Constructing Diverse Classifier Ensembles using Artificial Training Examples

Prem Melville and Raymond J. Mooney
Department of Computer Sciences
University of Texas
1 University Station, C0500
Austin, TX 78712
melville@cs.utexas.edu, mooney@cs.utexas.edu

Abstract

Ensemble methods like bagging and boosting that combine the decisions of multiple hypotheses are some of the strongest existing machine learning methods. The diversity of the members of an ensemble is known to be an important factor in determining its generalization error. This paper presents a new method for generating ensembles 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 both the base classifier and bagging (whereas boosting can occasionally decrease accuracy), and also obtains higher accuracy than boosting early in the learning curve when training data is limited.

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 [Dietterich, 2000]. For example, *boosting* [Freund and Schapire, 1996], an ensemble method that learns a series of "weak" classifiers each one focusing on correcting the errors made by the previous one, has been found to be one of the currently best generic inductive classification methods [Hastie *et al.*, 2001].

Constructing a *diverse* committee in which each hypothesis is as different as possible (decorrelated with other members of the ensemble) while still maintaining consistency with the training data is known to be a theoretically important property of a good committee [Krogh and Vedelsby, 1995]. 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 [Opitz and Shavlik, 1996; Rosen, 1996] are fairly complex and are not general *meta-learners* like bagging [Breiman, 1996] and boosting that can be applied to any base learner to produce an effective committee [Witten and Frank, 1999].

We present a new meta-learner (DECORATE, Diverse Ensemble Creation by Oppositional Relabeling of Artificial Training Examples) that uses an existing "strong" learner (one that provides high accuracy on the training data) to build an effective diverse committee in a fairly 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.

Boosting and bagging 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 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, and DECORATE, all using J48 decision-tree induction (a Java implementation of C4.5 [Quinlan, 1993] introduced in [Witten and Frank, 1999]) as a base learner. Cross-validated learning curves support the hypothesis that "DECORATEd trees" generally result in greater classification accuracy for small training sets.

2 Ensembles and Diversity

In an ensemble, the combination of the output of several classifiers is only useful if they disagree on some inputs [Krogh and Vedelsby, 1995]. We refer to the measure of disagreement as the *diversity* of the ensemble. There have been several methods proposed to measure ensemble diversity [Kuncheva and Whitaker, 2002] — usually dependent on the measure of accuracy. For regression, where the mean squared error is commonly used to measure accuracy, variance can be used as a measure of diversity. So the diversity of the i^{th} classifier on example x can be defined as $d_i(x) = [C_i(x) - C^*(x)]^2$, where $C_i(x)$ and $C^*(x)$ are the predictions of the i^{th} classifier and the ensemble respectively. For this setting Krogh et al 11995] show that the generalization error, E", of the ensemble can be expressed as $E = E- D$, where E and D are the mean error and diversity of the ensemble respectively.

For classification problems, where the 0/1 loss function is most commonly used to measure accuracy, the diversity of the i^{th} classifier can be defined as:

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

However, in this case the above simple linear relationship does not hold between *E, E* and *D.* But there is still strong reason to believe that increasing diversity should decrease ensemble error [Zenobi and Cunningham, 20011. The underlying principle of our approach is to build ensembles of classifiers that are consistent with the training data and maximize diversity as defined in (1).

3 DECORATE: Algorithm Definition

In DECORATE (see Algorithm 1), an ensemble is generated iteratively, learning a classifier at each iteration and 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 and also on 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_{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. If adding this new classifier to the current ensemble increases the ensemble training error, then we reject this classifier, else we add it to the current ensemble. This process is repeated until we reach the desired committee size or exceed the maximum number of iterations.

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 $Pc_i,y(x)$ is the 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:

$$P_y(x) = \frac{\sum_{C_i \in C^*} P_{C_i,y}(x)}{|C^*|}$$

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

3.1 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.

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, replacing zero probabilities with a small non-zero value. Labels are then selected, such that the probability of selection is inversely proportional to the current ensemble's predictions.

Algorithm 1 The DECORATE algorithm

Input:
$BaseLearn$ - base learning algorithm
T - set of m training examples $< (x_1, y_1), ..., (x_m, y_m) >$ with labels $y_j \in Y$
C_{size} - desired ensemble size
I_{max} - maximum number of iterations to build an ensemble
R_{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 $i < C_{size}$ and $trials < I_{max}$
7. Generate $R_{size} \times |T|$ training examples, R, based on distribution of training data
8. Label examples in R with probability of class labels inversely proportional to C^*'s predictions
9. $T = T \bigcup R$
10. $C' = BaseLearn(T)$
11. $C^* = C^* \bigcup \{C'\}$
12. $T = T - R$, remove the artificial data
13. Compute training error, ϵ', of C^* as in step 5
14. If $\epsilon' \leq \epsilon$
15. $i = i + 1$
16. $\epsilon = \epsilon'$
17. otherwise,
18. $C^* = C^* - \{C'\}$
19. $trials = trials + 1$

4 Experimental Evaluation

4.1 Methodology

To evaluate the performance of DECORATE we ran experiments on 15 representative data sets from the UCI repository [Blake and Merz, 19981 used in similar studies [Webb, 2000;

Quinlan, 1996]. We compared the performance of DECORATE to that of Adaboost, Bagging and J48, using J48 as the base learner for the ensemble methods and using the Weka implementations of these methods [Witten and Frank, 1999]. For the ensemble methods, we set the ensemble size to 15. Note that in the case of DECORATE, we only specify a maximum ensemble size, the algorithm terminates if the number of iterations exceeds the maximum limit 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_{size}, and found that the results do not vary much for the range 0.5 to 1. However, R_{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_{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 10-fold cross-validations. 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.

To compare two learning algorithms across all domains we employ the statistics used in [Webb, 2000], 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 *EA* and *EB* 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 so as to capture the degree to which algorithms out-perform each other in win or loss outcomes.

4.2 Results

Our results are summarized in Tables 1-3. 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.

DECORATE has more *significant* wins to losses over Bagging for all points along the learning curve (see Table 2). DECORATE also outperforms Bagging on the geometric mean 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.

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 data DECORATE produces higher accuracies on 6 out of 15 data sets.

It has been observed in previous studies [Webb, 2000; Bauer and Kohavi, 1999] 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 1.

On many data sets, DECORATE achieves the same or higher accuracy as Bagging and AdaBoost with many fewer training examples. Figure 1 show learning curves that clearly demonstrate this point. Hence, in domains where little data is available or acquiring labels is expensive, DECORATE has an advantage over other ensemble methods.

We performed additional experiments to analyze the role that diversity plays in error reduction. We ran DECORATE at 10 different settings of R_{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. Diversity of an ensemble is computed as the mean diversity of the ensemble members (as given by Eq. 1). 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 [Cunningham and Carney, 2000]). We found that the correlation coefficient between diversity and ensemble error reduction is 0.6225 ($p^1 \ll 10^{-50}$), which is fairly strong. 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 0.1552 ($p \ll 10^{-50}$). We note that even though this correlation 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.

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. Due to lack of space, we do not include the results for all 15 datasets, but present five representative datasets (see Figure 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 to 25.

[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.

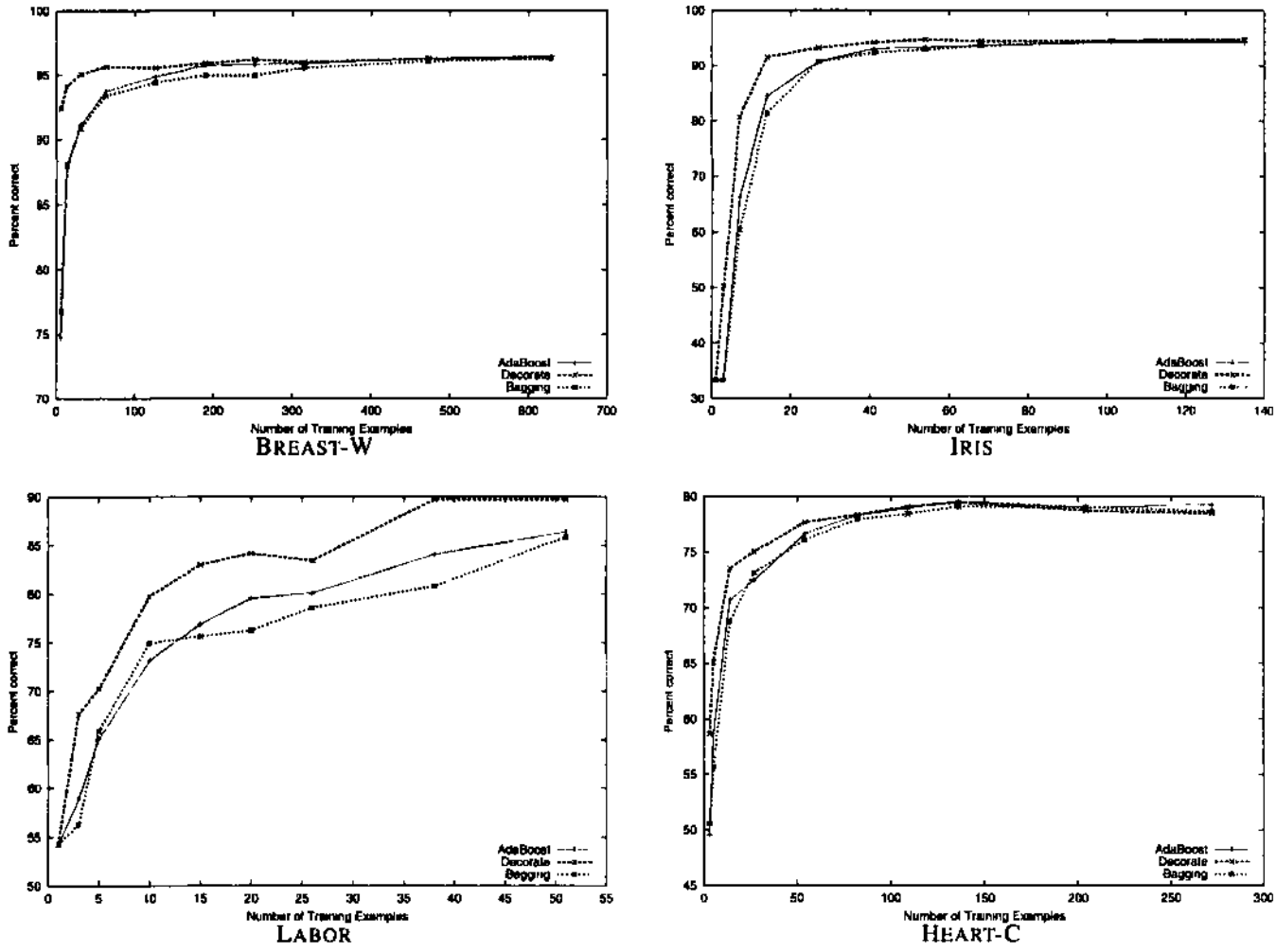

Figure 1: DECORATE compared to AdaBoost and Bagging

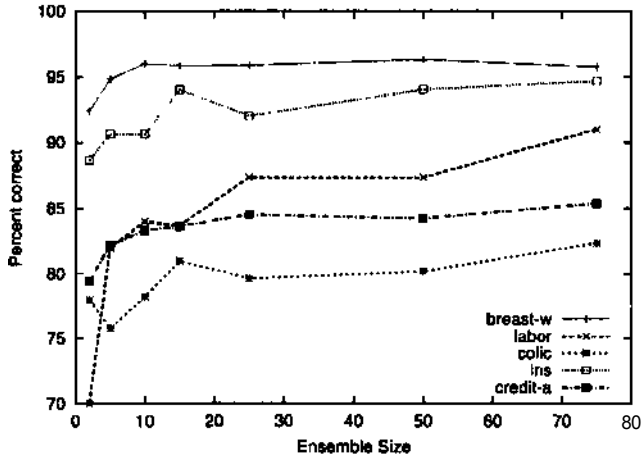

Figure 2: DECORATE at different ensemble sizes

5 Related Work

There have been some other attempts at building ensembles that focus on the issue of diversity. Liu et al [1999] and Rosen [1996] simultaneously train neural networks in an ensemble using a correlation penalty term in their error functions. Opitz and Shavlik [1996] 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. Zenobi et al [2001] build ensembles based on different feature subsets; where 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 lead to a "substantial" deterioration of the other; where "substantial" is defined by a pre-sct threshold.

In all these approaches, ensembles are built attempting to simultaneously optimize the accuracy and diversity of individual ensemble members. However, in DECORATE, our goal is to minimize *ensemble error* by increasing diversity. At no point does the training accuracy of the ensemble go below

Table 1: DECORATE vs J48

Dataset	1%	2%	5%	10%	20%	30%	40%	50%	75%	100%
anneal	**75.29**/72.49	**78.14**/75.31	**85.24**/82.08	**92.26**/89.28	**96.48**/95.57	**97.36**/96.47	**97.73**/97.3	**98.16**/97.93	98.39/98.35	98.71/98.55
audio	16.66/16.66	23.73/23.07	41.72/41.17	**55.42**/51.67	**64.09**/60.59	**67.62**/64.84	**70.46**/68.11	**72.82**/70.77	**77.8**/75.15	**82.1**/77.22
autos	24.33/24.33	29.6/29.01	**36.73**/34.37	42.89/41.22	52.2/50.53	**59.86**/53.92	**64.77**/59.68	**68.6**/65.24	**78**/73.15	**83.64**/81.72
breast-w	**92.38**/74.73	**94.12**/87.34	**95.06**/89.42	**95.64**/92.21	**95.55**/93.09	**95.91**/93.36	**96.2**/93.85	**96.01**/94.24	**96.28**/94.65	**96.31**/95.01
credit-a	71.78/69.54	74.83/**77.46**	80.61/81.57	83.09/82.35	84.38/84.29	84.68/84.59	**85.22**/84.41	**85.57**/84.78	85.61/85.43	85.93/85.57
Glass	31.69/31.69	**35.86**/32.96	**44.5**/38.34	**55.4**/46.62	**61.77**/54.16	**66.01**/60.63	**68.07**/61.38	**68.85**/63.69	**72.73**/67.53	**72.77**/67.77
heart-c	**58.66**/49.57	**65.11**/58.03	**73.55**/67.71	**75.05**/70.15	**77.66**/73.44	**78.34**/74.61	**79.09**/74.78	**79.46**/75.62	**78.74**/76.7	**78.48**/77.17
hepatitis	52.33/52.33	**71.95**/65.93	**76.59**/72.75	78.85/78.25	80.28/78.61	**81.14**/78.63	**81.53**/79.35	**81.68**/79.57	**82.37**/79.04	**82.43**/79.22
colic	**59.85**/52.85	**68.19**/65.31	74.91/74.37	78.45/79.94	81.81/82.71	82.47/83.41	82.74/83.55	83.5/**84.66**	83.93/**85.18**	85.24/85.16
iris	33.33/33.33	**50.87**/33.33	**80.67**/59.33	**91.27**/84.33	**93.07**/91.33	**94.4**/92.73	**95.07**/93	94.07/93.33	94.67/94.07	94.93/94.73
labor	54.27/54.27	54.27/54.27	**67.7**/58.93	**71.47**/64.77	**78.6**/70.07	**81.67**/73.7	**85.67**/75.17	**84.2**/75.8	**87.53**/77.4	**89.5**/78.8
lymph	48.39/48.39	**53.49**/46.64	**65.73**/60.39	**72.79**/68.21	**74.57**/70.79	**78.84**/73.58	**78.37**/74.53	**78.31**/73.34	**78.06**/75.63	**78.74**/76.06
segment	**67.94**/52.43	**80.75**/73.26	**89.52**/85.41	**92.87**/89.34	**94.99**/92.22	**95.82**/93.37	**96.54**/94.34	**96.93**/94.77	**97.56**/95.94	**98.02**/96.79
soybean	**19.37**/13.69	**32.12**/22.32	**55.55**/42.94	**73.51**/59.04	**84.63**/74.49	**88.52**/81.59	**90.37**/84.78	**91.35**/86.89	**92.85**/89.44	**93.81**/91.76
splice	**63.48**/59.92	67.56/**68.69**	77.34/77.49	82.62/82.58	**88.2**/87.98	90.46/90.44	91.82/91.77	92.5/92.4	93.41/93.47	93.92/**94.03**
Win/Draw/Loss	15/0/0	13/0/2	13/0/2	14/0/1	14/0/1	14/0/1	14/0/1	14/0/1	13/0/2	14/0/1
Sig. W/D/L	7/8/0	10/3/2	11/4/0	10/5/0	11/4/0	12/3/0	13/2/0	12/2/1	10/4/1	10/4/1
GM error ratio	0.858	0.8649	0.8116	0.8098	0.8269	0.8103	0.7983	0.8305	0.8317	0.8293

Table 2: DECORATE vs Bagging

Dataset	1%	2%	5%	10%	20%	30%	40%	50%	75%	100%
anneal	75.29/74.57	**78.14**/76.42	**85.24**/82.88	**92.26**/89.87	**96.48**/95.67	**97.36**/96.89	**97.73**/97.34	**98.16**/97.78	98.39/98.53	98.71/98.83
audio	**16.66**/12.98	23.73/23.68	**41.72**/38.55	**55.42**/51.34	**64.09**/61.76	67.62/66.9	70.46/70.29	72.82/73.07	77.8/77.32	**82.1**/80.71
autos	**24.33**/22.16	29.6/28	36.73/35.88	42.89/44.65	52.2/54.32	59.86/59.67	64.77/65.6	68.6/69.88	78/77.97	83.64/83.12
breast-w	**92.38**/76.74	**94.12**/88.07	**95.06**/90.88	**95.64**/93.41	**95.55**/94.42	**95.91**/94.95	**96.2**/94.95	**96.01**/95.55	96.28/96.07	96.31/96.3
credit-a	71.78/69.54	74.83/**77.99**	80.61/**82.58**	83.09/83.9	84.38/**85.13**	84.68/**85.78**	85.22/85.59	85.57/85.64	85.61/**86.12**	85.93/85.96
Glass	**31.69**/24.85	**35.86**/31.47	**44.5**/40.87	**55.4**/49.6	**61.77**/58.9	**66.01**/64.35	68.07/66.3	68.85/68.44	72.73/72	72.77/**74.67**
heart-c	**58.66**/50.56	**65.11**/55.67	**73.55**/68.77	**75.05**/73.17	**77.66**/76.12	78.34/77.9	79.09/78.44	79.46/79.11	78.74/79.05	78.48/78.68
hepatitis	52.33/52.33	**72.14**/63.18	76.8/75.2	79.48/78.64	80.7/80.42	81.81/81.07	81.65/81.22	**83.19**/81.06	**82.99**/80.87	82.62/81.34
colic	**58.37**/53.14	**66.58**/63.83	75.85/76.44	79.54/80.06	81.33/**83.04**	82.47/**83.58**	83.02/**83.98**	83.1/**84.47**	84.02/**85.4**	84.69/85.34
iris	33.33/33.33	**50.27**/33.33	**80.67**/60.47	**91.53**/81.4	93.2/90.67	**94.2**/92.33	**94.73**/92.87	**94.4**/93.6	94.53/94.47	94.67/94.73
labor	54.27/54.27	54.27/54.27	**67.63**/56.27	**70.23**/65.9	**79.77**/74.97	**83**/75.67	**84.17**/76.27	**83.43**/78.6	**89.73**/80.83	**89.73**/85.87
lymph	48.39/48.39	**53.62**/47.11	**65.06**/60.12	71.2/69.68	**76.74**/73.6	**78.84**/76.58	78.17/77.68	**78.99**/76.98	**79.14**/76.8	79.08/77.97
segment	**67.03**/55.88	**81.16**/76.36	**89.61**/87.42	**92.83**/91.01	**94.88**/93.4	**95.94**/94.65	**96.47**/95.26	**96.93**/95.82	**97.58**/96.78	**98.03**/97.41
soybean	**19.51**/14.56	**32.4**/24.58	**55.36**/47.46	**73.06**/65.45	**85.14**/79.29	**88.27**/85.05	**90.22**/87.89	**91.4**/89.22	**92.75**/91.56	**93.89**/92.71
splice	62.77/62.52	67.8/**72.36**	77.37/**80.5**	82.55/**85.44**	88.24/**89.5**	90.47/**91.44**	91.84/**92.4**	92.41/**93.07**	93.44/**94.06**	93.92/**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

that of the base classifier; however, this is a possibility with previous methods. Furthermore, none of the previous studies compared their methods with the standard ensemble approaches such as Boosting and Bagging (fOpitz and Shavlik, 1996] compares with Bagging, but not Boosting).

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 boosting. Therefore, DECORATE may be a preferable ensemble meta-learner for strong learners.

To our knowledge, the only other ensemble approach to utilize artificial training data is the active learning method introduced in ICohn *et al,* 1994]. The goal of the committee here is to select good new training examples rather than to improve accuracy using the existing training data. Also, 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.

6 Future Work and Conclusion

In our current approach, we are encouraging diversity using artificial training examples. However, in many domains, a large amount of unlabeled data is already available. We could exploit these unlabeled examples and label them as diversity data. This would allow DECORATE to act as a form of *semi-supervised learning* that exploits both labeled and unlabeled data [Nigam et al., 2000].

Our current study has used J48 as a base learner; however, we would expect similarly good results with other base learners. Decision-tree induction has been the most commonly used base learner in other ensemble studies, but there has been some work using neural networks and naive Bayes [Bauer and Kohavi, 1999; Opitz and Maclin, 1999]. Experiments on "DECORATing" other learners is another area for future work.

By manipulating artificial training examples, DECORATE is able to use a strong base learner to produce an effective, diverse ensemble. Experimental results demonstrate that the approach is particularly effective at producing highly accurate ensembles when training data is limited, outperforming both bagging and boosting low on the learning curve. The empirical success of DECORATE raises the issue of developing a sound theoretical understanding of its effectiveness. In gen-

Table 3: DECORATE vs AdaBoost

Dataset	1%	2%	5%	10%	20%	30%	40%	50%	75%	100%
anneal	75.29/73.02	78.14/77.12	85.24/87.51	92.26/94.16	96.48/97.13	97.36/97.95	97.73/98.54	98.16/98.8	98.39/99.23	98.71/99.68
audio	16.66/16.66	23.73/23.41	41.72/40.24	55.42/52.7	64.09/64.15	67.62/68.91	70.46/73.07	72.82/75.92	77.8/81.74	82.1/84.52
autos	24.33/24.33	29.6/29.71	36.73/34.2	42.89/43.28	52.2/56.13	59.86/62.2	64.77/69.14	68.6/72.03	78/80.28	83.64/85.28
breast-w	92.38/74.73	94.12/87.84	95.06/91.15	95.64/93.75	95.55/94.85	95.91/95.72	96.2/95.84	96.01/95.87	96.28/96.3	96.31/96.47
credit-a	71.78/68.8	74.83/75.3	80.61/79 68	83.09/81.14	84.38/83.04	84.68/84.22	85.22/84.13	85.57/84.58	85.61/84.93	85.93/85.42
Glass	31.69/31.69	35.86/32.93	44.5/40.71	55.4/49.78	61.77/58.03	66.01/64.33	68.07/66.93	68.85/68.69	72.73/74.69	72.77/76.06
heart-c	58.66/49.57	65.11/58.65	73.55/70.71	75.05/72.5	77.66/76.65	78.34/78.26	79.09/78.96	79.46/79.55	78.74/79.06	78.48/79.22
hepatitis	52.33/52.33	72.14/65.93	76.8/73.01	79.48/76.95	80.7/79.44	81.81/79.22	81.65/81.27	83.19/82.63	82.99/83.24	82.62/82.71
colic	58.37/52.85	66.58/67.18	75.85/72.85	79.54/77.17	81.33/79.36	82.47/79.24	83.02/79.51	83.1/80.22	84.02/80.59	84.69/81.93
iris	33.33/33.33	50.27/33.33	80.67/66.2	91.53/84.53	93.2/90.73	94.2/93	94.73/93.33	94.4/93.53	94.53/94.2	94.67/94.2
labor	54.27/54.27	54.27/54.27	67.63/58.93	70.23/65.1	79.77/73.2	83/76.9	84.17/79.57	83.43/80.1	89.73/84.07	89.73/86.37
lymph	48.39/48.39	53.62/46.64	65.06/60.54	71.2/69.57	76.74/74.16	78.84/78.62	78.17/80.35	78.99/79.88	79.14/80.96	79.08/81.75
segment	67.03/60.22	81.16/77.38	89.61/88.5	92.83/92.71	94.88/95.01	95.94/96.03	96.47/96.9	96.93/97.23	97.58/98	98.03/98.34
soybean	19.51/14.26	32.4/23.36	55.36/49.37	73.06/69.49	85.14/85.01	88.27/88.37	90.22/90.04	91.4/90.89	92.75/92.57	93.89/92.88
splice	62.77/65.11	67.8/73.9	77.37/82.22	82.55/86.13	88.24/88.27	90.47/89.82	91.84/90.8	92.41/90.78	93.44/92.63	93.92/93.59
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

eral, the idea of using artificial or unlabeled examples to aid the construction of effective ensembles seems to be a promising approach worthy of further study.

Acknowledgments

This work was supported by DARPA EELD Grant F30602-01-2-0571.

References

[Bauer and Kohavi, 1999] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting and variants. *Machine Learning,* 36, 1999.

iBlake and Merz, 1998] C. L. Blake and C. J. Merz. UCI repository of machine learning databases. http://www.ics.uci.edurmlearn/MLRepository.html,

[Breiman, 1996] Leo Breiman. Bagging predictors. *Machine Learning,* 24(2): 123-140,1996.

ICohn *etal,* 1994] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning,* 15(2):201-221,1994.

[Cunningham and Carney, 2000] P. Cunningham and J. Carney. Diversity versus quality in classification ensembles based on feature selection. In *11th European Conference on Machine Learning,* pages 109-116,2000.

[Dietterich, 2000] T. Dietterich. Ensemble methods in machine learning. In J. Kittler and F. Roli, editors, *First International* ***Workshop on Multiple Classifier Systems, Lecture*** *Notes in Computer Science,* pages 1-15. Springer-Verlag, 2000.

[Freund and Schapire, 1996] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In ***Proceedings of the 13th International Conference on Machine*** *Learning,* July 1996.

[Hastie*al,* 2001] Trevor Hastie, Robert Tibshirani, and **Jerome Friedman.** ***The Elements of Statistical Learning.*** Springer Verlag, New York, August 2001.

iKrogh and Vedelsby, 1995] A. Krogh and J. Vedelsby. Neural network ensembles, cross validation and active learning. **In *Advances in Neural Information Processing Systems*** 7, 1995.

iKuncheva and Whitaker, 2002] L. Kuncheva and C. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. ***submitted,*** **2002.**

[Liu and Yao, 1999] Y. Liu and X. Yao. Ensemble learning via negative correlation. *Neural Networks,* 12, 1999.

[Nigam *et al,* 2000] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning,* 39:103-134,2000.

[Opitz and Maclin, 1999J David Opitz and Richard Maclin. Popular ensemble methods: An study journal ***of Artificial Intelligence Research,*** **11.169-198,1999.**

[Opitz and Shavlik, 19961 D. Opitz and J. Shavlik. Actively searching for an effective neural-network ensemble. *Connection* ***Science,*** **8, 1996.**

[Quinlan, 1993] J. Ross Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo,CA, 1993.

[Quinlan, 1996] J. Ross Quinlan. Bagging, boosting, and **C4.5. In *Proceedings of the 13th National Conference on Artificial Intelligence,*** **August 1996.**

[Rosen, 1996] B. Rosen. Ensemble learning using decorrelated neural networks. *Connection Science,* 8, 1996.

[Webb, 2000] G. Webb. Multiboosting: A technique for combining boosting and wagging. *Machine Learning,* 40, 2000.

[Witten and Frank, 1999] Ian H. Witten and Eibe Frank. ***Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.*** **Morgan Kaufmann,** San Francisco, 1999.

[Zenobi and Cunningham, 2001] G. Zenobi and 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.**