Available online at www.sciencedirect.com
SCIENCE DIRECT®
Information Fusion 6 (2005) 99–111
INFORMATION FUSION
www.elsevier.com/locate/inffus

# Creating diversity in ensembles using artificial data

Prem Melville *, Raymond J. Mooney

*Department of Computer Sciences, University of Texas, 1 University Station C0500, Austin, TX 787120233, USA*

Received 31 October 2003; received in revised form 23 March 2004; accepted 2 April 2004
Available online 14 May 2004

**Abstract**

The diversity of an ensemble of classifiers is known to be an important factor in determining its generalization error. We present a new method for generating ensembles, DECORATE (Diverse Ensemble Creation by Oppositional Relabeling of Artificial Training Examples), that directly constructs diverse hypotheses using additional artificially constructed training examples. The technique is a simple, general meta-learner that can use any strong learner as a base classifier to build diverse committees. Experimental results using decision-tree induction as a base learner demonstrate that this approach consistently achieves higher predictive accuracy than the base classifier, Bagging and Random Forests. DECORATE also obtains higher accuracy than Boosting on small training sets, and achieves comparable performance on larger training sets.


*Keywords:* Artificial data; Decision trees; Bagging; Boosting; Random Forests

## 1. Introduction

One of the major advances in inductive learning in the past decade was the development of *ensemble* or *committee* approaches that learn and retain multiple hypotheses and combine their decisions during classification [11]. For example, *Boosting* [16] is an ensemble method that learns a series of ''weak'' classifiers each one focusing on correcting the errors made by the previous one; and it is currently one of the best generic inductive classification methods [18].

Constructing a *diverse* committee in which each hypothesis is as different as possible, while still maintaining consistency with the training data, is known to be a theoretically important property of a good ensemble method [19]. Although all successful ensemble methods encourage diversity to some extent, few have focused directly on the goal of maximizing diversity. Existing methods that focus on achieving diversity [21,30,34] are fairly complex and are not general *meta-learners* like Bagging [4] and Boosting which can be applied to any base learner to produce an effective committee [41].

We present a new meta-learner DECORATE (Diverse Ensemble Creation by Oppositional Relabeling of Artificial Training Examples) [24], that uses an existing ''strong'' learner (one that provides high accuracy on the training data) to build an effective diverse committee in a simple, straightforward manner. This is accomplished by adding different randomly constructed examples to the training set when building new committee members. These artificially constructed examples are given category labels that *disagree* with the current decision of the committee, thereby easily and directly increasing diversity when a new classifier is trained on the augmented data and added to the committee.

Methods such as Boosting, Bagging and Random Forests [5] provide diversity by sub-sampling or re-weighting the existing training examples. If the training set is small, this limits the amount of ensemble diversity that these methods can obtain. DECORATE ensures diversity on an arbitrarily large set of additional artificial examples. Therefore, one hypothesis is that it will

* Corresponding author. Tel.: +1-512-636-3455; fax: +1-512-471-8885.
*E-mail addresses:* melville@cs.utexas.edu (P. Melville), mooney@cs.utexas.edu (R.J. Mooney).

1566-2535/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.inffus.2004.04.001

result in higher generalization accuracy when the training set is small. This paper presents experimental results on a wide range of UCI data sets comparing Boosting, Bagging, Random Forests and DECORATE, using J48 decision-tree induction as a base learner. J48 is a Java implementation of C4.5 [31] introduced in [41]. Cross-validated learning curves support the hypothesis that "DECORATEd trees" generally result in greater classification accuracy for small training sets. In fact, even given large training sets, DECORATE outperforms Bagging and Random Forests, and is competitive with ADABOOST.

We claim that DECORATE's success is due to its explicit focus on *diversity* while constructing ensembles. We support this claim with additional experiments that show a strong correlation between the *diversity* of DECORATE ensembles and *error reduction*.

## 2. Ensemble diversity

It is well known that the combination of the output of several classifiers is only useful if they disagree on some inputs [17,38]. We refer to the measure of disagreement as the *diversity/ambiguity* of the ensemble. For regression problems, *mean squared error* is generally used to measure accuracy, and *variance* is used to measure diversity. In this setting, Krogh and Vedelsby [19] show that the generalization error, $E$, of the ensemble can be expressed as $E = \overline{E} - \overline{D}$; where $\overline{E}$ and $\overline{D}$ are the mean error and diversity of the ensemble respectively. This result implies that increasing ensemble diversity while maintaining the average error of ensemble members, should lead to a decrease in ensemble error. Unlike regression, for the classification task the above simple linear relationship does not hold between $E$, $\overline{E}$ and $\overline{D}$. But there is still strong reason to believe that increasing diversity should decrease ensemble error [42].

There have been several measures of diversity for classifier ensembles proposed in the literature. In a recent study, Kuncheva and Whitaker [20] compared 10 different measures of diversity. They found that most of these measures are highly correlated. However, to the best of our knowledge, there has not been a conclusive study showing which measure of diversity is the best to use for constructing and evaluating ensembles.

### 2.1. Our diversity measure

For our work, we use the disagreement of an ensemble member with the ensemble's prediction as a measure of diversity. More precisely, if $C_i(x)$ is the prediction of the $i$th classifier (in an ensemble) for the label of $x$; $C^*(x)$ is the prediction of the entire ensemble, then the diversity of the $i$th classifier on example $x$ is given by

$$d_i(x) = \begin{cases} 0: & \text{if } C_i(x) = C^*(x) \\ 1: & \text{otherwise} \end{cases} \quad (1)$$

To compute the diversity of an ensemble of size $n$, on a training set of size $m$, we average the above term:

$$\frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} d_i(x_j) \quad (2)$$

This measure estimates the probability that a classifier in an ensemble will disagree with the prediction of the ensemble as a whole. Our approach is to build ensembles that are consistent with the training data and that attempt to maximize this diversity term.

## 3. Bagging, Boosting and Random Forests

There have been many ensemble methods studied in the literature. In this paper, we compared our approach to the most popular methods—Bagging [4], ADABOOST [16] and Random Forests [5]. We discuss these methods in more detail below.

### 3.1. Bagging

In a Bagging ensemble, each classifier is trained on a set of $m$ training examples, drawn randomly with replacement from the original training set of size $m$. Such a training set is called a *bootstrap replicate* of the original set. Each bootstrap replicate contains, on average, 63.2% of the original training set, with many examples appearing multiple times. Predictions on new examples are made by taking the majority vote of the ensemble.

Bagging is typically applied to learning algorithms that are *unstable*, i.e., a small change in the training set leads to a noticeable change in the model produced. Since each ensemble member is not exposed to the same set of examples, they are different from each other. By voting the predictions of each of these classifiers, Bagging seeks to reduce the error due to variance of the base classifier. Bagging of *stable* learners, such as Naive Bayes, does not reduce error.

### 3.2. Boosting

There are several variations of Boosting that appear in the literature. When we talk about Boosting or ADABOOST, we refer to the ADABOOST. M1 algorithm described in [16] (see Algorithm 1). This algorithm assumes that the base learner can handle weighted example. If the learner cannot directly handle weighted examples, then the training set can be sampled according to a weight distribution to produce a new training set to be used by the learner. ADABOOST maintains a set of weights over the training examples; and in each iteration

$i$, the classifier $C_i$ is trained to minimize the weighted error on the training set. The weighted error of $C_i$ is computed and used to update the distribution of weights on the training examples. The weights of misclassified examples are increased and the weights on correctly classified examples are decreased. The next classifier is trained on the examples with this updated distribution and the process is repeated.

After training, the ensemble's predictions are made using a weighted vote of the individual classifiers: $\sum_i w_i C_i(x)$. The weight of each classifier, $w_i$, is computed according to its accuracy on the weighted example set it was trained on.

**Algorithm 1.** The AdaBoost.M1 algorithm

**Input:**

*BaseLearn*—base learning algorithm

$T$—set of $m$ training examples $\langle (x_1, y_1), \ldots, (x_m, y_m) \rangle$ with labels $y_j \in Y$

$I$—number of Boosting iterations

**Initialize** Distribution of weights on examples, $D_1(x_j) = 1/m$ for all $x_j \in T$

(1) For $i = 1$ to $I$

(2) Train base learner given the distribution, $D_i$, $C_i = BaseLearn(T, D_i)$

(3) Calculate error of $C_i$, $\epsilon_i = \sum_{x_j \in T, C_i(x_j) \neq y_j} D_i(x_j)$

(4) If $\epsilon_i > 1/2$ or then set $I = i - 1$ and abort loop

(5) Set $\beta_i = \epsilon_i / (1 - \epsilon_i)$

(6) Update weights,

$$D_{i+1}(x_j) = D_i(x_j) \times \begin{cases} \beta_i: & \text{if } C_i(x_j) = y_j \\ 1: & \text{otherwise} \end{cases}$$

(7) Normalize weights, $D_{i+1}(x_j) = \frac{D_{i+1}(x_j)}{\sum_{x_j \in T} D_{i+1}(x_j)}$

**Output:** The final hypothesis, $C^* = \arg\max_{y \in Y} \times \sum_{i: C_i(x) = y} \log \frac{1}{\beta_i}$.

AdaBoost is a very effective ensemble method that has been tested extensively by many researchers [1,13,22,32]. Applying AdaBoost to decision trees has been particularly successful, and is considered one of the best off-the-shelf classification methods [18]. The success of AdaBoost has lead to its use in a host of different applications, including text categorization [36], recommender systems [15], and named-entity extraction [8].

Despite its popularity, Boosting does suffer from some drawbacks. In particular, Boosting can fail to perform well given insufficient data [35]. This observation is consistent with the Boosting theory. Boosting also does not perform well when there is a large amount of classification noise (i.e., training examples with incorrect class labels) [13,27].

### 3.3. Random Forests

Breiman [5] introduces Random Forests, where he combines Bagging with random feature selection for decision trees. In this method, each member of the ensemble is trained on a bootstrap replicate as in Bagging. Decision trees are then grown by selecting the feature to split on at each node from $F$ randomly selected features. As in [5], we set $F$ to $\lfloor \log_2(k+1) \rfloor$, where $k$ is the total number of features. And we also do not perform any pruning on the random trees.

Dietterich [12] recommends Random Forests as the method of choice for decision trees, as it compares favorably to AdaBoost and works well even with noise in the training data. The focus of our work has been the development of ensemble methods that are *meta-learners*. Random Forests do not fall in this class, as they can only be applied to decision trees. However, as we applied our methods to tree induction we chose to also compare our results with Random Forests.

## 4. Decorate: Algorithm definition

In Decorate (see Algorithm 2), an ensemble is generated iteratively, first learning a classifier and then adding it to the current ensemble. We initialize the ensemble to contain the classifier trained on the given training data. The classifiers in each successive iteration are trained on the original training data combined with some artificial data. In each iteration, artificial training examples are generated from the data distribution; where the number of examples to be generated is specified as a fraction, $R_{\text{size}}$, of the training set size. The labels for these artificially generated training examples are chosen so as to differ maximally from the current ensemble's predictions. The construction of the artificial data is explained in greater detail in the following section. We refer to the labeled artificially generated training set as the *diversity data*. We train a new classifier on the union of the original training data and the diversity data, thereby forcing it to differ from the current ensemble. Therefore adding this classifier to the ensemble should increase its diversity. While forcing diversity we still want to maintain training accuracy. We do this by rejecting a new classifier if adding it to the existing ensemble decreases its accuracy. This process is repeated until we reach the desired committee size or exceed the maximum number of iterations.

**Algorithm 2.** The Decorate algorithm

**Input:**

*BaseLearn*—base learning algorithm

$T$—set of $m$ training examples $\langle (x_1, y_1), \ldots, (x_m, y_m) \rangle$ with labels $y_j \in Y$

$C_{\text{size}}$—desired ensemble size

$I_{\max}$—maximum number of iterations to build an ensemble

$R_{\text{size}}$—factor that determines number of artificial examples to generate

(1) $i = 1$
(2) $trials = 1$
(3) $C_i = BaseLearn(T)$
(4) Initialize ensemble, $C^* = \{C_i\}$
(5) Compute ensemble error, $\epsilon = \frac{\sum_{x_j \in T, C^*(x_j) \neq y_j} 1}{m}$
(6) While and $i < C_{\text{size}}$ and $trials < I_{\max}$
  (7) Generate $R_{\text{size}} \times |T|$ training examples, $R$, based on distribution of training data
  (8) Label examples in $R$ with probability of class labels inversely proportional to predictions of $C^*$
  (9) $T = T \cup R$
  (10) $C' = BaseLearn(T)$
  (11) $C^* = C^* \cup \{C'\}$
  (12) $T = T - R$, remove the artificial data
  (13) Compute training error, $\epsilon'$, of $C^*$ as in step 5
  (14) If $\epsilon' \leqslant \epsilon$
    (15) $i = i + 1$
    (16) $\epsilon = \epsilon'$
  (17) otherwise
    (18) $C^* = C^* - \{C'\}$
  (19) $trials = trials + 1$

To classify an unlabeled example, $x$, we employ the following method. Each base classifier, $C_i$, in the ensemble $C^*$ provides probabilities for the class membership of $x$. If $\widehat{P}_{C_{i,y}}(x)$ is the estimated probability of example $x$ belonging to class $y$ according to the classifier $C_i$, then we compute the class membership probabilities for the entire ensemble as

$$\widehat{P}_y(x) = \frac{\sum_{C_i \in C^*} \widehat{P}_{C_{i,y}}(x)}{|C^*|}$$

where $\widehat{P}_y(x)$ is the probability of $x$ belonging to class $y$. We then select the most probable class as the label for $x$, i.e., $C^*(x) = \arg\max_{y \in Y} \widehat{P}_y(x)$.

### 4.1. Construction of artificial data

We generate artificial training data by randomly picking data points from an approximation of the training-data distribution. For a numeric attribute, we compute the mean and standard deviation from the training set and generate values from the Gaussian distribution defined by these. For a nominal attribute, we compute the probability of occurrence of each distinct value in its domain and generate values based on this distribution. We use Laplace smoothing so that nominal attribute values not represented in the training set still have a non-zero probability of occurrence. In constructing artificial data points, we make the simplifying assumption that the attributes are independent. It is possible to more accurately estimate the joint probability distribution of the attributes; but this would be time consuming and require a lot of data. Furthermore, the results seem to indicate that we can achieve good performance even with the crude approximation we use.

In each iteration, the artificially generated examples are labeled based on the current ensemble. Given an example, we first find the class membership probabilities predicted by the ensemble. We replace zero probabilities with a small non-zero value and normalize the probabilities to make it a distribution. Labels are then selected, such that the probability of selection is inversely proportional to the current ensemble's predictions. So if the current ensemble predicts the class membership probabilities $\widehat{P}_y(x)$, then a new label is selected based on the new distribution $\widehat{P}'$, where

$$\widehat{P}'_y(x) = \frac{1/\widehat{P}_y(x)}{\sum_y 1/\widehat{P}_y(x)}$$

## 5. Why DECORATE should work

Ensembles of classifiers are often more accurate than their component classifiers if errors made by the ensemble members are uncorrelated [17]. By training classifiers on oppositely labeled artificial examples, DECORATE reduces the correlation between ensemble members. Furthermore, the algorithm ensures that the *training* error of the ensemble is always less than or equal to the error of the base classifier; which usually results in a reduction of *generalization* error. This leads us to our first hypothesis:

**Hypothesis 1.** On average, using the predictions of a DECORATE ensemble will improve on the accuracy of the base classifier.

We believe that diversity is the key to constructing good ensembles, and is thus the basis of our approach. Other ensemble methods also encourage diversity, but in different ways. Bagging implicitly creates ensemble diversity, by training classifiers on different subsets of the data. Boosting fosters diversity, by explicitly modifying the distributions of the training data given to subsequent classifiers. Random Forests produce diversity by training on different subsets of the data and feature sets. However, all these methods rely solely on the *training* data for encouraging diversity. So when the size of the training set is small, they are limited in the amount of diversity they can produce. On the other hand, DECORATE ensures diversity on an arbitrarily large set of additional artificial examples, while still exploiting all the available training data. This leads us to our next hypothesis:

**Hypothesis 2.** DECORATE will outperform Bagging, ADABOOST and Random Forests low on the learning curve i.e. when training sets are small.

## 6. Experimental evaluation

### 6.1. Methodology

To evaluate the performance of DECORATE we ran experiments on 15 representative data sets from the UCI repository [3] that were used in similar studies [32,40]. The data sets are summarized in Table 1. Note that the datasets vary in the numbers of training examples, classes, numeric and nominal attributes; thus providing a diverse testbed.

We compared the performance of DECORATE to that of ADABOOST, Bagging, Random Forests and J48, using J48 as the base learner for the ensemble methods and using the Weka implementations of these methods [41]. For the ensemble methods, we set the ensemble size to 15. Note that in the case of DECORATE we can only specify a *desired* ensemble size; the algorithm terminates if the number of iterations exceeds the maximum limit set even if the desired ensemble size is not reached. For our experiments, we set the maximum number of iterations in DECORATE to 50. We ran experiments varying the amount of artificially generated data, $R_{\text{size}}$ and found that the results do not vary much for the range 0.5–1. However, $R_{\text{size}}$ values lower than 0.5 do adversely affect DECORATE, because there is insufficient artificial data to give rise to high diversity. The results we report are for $R_{\text{size}}$ set to 1, i.e. the number of artificially generated examples is equal to the training set size.

The performance of each learning algorithm was evaluated using 10 complete runs of 10-fold cross-validation. In each 10-fold cross-validation, each data set is randomly split into 10 equal-size segments and results are averaged over 10 trials. For each trial, one segment is set aside for testing, while the remaining data is available for training. To test performance on varying amounts of training data, learning curves were generated by testing the system after training on increasing subsets of the overall training data. Since we would like to summarize results over several data sets of different sizes, we select different *percentages* of the total training-set size as the points on the learning curve.

Table 1
Summary of data sets

| Name | Cases | Classes | Attributes | |
|---|---|---|---|---|
| | | | Numeric | Nominal |
| Anneal | 898 | 6 | 9 | 29 |
| Audio | 226 | 6 | – | 69 |
| Autos | 205 | 6 | 15 | 10 |
| Breast-w | 699 | 2 | 9 | – |
| Credit-a | 690 | 2 | 6 | 9 |
| Glass | 214 | 6 | 9 | – |
| Heart-c | 303 | 2 | 8 | 5 |
| Hepatitis | 155 | 2 | 6 | 13 |
| Colic | 368 | 2 | 10 | 12 |
| Iris | 150 | 3 | 4 | – |
| Labor | 57 | 2 | 8 | 8 |
| Lymph | 148 | 4 | – | 18 |
| Segment | 2310 | 7 | 19 | – |
| Soybean | 683 | 19 | – | 35 |
| Splice | 3190 | 3 | – | 62 |

To compare two learning algorithms across all domains we employ the statistics used in [40], namely the win/draw/loss record and the geometric mean error ratio. The win/draw/loss record presents three values, the number of data sets for which algorithm A obtained better, equal, or worse performance than algorithm B with respect to classification accuracy. We also report the *statistically significant* win/draw/loss record; where a win or loss is only counted if the difference in values is determined to be significant at the 0.05 level by a paired $t$-test.

The geometric mean error ratio is defined as $\sqrt[n]{\prod_{i=1}^{n} \frac{E_A}{E_B}}$ where $E_A$ and $E_B$ are the mean errors of algorithm A and B on the same domain. If the geometric mean error ratio is less than one it implies that algorithm A performs better than B, and vice versa. We compute error ratios to capture the degree to which algorithms outperform each other in win or loss outcomes.

### 6.2. Results

Our results are summarized in Tables 2–5. Each cell in the tables presents the accuracy of DECORATE versus another algorithm. If the difference is statistically significant, then the larger of the two is shown in bold. We varied the training set sizes from 1–100% of the total available data, with more points lower on the learning curve since this is where we expect to see the most difference between algorithms. The bottom of the tables provide summary statistics, as discussed above, for each of the points on the learning curve.

The results in Table 2 confirm our hypothesis that combining the predictions of DECORATE ensembles will, on average, improve the accuracy of the base classifier. DECORATE almost always does better than J48, producing considerable reduction in error throughout the learning curve.

DECORATE has more *significant* wins to losses over Bagging for all points along the learning curve (see Table 3). DECORATE also outperforms Bagging on the geometric mean error ratio. This suggests that even in cases where Bagging beats DECORATE the improvement is less than DECORATE's improvement on Bagging on the rest of the cases.

Similar results are observed in the comparison of DECORATE with Random Forests (see Table 4). DECORATE exhibits superior performance through out the learning curve on both wins/loss records as well as error ratios. The poor performance of Random Forests

Table 2
DECORATE versus J48

| Dataset | 1% | 2% | 5% | 10% | 20% | 30% | 40% | 50% | 75% | 100% |
|---|---|---|---|---|---|---|---|---|---|---|
| Anneal | **75.29/** | **78.14/** | **85.24/** | **92.26/** | **96.48/** | **97.36/** | **97.73/** | **98.16/** | 98.39/ | 98.71/ |
| | 72.49 | 75.31 | 82.08 | 89.28 | 95.57 | 96.47 | 97.3 | 97.93 | 98.35 | 98.55 |
| Audio | 16.66/ | 23.73/ | 41.72/ | **55.42/** | **64.09/** | **67.62/** | **70.46/** | **72.82/** | **77.8/** | **82.1/** |
| | 16.66 | 23.07 | 41.17 | 51.67 | 60.59 | 64.84 | 68.11 | 70.77 | 75.15 | 77.22 |
| Autos | 24.33/ | 29.6/ | **36.73/** | 42.89/ | 52.2/ | **59.86/** | **64.77/** | **68.6/** | **78/** | **83.64/** |
| | 24.33 | 29.01 | 34.37 | 41.22 | 50.53 | 53.92 | 59.68 | 65.24 | 73.15 | 81.72 |
| Breast-w | **92.38/** | **94.12/** | **95.06/** | **95.64/** | **95.55/** | **95.91/** | **96.2/** | **96.01/** | **96.28/** | **96.31/** |
| | 74.73 | 87.34 | 89.42 | 92.21 | 93.09 | 93.36 | 93.85 | 94.24 | 94.65 | 95.01 |
| Credit-a | 71.78/ | 74.83/ | 80.61/ | 83.09/ | 84.38/ | 84.68/ | **85.22/** | **85.57/** | 85.61/ | 85.93/ |
| | 69.54 | **77.46** | 81.57 | 82.35 | 84.29 | 84.59 | 84.41 | 84.78 | 85.43 | 85.57 |
| Glass | 31.69/ | **35.86/** | **44.5/** | **55.4/** | **61.77/** | **66.01/** | **68.07/** | **68.85/** | **72.73/** | **72.77/** |
| | 31.69 | 32.96 | 38.34 | 46.62 | 54.16 | 60.63 | 61.38 | 63.69 | 67.53 | 67.77 |
| Heart-c | **58.66/** | **65.11/** | **73.55/** | **75.05/** | **77.66/** | **78.34/** | **79.09/** | **79.46/** | **78.74/** | **78.48/** |
| | 49.57 | 58.03 | 67.71 | 70.15 | 73.44 | 74.61 | 74.78 | 75.62 | 76.7 | 77.17 |
| Hepatitis | 52.33/ | **72.14/** | **76.8/** | 79.48/ | **80.7/** | **81.81/** | **81.65/** | **83.19/** | **82.99/** | **82.62/** |
| | 52.33 | 65.93 | 72.75 | 78.25 | 78.61 | 78.63 | 79.35 | 79.57 | 79.04 | 79.22 |
| Colic | **58.37/** | 66.58/ | 75.85/ | 79.54/ | 81.33/ | 82.47/ | 83.02/ | 83.1/ | 84.02/ | 84.69/ |
| | 52.85 | 65.31 | 74.37 | 79.94 | **82.71** | **83.41** | 83.55 | **84.66** | **85.18** | 85.16 |
| Iris | 33.33/ | **50.27/** | **80.67/** | **91.53/** | **93.2/** | **94.2/** | **94.73**/93 | **94.4/** | 94.53/ | 94.67/ |
| | 33.33 | 33.33 | 59.33 | 84.33 | 91.33 | 92.73 | | 93.33 | 94.07 | 94.73 |
| Labor | 54.27/ | 54.27/ | **67.63/** | **70.23/** | **79.77/** | **83**/73.7 | **84.17/** | **83.43/** | **89.73/** | **89.73/** |
| | 54.27 | 54.27 | 58.93 | 64.77 | 70.07 | | 75.17 | 75.8 | 77.4 | 78.8 |
| Lymph | 48.39/ | **53.62/** | **65.06/** | **71.2/** | **76.74/** | **78.84/** | **78.17/** | **78.99/** | **79.14/** | **79.08/** |
| | 48.39 | 46.64 | 60.39 | 68.21 | 70.79 | 73.58 | 74.53 | 73.34 | 75.63 | 76.06 |
| Segment | **67.03/** | **81.16/** | **89.61/** | **92.83/** | **94.88/** | **95.94/** | **96.47/** | **96.93/** | **97.58/** | **98.03/** |
| | 52.43 | 73.26 | 85.41 | 89.34 | 92.22 | 93.37 | 94.34 | 94.77 | 95.94 | 96.79 |
| Soybean | **19.51/** | **32.4/** | **55.36/** | **73.06/** | **85.14/** | **88.27/** | **90.22/** | **91.4/** | **92.75/** | **93.89/** |
| | 13.69 | 22.32 | 42.94 | 59.04 | 74.49 | 81.59 | 84.78 | 86.89 | 89.44 | 91.76 |
| Splice | **62.77/** | 67.8/ | 77.37/ | 82.55/ | **88.24/** | 90.47/ | 91.84/ | 92.41/ | 93.44/ | 93.92/ |
| | 59.92 | 68.69 | 77.49 | 82.58 | 87.98 | 90.44 | 91.77 | 92.4 | 93.47 | 94.03 |
| Win/draw/loss | 15/0/0 | 13/0/2 | 13/0/2 | 13/0/2 | 14/0/1 | 14/0/1 | 14/0/1 | 14/0/1 | 13/0/2 | 12/0/3 |
| Sig. W/D/L | 7/8/0 | 9/5/1 | 11/4/0 | 10/5/0 | 12/2/1 | 12/2/1 | 13/2/0 | 13/1/1 | 10/4/1 | 10/4/1 |
| GM error ratio | 0.8627 | 0.8661 | 0.8099 | 0.8104 | 0.8172 | 0.8056 | 0.8081 | 0.8251 | 0.8173 | 0. 8303 |

maybe because we are using only 15 trees. Random Forests may benefit from using larger ensembles; more so than other methods. However, to do a fair comparison we use the same ensemble size for all methods.

DECORATE outperforms ADABOOST early on the learning curve both on significant wins/draw/loss record and geometric mean ratio; however, the trend is reversed when given 75% or more of the data. Note that even with large amounts of training data, DECORATE's performance is quite competitive with ADABOOST—given 100% of the training data, DECORATE produces higher accuracies on 6 out of 15 data sets. It has been observed in previous studies [1,40] that while ADABOOST usually significantly reduces the error of the base learner, it occasionally increases it, often to a large extent. DECORATE does not have this problem as is clear from Table 2.

On many data sets, DECORATE achieves the same or higher accuracy as Bagging, ADABOOST or Random Forests with far fewer training examples. Fig. 1 shows learning curves that clearly demonstrate this point. Hence, in domains where little data is available or acquiring labels is expensive, DECORATE has a significant advantage over other ensemble methods.

### 6.3. DECORATE with large training sets

The learning curve evaluation clearly shows DECORATE's advantage when training sets are small. The results also indicate that DECORATE begins to lose out to ADABOOST with larger training sets. However, we claim that the performance of both systems on large training sets is comparable. To support this we ran additional experiments comparing DECORATE with ADABOOST on a larger collection of 33 UCI datasets. We ran 10-fold cross-validation using all the available training examples for each of the datasets. The results of this study are summarized in Table 6. We observe that on 25 of the 33 datasets there was no statistically significant difference between the two systems. And DECORATE significantly outperforms ADABOOST on four of the eight remaining datasets. We conjecture that when

Table 3
DECORATE versus Bagging

| Dataset | 1% | 2% | 5% | 10% | 20% | 30% | 40% | 50% | 75% | 100% |
|---|---|---|---|---|---|---|---|---|---|---|
| Anneal | 75.29/ | **78.14/** | **85.24/** | **92.26/** | **96.48/** | **97.36/** | **97.73/** | **98.16/** | 98.39/ | 98.71/ |
| | 74.57 | 76.42 | 82.88 | 89.87 | 95.67 | 96.89 | 97.34 | 97.78 | 98.53 | 98.83 |
| Audio | **16.66/** | 23.73/ | **41.72/** | **55.42/** | **64.09/** | 67.62/ | 70.46/ | 72.82/ | 77.8/ | **82.1/** |
| | 12.98 | 23.68 | 38.55 | 51.34 | 61.76 | 66. 9 | 70.29 | 73.07 | 77.32 | 80.71 |
| Autos | **24.33/** | 29.6/ | 36.73/ | 42.89/ | 52.2/ | 59.86/ | 64.77/ | 68.6/ | 78/ | 83.64/ |
| | 22.16 | 28 | 35.88 | 44.65 | 54.32 | 59.67 | 65.6 | 69.88 | 77.97 | 83.12 |
| Breast-w | **92.38/** | **94.12/** | **95.06/** | **95.64/** | **95.55/** | **95.91/** | **96.2/** | **96.01/** | 96.28/ | 96.31/ |
| | 76.74 | 88.07 | 90.88 | 93.41 | 94.42 | 94.95 | 94.95 | 95.55 | 96.07 | 96.3 |
| Credit-a | 71.78/ | 74.83/ | 80.61/ | 83.09/ | 84.38/ | 84.68/ | 85.22/ | 85.57/ | 85.61/ | 85.93/ |
| | 69.54 | **77.99** | **82.58** | 83.9 | **85.13** | **85.78** | 85.59 | 85.64 | **86.12** | 85.96 |
| Glass | **31.69/** | **35.86/** | **44.5/** | **55.4/** | **61.77/** | **66.01/** | 68.07/ | 68.85/ | 72.73/ | 72.77/ |
| | 24.85 | 31.47 | 40.87 | 49.6 | 58.9 | 64.35 | 66.3 | 68.44 | 72 | **74.67** |
| Heart-c | **58.66/** | **65.11/** | **73.55/** | **75.05/** | **77.66/** | 78.34/ | 79.09/ | 79.46/ | 78.74/ | 78.48/ |
| | 50.56 | 55.67 | 68.77 | 73.17 | 76.12 | 77.9 | 78.44 | 79.11 | 79.05 | 78.68 |
| Hepatitis | 52.33/ | **72.14/** | 76.8/ | 79.48/ | 80.7/ | 81.81/ | 81.65/ | **83.19/** | **82.99/** | 82.62/ |
| | 52.33 | 63.18 | 75.2 | 78.64 | 80.42 | 81.07 | 81.22 | 81.06 | 80.87 | 81.34 |
| Colic | **58.37/** | **66.58/** | 75.85/ | 79.54/ | 81.33/ | 82.47/ | 83.02/ | 83.1/ | 84.02/ | 84.69/ |
| | 53.14 | 63.83 | 76.44 | 80.06 | **83.04** | **83.58** | **83.98** | **84.47** | **85.4** | 85.34 |
| Iris | 33.33/ | **50.27/** | **80.67/** | **91.53/** | **93.2/** | **94.2/** | **94.73/** | **94.4/** | 94.53/ | 94.67/ |
| | 33.33 | 33.33 | 60.47 | 81.4 | 90.67 | 92.33 | 92.87 | 93.6 | 94.47 | 94.73 |
| Labor | 54.27/ | 54.27/ | **67.63/** | **70.23/** | **79.77/** | **83/** | **84.17/** | **83.43/** | **89.73/** | **89.73/** |
| | 54.27 | 54.27 | 56.27 | 65.9 | 74.97 | 75.67 | 76.27 | 78.6 | 80.83 | 85.87 |
| Lymph | 48.39/ | **53.62/** | **65.06/** | 71.2/ | **76.74/** | **78.84/** | 78.17/ | **78.99/** | **79.14/** | 79.08/ |
| | 48.39 | 47.11 | 60.12 | 69.68 | 73.6 | 76.58 | 77.68 | 76.98 | 76.8 | 77.97 |
| Segment | **67.03/** | **81.16/** | **89.61/** | **92.83/** | **94.88/** | **95.94/** | **96.47/** | **96.93/** | **97.58/** | **98.03/** |
| | 55.88 | 76.36 | 87.42 | 91.01 | 93.4 | 94.65 | 95.26 | 95.82 | 96.78 | 97.41 |
| Soybean | **19.51/** | **32.4/** | **55.36/** | **73.06/** | **85.14/** | **88.27/** | **90.22/** | **91.4/** | **92.75/** | **93.89/** |
| | 14.56 | 24.58 | 47.46 | 65.45 | 79.29 | 85.05 | 87.89 | 89.22 | 91.56 | 92.71 |
| Splice | 62.77/ | 67.8/ | 77.37/ | 82.55/ | 88.24/ | 90.47/ | 91.84/ | 92.41/ | 93.44/ | 93.92/ |
| | 62.52 | **72.36** | **80.5** | **85.44** | **89.5** | **91.44** | **92.4** | **93.07** | **94.06** | **94.53** |
| Win/draw/loss | 15/0/0 | 13/0/2 | 12/0/3 | 11/0/4 | 11/0/4 | 12/0/3 | 11/0/4 | 10/0/5 | 10/0/5 | 8/0/7 |
| Sig. W/D/L | 8/7/0 | 10/3/2 | 10/3/2 | 9/5/1 | 10/2/3 | 8/4/3 | 6/7/2 | 8/5/2 | 5/7/3 | 4/9/2 |
| GM error ratio | 0.8727 | 0.8785 | 0.8552 | 0.8655 | 0.8995 | 0.9036 | 0.8979 | 0.9214 | 0.9312 | 0.9570 |

the training set is large enough the classifiers produced may be reaching the Bayes-optimal performance, which makes improvements impossible. Such a ceiling effect has been observed in other empirical comparisons of ensemble methods [1]. However, by looking at performance on varying training set sizes we can get a better understanding of the relative effectiveness of two learners. Therefore we strongly believe that generating learning curves is crucial for making a good comparison between systems.

### 6.4. Diversity versus error reduction

Our approach is based on the claim that ensemble diversity is critical to error reduction. We attempt to validate this claim by measuring the correlation between diversity and error reduction. We ran DECORATE at 10 different settings of $R_{\text{size}}$ ranging from 0.1 to 1.0, thus varying the diversity of ensembles produced. We then compared the diversity of ensembles with the reduction in generalization error, by computing Spearman's rank correlation between the two. Diversity of an ensemble is computed as the mean diversity of the ensemble members (as given by Eq. (2)). We compared ensemble diversity with the *ensemble error reduction*, i.e., the difference between the average error of the ensemble members and the error of the entire ensemble (as in [10]). We found that the correlation coefficient between diversity and ensemble error reduction is 0.6602 ($p \ll 10^{-50}$), which is fairly strong. [1] Furthermore, we compared diversity with the *base error reduction*, i.e., the difference between the error of the base classifier and the ensemble error. The base error reduction gives a better indication of the improvement in performance of an ensemble over the base classifier. The correlation of diversity versus the base error reduction is 0.1607 ($p \ll 10^{-50}$). We note that even though this correlation

[1] The $p$-value is the probability of getting a correlation as large as the observed value by random chance, when the true correlation is zero [37].

Table 4
DECORATE versus Random Forests

| Dataset | 1% | 2% | 5% | 10% | 20% | 30% | 40% | 50% | 75% | 100% |
|---|---|---|---|---|---|---|---|---|---|---|
| Anneal | **75.29/** | **78.14/** | **85.24/** | **92.26/** | **96.48/** | 97.36/ | 97.73/ | 98.16/ | 98.39/ | 98.71/ |
| | 72.07 | 76.69 | 84.21 | 90.89 | 95.71 | **97.54** | **98.16** | **98.64** | **99.01** | **99.23** |
| Audio | **16.66/** | **23.73/** | **41.72/** | **55.42/** | **64.09/** | **67.62/** | **70.46/** | **72.82/** | **77.8/** | **82.1/** |
| | 12.98 | 20.47 | 26.61 | 30.73 | 41.93 | 51.14 | 57.05 | 60.69 | 69.43 | 73.47 |
| Autos | **24.33/** | 29.6/ | 36.73/ | 42.89/ | 52.2/ | 59.86/ | 64.77/ | 68.6/ | 78/ | 83.64/ |
| | 22.16 | **31.65** | **36.76** | **44.76** | **57.04** | **63.53** | **69.43** | **73.81** | **79.95** | **85.24** |
| Breast-w | **92.38/** | **94.12/** | **95.06/** | **95.64/** | **95.55/** | **95.91/** | **96.2/** | **96.01/** | **96.28/** | **96.31/** |
| | 81.52 | 88.7 | 92.07 | 93.49 | 94.37 | 94.94 | 95.41 | 95.77 | 95.84 | 95.85 |
| Credit-a | **71.78/** | **74.83/** | **80.61/** | **83.09/** | **84.38/** | **84.68/** | **85.22/** | **85.57/** | **85.61/** | **85.93/** |
| | 60.61 | 64.65 | 70.38 | 72.87 | 76.55 | 78.36 | 79.54 | 81.13 | 82.35 | 83.25 |
| Glass | **31.69/** | **35.86/** | **44.5/** | **55.4/** | **61.77/** | **66.01/** | **68.07/** | 68.85/ | 72.73/ | 72.77/ |
| | 24.85 | 31.79 | 42.19 | 52.84 | 59.96 | 63.4 | 67.06 | **69.14** | **73.55** | **76.4** |
| Heart-c | **58.66/** | **65.11/** | **73.55/** | **75.05/** | **77.66/** | **78.34/** | **79.09/** | **79.46/** | 78.74/ | 78.48/ |
| | 50.06 | 54.78 | 66.86 | 72.61 | 76.14 | 76.52 | 77.63 | 78.58 | **79.28** | **79.92** |
| Hepatitis | **52.33/** | **72.14/** | **76.8/** | **79.48/** | **80.7/** | **81.81/** | **81.65/** | **83.19/** | 82.99/ | 82.62/ |
| | 52.33 | 70.36 | 74.51 | 77.26 | 80.37 | 81.7 | 81 | 81.72 | **83.05** | **82.9** |
| Colic | **58.37/** | **66.58/** | **75.85/** | **79.54/** | **81.33/** | **82.47/** | **83.02/** | **83.1**/81 | **84.02/** | **84.69/** |
| | 52.73 | 56.62 | 64.52 | 68.03 | 74.6 | 77.15 | 79.54 | | 83.36 | 84.34 |
| Iris | **33.33/** | **50.27/** | **80.67/** | **91.53/** | **93.2/** | **94.2/** | **94.73/** | **94.4/** | **94.53/** | **94.67/** |
| | 33.33 | 47 | 67.07 | 83.33 | 91.13 | 94 | 94.47 | 94.33 | 94.4 | 94.2 |
| Labor | **54.27/** | **54.27/** | **67.63/** | **70.23/** | **79.77/** | **83/** | **84.17/** | **83.43/** | **89.73/** | **89.73/** |
| | 54.27 | 54.27 | 65.3 | 69.57 | 75.23 | 79.6 | 80.03 | 81.6 | 82.83 | 88.1 |
| Lymph | **48.39/** | **53.62/** | **65.06/** | **71.2/** | **76.74/** | **78.84/** | **78.17/** | **78.99/** | **79.14/** | 79.08/ |
| | 48.39 | 52.06 | 60.55 | 65.48 | 68.18 | 71.37 | 73.55 | 76.34 | 77.51 | **79.28** |
| Segment | **67.03/** | **81.16/** | **89.61/** | **92.83/** | **94.88/** | **95.94/** | **96.47/** | **96.93/** | **97.58/** | **98.03/** |
| | 59.46 | 74.16 | 86.45 | 91.25 | 94.16 | 95.42 | 95.99 | 96.39 | 97.18 | 97.59 |
| Soybean | 19.51/ | 32.4/ | **55.36/** | **73.06/** | **85.14/** | **88.27/** | **90.22/** | **91.4/** | **92.75/** | **93.89/** |
| | **25.82** | **38.3** | 54.57 | 66.52 | 78.4 | 83.94 | 87 | 88.54 | 90.73 | 91.38 |
| Splice | **62.77/** | **67.8/** | **77.37/** | **82.55/** | **88.24/** | **90.47/** | **91.84/** | **92.41/** | **93.44/** | **93.92/** |
| | 49.37 | 51.34 | 51.92 | 51.97 | 52.03 | 52.11 | 52.17 | 52.23 | 52.42 | 52.59 |
| Win/draw/loss | 14/0/1 | 13/0/2 | 14/0/1 | 14/0/1 | 14/0/1 | 13/0/2 | 13/0/2 | 12/0/3 | 10/0/5 | 9/0/6 |
| Sig.W/D/L | 10/4/1 | 8/6/1 | 10/5/0 | 13/2/0 | 11/3/1 | 10/4/1 | 10/3/2 | 7/6/2 | 7/6/2 | 6/5/4 |
| GM error ratio | 0.8603 | 0.8495 | 0.7814 | 0.7433 | 0.7486 | 0.7763 | 0.7915 | 0.8203 | 0.8171 | 0. 8364 |

is weak, it is still a *statistically significant* positive correlation. These results reinforce our belief that increasing ensemble diversity is a good approach to reducing generalization error.

### 6.5. Influence of ensemble size

To determine how the performance of DECORATE changes with ensemble size, we ran experiments with increasing sizes. We compared results for training on 20% of available data since the advantage of DECORATE is most noticeable low on the learning curve. The results were produced using 10-fold cross-validation. We present graphs of *accuracy* versus *ensemble size* for five representative datasets (see Fig. 2). The performance on other datasets is similar. We note, in general, that the accuracy of DECORATE increases with ensemble size; though on most datasets, the performance levels out with an ensemble size of 10–25.

In our main results in Section 6.2 we used committees of size 15 for all methods. However, different ensemble methods may be affected to varying extents by committee size. To verify that the other ensemble methods are not being disadvantaged by smaller ensembles, we ran additional experiments with ensemble size set to 100. Learning curves were generated as in Section 6.1 on the four datasets presented in Fig. 1. For these experiments, we set the maximum number of iterations in DECORATE to 300. The results of testing with larger ensembles is presented in Fig. 3. Apart from slight improvements in accuracies for all methods, the trends of the results are the same as with ensembles of size 15.

## 7. Related work

### 7.1. Explicit diversity-based approaches

DECORATE differs from ensemble methods, such as Bagging, in that it *explicitly* tries to foster ensemble diversity. There have been other approaches to using diversity to guide ensemble creation. We list some of them below.

Liu and Yao [21] and Rosen [34] simultaneously train neural networks in an ensemble using a correlation

Table 5
DECORATE versus ADABOOST

| Dataset | 1% | 2% | 5% | 10% | 20% | 30% | 40% | 50% | 75% | 100% |
|---|---|---|---|---|---|---|---|---|---|---|
| Anneal | **75.29**/ | 78.14/ | 85.24/ | 92.26/ | 96.48/ | 97.36/ | 97.73/ | 98.16/ | 98.39/ | 98.71/ |
| | 73.02 | 77.12 | **87.51** | **94.16** | **97.13** | **97.95** | **98.54** | **98.8** | **99.23** | **99.68** |
| Audio | 16.66/ | 23.73/ | **41.72**/ | **55.42**/ | 64.09/ | 67.62/ | 70.46/ | 72.82/ | 77.8/ | 82.1/ |
| | 16.66 | 23.41 | 40.24 | 52.7 | 64.15 | 68.91 | **73.07** | **75.92** | **81.74** | **84.52** |
| Autos | 24.33/ | 29.6/ | 36.73/ | 42.89/ | 52.2/ | 59.86/ | 64.77/ | 68.6/ | 78/ | 83.64/ |
| | 24.33 | 29.71 | 34.2 | 43.28 | **56.13** | **62.2** | **69.14** | **72.03** | **80.28** | **85.28** |
| Breast-w | **92.38**/ | **94.12**/ | **95.06**/ | **95.64**/ | **95.55**/ | 95.91/ | 96.2/ | 96.01/ | 96.28/ | 96.31/ |
| | 74.73 | 87.84 | 91.15 | 93.75 | 94.85 | 95.72 | 95.84 | 95.87 | 96.3 | 96.47 |
| Credit-a | **71.78**/ | 74.83/ | 80.61/ | **83.09**/ | **84.38**/ | 84.68/ | **85.22**/ | **85.57**/ | 85.61/ | 85.93/ |
| | 68.8 | 75.3 | 79.68 | 81.14 | 83.04 | 84.22 | 84.13 | 84.58 | 84.93 | 85.42 |
| Glass | 31.69/ | **35.86**/ | **44.5**/ | **55.4**/ | **61.77**/ | 66.01/ | 68.07/ | 68.85/ | 72.73/ | 72.77/ |
| | 31.69 | 32.93 | 40.71 | 49.78 | 58.03 | 64.33 | 66.93 | 68.69 | **74.69** | **76.06** |
| Heart-c | **58.66**/ | **65.11**/ | **73.55**/ | **75.05**/ | 77.66/ | 78.34/ | 79.09/ | 79.46/ | 78.74/ | 78.48/ |
| | 49.57 | 58.65 | 70.71 | 72.5 | 76.65 | 78.26 | 78.96 | 79.55 | 79.06 | 79.22 |
| Hepatitis | 52.33/ | **72.14**/ | **76.8**/ | **79.48**/ | 80.7/ | **81.81**/ | 81.65/ | 83.19/ | 82.99/ | 82.62/ |
| | 52.33 | 65.93 | 73.01 | 76.95 | 79.44 | 79.22 | 81.27 | 82.63 | 83.24 | 82.71 |
| Colic | **58.37**/ | 66.58/ | **75.85**/ | **79.54**/ | **81.33**/ | **82.47**/ | **83.02**/ | **83.1**/ | **84.02**/ | **84.69**/ |
| | 52.85 | 67.18 | 72.85 | 77.17 | 79.36 | 79.24 | 79.51 | 80.22 | 80.59 | 81.93 |
| Iris | 33.33/ | **50.27**/ | **80.67**/ | **91.53**/ | **93.2**/ | **94.2**/93 | **94.73**/ | **94.4**/ | 94.53/ | 94.67/ |
| | 33.33 | 33.33 | 66.2 | 84.53 | 90.73 | | 93.33 | 93.53 | 94.2 | 94.2 |
| Labor | 54.27/ | 54.27/ | **67.63**/ | **70.23**/ | **79.77**/ | **83**/76.9 | **84.17**/ | **83.43**/ | **89.73**/ | **89.73**/ |
| | 54.27 | 54.27 | 58.93 | 65.1 | 73.2 | | 79.57 | 80.1 | 84.07 | 86.37 |
| Lymph | 48.39/ | **53.62**/ | **65.06**/ | 71.2/ | **76.74**/ | 78.84/ | 78.17/ | 78.99/ | 79.14/ | 79.08/ |
| | 48.39 | 46.64 | 60.54 | 69.57 | 74.16 | 78.62 | **80.35** | 79.88 | **80.96** | **81.75** |
| Segment | **67.03**/ | **81.16**/ | **89.61**/ | 92.83/ | 94.88/ | 95.94/ | 96.47/ | 96.93/ | 97.58/ | 98.03/ |
| | 60.22 | 77.38 | 88.5 | 92.71 | 95.01 | 96.03 | **96.9** | **97.23** | **98** | **98.34** |
| Soybean | **19.51**/ | **32.4**/ | **55.36**/ | **73.06**/ | 85.14/ | 88.27/ | 90.22/ | 91.4/ | 92.75/ | **93.89**/ |
| | 14.26 | 23.36 | 49.37 | 69.49 | 85.01 | 88.37 | 90.04 | 90.89 | 92.57 | 92.88 |
| Splice | 62.77/ | 67.8/ | 77.37/ | 82.55/ | 88.24/ | 90.47/ | **91.84**/ | **92.41**/ | **93.44**/ | 93.92/ |
| | **65.11** | **73.9** | **82.22** | **86.13** | 88.27 | 89.82 | 90.8 | 90.78 | 92.63 | 93.59 |
| Win/draw/loss | 14/0/1 | 11/0/4 | 13/0/2 | 12/0/3 | 10/0/5 | 10/0/5 | 10/0/5 | 9/0/6 | 6/0/9 | 6/0/9 |
| Sig. W/D/L | 7/7/1 | 8/6/1 | 11/2/2 | 10/3/2 | 7/6/2 | 4/9/2 | 5/5/5 | 5/6/4 | 3/6/6 | 3 /6/6 |
| GM error ratio | 0.8812 | 0.8937 | 0.8829 | 0.9104 | 0.9407 | 0.9598 | 0.9908 | 0.9957 | 1.0377 | 1.0964 |

penalty term in their error functions. McKay and Abbass [23] use a similar method with a different penalty function. Brown and Wyatt [6] provide a good theoretical analysis of these methods, commonly referred to as Negative Correlation Learning. Opitz and Shavlik [30] and Opitz [28] use a genetic algorithm to search for a good ensemble of networks. To guide the search they use an objective function that incorporates both an accuracy and diversity term.

Turner and Ghosh [38] reduce the correlation between classifiers in an ensemble by exposing them to different feature subsets. They train $m$ classifiers, one corresponding to each class in a $m$-class problem. For each class, a subset of features that have a low correlation to that class is eliminated. The degree of correlation between classifiers can be controlled by the amount of features that are eliminated. This method, called *input decimation*, has been further explored in [39].

Zenobi and Cunningham [42] also build ensembles based on different feature subsets. In their approach, feature selection is done using a hill-climbing strategy based on classifier error and diversity. A classifier is rejected if the improvement of one of the metrics leads to a "substantial" deterioration of the other; where "substantial" is defined by a pre-set threshold.

All these approaches attempt to simultaneously optimize diversity and error of *individual* ensemble members. On the other hand, DECORATE focuses on reducing the error of the *entire* ensemble by increasing diversity. At no point does the training accuracy of the ensemble go below that of the base classifier; however, this is a possibility with previous methods. Furthermore, to the best of our knowledge, apart from [28], none of the previous studies compared their methods with standard ensemble approaches such as Boosting and Bagging.

Compared to boosting, which requires a "weak" base learner that does not completely fit the training data (boosting terminates once it constructs a hypothesis with zero training error), DECORATE requires a strong learner, otherwise the artificial diversity training data may prevent it from adequately fitting the real data. When applying boosting to strong base learners, they must first be appropriately weakened in order to benefit from

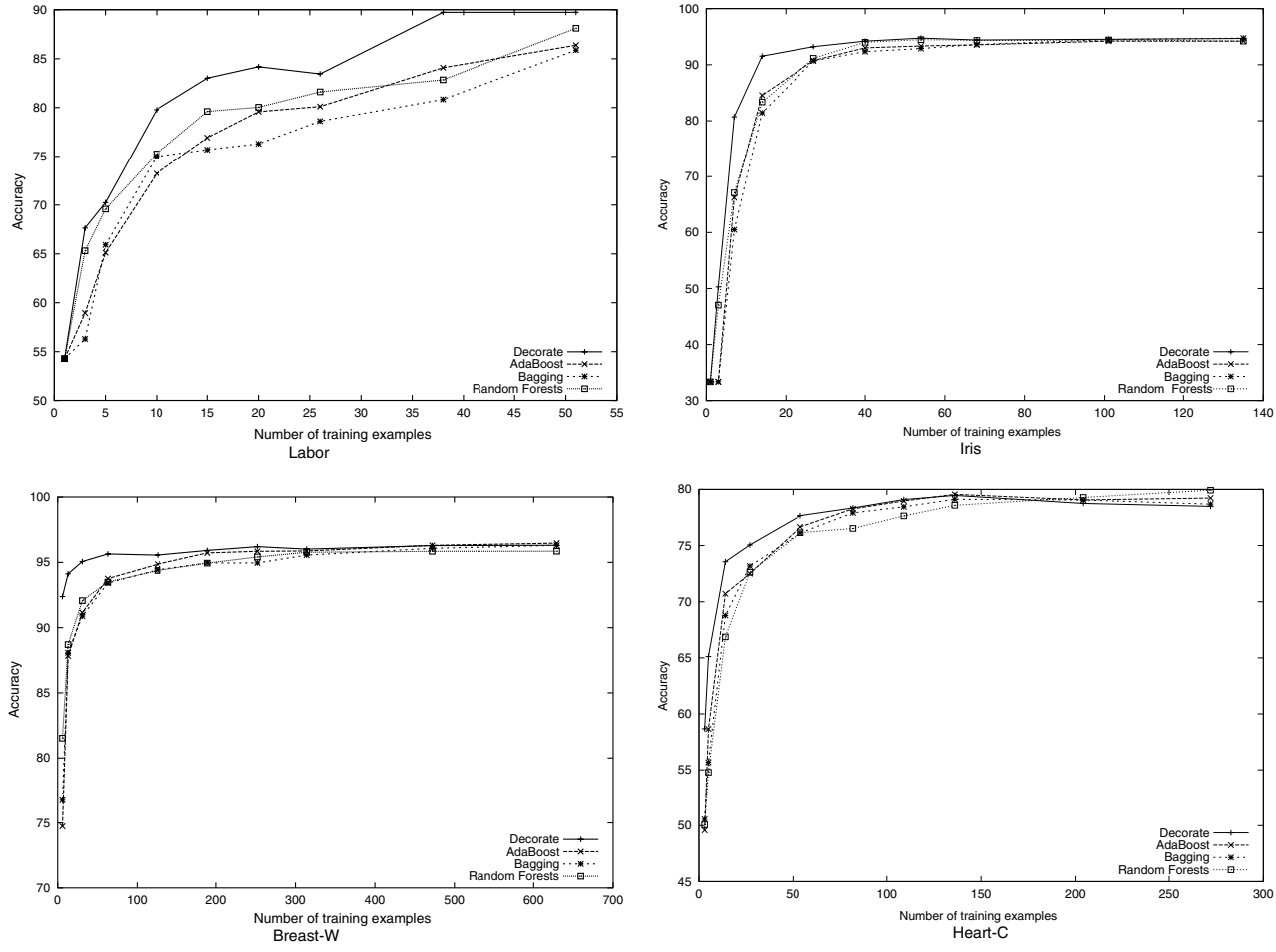


Fig. 1. DECORATE compared to ADABOOST, Bagging and Random Forests.

boosting, e.g., boosting pruned trees outperforms unpruned trees (which completely fit the training data).

### 7.2. Use of artificial examples

One ensemble approach that also utilizes artificial training data is the active learning method introduced in [7]. Rather than to improve accuracy, the goal of the committee here is to select good new training examples using the existing training data. The labels of the artificial examples are selected to produce hypotheses that more faithfully represent the entire version space rather than to produce diversity. Cohn's approach labels artificial data either all positive or all negative to encourage, respectively, the learning of more general or more specific hypotheses.

Another application of artificial examples for ensembles is Combined Multiple Models (CMMs) [14]. The aim of CMMs is to improve the comprehensibility of an ensemble of classifiers, by approximating it by a single classifier. Artificial examples are generated and labeled by a voted ensemble. They are then added to the original training set. The base learner is trained on this augmented training set to produce an approximation of the ensemble. The role of artificial examples here is to create less complex models, *not* to improve classification accuracy.

Craven and Shavlik [9] use artificial examples to learn decision trees from trained neural networks. As in CMMs, the goal here is to create more comprehensible models from existing classifiers. The artificial examples created are labeled by a given neural network, and then used in constructing an equivalent decision tree.

To prevent overfitting in neural networks often noise is added to the inputs during training. This is generally done by adding a random vector to the feature vector of each training example. These *perturbed* or *jittered* examples may also be considered as artificial examples. Quite often training with noise improves network generalization [2,33]. Adding noise to training examples differs from our method of constructing examples from the data distribution. Furthermore, unlike adding noise, DECORATE systematically labels artificial examples to improve generalization.

Table 6
DECORATE versus ADABOOST with large training sets

| Dataset | ADABOOST | DECORATE |
|---|---|---|
| Audio | 84.45 | 83.6 |
| Anneal | **99.55** | 98.66 |
| Colic | 83.13 | 85.58 |
| Balance-scale | 78.56 | **80.98** |
| Credit-g | 72.40 | 73.6 |
| Pima-diabetes | 72.52 | **75.52** |
| Glass | 76.58 | 72.34 |
| Heart-c | 81.15 | 77.51 |
| Heart-h | 78.56 | 79.98 |
| Credit-a | 85.94 | 87.39 |
| Autos | 86.33 | 85.79 |
| Kr-vs-kp | 99.56 | 99.41 |
| Labor | 88.33 | 83.00 |
| Lymph | 82.43 | 78.29 |
| Mushroom | 100.00 | 100.00 |
| Sonar | 80.29 | 82.21 |
| Soybean | 92.82 | **94.58** |
| Splice | 93.17 | 93.89 |
| Vehicle | 76.48 | 75.42 |
| Vote | 95.17 | 95.18 |
| Vowel | 93.94 | **96.87** |
| Breast-y | 67.88 | 68.21 |
| Breast-w | 96.42 | 96.85 |
| Heart-statlog | 81.11 | 81.85 |
| Hepatitis | 85.17 | 81.17 |
| Hypothyroid | **99.66** | 98.6 |
| Ionosphere | 93.75 | 92.6 |
| Iris | 92.67 | 93.33 |
| Primary-tumor | 40.09 | 44.53 |
| Segment | **98.57** | 97.97 |
| Sick | **99.23** | 98.49 |
| Waveform | 81.58 | 80.92 |
| Zoo | 96.18 | 94.18 |

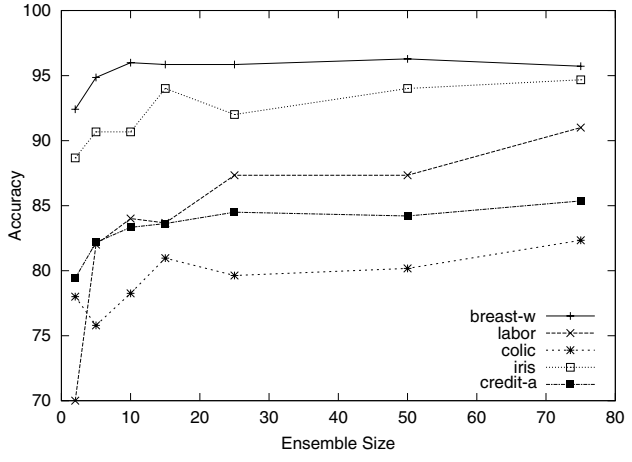


Fig. 2. DECORATE at different ensemble sizes.

## 8. Future work

Our current study has focused on building ensembles of decision trees. However, DECORATE being a meta-learner, can be applied to any learning algorithm. We plan to experiment with other base learners. In particular, we would like to apply DECORATE to neural networks and see how its diversity search compares with that of Negative Correlation Learning [21].

Recent studies have analyzed how different ensemble methods affect the contribution of *bias* and *variance* to generalization error [1,40]. We are currently analyzing the bias-variance decomposition of DECORATE ensembles to get a better understanding of their effectiveness.

ADABOOST and DECORATE both perform very well on large training sets. However, studies have shown that ADABOOST is very susceptible to noise in the training data [13,29]. In recent work, we have shown that DECORATE is more robust to noise than ADABOOST [27]. In the same work, we show that compared to Bagging and ADABOOST, DECORATE is also more resilient to missing features in the data.

Our current implementation of DECORATE attempts to increase ensemble diversity as defined by Eq. (2). However, there are several other definitions of diversity that have been explored in the literature. It is possible to generate and label artificial examples in DECORATE so as to maximize different measures of diversity. Comparing different definitions of diversity in DECORATE should provide us with more insight into which measure is the most beneficial in guiding the search for better ensembles.

The empirical success of DECORATE raises the issue of developing a sound theoretical understanding of its effectiveness. It would be particularly beneficial to prove that the DECORATE algorithm does indeed improve the bound on generalization error. Another area of future work is exploring how DECORATE relates to methods that attempt to maximize the margins on the training sample, such as ADABOOST.

In addition to improving classification accuracy in a traditional supervised setting, recent work has shown that DECORATE can be very effective for *active learning* [25]. DECORATE has also been successfully used for the task of *active feature acquisition* for classifier induction (i.e., given a feature acquisition budget, identify the instances with missing values for which acquiring complete feature information will result in the most accurate model) [26].

## 9. Conclusion

DECORATE is a simple yet effective method that uses diversity to guide ensemble construction. By manipulating artificial training examples, DECORATE is able to use a strong base learner to produce an accurate and diverse set of classifiers. Experimental results demonstrate that our approach produces highly accurate ensembles that outperform Bagging, ADABOOST and Random Forests low on the learning curve. Moreover,

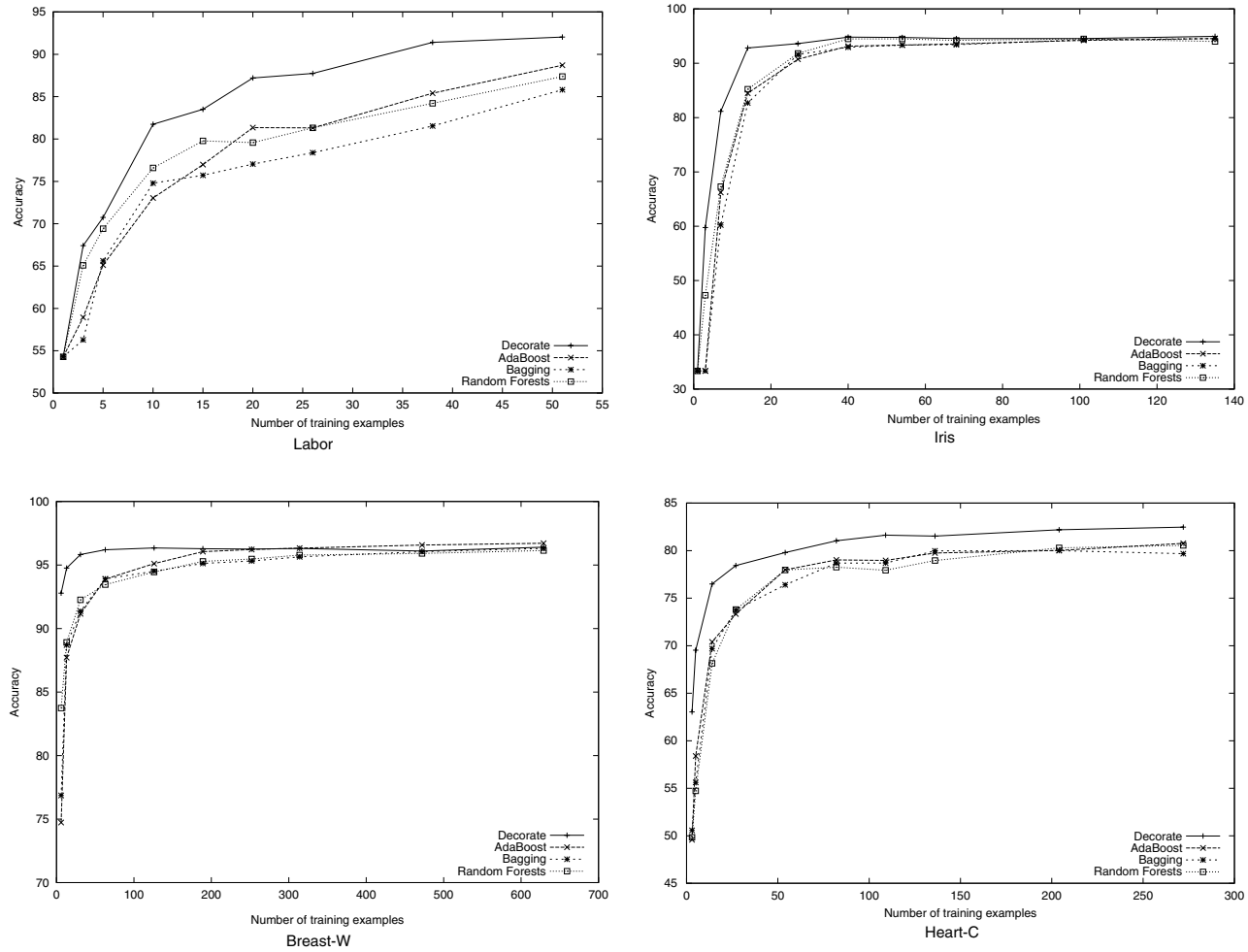


Fig. 3. Ensembles of size 100. DECORATE compared to ADABOOST, Bagging and Random Forests.

given large training sets, DECORATE outperforms Bagging and Random Forests, and is competitive with ADABOOST. In general, the idea of using artificial examples to foster diversity in the construction of ensembles seems to be a promising approach worthy of further study.

## Acknowledgements

We thank the anonymous reviewers for their helpful comments on the initial draft of this paper. This research was supported by DARPA grant HR0011-04-1-007.

## References

[1] E. Bauer, R. Kohavi, An empirical comparison of voting classification algorithms: bagging, boosting and variants, Machine Learning 36 (1–2) (1999) 105–139.
[2] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.
[3] C.L. Blake, C.J. Merz, UCI repository of machine learning databases, 1998, http://www.ics.uci.edu/~mlearn/MLRepository.html.
[4] L. Breiman, Bagging predictors, Machine Learning 24 (2) (1996) 123–140.
[5] L. Breiman, Random forests, Machine Learning 45 (1) (2001) 5–32.
[6] G. Brown, J.L. Wyatt, The use of the ambiguity decomposition in neural network ensemble learning methods, in: T. Fawcett, N. Mishra (Eds.), 20th International Conference on Machine Learning (ICML'03), Washington, DC, USA, August 2003, pp. 67–74.
[7] D. Cohn, L. Atlas, R. Ladner, Improving generalization with active learning, Machine Learning 15 (2) (1994) 201–221.
[8] M. Collins, Ranking algorithms for named-entity extraction: boosting and the voted perceptron, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL-02), 2002.
[9] M.W. Craven, J.W. Shavlik, Extracting tree-structured representations of trained networks, in: D.S. Touretzky, M.C. Mozer, M.E. Hasselmo (Eds.), Advances in Neural Information Processing Systems, vol. 8, The MIT Press, 1995, pp. 24–30.
[10] P. Cunningham, J. Carney, Diversity versus quality in classification ensembles based on feature selection, in: 11th European Conference on Machine Learning, 2000, pp. 109–116.
[11] T. Dietterich, Ensemble methods in machine learning, in: J. Kittler, F. Roli (Eds.), First International Workshop on Multiple

Classifier Systems, Lecture Notes in Computer Science, Springer-Verlag, 2000, pp. 1–15.
[12] T. Dietterich, The Handbook of Brain Theory and Neural Networks, in: Ensemble Learning, The MIT Press, 2002, pp. 405–408.
[13] T.G. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization, Machine Learning 40 (2) (2000) 139–157.
[14] P. Domingos, Knowledge acquisition from examples via multiple models, in: Proceedings of the Fourteenth International Conference on Machine Learning, Morgan Kaufmann, Nashville, TN, 1997, pp. 98–106.
[15] Y. Freund, R. Iyer, R.E. Schapire, Y. Singer, An efficient boosting algorithm for combining preferences, in: J.W. Shavlik (Ed.), Proceedings of ICML-98, 15th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, Madison, US, 1998, pp. 170–178.
[16] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, in: L. Saitta (Ed.), Proceedings of the Thirteenth International Conference on Machine Learning (ICML-96), July, Morgan Kaufmann, 1996, pp. 148–156.
[17] L.K. Hansen, P. Salamon, Neural network ensembles, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (10) (1990) 993–1001.
[18] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer Verlag, New York, 2001.
[19] A. Krogh, J. Vedelsby, Neural network ensembles, cross validation and active learning, Advances in Neural Information Processing Systems 7 (1995) 231–238.
[20] L. Kuncheva, C. Whitaker, Measures of diversity in classifier ensembles and their relationship with ensemble accuracy, Machine Learning 51 (2) (2003) 181–207.
[21] Y. Liu, X. Yao, Ensemble learning via negative correlation, Neural Networks (1999) 12.
[22] R. Maclin, D. Opitz, An empirical evaluation of bagging and boosting, in: Proceedings of the Fourteenth National Conference on Artificial Intelligence, AAAI Press, Providence, RI, 1997, pp. 546–551.
[23] R. McKay, H. Abbass, Analyzing anticorrelation in ensemble learning, in: Proceedings of 2001 Conference on Artificial Neural Networks and Expert Systems, Otago, New Zealand, 2001, pp. 22–27.
[24] P. Melville, R. Mooney, Constructing diverse classifier ensembles using artificial training examples, in: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 2003, pp. 505–510.
[25] P. Melville, R.J. Mooney, Diverse ensembles for active learning, Technical Report UT-AI-TR-04-312, University of Texas, Austin, 2004.
[26] P. Melville, M. Saar-Tsechansky, F. Provost, R. Mooney, Active feature acquisition for classifier induction, Technical Report UT-AI-TR-04-311, University of Texas, Austin, 2004.
[27] P. Melville, N. Shah, L. Mihalkova, R.J. Mooney, Experiments on ensembles with missing and noisy data, in: Proceedings of the Workshop on Multi Classifier Systems, 2004.
[28] D. Opitz, Feature selection for ensembles, in: Proceedings of 16th National Conference on Artificial Intelligence (AAAI), 1999, pp. 379–384.
[29] D. Opitz, R. Maclin, Popular ensemble methods: an empirical study, Journal of Artificial Intelligence Research 11 (1999) 169–198.
[30] D. Opitz, J. Shavlik, Actively searching for an effective neural-network ensemble, Connection Science (1996) 8.
[31] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, San Mateo, CA, 1993.
[32] J.R. Quinlan, Bagging, boosting, and C4.5, in: Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96), Portland, OR, August 1996, pp. 725–730.
[33] Y. Raviv, N. Intrator, Bootstrapping with noise: an effective regularization technique, Connection Science 8 (3–4) (1996) 356–372.
[34] B. Rosen, Ensemble learning using decorrelated neural networks, Connection Science 8 (1996) 373–384.
[35] R.E. Schapire, Theoretical views of boosting and applications, in: Proceedings of the Tenth International Conference on Algorithmic Learning Theory, 1999, pp. 13–25.
[36] R.E. Schapire, Y. Singer, Boostexter: a boosting-based system for text categorization, Machine Learning 39 (2/3) (2000) 135–168.
[37] C. Spatz, J. Johnston, Basic Statistics, third ed., Brooks/Cole Publishing Company, 1984, Chapter 9, pp. 201–202.
[38] K. Turner, J. Ghosh, Error correlation and error reduction in ensemble classifiers, Connection Science 8 (3–4) (1996) 385–403.
[39] K. Turner, N. Oza, Decimated input ensembles for improved generalization, in: International Joint Conference on Neural Networks, 1999.
[40] G. Webb, Multiboosting: a technique for combining boosting and wagging, Machine Learning 40 (2) (2000) 159–196.
[41] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann, San Francisco, 1999.
[42] G. Zenobi, P. Cunningham, Using diversity in preparing ensembles of classifiers based on different feature subsets to minimize generalization error, in: Proceedings of the European Conference on Machine Learning, 2001, pp. 576–587.