System of Boosting Voting with Multiple Learning Algorithms

[1]Chien-Hsing Chen, [2]Chung-Chian Hsu
[1]Department of Information Management
Hwa Hsia Institute of Technology
Taiwan
ktfive@gmail.com

[2]Department of Information Management
National Yunlin University of Science and Technology
Taiwan
hsucc@yuntech.edu.tw

Journal of Digital Information Management

ABSTRACT: *The Ensemble classifier has been an active research topic in the area of machine learning. In a classification task, the ensemble scheme determines a final class label from several individual results which are usually generated by several individual classifiers according to two major principles. The first is to use multiple learning algorithms to form the set of individual outcomes, and the second is to train a set of data fractions, which are generated from a given training dataset, with a weak learning algorithm to generate the set of results. The advantage of the first principle is on classification stability, whereas that of the second principle is on the accuracy improvement in terms of classification accuracy. Thus far, most studies in the literature have been based on either a study of the two principles of ensemble. In this paper we propose a new ensemble scheme to combine the two principles simultaneously. We evaluate the performance on a classification task. Experimental results on several UCI benchmark datasets demonstrate that the proposed framework achieves improved performance in terms of classification accuracy compared to the conventional approaches.*

Categories and Subject Descriptors
E. 1 [Data Structures]: I.1.2 [Algorithms]

General Terms: Learning algorithms, Data distribution, Machine learning

Keywords: Classification, Ensemble classifier, Majority voting, Boosting voting

Received: 12 September 2009, Revised 11 January 2010, Accepted 19 January 2010

1. Introduction

An aggregating ensemble classifier has been a well-accepted method to improve accurate performance or stability in the area of machine learning, since there has been no individual approach always yielding the best performance. Previous studies have shown that the ensemble classifier is successful in the classification task, with consistent and stable results, as well as with improved performance in terms of classification accuracy. The ensemble classifier has been studied in many research areas including pattern recognition, information retrieval, data mining, and machine learning. The reason behind the usage of the ensemble classifier is its simplicity and performance on real data (Narasimhamurthy 2005).

In a classification task, the ensemble scheme determines a final class label from a set of individual results which are usually generated from a set of individual classifiers corresponding to two major principles. The first is to use multiple learning algorithms to form the set of individual results, where each participative learning algorithm contributes independently a result (opinion, vote or suggestion). The set of results (usually odd numbers) are then to be aggregated into a final class label by the use of a majority voting scheme which is commonly employed in many areas of research, such as the research in (Lam and Suen 1997; Kuncheva 2002; Narasimhamurthy 2005). The second is to train a set of data fractions, which are all generated from a given training dataset, on a weak learning algorithm to generate several classifiers which can contribute the set of opinions. Such learning ensemble schemes like bagging (Breiman 1996), and boosting (Freund and Schapire 1996; Sun, Wang et al. 2006), are the paradigms inspiring numerous extensions of ensemble. Mainly, they use different strategies to form variant data fractions: bagging uses bootstrapping strategy, whereas boosting does sampling by observing data quality in the prior training phases.

Accordingly, the first principle of the ensemble scheme focuses on adopting multiple learning algorithms, while the other one of that focuses on generating variant data fractions trained on a weak learning algorithm. The advantage of the first principle is on stability (Lam and Suen 1997; Kuncheva 2002; Kuncheva and Whitaker 2003; Chen and Zhou 2005; Narasimhamurthy 2005), whereas that of the second principle is on accuracy improvement in terms of classification accuracy (Freund and Schapire 1995; Bauer and Kohavi 1999; Dietterich 2000; Sun, Wang et al. 2006).

However, most studies in the literature were based on either one of the two principles of the ensemble scheme. One may want to know whether there is an ensemble scheme can simultaneously preserve the property of the two principles. This led us to combine the two principles simultaneously and to present a new ensemble scheme concerning the use of the multiple learning algorithms and the generation of variant data fractions at the same time.

In this paper we present a new ensemble framework, namely boosting voting scheme, which is to perform an ensemble scheme integrating the set of the results of multiple learning algorithms with respect to variant data distributions. The framework attempts to achieve simultaneously stability and performance in terms of classification accuracy. The motivation is stimulated in that when a predicted result (class label) of a majority voting of a data instance is incorrect (the class label is aggregated from several classification learning algorithms), the data instance needs to be trained more often. In other words,

the data instance has to be learnt more frequently when it is not easy to be classified by the use of the majority voting scheme. The proposed framework differs from previous work that was based either on a majority voting scheme with multiple learning algorithms which has been used to improve classification stability or on only a weak learning algorithm as is necessary in a boosting and a bagging schemes to improve classification accuracy. In particular, the proposed framework uses a generation strategy of variant data fractions in boosting, because the boosting scheme gave more efforts on data quality during training (Freund and Schapire 1996; Sun, Wang et al. 2006).

This paper is organized as follows. Following the introduction (Section 1), Section 2 illustrates related work. Section 3 describes the detail of the proposed method. Section 4 describes the experiments performed on a set of UCI benchmark datasets. The conclusion is given in Section 5.

2. Related Work

An ensemble classifier has been useful in improving performance in terms of classification accuracy. Kuncheva mentioned that there are two principal tasks in an ensemble classifier (Kuncheva 2002). The first is to determine learning algorithms for the sake of producing classifiers, and the second is to use an ensemble fusion method. He argued that the outputs of a set (pool, committee, ensemble, or team) of the classifiers could be combined together to achieve higher accuracy than that of the best classifier. The author studied six fusion strategies, which were minimum, maximum, average, median, majority vote and oracle, to select a winner vote from a set of opinions. Recently, Kuncheva et al. derived upper and lower limits on the majority voting accuracy (Kuncheva, Whitaker et al. 2003). Their experimental results have demonstrated that the majority voting scheme with dependent classifiers potentially achieves a better performance than that with independent classifiers and an individual classifier. Narasimhamurthy formulated the majority voting problem as an optimization problem with linear constraints via the theoretical lower and upper bounds in a binary classification problem (Narasimhamurthy 2005). He also analyzed the experiments tested on a majority voting scheme for an odd number of classifiers and for an even number of ones. Freund and Schapire applied the multiplicative weighting technique to derive a new ensemble scheme which does not require any prior knowledge about the performance of the weak learning algorithm (Freund and Schapire 1995). Martinez-Munoz and Suarez argued the ensemble size is a major factor of driving the error rate during training (Martinez-Munoz and Suarez 2005). To use an ensemble scheme with different strategies is still an open issue.

The use of an ensemble scheme was usually based on two major principles. The first principle of ensemble classifier was to use multiple learning algorithms. It often focused on defining the weights for the combination of multiple classifiers or selecting multiple appropriate classifiers assumed independent from each other. For example, five methods which were Bayesian statistics, multiple linear regression, decision tree, neural network and support vector machine were got together for the use of classification in the research (Chen and Zhou 2005). The research in (Sinha and Huimin 2008) also integrated six methods, which were naive Bayes, logistic regression, decision tree, decision table, back-propagation neural network, k-nearest neighbor, and support vector machine, to solve classification problems. Other examples of ensemble classifier could be found in (Hansen and Salamon 1990; Wang, Fan et al. 2003; Won and Cho 2003; Plewczynski, Spieser et al. 2006; Banfield, Hall et al. 2007; derWalt and Barnard November 2006).

Another principle of ensemble classifier was to train a set of the data fractions, which were all generated from a given training dataset, on a weak learning algorithm to outcome numerous classifiers. Two schemes, which were a bagging and a boosting schemes, were the paradigms inspiring numerous extensions and have been applied in many research areas. First, a bagging scheme worked for combining multiple independent classifiers to yield a prediction class label via integrating their corresponding votes (Breiman 1996). It played a bootstrapping strategy to generate a set of training data fractions, in which the data distribution during training process seemed similar to the original distribution (Rodriguez and Kuncheva 2006). Each generated training data fraction was learnt by a learning algorithm, so that there was a set of classifiers, each of which contributed a vote with regards to its weight performed from the training performance. Second, a booting scheme was similar to bagging except in that the role of generation of training data fractions differed from each other. In boosting, each training data fraction was drawn with respect to the data distribution which was considered about the training data quality; thus the data distribution of a new generation was differed greatly from the old one. The literature (Freund and Schapire 1996; Elkan 1997; Dietterich 2000; Schwenk and Bengio 2000; Sun, Wang et al. 2006; Sun, Wang et al. 2006) reports experiments by using a boosting strategy, in which AdaBoost algorithm was much more popular and more successful with many applications, as mentioned in (Sun, Wang et al. 2006). Due to bagging and boosting, all of which were similar in that they provided a valuable ensemble framework based on only a learning algorithm with respect to variant data fractions, they have been used to improve performance in terms of classification accuracy in many applications.

Accordingly, the first principle of the ensemble scheme focuses on adopting multiple learning algorithms, while the other one focuses on generating variant data fractions for training a weak learning algorithm. However, in the literature reported above, there is no work addressing the relationship between multiple learning algorithms and multiple data fractions. Similarly, most studies in the literature focused on either a study of the multiple learning algorithms or the use of multiple data fractions drawn based on training result of a weak learning algorithm. This led us to combine the two principles simultaneously and to present a new ensemble scheme, namely boosting voting scheme, concerning the use of the multiple learning algorithms and the generation of variant data fractions at the same time.

3. Methodology

We present a new ensemble framework, namely boosting voting scheme, which is to perform an ensemble scheme integrating a set of the results of multiple learning algorithms with respect to variant data distributions. If a predicted result (class label) of a majority voting of a data instance is incorrect (the class label is aggregated from several classification learning algorithms), the data instance needs to be trained more often. In other words, the data instance has to be learnt more frequently when it is not easy to be classified by the use of the majority voting scheme.

We first give a definition of majority voting learning scheme and illustrate a running example to determine a class label with majority voting scheme. Then, we devise a framework to train a set of data fractions, which are all generated from a given

training dataset with respect to the results of the majority voting scheme, on a weak learning algorithm to generate classifiers.

3.1 Definition for Majority Voting Learning

Let X be a dataset containing a set of n data instances and let x_j be an instance which corresponds to its class label $y_j \in Y$. Denote $M = \{m_1, \ldots, m_l\}$ to be a set of classifiers, where m_i is the ith classifier that has a suggestion function $v_i(\cdot)$. An instance x_j has $|M|$ suggestions by the use of the set of classifiers M which forms a tuple of dimension $|M|$. An example, as reported in Table 1, illustrates that each instance has three suggestions, each of which is the result of a classifier and assumed as either "Y" or "N".

Instance	m_1	m_2	m_3
x_{19}	Y	Y	N
x_{23}	N	N	Y
x_{32}	Y	N	N

Table 1. Three tuples are generated by the use of $M = \{m_1, m_2, m_3\}$

Since a majority voting scheme with multiple learning algorithms has been a well-known integrated voting approach to generate a final class label, it is able to result a winner vote for x_j. The winner vote is chosen by using the majority voting scheme with M. The formula which is to generate a winner vote is written as

$$h_1(x_j) = \arg\max_{y \in Y} \sum_{i=1}^{|M|} \delta(v_i(x_j) = y) \qquad (1)$$

where $v_i(x_j)$ is a suggestion of the ith learning algorithm for a data instance x_j. The function $\delta()$ is a boolean function returning 1 while its argument is evaluated true and 0, otherwise. Hence, based on the example shown in Table 1, the results of these three instances as shown in Table 2 are obtained by the use of the formula (1).

Instance	$h_1(.)$
x_{19}	Y
x_{23}	N
x_{32}	N

Table 2. Results of the three instances by the use of the majority voting scheme with $M = \{m_1, m_2, m_3\}$

3.2 Boosting Voting Scheme

In this paper, we combine the principles of a majority voting scheme with multiple learning algorithms and a boosting algorithm (Freund and Schapire 1996; Elkan 1997; Schapire and Singer 1999; Schwenk and Bengio 2000) and then propose a new ensemble framework, namely boosting voting scheme, to achieve a classification task. The proposed framework differs from previous work which focuses on either a study of a majority voting scheme with multiple learning algorithms or a boosting algorithm. In other words, instead of multiple learning algorithms $m_1, m_2, \ldots, m_l$ used in a majority voting scheme or only a weak learning algorithm trained on multiple data fractions $X_1, X_2, \ldots, X_T$ which are generated from a training dataset X in the boosting algorithm, we perform a procedure allowing numerous iterations, which correspond to multiple data fractions and involve multiple learning algorithms to produce a winner vote.

The method of generating multiple data fractions can refer to the boosting algorithm (Freund and Schapire 1996); that is, a training data instance is drawn with respect to the previous

Input:

$\{x_j\}_{j=1}^{n} \in X$ with its class label $\{y_j\}_{j=1}^{n} \in Y$ is given

let $\{D_j(t)\}_{j=1}^{n} = 1/n$ for $t = 1$

the number of rounds T is assigned

set the number of learning algorithms

learning algorithm $m_i \in M$ for all i is determined

Do:

for $t = 1, 2, \ldots, T$ {

bootstrapping a data fraction X_t with respect to $\{D_j(t)\}_{j=1}^{n}$

for $\{x_j\}_{j=1}^{n}$ {

for $\{m_i\}_{i=1}^{|M|} \in M$ {

run $v_{i,t}(x_j)$

}

}

calculate $\{u_{i,t}\}_{i=1}^{|M|}$ in (3)

execute $h_t(x_j)$ using (2)

calculate w_t in (4)

update $D_j(t+1)$ in (7)

}

Output:

execute $H_{fin}(x_j)$ using (5)

Figure 1. The boosting voting algorithm

training quality. In other words, a data instance of which class label predicted by the majority voting scheme is incorrect will have a higher probability to be drawn and may be included in the training task at the next step. The algorithm of the framework proposed in this paper is written as follows (to see Fig. 1).

A majority voting scheme is a robust approach in hypothesizing a class label for a data instance. We extend the formula (1) to have repeated learning rounds to produce further hypotheses $h_t(x_j)$ instead of only one $h_1(x_j)$, where $h_t(.)$ represents the hypothesis for t rounds and $h_1(.)$ represents the single hypothesis. Therefore, a T-dimensional vector for the hypotheses is made for each of the data instances. Moreover, the learning property for the participative method m_i at the tth round can be studied. The formula (1) is then rewritten as follows:

$$h_t(x_j) = \arg\max_{y \in Y} \sum_{i=1}^{|M|} u_{i,t} \delta(v_{i,t}(x_j) = y) \qquad (2)$$

where $v_{i,t}$ is a suggestion function for the m_i approach at the tth round and $u_{i,t}$ is to measure the capability of m_i tested on a data fraction X_t at the tth round defined as follows:

$$u_{i,t} = \log \frac{1 - \tau_{i,t}}{\tau_{i,t}} \qquad (3)$$

where $\tau_{i,t}$ is the probability of a training error with the approach m_i at t^{th} round.

We can obtain the results by the use of formula (2). The results can be illustrated in Table 3. It differs from Table 2 in that it allows repeated learning rounds (the number of rounds in the example is set to 4).

Instance	$h_1(.)$	$h_2(.)$	$h_3(.)$	$h_4(.)$
x_{19}	Y	Y	N	Y
x_{23}	N	N	N	Y
x_{32}	N	Y	Y	Y

Table 3. Results of the three instances by the use of the $h(.)$ formula

For $h_t(.)$, $t = 1, 2, \ldots, T$, they can have different voting suggestions with respect to a set of data fractions, and have different confidences for a total of T rounds. Therefore, a prediction confidence w_t for a learning round t is written as follows:

$$w_t = \log \frac{1-\varepsilon_t}{\varepsilon_t} \tag{4}$$

where ε_t is the rate of training errors at the tth round. In other words, the value w_t is large, if the amount of the prediction class label for the training data instances at the tth round is large.

Different from previous work in the boosting scheme, our method performs multiple learning algorithms M to output the results for all iterations T and then integrates multiple voting hypotheses $\{h_t(x_j)\}_{t=1}^{T}$ to form a final suggestion $H_{fin}(x_j)$ for a data instance x_j, where T is an integer representing the total rounds in the procedure. The function of the final suggestion $H_{fin}()$ is defined as

$$H_{fin}(x_j) = \arg\max_{y \in Y} \sum_{t|\varepsilon_t \le \theta}^{T} w_t \times \delta(h_t(x_j) = y) \tag{5}$$

where θ is a constant representing an incorrect value of random suggestions. In other words, the tth round which error rate ε_t is not smaller than the threshold θ will not participate in the final suggestion. The threshold θ is formulated as

$$\theta = 1 - \frac{1}{|y|} \tag{6}$$

We can obtain the example which is from Table 3 by using formula (5). The results can be illustrated in Table 4 to show the final winner suggestion.

Instance	$H_{fin}(.)$
x_{19}	Y
x_{23}	N
x_{32}	Y

Table 4. Results of the three instances by the use of the $H_{fin}(.)$ formula

The training sample X_t at the t round is evolved from X_{t-1} by the bootstrapping strategy with respect to the learning property in the training process. It is worth mentioning that an instance has to be learnt more frequently when it is not easy to be classified. Therefore, an instance x_j will appear much more possible in X_t while gaining the wrong prediction class label in X_{t-1}. Similarly, an instance may be learned at the t +1 round when the result of $h_t(.)$ is incorrect. On the other hand, x_j contributing the correct class label in X_{t-1} may not appear in X_t. That is to say, the result of $h_t(x_j)$ which is wrong will increase the probability $D_j(t+1)$ as would be drawn at round $t+1$.

$$D_j\ (t+1) = \frac{D_j(t)}{Z_t} \times \begin{cases} 1 & \text{if } h_t(x_j) \neq y_i \\ \dfrac{\varepsilon_t}{1-\varepsilon_t} & \text{else} \end{cases} \tag{7}$$

where Z_t is a normalization value so as to satisfy $\{D_j(t)\}_{j=1}^{n} = 1$ for each t round.

4. Experiments

4.1 Experimental Dataset

We tested twelve datasets in the experiment. The experimental datasets are selected from a well-known UCI machine learning repository (UCI). The description of the datasets including dataset name, numbers of attributes and numbers of classes is listed in Table 5.

Dataset Name	Numbers of Attributes	Numbers of Classes
Bench	60	2
Chess	36	2
Glass	9	6
Hepatitis	19	2
HillValley	100	2
Iris	4	3
Mannorgraphic	5	2
Statlog	18	4
StatlogGaman	24	2
Wine	13	3
Yeast	8	10
Zoo	16	7

Table 5. Description of a set of UCI Benchmark Datasets

4.2 Experimental Setting

We test performance on a classification task by the use of ensemble schemes. A total of four ensemble schemes, namely boosting voting, boosting, bagging and majority voting, are employed to test performance for the twelve UCI benchmark datasets. The boosting voting scheme proposed in this paper is performed by the use of the procedure mentioned in Figure 1. The majority voting scheme with multiple learning algorithms we used follows equation (1) which assumes that the participative classifiers contribute their votes with equal weights. Both they have to determine a set of M classifiers as mentioned in Section 3.1. The K-nearest neighbor (KNN) algorithm is able to produce multiple classifiers by setting a different parameter value K. In practice, we use three classifiers based on the KNN algorithm with a different parameter K which includes 3, 5, and 7, respectively. In other words, the boosting voting scheme and the majority voting scheme are to use three classifiers (3NN, 5NN, and 7NN algorithm) together to perform the performances on the test of the twelve datasets.

Another two schemes are a majority voting scheme and a boosting majority scheme. They both need a single weak learning algorithm to do the classification task. Therefore, we use the KNN algorithm with K set to 5 for performance comparison with the other ensemble schemes. K is set to 5 because of its simplicity.

We run each ensemble scheme 10 times. For each time, we divide randomly the data instances into two parts: two-third as the training dataset and one-third as the testing dataset for the classification task in the experiment. For each time, we set the number of rounds of the four ensemble schemes to 10 (A literature in (Sun, Wang et al. 2006) set the boosting scheme for 10 generations) for the classification task.

We use the Euclidean formula to measure the distance between data instances for the use of KNN algorithm. The categorical attributes are measured as the range of $\{0, 1\}$ and the numerical attributes are measured as the range of $[0, 1]$.

4.3 Performance Comparison

An accuracy measure is applied to evaluate the performance. The performance comparison for the four ensemble classifiers is shown in Table 6. In Table 6, the first column describes the

dataset names, and the other columns are the performances of accuracy rate by different ensemble schemes.

Dataset Name	Boosting Voting	boosting	bagging	Majority Voting
Bench	0.50	0.47	0.47	0.45
Chess	0.75	0.72	0.70	0.69
Glass	0.63	0.64	0.56	0.56
Hepatitis	0.64	0.61	0.63	0.61
HillValley	0.50	0.49	0.48	0.46
Iris	0.87	0.86	0.87	0.88
Mannorgraphic	0.69	0.66	0.68	0.68
Statlog	0.59	0.56	0.56	0.54
StatlogGaman	0.66	0.68	0.67	0.67
Wine	0.93	0.89	0.89	0.90
Yeast	0.50	0.49	0.48	0.49
Zoo	0.88	0.87	0.86	0.84

Table 6. Accuracy Rates on twelve UCI benchmark datasets

The experimental results demonstrate that the boosting voting scheme is superior to the others in performing the classification task. The experiments show that the boosting voting scheme gains the best performance than the others in a total of nine datasets, which are “Bench”, “Chess”, “Hepatitis”, “HillValley”, “Mannorgraphic”, “Statlog”, “Wine”, Yeast”, and “Zoo”. The boosting scheme gains the best performance in two datasets, while the bagging and the majority voting gains best performance in one dataset.

We further illustrate the performance by the use of average accuracy measure and depict graphically the accuracy rate (accuracy rate in average) and variance by the boxplot analysis for the 10 times. From this analysis, we can observe not only the average accuracy but also the stability of classification revealed by the variance. The shorter box width represents that the scheme is more stable in terms of classification accuracy.

From the observation, the boosting voting scheme is able to achieve simultaneously the stability of classification and accuracy improvement. The majority voting scheme with three learning algorithms potentially achieve the stability on several datasets; however, it is not able to improve the accuracy rate. In contrast, the boosting scheme is able to improve the accuracy rate on several datasets; however, it has no stability in terms of classification accuracy.

In summary, the experimental results indicate that the boosting voting scheme outperforms the other three ensemble schemes in testing on a classification task. The research as reported in the literature has mentioned that a majority voting scheme with multiple learning algorithms was a powerful scheme in integrating the votes from several classifiers and was a well-accepted method to achieve excellent performance. Both the bagging scheme and the boosting scheme are well-known techniques to improve performance in terms of classification accuracy.

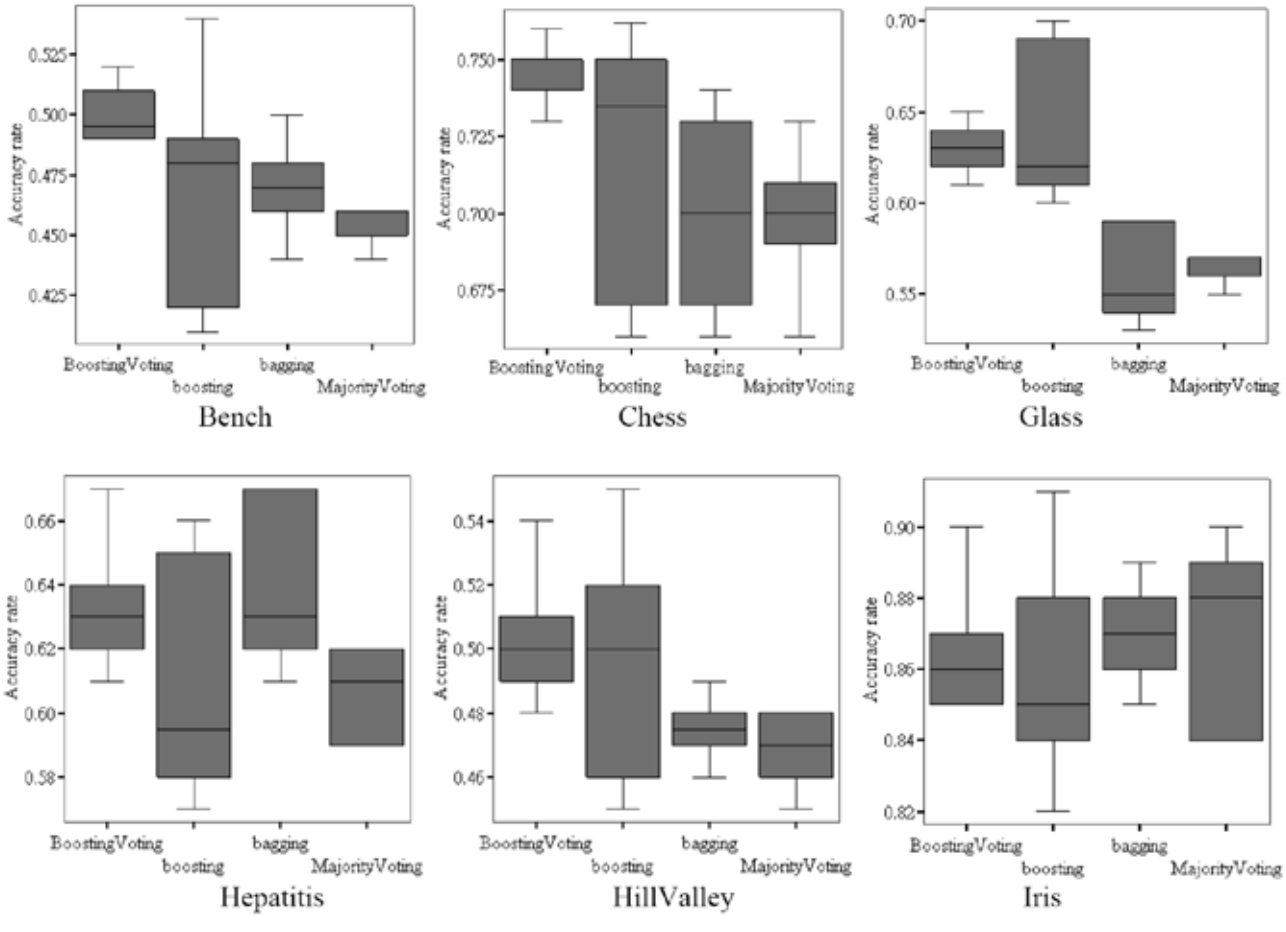

Figure 2. Boxplot analysis for the four ensemble schemes on the twelve UCI benchmark datasets

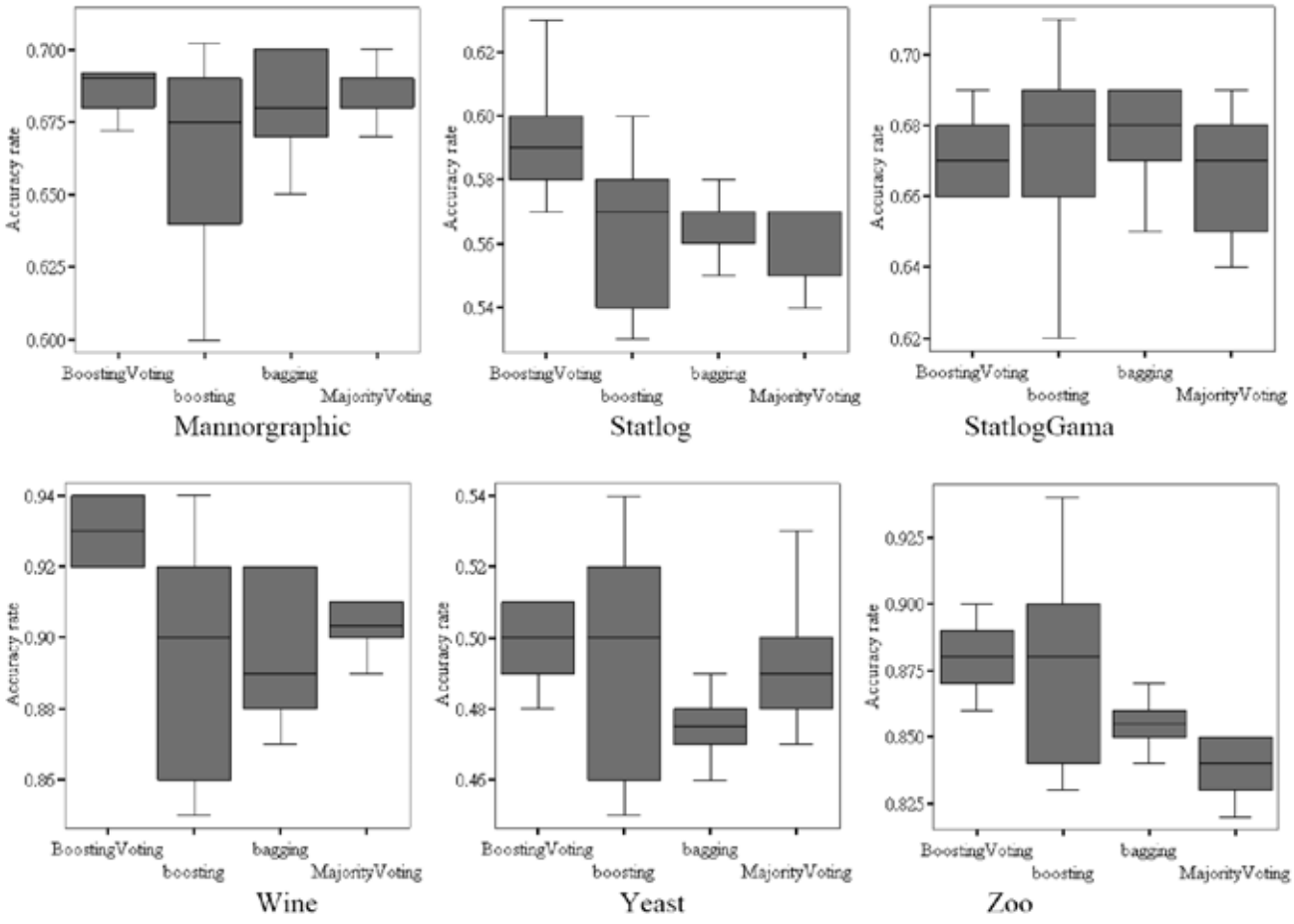

Figure 2. Boxplot analysis for the four ensemble schemes on the twelve UCI benchmark datasets—*Cont'd*

Importantly, the experiments demonstrate that the boosting voting scheme proposed in this paper is valuable in the classification task and show better and more stable performance than the majority voting scheme, the boosting scheme and the bagging scheme.

5. Conclusion

The ensemble classifier is to choose a winner vote from several opinions and is usually based on two major principles. The first focuses on the use of multiple learning algorithms and the second focuses on generating various data fractions for training a learning algorithm. The advantage of the first principle is on the stability (Lam and Suen 1997; Kuncheva 2002; Kuncheva and Whitaker 2003; Chen and Zhou 2005; Narasimhamurthy 2005), whereas that of the second principle is on the accuracy improvement in terms of classification accuracy (Freund and Schapire 1995; Bauer and Kohavi 1999; Dietterich 2000; Sun, Wang et al. 2006). However, most studies in the literature were only based on either one of the two principles. This led us to combine the two principles simultaneously and to present a new ensemble scheme concerning the use of the multiple learning algorithms and the generation of variant data fractions at the same time. In this paper we present a new ensemble scheme, namely boosting voting, allowing a set of learning algorithms with various data fractions, instead of only a majority voting scheme with multiple learning algorithms which has been commonly employed for improving prediction stability or a weak classification algorithm as is necessary in boosting scheme. The principal idea is that a data instance has to be learnt more frequently when it is not easy to be classified by the use of the majority voting scheme. From the experiment, we can find that the boosting voting scheme proposed in the paper achieves stability and accuracy improvement tested on several UCI benchmark datasets. In the future, we are interested in applying the boosting voting scheme for the analysis on various practical datasets.

6. Acknowledgement

This work is supported by National Science Council, Taiwan under grant NSC 98-2416-H-224-010-MY2.

References

[1] Banfield, R. E., Hall, L. O., Bowyer, K. W., Kegelmeyer, W. P. (2007). A Comparison of Decision Tree Ensemble CreationTechniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (1) 173-180.

[2] Bauer, E., Kohavi, R. (1999). An empirical comparison of voting classification algorithms: *Bagging, boosting, and variants. Machine Learning* 36 (1/2) 105-139.

[3] Breiman, L. (1996). Bagging Predictors. *Machine Learning* 24, 123-140.

[4] Chen, H., Zhou, H.-X. (2005). Prediction of solvent accessibility and sites of deleterious mutations from protein sequence. *Nucleic Acids Research* 33 (10) 3193-3199.

[5] Walt, C. M., van der., Barnard, E. (2006). Data characteristics that determine classifier performance. *In:* Proc. of the 6th Annual Symposium of the Pattern Recognition Association of South Africa. Parys, South Africa, November.

[6] Dietterich, T. G. (2000). An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees:

Bagging, Boosting, and Randomization. Machine Learning 40 (2) 139-157.

[7] Elkan, C. (1997). Boosting and Naive Bayesian learning. *Technical Report CS97-557*, Dept. Computer Science and Eng., Univ. of California, San Diego.

[8] Freund, Y., Schapire, R. E. (1995). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *In:* Proc. of the 2th European Conf. Computational Learning Theory.

[9] Freund, Y., Schapire, R. E. (1996). Experiments with a new Boosting algorithm. *In:* Proc. of the 13th International Conference on Machine Learning.

[10] Hansen, L. K., Salamon. P. (1990). Neural Network Ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12, 993-1001.

[11] Kuncheva, L. I. (2002). A Theoretical Study on Six Classifier Fusion Strategies. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 281-286.

[12] Kuncheva, L. I., Whitaker, C. J. (2003). Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy. *Machine Learning* 51, 181-207.

[13] Kuncheva, L. I., Whitaker, C. J., Shipp, C., Duin, R. (2003). Limits on the Majority Vote Accuracy in Classifier Fusion. *Pattern Analysis and Applications* 6, 22-31.

[14] Lam, L., Suen, C. (1997). Application of Majority Voting to Pattern Recognition: An Analysis of Its Behavior and Performance. *IEEE Transactions on Systems, Man, and Cybernetics* 27, 553-568.

[15] Martinez-Munoz, G., Suarez, A. (2005). Switching Class Labels to Generate Classification Ensembles. *Pattern Recognition* 38 (10) 1483-1494.

[16] Narasimhamurthy, A. (2005). Theoretical Bounds of Majority Voting Performance for a Binary Classification Problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 1988-1995.

[17] Plewczynski, D., Spieser, S. A. H., Koch, U. (2006). Assessing different classification methods for virtual screening. *Journal of Chem. Inf. Model* 46, 1098-1106.

[18] Rodriguez, J. J., Kuncheva, L. I. (2006). Rotation Forest: A New Classifier Ensemble Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, 1619-1630.

[19] Schapire, R. E., Singer, Y. (1999). Improved Boosting Algorithms Using Confidence-Rated Predictions. *Machine Learning* 37, 297-336.

[20] Schwenk, H., Bengio, Y. (2000). Boosting neural networks. *Neural Computation* 12, 1869-1887.

[21] Sinha, A. P., Huimin, Z. (2008). Incorporating domain knowledge into data mining classifiers: *An application in indirect lending. Decision Support Systems* 46, 287-299.

[22] Sun, Y., Wang, Y., Wong, A. K. C. (2006). Boosting an associative classifier. *IEEE Transactions on Knowledge and Data Engineering* 18, 988-992.

[23] UCI. *Center for Machine Learning and Intelligent Systems.* University of California Irvine Machine Learning Repository http://archive.ics.uci.edu/ml/datasets.html.

[24] Wang, H., Fan, W., Yu, P. S., Han, J. (2003). Mining Concept-Drifting Data Streams Using Ensemble Classifiers. *In:* Proc. of Ninth ACM SIGKDD Intelligent Conference Knowledge Discovery and Data Mining.

[25] Won, H. H., Cho, S. B. (2003). Neural network ensemble with negatively correlated features for cancer classification. *Lecture Notes in Computer Science* 2714, 1143-1150.