Factors Affecting Boosting Ensemble Performance on DNA Microarray data.

Geoffrey R. Guile and Wenjia Wang

Abstract— **Boosting techniques have been applied to DNA microarray data because their high dimensionality has made them difficult to analyze. However, classification performance varies between boosting algorithms. We have investigated factors affecting error in boosting ensemble classifiers on DNA Microarray data: number of training samples, number of boosting iterations, complexity of base learners and diversity of models. Specifically we have applied diversity measures to investigate the relationships between model type, model accuracy, diversity and ensemble accuracy.**

I. Introduction

Because of their high dimensionality and complexity, DNA microarray and other high dimensional biological data datasets have proven difficult to analyze, and machine learning techniques have to be applied. Ensemble methods such as bagging and boosting which produce a committee of classification models can be more accurate than those that produce a single model and have been shown to be effective for analyzing high dimensional data.

The reason for employing the ensemble learning approach is that while a classification algorithm may only be able to produce a model with slightly better accuracy than random guessing, the combined accuracy of several models will be greater than any single classifier, providing they are sufficiently diverse from each other, so that they do not make similar errors, even though the individual models may not have high accuracy themselves.

In order to produce an ensemble, boosting algorithms iteratively employ another algorithm known as the base learner to generate a series of models, which are combined into an ensemble. The base learner can be any algorithm normally used for classification or prediction, such as a neural network or a decision tree. At the start of the boosting process all samples have equal weights, after each successive iteration the accuracy of the model produced is measured and the samples weights are adjusted so that the weights of misclassified samples are increased while those of correctly classified samples are reduced. With successive iterations the base learner increasingly concentrates on the misclassified samples. The models produced are then combined into an ensemble voting committee. It is obvious that if all the models are identical, or nearly identical, there will be no advantage in having more than one, and so boosting algorithms are designed with in order to produce a high level of diversity.

While it is generally recognized that for a boosting ensemble to be an effective classifier there has to be a high level of diversity, to date there has been no published research where measures of diversity have been applied to boosting ensembles in order to investigate the relationships between model type, model accuracy, diversity and ensemble accuracy. Therefore, in order to understand why some boosting-based ensembles work better than others, we applied some standard diversity measures to investigate how ensemble diversity varied for four different boosting algorithms and when different depths of decision trees were used as base learners.

II. Related work

The accuracy of an ensemble is affected by a number of factors, including the accuracy of the individual models, the decision fusion strategy and diversity between the models within the ensemble. These factors were investigated and discussed by Wang [1], who found that ensembles built with the most accurate models were not necessarily the most accurate, and that adding some less accurate but more diverse models could improve the overall accuracy of an ensemble.

While the concept of diversity is easy to grasp intuitively, precisely defining and measuring it is not straightforward and there have been a number of measures of diversity proposed. However, there is no generally agreed standard diversity measure and there has been considerable interest in comparing the performance of different diversity measures. For example, Kuncheva and Whittaker [2] investigated the relationship between ensemble accuracy and the values obtained with ten different diversity measures. However, previous work has concentrated on comparing the performance of the diversity measures themselves, rather than applying them to investigate the performance of different boosting mechanisms.

Because microarray data are of high dimensionality and have proven difficult to analyze, a number of investigators have proposed ways of improving the performance of boosting algorithms with microarray data.

Dettling proposed the BagBoosting algorithm [3] which incorporates a bagging step into LogitBoost [4]. At each boosting iteration several sets of bagged samples are produced by random sampling with replacement. The base learner produces a separate decision stump model for each set of bagged samples and all the models from that iteration are then added to the ensemble. By using the bagging step to generate a set of models at each boosting round, the overall diversity of the ensemble should be increased.

G. R. Guile is a with the School of Computing Sciences, University of East Anglia, Norwich, Norfolk, NR4 7TJ, UK. (e-mail: g.guile@uea. ac.uk).

W. Wang is with the School of Computing Sciences, University of East Anglia, Norwich, Norfolk, NR4 7TJ, UK. (e-mail: wenjia.wang@uea.ac.uk).

978-1-4244-8126-2/10/$26.00 ©2010 IEEE

Long and Vega [5] investigated the effect of not replacing features once they had been used. They incorporated a non-replacement mechanism into AdaBoost and found that the non-replacement version gave lower error rates than standard AdaBoost. The feature non-replacement mechanism forces the base learner to use unused features to potentially increase the amount of information taken into consideration when constructing the ensemble and increase the diversity of the models.

We proposed LogitBoost-NR and BagBoosting-NR which incorporated feature non-replacement into LogitBoost and BagBoosting respectively and found that the non-replacement algorithms gave lower error rates than the unmodified versions, overall BagBosting-NR gave the lowest error rates for the classification of microarray data [6].

Most applications of boosting to microarray data have used decision stumps as the base learner. Decision trees can take feature interaction into account and can therefore be more accurrate than decision stumps, but they are more likely to suffer from overfitting. We investigated the use of varying depths of decision trees with LogitBoost and LogitBoost-NR and found that ensembles of decision trees generally have higher error rates than decision stumps with LogitBoost, but that with LogitBoost-NR the lowest error rates were achieved using decision trees deeper than stumps, although the error rates of the individual models in the ensembles were higher. We concluded that this was because the non-replacement mechanism was generating increased diversity, resulting in higher overall accuracy, despite the lower accuracy of the individual models [7].

III. Methods

Factors that can affect the accuracy of a boosting classification ensemble include:

1) The number of training samples.
2) The number of boosting iterations.
3) The complexity of the individual models.
4) The diversity between the models.
5) The sample weight adjustment mechanism.
6) The decision fusion strategy.

The number of training samples will affect ensemble accuracy since small numbers of samples are less likely to be representative of the entire dataset. However, different algorithms are still likely to give different performance, because some will be able to utilize the available information better than others.

The number of boosting iterations will have an effect on ensemble accuracy, and since the ensemble approach is designed to utilize increasing amounts of useful information with successive iterations, the accuracy should improve. However, there is likely to be a point reached where little improvement results because the boosting algorithm has utilized as much information as it is able to.

The complexity of individual models in the ensemble will affect the overall accuracy. Generally, a more complex base learner, such as a decison tree, should be more accurate than a simple base learner, such as a decison stump; and a more accurate base learner should result in a more accurate ensemble. However, this may not be the case if other factors do not remain constant. In our previous study [7] we found that when LogitBoost was applied to microarray data, using decision tree base learners rather than decision stumps always resulted in higher ensemble error, even though the models themselves were more accurate. We concluded that this was because the diversity of the tree models was lower than that of the stumps. Thus more complex models may be more accurate, but this will not necessarily result in more accurate ensembles.

The diversity between the models in the ensemble is of major importance. The entire rationale behind the ensemble approach is to have a diversity of models, so that the ensemble accuracy is greater than that of any individual model. Because there is no generally agreed standard diversity measure, for this study we decided to use three diversity measures, as described below in Section III-A.

Different mechanisms are employed for the sample weight adjustment at successive boosting rounds by different algorithms, for example LogitBoost uses binary log-likelihood and AdaBoost uses an exponential function. Dettling and Bühlmann [10] argued that this makes LogitBoost more robust than AdaBoost as it should be less sensitive to outliers, and showed that it does have slightly lower error rates. In a previous paper [6] we found that LogitBoost-NR gave slightly lower error rates than AdaBoost-NR, but the differences were much less than those between LogitBoost-NR and standard LogitBoost. In view of these previous results we conclude that the sample weight adjustment mechanism is not likely to have a major effect on ensemble performance and have not investigated its effects in this study.

The decision fusion strategy of most boosting algorithms is usually either weighted majority voting or simple majority voting. We have not investigated the effect of the decision fusion strategy for this paper, all the algorithms we have used employ weighted majority voting. We also note that Long and Vega [5] reported similar error rates for Arc-4-RW (which uses simple majority voting) and AdaBoost (which uses weighted majority voting), so the decision fusion strategy may not be as important for ensemble error rates as other factors.

A. Diversity measures

There are two main types of diversity measure: *pairwise* and *non-pairwise*. In a theoretical analysis Bian [9] found that pairwise diversity measures are affected by the number of base models in the ensemble, while non-pairwise measures are not. We therefore decided to use some standard non-pairwise measures of diversity to investigate diversity between the models in the ensembles we generated when applying the different boosting methods to microarray data. We applied three diversity measures: the coincident failure diversity *CFD* and generalized diversity *GD* measures of Partridge and Krzanowski [8], and entropy *E* as implemented by Kuncheva and Whittaker [2].

1) Entropy: We have followed Kuncheva and Whittaker's entropy-based measure of diversity [2], as follows: Let $\mathcal{D} = \{\mathbf{x}_1 \ldots \mathbf{x}_S\}$ be a set of gene expression data for S samples. The Entropy measure of diversity E can be defined as follows [2]. Denote by $m(\mathbf{x}_s)$ the number of classifiers in an ensemble that correctly recognize $\mathbf{x}_s$, then:

$$E = \frac{1}{S}\sum_{s=1}^{S} \frac{1}{(M - \lceil M/2 \rceil)} \times \min\{m(\mathbf{x}_s), M - m(\mathbf{x}_s)\}$$

E will vary between 0 (no diversity) and 1 (maximum diversity).

2) Generalized diversity: The generalized diversity measure, *CFD*, of Partridge and Krzanowski [8], is defined as follows, where p_m denotes the probability that m randomly chosen classifiers fail on a randomly chosen sample:

$$\begin{aligned} p(1) &= \sum_{m=1}^{M} \frac{m}{M} \times p_m \\ p(2) &= \sum_{m=1}^{M} \frac{1}{M} \times \frac{(m-1)}{(M-1)} \times p_m \\ GD &= 1 - \frac{p(2)}{p(1)} \end{aligned}$$

3) Coincident failure diversity: The coincident failure diversity measure, *CFD*, of Partridge and Krzanowski [8], is defined as follows:

$$CFD = \sum_{m=1}^{M} \frac{(M-m)}{(M-1)} \times f_m$$

Where

$$f_m = \frac{\text{num. samples misclassified by } m \text{ models}}{\text{num. samples misclassified by at least one model}}$$

The maximum possible value of *CFD* is 1, the minimum value of 0 is obtained when all models are identical regardless of their accuracy.

B. Boosting algorithms

The boosting algorithms we investigated were the LogitBoost algorithm of Friedman *et.al.* [4], [10], the BagBoosting algorithm of Dettling [3], and our own LogitBoost-NR and BagBoosting-NR, which were developed from LogitBoost and BagBoosting respectively [6]. These algorithms have all been previously applied to the analysis of microarray data and achieved good classification performance. Furthermore, BagBoosting was developed from LogitBoost and thus a comparison of the diversity achieved by the different algorithms should give an insight into the effectiveness of the different mechanisms incorporated into them. We have previously investigated the effect of tree depth on the performance of LogitBoost [7] and found that deeper decision tree base learners generally gave lower error rates than decision stumps, we therefore investigated the effect of decision tree depth on error rates and diversity with all four algorithms. We did not include AdaBoost in our comparison because we have previously found [6] that the classification error rates of LogitBoost and LogitBoost-NR were similar to those of AdaBoost and AdaBoost-NR respectively under comparable conditions, thus we would expect that the diversity levels would also be similar under comparable conditions.

C. Datasets used

For testing we used two benchmark DNA microarray datasets: the Colon dataset [11] and the Leukemia dataset [13]. The Leukemia dataset was preprocessed as in Dudoit *et.al.* [12]. The demography of the datasets is given in Table I.

TABLE I

DEMOGRAPHY OF DATASETS USED.

Dataset	Number of		
	Features (genes)	Samples	Classes
Colon	2000	62	2
Leukemia	3571	72	2

D. Code

The code for randomizing the data was written in Java version 1.5.0. Experiments were performed using the R Statistical Package version 2.8.0. The code for the boosting algorithms was modified from that of the R package "boost" of [10].

IV. EXPERIMENTS AND RESULTS

A. Standard testing proceedure

A dataset was randomly partitioned into a training set and a test. Except where the effect of the number of training samples was being investigated we used two thirds of the samples for training and one third for testing. Each experiment was repeated 50 times with different random partitions to test the consistency of the models. The partitioning used a random seed to ensure that the same set of partitions could be generated repeatedly, in order that experiments could be reproduced. No feature preselection was performed. A boosting algorithm was applied to each dataset using either decision stumps or decison trees for the base learners. We set the number of iterations of boosting, M, at 25, except where the effect of the number of iterations was being investigated. With BagBoosting and BagBoosting-NR 25 rounds of bagging were applied at each boosting step. We recorded the final ensemble error on the test data ($E_{\mathcal{E}}$), the mean model error on the test data (E_m) and the diversity of the ensemble using the three diversity measures for each of the 50 partitions of the data. The values reported are the means across the 50 partitions.

B. Number of samples

We investigated the effect of the number of training samples, N_{train}, using LogitBoost and LogitBoost-NR, varying N_{train} from 20 to 80% of samples, and with the maximum tree depth, d_{max}, set at 1, 2 and 3. The results from 50 random partitionings of the datasets are given in Tables II and III.

TABLE II

Error rates with different training set sizes, Colon dataset, LogitBoost(-NR).

N_{train} (%)	LogitBoost, d_{max} =			LogitBoost-NR, d_{max} =		
	1	2	3	1	2	3
12 (20%)	31.68	33.16	33.16	27.72	27.92	28.04
19 (30%)	26.98	31.02	32.37	21.49	21.53	20.60
25 (40%)	23.73	30.70	29.89	20.65	20.00	19.62
31 (50%)	20.64	28.64	29.35	19.81	18.51	18.19
37 (60%)	21.44	25.28	29.04	19.04	18.16	17.60
43 (70%)	19.79	23.05	26.21	19.26	17.26	16.42
50 (80%)	19.83	19.33	29.17	17.33	17.00	15.83

TABLE III

Error rates with different training set sizes, Leukemia dataset, LogitBoost(-NR).

N_{train} (%)	LogitBoost, d_{max} =			LogitBoost-NR, d_{max} =		
	1	2	3	1	2	3
14 (20%)	33.45	34.86	34.86	14.79	14.59	14.62
22 (30%)	18.60	20.92	20.92	8.04	8.12	8.24
29 (40%)	10.33	13.72	14.42	5.02	5.49	6.23
36 (50%)	9.17	12.50	13.89	4.83	4.45	4.78
43 (60%)	6.97	11.10	14.89	3.66	4.00	4.14
50 (70%)	6.09	8.64	13.00	3.73	3.73	3.91
58 (80%)	5.71	8.29	13.00	4.29	3.57	3.14

C. Number of boosting iterations

We investigated the effect of the number of boosting iterations, M, using LogitBoost and LogitBoost-NR. M was varied from 1 to 50, decision stumps were used as base learners. The results are summarized in Table IV.

TABLE IV

Error rates with different numbers of boosting iterations, LogitBoost(-NR).

M	Colon		Leukemia	
	LogitBoost	LogitBoost -NR	LogitBoost	LogitBoost -NR
1	26.10	26.10	13.50	13.50
5	24.48	24.48	7.17	5.67
10	22.10	21.81	6.08	4.00
15	20.57	19.91	6.25	3.67
20	20.57	19.62	6.00	3.50
25	20.00	19.05	6.25	3.17
30	19.52	18.10	6.17	2.84
35	19.14	18.00	6.08	3.25
40	19.33	18.67	6.08	3.25
45	19.43	18.57	6.08	3.25
50	18.95	18.67	6.25	3.42

D. Complexity, diversity and accuracy

We investigated the effects of model error and complexity on ensemble error and diversity using using LogitBoost, LogitBoost-NR, BagBoosting and BagBoosting-NR. The effect of the complexity of models was investigated by using d_{max} from 1 to 5. In each case $E_{\mathcal{E}}$, E_m, *CFD*, *GD*, and *E* across the 50 partitions were recorded. We found that overall, *CFD* was the most useful diversity measure, and Tables VII and VIII give the values of ensemble error, mean model error and *CFD* for the Colon and Leukemia data respectively, using BagBoosting and BagBoosting-NR. Tables V and VI give the values of ensemble error, mean model error and *CFD* using LogitBoost and LogitBoost-NR, for the Colon and Leukemia data respectively. Figures 1 and 2 show plots of ensemble error, mean model error and *CFD* for the Colon dataset, with LogitBoost and LogitBoost-NR, respectively. Tables VII and VIII give the values of ensemble error, mean model error and *CFD* using BagBoosting and BagBoosting-NR, for the Colon and Leukemia data respectively. Tables IX–XVI in the Appendix give the complete results including *GD* and *E*.

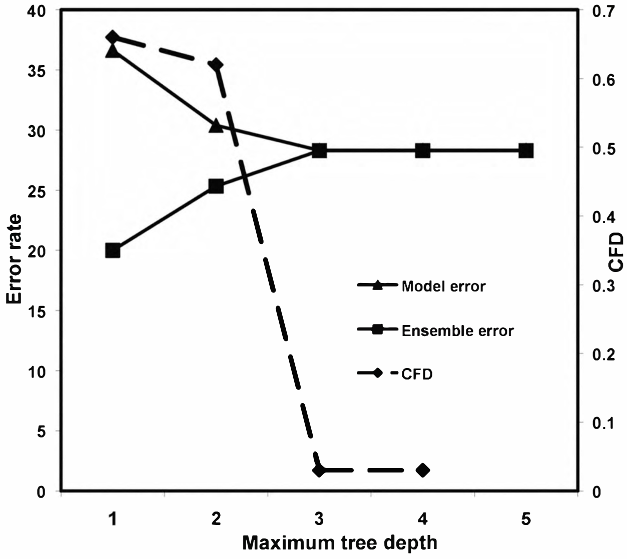

Fig. 1. Ensemble error rates, model error rates and *CFD* values at different tree depths, with Colon dataset, LogitBoost.

TABLE V

Error rates and diversity with Colon dataset, LogitBoost(-NR).

d_{max}	LogitBoost			LogitBoost-NR		
	$E_{\mathcal{E}}$	E_m	*CFD*	$E_{\mathcal{E}}$	E_m	*CFD*
1	20.00	36.62	0.66	19.05	37.12	0.65
2	25.33	30.39	0.62	18.10	31.50	0.71
3	28.29	28.31	0.03	17.91	28.56	0.73
4	28.29	28.31	0.03	17.91	28.54	0.73
5	28.29	28.31	0.03	17.91	28.53	0.73

V. Discussion and evaluation

With microarray data, only small numbers of samples are available, and so the performance with small numbers of training samples is important. Certain patterns can be seen in our results in Tables II and III. With the lowest number of samples the error rates were similar at all tree depths for the same algorithm, but those with LogitBoost-NR were much lower than those with standard LogitBoost.

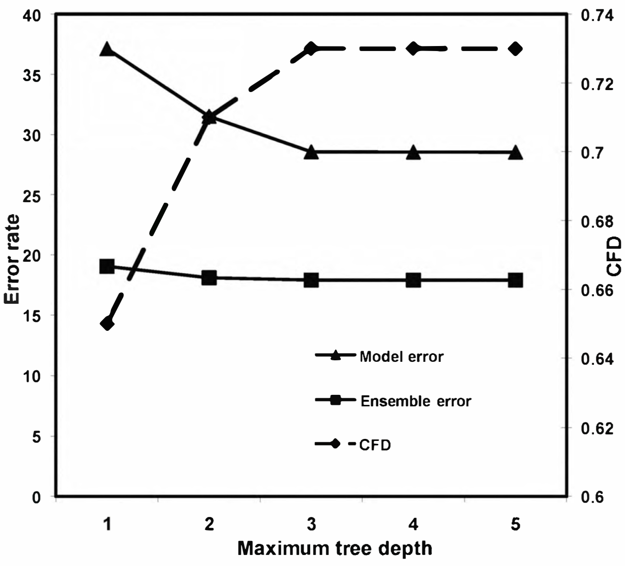

Fig. 2. Ensemble error rates, model error rates and *CFD* values at different tree depths, with Colon dataset, LogitBoost-NR.

TABLE VI

ERROR RATES AND DIVERSITY WITH LEUKEMIA DATASET, LOGITBOOST(-NR).

d_{max}	LogitBoost			LogitBoost-NR		
	$E_{\mathcal{E}}$	E_m	*CFD*	$E_{\mathcal{E}}$	E_m	*CFD*
1	6.42	17.66	0.67	3.25	21.85	0.81
2	10.33	15.69	0.57	3.83	17.68	0.84
3	15.42	15.42	0.00	2.58	16.54	0.85
4	15.42	15.42	0.00	2.58	16.54	0.85
5	15.42	15.42	0.00	2.58	16.54	0.85

For example, with the Colon data at 12 samples the error rate with LogitBoost was in the range 31.68–33.16% and with LogitBoost-NR in the range 27.72–28.04%. For both datasets, the error rates with the largest number of samples were considerably lower than with the smallest, however with standard LogitBoost the error rates with decision stumps were much lower than with trees of depth 3: in the case of the Colon data the error rate with 50 training samples was 19.83% using stumps, but 29.17% using trees of depth 3. Thus, with increasing numbers of training samples the performance of decision stumps as base learners improved compared to that with the deepest trees. In contrast, with LogitBoost-NR, the performance with all 3 depths of decision tree improved much more uniformly with increasing numbers of training samples. With the Colon data and 50 training samples, an error rate of 15.83% was achieved with trees of depth 3, which was lower than the 17.33% achieved with stumps. With the Leukemia data the overall pattern was comparable to that with the Colon data, except that the differences between the error rates with LogitBoost and LogitBoost-NR were greater. Overall, the lowest error rates were achieved using LogitBoost-NR, particularly with small numbers of training samples.

TABLE VII

RESULTS WITH COLON DATASET, BAGBOOSTING(-NR).

d_{max}	BagBoosting			BagBoosting-NR		
	$E_{\mathcal{E}}$	E_m	*CFD*	$E_{\mathcal{E}}$	E_m	*CFD*
1	17.43	21.69	0.65	16.95	21.93	0.68
2	18.38	20.18	0.57	17.72	20.40	0.59
3	17.91	20.90	0.55	18.00	20.87	0.57
4	18.10	20.74	0.55	18.19	20.78	0.57
5	17.91	20.99	0.55	18.10	20.82	0.57

TABLE VIII

RESULTS WITH LEUKEMIA DATASET, BAGBOOSTING(-NR).

d_{max}	BagBoosting			BagBoosting-NR		
	$E_{\mathcal{E}}$	E_m	*CFD*	$E_{\mathcal{E}}$	E_m	*CFD*
1	4.92	5.66	0.66	2.50	4.74	0.83
2	6.92	7.08	0.51	3.08	5.13	0.82
3	7.00	7.18	0.55	3.42	5.47	0.8
4	6.92	7.11	0.55	3.00	5.32	0.81
5	7.08	7.21	0.53	3.25	5.42	0.8

One issue that frequently occurs with boosting ensemble methods is that it is not always clear how many iterations of the base learner should be performed. Table IV gives the final error rates for different numbers of iterations of boosting using LogitBoost and LogitBoost-NR on the Colon, Leukemia datasets, with decision stumps as base learners. The error rate drops at different rates for different combinations of datasets and boosting algorithm, but most of the reduction in error had occurred by 25 iterations. We therefore consider that 25 iterations of boosting is a reasonable compromise between prediction accuracy and computing time for comparing the performance of the boosting algorithms and used it for the rest of our experiments.

The results of our investigation of model comlexity, accuracy and diversity showed that *CFD* had the closest relationship with ensemble accuracy, and we will limit our discussion of diversity to the results with *CFD*, the full results including *GD* and *E* are given in the Appendix. From Tables V and VI it can be see that for both datasets with LogitBoost *CFD* drops to zero, or nearly zero with $d_{max} \geq 3$, while the ensemble error rate increases and the model error rate decreases. At $d_{max} = 3$ the error rates of the ensemble and models are identical, while *CFD* has dropped to zero or nearly zero. The results for the Colon dataset are plotted in Figures 1 and 2. From these results it is clear that the change in performance of the ensembles as the tree depth is increased is associated with the change in ensemble diversity.

With LogitBoost-NR, which uses feature non-replacement, *CFD* increased as d_{max} was increased from 1 to 3, then remained constant, while model error decreased, then remained constant. The overall ensemble error at $d_{max} \geq 3$ was greater than at $d_{max} = 1$. Thus, the overall pattern was of increasing diversity associated with decreasing error as the tree depth was increased.

With the Colon data (Table VII) there was little difference between the error rates for BagBoosting and BagBoosting-NR, except with decision stumps as base learners. With decision stumps the final error rate of 16.95% was lower for BagBoosting-NR than the 17.43% with BagBoosting, but the error rate of the models was slightly higher (21.93% *vs.* 21.69%), suggesting higher diversity, which is confirmed by *CFD* being slightly higher.

For both BagBoosting and BagBoosting-NR with deeper trees, the error rates of the ensembles were higher than with stumps but the error rates of the models were lower, *eg.* for BagBoosting the final error rates with trees were in the range 17.91–18.38% and the models in the range 20.18–20.99%, while the figures with stumps were 17.43% and 21.69% respectively. This suggests lower diversity with trees, which is confirmed by the *CFD* values being lower with trees than with stumps (for BagBoosting 0.55–0.57 with trees, but 0.65 with stumps). We therefore conclude that, for the Colon dataset, the non-replacement mechanism in BagBoosting-NR has increased diversity compared to BagBoosting, and this is confirmed by the changes in values of *CFD*. We also conclude that of the three diversity measures used, the changes in *CFD* most closely reflect the true changes in diversity.

With the Leukemia data (Table VIII), when decision stumps were used as base learners, the final and model error rates for BagBoosting were 4.92% and 5.66% respectively.

For BagBoosting-NR, the final and model error rates of 2.50% and 4.74%, respectively, were both lower than with BagBoosting and the *CFD* value was 0.83. These results show the importance of both model accuracy and diversity for achieving low ensemble error rates, since the lowest error rate was with BagBoosting-NR which also had the lowest model error rate, and a *CFD* value of 0.83 which was much larger than the 0.66 with BagBoosting.

Overall, these results show that diversity is critical in order to achieve low error rates with boosting ensemble techniques, and that the lower error rates achieved by the non-replacement algorithms LogitBoost-NR and BagBoosting-NR were due to their achieving higher diversity than the standard versions.

VI. Conclusions

The effect of different numbers of training samples was investigated using LogitBoost and LogitBoost-NR. This is important because microarray datasets often contain small numbers of samples. We found that feature non-replacement enabled deeper decision trees to be used with smaller sampleset sizes than were required for LogitBoost. However, for very small training set sizes, decision stumps gave lower error rates, even with LogitBoost-NR. Thus, feature non-replacement is most useful for improving the performance of boosting algorithms when only small numbers of training samples are available, as is the case with DNA-MD.

We investigated the relationships between model complexity, model error and ensemble diversity and found that for LogitBoost and LogitBoost-NR changes in ensemble error rates with increasing tree depths are associated with changes in diversity.

In the case of BagBoosting and BagBoosting-NR, the lowest error rates were achieved using decision stump base learners, not decision trees, and the reason for this was because with these algorithms, the highest diversity was achieved with decision stumps.

Until now, the general concensus has been that boosting results in increased diversity, but there is no published work using diversity measures to establish how diversity varies under different boosting conditions. By using diversity measures we were able establish the degree of diversity within the boosting ensembles, our results showed that the non-replacement mechanism in the non-replacement algorithms results in higher diversity compared to the corresponding unmodified algorithms. It is this higher diversity that is responsible for their lower error rates.

References

[1] W. Wang. "Some fundamental issues in ensemble methods." In *Proc. IEEE International Joint Conference on Neural Networks, 2008. (IEEE World Congress on Computational Intelligence)*, pp. 2243–2250, 2008.

[2] L. I. Kuncheva and C. J. Whitaker. "Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy." *Machine Learning*, vol. 51, pp. 181–207, 2003.

[3] M. Dettling. "Bagboosting for tumor classification with gene expression data." *Bioinformatics*, vol. 20, pp. 3583–93, 2004.

[4] J. Friedman, T. Hastie, and R. Tibshirani. "Additive logistic regression: A statistical view of boosting." *Annals of Statistics*, vol. 28, pp. 337–407, 2000.

[5] P. M. Long and V. B. Vega. "Boosting and microarray data." *Machine Learning*, vol. 52, pp. 31–44, 2003.

[6] G. R. Guile and W. Wang. "Enhancing Boosting by Feature Non-replacement for Microarray Data Analysis." *Proc. IEEE International Joint Conference on Neural Networks, 2007.*, pp. 430–435, 2007.

[7] G. R. Guile and W. Wang. "Relationship Between Depth of Decision Trees and Boosting Performance." *Proc. IEEE International Joint Conference on Neural Networks, 2008. (IEEE World Congress on Computational Intelligence)*, pp. 2267–2274, 2008.

[8] D. Partridge and W. Krzanowski. "Software diversity: practical statistics for its measurement and exploitation". *Information and Software Technology*, vol. 39, pp. 707–717, 1997.

[9] S.Bian. *Data Mining Ensemble Hierarchy, Diversity and Accuracy.* PhD thesis, University of East Anglia, 2006.

[10] M. Dettling and P. Bühlmann. "Boosting for tumor classification with gene expression data." *Bioinformatics*, vol. 19, pp. 1061–1069, 2003.

[11] U. Alon *et.al.* "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays." *Proceedings of the National Academy of Sciences*, vol. 96, pp. 6745–6750, 1999.

[12] S. Dudoit, J. Fridlyand, and T. P. Speed. "Comparison of discrimination methods for the classification of tumors using gene expression data." *Journal of the American Statistical Association*, vol. 97, pp. 77–87, 2002.

[13] T.R. Golub, *et.al.* "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring." *Science*, vol. 286, pp. 531–537, 1999.

Appendix

The full results for the experiments using diversity measures are given here.

TABLE IX

Results with Colon dataset, LogitBoost.

d_{max}	$E_{\mathcal{E}}$	E_m	*CFD*	*GD*	*E*
1	20.00	36.62	0.66	0.56	0.62
2	25.33	30.39	0.62	0.34	0.27
3	28.29	28.31	0.03	0.00	0.00
4	28.29	28.31	0.03	0.00	0.00
5	28.29	28.31	0.03	0.00	0.00

TABLE X

Results with Colon dataset, LogitBoost-NR.

d_{max}	$E_{\mathcal{E}}$	E_m	*CFD*	*GD*	*E*
1	19.05	37.12	0.65	0.57	0.65
2	18.10	31.5	0.71	0.56	0.51
3	17.91	28.56	0.73	0.54	0.43
4	17.91	28.54	0.73	0.54	0.43
5	17.91	28.53	0.73	0.54	0.43

TABLE XI

Results with Leukemia dataset, LogitBoost.

d_{max}	$E_{\mathcal{E}}$	E_m	*CFD*	*GD*	*E*
1	6.42	17.66	0.67	0.59	0.28
2	10.33	15.69	0.57	0.39	0.15
3	15.42	15.42	0.00	0.00	0.00
4	15.42	15.42	0.00	0.00	0.00
5	15.42	15.42	0.00	0.00	0.00

TABLE XII

Results with Leukemia dataset, LogitBoost-NR.

d_{max}	$E_{\mathcal{E}}$	E_m	*CFD*	*GD*	*E*
1	3.25	21.85	0.81	0.74	0.44
2	3.83	17.68	0.84	0.75	0.35
3	2.58	16.54	0.85	0.75	0.33
4	2.58	16.54	0.85	0.75	0.33
5	2.58	16.54	0.85	0.75	0.33

TABLE XIII

Results with Colon dataset, BagBoosting.

d_{max}	$E_{\mathcal{E}}$	E_m	*CFD*	*GD*	*E*
1	17.43	21.69	0.65	0.34	0.2
2	18.38	20.18	0.57	0.34	0.2
3	17.91	20.90	0.55	0.34	0.2
4	18.10	20.74	0.55	0.34	0.19
5	17.91	20.99	0.55	0.34	0.19

TABLE XIV

Results with Colon dataset, BagBoosting-NR.

d_{max}	$E_{\mathcal{E}}$	E_m	*CFD*	*GD*	*E*
1	16.95	21.93	0.68	0.36	0.21
2	17.72	20.40	0.59	0.36	0.21
3	18.00	20.87	0.57	0.36	0.21
4	18.19	20.78	0.57	0.37	0.21
5	18.10	20.82	0.57	0.36	0.21

TABLE XV

Results with Leukemia dataset, BagBoosting.

d_{max}	$E_{\mathcal{E}}$	E_m	*CFD*	*GD*	*E*
1	4.92	5.66	0.66	0.51	0.06
2	6.92	7.08	0.51	0.48	0.06
3	7.00	7.18	0.55	0.48	0.06
4	6.92	7.11	0.55	0.50	0.05
5	7.08	7.21	0.53	0.50	0.05

TABLE XVI

Results with Leukemia dataset, BagBoosting-NR.

d_{max}	$E_{\mathcal{E}}$	E_m	*CFD*	*GD*	*E*
1	2.50	4.74	0.83	0.64	0.08
2	3.08	5.13	0.82	0.65	0.08
3	3.42	5.47	0.80	0.65	0.08
4	3.00	5.32	0.81	0.63	0.08
5	3.25	5.42	0.80	0.63	0.08