A Bias-Variance Analysis of Bootstrapped Class-Separability Weighting for 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

We investigate the effects, in terms of a bias-variance decomposition of error, of applying class-separability weighting plus bootstrapping in the construction of error-correcting output code ensembles of binary classifiers. Evidence is presented to show that bias tends to be reduced at low training strength values whilst variance tends to be reduced across the full range. The relative importance of these effects, however, varies depending on the stability of the base classifier type.

1. Introduction

When considering the errors made by statistical learning systems it is useful to group them under three headings: Bayes error, bias (actually measured as bias^2) and variance. The first of these is due to unavoidable noise but the latter two can be reduced by careful design. There is often a tradeoff between bias and variance [4] so that a high value of one implies a low value of the other.

In this paper we look at the bias and variance behaviour that is observed when error-correcting output code classifier ensembles are enhanced by the application of bootstrapping and class-separability weighting. Section 2 describes these concepts in more detail and section 3 shows the results of performing experiments on 11 multi-class UCI datasets; finally, in section 4 we present the conclusions to be drawn from this work.

2. Method

The use of error-correcting output code (ECOC) ensembles [2, 5] has proved to be highly successful in solving multi-class classification problems. 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. 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. The operation of the ECOC algorithm can be broken down into two distinct stages. The coding stage consists of applying the base classifiers to the input pattern $\mathbf{x}$ so as to construct vector of base classifier outputs $\mathbf{s}(\mathbf{x})$. The decoding stage consists of applying some decoding rule to this vector so as to make an estimate of the class label that should be assigned to the input pattern.

A generally desirable property of multiple classifier systems, of which ECOC is an example, is that there should be diversity among the individual classifiers in the ensemble [1]. One way of encouraging this is to apply *bootstrapping* [3] to the training set so that each base classifier is trained on a unique bootstrap replicate. These replicates are obtained from a given training set by repeated sampling with replacement and this results in further training sets which have, on average, 63% of the patterns in the original set but with some patterns repeated to form a set of the same size. Previous work [8] has shown that bootstrapping can reduce ECOC ensemble error and, in particular, it tends to avoid the problem of variance caused by over fitting the data at high training strength values[1]. A further potential benefit of bootstrapping is that the out-of-bootstrap (OOB) sample for each base classifier can be used for other purposes such as parameter tuning.

A commonly used ECOC decoding method is to base the classification decision on the minimum distance between $\mathbf{s}(\mathbf{x})$ and the vector of target outputs for each of the classes, using a distance metric such as Hamming or L^1. This, however, treats all base clas-

[1] By “training strength” we mean the parameter that controls the amount of effort that is put into training a base classifier. E.g. for neural networks this is the number of training epochs and for SVMs it is the cost parameter.

```
Inputs: matrix of training patterns $\mathbf{T} \in \mathbb{R}^{P \times M}$, binary coding matrix $\mathbf{Z} \in \{0,1\}^{N \times L}$, trained ECOC coding function $E : \mathbb{R}^M \mapsto [0,1]^L$ .
Outputs: weight matrix $\mathbf{W} \in [0,1]^{N \times L}$ where $\sum_{j=1}^{L} \mathbf{W}_{ij} = 1$, for $i = 1 \ldots N$.
Apply $E$ to each row of $\mathbf{T}$ and round to give prediction matrix $\mathbf{H} \in \{0,1\}^{P \times L}$.
Initialise $\mathbf{W}$ to $\mathbf{0}$.
for $c$ = 1 to $N$
  for $i$ = indices of training patterns belonging to class $c$
    for $j$ = indices of training patterns not belonging to class $c$
      let $d$ be the true class of the pattern $\mathbf{T}_j$.
      for k = 1 to $L$
        if $\mathbf{H}_{ik} = \mathbf{Z}_{ck}$ and $\mathbf{H}_{\mathrm{jk}} = \mathbf{Z}_{\mathrm{dk}}$, add 1 to $\mathbf{W}_{ck}$
          as both predictions for $\mathbf{T}_i$ and $\mathbf{T}_j$ are correct.
        if $\mathbf{H}_{ik} \neq \mathbf{Z}_{ck}$ and $\mathbf{H}_{\mathrm{jk}} \neq \mathbf{Z}_{\mathrm{dk}}$, subtract 1 from$\mathbf{W}_{ck}$
          as both predictions for $\mathbf{T}_i$ and $\mathbf{T}_j$ are incorrect.
      end
    end
  end
end
Reset all negative entries in $\mathbf{W}$ to 0.
Normalize $\mathbf{W}$ so that each row sums to 1.
```

Figure 1. Pseudo-code for computing the class-separability weight matrix.

sifiers as equal, and takes no account of variations in their reliability. In this paper we make use of a method known as class-separability weighting (CSEP). CSEP has been shown to improve ECOC accuracy, particularly when combined with bootstrapping (BS) [9, 10].

The CSEP weight, for each base classifier and target class combination, is computed by taking the difference between the counts of the accuracy and error correlations observed when the base classifier is applied to training patterns belonging to, and not belonging to, the given class. This procedure is detailed in Fig. 1.

3. Experiments

In this section we present the results of performing classification experiments on 11 multi-class datasets (see Table 1) obtained from the publicly available UCI repository [7].

For each dataset, ECOC ensembles of size 200 were constructed using a range of capacity[2] and training strength parameters. The base classifier types employed were multi-layer perceptron (MLP) neural networks, support vector machines (SVMs) with Gaussian kernel and SVMs with polynomial kernel. Each such combination was repeated 10 times with and without CSEP weighting and with and without bootstrapping. Each run used a different randomly chosen stratified training set and a different randomly generated ECOC coding matrix; for neural network base classifiers another source of random variation was the initial network weights. Each bootstrapped base classifier was trained on a separate bootstrap replicate drawn from the complete training set for that run. The CSEP weight matrix was, in all cases, computed from the full training set. 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. In measuring bias and variance we adopted the definitions of Kohavi and Wolpert [6].

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

Figure 2 shows the mean, over ten datasets[3], of variance and bias at a range of training strength parameter values. To allow a fair comparison, each dataset was evaluated at its respective optimal capacity parameter and the error curves were re-scaled so as to equalise the mean unmodified ensemble error across all datasets. The graphs show the effect of individually applying no modifications, bootstrapping, CSEP weighting and the CSEP+BS combination.

A number of comments may be made about Fig. 2. Firstly it should be noted that, when compared with the unmodified ensemble, the general effect of the CSEP+BS combination is to reduce variance across the full range of training strengths and to reduce bias at low training strength values. There is, however, a difference in degree, depending on which base classifier type

[2] The term *capacity* refers to the degree to which a classifier is able to model complex class boundaries.

[3] *thyroid* was omitted from Fig. 2 due to the fact that CSEP weighting produced an uncharacteristically large bias at low training strengths. We ascribe this anomalous behaviour to the fact that the set is extremely unbalanced, with 93% of samples belonging to just one of the three classes. Note however that, as shown in Fig. 3, CSEP+BS outperformed the unmodified ensemble at the optimal training strength.

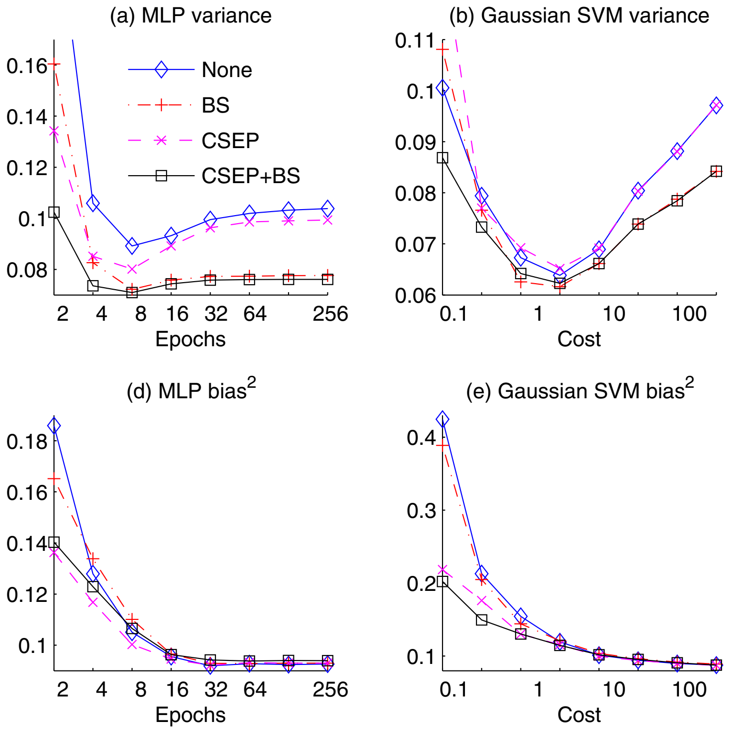

Figure 2. The effects of CSEP weighting and bootstrapping on bias and variance. The graphs show the weighted average error, taken over ten datasets, at a range of training strength values.

is used, with the MLP ensemble benefiting more than those based on SVMs. For the unstable MLP classifier, the main effect is one of variance reduction, with bias reduction being small or even slightly negative at all but the lowest training strengths. By contrast, for the stable SVM classifiers, variance reduction is less pronounced, especially where the training strength is close to the optimal value; bias reduction, however, is rather larger than that for MLPs and occurs over a wider range of training strengths.

An explanation of how CSEP+BS achieves these reductions in error may be obtained by looking at the individual graphs for bootstrapping and CSEP weighting on their own. It is apparent that the variance reduction at high training strengths is largely due to the effect of bootstrapping since the curve for CSEP+BS closely follows that for bootstrapping alone. This is to be expected, given that bootstrapping is known to reduce variance at high training strengths. Similarly, bias reduction at low training strengths may be ascribed to the effect of CSEP weighting because the two curves are again closely aligned. Again, this is reasonable since the effect of CSEP should be to reduce bias by giving more weight to accurate base classifiers. One surprising observation is that the CSEP+BS combination reduces variance at very low training strengths even when neither technique on its own appears to produce this effect sufficiently strongly. Indeed, in the case of the SVM classifiers the individual methods can actually increase variance but the combination of methods still tends to reduce it (or at least prevent its increase, as in the case of the polynomial SVM). We attribute this behaviour to the fact that the weights matrix for the CSEP+BS combination is made more reliable by being trained, in part, on independent data (i.e. the OOB set). It is thought that this acts to reduce the arbitrariness of decisions made by a lightly-trained ensemble.

For any given dataset, the benefits (or otherwise) of using CSEP+BS can best be determined by looking at the performance of the ensemble at its respective optimal base classifier parameter settings, both with and without the techniques being applied. In general this may mean that the capacity as well as training strength parameters differ. Fig. 3 shows the relative reduction in bias and variance at these optimal values brought about by CSEP+BS.

It can be observed that the results are broadly in accord with the findings from Fig. 2, with the MLPs

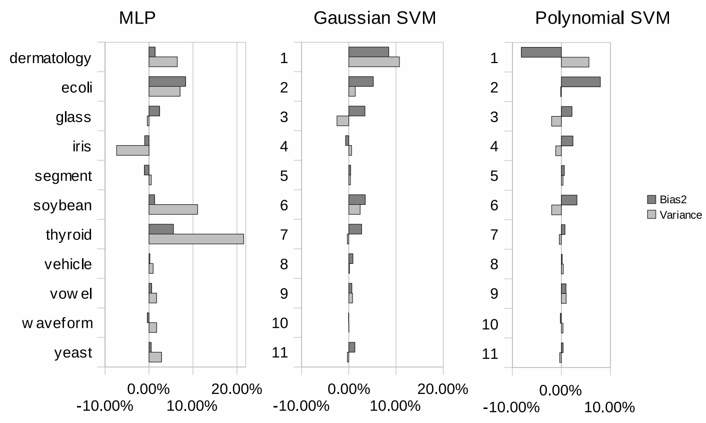

Figure 3. The relative reductions (positive values) in bias and variance error obtained by applying CSEP weighting and bootstrapping. Comparisons are made at the respective optimal base classifier parameter settings both with and without CSEP+BS.

tending to show larger benefits than the SVM ensembles. For MLP base classifiers the main gain is seen to be from variance reduction, with bias reduction playing a lesser role. In only one case (*iris*), was there any significant worsening of variance error. The situation for SVMs is reversed, with bias reduction being more consistently observed; there was only one case (*dermatology* with a polynomial SVM) where bias error increased significantly. In several instances a modest increase in variance was offset by a reduction in bias.

4. Discussion and Conclusions

In this paper we have examined the effect on bias and variance error of applying CSEP weighting and bootstrapping to ECOC ensembles. Evidence has been presented to show that these two techniques complement each other so that, when combined, they tend to reinforce each others strengths whilst avoiding their weaknesses. This tends to lead to a reduction in variance across the full range of training strengths and a reduction in bias at lower training strengths.

At optimal parameter settings, the relative importance of these effects vary depending on the nature of the base classifiers used. For unstable MLP classifiers the main benefit is one of variance reduction, whilst for the more stable SVM classifiers bias reduction tends to predominate.

5. Acknowledgements

This work was supported by EPSRC grant E061664/1.

References

[1] Brown G, Wyatt J, Harris R, Yao X. Diversity Creation Methods: A Survey and Categorisation. Journal of Information Fusion, 6(1), 2005.

[2] Dietterich TG, Bakiri G. Solving Multiclass Learning Problems via Error-Correcting Output Codes. *Journal of Artificial Intelligence Research* 2: 263-286, 1995.

[3] Efron B, Tibshirani RJ. An Introduction to the Bootstrap. Chapman & Hall, 1993.

[4] Geman S, Bienenstock E. Neural networks and the bias / variance dilemma. Neural Computation, 4:1-58, 1992.

[5] James G. Majority Vote Classifiers: Theory and Applications. PhD Dissertation, Stanford University, 1998.

[6] Kohavi R, Wolpert D. Bias plus variance decomposition for zero-one loss functions. *Proc. 13th International Conference on Machine Learning*, pp. 275-283, 1996.

[7] Merz CJ, Murphy PM. UCI Repository of Machine Learning Databases, 1998. http://www.ics.uci.edu/~mlearn/MLRepository.html.

[8] Smith RS, Windeatt T. The Bias Variance Trade-off in Bootstrapped Error Correcting Output Code Ensembles. *Proc. 8th International Conf. on Multiple Classifier Systems*, pp. 1-10, July, 2009.

[9] Smith RS, Windeatt T. Class-Separability Weighting and Bootstrapping in Error Correcting Output Code Ensembles. *Proc. 9th International Conf. on Multiple Classifier Systems*, April 2010, accepted.

[10] Windeatt T, Smith RS, Dias K. Weighted Decoding ECOC for Facial Action Unit Classification. *18th European Conference on Artificial Intelligence* (ECAI), pp. 26-30, Patras, Greece, July 2008.