Multistrategy Ensemble Learning: Reducing Error by Combining Ensemble Learning Techniques

Geoffrey I. Webb, *Member*, *IEEE*, and Zijian Zheng

Abstract—Ensemble learning strategies, especially Boosting and Bagging decision trees, have demonstrated impressive capacities to improve the prediction accuracy of base learning algorithms. Further gains have been demonstrated by strategies that combine simple ensemble formation approaches. In this paper, we investigate the hypothesis that the improvement in accuracy of multistrategy approaches to ensemble learning is due to an increase in the diversity of ensemble members that are formed. In addition, guided by this hypothesis, we develop three new multistrategy ensemble learning techniques. Experimental results in a wide variety of natural domains suggest that these multistrategy ensemble learning techniques are, on average, more accurate than their component ensemble learning techniques.

Index Terms—Boosting, bagging, ensemble learning, committee learning, multiboosting, bias, variance, ensemble diversity.

1 Introduction

Classification ensemble learning techniques have demonstrated powerful capacities to improve upon the classification accuracy of a base learning algorithm. Common to these approaches is the repeated application of the base learning algorithm to a sample derived from the available training data. Each application of the algorithm results in the generation of a new classifier which is added to the ensemble. To classify a new case, each member of the ensemble classifies the case independently of the others and then the resulting votes are aggregated to derive a single final classification.

It has been observed that an important prerequisite for classification ensemble learning to reduce test error is that it generate a diversity of ensemble members [7], [8], [14], [15]. If all ensemble members agree in all of their classifications, then the aggregated classification under any reasonable aggregation scheme will be identical to that of any individual ensemble member. Indeed, for ensembles of numeric predictors for which aggregation of predictions is by weighted numeric mean, it has been proven that increasing diversity between ensemble members without increasing their individual test error necessarily results in a decrease in ensemble test error [13]. This result does not extend directly to classification learning, however, as aggregation of classification predictions is not usually performed by weighted numeric mean. Nonetheless, using majority voting between ensemble members in a two-class domain, if diversity in predictions is maximized (as measured by the variance of the predictions) while maintaining a set test error rate for each individual ensemble member, so long as $e < 0.5$ and $t \geq \frac{1}{1-2e}$ rounded up to the next odd integer (where e is the test error for individual ensemble members and t is the ensemble size), the test error rate of the ensemble will be zero. This is because variance will be maximized when the proportion e of ensemble members make an error on each test case to be classified. If e is less than 0.5, then this will ensure that the majority vote favors the correct class for every case. The constraint on t is required to ensure that the rounding up of e required by the granularity of ensemble votes still results in a value less than 0.5 for each case to be classified. Turning from this theoretical result, in general, when the test error rate of individual classifiers is less than 0.5, increasing diversity in classifications between classifiers will tend to decrease test error as it will tend to dilute concentrations of errors to less than 0.5 of the votes for any given test case to be classified, hence tending to result in correct classification.

However, this insight is of less practical value for the generation of classification ensemble learning techniques than might at first be thought. This is because methods for increasing diversity within an ensemble usually come at a cost of also increasing the expected test error of the individual ensemble members. Without knowing the magnitude of the increase in the test error of the individual ensemble members, it is not possible to realistically assess the likely outcome of a particular trade off between diversity and individual error. Assessing the likely increase

- *G.I. Webb is with the School of Computer Science and Software Engineering, Monash University, Clayton, Victoria, 3800, Australia. E-mail: webb@infotech.monash.edu.*
- *Z. Zheng is with Microsoft Corporation, One Microsoft Way, Redmond, WA 98052-6399. E-mail: zijian@acm.org.*

Manuscript received 21 Feb. 2002; revised 19 Feb. 2003; accepted 5 May 2003. For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number 115933.

in individual error is not practical, however, as error estimation on the training data is likely to produce highly optimistic estimates. Nonetheless, the spectacular success of ensemble techniques demonstrates that they manage this trade off successfully in practice.

With these issues in mind, Webb [20] hypothesized that it would be advantageous to combine ensemble learning techniques that have the capacity to effectively manage this trade off because doing so will lead to further increases in internal diversity without undue increases in individual error and this can be expected to result in improved classification accuracy. These hypotheses led to the development of MultiBoosting [20], a technique that combines AdaBoost [10] and a variant of Bagging [4] called *Wagging* [2].

MultiBoosting has been demonstrated to attain most of Boosting's superior bias reduction together with most of Wagging's superior variance reduction. However, which mechanisms are responsible for this outcome remains an open question. This paper investigates the link in multistrategy ensemble learning between test error reduction and the generation of diversity in ensemble membership. Further, Boosting and Bagging/Wagging are not the only approaches to classification ensemble learning. This paper also explores the effect of increasing the diversity in ensemble membership by integrating the formation of ensembles by stochastic perturbation [9], [1], [21] with Boosting and Wagging.

2 Explanations for the Effectiveness of Ensembling

The spectacular success of ensemble learning has led to a number of investigations into the underlying mechanisms that support their powerful error reduction capabilities.

Breiman [5] argues that bagging can be viewed as classifying by application of an estimate of the central tendency for the base learner. This may serve to explain why bagging reduces variance. However, it is yet to be explained why such a reduction in variance should not be accompanied by a corresponding increase in error due to bias. Nonetheless, several studies have shown bagging to decrease variance without unduly affecting bias [7], [18], [2], [20].

A contrasting account of the performance of AdaBoost is provided by Friedman et al. [11]. They provide an account of AdaBoost in terms of additive logistic regression. They assert that boosting by reweighting "appears to be a purely 'bias' reduction procedure, intended to increase the flexibility of stable (highly biased) weak learners." Despite this account, a number of empirical studies have demonstrated AdaBoost to reduce both bias and variance [7], [18], [2], [20].

Two sets of studies with artificial data have shown AdaBoost to outperform bagging both in terms of bias and variance reduction [7], [18]. However, experiments with "natural" data sets seem to indicate that, in general, while AdaBoost is more effective at reducing bias than is bagging, bagging is the more effective at reducing variance [2], [20].

Freund and Schapire [10] prove that AdaBoost reduces error on the training data. However, they also note that this need not reduce error outside the training data. They suggest that structural risk minimization [19] might explain off-training set error reduction. However, subsequent empirical evidence has not supported this supposition [18].

Schapire et al. [18] attribute AdaBoost's ability to reduce off-training set error to its boosting the margins of the ensemble's weighted classifications. However, as evidence against this account, Breiman [6] has presented algorithms that are more effective than AdaBoost at increasing margins, but less effective at reducing test error.

Breiman [7] ascribes AdaBoost's error reduction to *adaptive resampling*. This is the construction of an ensemble by repeated sampling from a training set where the probability of a training case being included in a subsequent sample is increased if it has been misclassified by ensemble members learned from previous samples. Some support for this argument is provided by the success of an alternative adaptive resampling algorithm, arc-x4. However, while AdaBoost has been demonstrated to be equally effective at reducing error using either reweighting or resampling, arc-x4 has been shown to be much less effective using reweighting than using resampling [2]. This could be taken to indicate that AdaBoost does more than just adaptive resampling.

As can be seen, while investigation into the success of ensemble learning techniques has been extensive, no single account has received undisputed widespread support. In this context, this paper investigates the role of diversity between ensemble members in the effectiveness of ensemble learning.

3 Ensemble Learning Techniques

This section describes the ensemble learning techniques utilized in this work. All of the techniques take a base learning algorithm and a set of training data and then repeatedly apply the algorithm or a variant thereof to a sample from the data, producing a set of classifiers. These classifiers then vote to reach an ensemble classification.

Bagging applies the base algorithm to bootstrap samples from the training data. A bootstrap sample from n cases is formed by randomly selecting n cases with replacement. Wagging is similar to Bagging except that all training cases are retained in each training set, but each case is stochastically assigned a weight. In the current research, we follow Webb [20] in assigning weights from the exponential distribution. This is motivated by the observation that Bagging can be considered to be Wagging with allocation of weights from the Poisson distribution, a discrete distribution that results in discrete valued weights (each case is represented in the sample a discrete number of times). The exponential distribution is the continuous valued counterpart to the Poisson distribution and, hence, the use of the exponential distribution provides a natural extension of Bagging to utilization of fractional weights. Individual random instance weights (approximately) conforming to the exponential distribution are calculated by the following formula:

$$rand\_weight() = -\log\left(\frac{rand(1\ldots 999)}{1000}\right), \qquad (1)$$

where $rand(x\ldots y)$ returns a random integer value between x and y inclusive.

The resulting algorithm is presented as Algorithm 1. In general, Wagging is slightly less effective than Bagging at test error reduction, perhaps because the inclusion of every case in every training set tends to lead to lower variation between the resulting ensemble members [20]. Nonetheless, we utilize Wagging rather than Bagging in the current research as it interacts better with other ensemble learning algorithms, possibly because it includes all training cases, allowing the other algorithm access to all cases on every run.

Algorithm 1 The WAG algorithm

```
WAG($S$, $BaseLearn$, $t$)
inputs: $S$, a sequence of $m$ examples $\langle(x_1,y_1),\ldots,(x_m,y_m)\rangle$ with labels
            $y_i \in Y = \{1,...,k\}$.
        base learning algorithm $BaseLearn$.
        integer $t$ specifying the number of iterations.

1: for $i \leftarrow 1$ to $t$ do
2:     $S' \leftarrow S$ with random weights drawn from the exponential distribution.
3:     $C_i \leftarrow BaseLearn(S')$.
4: end for
5: output the final classifier: $C^*(x) = \underset{y\in Y}{argmax} \sum_{i=1}^{t} 1(C_i(x) = y)$.
```

Boosting is another approach to ensemble learning. The first ensemble member is formed by applying the base learning algorithm to the entire training set. Subsequent ensemble members are formed by applying the base algorithm to the training set but with cases reweighted to place higher weight on cases that are misclassified by existing ensemble members. The votes of ensemble members are weighted by a function that lowers the vote of a classifier that has lower accuracy on the weighted training set from which it was learned. We utilize a minor variant on the Bauer and Kohavi [2] variant of Freund and Schapire's [10] AdaBoost algorithm. This is presented as Algorithm 2.

Algorithm 2 The AdaBoost algorithm

```
AdaBoost($S$, $BaseLearn$, $t$)
inputs: $S$, a sequence of $m$ examples $\langle(x_1,y_1),\ldots,(x_m,y_m)\rangle$ with labels
            $y_i \in Y = \{1,...,k\}$.
        base learning algorithm $BaseLearn$.
        integer $t$ specifying the number of iterations.

1: $S' \leftarrow S$ with all instance weights set to 1.
2: for $i = 1$ to $t$ do
3:     $C_i \leftarrow BaseLearn(S')$.
4:     $\epsilon_i \leftarrow \frac{\sum_{x_j\in S':C_i(x_j)\neq y_j} weight(x_j)}{m}$ {the weighted error on the training set}
5:     if $\epsilon_i > 0.5$ then
6:         reset $S'$ to random weights from the exponential distribution.
7:         standardize $S'$ to sum to $m$.
8:         goto step 3.
9:     end if
10:    if $\epsilon_i = 0$ then
11:        set $\beta_i$ to $10^{-10}$
12:        reset $S'$ to random weights from the exponential distribution.
13:        standardize $S'$ to sum to $m$.
14:    else
15:        $\beta_i \leftarrow \frac{\epsilon_i}{(1-\epsilon_i)}$.
16:        for all $x_j \in S'$ do
17:            divide $weight(x_j)$ by $2\epsilon_i$ if $C_i(x_j) \neq y_j$ and $2(1-\epsilon_i)$ otherwise.
18:            if $weight(x_j) < 10^{-8}$, set $weight(x_j)$ to $10^{-8}$.
19:        end for
20:    end if
21: end for
22: output the final classifier: $C^*(x) = \underset{y\in Y}{argmax} \sum_{i:C_i(x)=y} log\frac{1}{\beta_i}$.
```

Bauer and Kohavi's [2] variant

- Uses a one step weight update process that is less subject to numeric underflow than the original two step process (Step 17).
- Prevents numeric underflow (Step 18).
- Continues producing more ensemble members beyond the point where $\epsilon \geq 0.5$ (Step 5). This measure is claimed to improve prediction accuracy [5], [2].

We further modify this approach by utilizing the exponential distribution for reweighting cases at Steps 6 and 12 and continuing to produce more ensemble members beyond a point where training error falls to zero (Step 10). These two measures are included for the sake of consistency between the various learning algorithms. All use the exponential distribution for random reweighting and all always produce an ensemble of size t. Note that this version of the algorithm may fail to terminate, entering an infinite loop through Steps 3 to 8. This did not occur in our experiments.

Stochastic Attribute Selection Committee Learning differs from the above techniques in that instead of perturbing the training set, it performs stochastic perturbations to the base learning algorithm on successive applications of the learner to a training set. A number of variations on this general approach have been explored [9], [1]. While our specific technique (described in more detail by Zheng and Webb [21]) differs in minor details from these previous approaches, we have no reason to believe that, in general, its performance would differ substantially from the alternatives. Our implementation, that uses C4.5 [16] Release 8 as the base learning algorithm, operates as follows: When learning a decision tree, C4.5 applies an information measure to each potential test selecting the test with the highest value of a criterion based on gain ratio [17]. C4.5SAS modifies this behavior by introducing a stochastic element to the selection process, allowing tests with lower values on the selection criterion to occasionally be selected. This is achieved by selecting a subset of the available attributes, with each available attribute having probability of 0.33 of inclusion. If there is an acceptable test on the attributes included in the subset, then the one that rates highest on the selection criterion is selected. If there is no such test among the selected attributes, the best test among all attributes is selected unless there is no acceptable test in which case a leaf is formed.

Our implementation of Stochastic Attribute Selection Committees, SASC, repeatedly applies C4.5SAS to the training data to create an ensemble of classifiers. This process is presented as Algorithm 3.

Algorithm 3 The SASC learning algorithm

```
SASC($Att$, $D$, $P$, $t$)
inputs: $Att$: a set of attributes,
        $D$: a training set represented using $Att$ and classes,
        $P$: the probability with which an attribute should be included in
             the set of attributes available at a node,
        $t$: the number of trials.

1: for $i \leftarrow 1$ to $t$ do
2:   $C_i \leftarrow$ C4.5SAS($Att$, $D$, $P$)
3: end for
4: output the final classifier: $C^*(x) = \underset{y \in Y}{argmax} \sum_{i=1}^{t} 1(C_i(x) = y)$.
```

4 Multistrategy Ensemble Learning Algorithms

We combine multiple approaches to ensemble learning motivated by the hypothesis that doing so will increase diversity between ensemble members, albeit at a cost of a small increase in individual error. Webb [20] hypothesized that this process would trade off diversity against individual error so as to decrease the resulting ensemble's test error. In the current work, we seek to evaluate this hypothesis and explore the role of ensemble member diversity in ensemble learning.

MultiBoosting has established that the combination of Boosting and Wagging can reduce test error. However, this does not answer the questions of whether this reduction can be attributed to an increase in the diversity of ensemble members or whether combinations of other forms of ensemble learners may also be productive. We address the latter question by exploring the space of combinations of Boosting, Wagging, and Stochastic Attribute Selection Committees. For our experimental work, we utilize the well-known C4.5 Release 8 [17] as the base learning algorithm.

To combine SASC with another method, we replace C4.5 with C4.5SAS as the base learning algorithm within the other method. To combine Wagging with Boosting we follow Webb's [20] MultiBoosting approach, Wagging subensembles formed by Boosting. The MB algorithm is presented in Algorithm 4. Note that, for consistency with the other approaches to combining base learning algorithms, we have modified MB to utilize stochastic weights for the first subensemble (Step 1) rather than initializing all weights to 1 as done by Webb [20]. Note also that this version of the algorithm, as is the case with our version of AdaBoost, may fail to terminate, entering an infinite loop through Steps 9 to 15. This did not occur in our experiments.

Algorithm 4 The MB algorithm

```
MB($S$, $BaseLearn$, $t$, $I$)
inputs: $S$, a sequence of $m$ examples $\langle(x_1, y_1), \ldots, (x_m, y_m)\rangle$ with labels
          $y_i \in Y = \{1, ..., k\}$.
        base learning algorithm $BaseLearn$.
        integer $t$ specifying the number of iterations.
        vector of integers $I_i$ specifying the iteration at which each
          subensemble $i \geq 1$ should terminate.

1: $S' \leftarrow S$ with random weights from the exponential distribution.
2: $l \leftarrow 1$.
3: for $i \leftarrow 1$ to $t$ do
4:   if $I_l = i$ then
5:     reset $S'$ to random weights from the exponential distribution.
6:     standardize $S'$ to sum to $m$.
7:     increment $l$.
8:   end if
9:   $C_i \leftarrow BaseLearn(S')$.
10:  $\epsilon_i \leftarrow \frac{\sum_{x_j \in S' : C_i(x_j) \neq y_j} weight(x_j)}{m}$ {the weighted error on the training set}
11:  if $\epsilon_i > 0.5$ then
12:    reset $S'$ to random weights from the exponential distribution.
13:    standardize $S'$ to sum to $m$.
14:    increment $l$.
15:    go to Step 9.
16:  else if $\epsilon_i = 0$ then
17:    set $\beta_i$ to $10^{-10}$
18:    reset $S'$ to random weights from the exponential distribution.
19:    standardize $S'$ to sum to $m$.
20:    increment $l$.
21:  else
22:    $\beta_i \leftarrow \frac{\epsilon_i}{(1 - \epsilon_i)}$.
23:    for all $x_j \in S'$ do
24:      divide $weight(x_j)$ by $2\epsilon_i$ if $C_i(x_j) \neq y_j$ and $2(1 - \epsilon_i)$ otherwise.
25:      if $weight(x_j) < 10^{-8}$ then
26:        $weight(x_j) \leftarrow 10^{-8}$
27:      end if
28:    end for
29:  end if
30: end for
31: output the final classifier: $C^*(x) = \underset{y \in Y}{argmax} \sum_{i:C_i(x)=y} log \frac{1}{\beta_i}$.
```

Together with the two base learning algorithms, C4.5 and C4.5SAS, combining the algorithms in all possible ways results in nine distinct algorithms, presented in Fig. 1.

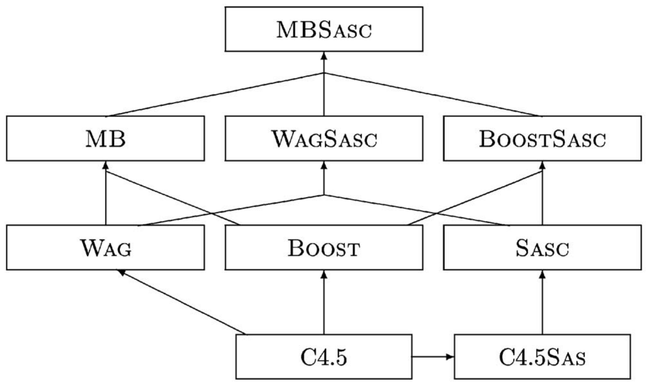

Fig. 1. Hierarchy of ensemble learning technique combinations.

5 EXPERIMENTS

We use the following notation:

- Y is the set of classes.
- $\mathcal{T}$ is a training set of example description-classification pairs.
- K is a classifier, a function from objects to classes.
- C_i is the ith classifier in ensemble C, a function from objects to classes.
- W_i is the weight given to the vote of C_i.
- t is the number of classifiers in ensemble C.
- x_i is the description of the ith case to be classified.
- y_i is the correct classification for the ith case to be classified.
- m is the number of cases to be classified.
- $\mathcal{L}$ is a learner, a function from training sets to classifiers.

We wish to evaluate the hypothesis that combining multiple distinct ensemble learning algorithms will tend to increase diversity in the predictions of ensemble members without unduly increasing the test error of the individual predictions of the ensemble members, resulting in a reduction in ensemble test error.

To evaluate this hypothesis, we need an operational measure of *diversity in the predictions of ensemble members*. We utilize the weighted statistical variance between the weighted predictions of the members of a classifier ensemble for this purpose:

$$diversity = \frac{\sum_{i=1}^{m}\left(1-\sum_{y\in Y}\left(\frac{\sum_{j=1}^{t}W_j 1(C_j(x_i)=y)}{\sum_{j=1}^{t}W_j}\right)^2\right)}{m}. \quad (2)$$

We also require an operational measure of *the test error of the individual predictions of ensemble members*. We utilize a weighted mean of the test error of the predictions of an ensemble's constituent classifiers for this purpose:

$$individual\ error = \frac{\sum_{i=1}^{m}\sum_{j=1}^{t}W_j 1(C_j(x_i)=y_i)}{m\sum_{j=1}^{t}W_j}. \quad (3)$$

We wish to examine the relationship between these two measures and error, which we define as

$$error = \frac{\sum_{i=1}^{m}1(K(x_i)\neq y_i)}{m}. \quad (4)$$

We further decompose error into bias and variance using Kohavi and Wolpert's [12] definitions thereof:

$$bias_x^2 = \frac{1}{2}\sum_{y\in Y}[P_{Y,X}(Y=y|X=x)-P_{\mathcal{T}}(\mathcal{L}(\mathcal{T})(x)=y)]^2, \quad (5)$$

$$variance_x = \frac{1}{2}\left(1-\sum_{y\in Y}P_{\mathcal{T}}(\mathcal{L}(\mathcal{T})(x)=y)^2\right), \quad (6)$$

$$\sigma_x = \frac{1}{2}\left(1-\sum_{y\in Y}P_{Y,X}(Y=y|X=x)^2\right). \quad (7)$$

The third term σ_x relates to irreducible error. We follow Kohavi and Wolpert's [12] practice of aggregating this value with bias due to the difficulty of estimating it from observations of classification performance.

We estimate all of the above terms using Webb's [20] procedure of performing 10 cycles of three-fold cross validation. This process ensures that each case is used in 20 training sets and ten test sets.

Armed with these measures and procedures we systematically explored the space of combinations of the three base ensemble learning algorithms by forming the following systems that realize the hierarchy illustrated in Fig. 1:

- **C4.5**: C4.5 Release 8, the base system.
- **C4.5**SAS: C4.5 modified to perform stochastic attribute selection as described in Section 3.
- WAG: Wagged ensembles of 100 decision trees using C4.5 as the base algorithm.
- BOOST: AdaBoost ensembles of 100 decision trees using C4.5 as the base algorithm.
- SASC: Stochastic attribute selection committees of 100 decision trees each formed by C4.5SAS.
- MB: MultiBoosted (Wagged subensembles formed by boosting) ensembles of 100 (10 $\times$ subensembles of size 10) decision trees using C4.5 as the base algorithm.
- BOOSTSASC: Boosted ensembles of 100 decision trees using C4.5SAS as the base algorithm.
- WAGSASC: Wagged ensembles of 100 decision trees using C4.5SAS as the base algorithm.
- MBSASC: MultiBoosted ensembles of 100 (10 $\times$ subensembles of size 10) decision trees using C4.5SAS as the base algorithm.

These various algorithms were applied to the 41 data sets from the UCI repository [3] described in the Appendix. Ensembles of size 100 were used as a compromise between greater compute times required by larger ensembles and the ever-decreasing average-case marginal improvement in error that can be expected from larger ensemble sizes.

Unfortunately, space constraints prevent the presentation of results at the individual data set level. Figs. 2, 3, 4, 5, and 6 and Tables 1, 2, 3, 4, and 5 provide high-level summaries of these results. The summary tables have the following format, where *row* indicates the mean value on a data set for the algorithm with which a row is labeled, while

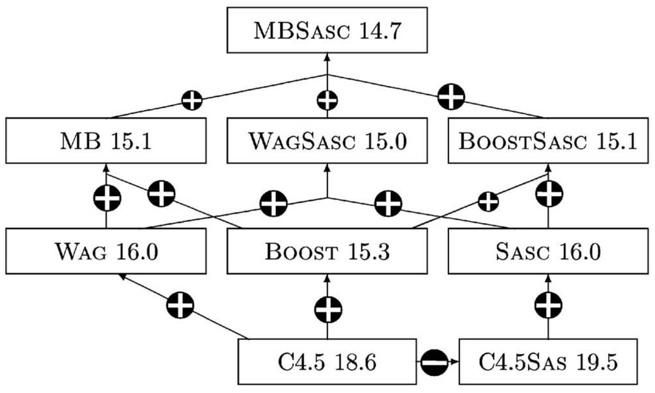

Fig. 2. Hierarchy of error outcomes.

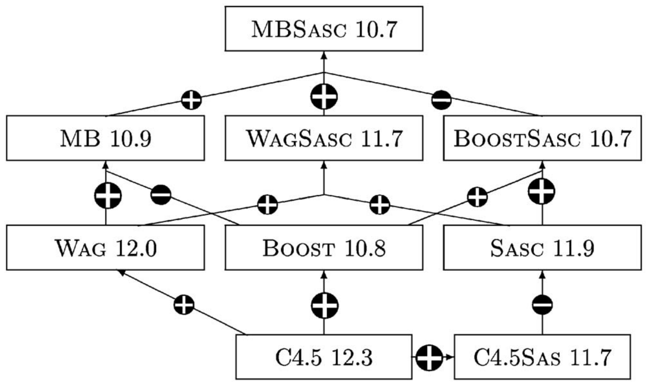

Fig. 3. Hierarchy of bias outcomes.

col indicates the mean value for the algorithm with which the column is labeled. Rows labeled $\dot{r}$ present the geometric mean of the value ratio *col*/*row*. Rows labeled s present the win/draw/loss record, where the first value is the number of data sets for which $col < row$, the second is the number for which $col = row$, and the last is the number for which $col > row$. Rows labeled p present the result of a two-tailed sign test on the win-loss record. This is the two-tailed probability of obtaining the observed record of wins to losses, or more extreme, if wins and losses were equiprobable random events. Note, these values have not been corrected to control experiment-wise type-1 error, but are in most cases so low that standard statistical corrections would not affect significance test outcomes. The figures depict the hierarchy of Fig. 1, with mean value across all data sets listed against each algorithm. The lines climbing up the hierarchy are labeled with an indication of the relative win/draw/loss record. An improvement in the win/draw/loss record is labeled with a "+" and a decline by a "-." Where the difference is statistically significant at the 0.05 level, the label is large; otherwise, it is small.

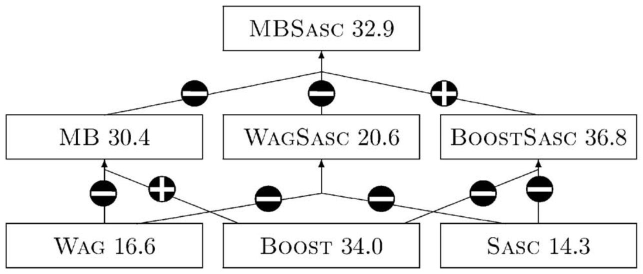

Fig. 5. Hierarchy of diversity outcomes.

5.1 Error, Bias, and Variance

Fig. 2 shows that the mean error invariably drops as we climb the hierarchy of ensemble technique combinations. Table 1 shows that all ensemble techniques have significantly better win/draw/loss records than C4.5. Moving from a single strategy to a combination of two strategies, in every case the error ratio and win/draw/loss record favors the multistrategy approach over each of its constituent strategies. The win/draw/loss record significantly favors the multistrategy approach over the constituent strategy in every case except for comparing BOOST against BOOST-SASC. Moving from combinations of two strategies to the combination of all three strategies, the error ratio and win/draw/loss record in each case favors MBSASC, but the advantage on the win/draw/loss record is only significant against BoostSASC. While the failure to obtain significant advantages at the top of the hierarchy leaves room for interpretation about whether MBSASC holds a general advantage over each of its two-strategy constituents, it at the very least appears clear that it does not hold a significant general disadvantage.

Turning to bias, all of the ensemble techniques have lower bias, favorable bias ratios, and favorable win/draw/loss records with respect to C4.5. However, the win/draw/loss records only indicate significant benefits for the ensemble techniques that incorporate boosting as a component technique. Climbing the hierarchy, MB shows a

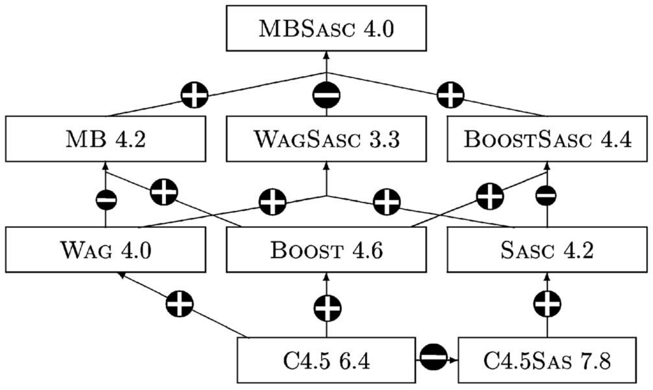

Fig. 4. Hierarchy of variance outcomes.

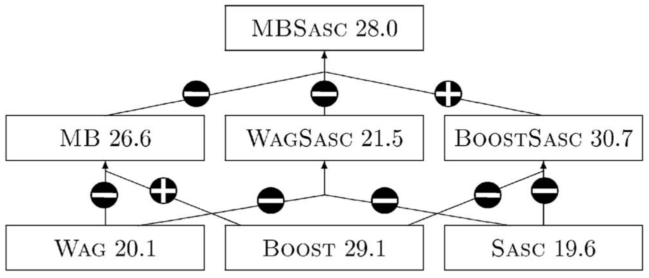

Fig. 6. Hierarchy of individual error outcomes.

TABLE 1
Error Rate Comparison Results

		C4.5SAS	BOOST	WAG	SASC	MB	BOOSTSASC	WAGSASC	MBSASC
	$\dot{r}$	1.07	0.76	0.86	0.84	0.75	0.75	0.79	0.73
C4.5	s	13/0/28	34/2/5	35/3/3	32/4/5	36/1/4	34/0/7	34/2/5	36/0/5
	p	0.0275	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
	$\dot{r}$		0.71	0.80	0.78	0.70	0.69	0.74	0.68
C4.5SAS	s		34/2/5	39/0/2	37/0/4	37/0/4	33/1/7	39/1/1	36/2/3
	p		<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
	$\dot{r}$			1.14	1.10	0.99	0.98	1.05	0.96
BOOST	s			13/1/27	11/2/28	25/6/10	22/4/15	19/1/21	27/4/10
	p			0.0385	0.0095	0.0167	0.3240	0.8746	0.0076
	$\dot{r}$				0.97	0.87	0.86	0.92	0.84
WAG	s				20/6/15	28/4/9	26/1/14	31/2/8	28/1/12
	p				0.4996	0.0026	0.0807	0.0003	0.0166
	$\dot{r}$					0.90	0.89	0.95	0.87
SASC	s					33/1/7	29/0/12	28/3/10	34/0/7
	p					<0.0001	0.0115	0.0051	<0.0001
	$\dot{r}$						0.99	1.06	0.97
MB	s						17/6/18	17/4/20	22/6/13
	p						>0.9999	0.7428	0.1755
BOOST	$\dot{r}$							1.07	0.98
+SASC	s							16/0/25	26/7/8
	p							0.2110	0.0029
WAG+	$\dot{r}$								0.91
SASC	s								24/2/15
	p								0.1996

marginally worse aggregate mean bias and bias ratio relative to BOOST and the win/draw/loss record approaches significance at the 0.05 level. In contrast, BOOSTSASC shows marginal improvements in mean bias and bias ratio, but the very small advantage in win/draw/loss record is definitely not significant. Relative to SASC, BOOSTSASC demonstrates clear wins on all metrics, but WAGSASC demonstrates only a minor win on mean bias, no win on bias ratio, and an extremely slim, far from significant win on win/draw/loss record. With respect to WAG, MB demonstrates clear wins on all metrics, while WAGSASC demonstrates marginal wins and the win on win/draw/loss record is not significant. Proceeding to the top of the hierarchy, MBSASC demonstrates a clear win over WAGSASC on all metrics, but has at most marginal, insignificant, wins against MB and actually has a worse win/drawn/loss record, albeit not significantly, than BOOSTSASC. In summary, combining Boosting with Wagging or Stochastic Attribute Selection appears to have a beneficial effect with respect to bias, but the combination of Wagging with Stochastic Attribute Selection appears to have little effect in this respect.

Turning our attention next to variance, a very different pattern appears. Again, all of the ensemble techniques outperform C4.5 on all metrics. Adding either WAG or SASC to another technique, including each other, always produces a substantial benefit on all metrics, including a significant improvement in win/draw/loss record relative to the other technique on its own. In contrast, however, adding BOOST to another technique always results in a decrease in performance on all metrics (excepting small, nonsignificant, improvements in win/draw/loss record for MB against WAG and BOOSTSASC against SASC), sometimes a substantial decrease and, in one case (MBSASC against WAGSASC), a significant worsening of the win/draw/loss record.

In summary, these results suggest that BOOST is primarily a bias reduction technique. Although it performs significant variance reduction, it is not as effective at this as WAG or SASC. Combining BOOST with WAG or SASC produces significant benefits in bias reduction over each of WAG and SASC in isolation, without a serious decline vis a vis BOOST. In contrast, WAG and SASC are primarily variance reduction techniques. Combining either with another technique results in variance reduction vis a vis

TABLE 2
Bias Comparison Results

		C4.5SAS	BOOST	WAG	SASC	MB	BOOSTSASC	WAGSASC	MBSASC
	$\dot{r}$	0.96	0.80	0.98	0.94	0.81	0.80	0.94	0.80
C4.5	s	30/2/9	33/2/6	21/2/18	21/2/18	34/1/6	31/0/10	24/1/16	32/1/8
	p	0.0011	<0.0001	0.7493	0.7493	<0.0001	0.0015	0.2682	0.0002
	$\dot{r}$		0.84	1.02	0.98	0.85	0.83	0.98	0.83
C4.5SAS	s		30/0/11	10/5/26	12/6/23	29/1/11	27/2/12	13/2/26	26/2/13
	p		0.0043	0.0113	0.0895	0.0064	0.0237	0.0533	0.0533
	$\dot{r}$			1.22	1.17	1.01	0.99	1.17	1.00
BOOST	s			6/1/34	8/1/32	11/7/23	19/4/18	8/0/33	16/4/21
	p			<0.0001	0.0002	0.0576	>0.9999	0.0001	0.5114
	$\dot{r}$				0.96	0.83	0.81	0.96	0.81
WAG	s				21/4/16	35/1/5	35/0/6	24/1/16	36/0/5
	p				0.5114	<0.0001	<0.0001	0.2682	<0.0001
	$\dot{r}$					0.87	0.85	1.00	0.85
SASC	s					30/3/8	35/0/6	22/1/18	34/1/6
	p					0.0005	<0.0001	0.6358	<0.0001
	$\dot{r}$						0.98	1.15	0.98
MB	s						18/10/13	9/0/32	21/4/16
	p						0.4731	0.0004	0.5114
BOOST	$\dot{r}$							1.18	1.00
+SASC	s							7/1/33	13/8/20
	p							<0.0001	0.2962
WAG+	$\dot{r}$								0.85
SASC	s								31/3/7
	p								0.0001

the other technique on its own without a serious increase in variance in comparison to WAG or SASC in isolation. Putting all of these factors together, combining techniques is generally beneficial with respect to error reduction because there is always a benefit either in terms of bias or variance reduction against each constituent technique, which is usually gained without a substantial or significant loss with respect to the other constituent of error.

5.2 Diversity and Individual Error

Webb's [20] original motivation for the development of MultiBoosting and the subsequent multistrategy ensemble learning techniques was that combining ensemble learning strategies would increase diversity without unduly affecting the error of individual members of the ensemble. Tables 4 and 5 show that this is indeed the case when one considers every step from a constituent ensemble learning technique to a multistrategy technique except for BOOST to MB and BOOSTSASC to MBSASC. The steps from WAG to MB, BOOST to BOOSTSASC, SASC to BOOSTSASC, WAG to WAGSASC, SASC to WAGSASC, MB to MBSASC, and WAGSASC to MBSASC all result in increases in diversity accompanied by small but significant increases in individual error but decreases in ensemble error. However, adding WAG to BOOST or MBSASC has the opposite effect on diversity and individual error. In both cases, both diversity and individual error significantly decrease. However, in both cases, this nonetheless results in a decrease in ensemble error. We have the unexpected outcome that the original motivation for MultiBoosting appears to apply to other combinations of standard classifier ensemble learning techniques, but not to the MultiBoost combination of WAG and BOOST!

Having observed this phenomenon, it is straightforward to explain. Compared with WAG and SASC, BOOST has very high diversity. It is credible that the Boosting process will tend to drive diversity up at ever-increasing rates as ensemble size increases. This is due to the manner in which Boosting attempts to focus the learner on areas of the instance space that previous ensemble members fail to handle adequately. By definition, this means that it is attempting to make the latter ensemble members systematically differ in their classifications from prior members. However, this process will also drive up the individual error of each successive ensemble member when applied to the domain as a whole because each successive member concentrates primarily on ever-decreasing areas of the total

TABLE 3
Variance Comparison Results

		C4.5SAS	BOOST	WAG	SASC	MB	BOOSTSASC	WAGSASC	MBSASC
	$\dot{r}$	1.25	0.75	0.65	0.60	0.69	0.72	0.51	0.65
C4.5	s	6/2/33	34/2/5	38/2/1	37/3/1	34/2/5	33/3/5	37/3/1	34/3/4
	p	$<$0.0001	$<$0.0001	$<$0.0001	$<$0.0001	$<$0.0001	$<$0.0001	$<$0.0001	$<$0.0001
	$\dot{r}$		0.60	0.53	0.48	0.56	0.58	0.41	0.52
C4.5SAS	s		36/0/5	40/0/1	39/0/2	37/0/4	36/0/5	40/0/1	38/0/3
	p		$<$0.0001	$<$0.0001	$<$0.0001	$<$0.0001	$<$0.0001	$<$0.0001	$<$0.0001
	$\dot{r}$			0.87	0.80	0.93	0.96	0.68	0.87
BOOST	s			20/5/16	18/5/18	31/9/1	21/11/9	31/3/7	32/6/3
	p			0.6177	$>$0.9999	$<$0.0001	0.0428	0.0001	$<$0.0001
	$\dot{r}$				0.92	1.06	1.10	0.78	1.00
WAG	s				21/5/15	19/6/16	18/5/18	34/4/3	21/5/15
	p				0.4050	0.7359	$>$0.9999	$<$0.0001	0.4050
	$\dot{r}$					1.16	1.20	0.85	1.09
SASC	s					21/8/12	21/7/13	29/9/3	26/4/11
	p					0.1628	0.2295	$<$0.0001	0.0201
	$\dot{r}$						1.03	0.74	0.94
MB	s						11/10/20	30/3/8	24/7/10
	p						0.1496	0.0005	0.0243
BOOST	$\dot{r}$							0.71	0.91
+SASC	s							31/4/6	32/7/2
	p							$<$0.0001	$<$0.0001
WAG+	$\dot{r}$								1.27
SASC	s								12/4/25
	p								0.0470

instance space. Successive ensemble members have ever-increasing individual error in order to gain the ever increasing diversity, a trade off that, in practice, results in ever-diminishing benefit in terms of reduction of overall ensemble error. Credibility is lent to this argument when one compares the diversity and individual error of AdaBoost ensembles of size 10 and 100. To this end, the AdaBoost experiments were rerun on the same data set cross-validation partitions, but with an ensemble size of 10. With respect to diversity, in contrast to the mean across all data sets of 0.340 obtained for boosted ensembles of size 100, boosted ensembles of size 10 had a mean of 0.276. With respect to individual error, the mean dropped from 0.291 to 0.260 for ensembles of size 10. With respect to both measures, the mean on every data set was lower for ensemble size 10 than for size 100.

MultiBoosting creates boosted subensembles of size 10. Compared with the mean diversity and individual error obtained for boosted ensembles of this size, MultiBoosting does lead to increases. However, it is clear that MultiBoosting compared with Boosting ensembles of size 100 leads to decreases in diversity and individual error. By creating boosted subensembles of size 10, MultiBoosting delivers lower internal error than Boosting. However, this improvement in internal error comes at a cost of a slight decrease in diversity. Contrary to our expectations, rather than increasing diversity vis a vis Boosting, MultiBoosting decreases diversity, but, in practice does so in a manner that forms a better diversity against individual error trade off than that formed by Boosting alone.

Other than these two cases where Wagging is combined with Boosting, the results are, however, consistent with our expectations that combining ensemble techniques would result in increased diversity and individual error resulting in trade offs that reduce overall ensemble error.

6 CONCLUSIONS

This paper has examined techniques for combining simple ensemble learning approaches with the aim of exploring the relationship between ensemble member diversity and ensemble error. The results strongly support the proposition that combining effective ensemble learning strategies is conducive to reducing test error. A specific hypothesis about this effect was examined—that combining ensemble learning strategies would increase diversity at the cost of a small increase in individual test error resulting in a trade off that reduced overall ensemble test error. While this hypothesis

TABLE 4
Diversity Comparison Results

		Wag	Sasc	MB	BoostSasc	WagSasc	MBSasc
	$\dot{r}$	0.37	0.29	0.85	1.09	0.47	0.93
Boost	s	41/0/0	41/0/0	41/0/0	2/0/39	41/0/0	28/1/12
	p	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	0.0166
	$\dot{r}$		0.78	2.30	2.96	1.29	2.51
Wag	s		26/1/14	0/0/41	0/0/41	1/1/39	0/0/41
	p		0.0807	<0.0001	<0.0001	<0.0001	<0.0001
	$\dot{r}$			2.95	3.79	1.65	3.22
Sasc	s			0/0/41	0/0/41	0/0/41	0/0/41
	p			<0.0001	<0.0001	<0.0001	<0.0001
	$\dot{r}$				1.29	0.56	1.09
MB	s				0/0/41	41/0/0	1/0/40
	p				<0.0001	<0.0001	<0.0001
Boost	$\dot{r}$					0.43	0.85
+Sasc	s					41/0/0	41/0/0
	p					<0.0001	<0.0001
Wag+	$\dot{r}$						1.95
Sasc	s						0/0/41
	p						<0.0001

TABLE 5
Individual Error Rate Comparison Results

		Wag	Sasc	MB	BoostSasc	WagSasc	MBSasc
	$\dot{r}$	0.57	0.56	0.86	1.07	0.62	0.92
Boost	s	41/0/0	41/0/0	41/0/0	4/0/37	41/0/0	32/0/9
	p	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	0.0004
	$\dot{r}$		0.98	1.52	1.89	1.10	1.63
Wag	s		29/0/12	0/0/41	0/0/41	8/0/33	0/0/41
	p		0.0115	<0.0001	<0.0001	0.0001	<0.0001
	$\dot{r}$			1.55	1.92	1.11	1.66
Sasc	s			0/0/41	0/0/41	0/0/41	0/0/41
	p			<0.0001	<0.0001	<0.0001	<0.0001
	$\dot{r}$				1.24	0.72	1.07
MB	s				0/0/41	40/0/1	3/0/38
	p				<0.0001	<0.0001	<0.0001
Boost	$\dot{r}$					0.58	0.86
+Sasc	s					41/0/0	41/0/0
	p					<0.0001	<0.0001
Wag+	$\dot{r}$						1.49
Sasc	s						0/0/41
	p						<0.0001

was consistent with the results obtained when stochastic attribute selection was combined with another ensemble learning strategy, it was not consistent with the results for the MultiBoosting approach to combining Boosting and Wagging, where, compared with the Boosting-based strategy (AdaBoost alone or AdaBoost combined with SASC), the

TABLE 6
Description of Learning Tasks

Domain	Size	No. of	No. of Att.	
		Classes	Num	Discr
Adult	48842	4	6	7
Annealing	898	6	6	32
Audiology	226	24	0	69
Automobile	205	7	15	10
Balance scale	625	3	4	0
Breast (S)	286	2	0	9
Breast (W)	699	2	9	0
Chess (KR-KP)	3169	2	0	36
Credit (A)	690	2	6	9
Credit (G)	1000	2	7	13
Discordant	3772	2	7	22
Echocardiogram	131	2	6	1
Glass	214	6	9	0
Heart	270	2	7	6
Heart (C)	303	2	13	0
Heart (H)	294	2	13	0
Hepatitis	155	2	6	13
Horse colic	368	2	7	15
House votes 84	435	2	0	16
Hypo	3772	5	7	22
Iris	150	3	4	0
Labor	57	2	8	8
LED 24	200	10	0	24
Letter	20000	26	16	0
Liver disorders	345	2	6	0
Lymphography	148	4	0	18
NetTalk letter	5438	163	0	7
NetTalk stress	5438	5	0	7
NetTalk phoneme	5438	52	0	7
New thyroid	215	3	5	0
Pima diabetes	768	2	8	0
Postoperative	90	3	1	7
Promoters	106	2	0	57
Segment	2310	7	19	0
Sick	3772	2	7	22
Sonar	208	2	60	0
Soybean large	683	19	0	35
Splice junction	3177	3	0	60
Vehicle	846	4	18	0
Waveform-21	300	3	21	0
Wine	178	3	13	0

combination appears to have the effect of reducing individual test error at the cost of a small reduction in diversity, a different trade off which nonetheless results in reduced ensemble test error.

The success of these ensemble learning techniques mandates further investigation. We have examined only a single base learner and only one ensemble size. Our expectation is that the results will generalize to other base learners and ensemble sizes, but this belief warrants evaluation. This paper has highlighted the trade off between diversity and individual test error in ensemble learning strategies. Research into how this trade off should be managed and how to identify when a particular trade off is likely to be productive are likely prove fruitful areas for future investigation.

The computational overheads of combining ensemble learning strategies are negligible. The same number of ensemble members are learned and, hence, the same numbers of calls to the base learner are required. Indeed, the strategy of wagging subcommittees formed by boosting can support greater computational efficiency by allowing parallelism of a form not readily possible with boosting alone. However, despite negligible computational cost, our experiments have shown that appreciable and reasonably consistent reductions in test error can be obtained. There appears to be no reason not to combine ensemble learning strategies in a learning scenario for which ensemble learning is appropriate.

APPENDIX

DATA SETS

Forty-one natural domains from the UCI machine learning repository are used. Table 6 summarizes the characteristics of these domains, including data set size, the number of classes, the number of numeric attributes, and the number of discrete attributes. This test suite covers a wide variety of different domains with respect to data set size, the number of classes, the number of attributes, and types of attributes.

ACKNOWLEDGMENTS

The Breast Cancer (S) and Lymphography data sets were provided by the Ljubljana Oncology Institute, Slovenia. Thanks to the UCI Repository's maintainers and donors for providing access to the data sets used herein.

REFERENCES

[1] K. Ali, "Learning Probabilistic Relational Concept Descriptions," PhD thesis, Dept. of Information and Computer Science, Univ. of California, Irvine, 1996.
[2] E. Bauer and R. Kohavi, "An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants," *Machine Learning,* vol. 36, pp. 105-139, 1999.
[3] C. Blake and C.J. Merz, "UCI Repository of Machine Learning Databases [Machine-readable data repository]," Univ. of California, Dept. of Information and Computer Science, Irvine, 2001.
[4] L. Breiman, "Bagging Predictors," *Machine Learning,* vol. 24, pp. 123-140, 1996.
[5] L. Breiman, "Bias, Variance, and Arcing Classifiers," Technical Report 460, Statistics Dept., Univ. of California, Berkeley, 1996.
[6] L. Breiman, "Arcing the Edge," Technical Report 486, Statistics Dept., Univ. of California, Berkeley, 1997.
[7] L. Breiman, "Arcing Classifiers," *The Annals of Statistics,* vol. 26, no. 3, pp. 801-849, 1998.
[8] T.G. Dietterich, "An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization," *Machine Learning,* vol. 40, no. 2, pp. 139-158, 2000.
[9] T.G. Dietterich and E.B. Kong, "Machine Learning Bias, Statistical Bias, and Statistical Variance of Decision Tree Algorithms," technical report, Dept. of Computer Science, Oregon State Univ., 1995.

[10] Y. Freund and R.E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *J. Computer and System Sciences,* vol. 55, no. 1, pp. 119-139, 1997.
[11] J.H. Friedman, T. Hastie, and R. Tibshirani, "Additive Logistic Regression: A Statistical View of Boosting," *Annals of Statistics,* vol. 28, no. 2, pp. 337-374, 2000.
[12] R. Kohavi and D. Wolpert, "Bias Plus Variance Decomposition for Zero-One Loss Functions," *Proc. 13th Int'l. Conf. Machine Learning,* pp. 275-283, 1996.
[13] A. Krogh and J. Vedelsby, "Neural Network Ensembles, Cross Validation, and Active Learning," *Advances in Neural Information Processing Systems,* G. Tesauro, D. Touretzky, and T. Leen, eds., vol. 7, pp. 231-238, 1995.
[14] S. Lee and J.F. Elder, "Bundling Heterogeneous Classifiers with Advisor Perceptrons," Technical Report 97-1, Elder Research, Charlottesville, Va., 1997.
[15] D. Margineantu and T.G. Dietterich, "Pruning Adaptive Boosting," *Proc. 14th Int'l. Conf. Machine Learning (ICML-97),* pp. 211-218, 1997.
[16] J.R. Quinlan, *C4.5: Programs for Machine Learning.* San Mateo, Calif.: Morgan Kaufmann, 1993.
[17] J.R. Quinlan, "Improved Use of Continuous Attributes in C4.5," *J. Artificial Intelligence Research,* vol. 4, pp. 77-90, 1996.
[18] R.E. Schapire, Y. Freund, P. Bartlett, and W.S. Lee, "Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods," *The Annals of Statistics,* vol. 26, no. 5, pp. 1651-1686, Oct. 1998.
[19] V.N. Vapnik, *Estimation of Dependencies Based on Empirical Data.* Springer-Verlag, 1982.
[20] G.I. Webb, "MultiBoosting: A Technique for Combining Boosting and Wagging," *Machine Learning,* vol. 40, no. 2, pp. 159-196, 2000.
[21] Z. Zheng and G.I. Webb, "Stochastic Attribute Selection Committees," *Proc. 11th Australian Joint Conf. Artificial Intelligence,* pp. 321-332, 1998.

Geoffrey I. Webb received the BA and PhD degrees in computer science from La Trobe University. He is a research professor on the Faculty of Information Technology, Monash University. From 1998 to 2002, he held a personal chair in computer science at Deakin University. His research interests include machine learning, data mining, and user modeling. Dr. Webb has published more than 100 scientific papers. He is the author of the commercial data mining software package *Magnum Opus*. He is a member of the editorial boards of *Machine Learning*, *User Modeling and User-Adapted Interaction*, and *Knowledge and Information Systems*. He is a member of the IEEE.

Zijian Zheng received the PhD degree in machine learning from the University of Sydney, Australia, in 1996 and then worked three years as a senior research fellow in the areas of machine learning and data mining at Deakin University, Australia. He is a software design engineer at Microsoft, developing data mining techniques. In 1999, Dr. Zheng started his industrial career at Blue Martini Software, USA, where he worked on data mining development, research, and practice with commercial Web, transactional, campaign, and customer data. Dr. Zheng also worked on data mining with semiconductor data at IDS Software Systems.