A TWO-STAGE COMMITTEE MACHINE OF NEURAL NETWORKS

Jen-Feng Wang, Chinson Yeh, Chen-Wen Yen*, and Mark L. Nagurka

ABSTRACT

In solving pattern recognition problems, many ensemble methods have been proposed to replace a single classifier by a classifier committee. These methods can be divided roughly into two categories: serial and parallel approaches. In the serial approach, component classifiers are created by focusing on different parts of the training set in different learning phases. In contrast, without paying special attention to any part of the dataset, the parallel approach generates classifiers independently. By integrating these two techniques and by using a neural network approach for the base classifier, this work proposes a design method for a two-stage committee machine. In the first stage of the approach the entire dataset is used to train an averaging ensemble. Based on the classification results of this first stage, hard-to-classify samples are selected and sent to the second stage. To improve the classification accuracy for these samples, a computationally more intensive bagging ensemble is employed in the second stage. These two neural network ensembles work in series whereas the component neural networks in each of the ensembles are trained in parallel. Experimental results demonstrate the accuracy and robustness of the proposed approach.

Key Words: neural network, committee machine, bagging, AdaBoost.

I. INTRODUCTION

It has been found, both theoretically and empirically, that combining the results of multiple classifiers can create a classifier committee that is generally more accurate than any of the component classifiers. In general, these classifier combination methods can be divided into two categories: parallel and serial approaches. In the former approach, classifiers are trained in parallel using the same dataset. These independently trained classifiers are then combined using rules such as averaging (Haykin, 1999; Kuncheva, 2002) voting (Lam and Suen, 1997; Windeatt, 2003), multiplying (Tax *et al*., 2000; Alexandre *et al*., 2001), linear combination (Hashem, 1997; Ueda, 2000; Fumera and Roli, 2004), and other methods.

*Corresponding author. (Tel.: +886-7-3482965; fax: +886-7-5254299; Email address: vincen@mail.nsysu.edu.tw)

J. F. Wang, C. Yeh and C. W. Yen are with theDepartment of Mechanical and Electro-Mechanical Engineering, National Sun Yat-Sen University, Kaohsiung 80024, Taiwan, R.O.C.

M. L. Nagurka is with the Department of Mechanical Engineering, Marquette University, Milwaukee, WI 53201-1881 USA.

In the serial approach a committee machine is constructed by increasing the number of classifiers one-at-a-time. The dataset used to train each classifier is chosen based on the performance of the earlier classifiers in the series. The goal is to produce new classifiers that can somehow compensate for the weakness of the existing classifiers. A well-known example of the serial approach is boosting (Schapire, 2002), which tries to generate new classifiers that are better able to correctly classify samples for which the current committee performance is poor.

Compared with the serial approach, a limitation of the parallel approach is that its classifiers are independently designed and the interaction among the classifiers has thus not been utilized to enhance the performance of the committee. As a result, classifiers designed by the parallel approach tend to be less accurate than those constructed by the serial approach. On the other hand, a drawback of the serial approach is that in dealing with a noisy dataset it can easily overfit, since it is typically designed to focus more on the misclassified examples which in some cases are the noisy data.

A goal of this work is to develop a committee machine design method that combines the advantages

of the serial and parallel approaches. It relies on neural networks for the base classifier. In analogy with the serial approach, the proposed approach divides the classification process into two stages. In particular, after using an averaging ensemble in the first stage, the second stage tries to improve the classification accuracy by focusing on a selected data subset which is more susceptible to classification error than the remaining dataset. To preserve the advantages of the parallel approach, these two serially combined neural ensembles are both constructed using a conventional parallel approach.

This paper is organized as follows: The following section discusses the basic ideas of two of the most popular classifier combination methods, bagging and boosting. Also addressed in Section II is the potential instability problem of classifiers. The proposed approach is developed in Section III. Section IV presents experimental results that demonstrate the method's accuracy and robustness. Conclusions are offered in Section V.

II. BAGGING AND BOOSTING

The approaches of bagging (Breiman, 1996) and boosting (Schapire, 2002) have received extensive attention. Compared with conventional classifier combination methods, a distinct feature shared by both approaches is the use of the resampling technique. A fundamental difference between them is that bagging constructs classifiers in parallel whereas boosting creates classifiers in series.

In bagging (or bootstrap aggregating), each classifier is trained on a set of N examples, drawn randomly with replacement from the original training set of size N. Training sets generated in such a way are called bootstrap replicates of the original set (Efron and Tibshirani, 1993). Specifically, with the neural network as the base classifier, bagging works as follows. First, the original training set is represented as:

$$T = \{(\boldsymbol{x}_i, y_i),\ i = 1, \cdots, N\}, \tag{1}$$

where $\boldsymbol{x}_i$ is the input vector, y_i is the output of the ith sample and N is the number of the data. Trained by training set T, given an input $\boldsymbol{x}$, the output of the neural network can be denoted as $f(\boldsymbol{x}, T)$. Next, using a resampling technique to produce a sequence of bootstrap replicates $\{G_k, k = 1, \cdots, K\}$ from T, one can construct K component neural networks by using these K bootstrap replicates. With $\boldsymbol{x}$ as the input, the output of the kth component neural network can be expressed as $f(\boldsymbol{x}, G_k)$. Finally, for a sample $\boldsymbol{x}$, the classification result can be determined using the following averaging rule:

$$f_B(\boldsymbol{x}) = \frac{1}{K}\sum_{k=1}^{K} f(\boldsymbol{x}, G_k)\ . \tag{2}$$

As a serial approach, boosting tries to combine a series of "weak" classifiers (whose individual performance is only slightly better than random guessing) to form a stronger classifier. Several variants of the boosting approach have been proposed. Here, this paper considers the well known AdaBoost method (Freund and Schapire, 1997), which has been successfully applied to many pattern recognition problems (Maclin and Opitz, 1997; Bauer and Kohavi, 1999). The basic idea of AdaBoost is to generate a series of classifiers using reweighed versions of the training set T. Specifically, by weighting all samples equally at the beginning, the dataset F_j used to train the jth classifier is created by sampling examples that are incorrectly classified by previous classifiers more frequently. This allows hard-to-classify samples to have ever-increasing influence since a subsequent classifier will focus more on the samples that have been misclassified by its predecessors. Finally, the results of these component classifiers can be integrated using the following linear combination rule:

$$f_{AB}(\boldsymbol{x}) = \sum_{j=1}^{J} w_j f(\boldsymbol{x}, F_j)\ , \tag{3}$$

where J is the number of component classifiers. The weighting coefficients w_j are determined such that those more accurate classifiers will have more influence on the final classification decision.

Both bagging and boosting have been found to be effective in working with unstable classifiers, such as decision trees and neural networks. The instability of a classifier can be measured by the sensitivity of the classification accuracy to the training set perturbation. For neural networks, this instability property is particularly noticeable for problems that have small training sets. The reason is that, without sufficient training data, it is unlikely that a neural network can perform accurate learning. Consequently, neural networks built on small training sets are usually biased. They may have a large error variance and are thus relatively unstable. In addition to increasing the size of the training set, this instability problem can be partially resolved by bagging or boosting since these two methods can reduce the variance of the classifier (Bauer and Kohavi, 1999; Breiman, 1998).

Previous studies (Maclin and Opitz,1997; Opitz and Maclin,1999; Dietterich, 2000; Chan *et al.*, 2001) have shown that both bagging and boosting approaches can improve the classification accuracy over individual classifiers including decision trees and neural networks. The studies also found that the bagging approach tends to be more consistent than the

boosting approach, despite the fact that the latter is often more accurate when applied properly. However, in dealing with problems with substantial noise, the bagging approach typically outperforms the boosting approach. This can be explained by noticing that the boosting may overfit the training set since the later classifiers may over-emphasize examples that are actually noisy data. This phenomenon will be observed in the experimental studies.

III. THE PROPOSED METHOD

The two-stage classification process proposed here is conceptually and operationally simple. The first stage uses an averaging neural ensemble to perform learning. In similarity to the boosting approach that trains individual classifiers sequentially to learn harder and harder parts of the classification problem, hard-to-classify samples are selected and sent to the second stage for further processing. Since the bagging approach has been shown to be effective in improving the accuracy of neural classifiers (Opitz and Maclin, 1997) and is also robust to noise, the second stage of the proposed approach adopts a bagging ensemble to classify this hard-to-classify training subset. The key issue then is how to determine this hard-to-classify training subset so that the overall classification accuracy can be maximized. This issue can be divided further into two subproblems. First, a criterion is needed to determine which samples are more difficult to classify than others. Second, the optimal size of the hard-to-classify training subset must be determined in order to minimize the classification error.

In responding to the first problem, a criterion is proposed to quantify the degree-of-classification-difficulty for the training samples. This criterion is closely related to the following rule for classification. Typically, a classifier assigns an input vector $\boldsymbol{x}$ to class C_k if

$$d_k(\boldsymbol{x}) > d_j(\boldsymbol{x}) \text{ for } k, j = 1, \cdots, M \text{ and } j \neq k, \quad (4)$$

where $d_1(x)$, $d_2(x)$, $\cdots$ $d_M(x)$ is a set of discriminant functions used by the classifier and M is the number of classes. To minimize the probability of misclassification, the Bayes' rule for minimum error chooses the discriminant function $d_k(x)$ as the posterior probability associated with class C_k, that is,

$$d_k(\boldsymbol{x}) \approx P(C_k|\boldsymbol{x}). \quad (5)$$

It has been shown that the outputs of a neural network can be trained to approximate the posterior probabilities (Bishop, 1995). For example, for a two-class problem, this can be achieved by first setting the number of the neural network outputs to two and by specifying the target outputs to be [1, 0] and [0, 1] for C_1 and C_2 samples, respectively. Next, after defining the error as the difference between target and actual outputs and by using the mean square error as the objective function to be minimized, the first and second outputs of the neural network can be trained to approximate the posterior probabilities of C_1 and C_2, respectively. As a result, these two neural network outputs are typically chosen as the discriminant functions $d_1(x)$ and $d_2(x)$.

By defining training error $e(x)$ as the difference between the desired and actual neural network outputs and by representing the training error associated with $d_1(x)$ and $d_2(x)$ as $e_1(x)$ and $e_2(x)$, respectively, it follows that:

$$d_1(\boldsymbol{x}) + e_1(\boldsymbol{x}) = P(C_1|\boldsymbol{x}), \quad (6)$$

$$d_2(\boldsymbol{x}) + e_2(\boldsymbol{x}) = P(C_2|\boldsymbol{x}). \quad (7)$$

The difference between the two discriminant functions is then

$$d_1(\boldsymbol{x}) - d_2(\boldsymbol{x}) = P(C_1|\boldsymbol{x}) - P(C_2|\boldsymbol{x}) + e_2(\boldsymbol{x}) - e_1(\boldsymbol{x}). \quad (8)$$

Based on the classification rule of Eq. (4), the Bayes' rule for minimum error will be violated if

$$\text{sgn}(d_1(\boldsymbol{x}) - d_2(\boldsymbol{x})) \neq \text{sgn}(P(C_1|\boldsymbol{x}) - P(C_2|\boldsymbol{x})), \quad (9)$$

where sgn(·) represents the signum function.

According to Eq. (9), the Bayes' rule for minimum error will not be violated as long as the following inequality is satisfied.

$$|P(C_1|\boldsymbol{x}) - P(C_2|\boldsymbol{x})| > |e_1(\boldsymbol{x}) - e_2(\boldsymbol{x})| \quad (10)$$

Eq. (10) shows that a sample is more tolerant of training error if it has a larger value of $|P(C_1|\boldsymbol{x}) - P(C_2|\boldsymbol{x})|$. Therefore, for a two-class problem, the term $|P(C_1|\boldsymbol{x}) - P(C_2|\boldsymbol{x})|$ seems to be a reasonable index for characterizing the degree-of-classification-difficulty. Unfortunately, the posterior probabilities are typically unknown. However, since the outputs of a neural network can be trained to approximate the posterior probabilities, $|P(C_1|\boldsymbol{x}) - P(C_2|\boldsymbol{x})|$ is replaced by $|d_1(\boldsymbol{x}) - d_2(\boldsymbol{x})|$ as a measure for the degree-of-classification-difficulty.

The next problem is to determine how many samples the hard-to-classify training subset should contain. To resolve this problem, two relevant properties of bagging (Breiman, 1998) are considered:

1. With judicious component classifiers, the committee machine designed by bagging can provide a

near optimal solution. However, if the component classifiers are not sufficiently accurate, bagging can also degrade the classification accuracy.

2. By defining the instability of a classifier as the sensitivity of the classifier's accuracy to the content of the training set, applying bagging to unstable classifiers usually improves the classification results. In contrast, applying bagging to stable classifiers does not seem to be useful.

These two properties indicate that the success of bagging depends heavily on the accuracy and instability of the component classifiers. For the proposed approach, accuracy and instability are two conflicting requirements. Specifically, increasing the size of the hard-to-classify training subset can improve the accuracy and reduce the instability of the component neural networks at the same time. Therefore, determining the optimal size for the hard-to-classify training subset so that the accuracy and instability requirements can be met simultaneously becomes a key issue for the proposed approach.

An analytical solution that can determine the optimal size of the hard-to-classify training subset by balancing these two conflicting requirements of accuracy and instability does not seem to be available. A computationally intensive but conceptually simple technique to resolve this difficulty is to perform a systematic search by incrementally increasing the size of the hard-to-classify training subset. Based on this idea and the proposed measure for degree-of-classification-difficulty, the approach here can be expressed procedurally (below). In developing the approach, attention is restricted to two-class problems. This does not limit the applicability of the approach since a multiclass problem can always be converted into a series of two-class problems (Knerr *et al.*, 1990; Anand *et al.*, 1995).

The design process of the proposed two-stage committee machine consists of the following steps:

(1) Construct an averaging neural ensemble and record the resulting $|d_1(\boldsymbol{x}) - d_2(\boldsymbol{x})|$ value for every sample.
(2) Set $a = 0$.
(3) Divide the training set into two subsets Z_1 and Z_2 by letting Z_1 contains the samples whose $|d_1(\boldsymbol{x}) - d_2(\boldsymbol{x})|$ value is larger than a and Z_2 contains the remaining samples.
(4) Construct a bagging ensemble by using Z_2 as its training set.
(5) After using the averaging ensemble to classify Z_1 and the bagging ensemble to classify Z_2, determine the total number of incorrectly classified samples.
(6) Set $a = a + \Delta a$ and continue the solution process by going back to step 3 until Z_1 becomes an empty set. In this work, the parameter Δa is set to 0.01.
(7) From all the tested two-state committee machines, select the one that yields the smallest classification error as the final design. Signify the corresponding a as a^*.

In the procedure the optimal size for the hard-to-classify training subset is determined by testing every quantized value for parameter a. Initially, with parameter a set to zero, the two-stage committee machine is essentially an averaging ensemble. After parameter a reaches its maximum, the two-stage committee machine transforms to a standard bagging ensemble. As a result, the proposed approach is expected to perform no worse than the averaging and bagging ensembles. The tradeoff of this flexibility for adjusting parameter a is that the computational cost increases linearly with the number of quantization levels of parameter a. However, this computational cost is only required for the classifier design phase, which is basically a one-time process and is typically executed off-line. For the repetitive classification tasks, which often require on-line operation, the computational cost of the approach is of the same order as that of the conventional committee machine. To perform classification using the proposed two-stage committee machine, the sample is first classified by the averaging neural ensemble. Its classification result is accepted if its $|d_1(\boldsymbol{x}) - d_2(\boldsymbol{x})|$ value is no less than a^*. Otherwise, this sample is considered to be a hard-to-classify sample and the bagging ensemble is used to determine its membership.

IV. EXPERIMENTAL RESULTS

This section compares the proposed approach with neural ensembles constructed by an averaging rule, bagging and AdaBoost. In implementing these ensemble methods, the multilayered perceptron (MLP) is chosen as the base classifier. Additionally, in constructing the AdaBoost neural ensembles, AdaBoost.M2, which improves upon some of the properties of the original AdaBoost algorithm, is used to train the MLPs (Freund and Schapire, 1997).

The tested datasets are real-world problems obtained from the UCI repository of Machine Learning Databases and Domain Theories (Blake and Merz, 1998). The contents of these datasets are summarized in Table 1. In comparing the tested methods, each dataset is divided into training, validation and testing subsets with an 8:1:1 ratio. The training subset is used to adjust the connection weights of the MLP; the validation subset is used by the early-stop technique to avoid overfitting. The testing subset is used to characterize the generalization accuracy of the MLP. For the sake of reliability, the training process

Table 1 Summary of the tested datasets

Dataset	Number of features	Number of samples
Sonar	60	208
Hepatitis	19	155
Horse-Colic	22	368
Heart-Statlog	13	270
Heart-C	13	303
Heart-H	12	294
House-Votes-84	16	435
German (Credit-g)	24	1000
Credit-a	15	690
Australian	14	690
Breast cancer	9	699
Kr-vs-Kp	36	3196
Diabetes	8	768

Table 2 Summary of the means and standard deviation of the testing subset accuracy (without noise)

Dataset	Testing accuracy % (Mean ± Standard deviation)			
	Averaging	Bagging	AdaBoost	Two stage
Sonar	75.87 ± 8.64	78.26 ± 8.34	78.15 ± 8.35	77.89 ± 8.39
Hepatitis	81.07 ± 5.96	85.36 ± 7.08	87.06 ± 7.18	86.34 ± 7.47
Horse-Colic	87.33 ± 5.04	88.50 ± 4.73	88.38 ± 4.93	88.48 ± 4.79
Heart-Statlog	82.92 ± 7.11	84.60 ± 6.76	84.09 ± 6.97	84.33 ± 7.03
Heart-C	83.21 ± 6.46	84.21 ± 6.24	83.94 ± 6.24	84.22 ± 6.23
Heart-H	95.10 ± 4.38	96.88 ± 3.10	98.02 ± 2.52	97.75 ± 2.58
House-Votes-84	93.53 ± 3.86	95.21 ± 2.94	95.78 ± 2.77	95.51 ± 2.95
German (Credit-g)	70.73 ± 1.89	76.14 ± 3.20	75.06 ± 3.03	76.72 ± 3.50
Credit-a	86.16 ± 3.95	86.74 ± 3.87	86.62 ± 3.85	86.73 ± 3.85
Australian	86.49 ± 3.85	86.96 ± 3.70	86.85 ± 3.71	87.18 ± 3.75
Breast cancer	96.00 ± 2.35	96.41 ± 2.10	96.47 ± 2.12	97.21 ± 1.89
Kr-vs-Kp	95.11 ± 1.27	95.98 ± 1.07	96.13 ± 1.09	96.41 ± 1.02
Diabetes	76.16 ± 4.34	76.40 ± 4.45	76.50 ± 4.40	76.63 ± 4.36

was repeated 1000 times using randomly partitioned training, validation and testing subsets. To characterize the generalization performances of the tested ensemble methods, the accuracy reported is the average of the testing subset classification accuracy.

In addition to classification accuracy, this work also compares the robustness of the tested ensemble methods by introducing noise into the datasets (Demiriz *et al*., 2002; Windeatt, 2006). In particular, this work "injects" α% of noise into the datasets by randomly selected α% of the training and validation subset samples to alter their class memberships. In order to test the true generalization performance of the classification methods, the testing subset samples remain unchanged. The means of the testing accuracy for the original datasets and datasets with 5%, 10% and 15% of noise are summarized in Tables 2-5. The best classification results are highlighted (shown as shaded cells in the table) for every tested problem.

Based on the results of Tables 2 to 5, the classification accuracy of the proposed approach can be compared with that of the other three tested methods in a pairwise manner. Since there are thirteen problems, such an arrangement requires thirty nine comparisons for each noise level. These comparative results are summarized in Table 6. The white (black) boxes in Table 6 signify that the proposed approach has a higher (lower) classification accuracy than the competing method. From Table 6, for the noise free cases, the number of white and black boxes is 31 and 8, respectively. For simplicity, the result is expressed as 31W-8B. For the problems with 5%, 10% and 15% noise, Table 6 shows that the proposed approach results in 32W-7B, 32W-7B and 34W-5B, respectively. Summarizing results from all four tested noise levels, the proposed approach yields an overall result of 129W-27B.

To more rigorously compare the performance of

Table 3 Summary of the mean and standard deviation of the testing subset accuracy (5% noise)

Dataset	Testing accuracy % (Mean ± Standard deviation)			
	Averaging	Bagging	AdaBoost	Two stage
Sonar	75.28 ± 8.61	77.72 ± 8.37	77.10 ± 8.70	77.83 ± 8.37
Hepatitis	76.90 ± 2.10	81.46 ± 5.87	84.58 ± 7.09	84.68 ± 6.86
Horse-Colic	71.49 ± 10.60	88.24 ± 4.85	87.19 ± 5.89	88.21 ± 4.89
Heart-Statlog	81.24 ± 8.22	84.30 ± 6.71	84.00 ± 6.87	84.14 ± 6.81
Heart-C	83.05 ± 6.52	84.33 ± 6.17	84.03 ± 6.21	84.03 ± 6.35
Heart-H	94.55 ± 4.72	96.94 ± 3.21	96.42 ± 3.60	97.59 ± 2.76
House-Votes-84	93.08 ± 4.06	95.14 ± 3.03	94.70 ± 3.29	95.44 ± 3.04
German (Credit-g)	70.00 ± 0.11	71.21 ± 2.18	71.74 ± 2.38	74.93 ± 3.23
Credit-a	85.91 ± 4.04	86.51 ± 3.86	86.53 ± 3.86	86.52 ± 3.87
Australian	86.10 ± 3.86	86.57 ± 3.77	86.68 ± 3.75	86.68 ± 3.84
Breast cancer	95.48 ± 2.43	96.22 ± 2.18	95.81 ± 2.28	96.97 ± 1.97
Kr-vs-Kp	94.71 ± 1.32	95.60 ± 1.15	95.25 ± 1.21	96.04 ± 1.06
Diabetes	75.78 ± 4.41	76.63 ± 4.30	76.46 ± 4.23	76.47 ± 4.34

Table 4 Summary of the mean and standard deviation of the testing subset accuracy (10% noise)

Dataset	Testing accuracy % (Mean ± Standard deviation)			
	Averaging	Bagging	AdaBoost	Two stage
Sonar	73.79 ± 9.42	76.85 ± 8.78	76.14 ± 8.83	76.41 ± 8.51
Hepatitis	76.51 ± 0.53	77.42 ± 2.94	81.35 ± 6.49	81.16 ± 5.76
Horse-Colic	63.46 ± 2.38	80.03 ± 10.33	77.92 ± 11.59	87.74 ± 4.70
Heart-Statlog	66.38 ± 11.79	83.25 ± 6.96	81.20 ± 8.97	83.86 ± 6.86
Heart-C	81.86 ± 7.18	84.10 ± 6.20	83.64 ± 6.38	83.71 ± 6.33
Heart-H	91.13 ± 8.80	96.50 ± 3.52	95.21 ± 4.38	97.24 ± 3.05
House-Votes-84	92.36 ± 4.25	94.82 ± 3.23	93.83 ± 3.77	95.34 ± 3.04
German (Credit-g)	70.00 ± 0.00	70.00 ± 0.21	70.19 ± 0.93	70.52 ± 1.51
Credit-a	85.64 ± 4.02	86.26 ± 3.96	86.24 ± 3.96	86.17 ± 3.98
Australian	85.75 ± 3.87	86.33 ± 3.80	86.40 ± 3.79	86.29 ± 3.85
Breast cancer	94.96 ± 2.69	95.59 ± 2.39	95.05 ± 2.53	96.76 ± 2.05
Kr-vs-Kp	94.29 ± 1.34	95.13 ± 1.21	94.58 ± 1.34	95.77 ± 1.16
Diabetes	72.38 ± 5.43	76.10 ± 4.18	75.14 ± 4.11	76.33 ± 4.27

Table 5 Summary of the mean and standard deviation of the testing subset accuracy (15% noise)

Dataset	Testing accuracy % (Mean ± Standard deviation)			
	Averaging	Bagging	AdaBoost	Two stage
Sonar	69.80 ± 10.74	75.79 ± 8.79	74.56 ± 9.30	75.95 ± 8.92
Hepatitis	76.47 ± 0.00	76.52 ± 0.56	77.56 ± 3.68	77.23 ± 2.79
Horse-Colic	63.16 ± 0.00	65.59 ± 6.33	67.43 ± 8.87	81.79 ± 8.42
Heart-Statlog	56.60 ± 4.75	75.90 ± 10.98	73.06 ± 13.02	82.93 ± 6.74
Heart-C	71.49 ± 12.50	83.21 ± 6.49	81.90 ±7.98	83.30 ± 6.42
Heart-H	67.56 ± 9.35	93.66 ± 6.49	90.05 ± 10.34	96.45 ± 3.58
House-Votes-84	87.48 ± 10.20	94.27 ± 3.51	92.54 ± 4.24	95.09 ± 3.19
German (Credit-g)	70.00 ± 0.00	70.00 ± 0.00	69.997 ± 0.17	70.01 ± 0.12
Credit-a	85.30 ± 4.25	85.96 ± 4.01	85.94 ± 3.95	85.77 ± 3.97
Australian	85.04 ± 4.45	86.08 ± 3.86	86.02 ± 3.95	85.99 ± 3.89
Breast cancer	93.94 ± 2.86	94.79 ± 2.53	94.17 ± 2.63	96.49 ± 2.18
Kr-vs-Kp	93.81 ± 1.40	94.53 ± 1.32	93.89 ± 1.42	95.36 ± 1.21
Diabetes	66.45 ± 3.33	71.41 ± 4.95	70.67 ± 4.32	75.03 ± 4.16

Table 6 Summary of classification accuracy comparison results for the proposed approach

Datasets	Noise level											
	0%			5%			10%			15%		
	Methods: A (Averaging), B (Bagging), AB (AdaBoost)											
	A	B	AB	A	B	AB	A	B	AB	A	B	AB
Sonar		■	■					■				
Hepatitis			■						■			■
Horse-Colic		■			■							
Heart-Statlog		■			■							
Heart-C					■	■		■				
Heart-H			■									
House-Votes-84			■									
German (Credit-G)												
Credit-A		■				■		■	■		■	■
Australian						■		■	■		■	■
Breast cancer												
Kr-vs-Kp												
Diabetes					■							

the proposed approach with the other three ensemble methods, a one-sided t-test was conducted to compare classification results. In each of these comparisons, the approach is considered to have won (lost) the comparison if its mean testing accuracy is statistically significantly higher (lower) than that of the competing method. A p-value less than 0.01 is considered statistically significant, and a statistically insignificant result is considered to be a tie. To visualize the overall performance of the proposed approach, Table 7 summarizes its statistical test results by representing wins, losses and ties with white, black and grey boxes, respectively. From this table, for the noise free cases, the number of white, black and grey boxes is 21, 1 and 17, respectively. This result is reported as 21W-1L-17T signifying 21 wins, 1 loss and 17 ties. Similarly, Table 7 also shows that for the 5%, 10% and 15% noise level cases, the proposed approach yields 24W-0L-15T, 28W-0L-11T and 29W-0L-10T, respectively. The overall result for all four noise levels is 102W-1L-53T.

A comparison of the overall results of Table 6 (129W-27B) and Table 7 (102W-1L-53T) sheds some light on the performance of the proposed approach. First, both tables show that, among the four tested ensemble methods, the proposed approach has achieved the best overall classification results. Second, among the 129 white boxes of Table 6, where the proposed approach has lower classification error than the competing method, 102 of these results are statistically significant. In contrast, among the 27 black boxes of Table 6, where the proposed approach has lower classification accuracy than the competing method, only one of them achieves statistical significance.

Tables similar to Table 7 can be established for the other three tested methods. Although they are omitted for brevity, their overall results for the number of wins, losses and ties are summarized in Table 8, leading to the following observations:

1. First, the averaging method has the largest number of losses and least number of wins in all four tested noise levels. For the tested cases, it has not won any comparison. Consequently, despite its simplicity, an averaging neural ensemble is not recommended as a general solver for classification problems.
2. As addressed in Section II, the classification accuracy of bagging depends on the size of the training set. With the flexibility of adjusting the training set size for its bagging ensemble, the proposed approach is expected to perform at least as well as bagging. This expectation is confirmed from the results of Table 7 showing that, when compared with the proposed approach, bagging has not won any statistical comparison.
3. For the noise free datasets, next to the proposed approach, the AdaBoost has the second largest number of wins. However, as the noise level increases from 0%, 5%, 10% to 15%, the number of

Table 7 Summary of statistical significance test results for the proposed approach

Datasets	Noise level											
	0%			5%			10%			15%		
	Methods: A (Averaging), B (Bagging), AB (AdaBoost)											
	A	B	AB	A	B	AB	A	B	AB	A	B	AB
Sonar												
Hepatitis												
Horse-Colic												
Heart-Statlog												
Heart-C												
Heart-H												
House-Votes-84												
German (Credit-g)												
Credit-a												
Australian												
Breast cancer												
Kr-vs-Kp												
Diabetes												

Table 8 Summary of the numbers of wins, losses and ties

Noise level	Number of wins (W), losses (L) and ties (T)											
	Averaging			Bagging			AdaBoost			Two stage		
	W	L	T	W	L	T	W	L	T	W	L	T
0%	0	36	3	13	8	18	16	4	19	21	1	17
5%	0	39	0	18	7	14	15	11	13	24	0	15
10%	0	37	2	19	9	11	14	15	10	28	0	11
15%	0	34	5	20	10	9	12	17	10	29	0	10

wins achieved by the AdaBoost decreases from 16, 15, 14 to 12. In contrast, under the same conditions, the number of wins for bagging increases from 13, 18, 19 to 20. These results are in agreement with previous findings that show that bagging is more robust to noise than AdaBoost.

4. In similarity to bagging, the number of wins of the proposed approach improves as the noise level increases. The proposed approach employs bagging ensemble at the second stage of its classification process and thus preserves bagging's robustness with respect to noise. The results of Table 7 show that, for the tested problems, the proposed approach is even more robust than bagging. Specifically, as the noise level increases from 0%, 5%, 10% to 15%, when compared against bagging, the statistical comparison results of the proposed approach improves from 5W-0L-8T, 5W-0L-8T, 7W-0L-6T to 8W-0L-5T.
5. Finally, Table 9 is given to demonstrate the consistency of the proposed approach. By defining the error margin as the difference between the smallest classification error obtained by the tested methods and the classifier under investigation, Table 9 summarizes the largest error margins (LEM) for all the tested methods and for all four tested noise levels. In essence, LEM represents the worst performance of a classifier in solving the thirteen benchmark problems. As shown by Table 9, the two-stage method has the smallest LEM for all four tested noise levels. In particular, for the noise free case, LEMs of the conventional methods are at least twice as large as those of the proposed approach. Moreover, for the 15% noise level case, LEMs of earlier classification methods are at least 43.5 larger than those of the two-stage method.

Table 9 Summary of the largest error margin (LEM) for the tested classification methods

Noise level	The largest error margin among 13 benchmark results			
	Averaging	Bagging	AdaBoost	Two stage
0%	5.99	1.70	1.66	0.72
5%	16.75	3.72	3.19	0.30
10%	24.28	7.71	9.82	0.44
15%	28.89	16.2	14.36	0.33

In summary, the proposed approach compares very favorably with the other three tested methods in terms of accuracy, robustness and consistency.

V. CONCLUSION

By adopting a neural network as the base classifier, this paper introduced a two-stage committee machine to solve classification problems. The first stage of the proposed method was the conventional averaging ensemble. The second stage employed a bagging ensemble to focus on hard-to-classify samples. In developing this classification method, two key problems needed to be resolved. First, a criterion was needed to determine which samples are more difficult to classify than the others. Based on the idea of Bayes' rule and by utilizing the property that a neural network can be trained to approximate the posterior probabilities, this work employed the absolute value of the neural network output difference as a measure for the degree-of-classification-difficulty.

The second key problem was how to determine the optimal size of the hard-to-classify training subset for the bagging ensemble so that the classification accuracy can be maximized. The importance of this problem lies in the fact that the performance of a bagging ensemble depends heavily on the size of its training set. Since an analytical solution does not seem to be feasible, at the expense of computational cost, this problem is resolved by incrementally expanding the size of the hard-to-classify training subset. The subset that yields the highest classification accuracy is selected for the final design. Considering the heavy computational cost required by this incremental procedure, one of the lines of promising future work is to develop a more efficient approach to determine the optimal size of the hard-to-classify training subset.

Finally, a series of experiments were conducted to compare the proposed approach with other well known committee machine design methods including the averaging rule, bagging and AdaBoost. The experimental results validate the accuracy and the robustness of the proposed approach.

ACKNOWLEDGMENTS

This research was supported in part by the National Science Council of the Republic of China under Grant Number NSC 92-2320-B-039-0292.

NOMENCLATURE

a	Threshold value for the output of the two-stage committee machine
a^*	Optimal value of a
C_k	kth class
d_j	jth output of the neural network
$e(\boldsymbol{x})$	Neural network training error associated with input $\boldsymbol{x}$
$f(\boldsymbol{x}, T)$	Output of the neural network when the training set is T and input vector is $\boldsymbol{x}$
f_{AB}	Output of the AdaBoost committee machine
f_B	Output of the bagging committee machine
F_j	Dataset that is employed to train the jth component classifier of the AdaBoost committee machine
G_k	Dataset that is used to train the kth component classifier of the Bagging committee machine
M	Total number of classes
N	Total number of data
$P(C_k\|\boldsymbol{x})$	Posterior probability of $\boldsymbol{x}$ in association with class C_k
T	Training set
w_j	Weighting coefficient for the jth component classifier of the committee machine
$\boldsymbol{x}_i$	Input vector of the ith sample
$\boldsymbol{y}_i$	Output vector of the ith sample
Z_1	Training subset that is used to train the averaging ensemble for the two-stage committee machine
Z_2	Training subset that is used to train the bagging ensemble for the two-stage committee machine

REFERENCES

Alexandre, L. A., Campilho, A. C., and Kamel, M., 2001, "On Combining Classifiers Using Sum and

Product Rules," *Pattern Recognition Letters*, Vol. 22, No. 12, pp. 1283-1289.

Anand, R., Mehrotra, K., Mohan, C. K., and Ranka, S., 1995, "Efficient Classification for Multiclass Problems Using Modular Neural Networks," *IEEE Transactions on Neural Networks*, Vol. 6, No. 1, pp. 117-124.

Bauer, E., and Kohavi, R., 1999, "An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting and Variants," *Machine Learning*, Vol. 36, No. 1-2, pp. 105-139.

Bishop, C. M., 1995, *Neural Networks for Pattern Recognition*, Oxford University Press, NY, USA.

Blake, C. L., and Merz., C. J., 1998, *UCI Repository of Machine Learning Databases*, University of California, Department of Information and Computer Science, Irvine, CA, Available: http://www.ics.uci.edu/~mlearn/MLRepository.html.

Breiman, L., 1996, "Bagging Predictors," *Machine Learning*, Vol. 24, No. 2, pp. 123-140.

Breiman, L., 1998, "Arcing Classifiers," *The Annals of Statistics*, Vol. 26, No. 3, pp. 801-824.

Chan, C. W., Huang, C., and DeFries, R., 2001, "Enhanced Algorithm Performance for Land Cover Classification from Remotely Sensed Data Using Bagging and Boosting," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 39, No. 3, pp. 693-695.

Demiriz, A., Bennett, K. P., and Shawe-Taylor, J., 2002, "Linear Programmingboosting Via Column Generation," *Machine Learning*, Vol. 46, No. 1-3, pp. 225-254.

Dietterich, T. G., 2000, "An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization," *Machine Learning*, Vol. 40, No. 2, pp. 139-157.

Efron, B., and Tibshirani, R., 1993, *An Introduction to Bootstrap*, Chapman and Hall, London, UK.

Freund, Y., and Schapire, R., 1997, "A Decision-theoretic Generalization of On-line Learning and an Application to Boosting," *Journal of Computer and System Sciences*, Vol. 55, No. 1, pp. 119-139.

Fumera, G., and Roli, F., 2004, "Analysis of Error-reject Trade-off in Linear Combined Multiple Classifiers," *Pattern Recognition*, Vol. 37, No. 6, pp. 1245-1265.

Hashem, S., 1997, "Optimal Linear Combinations of Neural Networks," *Neural Networks*, Vol. 10, No. 4, pp. 599-614.

Haykin, S., 1999, *Neural Network–A Comprehensive Foundation*, 2nd ed., Prentice-Hall, Upper Saddle River, NJ, USA.

Knerr, S., Personnaz, L., and Dreyfus, G., 1990, "Single-layer Learning Revisited: A Stepwise Procedure for Building and Training a Neural Network," *Neurocomputing: Algorithms, Architectures and Applications*, F. Fogelman Soulié and J. Hérault ed., Springer-Verlag, Vol. F68 of NATO ASI Series., pp. 41-50.

Kuncheva, L. I., 2002, "A Theatrical Study on Six Classifier Fusion Strategies," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 2, pp. 281-286.

Maclin, R., and Opitz, D., 1997, "An Empirical Evaluation of Bagging and Boosting," *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, Providence, USA, pp. 546-551.

Lam, L., and Suen, C. Y., 1997, "Application of Majority Voting to Pattern Recognition: an Analysis of Its Behavior and Performance," *IEEE Transactions on Systems, Man, and Cybernetics–Part A: Systems and Humans*, Vol. 27, No. 5, pp. 553-568.

Opitz, D. W., and Maclin, R. F., 1997, "An Empirical Evaluation of Bagging and Boosting for Artificial Neural Networks," *IEEE International Conference on Neural Networks*, Huston, USA.

Opitz, D., and Maclin, R., 1999, "Popular Ensemble Methods: An Empirical Study," *Journal of Artificial Intelligence Research*, Vol. 11, No. 1, pp. 169-198.

Schapire, R. E., 2002, "The Boosting Approach to Machine Learning: An Overview," *MSRI Workshop on Nonlinear Estimation and Classification*, AT&T Labs, Florham Park, NJ, USA.

Tax, D. M. J., Breukelen, M. V., Duin, R. P. W., and Kittler, J., 2000, "Combining Multiple Classifiers by Averaging or Multiplying?," *Pattern Recognition*, Vol. 33, No. 9, pp. 1475-1485.

Ueda, N., 2000, "Optimal Linear Combination of Neural Networks for Improving Classification Performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 2, pp. 207-215.

Windeatt, T., 2003, "Vote Counting Measures for Ensemble Classifiers," *Pattern Recognition*, Vol. 36, No. 12, pp. 2743-2756.

Windeatt, T., 2006, "Accuracy/diversity and Ensemble MLP Classifier Design," *IEEE Transactions on Neural Networks*, Vol. 17, No. 5, pp. 1194-1211.

Manuscript Received: Apr. 27, 2007
Revision Received: June 30, 2008
and Accepted: July 30, 2008