Chapter 1
Bias-Variance Analysis of ECOC and Bagging Using Neural Nets

Cemre Zor, Terry Windeatt and Berrin Yanikoglu

Abstract One of the methods used to evaluate the performance of ensemble classifiers is bias and variance analysis. In this chapter, we analyse bootstrap aggregating (bagging) and Error Correcting Output Coding (ECOC) ensembles using a bias-variance framework; and make comparisons with single classifiers, while having Neural Networks (NNs) as base classifiers. As the performance of the ensembles depends on the individual base classifiers, it is important to understand the overall trends when the parameters of the base classifiers -nodes and epochs for NNs-, are changed. We show experimentally on 5 artificial and 4 UCI MLR datasets that there are some clear trends in the analysis that should be taken into consideration while designing NN classifier systems.

1.1 Introduction

Within machine learning research, many techniques have been proposed in order to understand and analyse the success of ensemble methods over single classifiers. One of the main approaches considers tightening the generalization error bounds by using the margin concept [17]. Though theoretically interesting, bounds are not usually tight enough to be used in practical design issues. Another method used to show why ensembles work well is bias and variance analysis. In this chapter, we try to analyse the success of bootstrap aggregating (bagging) [8] and Error Correcting Output Coding (ECOC) [4] as ensemble classification techniques, by using Neural Networks (NNs) as the base classifiers and zero-one loss as the loss function within the bias and variance framework of James [13]. As the characteristics

Cemre Zor and Terry Windeatt
Centre for Vision, Speech and Signal Processing, University of Surrey, UK, GU2 7XH
e-mail: (c.zor,t.windeatt)@surrey.ac.uk

Berrin Yanikoglu
Sabanci University, Tuzla, Istanbul, Turkey, 34956 e-mail: berrin@sabanciuniv.edu

of the ensemble depend on the specifications of the base classifiers, having a detailed look at the parameters of the base classifiers within the bias-variance analysis is of importance. Comparisons of bagging and ECOC ensembles with single classifiers have been shown through various experiments by changing these parameters, namely nodes and epochs of NN base classifiers. Similar work for bagged Support Vector Machines (SVMs) within Domingos' bias-variance framework [6] can be found in [22].

1.1.1 Bootstrap Aggregating (Bagging)

Bagging [8] is a commonly used ensemble method, which suggests aggregating the decisions of base classifiers trained on bootstrapped training sets.

Using the idea of bootstrapping, i training sets, $D_{1,2..,i}$, are formed by uniformly sampling elements from the main training set D, with replacement. Note that on average about 37% of each D_i is replicated. By bootstrapping, a close enough approximation to random & independent data generation from a known underlying distribution is expected to be achieved [3]. The training sets created are later used to train the base classifiers; and classification is performed by combining their decisions through majority voting.

1.1.2 Error Correcting Output Coding (ECOC)

ECOC is an ensemble technique [4], in which multiple base classifiers are created and trained according to the information obtained from a pre-set binary *code matrix*. The main idea behind this procedure is to solve the original multi-class problem by combining the decision boundaries obtained from simpler two-class decompositions. The original problem is likely to be more complex compared to the sub-problems into which it is decomposed, and therefore the aim is to come up with an easier and/or more accurate solution using the sub-problems rather than trying to solve it by a single complex classifier.

The base classifiers are actually two-class classifiers (dichotomizers), each of which is trained to solve a different bi-partitioning of the original problem. The bi-partitions are created by combining the patterns from some predetermined classes together and relabeling them. An example bi-partitioning of an $N > 2$ class dataset would be by having the patterns from the first 2 classes labeled as $+1$ and the last $N-2$ classes as -1. The training patterns are therefore separated into two super-classes for each base classifier, and the information about how to create these super-classes is obtained from the ECOC matrix.

Consider an ECOC matrix C, where a particular element C_{ij} is an element of the set $(+1,-1)$. Each C_{ij} indicates the desired label for class i, to be used in training the base classifier j; and each row, called a *codeword*, represents the desired output

for the whole set of base classifiers for the class it indicates. Figure 1.1 shows an ECOC matrix for a 4-class problem for illustration purposes.

	b1	b2	b3	b4	b5
c1	+1	+1	+1	-1	-1
c2	+1	-1	-1	+1	-1
c3	+1	+1	-1	-1	-1
c4	-1	-1	-1	+1	+1

Fig. 1.1 An example ECOC matrix for a 4-class problem. b1,..,5 indicate the names of columns to be trained by base classifiers 1,..5; and c1,..,4 indicate names of rows dedicated to classes 1,..,4.

During testing (decoding), a given test sample is classified by computing the similarity between the output (hard or soft decision) of each base classifier and the codeword for each class, by using a distance metric such as the Hamming or the Euclidean distance. The class with the minimum distance is then chosen as the estimated class label.

As the name of the method implies, ECOC can handle incorrect base classification results up to a certain degree. Specifically, if the minimum Hamming distance (*HD*) between any pair of codewords is d, then up to $\lfloor (d-1)/2 \rfloor$ single bit errors can be corrected. In order to help with the error correction in the testing stage, the code matrix is advised to be designed to have large Hamming distances between the codewords of different classes. Moreover, when deterministic classifiers such as Support Vector Machines (SVMs) are used as base classifiers, the *HD* between a pair of columns should also be large enough so that the outputs of the base classifiers are uncorrelated [4] and their individual errors can be corrected by the ensemble. Many variations of the design of the ECOC matrix, namely *encoding*; and the test stage, namely *decoding*, have been suggested in the literature so far and are still open problems.

While the errors made by individual dichotomizers may be corrected using the ECOC approach, the encoding of the code matrix is also being researched in order to make the method more powerful. A straightforward approach in design is to have an *exhaustive code*, which includes all possible bi-partitionings of the problem. This means that for a problem consisting of n classes, an ECOC matrix of size $n \times (2^{n-1}-1)$ is created[1]. Apart from the use of exhaustive codes which are computationally expensive and do not guarantee the best performance, some commonly used data-independent techniques such as the one-versus-all, one-versus-one, dense random and sparse random [1] coding schemes have been suggested for the ECOC matrix encoding. Furthermore, data dependent ECOC designs, in which training

[1] The number of classes is calculated as $2^{n-1}-1$ after removing the complementary and the all-zero or all-one columns [4].

data is used to create coding matrices meaningful within the input data domain, have also gained importance. As an example for the data dependent ECOC design, intelligent creation of binary column classifiers which can better fit the decision boundaries of the problem training set can be given [7]. This is obtained through splitting the original set of classes into sub-classes using the training data. Most importantly, although problem dependent coding approaches provide successful outcomes, it has been theoretically and experimentally proven that the randomly generated long or deterministic equi-distant code matrices are also close to optimum performance when used with strong base classifiers [12, 14]. This is why, *long-random* codes have also been used for the experiments in this chapter. It has also been shown that in real life scenarios that equidistant codes are superior at least for shorter codes; but as length of code word is increased, the coding/decoding strategy becomes less significant [24]. Finally for encoding, note that the use of ternary ECOC matrices [1], where a zero symbol is used to leave a class out of the consideration of a dichotomizer, has also gained interest within the field.

There are many ways used in the decoding of the ECOC matrix apart from the usual *HD* decoding. It is a common convention that the decoding of the problem-dependent ECOC matrices is performed in accordance with their encoding. As common examples of decoding; general weighted decoding approaches, together with Centroid of Classes, Least Squares and Inverse Hamming Distance methods can be listed [24].

As a final point, it should be mentioned that many static and dynamic pruning methods can also be applied to ECOC (e.g. column selection), just like any other ensemble method, so as to increase the efficiency and accuracy.

1.1.3 Bias and Variance Analysis

Bias and variance analysis plays an important role in ensemble classification research due to the framework it provides for classifier prediction error decomposition.

Initially, Geman has decomposed prediction error into bias and variance terms under the regression setting using squared-error loss [10]. This decomposition brings about the fact that a decrease/increase in the prediction error rate is caused by a decrease/increase in bias, or in variance, or in both. Extensions of the analysis have been carried out on the classification setting, and later applied on different ensemble classifiers in order to analyse the reason behind their success over single classifiers. It has been shown that the reason for most of the ensembles to have lower prediction rates is due to the reductions they offer in sense of both bias and variance.

However, the extension of the original theoretical analysis on regression has been done in various ways by different researchers for classification; and there is no standard definition accepted. Therefore, the results of the analyses also differ from each other slightly. Some of the definitions/frameworks that have gained interest within the research field are given by Breiman [3], Kohavi and Wolpert [15], Dietterich and

Kong [16], Friedman [9], Wolpert [25], Heskes [11], Tibshirani [19], Domingos [6] and James [13].

Although there are dissimilarities in-between the frameworks, the main intuitions behind each are similar. Consider a training set T with patterns (x_i, l_i), where x represents the feature vector and l the corresponding label. Given a test pattern, an optimal classifier model predicts a decision label by assuring the lowest expected loss over all possible target label values. This classifier, which is actually *the Bayes classifier* when used with the zero-one loss function, is supposed to know and use the underlying likelihood probability distribution for the input dataset patterns/classes. If we call the decision of the optimal classifier as the optimal decision (*OD*); then for a given test pattern (x_i, l_i), $OD = argmin_\alpha E_t[L(t,\alpha)]$ where L denotes the loss function used, and l the possible target label values.

The estimator, on the other hand, is actually an averaged classifier model. It predicts a decision label by assuring the lowest expected loss over all labels that are created by classifiers trained on different training sets. The intrinsic parameters of these classifiers are usually the same, and the only difference is the training sets that they are trained on. In this case, instead of minimizing the loss over the target labels using the known underlying probability distribution as happens in the optimal classifier case, the minimization of the loss is carried out for the set of labels which are created by the classifiers trained on various training sets. If the decision of the estimator is named as the expected estimator decision (*EED*); then for a given test pattern (x_i, l_i), $EED = argmin_\alpha E_l[L(l,\alpha)]$ where L denotes the loss function used, and l the label values obtained from the classifiers used. For regression under the squared-error loss setting, the *OD* is the *mean* of the target labels while the *EED* is the *mean* of the classifier decisions obtained via different training sets.

Bias can mainly be defined as the difference or the distance between *OD* and *EED*. Therefore, it emphasizes how effectively the optimal decision can be predicted by the estimator. The type of the distance metric used depends on the loss function of the classification. On the other hand, variance is calculated as the expected loss exposed by the classifiers, which are trained on different training sets, while predicting *OD*. So, it shows how sensible the estimate is, against variations in the training data [9].

The problem with the above mentioned definitions of bias and variance is that most of them are given for specific loss functions such as the zero-one loss. It is difficult to generalize them for the other loss functions; usually new definitions are given for each new loss function. Secondly, as for the definitions which are proposed to be applicable for all loss functions, the problem of failing to satisfy the additive decomposition of the prediction error defined in [10] exists.

The definition of James [13] has advantages over the others as it proposes to construct a bias and variance scheme which is generalizable to any symmetric loss function, while assuring the additive prediction error decomposition by utilizing two new concepts called *systematic effect* (*SE*) and *variance effect* (*VE*). These concepts also help realizing the effects of bias and variance on the prediction error.

Some characteristics of the other definitions which make James' more preferable are as follows:

1. Dietterich allows a negative variance and it is possible for the Bayes classifier to have positive bias.
2. Experimentally, the trends of Breiman's bias and variance closely follow James' *SE* and *VE* respectively. However, for each test input pattern, Breiman separates base classifiers into two sets, as biased and unbiased; and considers each test pattern only to have either bias or variance accordingly.
3. Kohavi and Wolpert also assign a nonzero bias to the Bayes classifier but the Bayes error is absorbed within the bias term. Although it helps avoid the need to calculate the Bayes error in real datasets through making unwarranted assumptions, it is not preferable since the bias term becomes too high.
4. The definitions of Tibshirani, Heskes and Breiman are difficult to generalize and extend for the loss functions other than the ones for which they were defined.
5. Friedman proposes that bias and variance do not always need to be additive.

In addition to all these differences, it should also be noted that the characteristics of bias and variance of Domingos' definition are actually close to James', although the decomposition can be considered as being multiplicative [13].

In the literature, attempts have been made to explore the bias-variance characteristics of ECOC and bagging ensembles. Examples can be found in [13, 16, 3, 18, 5, 22]. In this chapter, a detailed bias-variance analysis of ECOC and bagging ensembles using NNs as base classifiers is given while systematically changing parameters, namely nodes and epochs, based on James' definition.

We start by taking a detailed look at the bias and variance framework of James in the next section.

1.2 Bias and Variance Analysis of James

James [13] extends the prediction error decomposition, which is initially proposed by Geman et al [10] for squared error under regression setting, for all symmetric loss functions. Therefore, his definition also covers zero-one loss under classification setting, which we use in the experiments.

In his decomposition, the terms *systematic effect* (*SE*) and *variance effect* (*VE*) satisfy the additive decomposition for all symmetric loss functions, and for both real valued and categorical predictors. They actually indicate the effect of bias and variance on the prediction error. For example, a negative *VE* would mean that variance actually helps reduce the prediction error. On the other hand, the bias term is defined to show the average distance between the response and the predictor; and the variance term refers to the variability of the predictor. As a result, both the meanings and the additive characteristics of the bias and variance concepts of the original setup have been preserved. Following is a summary of the bias-variance derivations of James:

For any symmetric loss function L, where $L(a,b) = L(b,a)$:

$$\begin{aligned} E_{Y,\tilde{Y}}[L(Y,\tilde{Y})] &= E_Y[L(Y,SY)] + E_Y[L(Y,S\tilde{Y}) - L(Y,SY)] \\ &\quad + E_{Y,\tilde{Y}}[L(Y,\tilde{Y}) - L(Y,S\tilde{Y})] \\ prediction\,error &= Var(Y) + SE(\tilde{Y},Y) + VE(\tilde{Y},Y) \end{aligned} \tag{1.1}$$

where $L(a,b)$ is the loss when b is used in predicting a, Y is the response and $\tilde{Y}$ is the predictor. $SY = argmin_\mu E_Y[L(Y,\mu)]$ and $S\tilde{Y} = argmin_\mu E_Y[L(\tilde{Y},\mu)]$. We see here that prediction error is composed of the variance of the response (irreducible noise), *SE* and *VE*.

Using the same terminology, the bias and variance for the predictor are defined as follows:

$$\begin{aligned} Bias(\tilde{Y}) &= L(SY,S\tilde{Y}) \\ Var(\tilde{Y}) &= E_{\tilde{Y}}[L(\tilde{Y},S\tilde{Y})] \end{aligned} \tag{1.2}$$

When the specific case of classification problems with zero-one loss function is considered, we end up with the following formulations:

$L(a,b) = I(a \neq b)$, $Y\,\varepsilon\,\{1,2,3..N\}$ for an N class problem, $P_i^Y = P_Y(Y=i)$, $P_i^{\tilde{Y}} = P_{\tilde{Y}}(\tilde{Y}=i)$, $ST = argmin_i E_Y[I(Y \neq i)] = argmax_i P_i^Y$

Therefore,

$$\begin{aligned} Var(Y) &= P_Y(Y \neq SY) = 1 - max_i P_i^Y \\ Var(\tilde{Y}) &= P_{\tilde{Y}}(\tilde{Y} \neq S\tilde{Y}) = 1 - max_i P_i^{\tilde{Y}} \\ Bias(\tilde{Y}) &= I(S\tilde{Y} \neq SY) \\ VE(\tilde{Y},Y) &= P(Y \neq \tilde{Y}) - P_Y(Y \neq S\tilde{Y}) = P_{S\tilde{Y}}^Y - \sum_i P_i^Y P_i^{\tilde{Y}} \\ SE(\tilde{Y},Y) &= P_Y(Y \neq S\tilde{Y}) - P_Y(Y \neq SY) = P_{SY}^Y - P_{S\tilde{Y}}^Y \end{aligned} \tag{1.3}$$

where $I(q)$ is 1 if q is a true argument and 0 otherwise.

1.3 Experiments

1.3.1 Setup

In the experiments, 3 classification methods have been analysed: Single classifier, bagging, and ECOC. In each case, 50 classifiers are created for bias-variance analysis. Each of these 50 classifiers is either a single classifier, or an ensemble consisting of 50 bagged classifiers or ECOC matrices of 50 columns. Due to the reasons explained in Sect. 1.1.2, the ECOC matrices are created by randomly assigning binary values to each matrix cell; and Hamming Distance is used as the metric in the decod-

ing stage. The optimization method used in NNs is the Levenberg-Marquart (LM) technique; the level of training (epochs) varies between 2 and 15; and the number of nodes between 2 and 16.

In artificial dataset experiments, training sets are created by simple random sampling from the infinite data at hand to be used in training 50 classifiers for bias/variance analysis. The number of training patterns per classifier is equal to 300, and the number of test patterns is 18000. Experiments have been repeated 10 times using different test data (also generated via simple random sampling) together with different training data and ECOC matrices in each run, and the results are averaged. In the two-class problem experiments, ECOC has not been used as it is a multi-class classification technique. Applying ECOC in such cases would be nothing different than applying bagging; effect of bootstrapping in bagging would be similiar to the effect of the random initial weights of LM in ECOC.

For the UCI datasets having separate test sets, the training sets for each of the 50 classifiers are created from the finite dataset at hand using bootstrapping. Bootstrapping is expected to be a close enough approximation to random & independent data generation from a known underlying distribution [3].The analysis has been performed just once without repetition, as the test set is given/fixed. However, the results of the ECOC setting are averaged over 10 iterations with different matrices.

As for the UCI datasets without separate test sets, the *ssCV* cross-validation method of Webb and Conilione [23], which allows the usage of the whole dataset both in training and test stages, has been implemented. Within a number of iterations, all data is effectively used for both training and testing in each of the 50 classifiers, and the overall results for bias and variance analysis are recorded. The procedure is not repeated as the iterations within *ssCV* already cover the whole dataset. However, 10 different ECOC matrices are again used in ECOC setting, and results are averaged. Note that in *ssCV*, the shortcomings of the hold-out approach like the usage of small training and test sets; and the lack of inter-training variability control between the successive training sets has been overcome. In our experiments, we set the inter-training variability constant δ to $1/2$.

The diagram in Fig. 1.2 visualizes the experimental setup. Experiments have been carried out on 5 artificial and 4 UCI MLR [2] datasets; three of the artificial datasets being created according to Breiman's description in [3]. Detailed information about the sets can be found in Table 1.1.

The Bayes error, namely $Var(Y)$ for the zero-one loss function (see Eq. 1.3), is analytically calculated for the artificial datasets, as the underlying likelihood probability distributions are known. As for the real datasets, usually either the need for the underlying probability distributions has been overcome by assuming zero noise level [6], or some heuristic methods like using nearest neighbours [13] have been proposed to estimate the underlying probability distributions and therefore the Bayes error in the literature. The first approach has the shortcoming of a wrong estimate on bias. Therefore, we also use a heuristic method in our experiments to do the estimation. Our motivation is to find the optimal classifier parameters giving the lowest error rate possible, through cross-fold validation (*CV*); and then to use these parameters to construct a classifier which is expected to be close enough to the Bayes

Table 1.1 Summary of the datasets used

	Type	# Training Samples	# Test Samples	# Attributes	# Classes	Bayes Error (%)
TwoNorm [3]	Artificial	300	18000	20	2	2.28
ThreeNorm [3]	Artificial	300	18000	20	2	10.83
RingNorm [3]	Artificial	300	18000	20	2	1.51
ArtificalMulti1	Artificial	300	18000	2	5	21.76
ArtificalMulti2	Artificial	300	18000	3	9	14.33
Glass Identification	UCI	214*	-	10	6	38.66
Dermatology	UCI	358*	-	33	6	9.68
Segmentation	UCI	210	2100	19	7	4.21
Yeast	UCI	1484*	-	8	10	43.39

*: The total number of the elements of the UCI datasets without separate test sets, are listed under # of training samples.

classifier. The constructed classifier is then used to calculate the output probabilities per pattern in the dataset. For this, we first find an optimal set of parameters for RBF SVMs by applying 10 fold *CV*; and then, obtain the underlying probabilities by utilizing the leave-one-out approach. Using the leave-one-out approach instead of training and testing with the whole dataset using the previously found *CV* parameters helps us avoid overfitting. It is assumed that the underlying distribution stays almost constant for each fold of the leave-one-out procedure.

1.3.2 Results

In this section, some clear trends found in the analysis are discussed under three subsections: the prediction error, convergence to the prediction error, and bias/variance versus *SE*/*VE*. In the first sub-section, the comparison of the prediction error rates is made for the bagging, ECOC and single classifiers, while in the second one the convergence points in sense of number of nodes and epochs where the prediction error converges to its optimum are discussed. Finally in the third sub-section, the relationship between bias/variance and SE/VE is analysed.

Although the observations are made using 9 datasets, for brevity reasons we only present a number of representative graphs.

The Prediction Error

Prediction errors obtained by bagging and ECOC ensembles are always lower than those of the single classifier, and the reduction in the error is almost always a result of reductions both in *VE* and in *SE*. This observation means that the contributions of bias and (predictor) variance to the prediction error are smaller when ensembles are used (Fig. 1.3, Fig. 1.4). However, note that reductions in *VE* have greater magni-

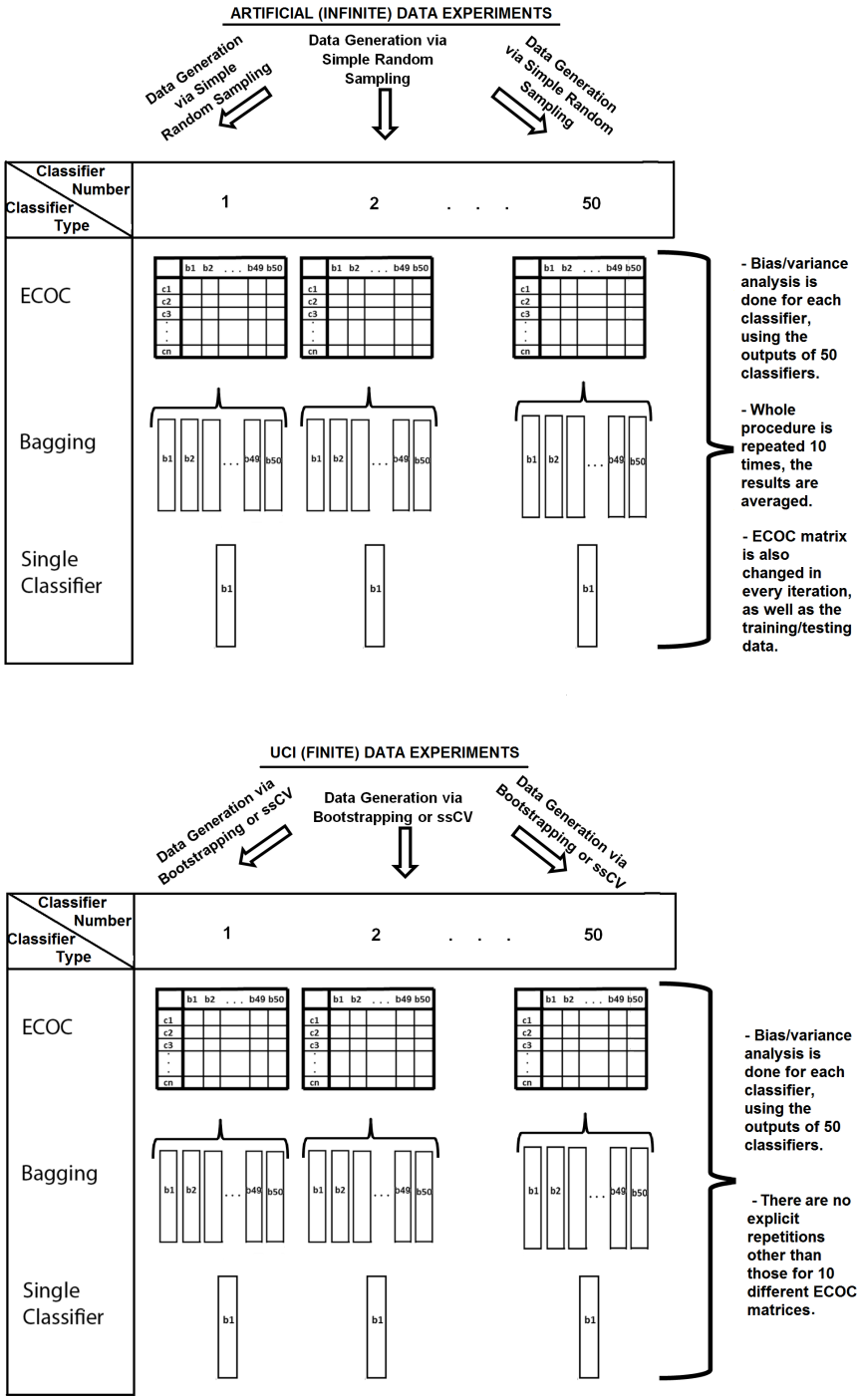

Fig. 1.2 Diagram illustrating the experimental setup for Artificial and Real (UCI MLR) datasets

tude, and in two-class problems, the reduction in *SE* is almost zero (Fig. 1.5). As for bias and variance themselves, it has been observed that ECOC and bagging induce reduction in both, especially in variance, in almost all the cases. The fact that NNs are high variance/low bias classifiers also plays a role in these observations, where the high variance is more easily reduced compared to the already lower bias and *VE* is reduced more than *SE*. In [3] and [16], bagging and ECOC are also stated to have low variance in the additive error decomposition, and Kong-Dietterich framework [16] also acknowledges that ECOC reduces variance.

Convergence to the Prediction Error

It is observed that the convergence of bagging ensemble to the optimal prediction error is usually achieved at a lower number of epochs compared to those of single classifier; and ECOC ensemble convergence is often at lower epochs than bagging (Fig. 1.3, Fig. 1.4, Fig. 1.5). The prediction errors are also in the same descending order: single classifier, bagging and ECOC; except when complex networks with high number of nodes and epochs are used. Under these circumstances *VE*, *SE*, and therefore the prediction errors of both ECOC and bagging are similar. However, it should also be noted that ECOC outperforms bagging in sense of speed due to the fact that it divides multi-class into multiple two-class problems.

It is also almost always the case that the prediction error of ECOC converges to its optimum in 2 nodes, whereas a single classifier requires higher number of nodes. Moreover, for ECOC, the optimal number of epochs is also lower than or equal to that of the single classifier. In other words, compared to a single classifier trained with high number of epochs and nodes, an ECOC can yield better results with fewer nodes and epochs. The trend is similar when bagging is considered; usually standing between the single classifier and ECOC in sense of accuracy and convergence.

Bias/Variance versus SE/VE

For the single classifier we see that *VE* does not necessarily follow the trend of variance. This happens especially when the number of nodes and epochs is small, that is when the network is relatively weak (Fig. 1.3, Fig. 1.4). In this scenario, the variance decreases while *VE* increases. This is actually an expected observation as having high variance helps hitting the right target class when the network is relatively less decisive. However, ensemble methods do not show this property as much as the single classifier. A possible explanation might be that each ensemble classifier already makes use of variance coming from its base classifiers; and this compensates for the decrease in *VE* of single classifiers with high variance, in weak networks. Therefore, more variance among ensemble classifiers does not necessarily help having less *VE*.

In the above mentioned scenario of *VE* showing an opposite trend of variance, the bias-variance trade-off can be observed. At the points where the *VE* increases,

SE decreases to reveal an overall decrease in the prediction error. However, these points are not necessarily the optimal points in terms of the prediction error; the optima are mostly where there is both *VE* and *SE* reduction (Fig. 1.4). Apart from this case, bias and variance are mostly correlated with *SE* and *VE* respectively (Fig. 1.4, Fig. 1.5). This is also pointed out in [13].

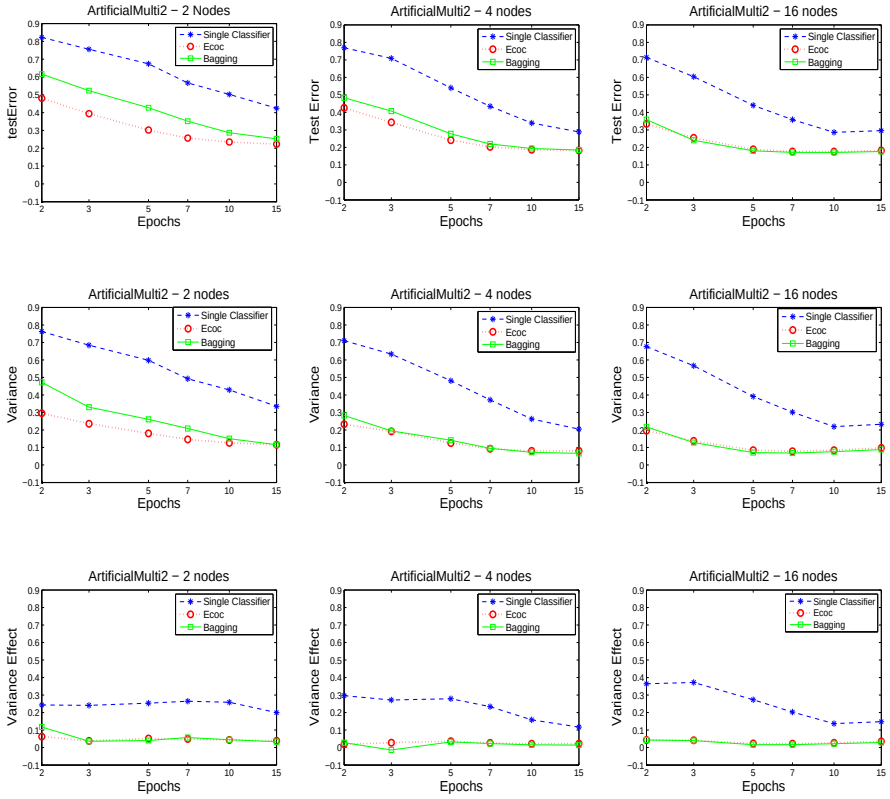

Fig. 1.3 Bias-Variance Analysis for ArtificalMulti2 data. First Row: Overall prediction error. Second Row: Variance. Third Row: Variance effect. First Column: For 2 Nodes. Second Column: For 4 Nodes. Third Column: For 16 Nodes.
Dashed blue lines (starred) indicate the results for single classifier, dotted red (circled) for ECOC and solid green (squared) for bagging.

1.4 Discussion

By analysing bagging, ECOC and single classifiers consisting of NNs through the bias-variance definition of James, we have found some clear trends and relation-

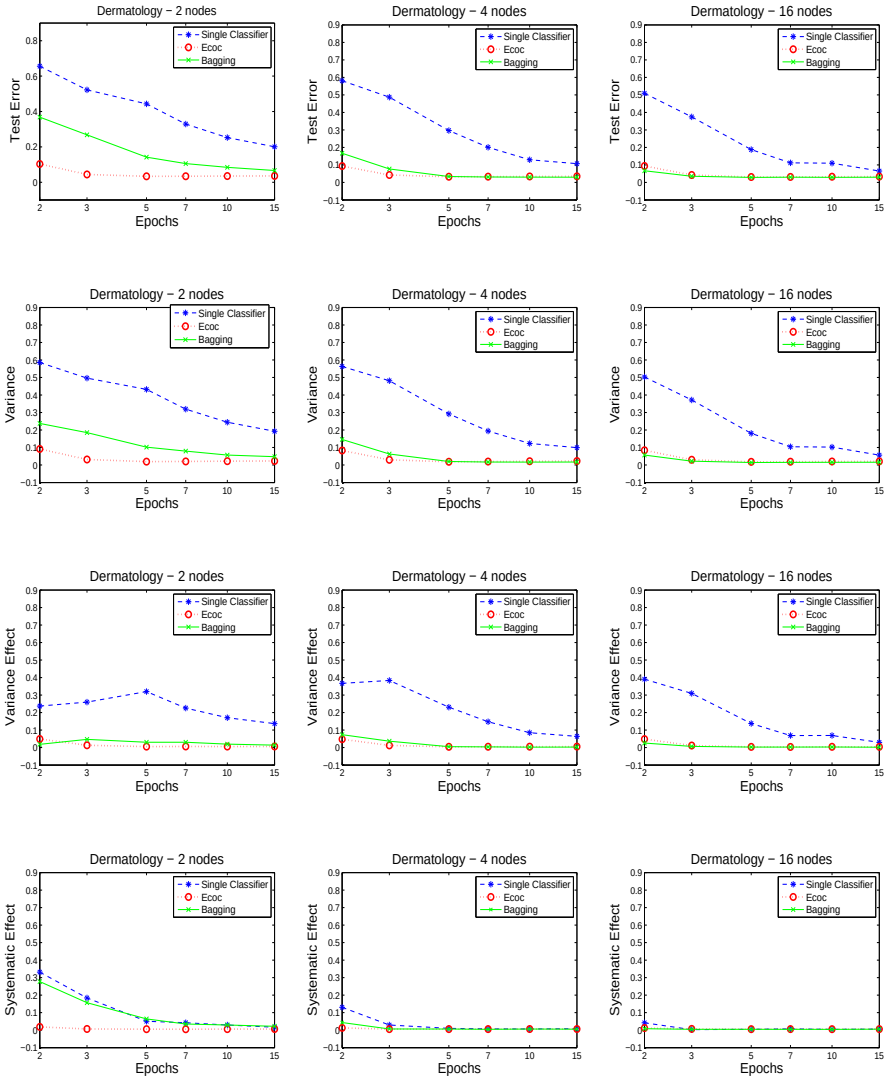

Fig. 1.4 Bias-Variance Analysis for Dermatology data. First Row: Overall prediction error. Second Row: Variance. Third Row: Variance effect. Fourth Row: Systematic effect. First Column: For 2 Nodes. Second Column: For 4 Nodes. Third Column: For 16 Nodes.
Dashed blue lines (starred) indicate the results for single classifier, dotted red (circled) for ECOC and solid green (squared) for bagging.

ships that offer hints to be used in classifier design. For multi-class classification problems, the increase in the overall prediction performance obtained with ECOC makes it preferable over single classifiers. The fact that it converges to the optimum with smaller number of nodes and epochs is yet another advantage. It also outperforms bagging mostly, while in other cases gives similar results. As for the two-class problems, bagging always outperforms the single classifier, and the optimum number of nodes and epochs is relatively smaller.

The increase in the performance of bagging and ECOC is a result of the decrease in both variance effect and systematic effect, although the reductions in the magnitude of the variance effect are bigger. Also, when the NNs are weak, that is when they have been trained with few nodes and epochs, we see that the trends of variance and variance effect might be in opposite directions in the single classifier case. This implies that having high variance might help improve the classification performance in weak networks when single classifiers are used. However, they are still outperformed by ensembles, which have even lower variance effects.

As for further possible advantages of ensembles, the fact that they are expected to avoid overfitting might be shown by using more powerful NNs with higher number of nodes, or other classifiers such as SVMs that are more prone to overfitting. Future work is also aimed at understanding and analysing the bias-variance domain within some mathematical frameworks such as [21, 20] and using the information in the design of ECOC matrices.

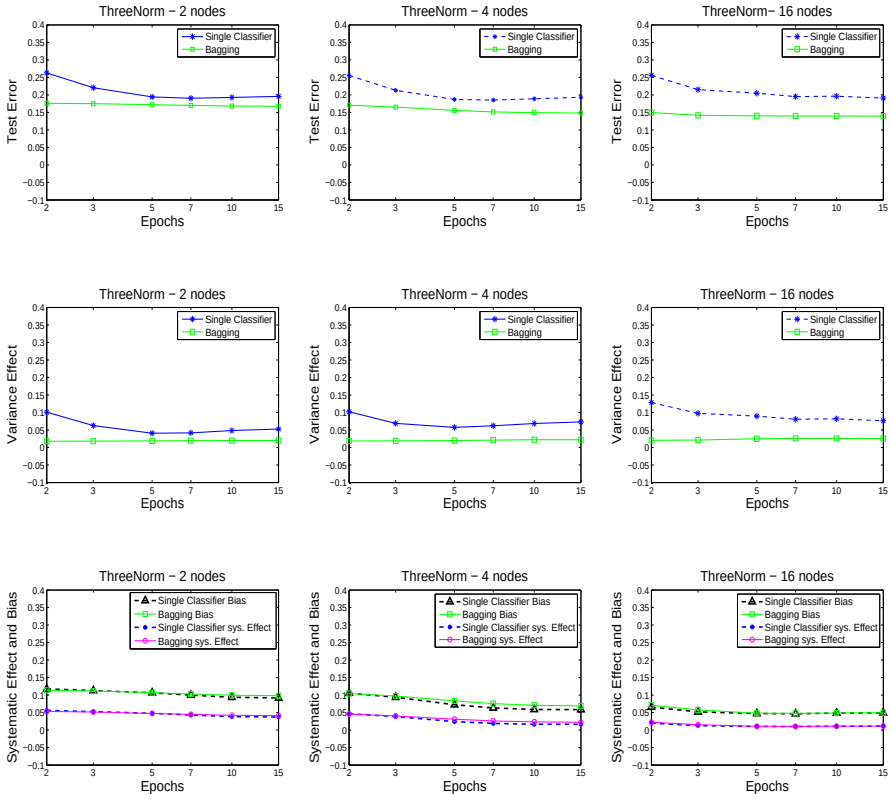

Fig. 1.5 Bias-Variance Analysis for ThreeNorm data. First Row: Overall prediction error. Second Row: Variance effect. Third Row: Systematic effect and Bias. First Column: For 2 Nodes. Second Column: For 4 Nodes. Third Column: For 16 Nodes.
In the first two rows, dashed blue lines (starred) indicate the results for single classifier, and solid green (squared) for bagging. In the third row, dashed black (with triangles) & dashed blue (starred) lines indicate the results for single classifier bias and systematic effect respectively; and solid green (squared) & magenta (circled) for those of bagging.

References

1. Allwein, E., Schapire, R., Singer, Y.: Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers. Journal of Machine Learning Research 1. 113–141 (2002)
2. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository, http://www.ics.uci.edu/~mlearn/MLRepository.html. School of Information and Computer Science, University of California, Irvine, CA (2007)
3. Breiman L.: Arcing classifiers. The Annals of Statistics, 26(3), 801–849 (1998)
4. Dietterich, T.G., Bakiri, G.: Solving multi-class learning problems via Error-Correcting Output Codes. J. Artificial Intelligence Research 2. 263–286 (1995)
5. Domingos, P.: Why does bagging work? A Bayesian account and its implications. Proceedings of the Third International Conference on Knowledge Discovery and Data

Mining, pp. 155–158. AAAI Press, Newport Beach, California, USA (1997)
6. Domingos. P.: A Unified Bias-Variance Decomposition for Zero-One and Squared Loss. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence, pp. 564–569. The MIT Press, Austin, Texas, USA (2000)
7. Escelara, S., Tax, D. M. J., Pujol, O., Radeva, P. and Duin R.P.W.: Subclass Problem-Dependent Design for Error-Correcting Output Codes. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(6), pp. 1041–1054 (2008)
8. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm.In: Proceedings Thirteenth International Conference on Machine Learning, pp. 148–156. Morgan Kaufmann, Bari, Italy (1996)
9. Friedman, J. H.: On bias, variance, 0/1 loss and the curse of dimensionality. Data Mining and Knowledge Discovery, 1, pp. 55–77 (1997)
10. Geman, S., Bienenstock, E., Doursat R.: Neural networks and the bias/variance dilemma. Neural Computation, vol. 4, no. 1, pp. 1–58 (1992)
11. Heskes, T.: Bias/Variance Decomposition for Likelihood-Based Estimators. Neural Computation, 10, pp. 1425–1433 (1998)
12. James, G. M.: Majority Vote Classifiers: Theory and Applications. PhD Thesis, Department of Statistics, University of Standford (1998)
13. James, G.: Variance and Bias for General Loss Functions. Machine Learning, 51(2), 115–135 (2003)
14. James, G. M., Hastie, T.: The Error Coding Method and PICT's. Computational and Graphical Statistics, vol. 7, no. 3, pp. 377-387 (1998)
15. Kohavi, R., & Wolpert, D. H.: Bias plus variance decomposition for zero-one loss functions. In: Proceedings Thirteenth International Conference on Machine Learning, pp. 275–283. Morgan Kaufmann, Bari, Italy (1996)
16. Kong, E. B., Dietterich, T. G.: Error-correcting output coding corrects bias and variance. In: Proceedings of the Twelfth International Conference on Machine Learning, pp. 313–321. ACM, New Orleans, LA, USA (1995)
17. Schapire, R. E., Freund, Y., Bartlett, P., Lee, W. S.: Boosting the margin: a new explanation for the effectiveness of voting methods. The Annals of Statistics, 26(5), pp. 1651–1686 (1998)
18. Smith, R. S., Windeatt, T.: The bias variance trade-off in bootstrapped Error correcting Output Code ensembles. Workshop Multiple Classifier Systems, pp. 1–10. Springer-Verlag, Reykjavik, Iceland (2009)
19. Tibshirani, R.: Bias, variance and prediction error for classification rules. Technical Report, University of Toronto, Toronto, Canada (1996)
20. Tumer, K., Ghosh, J.: Analysis of decision boundaries in linearly combined neural classifiers. Pattern Recognition, 29(2), pp. 341–348 (1996)
21. Tumer, K., Ghosh, J.: Error correlation and error reduction in ensemble classifiers. Connection Science 8 (3-4), pp. 385–403 (1996)
22. Valentini, G., Dietterich, T.: 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)
23. Webb, G.I., Conilione, P.: Estimating bias and variance from data. Technical Report (2005)
24. Windeatt, T., Ghaderi R.: Coding and Decoding Strategies for Multi-class Learning Problems. Information Fusion, 4(1), pp. 11–21 (2003)
25. Wolpert, D. H.: On bias plus variance. Neural Computation. 9, pp. 1211–1244 (1996)