The Bias Variance Trade-off in Bootstrapped Error Correcting Output Code Ensembles

R.S.Smith and T.Windeatt

Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, Surrey GU2 7XH, UK
{Raymond.Smith, T.Windeatt}@surrey.ac.uk.

Abstract. By performing experiments on publicly available multi-class datasets we examine the effect of bootstrapping on the bias/variance behaviour of error-correcting output code ensembles. We present evidence to show that the general trend is for bootstrapping to reduce variance but to slightly increase bias error. This generally leads to an improvement in the lowest attainable ensemble error, however this is not always the case and bootstrapping appears to be most useful on datasets where the non-bootstrapped ensemble classifier is prone to overfitting.

1 Introduction

When considering the errors made by statistical pattern classifiers it is useful to group them under three headings. Firstly there is the unavoidable error, known as *Bayes error*, which is caused by noise in the process that generates the patterns. A second source of error is *variance*; this is caused by the sensitivity of a learning algorithm to the chance details of a particular training set and causes slightly different training sets to produce classifiers that give different predictions for some patterns. Thirdly there are errors caused by *bias* in a learning algorithm; here the problem is that the classifier is unable, for whatever reason, to adequately model the class decision boundaries in the pattern feature space. When training a classifier there is often a tradeoff between bias and variance [8] so that a high value of one implies a low value of the other.

A successful approach to constructing multi-class classifiers has proved to be that of error-correcting output code (ECOC) ensembles [6,9]. In this approach the multi-class problem is decomposed into a series of 2-class problems, or dichotomies, and a separate base classifier trained to solve each one. These 2-class problems are constructed by repeatedly partitioning the set of target classes into pairs of super-classes so that, given a large enough number of such partitions, each target class can be uniquely represented as the intersection of the super-classes to which it belongs. The classification of a previously unseen pattern is then performed by applying each of the base classifiers so as to make decisions about the super-class membership of the pattern. Redundancy can be introduced into the scheme by using more than the minimum number of base classifiers and this allows errors made by some of the classifiers to be corrected by the ensemble

as a whole. It has been shown [10,12] that ECOC reduces both bias and variance when compared with a single multi-class classifier.

A generally desirable property of multiple classifier systems (MCS), of which ECOC is an example, is that there should be *diversity* among the individual classifiers in the ensemble [4,15]. By this is meant that the errors made by component classifiers should, as far as possible, be uncorrelated so that the error correcting properties of the ensemble can have maximum effect. One way of achieving this is to apply bootstrapping to the training set so that each base classifier is trained on a unique bootstrap replicate. These are created from the original training set by repeated sampling with replacement. This creates a training set which has, on average, 63% of the patterns in the original set but with some patterns repeated to form a training set of the same size.

When bootstrapping is used in a majority voting ensemble of identical classifiers it leads to the technique of *bagging* [2]. This is known to reduce variance at the cost of increased bias [3,7], particularly when using an unstable classifier such as MLP. The situation with ECOC bootstrapping is somewhat analogous to a bagged ensemble; the difference, however, is that in the latter case each classifier is trained to solve an identical problem, whereas the ECOC base classifiers are trained to solve different sub-problems.

One of the advantages of the ECOC approach is that it makes it possible to perform multi-class classification by using base classifier algorithms that are more suited to solving 2-class problems. Examples include support vector machines (SVMs) [5] and multi-layer perceptron (MLP) neural networks [1]. In this paper we investigate experimentally three types of base classifier, namely single hidden layer MLPs, Gaussian kernel SVMs and polynomial kernel SVMs. Each of these base classifier types can be regarded as being controlled by two main parameters which respectively control the *capacity* and the *training strength* of the learning algorithm. The term *capacity* [5] refers to the ability of an algorithm to learn a training set with low or zero training error. By *training strength* we mean the amount of effort that is put into training the classifier to learn the details of a given training set. Intuitively, high capacity tends to imply a low bias and high training strength tends to imply high variance. For a given dataset and learning algorithm therefore, there is often a tradeoff between the values of these two parameters.

2 Kohavi-Wolpert Definition of Bias and Variance

The statistical concepts of bias, variance and noise originally emerged from regression theory. In this context they can be defined in such a way that the squared loss can be expressed as the sum of noise, bias^2 and variance. The goal of generalising these concepts to classification problems, using a 0-1 or other loss function, has proved elusive and several alternative definitions have been proposed (see [10] for a summary). In fact it is shown in [10] that, for a general loss function, these concepts cannot be defined in such a way as to possess all

desirable properties simultaneously. For example the different sources of error may not be additive, or it may be possible for variance to take negative values.

In this study we adopt the Kohavi-Wolpert definitions [11]. Let X be a random variable representing input patterns and Y a random variable representing the target classes. Consider a learning algorithm $\mathcal{L}$ which, given a training set T, produces a classification function $\mathcal{L}(T)$ which maps X to Y. Then the Kohavi-Wolpert definitions of bias, variance and total error are given by the following equations:

$$bias_x^2 = \frac{1}{2} \sum_{y \in Y} \left[\hat{P}_{Y,X} \left(Y = y | X = x \right) - \hat{P}_T \left(\mathcal{L}(T)(x) = y \right) \right]^2 - D_x \tag{1}$$

$$variance_x = \frac{1}{2} \left[1 - \sum_{y \in Y} \hat{P}_T \left(\mathcal{L}(T)(x) = y \right)^2 \right] + D_x \tag{2}$$

$$D_x = \frac{1}{2} \sum_{y \in Y} \hat{P}_T \left(\mathcal{L}(T)(x) = y \right) \left[1 - \hat{P}_T \left(\mathcal{L}(T)(x) = y \right) \right] / \left(N_T - 1 \right) \tag{3}$$

$$error_x = \hat{P}_{Y,X} \left(\mathcal{L}(T)(x) \neq Y | X = x \right) = bias^2 + variance \tag{4}$$

Here $\hat{P}_{Y,X}\left(Y = y | X = x\right)$ is the empirical probability that the actual class of pattern x is y; in practice this takes the value 1 for a particular value of y and 0 for all others. $\hat{P}_T\left(\mathcal{L}(T)(x) = y\right)$ is the empirical probability, taken over a collection of N_T training sets, that the learning algorithm produces a classifier that assigns pattern x to target class y. D_x is a de-biasing term which ensures that the estimates of $bias_x^2$ and $variance_x$ are reliable for small values of N_T. The ability to apply this correction is one of the advantages of the Kohavi-Wolpert definitions; for example in [11] it is shown to lead to stable results using just 10 sample training sets.

Another advantage of the above definitions is that they give an additive decomposition of error. A major problem, however, is that there is no separate allowance for Bayes error. The rationale for this is that, on realistic datasets, this component of error cannot be estimated because the sampling is rarely dense enough to allow the probabilities of different classes to be estimated at a fixed value of x (a method for overcoming this problem has, however, been proposed in [10]). In effect, the Bayes error component is absorbed into the bias^2 term, thus giving a value which is biased too high. In this study, however, we are interested only in changes to bias and variance as the base classifier parameters are varied, and so this issue does not affect the conclusions of the paper.

3 ECOC Base Classifier Parameters

As noted in section 1, we wish to characterise the base classifier parameters as those which control the capacity of the classifier and those which control the training strength. In the case of single hidden layer MLPs, a natural choice is to take the number of hidden nodes and the number of training epochs respectively.

For SVM base classifiers note that the objective function to be minimised during training [5] has the form $\|\mathbf{w}\|^2 + C\sum_i \xi_i$ where $\mathbf{w}$ is the weight vector to be computed, C is a cost parameter and ξ_i are slack variables. The value of C controls the tradeoff between exactly fitting the training data (by driving the ξ_i towards zero) and maximising the margin (by driving $\|\mathbf{w}\|$ towards zero). It follows that C fulfils the role of the training strength parameter, with high values leading to the training set being modelled more precisely.

The choice of capacity parameter for SVMs depends on the kernel function being used. The Gaussian kernel has the form $exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x}-\mathbf{y}\|^2\right)$ where σ controls the diameter of the sphere of influence around each support vector. For this kernel $1/\sigma^2$ is a suitable choice for the capacity parameter (the inverse is taken in order to ensure that capacity increases as the parameter value increases). Some pictorial examples of the effect of varying C and σ can be found in [14]. For the polynomial kernel function $(\mathbf{x}\cdot\mathbf{y}+1)^d$ the capacity is determined by the degree parameter d.

4 Experiments

In this section we present the results of performing classification experiments on 11 multi-class datasets obtained from the publicly available UCI repository [13]. The characteristics of these datasets in terms of size, number of classes and number of features are given in table 1.

Table 1. Experimental datasets showing the number of patterns, classes, continuous and categorical features.

Dataset	Num. Patterns	Num. Classes	Cont. Features	Cat. Features
dermatology	366	6	1	33
ecoli	336	8	5	2
glass	214	6	9	0
iris	150	3	4	0
segment	2310	7	19	0
soybean	683	19	0	35
thyroid	7200	3	6	15
vehicle	846	4	18	0
vowel	990	11	10	1
waveform	5000	3	40	0
yeast	1484	10	7	1

For each dataset, ECOC ensembles of size 200 were constructed using each of three base classifier types and a range of base classifier capacity and training strength parameters. Each such combination was repeated 10 times with different randomly chosen stratified training sets and different randomly generated ECOC

coding matrices; for neural network base classifiers another source of random variation was the initial network weights. In each run the data was normalised to make the training set have zero mean and unit variance. The ECOC code matrices were constructed in such a way as to have balanced numbers of 1s and 0s in each column. Training sets were based on a 20/80 training/test set split. Each experiment was repeated with and without bootstrapping being applied to the construction of the individual base-classifier training sets. In total this led to 27,900 experimental runs being performed. For each unique combination of parameters and algorithms, the Kohavi-Wolpert bias^2, variance and total error were calculated in accordance with Eqns. 1 to 4 over the 10 randomised runs.

The base classifier types employed were single-hidden layer MLP neural networks using the Levenberg-Marquardt training algorithm, SVMs with Gaussian kernel and SVMs with polynomial kernel. The neural networks were constructed as a single hidden layer of perceptrons, with the number of nodes ranging from 2 to 16 and the number of training epochs from 2 to 1024. For Gaussian SVMs the width parameter σ was varied between 1 and 8, whilst for polynomial SVMs degrees of 1,2,3 and 4 were used. The cost parameter of SVMs was varied between 10^{-3} and 10^3. In all cases, apart from polynomial degrees, the base classifier parameters were varied in geometric progression.

For reference purposes a complete list of the lowest ensemble errors obtained in these experiments, for each base classifier type, is given in Table 2

Table 2. The lowest percentage ensemble error values obtained using three types of ECOC base classifier. Error values are shown with the application of bootstrapping (BS) and without ($\overline{\text{BS}}$).

Dataset	Neural Network		Gaussian SVM		Polynomial SVM	
	BS	$\overline{\text{BS}}$	BS	$\overline{\text{BS}}$	BS	$\overline{\text{BS}}$
dermatology	**3.3**	4.8	**2.9**	**2.9**	**2.8**	3.2
ecoli	**16.8**	18.3	15.1	**15.0**	**15.7**	16.0
glass	**36.2**	36.8	35.5	**35.3**	**37.2**	38.0
iris	**4.8**	5.1	**5.0**	5.7	5.5	**5.3**
segment	**4.0**	**4.0**	**5.7**	**5.7**	**5.7**	6.1
soybean	9.7	**9.3**	8.4	**7.8**	8.3	**8.2**
thyroid	2.7	**2.6**	**2.7**	2.8	**2.9**	3.4
vehicle	**20.9**	22.1	**22.1**	22.2	**23.1**	23.5
vowel	23.2	**21.3**	21.3	**20.9**	26.4	**25.9**
waveform	**14.9**	16.7	**14.3**	14.4	**14.4**	14.5
yeast	41.9	**41.4**	41.2	**41.1**	**42.0**	**42.0**

4.1 Bias-Variance Tradeoff

Some representative examples of the bias-variance behaviour observed in these experiments are illustrated in Fig. 1. Here the effect is shown, both with and

without bootstrapping, of increasing the training strength parameter for various datasets and base classifier types. For each graph the base classifier capacity parameter is fixed at the optimal value obtained on the test set for the given dataset.

A number of observations can be made about Fig. 1. We discuss first examples (b) to (e) where it can be seen that, as expected, there is a general tendency for the bias^2 to decrease and the variance to increase as the training strength increases. The effect of bootstrapping on variance is to lessen this increase, particularly for higher values of the training strength parameter. It can be observed that there is a tendency for bootstrapping to slightly increase the bias^2 error, although the effect is usually small and the curves generally lie very close to each other. It is noteworthy that the bootstrapped variance and total error curves tend to level out at a lower value than the non-bootstrapped versions; this indicates that bootstrapping makes the ensemble more resistant to overfitting the data at high training strengths.

At some point there is an optimal tradeoff between bias and variance where the total error is minimised. The position of the optimum may vary depending on whether bootstrapping is used or not (e.g. (c) and (d)) or it may be the same in both cases (e.g. (b) and (e)). Whether bootstrapping reduces the total error depends on the values of bias^2 and variance at the optimal tradeoff points. In examples (b) to (d) the variance reduction induced by bootstrapping is sufficient to lead to a significant overall reduction in error despite any slight increase in bias. The benefit of bootstrapping tends to be lower, or even negative, however when the optimal bias/variance tradeoff occurs at low training strengths; this is because, as in case (e), the divergence between the variance curves is insufficient, at this point, to significantly impact the total error.

The behaviour observed on some datasets, for example case (a) of Fig. 1, can be different from that described above. Here the ECOC classifier does not exhibit a pronounced tendency to overfit the data at high training strengths (as in cases (b) to (e)) and, as a result, variance is not reduced by bootstrapping. In fact, in example (a) both the bias^2 and variance curves of the bootstrapped ensemble lie slightly above those of the non-bootstrapped version, so bootstrapping leads to an overall increase in total error.

4.2 Bootstrapping vs. Non-Bootstrapping

In section 4.1 we examined the classification behaviour of ECOC ensembles under conditions of identical base classifier capacity and varying training strengths. In order to compare the performance of bootstrapped versus non-bootstrapped ensembles, however, it is necessary to look at them under optimal conditions and this may require the base classifier capacity, as well as training strength, to be different. The examples of Fig. 1 were chosen from cases where the optimal capacity was found to be the same for both types of ensemble, but this is not always the case. Due perhaps to its more stochastic behaviour, the neural network base classifier was found to be particularly prone to this phenomenon, with only 3 out of the 11 datasets requiring the same capacity parameter. For example on

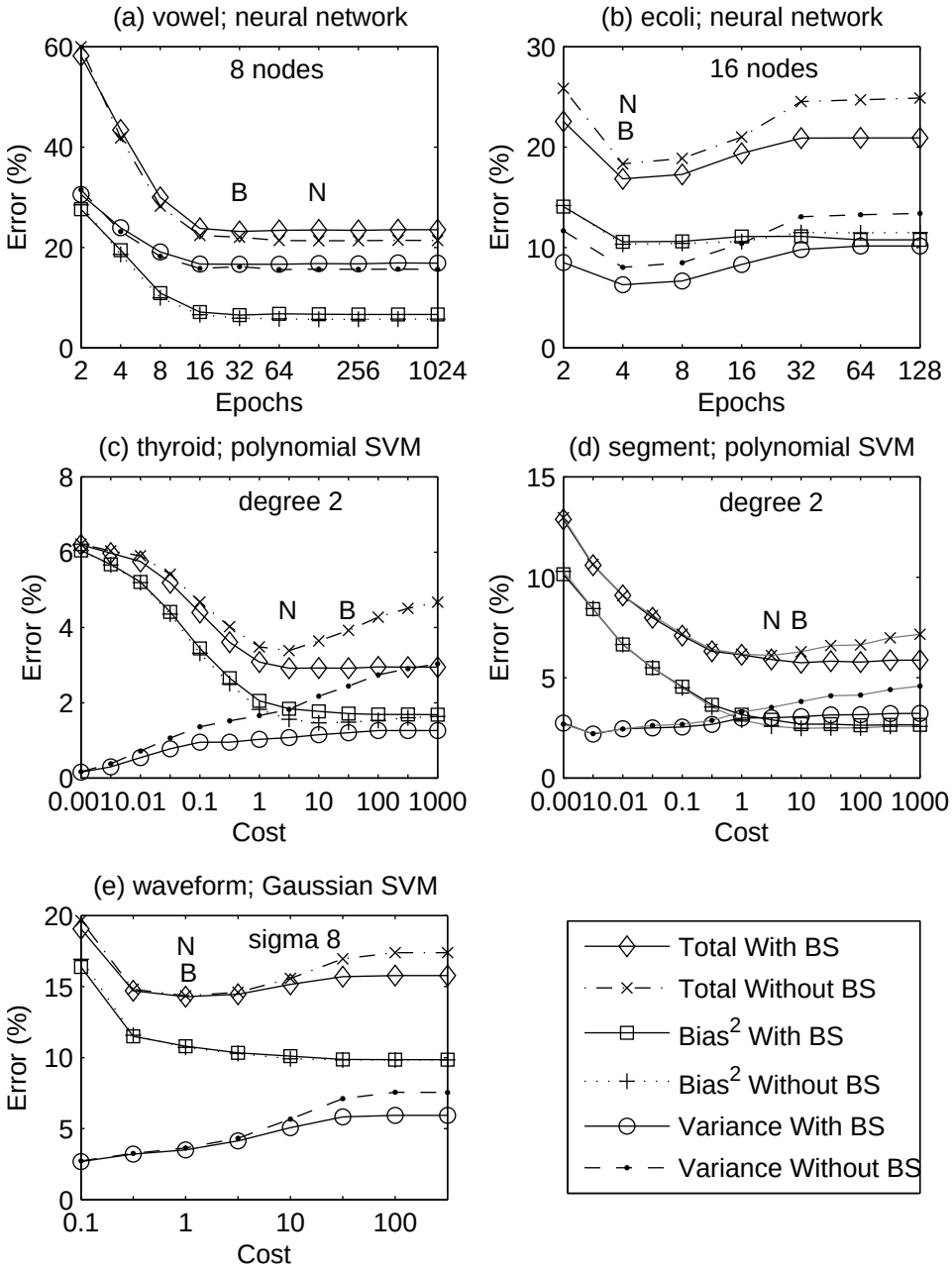

Fig. 1. Some example ensemble test-set bias, variance and total error curves as the ECOC base classifier training strength parameter is varied. These are shown with and without the application of bootstrapping during ensemble construction. 'B' and 'N' respectively mark the positions of minimum error with and without bootstrapping.

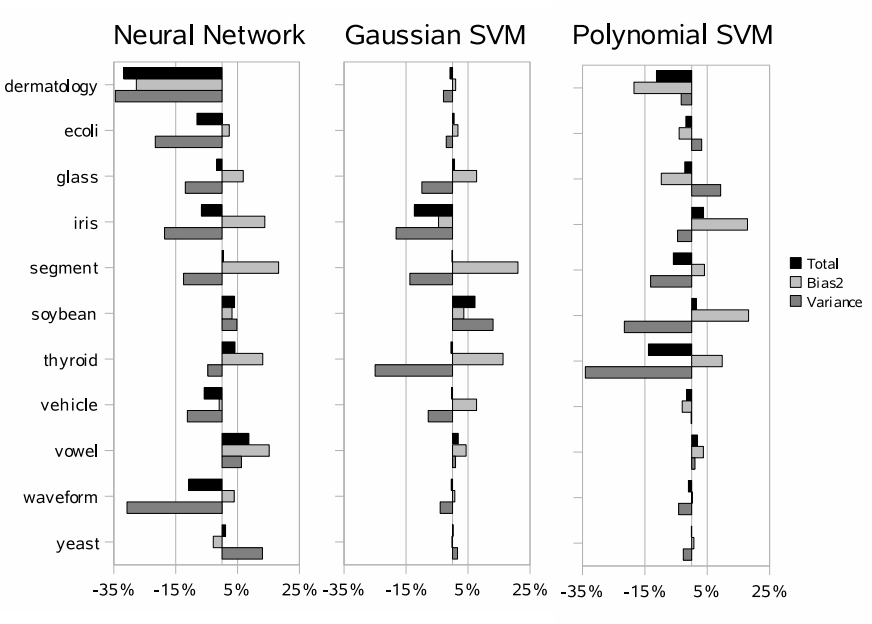

Fig. 2. The effect of bootstrapping on ECOC for three types of base classifier. Figures show the relative percentage change in the lowest attainable ensemble Kohavi-Wolpert test error components that result from bootstrapping. Negative values imply that bootstrapping leads to a reduction.

the yeast dataset this classifier was optimal at 4 nodes and 16 training epochs when bootstrapping was used but 8 nodes and 8 epochs when not.

Fig. 2 shows the relative percentage change[1] in test-set bias^2, variance and total error which resulted when the bootstrapped ensemble was compared with the non-bootstrapped version at their respective points of minimum total error. It can be seen from this that the general pattern is for bootstrapping to reduce variance but to increase bias and that this leads to a net reduction in total error. This pattern of behaviour is confirmed by Table 3 which shows the relative percentage changes averaged over the 11 datasets. There are, however, deviations from this pattern for individual datasets. For example dermatology, when using the neural network or polynomial SVM base classifiers, leads to the bias^2 at the point of bootstrapped minimum total error being significantly less than that obtained when bootstrapping is not applied.

[1] By relative percentage change we mean the value $100\,(v - v_{BS})\,/v$ where v and v_{BS} are measured without and with bootstrapping respectively.

Table 3. The average, over 11 datasets, of the effect of applying bootstrapping to ECOC. Figures show the mean percentage relative change in the lowest attainable ensemble Kohavi-Wolpert test error measures. Negative values imply that bootstrapping leads to a reduction.

Base Classifier	Total	$Bias^2$	Variance
Neural Network	-4.22	4.12	-11.05
Gaussian SVM	-0.36	5.44	-6.18
Polynomial SVM	-2.78	1.77	-6.38

5 Discussion and Conclusions

The main contribution of this paper to our understanding of ensemble classifiers is to shed light on how the bootstrapping of ECOC ensembles affects performance, not just in terms of overall classification error, but also how that error breaks down into its bias and variance components. Evidence has been presented to show that bootstrapping generally tends to lessen the impact of variance when compared with non-bootstrapped ensembles. This tends to be particularly noticeable at high values of the training strength parameter, leading to a reduced tendency to overtrain. The relative reduction in variance is, however, often achieved at the expense of a slight increase in the $bias^2$ component - a pattern of behaviour that is reminiscent of that observed in bagged ensembles [7].

Whilst the net effect of bootstrapping is usually to reduce the overall error that can be attained at optimal base classifier parameter settings, this is not universally the case and bootstrapping appears to be most useful on datasets for which the non-bootstrapped ensemble is prone to overfitting. This is to be expected since the latter type of dataset implies that variance error plays a more prominent role in determining the ensemble error.

Future work will be directed towards characterising more precisely the relationship between the properties of the dataset and the effect of ECOC bootstrapping. For example, when the available dataset is small, as with iris, it is likely that further reducing the base classifier training data by bootstrapping may lead to the introduction of bias. This cannot be the complete explanation, however, as increases in bias can also be observed on larger datasets such as thyroid and segment. Further investigation is required and it is hoped that this will lead to a theory that predicts when bootstrapping is advantageous.

6 Acknowlegements

This work was supported by EPSRC grant E061664/1.

References

1. Bishop MC. *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, 1995.

2. Breiman L. Bagging Predictors. *Machine Learning*, 24(2): 123-140, 1994.
3. Breiman L. Arcing Classifiers. *Annals of Statistics* 26(3): 801-849, 1998.
4. Brown G, Wyatt J, Harris R, Yao X. Diversity Creation Methods: A Survey and Categorisation. *Journal of Information Fusion*, 6(1), 2005.
5. Burges CJC. A Tutorial on Support Vector Machines for Pattern Recognition. *Knowledge Discovery and Data Mining*, 2(2), 1998.
6. Dietterich TG, Bakiri G. Solving Multiclass Learning Problems via Error-Correcting Output Codes. *Journal of Artificial Intelligence Research* 2: 263-286, 1995.
7. Dietterich TG, Kong EB. Machine learning bias, statistical bias, and statistical variance of decision tree algorithms. Technical Report, Dept. of Computer Science, Oregon State University. 1995.
8. Geman S, Bienenstock E. Neural networks and the bias / variance dilemma. Neural Computation, 4:1-58, 1992.
9. James G. Majority Vote Classifiers: Theory and Applications. PhD Dissertation, Stanford University, 1998.
10. James G. Variance and Bias for General Loss Functions. *Machine Learning*, 51 (2), 115-135, 2003.
11. Kohavi R, Wolpert D. Bias plus variance decomposition for zero-one loss functions. *Proc. 13th International Conference on Machine Learning*, pp. 275-283, 1996.
12. Kong EB, Dietterich TG. Error-correcting output coding corrects bias and variance. *Proc. 12th International Conference on Machine Learning*, pages 313–321, 1995.
13. Merz CJ, Murphy PM. UCI Repository of Machine Learning Databases, 1998. http://www.ics.uci.edu/~mlearn/MLRepository.html.
14. Valentini G, Dietterich TG. Bias-Variance Analysis of Support Vector Machines for the Development of SVM-Based Ensemble Methods. *Journal of Machine Learning Research* Vol. 5, pp. 725-775, 2004.
15. Windeatt T. Accuracy/ Diversity and Ensemble Classifier Design, *IEEE Trans Neural Networks*, 17(4), July, 2006.