# Class-Separability Weighting and Bootstrapping in 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.** A method for applying weighted decoding to error-correcting output code ensembles of binary classifiers is presented. This method is sensitive to the target class in that a separate weight is computed for each base classifier and target class combination. Experiments on 11 real-world datasets show that the method tends to improve classification accuracy when using neural network or support vector machine base classifiers. It is further shown that weighted decoding combines well with the technique of bootstrapping to improve classification accuracy still further.

## 1 Introduction

The use of error-correcting output code (ECOC) ensembles [5,8] 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. 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.

The operation of the ECOC algorithm can be broken down into two distinct stages - the coding stage and the decoding stage. 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 commonly used 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 classifiers as equal, and takes no account of variations in their reliability. In this paper we describe a method for weighting the base classifier outputs so as to obtain improved ensemble accuracy. The weighting coefficients are computed from a statistic, known as the class-separability statistic. This algorithm assigns different weights to each base classifier and target class combination. Class-separability weighting (CSEP) was shown in [12] to be useful in the field of face-expression recognition. Here we show that it can also be beneficial when applied to general classification problems, as exemplified by 11 UCI datasets [9].

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. In this paper we investigate experimentally three types of base classifier, namely multi-layer perceptron (MLP) neural networks [1], Gaussian kernel support vector machines (SVMs) and polynomial kernel SVMs [3]. It is useful to regard each of these base classifier types as being controlled by two main parameters which respectively determine the *capacity* and the *training strength* of the learning algorithm. The term *capacity* [3] 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. For the three types of base classifier considered, the capacity parameter is, respectively, the number of hidden nodes, the Gaussian gamma parameter and the polynomial degree parameter. The training strength parameter is the number of training epochs for MLPs and the cost parameter for both types of SVMs.

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 [2,11]. 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 encouraging this is to apply *bootstrapping* [7] to the training set so that each base classifier is trained on a unique bootstrap replicate. These are obtained 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. Previous work [10] has shown that bootstrapping often reduces ensemble error and, in particular, it tends to avoid the problem of overfitting the data at high training strength values.

The remainder of this paper is structured as follows. The technique of applying class-separability weighting to the decoding of outputs from ECOC ensembles is described in detail in section 2. An experimental investigation of the effect of using this weighting scheme, with and without bootstrapping, is presented in section 3. Finally, section 4 summarises the conclusions to be drawn from this work.

## 2 ECOC Weighted Decoding

The ECOC method consists of repeatedly partitioning the full set of $N$ classes $\Omega$ into $L$ super-class pairs. The choice of partitions is represented by an $N \times L$ binary *coding matrix* $\mathbf{Z}$. The rows $\mathbf{Z}_i$ are unique *codewords* that are associated with the individual target classes $\omega_i$ and the columns $\mathbf{Z}^j$ represent the different super-class partitions. Denoting the $j$th super-class pair by $\mathrm{S}^j$ and $\overline{\mathrm{S}^j}$, element $Z_{ij}$ of the coding matrix is set to 1 or 0[1] depending on whether class $\omega_i$ has been put into $\mathrm{S}^j$ or its complement. A separate base classifier is trained to solve each of these 2-class problems.

Given an input pattern vector $\mathbf{x}$ whose true class $y(\mathbf{x}) \in \Omega$ is unknown, let the soft output from the $j$th base classifier be $s_j(\mathbf{x}) \in [0,1]$. The set of outputs from all the classifiers can be assembled into a vector $\mathbf{s}(\mathbf{x}) = [s_1(\mathbf{x}), \ldots, s_L(\mathbf{x})]^{\mathrm{T}} \in [0,1]^L$ called the *output code* for $\mathbf{x}$. Instead of working with the soft base classifier outputs, we may also first harden them, by rounding to 0 or 1, to obtain the binary vector $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), \ldots, h_L(\mathbf{x})]^{\mathrm{T}} \in \{0,1\}^L$. The principle of the ECOC technique is to obtain an estimate $\hat{y}(\mathbf{x}) \in \Omega$ of the class label for $\mathbf{x}$ from a knowledge of the output code $\mathbf{s}(\mathbf{x})$ or $\mathbf{h}(\mathbf{x})$.

In its general form, a *weighted* decoding procedure makes use of an $N \times L$ weights matrix $\mathbf{W}$ that assigns a different weight to each target class and base classifier combination. The class decision, based on the $L^1$ metric, is made as follows:

$$\hat{y}(\mathbf{x}) = \arg\min_{\omega_i} \sum_{\mathrm{j}=1}^{\mathrm{L}} \mathbf{W}_{\mathrm{ij}} \left| \mathrm{s}_{\mathrm{j}}(\mathbf{x}) - \mathbf{Z}_{\mathrm{ij}} \right|, \tag{1}$$

where it is assumed that the rows of $\mathbf{W}$ are normalized so that $\sum_{j=1}^{L} \mathbf{W}_{ij} = 1$ for $i = 1 \ldots N$. If the base classifier outputs $s_j(\mathbf{x})$ in eqn. 1 are replaced by hardened values $h_j(\mathbf{x})$ then this describes the weighted Hamming decoding procedure.

The values of $\mathbf{W}$ may be chosen in different ways. For example, if $\mathbf{W}_{ij} = \frac{1}{L}$ for all $i, j$ then the decoding procedure of eqn. 1 is equivalent to the standard unweighted $L^1$ or Hamming decoding scheme. In this paper we make use of the class separability measure [11,12] to obtain weight values that express the ability of each base classifier to distinguish members of a given class from those of any other class.

In order to describe the class-separability weighting scheme, the concept of a correctness function must first be introduced: given a pattern $\mathbf{x}$ which is known to belong to class $\omega_i$, the correctness function for the $j$'th base classifier takes the value 1 if the base classifier makes a correct prediction for $\mathbf{x}$ and 0 otherwise:

$$C_j(\mathbf{x}) = \begin{cases} 1 & \text{if } h_j(\mathbf{x}) = \mathbf{Z}_{ij} \\ 0 & \text{if } h_j(\mathbf{x}) \neq \mathbf{Z}_{ij} \end{cases}. \tag{2}$$

We also consider the complement of the correctness function $\overline{C}_j(\mathbf{x}) = 1 - C_j(\mathbf{x})$ which takes the value 1 for an incorrect prediction and 0 otherwise.

[1] Alternatively, the values +1 and -1 are often used.

For a given class index $i$ and base classifier index $j$, the class-separability weight measures the difference between the positive and negative correlations of base classifier predictions, ignoring any base classifiers for which this difference is negative:

$$\mathbf{W}_{ij} = \max\left\{0, \frac{1}{K_i}\left[\sum_{\substack{\mathbf{p}\in\omega_i \\ \mathbf{q}\notin\omega_i}} C_j(\mathbf{p})\,C_j(\mathbf{q}) - \sum_{\substack{\mathbf{p}\in\omega_i \\ \mathbf{q}\notin\omega_i}} \overline{C}_j(\mathbf{p})\,\overline{C}_j(\mathbf{q})\right]\right\}, \quad (3)$$

where patterns $\mathbf{p}$ and $\mathbf{q}$ are taken from a fixed training set $T$ and $K_i$ is a normalization constant that ensures that the $i$'th row of $\mathbf{W}$ sums to 1. An algorithm for computing $\mathbf{W}$ is summarised in fig. 1.

**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 the predictions for both patterns $\mathbf{T}_i$ 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 the predictions for both patterns $\mathbf{T}_i$ 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.

**Fig. 1.** Pseudo-code for computing the class-separability weight matrix for ECOC.

## 3 Experiments

In this section we present the results of performing classification experiments on 11 multi-class datasets obtained from the publicly available UCI repository [9]. 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 capacity and training strength parameters. Each such combination was repeated 10 times with and without CSEP weighting and with and without bootstrapping. In total this led to 56,000 experimental runs being performed. 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. When bootstrapping was used, each 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. In each run the data was normalized so that the training set had 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.

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 MLPs were constructed as a single hidden layer of perceptrons, with the number of hidden nodes ranging from 2 to 16 and the number of training epochs from 2 to 1024. For Gaussian SVMs the width parameter gamma 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.

Table 2 compares the effect, on ensemble generalisation accuracy, of using CSEP weighted decoding and bootstrapping in different combinations. For each such combination and each base-classifier algorithm it shows the number of datasets for which rank 1 accuracy was achieved (with the scores being divided in the case of two or more equally ranked classifiers). It also shows the mean ranking, taken over the 11 datasets, achieved by each combination together with

the mean best-case ensemble error and the percentage reduction in this error[2]. The evidence of this table is that both bootstrapping and CSEP weighting on their own do tend to produce some improvement in classifier accuracy, with the latter algorithm being somewhat more effective than the former. This is shown by higher rank 1 counts, lower mean rank values and lower test errors. It is striking, however, that the greatest benefit is obtained when *both* techniques are used, indicating that their effects are mutually complementary so that they can be combined to good effect. It is also noticeable that, perhaps due to its more stochastic nature, the MLP base classifier shows the greatest reduction in mean test error, with the deterministic SVM classifiers benefitting to a lesser degree.

**Table 2.** Comparison of the merits of four combinations of algorithms, namely standard ECOC, ECOC with bootstrapping (BS), ECOC with CSEP weighted decoding and ECOC with both bootstrapping and weighted decoding.

| | Standard | BS | CSEP | BS+CSEP |
|---|---|---|---|---|
| Rank 1 count | | | | |
| MLP | 0 | 1 | 3 | **7** |
| Gaussian SVM | 0.5 | 1.5 | 2.5 | **6.5** |
| Polynomial SVM | 1 | 1 | 3 | **6** |
| Mean rank | | | | |
| MLP | 3.18 | 2.82 | 2.36 | **1.64** |
| Gaussian SVM | 3.09 | 2.91 | 2.18 | **1.55** |
| Polynomial SVM | 3.18 | 2.55 | 2.55 | **1.73** |
| Mean best-case ensemble test error (%) | | | | |
| MLP | 16.54 | 16.23 | 16.16 | **15.78** |
| Gaussian SVM | 15.84 | 15.88 | 15.77 | **15.65** |
| Polynomial SVM | 16.98 | 16.81 | 16.88 | **16.63** |
| Relative decrease in mean test error (%) | | | | |
| MLP | - | 1.87 | 2.30 | **4.59** |
| Gaussian SVM | - | -0.25 | 0.44 | **1.20** |
| Polynomial SVM | - | 1.00 | 0.59 | **2.06** |

Further evidence for these findings can be seen in Fig. 2. This shows the mean ensemble test error, taken over all datasets, at the optimal base classifier capacity and over a range of training strength values. It is immediately apparent, from an inspection of this figure, that the best results tend to be obtained using CSEP weighted decoding and bootstrapping in combination. Bootstrapping alone tends to reduce ensemble error and also makes the ensemble less susceptible to overtraining at high values of the training strength parameter. When CSEP weighting is added to bootstrapping there is a further consistent reduction in ensemble error over the range of training strength values. This improvement tends to be most pronounced at low values of training strength, but is still observ-

[2] Calculated as 100 x (original error - new error)/original error.

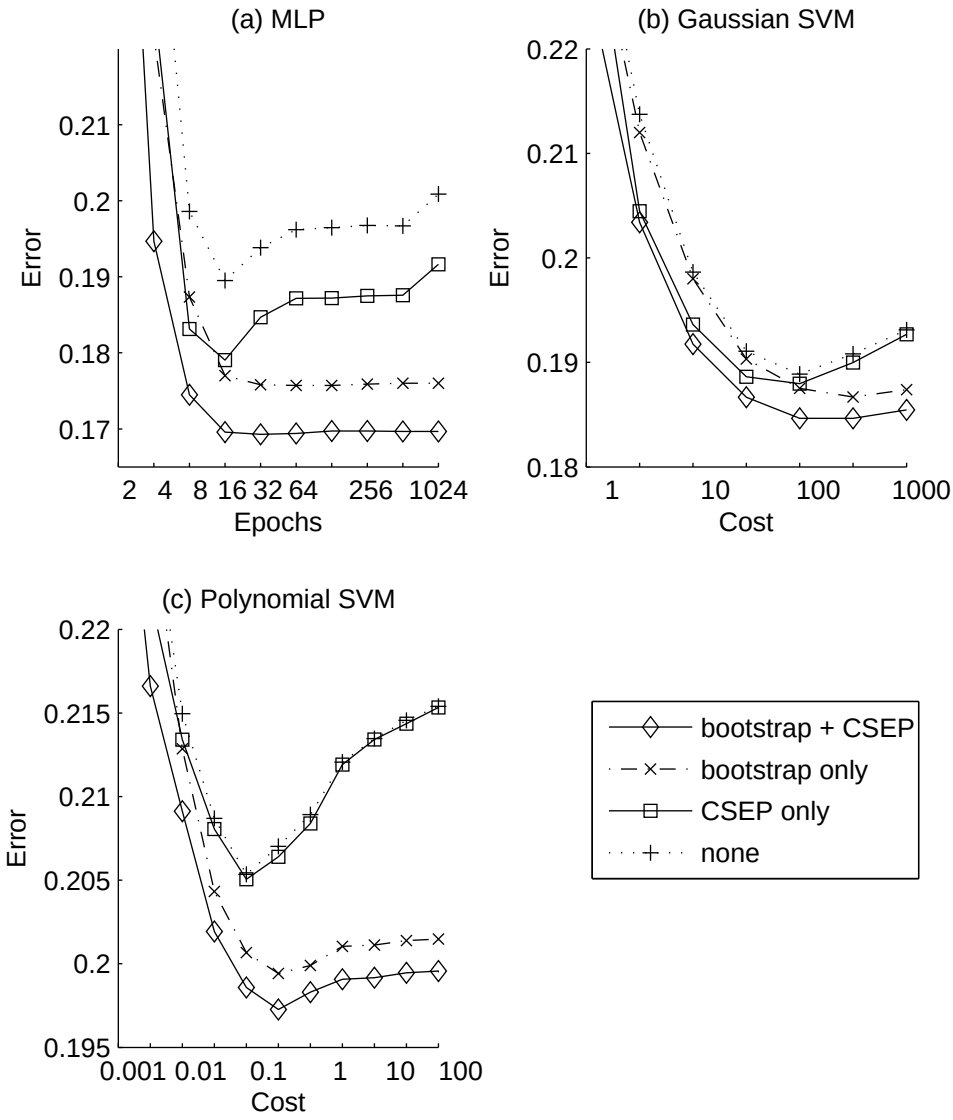


**Fig. 2.** The effects of class-separation weighting and bootstrapping on ensemble test error over a range of training strength values. These graphs show the mean error rate, taken over all datasets, at optimal base classifier capacity. The capacity parameters were (a) 8 hidden nodes, (b) gamma = 4, (c) degree = 2.

**Table 3.** Comparison of lowest ensemble error attained using standard ECOC and bootstrapped ECOC with weighted decoding (BS+CSEP). All values are expressed as percentages.

| Data Set | MLP | | | Gaussian SVM | | | Polynomial SVM | | |
|---|---|---|---|---|---|---|---|---|---|
| | Std. ECOC | BS + CSEP | relative decrease | Std. ECOC | BS + CSEP | relative decrease | Std. ECOC | BS + CSEP | relative decrease |
| dermatology | 4.86 | **3.07** | 36.83 | 2.97 | **2.90** | 2.35 | 3.21 | **2.97** | 7.56 |
| ecoli | 17.48 | **15.08** | 13.73 | 14.66 | **14.00** | 4.50 | 15.68 | **14.44** | 7.89 |
| glass | 37.16 | **36.64** | 1.40 | 35.76 | **35.69** | 0.22 | 38.57 | **37.71** | 2.22 |
| iris | 5.25 | **5.00** | 4.76 | 5.83 | **5.08** | 12.86 | **5.58** | 5.75 | -2.99 |
| segment | **3.92** | 3.94 | -0.51 | 5.64 | **5.59** | 0.96 | 6.08 | **5.69** | 6.50 |
| soybean | 9.39 | **9.04** | 3.73 | **7.90** | 8.06 | -2.10 | **8.24** | 8.25 | -0.02 |
| thyroid | 2.57 | **1.95** | 24.12 | 2.77 | **2.69** | 2.94 | 3.39 | **2.90** | 14.54 |
| vehicle | 22.22 | **20.76** | 6.57 | 22.38 | **22.04** | 1.52 | 23.66 | **23.08** | 2.44 |
| vowel | **21.14** | 22.42 | -6.05 | **20.83** | 20.86 | -0.12 | **25.85** | 25.86 | -0.05 |
| waveform | 16.70 | **14.75** | 11.68 | 14.48 | **14.41** | 0.45 | 14.59 | **14.45** | 0.97 |
| yeast | 41.22 | **40.97** | 0.61 | 41.02 | **40.85** | 0.41 | 41.90 | **41.83** | 0.18 |
| mean | 16.54 | **15.78** | 8.81 | 15.84 | **15.65** | 2.18 | 16.98 | **16.63** | 3.57 |

able at higher values of this parameter. In the absence of bootstrapping, CSEP weighting still leads to a reduction in ensemble error but the effect is more classifier dependent, with MLPs gaining the greatest benefit and polynomial SVMs the least. Again, the error reduction achieved by CSEP weighting is greatest at low values of training strength.

In the remainder of this section we look in more detail at the effects of applying bootstrapping and CSEP weighted decoding in combination. Table 3 shows the error levels measured on each of the test sets for each of the base classifier types when the base classifier parameters were optimised so as to minimise ensemble test error. Also shown is the percentage relative reduction in error achieved by bootstrapping plus CSEP weighting.

It can be seen from this table that, in the majority of cases (26/33), bootstrapping plus CSEP weighting did lead to a reduction in ensemble error. The size of this reduction was greatest when using an MLP base classifier but was nevertheless observable for the SVM base classifiers also.

There is also evidence that, for MLP base classifiers, bootstrapping plus CSEP weighted decoding has the desireable property that it tends to require simpler classifiers with fewer hidden nodes than standard ECOC. Table 4 shows the optimal numbers of hidden nodes, with and without bootstrapping plus weighted decoding. It can be seen that in 7/11 cases the former combination required fewer nodes, with the converse being true in only 2/11 cases. Also shown in table 4 is the number of training epochs required for optimal performance and it can be seen that the picture here is more evenly balanced between the two methods.

**Table 4.** Optimal numbers of hidden nodes and training epochs for MLP base classifiers with and without bootstrapping plus weighted decoding.

| Dataset | Nodes | | Epochs | |
|---|---|---|---|---|
| | Standard | BS+CSEP | Standard | BS+CSEP |
| dermatology | **4** | 8 | 16 | **4** |
| ecoli | 16 | **8** | 4 | 4 |
| glass | 16 | **8** | **8** | 64 |
| iris | 8 | **2** | 4 | 4 |
| segment | 16 | **8** | **32** | 64 |
| soybean | 8 | 8 | 4 | 4 |
| thyroid | 8 | **4** | **32** | 64 |
| vehicle | 8 | **4** | **8** | 16 |
| vowel | 8 | 8 | 128 | **32** |
| waveform | **4** | 16 | 8 | **4** |
| yeast | 8 | **4** | 8 | 8 |

## 4 Discussion and Conclusions

In this paper we have shown, by performing experiments on 11 real world multi-class datasets, that the techniques of bootstrapping and class-separability (CSEP) weighting each tend reduce ECOC ensemble error. Bootstrapping affects the coding stage; it tends to increase diversity and to make the ensemble resistant to overfitting, especially at high values of the training strength parameter. CSEP weighting affects the decoding stage by taking account of the different performances of the base classifiers with respect to each target class.

It has been shown that these two algorithms complement each other and thus combine well together to produce a greater reduction in ensemble error than either of the methods individually. One reason for this may be related to the fact that a side-effect of bootstrapping is to reduce the training set of each base classifier to a subset of the available training set. It seems likely that this benefits CSEP weighting because the weight matrix, which is calculated using the full training set, will tend to be less biased by virtue of the fact that some of the training patterns will not have been used for base classifier training. In effect this is similar to using a hold-out set for CSEP training.

The greatest benefit from CSEP weighting was observed when using MLPs as base classifiers. In this context it was also observed that the method has the desireable property that it tends to lead to simpler MLPs, requiring fewer hidden nodes for optimal performance. When deterministic base classifier algorithms such as SVMs were used, class-separability weighting was still found to be of benefit but to a lesser degree.

Future work will focus on characterizing how CSEP weighting improves performance in terms of a bias-variance decomposition of error.

## 5 Acknowledgements

This work was supported by EPSRC grant E061664/1. Thanks are also due to the providers of the prtools [6] and libsvm [4] software.

## References

1. Bishop MC. *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, 1995.
2. Brown G, Wyatt J, Harris R, Yao X. Diversity Creation Methods: A Survey and Categorisation. Journal of Information Fusion, 6(1), 2005.
3. Burges CJC. A Tutorial on Support Vector Machines for Pattern Recognition. *Knowledge Discovery and Data Mining*, 2(2), 1998.
4. Chih-Chung Chang, Chih-Jen Lin. LIBSVM : a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm
5. Dietterich TG, Bakiri G. Solving Multiclass Learning Problems via Error-Correcting Output Codes. *Journal of Artificial Intelligence Research* 2: 263-286, 1995.
6. Duin RPW, Juszczak P, Paclik P, Pekalska E, D. de Ridder, Tax DMJ, Verzakov S. PRTools 4.1, A Matlab Toolbox for Pattern Recognition, Delft University of Technology, 2007.
7. Efron B, Tibshirani RJ. An Introduction to the Bootstrap. Chapman & Hall, 1993.
8. James G. Majority Vote Classifiers: Theory and Applications. PhD Dissertation, Stanford University, 1998.
9. Merz CJ, Murphy PM. UCI Repository of Machine Learning Databases, 1998. http://www.ics.uci.edu/~mlearn/MLRepository.html.
10. 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.
11. Windeatt T. Accuracy/ Diversity and Ensemble Classifier Design, *IEEE Trans Neural Networks*, 17(4), July, 2006.
12. 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.