train/test data split for four data sets taken from the stat log project [3]. Fourteen other data sets were split into two halves randomly, for train and test. This was done 10 times for each of the data sets. The maximum and minimum accuracy results were deleted and the other eight runs were averaged. There was no evaluation of statistical significance. The conclusion was that random subspaces was better for data sets with a large number of attributes. This result, and some results from other papers listed below, conflict with our conclusions. We discuss those conflicts in Section 5.

Breiman’s random forest approach to creating an ensemble also utilizes a random choice of attributes in the construction of each CART decision tree [6, 5]. However, a random selection of attributes occurs at each node in the tree. Potential tests from these random attributes are evaluated and the best one is chosen. So, it is possible for each of the attributes to be utilized in the tree. The number of random attributes chosen for evaluation at each node is a variable in this approach. Additionally, bagging is used to create the training set for each of the trees that make up the random forests. In [5], random forest experiments were

conducted on 20 data sets and compared with Adaboost on the same data sets. Ensembles of 100 trees were built for the random forests and 50 for Adaboost. For the zip-code data set 200 trees were used. A random 10% of the data was left out of the training set to serve as test data. This was done 100 times and the results averaged. The random forest with a single attribute randomly chosen at each node was better than Adaboost on 11 of the 20 data sets. There was no evaluation of statistical significance. It was significantly faster to build the ensembles using random forests. In the experiments in this paper, we consider random subsets of size 1, 2 and $\lfloor \log_2(n) + 1 \rfloor$ where n , is the number of attributes.

Dietterich introduced an approach which he called randomized C4.5 [9], which comes under our more general description of random trees. In this approach, at each node in the decision tree the 20 best tests are determined and the actual test used is randomly chosen from among them. With continuous attributes, it is possible that multiple tests from the same attribute will be in the top 20. All tests (in C4.5) must be kept to determine the best 20, which can make this approach memory intensive. Dietterich experimented with 33 data sets from the UC Irvine repository. For all but three, a 10-fold cross validation was done. The best result from a pruned (certainty factor of 10) or unpruned ensemble was reported. Test results were evaluated for statistical significance at the 95% confidence level. It was found that randomized C4.5 was better than C4.5 14 times and equivalent 19 times. It was better than bagging with C4.5 6 times, worse 3 times and equivalent 24 times. From this, it was concluded that the approach tends to produce an equivalent or better ensemble than bagging. It has the advantage that you do not have to create multiple instances of a training set.

3 Algorithm Modifications

We describe our implementation of random forests and a modification to Dietterich's randomized C4.5 method. In the random forest implementation, in the event that the attribute set randomly chosen provides a negative information gain, our approach is to randomly re-choose attributes until a positive information gain is obtained. This enables each test to improve the purity of the resultant leaves to at least some degree. The same approach was also used in the WEKA system [15].

We have made a modification to the randomized C4.5¹ ensemble creation method in which only the best test from

¹On a code implementation note, we have added a `-pure` flag which allows trees to be grown to single example leaves, which we call pure trees. MINOBS is set to one (which means a test will be attempted any time there are two or more examples at a node), tree collapsing is not allowed and dynamic changes in the minimum number of examples in a branch for a test to be used are not allowed. All unpruned trees were built with the pure flag.

each attribute is allowed to be among the best set (of a given size) from which one is randomly chosen. We will call it the random tree B (RTB) approach. A slightly more memory efficient perturbation of this approach is to keep \sqrt{n} best attributes to randomly choose. In the rest of the paper, we will call our ensemble creation method RTB and Dietterich's original method random trees.

4 Experimental Results

Experiments were done on 28 data sets; 26 from the UC Irvine repository [12], credit-g from NIAAD (www.liacc.up.pt/ML) and phoneme from the ELENA project. The data sets have from 4 to 69 attributes and the attributes are a mixture of continuous and nominal² values. The ensemble size was 200 trees for the Dietterich and RTB approaches. There were 100 trees used in the random forest approach and in the ensemble for the random subspace approach. The size of the ensembles was chosen to allow for comparison with previous work (and corresponds with those authors' recommendations).

For the RTB approach, we used a random test from the 20 attributes with maximal information gain and a random test from the square root of the number of attributes, which of course will vary with the size of the attribute space. In the random subspace approach of Ho, exactly half ($\lceil n/2 \rceil$) of the attributes were chosen each time. For the random forest approach, we used a single attribute, 2 attributes and $\lfloor \log_2 n + 1 \rfloor$ attributes (which will be abbreviated as Random Forests-Ig in the following).

For each data set, a 10-fold cross validation was done. For each fold, an ensemble is built by each method and tested on the held out data. We also built a single C4.5 tree, with default pruning, on each of the folds. The accuracy of each ensemble method is compared against the single default pruned decision trees. The ensembles consist solely of unpruned trees.

For these experiments, with 8 classification methods and 28 data sets, there are 224 comparisons, and so about 11 errors in the comparisons at the 95% confidence level. Hence, we look at statistical significance at the 99% level, as shown in Table 1. Compared to C4.5 a random forest ensemble created using $\log_2 n + 1$ attributes is very good and RTB-20 is the best by a rather small increment. Random subspaces ties for the most times as statistically significantly more accurate than C4.5, but is also less accurate the most times. Several ensemble algorithms are very close and hard to pick among. We can create a summary score for each ensemble algorithm by providing 1 point for a win, and 1/2 point for a tie. Using this scoring approach, the random forest approaches have a score of 16.5 for 2 and Ig attributes

²As done by Dietterich, the attribute physician-fee-freeze has been left out of the voting data set to make it more difficult.

Table 1. Statistical significance at 99% level: + indicates more accurate, - indicates no difference, X means less accurate than C4.5.

Data Set	RTB sqrt	RTB 20	RT	RS	Bag ging	RF 1	RF 2	RF lg
anneal	-	-	-	-	-	-	-	-
audiology	-	-	-	-	-	-	-	-
autos	-	-	-	-	-	-	-	-
breast-y	-	-	-	-	-	-	-	-
breast-w	-	+	+	+	-	-	-	-
glass	-	-	-	+	-	-	+	-
heart-v	-	-	-	-	-	-	-	-
heart-s	-	-	-	-	-	-	-	-
heart-h	-	-	-	-	-	-	-	-
heart-c	-	-	-	-	-	-	-	-
iris	-	-	-	-	-	-	-	-
hepatitis	-	+	-	+	-	+	+	+
hypo	-	-	-	-	-	-	-	-
horse-colic	-	-	-	-	-	-	-	-
waveform	+	+	+	+	+	+	+	+
voting	-	-	-	-	-	-	-	-
vehicle	-	-	-	-	-	-	-	-
soybean	-	-	-	-	-	-	-	-
sonar	+	+	+	+	-	+	+	+
sick	-	-	-	X	-	X	-	-
primary	-	-	-	-	-	-	-	-
phoneme	-	+	-	X	+	+	+	+
lymph	-	+	-	-	-	-	-	-
labor	-	-	-	-	-	-	-	-
krkp	-	-	-	X	-	X	-	-
credit-g	+	-	+	+	+	+	-	+
credit-a	-	-	-	-	-	-	-	-
pima	-	-	-	-	-	-	-	-
Summary								
Better	3	6	4	6	3	5	5	5
Similar	25	22	24	19	25	21	23	23
Worse	0	0	0	3	0	2	0	0
Score	15.5	17	16	15.5	15.5	15.5	16.5	16.5

RT = Random Trees, RS = Random Subspaces, RF = Random Forests.

with RTB-20 at 17. On the other hand, most of the others amassed 15.5 points.

An interesting question is how would these approaches rank if the average accuracy, regardless of significance, was the criterion. In this case random forests-lg and bagging appear the best (22.5 and 21.5 points respectively). The other random forest approaches are at 21 points with random trees and random subspaces at 20. Now, RTB-20 is the weakest approach at 18.5 points. Clearly, utilizing statistical significance tests changes the conclusions that one would make given these experimental results. It is worth noting that all

Table 2. Pairwise win-lose-tie comparisons with significance at the 99% level.

wins-loses-ties row versus col	C4.5	RTB-20	Random Forests-lg
Bagging	3-0-25	1-1-26	0-1-27
Random Forests-lg	5-0-23	3-1-24	
RTB-20	6-0-22		

scores are well above 14 which means they are each better than growing a single pruned tree on average.

At the outset of the study, it was expected that one or more of these approaches would be an unambiguous winner over bagging in terms of accuracy. This was not the case, despite the earlier observation that, for instance, RTB-20 and random forests-lg seem to be better than a single C4.5 tree more often than bagging. When the two most competitive techniques are compared *directly* to bagging and each other (using the same methods for evaluating statistical significance at 99%), the results are as in Table 2. There we see bagging proves equivalent to RTB-20 and has one loss compared to random forests-lg. It was shown to be slightly worse than random trees (randomized C4.5) in previous work.

5 Discussion

Since the random forest approach utilizes bagging to create the training sets for the trees of its ensembles, one might expect that its accuracy was less than C4.5 on some of the same data sets for which bagging was less accurate. We found that random forests with one, two or $\log_2 n + 1$ random attributes to choose from was able to outperform C4.5 when bagging was worse two times for the first two approaches and three times with random forests-lg attributes. It was better than bagging was when compared with C4.5 twice when using two attributes. There were two cases in which random forests were worse than C4.5 when a bagged ensemble was better.

All the data sets used here, except Pima, were also used in the original randomized C4.5 paper [9], which found no losses to C4.5 at the 95% level. Our study finds only one loss. However, the previous study finds more wins in the (14 of 33 data sets) than we do. The difference could be due to our use of release 8 of C4.5, which is better at handling continuous valued attributes. Another big difference is that we utilized only unpruned trees. Dietterich chose the best of the pruned (certainty factor of 10) and unpruned trees.

In the random forests work, the ensembles obtained were compared with those obtained from Adaboost. On 19 data sets it was better 11 times and worse 8 times. There was no statistical test used to determine if the wins and losses were significant. Boosting is usually better than bagging unless there is noise in the data set [1]. We have nine data sets in common. It is difficult to draw direct conclusions, but this approach is one of the most competitive, which one would expect given the results in [5].

There are five data sets in common from the random subspaces paper [10]. In the experiments reported in the original paper random subspaces was better on all of these data sets. Here, at the 99% confidence level it is better once, worse once, and equivalent three times. We do not know

what release of C4.5 was used. However, a twofold cross validation was done 10 times and the outliers were removed (highest accuracy and lowest accuracy) with the remaining 8 averaged. Using a twofold cross validation the training set will be significantly smaller. The “data starvation” in the training set probably hurts the accuracy of the single tree more than it hurts the accuracy of the ensemble. Other work has shown that ensembles can recover accuracy with reduced training set sizes [7].

Random subspaces was not expected to do well when there are a small number of attributes. Its performance is less than a single classifier for Phoneme which has just five attributes and this was not unexpected. Also, it is no better than a single classifier on the Iris and Pima data sets which have only 4 and 8 attributes respectively. So, it was perhaps a lower performer partly due to the data sets chosen.

There are some computational advantages to random trees and random forests. Utilizing random trees it is not necessary to re-sample the training data in creating the individual trees. Random forests use a relatively small number of attributes in determining a test at a node which makes the tree faster to build.

It is possible to use the out of bag error to decide when to stop adding classifiers to a random forest ensemble or bagged ensemble. A stopping criterion of the error leveling off suffices. This, perhaps, would boost the performance of the random forests on the data sets utilized here.

Random trees and random forests can only be directly used to create ensembles of decision trees. The random subspace approach, which is less competitive than bagging, but faster because it uses less attributes, could be utilized with other learning algorithms such as neural networks.

Given the results presented here, it is perhaps worthwhile to explicitly consider the question — what would constitute a convincing experimental demonstration that a new technique achieves a general improvement in accuracy over simple bagging? Certainly the experiments should involve a “large” number of different datasets, say, in the range of 30 or more. Also, the comparison on each individual dataset should be in terms of whether or not the new technique achieves a statistically significant increase in accuracy. For this point, a paired t test on 10-fold or 20-fold cross-validation seems appropriate. The issue then becomes, on what fraction of the datasets should the new technique achieve a statistically significant increase in accuracy in order for us to accept that it offers a general improvement over bagging?

Acknowledgments: This research was partially supported by the Department of Energy through the ASCI Views Data Discovery Program, Contract number: DE-AC04-76DO00789 and the National Science Foundation under grant EIA-0130768.

References

- [1] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36(1,2):105–139, 1999.
- [2] K. Bowyer, N. Chawla, J. T.E. Moore, L. Hall, and W. Kegelmeyer. A parallel decision tree builder for mining very large visualization datasets. In *IEEE Systems, Man, and Cybernetics Conference*, pages 1888–1893, 2000.
- [3] P. Brazdil and J.Gama. The statlog project- evaluation / characterization of classification algorithms. Technical report, The STATLOG Project- Evaluation / Characterization of Classification Algorithms, <http://www.ncc.up.pt/liacc/ML/statlog/>, 1998.
- [4] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.
- [5] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [6] L. Breiman, J. Friedman, R. Olshen, and P. Stone. *Classification and Regression Trees*. Wadsworth International Group, Belmont, CA., 1984.
- [7] N. Chawla, T. Moore, L. Hall, K. Bowyer, W. Kegelmeyer, and C. Springer. Distributed learning with bagging-like performance. *Pattern Recognition Letters*, 24:455–471, 2003.
- [8] M. Collins, R. Schapire, and Y. Singer. Logistic regression, AdaBoost and Bregman distances. In *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, pages 158–169, 2000.
- [9] T. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, 2000.
- [10] T. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.
- [11] G. Hulten and P. Domingos. Learning from infinite data in finite time. In *Advances in Neural Information Processing Systems 14*, pages 673–680, Cambridge, MA, 2002. MIT Press.
- [12] C. Merz and P. Murphy. *UCI Repository of Machine Learning Databases*. Univ. of CA., Dept. of CIS, Irvine, CA. <http://www.ics.uci.edu/~mllearn/MLRepository.html>.
- [13] J. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1992. San Mateo, CA.
- [14] R. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.
- [15] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco, 1999.