

# Comparing Pure Parallel Ensemble Creation Techniques against Bagging

Lawrence O. Hall, Kevin W. Bowyer<sup>1</sup>, Robert E. Banfield, Divya Bhadoria, W. Philip Kegelmeyer<sup>2</sup> and Steven Eschrich

Department of Computer Science & Engineering  
University of South Florida  
Tampa, Florida 33620-5399

<sup>1</sup> Computer Science & Engineering  
384 Fitzpatrick Hall  
Notre Dame, IN 46556

<sup>2</sup> Sandia National Labs, Biosystems Research Department, PO Box 969, MS 9951  
Livermore, CA 94551-0969, USA

{hall, rbanfiel, dbhadori}@csee.usf.edu, kwb@cse.nd.edu, wpk@ca.sandia.gov

## Abstract

We experimentally evaluate bagging and seven other randomization-based approaches to creating an ensemble of decision-tree classifiers. Unlike methods related to boosting, all of the eight approaches create each classifier in an ensemble independently of the other classifiers in the ensemble. Bagging uses randomization to create multiple training sets. Other approaches, such as those of Dietterich, apply randomization in selecting a test at a given node of a tree. Then there are approaches, such as Breiman’s random forests and Ho’s random subspaces, which apply randomization in the selection of attributes to be used in building the tree. 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

There are now a number of ways to create an ensemble of decision tree classifiers [1, 2]. This paper compares four methods of creating an ensemble without incrementally focusing on misclassified examples as in boosting [3, 4]. Variations of two of the methods give us eight approaches to compare. 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 [5, 6, 7] because each classifier in an ensemble can be built at the same time if processors are available.

Bagging [8], three variations of random forests [9], three variations of randomized C4.5 [10] (which we will call by the more general name “random trees”), and random subspaces [11] are compared. Their classification accuracy is evaluated through a series of 10-fold stratified cross validation experiments on 28 data sets taken mostly from the UC Irvine repository [12]. The base classifier is a modification of the C4.5 release 8 [13] decision tree building software that we call USFC4.5. USFC4.5 produces identical output to C4.5 release 8 with default settings, but has significant added functionality.

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. Also, based on our experimental results, none of the other approaches can reliably be said to improve on the accuracy obtained with C4.5. 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 [14]. Fourteen other data sets were used by splitting them into two halves randomly. Each half was used as a training set with the other half used as a test set. 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. Ho's method tended to not be as good with a smaller number of attributes and a small number of examples or a small number of attributes and a large number of classes. This approach is interesting for large data sets with a significant number of attributes because it requires less time and memory to build each of the classifiers.

Breiman's random forest approach to creating an ensemble also utilizes a random choice of attributes in the construction of each CART decision tree [15, 9]. However, a random selection of attributes occurs at each node in the decision 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. We utilized random subsets of size 1, 2 and  $\lceil \log_2(n) + 1 \rceil$  where  $n$ , is the number of attributes.

Random forest experiments were conducted on 20 data sets and compared with Adaboost on the same data sets in [9]. Ensembles of 100 decision trees were built for the random forests and 50 decision trees 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.

Dietterich introduced an approach which he called randomized C4.5 [10], 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 when there are many continuous attributes which have many values that can be used in a binary test.

Dietterich experimented with 33 data sets from the UC Irvine repository. For all but three of them a 10-fold cross validation approach was followed. The best result from a pruned or unpruned ensemble was reported. Pruning was done with a certainty factor of 10. The 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 [16].

We have made a modification to the randomized C4.5<sup>1</sup> ensemble creation method in which only the best test from each attribute is allowed to be among the best set (of a given size) from which one is randomly chosen. This allows the algorithm to be more memory

---

<sup>1</sup>On 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.

Table 1: Description of data sets attributes and size.

Data Set	# attributes	# continuous attributes	# examples	# classes
anneal	38	6	898	6
audiology	69	0	226	24
autos	25	15	205	7
breast-y	9	0	286	2
breast-w	9	9	699	2
glass	9	9	214	7
heart-v	13	5	200	2
heart-s	13	5	123	2
heart-h	13	5	294	2
heart-c	13	5	303	2
iris	4	4	150	3
hepatitis	19	6	155	2
hypo	25	7	3163	2
horse-colic	22	8	368	2
waveform	21	21	5000	3
voting	15	0	435	2
vehicle	18	18	846	4
soybean	35	0	683	19
sonar	60	60	208	2
sick	29	7	3772	2
primary	17	0	339	22
phoneme	5	5	5404	2
lymph	18	3	148	4
labor	16	8	57	2
krkp	36	0	3196	2
credit-g	20	7	1000	2
credit-a	15	6	690	2
pima	8	8	768	2

efficient when there are a large number of continuous valued attributes. 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 NIADD ([www.liacc.up.pt/ML](http://www.liacc.up.pt/ML)) and phoneme from the ELENA project. The data sets, described in Table 1, have from 4 to 69 attributes and the attributes are a mixture of continuous and nominal<sup>2</sup> 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  $\lceil \log_2 n + 1 \rceil$  attributes (which will be abbreviated as Random Forests-lg 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. This allows for statistical comparisons between approaches to be made. We also built a single C4.5 decision 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. A paired t-test is used to determine whether a particular ensemble approach is better than or worse than building a single decision tree at a particular confidence level.

Table 2 shows the comparative results at the 95% confidence interval. All ensemble creation techniques from bagging to the right of the table utilized bagging in creating

---

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

Table 2: Statistically significantly better than C4.5 at the 95% confidence interval: A + indicates more accurate, a - indicates no difference detected, and an X means the ensemble approach is less accurate than C4.5.

Data Sets	RTB-sqrt	RTB-20	Random Trees	Random Subspaces	Bagging	Random Forests-1	Random Forests-2	Random Forests-lg
anneal	+	+	+	-	-	+	-	+
audiology	-	-	-	+	+	-	-	-
autos	-	-	-	-	-	-	-	-
breast-y	-	-	-	-	-	-	-	-
breast-w	+	+	+	+	+	+	+	+
glass	+	-	+	+	-	+	+	+
heart-v	-	-	-	-	-	-	-	-
heart-s	-	-	-	-	-	-	-	-
heart-h	-	-	-	-	-	+	-	-
heart-c	-	-	-	+	-	+	+	-
iris	-	-	-	-	-	-	-	-
hepatitis	-	+	+	+	+	+	+	+
hypo	-	-	-	X	-	X	X	-
horse-colic	-	-	-	-	-	-	-	-
waveform	+	+	+	+	+	+	+	+
voting	-	-	-	-	-	-	+	-
vehicle	-	-	-	-	-	-	-	-
soybean	-	-	-	-	-	-	-	-
sonar	+	+	+	+	+	+	+	+
sick	+	+	-	X	-	X	-	-
primary	-	-	-	-	-	-	-	-
phoneme	+	+	-	X	+	+	+	+
lymph	-	+	+	-	-	-	-	-
labor	-	-	-	-	-	-	-	-
krkp	-	-	-	X	-	X	X	-
credit-g	+	+	+	+	+	+	+	+
credit-a	X	-	X	-	-	-	-	-
pima	-	-	-	-	-	-	-	-
Summary								
Better	8	9	8	8	7	10	9	8
Similar	19	19	19	16	21	15	17	20
Worse	1	0	1	4	0	3	2	0
Score	17.5	18.5	17.5	16	17.5	17.5	17.5	18

training sets. There's a double line in each table separating the approaches that use bagging from those that do not. For 13 of the data sets none of the randomizing ensemble approaches could produce an improvement over C4.5. On four data sets, all techniques showed accuracy improvement. The (slightly) best ensemble building approaches appear to be our modification to the random tree approach of Dietterich (RTB-20), which is better 9 times and, random forests-lg which is better 8 times. Both are **never** worse. The others are about the same. Random forests with one attribute had the most wins at 10, but also the most losses at 3. Random trees had 8 wins and 1 loss. The only approach that is somewhat separated from the rest is random subspaces with 8 wins and 4 losses.

We can create a summary score for each ensemble algorithm by providing 1 point for a win, and 1/2 point for a tie. At the 95% confidence level the top performing ensemble methods are RTB-20 (18.5 points) and random forests-lg (18 points). All other approaches score 17.5 points except for Random subspaces, the lowest performing method, at 16 points.

For these experiments, there will be about 11 errors in the comparisons in the table at the 95% confidence level. Hence, in order to be more certain about our conclusions, we also look at statistical significance at the 99% level, as shown in Table 3. The same ensemble building algorithms appear better than C4.5 and perform similarly. In particular, 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. Several ensemble algorithms are very close and hard to pick among. In this case using the scoring approach, the random forest approaches have a score of 16.5 for 2 and lg attributes with RTB-20 at 17. On the other hand, most of the others amassed 15.5 points which is slightly worse.

An interesting question is how would these approaches rank if the average accuracy, regardless of significance, was the only criterion. Table 4 shows that 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

Table 3: Statistically significantly better than C4.5 at the 99% confidence interval: A + indicates more accurate, a - indicates no difference detected, and an X means the ensemble approach is less accurate than C4.5.

Data Set	RTB-sqrt	RTB-20	Random Trees	Random Subspaces	Bagging	Random Forests-1	Random Forests-2	Random Forests -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

Table 4: Better than C4.5 on average. A + indicates more accurate, a - indicates no difference detected, and an X means the ensemble approach is less accurate than C4.5.

Data Sets	RTB-sqrt	RTB-20	Random Trees	Random Subspaces	Bagging	Random Forests-1	Random Forests-2	Random Forests-lg
anneal	+	+	+	+	+	+	+	+
audiology	+	+	+	+	+	+	+	+
autos	+	X	+	+	X	X	X	+
breast-y	X	X	X	X	X	X	X	X
breast-w	+	+	+	+	+	+	+	+
glass	+	+	+	+	+	+	+	+
heart-v	X	X	X	-	X	+	+	+
heart-s	X	X	X	-	X	X	X	X
heart-h	X	X	X	+	X	+	+	+
heart-c	+	+	+	+	+	+	+	+
iris	-	-	+	X	-	+	X	-
hepatitis	+	+	+	+	+	+	+	+
hypo	X	X	X	X	X	X	X	X
horse-colic	X	+	X	X	+	X	+	+
waveform	+	+	+	+	+	+	+	+
voting	+	+	+	+	+	+	+	+
vehicle	+	+	+	+	+	+	+	+
soybean	+	+	X	+	+	+	+	+
sonar	+	+	+	+	+	+	+	+
sick	+	+	+	X	+	X	X	X
primary	+	+	+	+	+	+	+	+
phoneme	+	+	+	X	+	+	+	+
lymph	+	+	+	+	+	+	+	+
labor	+	+	+	+	+	+	+	+
krkp	X	X	+	X	+	X	X	X
credit-g	+	+	+	+	+	+	+	+
credit-a	X	X	X	+	+	+	+	+
pima	X	X	+	+	+	+	+	+
Summary								
Better	18	18	20	19	21	21	21	22
Same	1	1	0	2	1	0	0	1
Worse	9	9	8	7	6	7	7	5
Score	18.5	18.5	20	20	21.5	21	21	22.5

Table 5: Pairwise comparisons between the three most competitive ensemble creation algorithms. Each cell contains the number of wins, losses and ties between the algorithm in that row and the algorithms in that column with significance at the 99% level.

	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		

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

## 5 Discussion

### 5.1 Random Forests and Bagging

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. In Table 4, we can see 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.

### 5.2 Comparison Against Prior Results in the Literature

All the data sets used here, except Pima, were also used in the original randomized C4.5 paper [10]. Comparing with C4.5 there were never any losses at the 95% confidence level. This is almost true in this study (1 loss). What is different is that there were more wins

in the previous study (14 of 33 data sets). The difference could be due to the release of C4.5 utilized, we use release 8 and release 1 was used in the previous study. Release 8 is better at handling continuous valued attributes. However, another big difference is that we utilized only unpruned ensembles. Dietterich chose the best of the pruned (certainty factor of 10) and unpruned ensembles arguing that the choice to prune might always be correctly determined by doing cross validation on the training set (though this was apparently not evaluated on these data sets).

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 [9].

There are five data sets in common from the random subspaces paper [11]. 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. It’s 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.

### 5.3 Other Ensemble Methods vs. Bagging

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 observation earlier in this paper 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 5, evaluated here at the 99% significance level. 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. The release of C4.5 compared here is a later one that handles continuous attributes better [17] and perhaps that accounts for the difference. More likely, it is the fact that we used only unpruned ensembles rather than choosing the best of the pruned or unpruned ensemble. One would expect random forests to be clearly better than bagging given previous results. Perhaps our implementation choice of re-choosing a random attribute in the case of negative information gain is causing lower performance, though this is not intuitive. Perhaps CART trees benefit more from a random forests approach.

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 of the ensemble. This can represent a small to large time savings depending on implementation. 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.

## **5.4 Determining an Ensemble Building Method is Better than Bagging**

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? One possible criteria is to use a McNemar one-tailed test for statistical significance of the frequency of statistically significant differences across datasets. This would essentially be a sign test on the number of datasets for which the new technique achieves a statistically significant increase in accuracy, versus the number of datasets for which bagging achieves a statistically significant increase in accuracy. For example, if a new technique is statistically significantly better than bagging on nine datasets, and bagging is not statistically significantly better than the new technique on any dataset,

then we would conclude that the new technique achieved a general improvement over bagging; that is, demonstrated a statistically significant improvement a significant fraction of the time.

## 6 Summary

This paper compares several methods of building ensembles of decision trees. In particular two versions of the randomized C4.5 method introduced by Dietterich [10] (which we call by the more general term random trees), random subspaces [11], random forests [9] and bagging are compared. All experiments used a 10-fold cross validation approach to compare average accuracy. The accuracy of the various ensemble building approaches was compared with building a single C4.5 release 8 pruned decision tree utilizing the default parameters. All trees in the ensemble were unpruned. The comparison was done on 28 data sets with 26 taken from the UC Irvine repository [12] and the others publicly available.

The ensemble size was 200 trees for the random tree approaches. It was 100 trees for the random forests, random subspace and bagging approaches. The ensemble size was chosen to match what had been utilized in previous work [9, 10]. Statistical significance tests were done to determine whether each of the ensemble methods was statistically significantly more accurate than or less accurate than building a single tree on each of the data sets.

The ensemble approaches were generally as accurate or more accurate than building a single tree. None of the approaches was unambiguously more accurate than bagging. Surprisingly, bagging was found to be the most accurate approach in head-to-head comparisons except against random forests-lg where it was less accurate with significance one time out of twenty eight. Random trees and random forests are very competitive with bagging and are clearly better than building a single tree. The accuracy of the random subspace approach fluctuated and was typically less than the other approaches. It is notable that a random forest built utilizing a single randomly chosen attribute to create each

test in the decision tree was among the most accurate classification methods. It is also fast to compute a tree in this way.

**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] Robert Brylla, Ricardo Gutierrez-Osuna, and Francis Quek. Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets. *Pattern Recognition*, 36:1291–1302, 2003.
- [3] M. Collins, R.E. 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.
- [4] R.E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.
- [5] 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.
- [6] K.W. Bowyer, N.V. Chawla, Jr. T.E. Moore, L.O. Hall, and W.P. Kegelmeyer. A parallel decision tree builder for mining very large visualization datasets. In *IEEE Systems, Man, and Cybernetics Conference*, pages 1888–1893, 2000.

- [7] N.V. Chawla, T.E. Moore, L.O. Hall, K.W. Bowyer, W.P. Kegelmeyer, and C. Springer. Distributed learning with bagging-like performance. *Pattern Recognition Letters*, 24:455–471, 2003.
- [8] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.
- [9] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [10] T.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.
- [11] T.K. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.
- [12] C.J. Merz and P.M. Murphy. *UCI Repository of Machine Learning Databases*. Univ. of CA., Dept. of CIS, Irvine, CA. <http://www.ics.uci.edu/~mlearn/MLRepository.html>.
- [13] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1992. San Mateo, CA.
- [14] 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.
- [15] L. Breiman, J.H. Friedman, R.A. Olshen, and P.J. Stone. *Classification and Regression Trees*. Wadsworth International Group, Belmont, CA., 1984.
- [16] Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco, 1999.

- [17] J.R. Quinlan. Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Research*, 4:77–90, 1996.