Accuracy/Diversity and Ensemble MLP Classifier Design

Terry Windeatt
Centre for Vision, Speech and Signal Proc (CVSSP), School of Electronics and Physical Sciences
University of Surrey, Guildford, Surrey, United Kingdom GU2 7XH
t.windeatt@surrey.ac.uk

Abstract: The difficulties of tuning parameters of MLP classifiers are well known. In this paper, a measure is described that is capable of predicting the number of classifier training epochs for achieving optimal performance in an ensemble of MLP classifiers. The measure is computed between pairs of patterns on the training data, and is based on a spectral representation of a Boolean function. This representation characterises the mapping from classifier decisions to target label, and allows accuracy and diversity to be incorporated within a single measure. Results on many benchmark problems, including the ORL face database demonstrate that the measure is well correlated with base classifier test error, and may be used to predict the optimal number of training epochs. While correlation with ensemble test error is not quite as strong, it is shown in this paper that the measure may be used to predict number of epochs for optimal ensemble performance. Although the technique is only applicable to two-class problems, it is extended here to multi-class through Output Coding. For the Output Coding technique, a random code matrix is shown to give better performance than One-per-class code, even when the base classifier is well-tuned.

Keywords: Boolean, ECOC, Diversity, Multiple Classifiers, Face Identification

1 Introduction

Multi-layer perceptrons (MLP) make powerful classifiers that may provide superior performance compared with other classifiers, but are often criticized for the number of free parameters. Most commonly, parameters are set with the help of either a validation set or cross-validation techniques [1]. However, there is no guarantee that a pseudo-test set is representative, and for many pattern recognition problems there is insufficient data to rely on this approach. Cross-validation can also be time-consuming and biased [2]. For realistic problems, slow convergence and lack of guarantee of global minima are further drawbacks of MLP training [3].

Ensemble classifiers, also called committees or Multiple Classifier Systems (MCS) offer a way of solving some of these problems. The idea of combining multiple classifiers is based on the observation that achieving optimal performance in combination is not necessarily consistent with obtaining the best performance for an individual (base) classifier. The rationale is that it may be easier to optimise the design of a combination of relatively simple classifiers than to optimise the design of a single complex classifier. An MLP with random

starting weights is a suitable base classifier since randomisation has shown to be beneficial in the MCS context. Random selection has been successfully applied to training sets (Bootstrapping [4]), to feature sets (random subsets [5]) and to output labels [6]. Problems of local minima and computational slowness may be alleviated by the MCS approach of pooling together the decisions obtained from locally optimal classifiers. However, there is still the problem of tuning base classifiers, and the main focus of the paper concerns this issue. The architecture envisaged is a simple MCS framework in which there are B parallel MLP base classifiers.

Although it is known that diversity among base classifiers is a necessary condition for improvement in ensemble performance, there is no general agreement about how to quantify the notion of diversity among a set of classifiers. The desirability of using negatively correlated base classifiers in an ensemble is generally recognised, and in [7] the relationship between diversity and majority vote accuracy is characterized with respect to classifier dependency. Experimental evidence in [8] casts doubt on the usefulness of diversity measures for predicting majority vote accuracy. Diversity measures can be categorised into pair-wise and non-pair-wise, but to apply pair-wise measures to finding overall diversity it is necessary to average over the classifier set. These pair-wise diversity measures are normally computed between pairs of classifiers and take no account explicitly of the target labels. A spectral measure that combines accuracy and diversity for two-class problems is described in this paper. It is calculated between pairs of patterns, and is based on the spectral representation of a Boolean function that was first proposed for two-class problems in [9], and later developed in the context of MCS in [10]. It was shown for two-class problems in [11] that over-fitting could be detected by observing the spectral measure computed on the training set as it varies with base classifier complexity.

Realistic learning problems are in general ill-posed [12], thereby violating one or more of the properties of continuity, uniqueness and existence. The consequence of not being well-posed is that any attempt to automate the learning task requires some assumptions. The only assumption used here is that base classifier complexity is varied over a suitable range. The spectral measure was tested in [11] for two-class problems and shown to correlate well with base classifier test error. However, the upper limit on number of training epochs was shown to be quite critical. A contribution of this paper is to show that the incorporation of bootstrapping for estimating the measures enables good correlation over a wider range of base classifier complexity. A second contribution is to extend the method to solving multi-class problems (defined as k-class, $k > 2$) through Error-Correcting Output Coding (ECOC). Although the method was first proposed in [20], here it is tested on a greater number of benchmark datasets including a face recognition database. A third contribution is to show, in the presence of label noise, that the one per class (OPC) code is inferior to ECOC even when the base classifiers are well-tuned.

The spectral measure is defined in Section 2, and put in context of pair-wise diversity measures in Section 3. The Output Coding approach to solving multi-class problems is described in Section 4, and the face recognition database explained in Section 5. Experimental evidence in Section 6 includes test error rate plots as number of training epochs is systematically varied, as well as tables of correlation coefficients between test error and all the measures defined in Section 2 and Section 3.

2 Spectral Measure

Before providing a mathematical formulation of the spectral measure, a more intuitive description will be attempted. The idea is to represent each training pattern by the binary decisions of the multiple classifiers, giving rise to a binary-to-binary mapping with respect to binary target labels. For the (unrealistic) case that the mapping is completely specified, a search is made for all pattern pairs that have identical classifier decisions except one. That component is negatively or positively correlated with respect to the target class. By summing the individual correlations, the spectral measure for a pattern is defined as the normalised difference between total positive and negative correlations. For the (realistic) case of an incompletely specified mapping, all pattern pairs contribute to the total correlation, not just those that are unit Hamming Distance apart.

Initially two-class supervised learning problems are considered, with the label given to each pattern X_m denoted by $\omega_m = f(X_m)$ where $m = 1 \ldots \mu$ and $\omega_m \in \{0,1\}$ or $\{+1,-1\}$. Here f is the unknown function that maps X_m to the target label ω_m. It is assumed that there are B parallel single hidden-layer MLP base classifiers and that X_m is a B-dimension vector formed from the outputs of the B classifiers (ξ_{mi}, $i = 1 \ldots B$) applied to the original patterns which in general are real-valued and of arbitrary dimension. Therefore, we may represent the *mth* pattern by

$$X_m = (\xi_{m1}, \xi_{m2}, \ldots, \xi_{mB}) \qquad (1)$$

where $\xi \in \{x^s, x, x^d\}$, defined by

$x^s \in [0,1]$ is the soft decision in the interval

$x \in \{0,1\}$or $\{+1,-1\}$ is the hard (binary) decision formed by hardening x^s

$x^d \in \{0,1\}$ is the binary decision conventionally used for calculating diversity measures, where a correct classification is indicated by $x_{mi}^d = 1$ if and only if $x_{mi} = \omega_m$

In this section, $f(X_m)$ is a binary-to-binary mapping between classifier outputs and target labels with x (rather than x^s) in (1) representing a vertex in the B-dimensional binary hypercube. In [9], a spectral transform of $f(X)$ is proposed for characterising this mapping. These mappings are derived from the Hadamard transform T_n defined recursively as follows

$$T_n = \begin{bmatrix} T_{n-1} & T_{n-1} \\ T_{n-1} & -T_{n-1} \end{bmatrix} \text{ where } T_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2}$$

The transforms derived from (2) give rise to spectral coefficients [13] so that

$$s_0 = \sum_{m=1}^{\mu} \omega_m , \quad s_i = \sum_{m=1}^{\mu} X_{mi} * \omega_m , \quad s_{ij} = \sum_{m=1}^{\mu} ((X_{mi} \oplus X_{mj}) * \omega_m), \ \ldots\ldots$$

s_0	correlation between $f(X)$ and constant	
s_i $i=1...n$	correlation between $f(X)$ and x_i	(3)
s_{ij} $i,j = 1...n, i \neq j$	correlation between $f(X)$ and $x_i \oplus x_j$	
s_{ijk} $i,j,k = 1..n, i \neq j \neq k$	correlation between $f(X)$ and $x_i \oplus x_j \oplus x_k$	
..... and continues for fourth order and above		

where $\oplus$ is logic exclusive-OR.

In [9], first order coefficients s_i in (3) are computed by searching for pairs of binary patterns, one from each class, that differ in only a single component. The classifier representing that component is said to be sensitive in that a change in the classifier decision indicates a change in class label. For a completely specified Boolean function (truth table available), the *mth* pattern component x_{mj} is assigned sensitivity σ_{mj} ($j=1,2,...B$) as follows

$$\sigma_{mj}^{+} = x_{mj} \oplus x_{nj} = 1, \qquad \sum_{k=1}^{B} (x_{mk} \oplus x_{nk}) = 1, \quad x_{mj} = \omega_m \neq \omega_n \tag{4}$$

$$\sigma_{mj}^{-} = x_{mj} \oplus x_{nj} = 1, \qquad \sum_{k=1}^{B} (x_{mk} \oplus x_{nk}) = 1, \quad x_{mj} = \omega_n \neq \omega_m ,$$

and $\sum_{j=1}^{B} (x_{mj} \oplus x_{nj})$ is the Hamming Distance

Applying (4) involves a search in which each pattern X_m of one class, is paired with patterns of the other class that are unit Hamming Distance apart, and setting $\sigma_{mj}^{+} = 1$ *if* $x_{mj} = \omega_n$ and $\sigma_{mj}^{-} = 1$ otherwise. The search

process is identical to the first stage of logic minimisation, a description of which can be found in any standard textbook on combinational logic.

In [11], a technique known as spectral summation is described, in which contribution σ_{mj} associated with pattern component x_{mj} can be added to compute first order spectral coefficients s_i in (3). Spectral summation is described in [13], and the idea of separation into positive and negative contributions was first proposed in [9]. The existence of excitatory and inhibitory contributions $\sum_{m=1}^{\mu} \sigma_{mj}^{+} > 0$ and $\sum_{m=1}^{\mu} \sigma_{mj}^{-} > 0$ for given j provides evidence that the set of patterns is non-separable in the jth component [10]. For details of separable and non-separable Boolean functions, see reference [13]. To clarify the computation of sensitivity and spectral summation pseudo-code is provided in Figure 1.

The difference between the positive and negative contributions gives the first order spectral coefficients, as illustrated in the following example of a non-separable Boolean function

$$f(X) = \bar{x}_1 x_2 + x_1 \bar{x}_2 + x_2 x_3 \tag{5}$$

The truth table in {+1,-1} rather than {0,1} coding for the function defined in (5) is given by

id	1	2	3	class
X_1	1	1	1	1
X_2	-1	1	1	-1
X_3	1	-1	1	-1
X_4	-1	-1	1	1
X_5	1	1	-1	1
X_6	-1	1	-1	-1
X_7	1	-1	-1	-1
X_8	-1	-1	-1	-1

The truth table ordering defines the spectral coefficient ordering [13], which is computed as follows for T_3 in equation (2)

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} -2 \\ +2 \\ +2 \\ +6 \\ +2 \\ -2 \\ -2 \\ +2 \end{bmatrix} \begin{matrix} s_0 \\ s_1 \\ s_2 \\ s_{12} \\ s_3 \\ s_{13} \\ s_{23} \\ s_{123} \end{matrix}$$

By comparing the truth table and the transformation matrix it may be seen that first order coefficients s_i, i = 1,2,3 represent the correlation between *f(X)* and x_i. Similarly the second order coefficients represent correlation between *f(X)* and $x_i \oplus x_j$ as defined in (3). Now consider the result of applying (4) to class 1 patterns (X_1, X_4, X_5) of function *f(X)* in (5). For notational convenience, binary component *x* has its associated sensitivity σ represented as superscript x^{σ} as follows

$$\begin{array}{ccc} s_1 & s_2 & s_3 \\ \begin{bmatrix} +1^1 & +1^1 & +1^0 \\ -1^1 & -1^1 & +1^1 \\ +1^1 & +1^1 & -1^0 \end{bmatrix} & & \begin{matrix} X_1 \\ X_4 \\ X_5 \end{matrix} \end{array}$$

An alternative calculation of the spectral coefficients is obtained by applying spectral summation to the class 1 patterns. The three rows (X_1, X_4, X_5) represent class 1 binary patterns and the first order coefficients are calculated from the three columns by adding (x_j = +1) or subtracting (x_j = -1) when $\sigma_{mj} = 1$ $(m=1,4,5)$, with no contribution when $\sigma_{mj} = 0$. The contribution is doubled, since spectral summation from the class 1 patterns is identical to spectral summation from class –1 patterns, as explained in the notes for the pseudo-code in Figure 1.To calculate higher order coefficients, the first order contributions are added for the respective columns (in this paper we are only using first order coefficients).

e.g. s_1 = 2 * (1-1+1) = +2, using column 1.

s_{12} = 2 * ((1 * 1) + (-1 * -1) + (1 * 1)) = +6 using column 1,2.

s_{123} = 0 + 2 * (-1 * -1 * 1) + 0 = +2 using column 1,2,3.

Consider separating the positive and negative contributions so that $\sum_{m=1}^{8} \sigma_{mj}^{+} / \sum_{m=1}^{8} \sigma_{mj}^{-}$ $(j = 1,2,3)$ = [4/2,4/2,2/0]. When both positive and negative contributions are non-zero, for given *j*, a function violates the 1-monotonicity constraint and is therefore non-separable. From another perspective, in order to implement the function with a Threshold Logic Unit (TLU) the implication is that the weight on line *j* needs to be both positive and negative. That is, a single TLU cannot implement the function. The function defined in (5) is therefore not 1-monotonic in the first two components. Note that even if a function is 1-monotonic, it may still be non-separable due to violation of higher montonicity constraints [13]. It is difficult to give an intuitive explanation of the meaning of the spectral coefficients, since the positive and negative correlations cancel. However, by keeping the

correlations separate we can determine the evidence for overall positive and negative correlation, which gives more information than the spectral coefficients by themselves. If $\sum_{m=1}^{8} \sigma_{mi}^{+} > \sum_{m=1}^{8} \sigma_{mj}^{+}$ the evidence is that classifier *i* is more positively correlated than classifier *j* (for example classifier 1 is more positively correlated than classifier 3).

Clearly, for a realistic learning problem the unknown binary-to-binary function *f* will not be completely specified. However, the concept of spectral summation is still applicable even if the function is noisy, incompletely specified and perhaps contradictory, as is the case for pattern recognition problems. To estimate the coefficients, it is assumed that the pattern components, which in our framework are outputs of binary classifiers, are independent. Therefore, each classifier provides evidence that a pattern is positively or negatively correlated with all patterns of the other class. The *mth* pattern component x_{mj} is assigned σ_{mj} (*j=1,2,...B*) as follows

$$\sigma_{mj}{}^{c} = \sum_{n=1}^{\mu} x_{mj} \oplus x_{nj} \qquad (6)$$

where correlation c = + if $x_{mj} = \omega_m \neq \omega_n$ and c = – if $x_{mj} = \omega_n \neq \omega_m$,

In (6), in contrast to (4), the contribution for each pattern component comes from all patterns of the other class, not just nearest neighbours. The pseudo-code for the computation is shown in Figure 1, in which the conditional (marked ****) is removed. Note that any classifier that correctly classifies one pattern of a pattern pair, but incorrectly classifies the other, does not contribute to the summation. After applying (6) the *jth* component x_{mj} of a pattern pair has associated σ_{mj}^{-} only if the *jth* base classifier mis-classifies both patterns. Therefore we expect that a pattern with relatively large $\sum_{j=1}^{B} \sigma_{mj}{}^{-}$ is likely to come from regions where the two classes overlap. We now define a measure for each pattern that represents the difficulty of separating that pattern from patterns of the other class. It is based on a summation of contributions, relative to the total number of contributions. For any pattern, say the *nth* pattern, σ'_n (where prime (′) indicates that a measure is computed across pattern pairs of opposite class rather than across classifiers as in Section 3), sums the difference between excitatory and inhibitory contributions, normalised so that $-1 \le \sigma'_n \le 1$

$$\sigma'_n = \frac{\mathbf{1}}{\boldsymbol{K}} \times \sum_{j=1}^{B} \left(\frac{\sigma_{nj}{}^{+}}{\sum_{m=1}^{\mu} \sigma_{mj}{}^{+}} - \frac{\sigma_{nj}{}^{-}}{\sum_{m=1}^{\mu} \sigma_{mj}{}^{-}} \right) , \; K = \sum_{j=1}^{B} \left(\frac{\sigma_{nj}{}^{+}}{\sum_{m=1}^{\mu} \sigma_{mj}{}^{+}} + \frac{\sigma_{nj}{}^{-}}{\sum_{m=1}^{\mu} \sigma_{mj}{}^{-}} \right) \qquad (7)$$

In (7) σ'_n may be compared with the margin for the *nth* pattern. The margin of a training example is defined as the difference between the weight given to the correct class and the maximum weight given to any of the other classes. It is defined as a number between –1 and +1, and is positive for a correct classification. Furthermore, the absolute value of the margin represents confidence of classification. For a two-class problem, the margin M_n' for *nth* training pattern X_n is given by

$$M'_n = \frac{f(X_n)\sum_{j=1}^{B}\alpha_j x_{nj}}{\sum_{j=1}^{b}|\alpha_j|} \qquad (8)$$

where α_j is the weight associated with *jth* base classifier

Note that margin in (8) for majority vote ($\alpha_j = 1/B$) is identical to unnormalised s_0 defined in (3), so that the margin may be regarded as a special case of spectral summation. Cumulative Distribution graphs [14] can be defined similar to that for margin, that is $g(\sigma'_n)$ versus σ'_n where $g(\sigma'_n)$ is the fraction of patterns with value at least σ'_n. In this paper, a single measure for a set of patterns is obtained by taking the mean over positively correlated patterns, which represents the area under the Cumulative Distribution Graph [11]

$$\sigma' = \frac{1}{\mu}\sum_{n=1}^{\mu}\sigma'_n \qquad \sigma'_n > 0 \qquad (9)$$

$$M' = \frac{1}{\mu}\sum_{n=1}^{\mu}M'_n, \qquad M'_n > 0 \qquad (10)$$

Since σ'_n and M'_n in (9) and (10) vary between –1 and +1, σ' and M' vary between *0 and 1. W*hen σ' is plotted as base classifier complexity is varied (Section 6), the curves may be interpreted as (1 - mean over negatively correlated patterns).

3 Pair-wise diversity measures

Various approaches to defining diversity, and to determining the relationship between diversity and accuracy, have been proposed. For our study we consider pair-wise diversity measures, and follow the notation x^d in (1) and used in [7], in which the output of a classifier is defined to be 1 if and only if a pattern is correctly classified.

Although diversity measures are conventionally calculated over base classifiers, it is also possible to compute them over patterns [15]. If the B classifier decisions for μ patterns form a μ x B binary matrix, conventional pair-wise diversity measures [7] are computed between pairs of rows independent of class label. In contrast, the spectral measure in Section 2 is defined between pairs of columns, where each one of the pair represents a pattern chosen with different class label.

Diversity over classifiers

Let the *jth* classifier output for the *pth* pattern under this labelling scheme be a μ-dimensional binary vector given by x_{pj}^{d} *where p = 1,… μ.*. The following counts are defined for *ith* and *jth* classifiers

$$N_{ij}^{00} = \sum_{p=1}^{\mu} \overline{x}_{pi}^{d} \wedge \overline{x}_{pj}^{d}, \quad N_{ij}^{11} = \sum_{p=1}^{\mu} x_{pi}^{d} \wedge x_{pj}^{d}, \quad N_{ij}^{10} = \sum_{p=1}^{\mu} x_{pi}^{d} \wedge \overline{x}_{pj}^{d}, \quad N_{ij}^{01} = \sum_{p=1}^{\mu} \overline{x}_{pi}^{d} \wedge x_{pj}^{d} \qquad (11)$$

where $\wedge$ is logical AND and $\overline{x}^{d}$ is the logical complement of x^{d}

For example if the first two columns of a μ x 6 binary matrix is given by $\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}'$ we have $N_{12}^{00} = 1$, $N_{12}^{11} = 2$, $N_{12}^{10} = 2$, $N_{12}^{01} = 1$.

The Q statistic, Correlation coefficient (ρ), and Double Fault (F) measures defined in [7], all increase with decreasing diversity. Here an Agreement (A) measure is defined as (1 – Disagreement) to make it also increase with decreasing diversity so that

$$Q_{ij} = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}} \qquad (12)$$

$$\rho_{ij} = \frac{N^{11}N^{00} - N^{01}N^{10}}{\sqrt{(N^{11} + N^{10})(N^{01} + N^{00})(N^{11} + N^{01})(N^{10} + N^{00})}} \qquad (13)$$

$$A_{ij} = 1 - \frac{N^{01} + N^{10}}{N^{11} + N^{00} + N^{01} + N^{10}} \qquad (14)$$

$$F_{ij} = \frac{N^{00}}{N^{11} + N^{00} + N^{01} + N^{10}} \qquad (15)$$

where the *ij* subscripts for N in (12) to (15) have been omitted for convenience

The mean diversity measure $\Delta \in \{Q, \rho, A, F\}$ over B classifiers is given by

$$\Delta = \frac{2}{B(B-1)} \sum_{i=1}^{B-1} \sum_{j=i+1}^{B} \Delta_{ij} \tag{16}$$

Diversity over patterns

For comparison, analogous to the spectral measure of correlation (6), we propose to calculate diversity measures over patterns between the two classes using counts as follows

$$\tilde{N}_{mn}^{00} = \sum_{j=1}^{B} (\bar{x}_{mj}^{d} \wedge \bar{x}_{nj}^{d}),\ \tilde{N}_{mn}^{11} = \sum_{j=1}^{B} (x_{mj}^{d} \wedge x_{nj}^{d}),\ \tilde{N}_{mn}^{10} = \sum_{j=1}^{B} (x_{mj}^{d} \wedge \bar{x}_{nj}^{d}),\ \tilde{N}_{mn}^{01} = \sum_{j=1}^{B} (\bar{x}_{mj}^{d} \wedge x_{nj}^{d}), \tag{17}$$

where $\omega_m \neq \omega_n$

The *nth* pattern may then be given a measure of diversity Δ_n' by applying equations (12) to (15), and summing over all patterns of the other class. Similar to (9) and (10) we produce a single measure Δ' over all patterns as follows

$$\Delta' = \sum_{n=1}^{\mu} \Delta_n',\ \Delta_n' > 0 \tag{18}$$

The measures defined in (18) are experimentally compared in Section 6.

Now σ_n' defined in (7) may be formulated in the notation used for diversity measures as follows

$$\sigma_n' = \frac{1}{\tilde{K}} \left(\frac{\tilde{N}_n^{11}}{\sum_{m=1}^{\mu} \tilde{N}_m^{11}} - \frac{\tilde{N}_n^{00}}{\sum_{m=1}^{\mu} \tilde{N}_m^{00}} \right),\ \tilde{K} = \left(\frac{\tilde{N}_n^{11}}{\sum_{m=1}^{\mu} \tilde{N}_m^{11}} + \frac{\tilde{N}_n^{00}}{\sum_{m=1}^{\mu} \tilde{N}_m^{00}} \right) \tag{19}$$

From (19) and (9) σ' is the probability that $\left(\frac{\tilde{N}_n^{11}}{\sum_{m=1}^{\mu} \tilde{N}_m^{11}} - \frac{\tilde{N}_n^{00}}{\sum_{m=1}^{\mu} \tilde{N}_m^{00}} \right) > 0$, and may be considered as a measure of class separability. It provides an intuitive explanation of why we may expect that σ' correlates well with base classifier test error, and in particular peaks at the same number of training epochs. Consider the example of two overlapping Gaussians representing a two-class problem, with the Bayes boundary assumed to be placed where the probability density curves cross. Let the overlapping region be defined as the tails of the two Gaussians with respect to the Bayes boundary. If base classifiers are capable of approximating the Bayes boundary, by definition

an optimal base classifier will incorrectly classify all patterns in the overlapping region and correctly classify all other patterns. Now consider the situation that the complexity of the base classifiers increases beyond optimal, so that some patterns in the overlapping region become correctly classified and some of the remaining patterns become incorrectly classified. The result is that there is greater variability in classification among patterns close to the Bayes boundary, and it is more difficult to separate them. The probability represented by σ' decreases as complexity increases since $\tilde{N}^{00}$ is more evenly distributed over all patterns, leading to a reduction in positively correlated patterns. The effect on the Cumulative Distribution Graphs, defined in Section 3, is shown in reference [11]. However, if the base classifier becomes too powerful, eventually all patterns are correctly classified and $\tilde{N}^{00} \rightarrow 0$ and $\sigma' \rightarrow 1$, so it is expected that σ' would start to increase as seen in Figure 12 in Section 6.

It is also possible to understand how σ' may be used to predict test error by appealing to the notions of Bias and Variance, which are motivated by analogous concepts in regression theory. However, there are difficulties with the various Bias/Variance definitions for 0/1 loss function. Firstly, a comparison of Bias/Variance definitions [16] shows that no single definition satisfies zero Bias and zero Variance for Bayes classifier, together with additive Bias and Variance decomposition of error. Secondly, the effect of bias and variance on error rate cannot be guaranteed, and it is easy to think of example probability distributions for which the effect is counter-intuitive [16] [17]. Thirdly, there is the practical difficulty that the Bayes classification needs to be known or estimated. Breiman's definition [17] is based on defining Variance as the component of classification error that is eliminated by aggregation. Patterns are divided into two sets, the Bias set B containing patterns for which the Bayes classification disagrees with the ensemble classifier and the Unbias set U containing the remainder. Bias is computed using B patterns and Variance is computed using U patterns, but both Bias and Variance are defined as the difference between the probabilities that the Bayes and base classifier predict the correct class label. Breiman's definition was used in [11] to explain how σ' can predict generalisation error. As base classifier complexity increases beyond optimal, bias initially stays low and variance increases, and this leads to a reduction in correlation.

In [11] it was shown that σ' correlates well with base classifier test error, but was dependent on choosing a suitable value for the upper limit on number of training epochs to limit classifier complexity. In this paper, bootstrapping [18] is incorporated to improve the estimate of σ'. Bootstrapping is a popular ensemble technique and implies that training patterns are randomly sampled with replacement, so that approximately one third of patterns are removed and the remaining patterns occur one or more times. Experimental evidence in Section 6

shows that bootstrapping improves the correlation of σ' with classifier test error in the over-fit region, that is when the number of training epochs is increased beyond optimal.

4 Output Coding and Multi-Class Problems

Error-Correcting Output Coding (ECOC) is a well-established method [19] for solving multi-class problems by decomposition into complementary two-class problems. It is a two-stage process, coding followed by decoding, but there is some discussion about whether the error-correcting aspect is relevant to its performance [16]. It seems more appropriate to refer to the technique as Output Coding, in recognition of the variety of ways of producing codes that make no explicit reference to error-correction properties. The idea of using codes was originally based on modelling the multi-class learning task as a communication problem in which class information is transmitted over a channel. Although errors arise from a variety of causes, including estimation errors from the measurement process and base classifier learning algorithm, the effect is assumed to be to introduce classification errors into the ECOC feature vector representing the pattern. The main motivation was to correct these errors in decoding.

The coding step is defined by the binary *k x B* code word matrix Z that has one row (code word) for each of *k* classes, with each column defining one of B sub-problems that use a different labelling. Assuming each element of Z is a binary variable z, a training pattern with target class ω_l *(l = 1... k)* is re-labelled as class Ω_1 if $Z_{ij} = z$ and as class Ω_2 if $Z_{ij} = \bar{z}$. The two super-classes Ω_1 and Ω_2 represent, for each column, a different decomposition of the original problem. For example, if a column of Z is given by $[0\ 1\ 0\ 0\ 1]^T$, this would naturally be interpreted as patterns from class 2 and 5 being assigned to Ω_1 with remaining patterns assigned to Ω_2. This is in contrast to the conventional One-per-class (OPC) code, which can be defined by the diagonal *k x k* code matrix $\{Z_{ij} = 1$ if and only if i = j$\}$.

In the test phase, the *jth* classifier produces an estimated probability $\hat{q}_j$ that a test pattern comes from the super-class defined by the *jth* decomposition. The *pth* test pattern is assigned to the class that is represented by the closest code word, where distance of the *pth* pattern to the *ith* code word is defined as

$$D_{pi} = \sum_{j=1}^{B} \alpha_{jl} \left| Z_{ij} - \hat{q}_{pj} \right| \qquad l = 1,...k \tag{20}$$

where α_{jl} allows for *lth* class and *jth* classifier to be assigned a different weight [20]. Hamming decoding is denoted in (20) by $\{\alpha=1,\ \hat{q} \equiv x\}$ and L^1 norm decoding by $\{\alpha=1,\ \hat{q} \equiv x^s\}$ where *x* and x^s are defined in (1). Many types of decoding are possible, but theoretical and experimental evidence indicates that, providing a

problem-independent code is long enough and base classifier is powerful enough, performance is not much affected. However, it is shown in [20] that when base classifiers are sub-optimal and vary in accuracy weighted decoding gives better performance. In this paper Hamming and L^1 norm decoding are compared, and the emphasis is on optimising base classifier complexity using these simple decoding schemes.

In addition to the Bayes error, errors due to individual classifiers and due to the combining strategy can be distinguished. This can be further broken down into errors due to sub-optimal decomposition and errors due to the distance-based decision rule. If it is assumed that each classifier provides exactly the probability of respective super-class membership, with posterior probability of *lth* class represented by q_{pl} ($l = 1 \ldots k$), from equation (20), assuming $\alpha_{jl}=1$, it is shown in [21] that

$$D_{pi} = \sum_{j=1}^{B} \left| \left(\sum_{l=1}^{k} q_{pl} Z_{lj} \right) - Z_{ij} \right| = (1 - q_{pi}) \sum_{j=1}^{B} \left| Z_{lj} - Z_{ij} \right| \qquad (21)$$

Equation (21) tells us that D_{pi} is the product of $(1-q_{pi})$ and Hamming Distance between code words, so that when all pairs of code words are equidistant, minimising D_{pi} implies maximising posterior probability, which is equivalent to Bayes rule. Therefore any variation in Hamming distance between pairs of code words will reduce the effectiveness of the combining strategy. From (20) and (21) it may also be shown that variance of $\hat{q}$ is proportional to correlation between base classifiers and inversely proportional to the square of Hamming Distance between code words [27]. In [22] it is shown that maximising the minimum Hamming Distance between code words implies minimising upper bounds on generalisation error.

In classical coding theory, theorems on error-correcting codes guarantee a reduction in the noise in a communication channel, but the assumption is that errors are independent. When applied to machine learning the situation is more complex, in that error correlation depends on the data set, base classifier as well as the code matrix Z. In the original ECOC approach [19], heuristics were employed to maximise the distance between the columns of Z to reduce error correlation. Hadamard matrices, defined in (2), maximise distance between rows and columns and were used in [23], in which it was shown that training error is bounded by

$$\binom{B}{2} \Big/ \binom{d/2}{2} \Theta \leq \frac{4B(B-1)}{d(d-2)} \Theta$$

where Θ is an upper bound on probability of error correlation, and d is the minimum distance between code words.

The considerations explained above impose stringent requirements on choice of code word columns and rows, and finding optimal code matrices satisfying these properties is a complex problem. Codes are normally binary and problem-independent, but there has been recent interest in adaptive, problem-dependent and non-binary codes [24] [25]. In reference [25] it is proved that an adaptive method is NP-complete. Random codes, provided that they are long enough, have frequently been employed with almost as good performance [21]. It would seem to be a matter of individual interpretation whether long random codes may be considered to approximate required error-correcting properties. In this paper a random code matrix with near equal split of classes (approximately equal number of 1's in each column) is chosen, as proposed in [26]. An experimental comparison using random code with no constraint on number of labels, and OPC code is provided in Section 6.

5 Face Recognition Database

Facial images are a popular source of biometric information since they are relatively easy to acquire. However, automated face recognition systems often perform poorly due to small number of relatively high-dimensional training patterns, which can lead to poor generalisation through over-fitting. Face recognition is an integral part of systems designed for many applications including identity verification, security, surveillance and crime-solving. Improving their performance is known to be a difficult task, but one approach to improving accuracy and efficiency is provided by the method of Output Coded ensemble classifiers [27].

A typical face recognition system consists of three functional stages. In the first stage, the image of a face is registered and normalised. Since face images differ in both shape and intensity, *shape alignment* (geometric normalisation) and *intensity correction* (photometric normalisation) can improve performance. The second stage is feature extraction in which discriminant features are extracted from the face region. Finally, there is the matching stage in which a decision-making scheme needs to be designed depending on the task to be performed. In identification, the system classifies a face from a database of known individuals, while in verification the system should confirm or reject a claimed identity. To match a probe image to a face in a database, two methods are generally used. In geometric feature-based matching, relative positions and other parameters of distinctive features such as eyes, mouth and nose are extracted. The alternative is to consider the global image of the face as with methods such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA).

Although it is possible to use grey levels directly, this is usually not computationally feasible without reducing the size of the image. Normally, better results are obtained if features are first extracted. A popular approach is

PCA, but by itself PCA is not adequate for the face recognition task since projection directions only maximise the total scatter across all classes. Therefore we use LDA which requires computation of the between-class scatter matrix, S_B and the within-class scatter matrix, S_W. The objective of LDA is to find the transformation matrix, W_{opt}, that maximises the ratio of determinants $\left|w^T S_B w\right| / \left|w^T S_W w\right|$. W_{opt} is known to be the solution of the following eigenvalue problem.

$$S_B - S_W \Lambda = 0 \tag{22}$$

where Λ is a diagonal matrix whose elements are the eigenvalues of matrix $S_W^{-1} S_B$. Since in practice S_W in (22) is nearly always singular, dimensionality reduction is first achieved by PCA before solving the eigenvalue problem.

The database used is the ORL (Olivetti Research Laboratory http//www.cam-orl.co.uk), consisting of four hundred images of forty individual faces with some variation in lighting, facial expression, facial hair, pose and spectacles. The background was controlled with subjects in an upright frontal position, although small variation in rotation and scale was allowed. The advantage of this database is that it can be used without need for a face detection algorithm or any other pre-processing, so that there is a fairer comparison with the results obtained by other researchers. In our experiments, images have been projected to forty-dimensions using PCA and subsequently to a twenty-dimension feature space using LDA. It is treated as a forty-class face identification problem with the four hundred images randomly split into training/testing patterns. A comparison of results on this database is given in [3], using 50/50 split, but the number of experimental runs is smaller than used in our experiments. As pointed out in [3], some researchers do not specify the split that they have used, and some only base their results on one run, so that care is needed before making any comparison.

6 Experimental Evidence

The main purpose of these experiments is to determine how well the measures defined in Section 2 and Section 3 correlate with test error as the number of training epochs of single hidden-layer MLP base classifiers are systematically varied. All the measures are computed on the training data and the datasets use random training/testing split, respecting the class distribution as closely as possible. Experiments are repeated with and without bootstrapping ten times, except for the face database which is repeated twenty times. For example, if epochs and nodes are varied each node-epoch combination is repeated ten or twenty times, and a 50/50 split for

the ORL face database implies five training images per class. Each repetition uses the same base classifier parameters, which are identical for all classifiers. Variation for each classifier run arises from one or more of three sources, (i) random initial weights, (ii) bootstrapping, and (iii) the problem decomposition defined by the respective code matrix column. For multi-class problems, three different coding matrices have been used (i) random code with approximately equal split of labels (REQ) (ii) random code (RAN), all ones and all zeros removed (iii) One-per-class (OPC) code repeated B/k times.

Natural benchmark problems, selected from [28] and [29] are shown in Table 1 with numbers of patterns, classes, continuous and discrete features. For datasets with missing values the scheme suggested in [28] is used. Unless otherwise stated, two-class problems use one hundred classifiers (B = 100), multi-class problems use two hundred classifiers (k x 200 coding matrix), and the face database uses five hundred classifiers (40 x 500 code matrix). The Resilient BackPropagation training algorithm [30] with default parameters is used throughout. For the two-class problems, similar experiments were presented in [31] but no bootstrapping was applied and the Levenberg-Marquardt training algorithm was used. It is not the intention in this paper to perform a detailed comparison between the two algorithms, but it may be worth noting that the only observed difference was in the number of epochs required for optimal performance. The maximum number of epochs for Resilient BackPropagation was chosen to achieve over-fitting similar to Levenberg-Marquardt.

For the benchmark datasets tested in this paper, none except *Diabetes* over-fitted for the range of base classifier complexity values considered. To encourage over-fitting most experiments were carried out with 20/80 training/testing split and varying classification noise, in which a percentage of patterns of each class were selected at random (without replacement), and each target label changed to a class chosen at random (from patterns of the remaining classes). Unless otherwise stated the number of epochs is varied from 2 to 1024 using a log scale. The tables of correlation coefficients for the measures defined in Section 2 and Section 3 are with respect to specified test error as number of epochs is varied.

Two-class datasets

Figure 2 (a) – (d) give base classifier and majority vote (MAJ) test and train error rates for *Diabetes* 50/50 + 0% classification noise with [2,4,8,16] hidden nodes and with bootstrapping applied. Minimum test error occurs at 8-16 epochs, for which neither base classifier nor ensemble test error is much affected by number of nodes. Figure 2 (e) (f) shows the difference in error rate between MAJ and SUM (where SUM is defined using soft outputs x^s in (1)), indicating that SUM only gives lower test error than MAJ for 2, 4 epochs. Figure 3 shows four measures σ', Q, M′, Q′ defined in (9) (16) (10) (18), and it is emphasised again that all measures are computed on

the training set. It may be seen that both σ' and Q appear to be good predictors of base classifier test error in Figure 2 (a). However, it will be shown in Figure 5 that Q is poorly correlated on average over all two-class datasets, unless classification noise is added.

In Figure 4 is shown the mean test error, σ' and Q over all 20/80 two-class datasets using 8 hidden-node bootstrapped base classifiers for [0,20,40] % noise. It is perhaps surprising that, even though there are seven different datasets, the mean curves show a meaningful trend as number of epochs is varied. The minimum base classifier test error occurs on average at 8 epochs and both σ' and Q peak at 8 epochs. Note also that minimum of MAJ test error also occurs at 8 epochs for 0% noise but as noise is increased the minimum occurs at fewer number of epochs. Figure 5 (with Bootstrapping) and Figure 6 (without Bootstrapping) show the mean correlation coefficients over all seven datasets for all measures. Each measure is grouped as six vertical bars in the order BASE (0,20% noise), MAJ (0,20% noise), SUM (0,20% noise) and demonstrate that bootstrapping makes a significant difference to the estimation of these measures. With bootstrapping applied, spectral measure σ' is the only measure that is strongly (negatively) correlated with base classifier test error for both 0 and 20 % noise. Without bootstrapping σ' is weakly (positively) correlated with base classifier test error. Q and ρ are strongly negatively correlated when 20% classification noise is added but poorly correlated with 0% noise. Table 3 shows the number of datasets for Figure 5 for which there is ninety-five percent confidence that the correlation would not be as large as the observed value by random chance. Although σ' is less well correlated with ensemble test error, it correlates more strongly than other measures (except 20 % noise for Q, ρ, F which are equally strongly negatively correlated).

The effect of bootstrapping on the base classifier error rates is shown in Figure 7, indicating that bootstrapping increases training error as number of epochs is increased. However, when averaged over all datasets, epochs and noise levels bootstrapping makes little difference to ensemble test error rate, improving it by 0.54 percent.

The best of the mean majority vote error rates for individual two-class problems are shown in Table 2, along with corresponding Std error and number of epochs at which ensemble error is minimised. The datasets have 20/80 train/test split with 8 hidden nodes, and bootstrapping is applied. In order to determine the effect of using the spectral measure σ' to predict ensemble test error, the fourth column of Table 2 shows the ratio of the error at the number of epochs predicted by σ' and the error at the optimal number of epochs. The final column of Table 2 shows the ratio when 20% classification noise is added. We may conclude that at 0% noise σ' is a good predictor of the number of epochs for optimal ensemble test error. The conclusion extends to 20% noise, with the possible

exception of datasets *cancer* and *heart* for which the ratios are 1.17 and 1.13 respectively. Only the ratios for σ' are provided since Figure 5 shows that σ' is the only measure that is strongly correlated for both 0 and 20 % noise.

Multi-class datasets

All measures calculated over patterns defined in Section 3 assume a binary-to-binary mapping for two-class problems. For a multi-class problem, *k* binary-to-binary mappings can be defined by replacing decoding with an additional coding stage using *k x k* OPC code. The mean measure is computed over *k* mappings. Figure 8 shows the mean test error rates, σ', Q over all eleven 20/80 multi-class problems with 8 nodes and [0,20,40] % noise with bootstrapping applied. Note that base classifier error in Figure 8 (a) is the mean over *B* columns of the code matrix that define the two-class decompositions. Minima for both base classifier and Hamming Decoded test error occur at 16-32 epochs for 20 and 40 % noise but there is no over-fitting of Hamming Decoded error at 0 % noise. Measures σ' and Q appear to be good predictors of base classifier test error except that at 0 % noise Q is not well correlated, as with two-class problems. Figure 11 shows σ' without bootstrapping, and compared with Figure 8 indicates that σ' does not predict over-fitting unless bootstrapping is applied. The result for VEHICLE dataset in Figure 12 shows that as number of epochs is increased, σ' levels off before continuing to increase. This is not unexpected, as explained in Section 3, since as classifier complexity increases far beyond optimal, all patterns will eventually become correctly classified and $\sigma' \rightarrow 1$. When averaged over all datasets, epochs and noise levels, bootstrapping makes little difference to the ensemble test error, improving it by just 0.40 percent. The effect of number of classifiers (columns of the code matrix) is given in Figure 13, in which it is shown that increasing the number beyond one hundred has little effect.

Figure 9 (with Bootstrapping) and Figure 10 (without Bootstrapping) show the mean correlation coefficients associated with Figure 8 and Figure 11. The correlation is shown for base classifier, Hamming and L^1 norm decoding and demonstrates that σ' is strongly negatively correlated with base classifier test error. Table **4** shows the number of datasets for Figure 9, for which there is ninety-five percent confidence that the correlation would not be as large as the observed value by random chance. From Table **4** and Figure 9 it may be observed that σ' is better correlated with base classifier test error than other measures for 0 and 20 % noise. As with two-class datasets, σ' is seen to be less well correlated with ensemble test error, but better than other measures (with 20 % noise Q, ρ, F are equally strongly negatively correlated). The one dataset that is not significantly correlated is *ecoli*, which is a difficult eight class problem having nearly half the patterns in one class and with three classes having only nine patterns between them.

Figure 14 shows the difference between random codes REQ and RAN, demonstrating that equal split of labels has beneficial effect on ensemble test error for fewer epochs. Figure 15 shows the difference between REQ and OPC codes, and indicates that for 0% noise OPC gives lower ensemble test error for fewer than optimal number of epochs. This result is supported by a recent study in [32], which claims that OPC (also called OVA One-Vs-All) is as good as any other code if base classifier is tuned. However, when classification noise is added, random code REQ is better on average by 0.4 % over 32-1024 epochs. Also when noise is added the optimal number of epochs for ensemble test error is 8 – 16 (Figure 8) and at this value OPC has 2-3 % higher test error. Note in Figure 14 and Figure 15 that even though base classifier train and test error for REQ and RAN is higher, the ensemble test error is lower than it is for OPC.

Table 5 shows the same information as Table 2 for multi-class problems. The ratios in the last two columns demonstrate that, while σ' may be used for predicting ensemble test error, the results for datasets *thyroid* and *segment* are far from optimal.

Orl Database

Figure 16 shows error rates, σ', Q for 50/50 60/40 70/30 and 80/20 splits with 16 nodes and bootstrapping applied. As with two-class and multi-class benchmark problems, σ' appears to be strongly negatively correlated with test error. With no bootstrapping, there is a mean improvement in ensemble error rate over all epochs and splits of 0.11 %. The effect of noise [0 20 40] % is shown in Figure 17, again demonstrating the ability to predict the number of epochs at which base classifier test error is minimum. The correlation of σ' with base classifier test error is significant and it is the only measure to be significant at three noise levels. The difference between random and OPC codes is shown in Figure 18, showing the inferiority of OPC code.

There is interest in achieving lowest error rate for the ORL database. In reference [3] there is a comparison of the minimum 50/50 test error rates achieved by various researchers. The authors of [3] achieve the lowest rate of 1.92 % based on 6 runs, with some basing their results on three or fewer runs, but in general the standard deviations were not given. In an earlier study [33] the authors report 2.7 % error rate with Std 0.6% using ten runs. In our experiments based on twenty runs, we achieved a mean 3.98 % with Std 1.89 %. It is difficult to make a fair comparison when different numbers of runs are used to compute the error rates, as evidenced by three recent studies that use ORL 50/50 split. In agreement with a our results [34], a mean error rate of 4% based on twenty runs is reported. In contrast, an error rate of 2.5 % is reported in [35] based on ten runs, yet in [36] 0% error rate is claimed but it is not clear how many runs the result is based upon. To appreciate the difficulties, note that in our

experiments assuming that the 20 error rates are ordered, the top 6 average 6.25 % while the bottom 6 average 1.92 %. For random 80/20 split, it may be seen from Figure 16 that our best error rate is 0.9 %. Overall, the results using our technique of tuning multiple MLP classifiers, are comparable but there is no claim of superiority compared with other methods.

Discussion

The emphasis in this paper has been on predicting test error as number of training epochs is varied. It is shown that bootstrapping improves the correlation between test error and the spectral measure defined in (9). Furthermore the correlation is significant for a range of two-class and multi-class problems that include both continuous and discrete features, in contrast to [20]. The spectral measure is shown to be well correlated with base classifier test error, and may be used to predict optimal number of training epochs. While correlation with ensemble test error is not quite as strong, it is shown that the measure may be used to predict number of epochs for optimal ensemble performance.

Another way of systematically varying base classifier complexity is to change the number of hidden nodes. When the base classifier is well-tuned, ensemble test error appears to be relatively insensitive to number of nodes as shown in Figure 2 and reported in earlier publications [11] and [20]). In agreement with [37], it appears that a useful design strategy is to start with a network with a large number of nodes, and to use early-stopping. Further study is required before drawing any conclusions about the effect of increasing the number of hidden nodes above 16.

The benefit of using complex codes for Output Coding has recently been called into question, and in [32] it is shown experimentally that OPC code is no worse than other codes if the base classifier is well-tuned. However, it was explicitly recognised in [32] that the result was established using UCI benchmark datasets, with no classification noise added. We have seen that it may be difficult to over-fit these problems, and with no added noise the results in [32] are generally supported. However, when noise is added to the datasets, OPC is shown to be inferior. A possible explanation is that the noise tends to de-correlate the two-class decompositions, and therefore enhance the error-correcting capability. For the ORL database, random codes give better performance than OPC even without added noise. More complex codes, which satisfy the criteria discussed in Section 4, have not been considered in this paper, but were investigated in [21].

A preliminary study was carried out to see if σ' may be used to perform feature selection. LDA features are ordered, so the upper limit of number of features was varied from 5-35. Figure 19 shows the corresponding test

error for [8 16 32] nodes at 50 training epochs. It appears to show that σ' may be used to predict the number of features for minimising base classifier test error, and this could be a useful direction for future study.

6. Conclusion

It is shown experimentally that, over a range of k-class datasets, $k \geq 2$, a pair-wise measure computed over training patterns is well correlated with base classifier test error when number of training epochs of MLP base classifiers are systematically varied. Bootstrapping significantly improves the estimate of this measure, while making little difference to the ensemble test error. It is also demonstrated that correlation of the spectral measure with ensemble test error is not as strong. These can be thought of as two separate problems, the first being concerned with the prediction of over-fitting of the base classifier and which is the main focus of this paper. The second problem is to determine the relationship between ensemble and base classifier test error. The evidence in this paper is that for $k > 2$ minimum ensemble test error generally occurs at a higher number of epochs, while for $k = 2$ minimum ensemble test error generally occurs at fewer epochs compared with the minimum base classifier test error. It appears that the Output Coding method, by virtue of decomposition into artificial two-class problems, is resistant to over-fitting of the base classifier. Furthermore it has been shown experimentally that the error-correcting capability of ECOC may lead to superior results compared with OPC code, even when the base classifier is well-tuned.

Assumptions

μ^+ is the number of class 1 patterns
μ^- is the number of class -1 patterns
B is the pattern dimension
X^+ is a binary (-1,+1) matrix of class 1 patterns with μ^+ rows and B columns
X^- is a binary (-1,+1) matrix of class -1 patterns with μ^- rows and B columns
SS^+ is the sensitivity matrix for class 1 patterns with μ^+ rows and B columns
SS^- is the sensitivity matrix for class -1 patterns with μ^- rows and B columns
S^+ is the B-dimensional vector resulting from spectral summation of class 1 patterns
S^- is the B-dimensional vector resulting from spectral summation of class -1 patterns

Computation of SS^- which is initialised to all zeros

for j = 1: μ^+
  for i = 1: μ^-
    if $\sum_{j=1}^{B}(X^+(j,:) \oplus X^-(i,:)) == 1$ ****
      for m = 1:B
        if $X^+(j,m) \neq X^-(i,m)$
          $SS^-(i,m) = SS^-(i,m) + 1$
        end
      end
    end ****
  end
end

Computation of S^- which is initialised to all zeros

for j = 1: B
  for i = 1: μ^-
    $S^-(j) = S^-(j) + (-1 * X^-(i,j) * SS^-(i,j))$
  end
end

Notes:

(i) The class 1 sensitivity matrix SS^+ and spectral summation vector S^+ may be computed similar to SS^- and S^- and the result is that S^+ is identical to S^-. This equivalence is an example of the principle of duality in Boolean logic. For example, the –1 in the computation of S^- is due to class –1 so that for S^+ becomes +1.

(ii) The conditional marked ******** checks whether there exists unit hamming distance between the jth class 1 pattern and the ith class –1 pattern. Therefore sensitivity matrix SS^- contains binary (0,1) values by virtue of this conditional. For realistic problems, where the truth table is not available, this conditional is removed.

(iii) Each element of the sensitivity matrix is the absolute value for sensitivity, defined in Section 2

$\sigma^+_{mi} = SS^+(m,j)$ $\sigma^-_{mi} = SS^-(m,j)$

Figure 1: Pseudo-code for determining sensitivities and spectral summation for a two-class problem, divided into one set of class 1 and one set of class –1 patterns

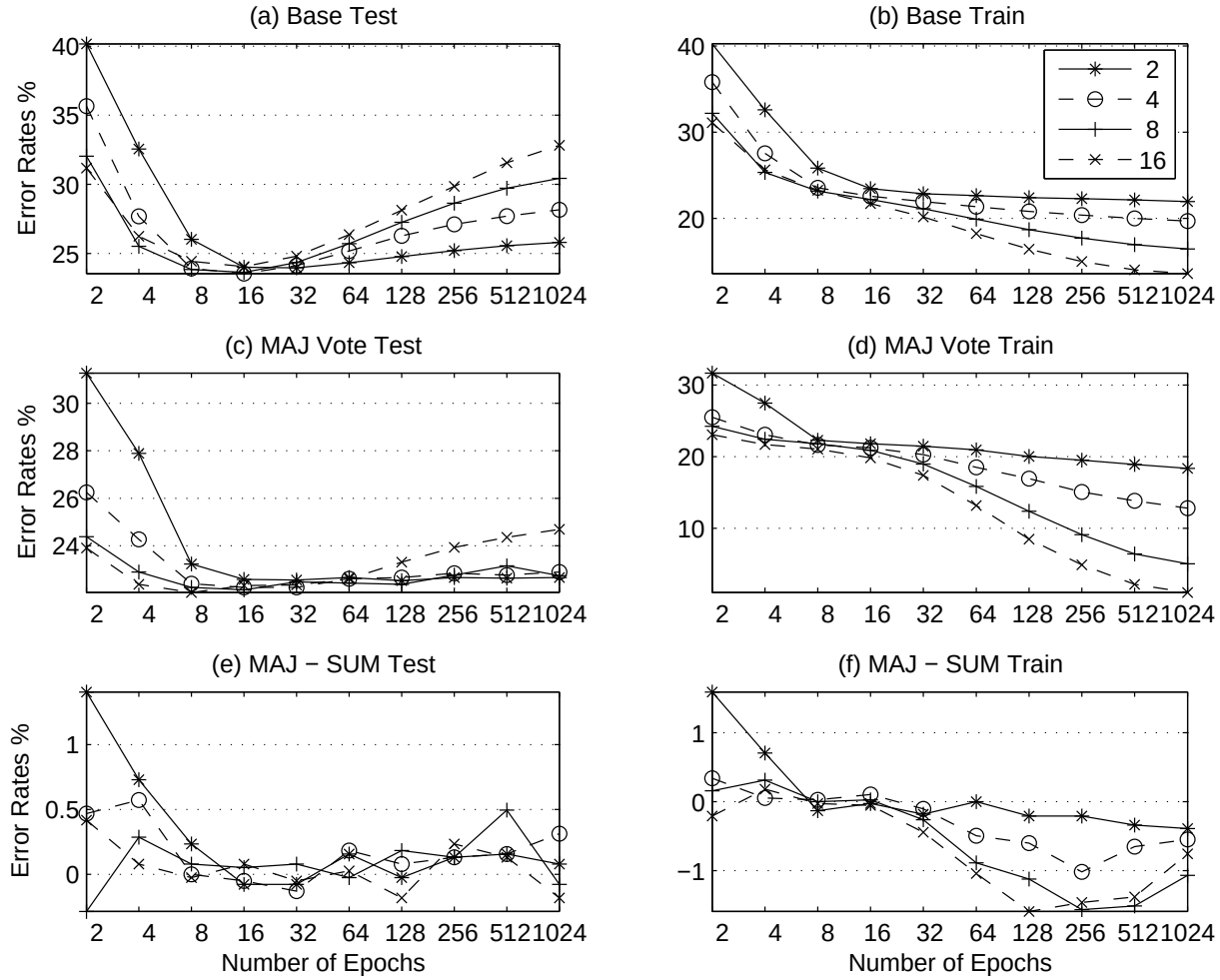

Figure 2: (a) – (d) Train and Test Error rates for Diabetes 50/50 with [2,4,8,16] nodes and Bootstrapping applied
(e) (f) difference in error rate between Majority Vote and Sum

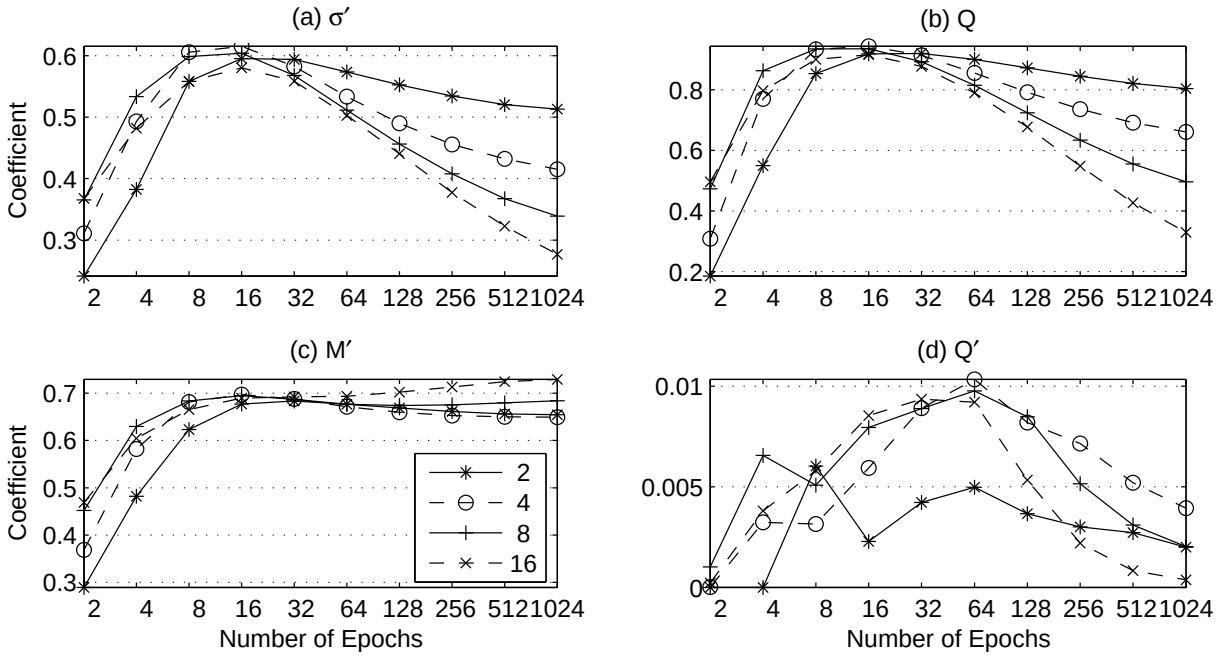

Figure 3: Measures for Diabetes 50/50 for [2,4,8,16] nodes and Bootstrapping applied

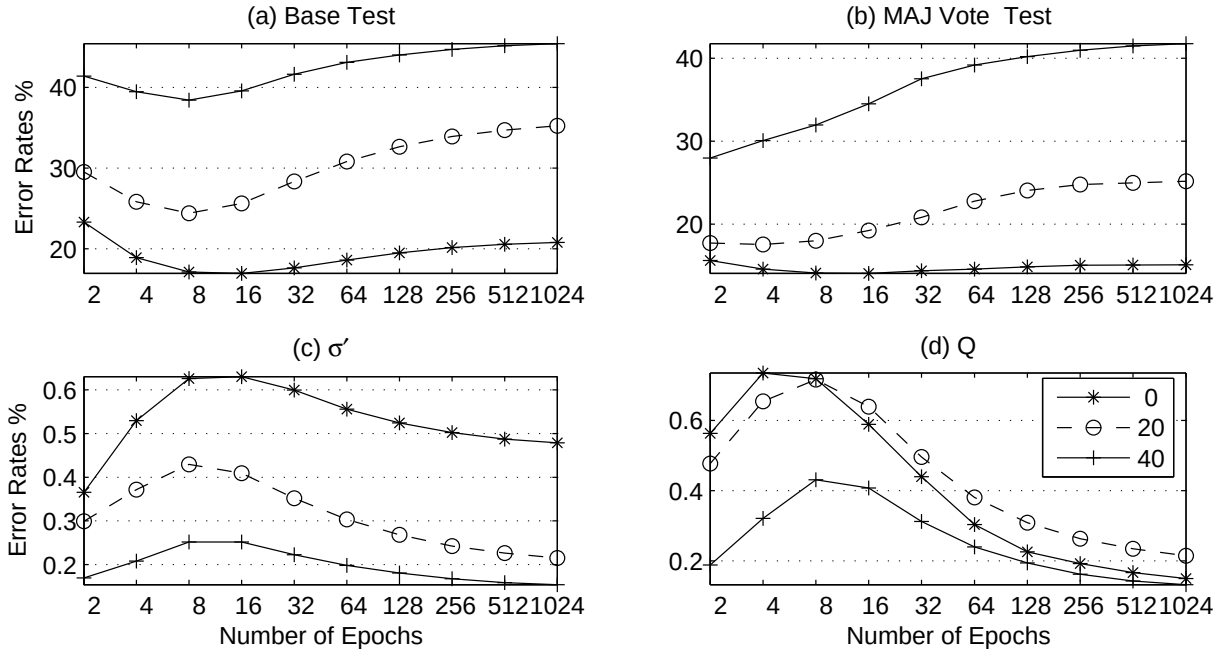

Figure 4: mean test error, σ', Q over seven 20/80 two-class datasets using 8 hidden-node bootstrapped base classifiers for [0,20,40] % noise

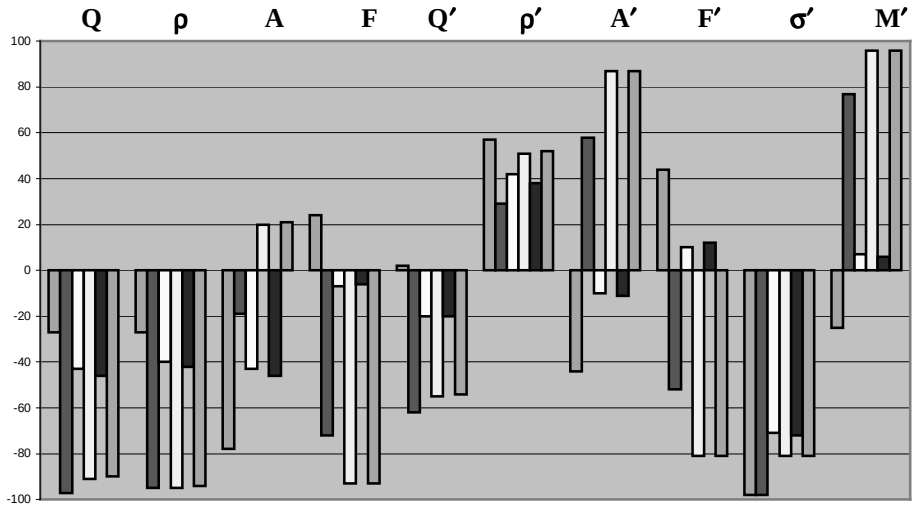

Figure 5: Mean correlation coefficient (x100) of test error with respect to epochs, over seven two-class datasets with 20/80 train/test split, [0,20]% classification noise and bootstrapping applied. Coefficient for each measure is grouped in the order BASE (0, 20), MAJ (0,20), SUM (0,20)

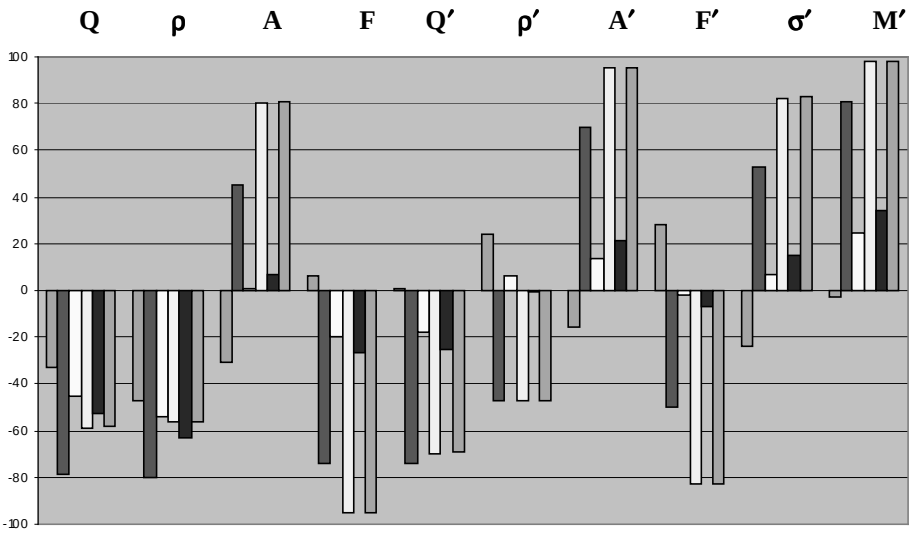

Figure 6: Mean correlation coefficient (x100) of test error with respect to epochs, over seven two-class datasets with 20/80 train/test split, [0,20]% classification noise and *NO* bootstrapping applied. Coefficient for each measure is grouped in the order BASE (0, 20), MAJ (0,20), SUM (0,20)

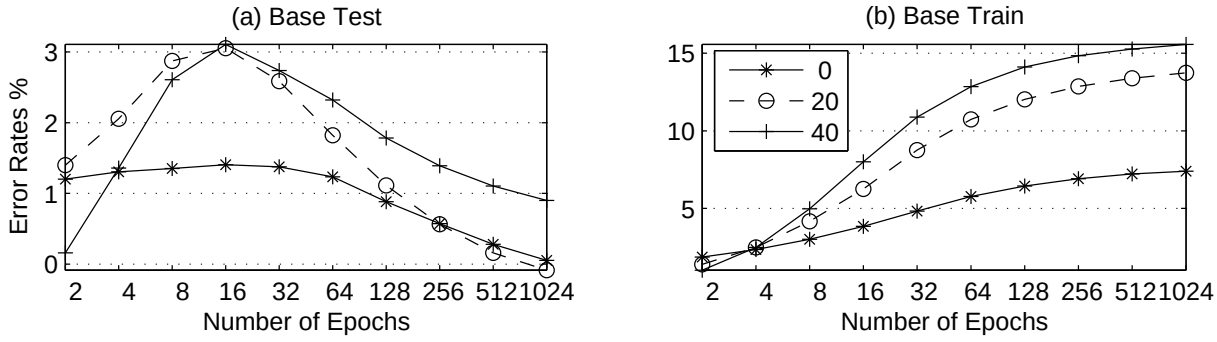

Figure 7: Mean base classifier bootstrapped error rate minus non-bootstrapped error rate over seven 20/80 two-class datasets using 8 hidden-nodes and [0,20,40] % noise

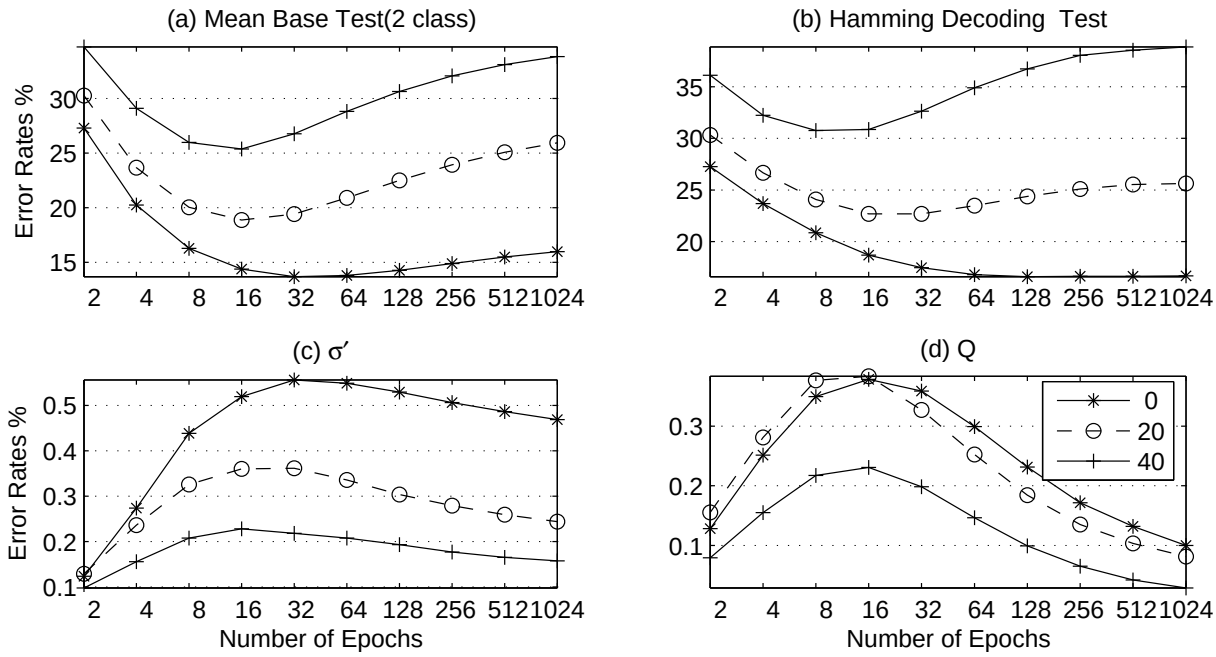

Figure 8: Mean test error rates, σ′, Q over eleven 20/80 multi-class problems using 8 hidden-node bootstrapped base classifiers for [0,20,40] % noise

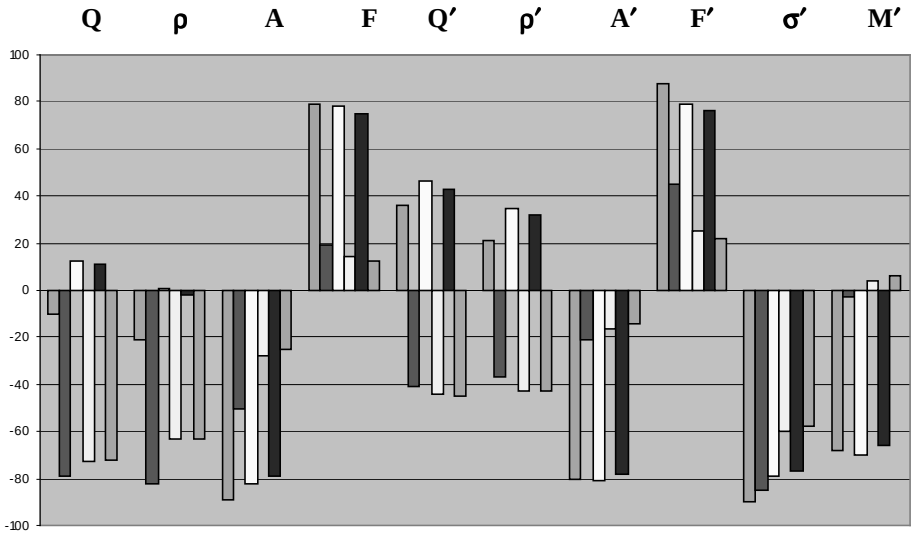

Figure 9: Mean correlation coefficient (x100) of base classifier, Hamming and L1 norm decoded test error with respect to epochs, over eleven multi-class datasets with 20/80 train/test split, [0,20]% classification noise and bootstrapping applied. Coefficient for each measure is grouped in the order BASE (0, 20), HAMMING (0,20), L1 NORM (0,20)

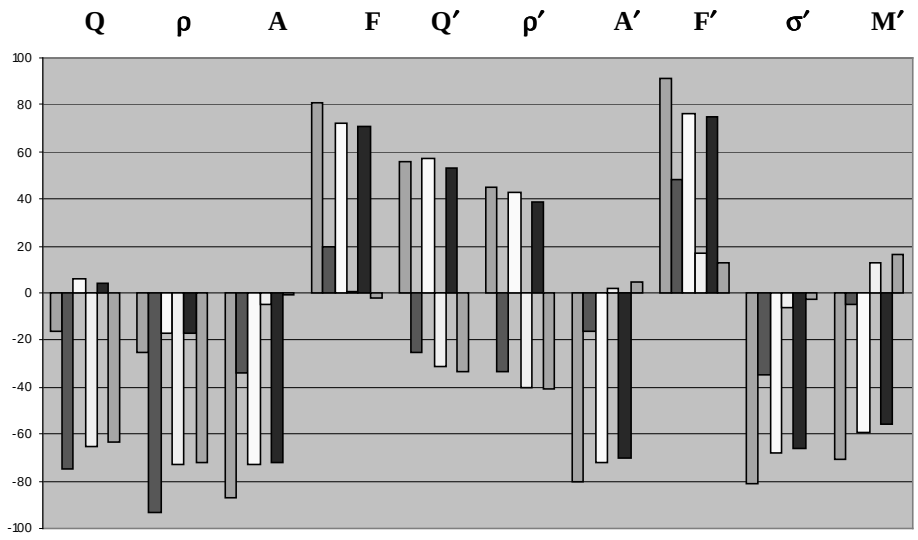

Figure 10: Mean correlation coefficient (x100) of base classifier, Hamming and L1 norm decoded test error with respect to epochs, over eleven multi-class datasets with 20/80 train/test split, [0,20]% classification noise and NO bootstrapping applied. Coefficient for each measure is grouped in the order BASE (0, 20), HAMMING (0,20), L1 NORM (0,20)

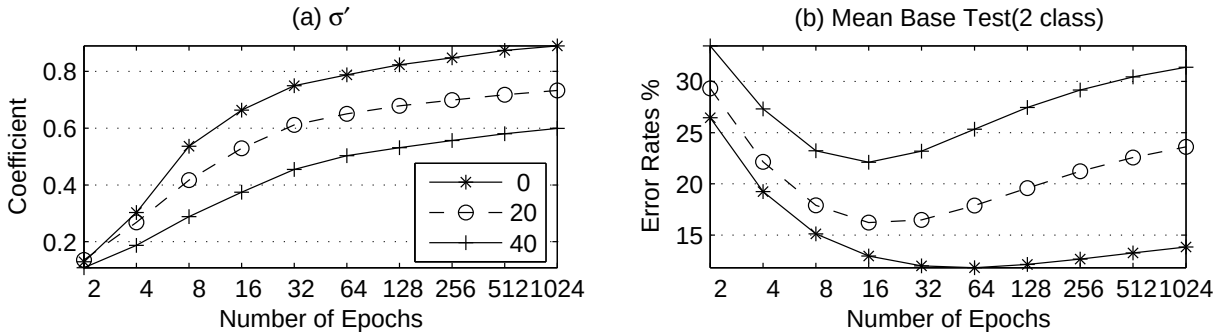

Figure 11: Mean test error rates, σ′ over eleven 20/80 multi-class problems using 8 hidden-node *non*-bootstrapped base classifiers for [0,20,40] % noise

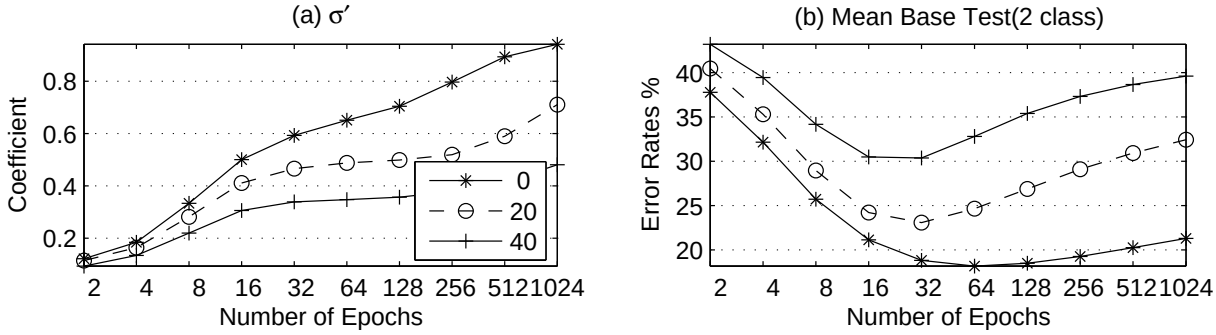

Figure 12: mean test error rates, σ′, Q for VEHICLE 20/80 dataset using 8 hidden-node *non*-bootstrapped base classifiers for [0,20,40] % noise

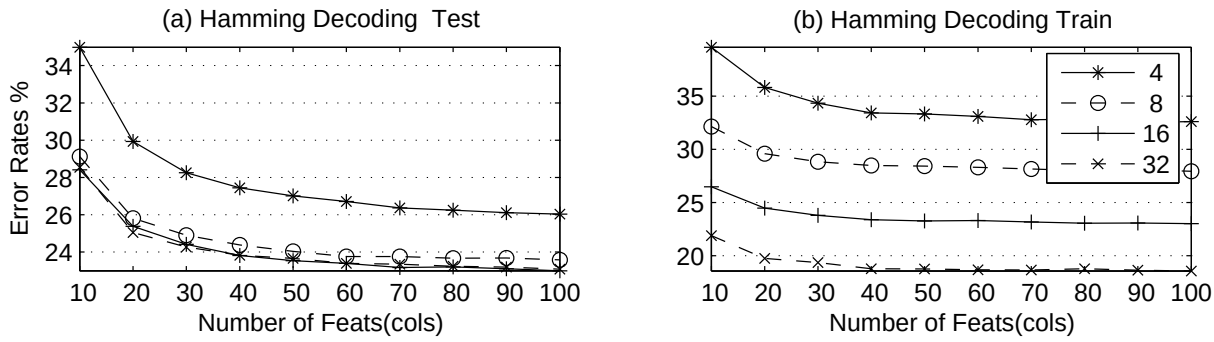

Figure 13: mean ensemble error rates as number of classifiers is varied over eleven 20/80 multi-class problems using 8 hidden-node *non*-bootstrapped base classifiers for [4 8 16 32] epochs

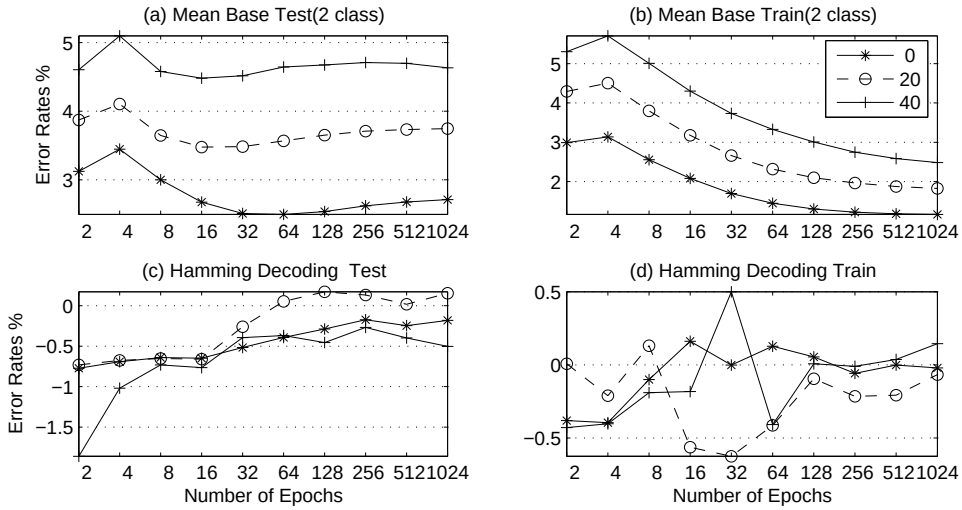

Figure 14: Error rates for *Random code with equal split of labels* (REQ) minus *Random code* (RAN) over eleven 20/80 multi-class problems 16 hidden-node and [0 20 40] % noise

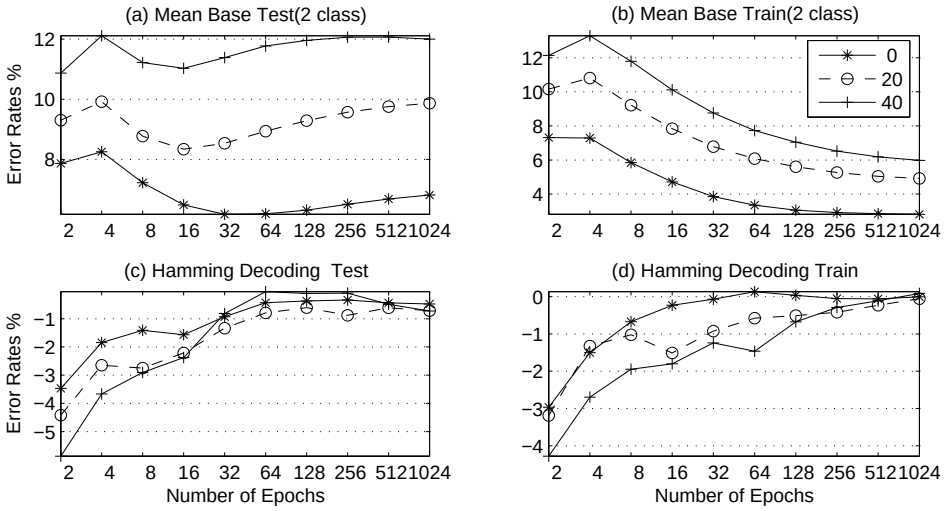

Figure 15: Error rates for *Random code with equal split of labels* (REQ) minus *OPC code* over eleven 20/80 multi-class problems 16 hidden-node and [0 20 40] % noise

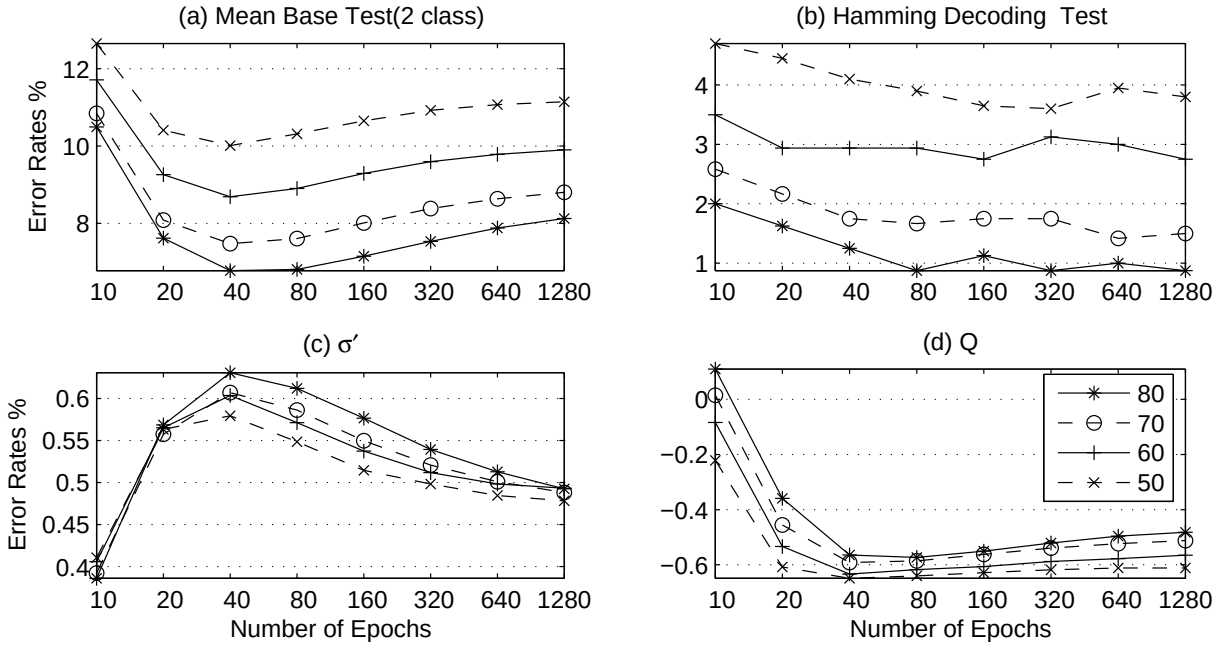

Figure 16: Test error, σ′, Q for ORL database using 16 hidden-node bootstrapped base classifiers for [50/50,60/40,70/30,80/20] train/test splits

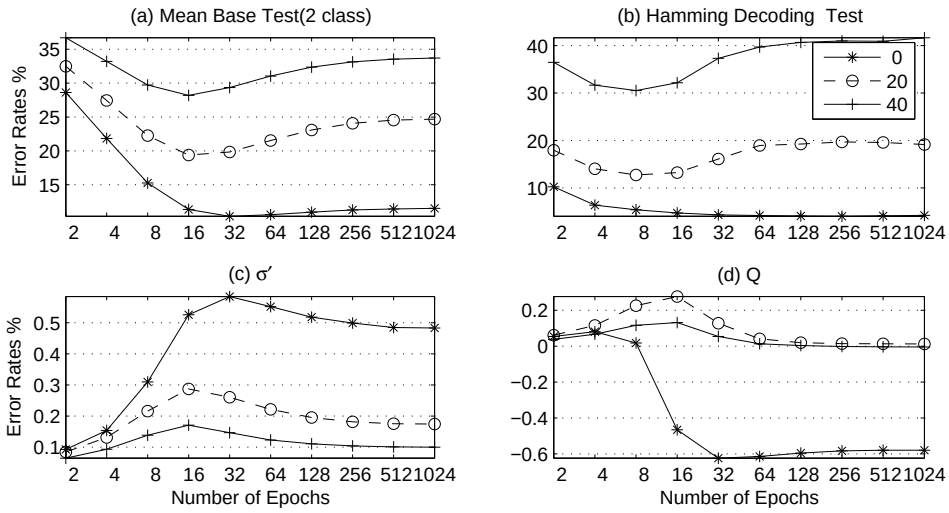

Figure 17: test error, σ', Q for ORL 50/50 database using 16 hidden-node bootstrapped base classifiers for [0,20,40] % noise

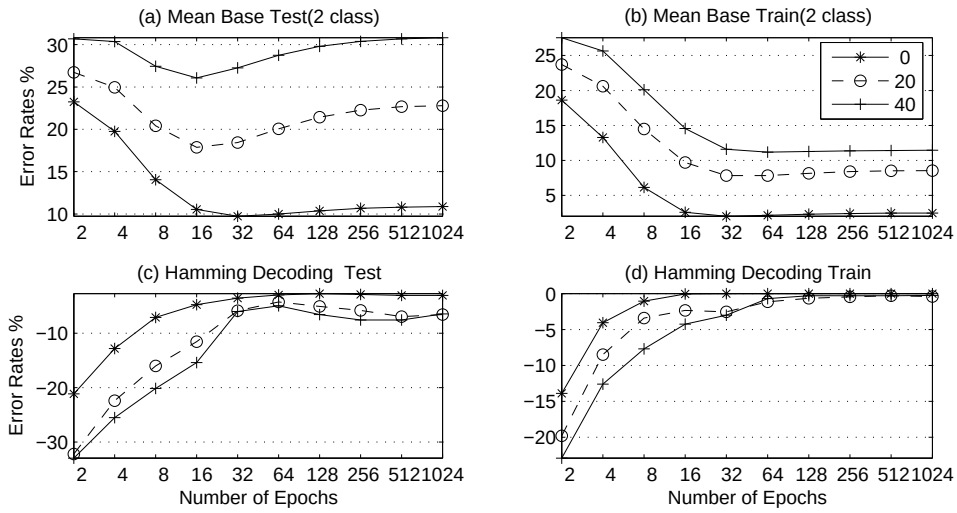

Figure 18: Error rates for *Random code with equal split of labels* (REQ) minus *OPC code* for ORL 50/50 16 hidden-nodes and [0 20 40] % noise

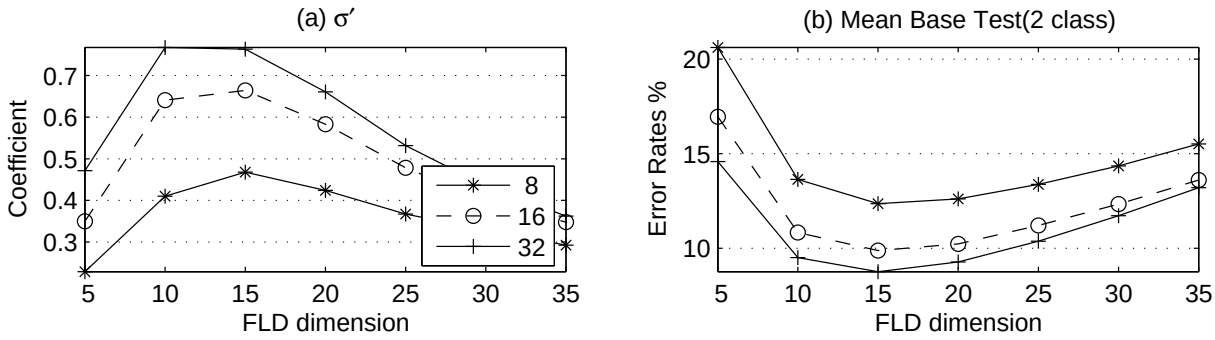

Figure 19: Test error, σ′, Q for ORL 50/50 database versus number of FLD features, using [8,16,32] hidden-node bootstrapped base classifiers

DATASET	#pat	#class	#con	#dis
cancer	699	2	0	9
card	690	2	6	9
credita	690	2	3	11
dermatology	366	6	1	33
diabetes	768	2	8	0
ecoli	336	8	5	2
glass	214	6	9	0
heart	920	2	5	30
iris	150	3	4	0
ion	351	2	31	3
segment	2310	7	19	0
soybean	683	19	0	35
thyroid	7200	3	6	15
vehicle	846	4	18	0
vote	435	2	0	16
vowel	990	11	10	1
wave	5000	3	21	0
yeast	1484	10	7	1

Table 1: Benchmark Datasets showing numbers of patterns, classes, continuous and discrete features

DATASET	Mean Error	Std. Error	Opt # epoch	Ratio 0 %	Ratio 20 %
cancer	3.5	0.4	8	1.05	1.17
card	16.3	0.9	8	1	1.03
credita	16	1.1	16	1	1
diabetes	24.3	1.5	8	1	1.02
heart	17.7	1.3	2	1.03	1.13
ion	12.1	1.6	16	1	1
vote	5.1	1.2	128	1.03	1.06

Table 2: Mean and Std minimum majority vote error rates for two-class problems (20/80 train/test split and 8 hidden nodes), showing optimal number of epochs and ratios with respect to error rate predicted by σ′ for 0% and 20% classification noise

	BASE		MAJ		SUM	
MEAS	0	20	0	20	0	20
Q	4	7	5	7	5	7
ρ	4	7	5	7	4	7
A	6	1	3	0	3	0
F	3	5	4	7	4	7
Q′	5	6	4	3	3	3
ρ′	3	1	3	4	3	4
A′	3	2	3	7	3	7
F′	4	2	4	7	3	7
σ′	7	7	5	6	5	6
M′	3	6	3	7	4	7

Table 3: Number of two-class datasets out of seven for which the correlation coefficient shown in Figure **5** is significant at the ninety-five percent confidence level

	BASE		HAMMING		L^1 NORM	
MEAS	0	20	0	20	0	20
Q	3	9	3	6	3	6
ρ	4	11	2	7	2	7
A	10	4	9	4	8	3
F	8	2	10	6	8	6
Q′	5	4	6	7	5	8
ρ′	7	7	8	6	7	7
A′	9	2	10	6	8	6
F′	10	5	9	4	8	5
σ′	10	9	9	7	6	6
M′	7	2	9	8	7	8

Table 4: Number of multi-class datasets out of eleven for which the correlation coefficient shown in Figure **9** is significant at the ninety-five percent confidence level

DATASET	Mean Error	Std. Error	Opt # epoch	Ratio 0 %	Ratio 20 %
dermatology	3.7	0.9	512	1.06	1.04
ecoli	13.9	2	16	1.1	1.27
glass	34.5	3.4	32	1.01	1.01
iris	4.8	0.8	32	1.02	1.0
segment	4.6	0.5	1024	1.33	1.15
soybean	8	1.1	16	1.02	1.09
thyroid	1.8	0.2	1024	1.28	1.6
vehicle	23.5	1.4	1024	1.14	1.07
vowel	22.9	3.2	1024	1.11	1.01
wave	14.8	0.7	16	1.01	1.01
yeast	41.7	1.1	64	1	1.03

Table 5: Mean and Std minimum Hamming Decoded error rates for multi-class problems (20/80 train/test split and 8 hidden nodes) showing optimal number of epochs and ratios with respect to error rate predicted by σ′ for 0% and 20% classification noise

References

1 L.K. Hansen and P. Salamon, Neural Network Ensembles, IEEE Trans. PAMI, vol.12, 9931001, 1990.

2 T. Bylander, Estimating generalisation error on two-class datasets using out-of-bag estimates, Machine Learning 48, 2002, 287-297.

3 M. J. Er, S. Wu and H. L. Toh, Face Recognition with RBF Neural Networks, IEEE Trans. On Neural Networks, 13 (3), 2002, 697-710.

4 L. Breiman, Bagging Predictors, Machine Learning, 24(2), (1997) 123-40.

5 T. K. Ho**,** The Random Subspace Method for Constructing Decision Forests **,** IEEE Trans. PAMI, 1998, 832 - 844

6 L. Breiman, Randomizing Outputs to Increase Prediction Accuracy, Machine Learning 40 (3), 2000, 229-242.

7 L. I. Kuncheva and C.J. Whitaker, Measures of Diversity in Classifier Ensembles, Machine Learning 51, 2003, 181-207.

8 A. Narasimhamurthy, Evaluation of Diversity Measures for Binary Classifier Ensembles, Proc. 6th Int. Workshop Multiple Classifier Systems, Editors: N. C. Oza, R. Polikar, F. Roli and J. Kittler, Seaside, Calif, June, 2005, Lecture notes in computer science, Springer-Verlag, 267-277.

9 T. Windeatt and R. Tebbs, Spectral Technique for Hidden Layer Neural Network Training, Pattern Recognition Letters, Vol.18(8), 1997, 723-731.

10 T. Windeatt, Recursive Partitioning for Combining Multiple Classifiers, Neural Processing Letters 13(3), June 2001, 221-236.

11 T. Windeatt, Vote Counting Measures for Ensemble Classifiers, Pattern Recognition 36(12), 2003, 2743-2756.

12 A. N. Tikhonov and V. A. Arsenin, Solutions of Ill-posed Problems, Winston & Sons, Washington, 1977.

13 S. L. Hurst, D. M. Miller and J. Muzio, Spectral Techniques in Digital Logic, Academic Press, 1985.

14 R. E. Schapire, Y. Freund and P. Bartlett, Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods, The Annals of Statistics 26(5), (1998) 1651-1686.

15 L. I. Kuncheva, That Elusive Diversity in Classifier Ensembles, Proc. Iberian Conf. On Pattern Recognition and Image Analysis, Mallorca, Spain, Lecture Notes in Computer Science, Springer-Verlag, 2003, 1126-1138.

16 G. James, Variance and Bias for General Loss Functions, Machine Learning, 51 (2) 2003, 115-135.

17 L. Breiman, Arcing Classifiers, The Annals of Statistics 26(3), (1998) 801-849.

18 B. Efron and R.J. Tibshirani, An Introduction to the Bootstrap, Chapman & Hall, 1993.

19 T. G. Dietterich and G. Bakiri, Solving Multi-class Learning Problems via Error-Correcting Output Codes, J. Artificial Intelligence Research 2, 1995, 263-286.

20 T. Windeatt, Spectral Measure for Multi-class Problems, Proc. 4th Int. Workshop Multiple Classifier Systems, Editors: F. Roli, J. Kittler and T. Windeatt, Cagliari, Italy, June, 2004, Lecture notes in computer science, Springer-Verlag, 184-193.

21 T. Windeatt and R. Ghaderi., Coding and Decoding Strategies for Multi-class Learning Problems, Information Fusion, 4(1), 2003, 11-21.

22 E. L. Allwein, R. E. Schapire and Y. Singer, Reducing Multi-class to Binary: A Unifying Approach for Margin Classifiers, J. Machine Learning Research 1, 2000, 113-141.

23 V. Guruswami and A. Sahai, Multi-class Learning, Boosting, and Error-Correcting Codes, Proc twelth Conf on Computational Learning Theory, ACM Press, Santa Cruz, Calif, July,1999,

24 K. Crammer and Y. Singer, Improved Output Coding for Classification Using Continuous Relaxation, in T.G.Dietterich, S. Becker, and Z. Ghahramani (eds.) *Advances in Neural Information Processing Systems 14*. MIT Press, Mass., 2002.

25 K.Crammer, On the Learnability and Design of Output Codes for Multi-class Problems, Machine Learning, 47 (2), 2002, 201-233.

26 R. E. Schapire, Using Output Codes to Boost Multi-class Learning Problems, 14th Int. Conf. of Machine Learning, Morgan Kaufman, 1997, 313--321.

27 J. Kittler, R. Ghaderi, T. Windeatt and J. Matas, Face Verification via Error Correcting Output Codes, Image and Vision Computing, Volume 21, (13-14), 2003, 1163-1169.

28 L. Prechelt, Proben1: A Set of Neural nNtwork Benchmark Problems and Benchmarking Rules, Tech Report 21/94, Univ. Karlsruhe, Germany, 1994.

29 C.J. Merz and P. M. Murphy, UCI Repository of Machine Learning Databases, 1998, http://www.ics.uci.edu/~mlearn/MLRepository.html

30 M. Riedmiller and H. Braun, A Direct Adaptive Method for Faster Backpropagation Learning: The {RPROP} Algorithm, Proc. Intl. Conf. on Neural Networks,San Francisco, Calif., 1993, 586—591.

31 T. Windeatt, Diversity Measures for Multiple Classifier System Analysis and Design, Information Fusion, 6 (1), 2004, 21-36.

32 R. Rifkin and A. Klautau, In defense of One-vs-All Classification, J. Machine Learning Research 5, 2004, 101-141.

33 G.L. Marcialis and F. Roli, Fusion of Appearance-Based Face Recognition Algorithms, Pattern Analysis and Applications, 7(2), 2004,151-163.

34 M. Wang and S. Chen, Enhanced FMAM Based on Empirical Kernel Map, IEEE Trans. Neural Networks, 16(3), 2005, 557-564.

35 M. E. Er, W. Chen and S. Wu, High Speed Recognition Based on Discrete Cosine Transform and RBF Neural Networks, IEEE Trans. Neural Networks 16(3). 2005, 679-691.

36 H. Zhang, B. Zhang, W. Huang and Q. Tian, Gabor Wavelet Associative Memory for Face Recognition, IEEE Trans. Neural Networks 16(1), 2005, 275-278.

37 R. Caruna, S. Lawrence and L. Giles, Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early Stopping, Neural Information Processing Systems, Denver, Colorado, November 28-30, 2000.