Coding and Decoding strategies for multi-class learning problems

Terry Windeatt and Reza Ghaderi
Centre for Vision, Speech and Signal
Processing (CVSSP)
University of Surrey, Guildford, Surrey, GU2 5XH, U.K.
Fax: +44(0) 1483 689554
Phone: +44(0) 1483 689286
T.Windeatt@eim.surrey.ac.uk

August 2, 2002

for submission to Information Fusion

Latex

Coding and Decoding strategies for multiclass learning problems

Abstract

It is known that the Error Correcting Output Code (ECOC) technique, when applied to multiclass learning problems, can improve generalisation performance. One reason for the improvement is its ability to decompose the original problem into complementary two-class problems. Binary classifiers trained on the sub-problems are diverse and can benefit from combining using a simple distance-based strategy. However there is some discussion about why ECOC performs as well as it does, particularly with respect to the significance of the coding/decoding strategy. In this paper we consider the binary (0,1) code matrix conditions necessary for reduction of error in the ECOC framework, and demonstrate the desirability of equidistant codes. It is shown that equidistant codes can be generated by using properties related to the number of 1's in each row and between any pair of rows. Experimental results on synthetic data and a few popular benchmark problems show how performance deteriorates as code length is reduced for six decoding strategies.

Keywords: decision level fusion, multiple classifiers, error-correcting, ECOC, binary coding

1 Introduction

For some classification problems, both two class and multiclass, it is known that the lowest error rate is not always reliably achieved by trying to design a single best classifier. An alternative approach is to employ a set of relatively simple sub-optimal classifiers and to determine a combining strategy that pools together the results. Although various systems of multiple classifiers have been proposed most use similar constituent classifiers, which are often called base classifiers. A necessary condition for improvement by combining is that the results of the base classifiers are not too well correlated, as discussed in [29]. While one strategy to promote independence is to employ different types of base classifier, there are some popular approaches for reducing correlation that are based on perturbing feature sets, perturbing training sets or injecting randomness [9]. For example two well-known training set perturbation methods are Bagging [6] and Boosting [12]. All these perturbation techniques have in common that each base classifier handles the same problem in the sense that the class labelling is identical.

There is another type of correlation reduction technique, aimed solely at multiclass problems, that perturbs class labels. In a method like Error Correcting Output Coding (ECOC) each base classifier solves a sub-problem that uses a different class labelling. Techniques like binary decision clustering [30] and pairwise coupling [14] may also be considered in this category. There are several motivations for decomposing the original multiclass problem into separate and complementary two-class problems. Firstly some accurate and efficient two-class classifiers do not naturally scale up to multiclass. Attention can then be focused on developing an effective technique for the two-class case, without having to consider explicitly the design and automation of the multiclass

classifier. It is also hoped that the parameters of a simple classifier run several times are easier to determine than a complex classifier run once and may facilitate more efficient solutions. Finally, solving different 2-class sub-problems, perhaps repeatedly with random perturbation, may help to reduce error in the original problem.

It needs to be remembered however, that even if the perturbation technique successfully produces diverse classifiers there is still the need to choose or design a suitable combining strategy. Bagging and Boosting originally used respectively the majority and weighted vote, which are both hard-level combining strategies. By hard-level we mean that a single-hypothesis decision is taken for each base classifier, in contrast with soft-level which implies a measure of confidence associated with the decision. The ECOC method was originally motivated by error-correcting principles, and used a Hamming Distance-based hard-level combining strategy. When it could be shown that ECOC produced reliable probability estimates [20], the decision-making strategy was changed to soft-level (L^1 norm equation (2)). There is recent evidence to suggest that the probability estimation error distribution may help determine whether a hard or soft-level combining strategy is more appropriate [18].

ECOC has successfully been used in many application areas [1, 3, 4, 11, 19, 22, 25, 28, 32] but despite many reports of improved performance there is some discussion about why it works well. In particular, it is known that a long random code appears to perform as well or better than a code designed for its error-correcting properties [11, 17, 25], and this has brought into question the significance of the coding strategy. Although the measures of bias/variance and margin have been developed for systems of multiple classifiers, there are recognised shortcomings with these theoretical ap-

proaches [7, 15]. There is no general agreement about which definition to use for bias and variance applied to classification problems, and while the margin concept leads to theoretical bounds on the generalisation error [2], the bounds are not necessarily tight. These ideas do appear to provide useful perspectives for the way systems of multiple classifiers operate, but it is difficult to use the measures to assign specific causes of error. As an example, consider [20] in which it is shown using synthetic data that parts of decision boundaries in ECOC are learned repeatedly, and that consequently both bias and variance according to their definition is reduced. The difficulty is that the effects of variance and bias on classification performance are not guaranteed when defined for the 0-1 loss function, as pointed out in [16]. Also in the case of real datasets there is the difficulty that both noise and Bayes rate are usually neither known nor easy to estimate. In the absence of a complete theory, empirical results continue to provide useful insights and a practical approach to characterising ECOC is to compare error rates for variants of the ECOC strategy.

The paper is organised as follows. The output coding concept is discussed in Section 2 and the original ECOC algorithm is briefly reviewed in Section 3.1. In the ECOC method the coding matrix defines a mapping between spaces (equation (3)), and in section 3.2 we discuss sources of error from this perspective. Properties of equidistant codes are derived in section 3.3 and a method for producing equidistant codes is described in section 3.4. The decision-making strategy may be viewed as classification based on decoding base classifier outputs, and in Section 4 we look at five decoding strategies. Three strategies are based on recovering individual class probabilities, one is based on defining centroid of classes and one is based on Dempster-Shafer combin-

ing. Experimental results on synthetic data using Bayesian base classifiers with added Gaussian noise and a few popular real benchmark problems using MLP base classifier are given in Section 5.

2 Output coding

First let us motivate the need for a suitable output coding by discussing the case of Multi-layer Perceptron (MLP) network, which we use in our experiments in section 5.5. A single multiple output MLP can handle a multiclass problem directly. The standard technique is to use a k-dimensional binary target vector that represents each one of k classes using a single binary value at the corresponding position, for example $[0, ...0, 1, 0, ...0]$ which is sometimes referred to as one-per-class (OPC) encoding. The reason that a single multiclass MLP is not a suitable candidate for use with a more elaborate output coding scheme is that all nodes share in the same training, so errors are far from independent and there is not much benefit to be gained from combining [23]. However a 2-class MLP is a suitable base classifier, and independence among classifiers is achieved by the problem decomposition defined by the coding method, as well as by injection of randomness through the starting weights. Of course, no guarantee can be given that a single MLP with superior performance will not be found, but the assumption is that even if one exists its parameters would be more difficult to determine.

An alternative to OPC is distributed output coding [26], in which k binary vectors are assigned to the k classes on the basis of meaningful features corresponding to each bit position. For this to provide a suitable decomposition some domain knowledge

is required so that each classifier output can be interpreted as a binary feature which indicates the presence or otherwise of a useful feature of the problem at hand. The vectors are treated as code words so that a test pattern is assigned to the class that is closest to the corresponding code word. It is this method of assigning, which is analogous to the assignment stage of error-correcting coding, that provides the motivation for employing ECOC in classification.

The first stage of the ECOC method, as described in section 3.1, gives a strategy to decompose a multiclass problem into complementary two-class sub-problems. The expectation is that the decomposition is beneficial because the two-class problems can be more reliably solved. The second stage of the ECOC method is the decoding step, which was originally based on error-correcting principles under the assumption that the learning task can be modelled as a communication problem, in which class information is transmitted over a channel [10]. In this model, errors introduced into the process arise from various sources including the learning algorithm, features and finite training sample. For a two-class problem, classification errors can be one of two types, either predicted class Ω_1 for target class Ω_2 or predicted class Ω_2 for target class Ω_1. The motivation for encoding multiple classifiers using an error-correcting code with Hamming Distance-based decoding was to provide error insensitivity with respect to individual classification errors. From the transmission channel viewpoint, we would expect that the one-per-class and distributed output coding matrices would not perform as well as the ECOC matrix, because of inferior error-correcting capability.

3 ECOC and coding strategies

When the ECOC technique was first developed it was believed that the ECOC code matrix should be designed to have certain properties to enable it to generalise well [10, 11]. Various coding strategies have been proposed, but most ECOC code matrices that have been investigated previously are binary and problem-independent, that is pre-designed. Random codes have received much attention, and were first mentioned in [10] as performing well in comparison with error-correcting codes. In [11] random, exhaustive, hill-climbing search and BCH coding methods were used to produce ECOC code matrices for different column lengths. Random codes were investigated in [25] for combining Boosting with ECOC, and it was shown that a random code with a near equal column split of labels was theoretically better. Random codes were also shown in [17] to give Bayesian performance if pairs of code words were equidistant, and it was claimed that a long enough random code would not be outperformed by a pre-defined code. In [33] a random assignment of class to codeword was suggested in order to reduce sensitivity to code word selection.

Although various heuristics have been employed to produce better binary problem-independent codes there appears to be little evidence to suggest that performance significantly improves by a clever choice of code, except that OPC is usually inferior [2, 10, 11]. Recently a three-valued code [2] was suggested which allows specified classes to be omitted from consideration (don't care for third value), thereby permitting integrated representation of methods such as all-pairs-of-classes [14]. Another recent development is described in [8] in which problem-dependent codes are investigated and it is claimed that designed continuous codes show more promise than designed discrete

codes.

3.1 Original ECOC algorithm

In the ECOC method, a $k \times b$ binary code word matrix C has one row (code word) for each of k classes, with each column defining one of b sub-problems that use a different labelling. Specifically, for the jth sub-problem, a training pattern with target class w_i ($i = 1...k$) is re-labelled either as class Ω_1 or as class Ω_2 depending on the value of C_{ij} (typically zero or one). One way of looking at the re-labelling is to consider that for each column the k classes are arranged into two super-classes Ω_1 and Ω_2.

A test pattern is applied to the b trained classifiers forming vector

$$\mathbf{y} = [y_1, y_2, ...y_b]^T \tag{1}$$

in which y_j is the real-valued output of jth base classifier.

The distance between output vector and code word for each class is given by

$$L_i^1 = \sum_{j=1}^{b} |C_{ij} - y_j| \tag{2}$$

Equation (2) represents the L^1 norm or Minkowski distance, and the decoding rule is to assign a test pattern to the class corresponding to closest code word $ArgMin_i(L_i^1)$

3.2 Coding and Errors

According to error-correcting theory, an ECOC matrix designed to have d bits error-correcting capability will have a minimum Hamming Distance $2d+1$ between any pair of code words. Assuming each bit is transmitted independently, it is then possible to correct a received pattern having d or fewer bits in error, by assigning the pattern to the code word closest in Hamming distance. While in practice errors are not independent,

the experimental evidence is that application of the ECOC method does lead to reduced test error rate. From the perspective of error-correcting theory, it is therefore desirable to use a matrix C containing code words having high minimum Hamming Distance between any pair. Besides the intuitive reason based on error-correcting theory, this distance property has been confirmed from other perspectives. In [2] it was shown that a high minimum distance between any pair implies a reduced upper bound on the generalisation error, and in [15] it was shown for a random matrix that if the code is equidistant, then decision-making is optimum.

Maximising Hamming Distance between any pair of code words is intended to remove individual classification errors on the re-labelled training sets, but even if classifiers are perfect (Bayesian) there will still be errors due to decoding. The decoding errors can be categorised into those due to inability of sub-problems to represent the main problem, and those due to the distance-based decision rule. Sub-problems are more independent and likely to benefit from combining if Hamming distance between columns is maximised, remembering that a column and its complement represent identical classification problems [11]. The distance-based effect on decoding error can be understood by analysing the relationship between decoding strategy and Bayes decision rule. Consider that the decomposition of a multiclass classification problem into binary sub-problems in ECOC can be interpreted as a transformation between spaces from the original output $\mathbf{q}$ to $\mathbf{p}$, given in matrix form by

$$\mathbf{p} = C^T \mathbf{q} \tag{3}$$

where $\mathbf{q}$ are individual class probabilities

Using the distance-based decision rule from (equation (2)) and equation (3)

$$L_i^1 = \sum_{j=1}^{b} |(\sum_{l=1}^{k} q_l C_{lj}) - C_{ij}| \tag{4}$$

and knowing that $\sum_{l=1}^{k} q_l = 1$ we have

$$L_i^1 = (1 - q_i) \sum_{j=1}^{b} |C_{ij} - C_{lj})| \tag{5}$$

From equation (5), we see that L_i^1 is the product of $1 - q_i$ and Hamming Distance between code words. When all pairs of code words are equidistant, minimising L^1 implies maximising posterior probability which is equivalent to Bayes rule

$$ArgMax_i(q_i) = ArgMin_i(L_i^1) \tag{6}$$

Therefore, any variation in Hamming Distance between pairs of code words will reduce the effectiveness of the decoding strategy.

3.3 Properties of Equidistant Codes

Consider the situation that C is an equidistant code with d bits error-correcting capability, so that $\sum_{l=1}^{b} |C_{il} - C_{jl}| = 2d + 1$ for any pair i, j. Since Hamming Distance between pair i, j is the sum of the number of ones in row i and row j minus number of common ones between i, j we may write

$$\sum_{l=1}^{b} C_{il} + \sum_{l=1}^{b} C_{jl} - 2 \sum_{l=1}^{b} C_{il} C_{jl} = 2d + 1 \tag{7}$$

similar equations can be written for pair i, k and pair j, k

$$\sum_{l=1}^{b} C_{il} + \sum_{l=1}^{b} C_{kl} - 2 \sum_{l=1}^{b} C_{il} C_{kl} = 2d + 1 \tag{8}$$

$$\sum_{l=1}^{b} C_{jl} + \sum_{l=1}^{b} C_{kl} - 2\sum_{l=1}^{b} C_{jl}C_{kl} = 2d + 1 \tag{9}$$

From equations (7), (8),(9) after re-arranging

$$\sum_{l=1}^{b} C_{il}C_{jl} = \sum_{l=1}^{b} C_{kl}C_{jl} = \sum_{l=1}^{b} C_{kl}C_{il} = m \tag{10}$$

where m is number of common bits in code word, and

$$\sum_{l=1}^{b} C_{il} = \sum_{l=1}^{b} C_{kl} = \sum_{l=1}^{b} C_{jl} = n \tag{11}$$

where n is the number of ones in each row. Therefore if C is an equidistant matrix, the number of ones in different rows are the same, and the number of common ones between any pair of rows is equal.

3.4 Code selection

From section 3.2, the main considerations in designing ECOC matrices are as follows

- minimum Hamming Distance between rows (error-correcting capability)
- variation of Hamming Distance between rows (effectiveness of decoding)
- number of columns (repetition of different parts of sub-problems)
- Hamming Distance between columns and complement of columns (independence of base classifiers)

From the theory of error-correcting codes [24] we know that finding a matrix with long code words, and having maximum and equal distance between all pairs of rows is

complex. In the experiments in Section 5, we compare random, equidistant and non-equidistant code matrices as number of columns is varied, but do not address explicitly the distance requirement between columns. Lack of experimental results on equidistant codes in previous work can be attributed to the difficulty in producing them. We produce equidistant codes by using the BCH method [24], which employs algebraic techniques from Galois field theory. Although BCH has been used before for ECOC, our implementation is different in that we first over-produce the number of rows (BCH requires number to be power of 2), before using properties (10) and (11) to select a subset of rows. Of course these properties provide only necessary conditions for equidistant codes and so cannot be used to generate the codes in isolation.

4 Decoding Strategies

A variety of soft-level decoding strategies have been proposed for ECOC besides L^1 norm, referred to here as L1-ECOC. Other decoding strategies discussed in this section include three approaches for recovering individual class probabilities, which are Least Squares (LS-ECOC), Inverse Hamming Distance (IHD-ECOC) and Linear (Lin-ECOC). For these three methods the classification decision is based on maximising the probability estimate. Two further methods are also described, Dempster-Shafer (DS-ECOC) based on maximising probability mass function [27] and Centroid (Cen-ECOC), based on minimising Euclidean distance to centroid of classes. The effect of codes on performance has not previously been investigated for the other decoding strategies in the same way as it has for L1-ECOC.

The original Hamming Distance decoding strategy (HD-ECOC) was shown in [20]

to be analogous to majority vote over the classes. Besides L1-ECOC, LS-ECOC has been the most extensively investigated. The justification for LS-ECOC in terms of probability estimation was reported in [21], and in [15] LS-ECOC was extended by incorporating ridged regression when b is small. In [15] it was also shown that for certain geometrical arrangements of patterns Cen-ECOC is to be preferred to LS-ECOC. From the perspective of computational learning theory, it was shown in [2] that a loss-based decoding scheme theoretically and experimentally outperforms HD-ECOC for a variety of codes.

4.1 Least squares approach to recovering probabilities (LS-ECOC)

Recovering individual class probabilities from super-class probabilities is easily accomplished if the individual probability estimates are exact and if the k independent equations that result from equation (3) can be solved. In practice, base classifiers will not produce correct probabilities but we can model the error assuming that the expected value of the error is zero for unbiased probability estimates [13]. In general C^T is not a square matrix, and so does not have an inverse, but a solution to equation (3) can be found using the method of least squares, which means finding $\hat{\mathbf{q}}$ which minimises a cost function such as

$$R_p = (\hat{\mathbf{p}} - Z^T\mathbf{q})^T(\hat{\mathbf{p}} - Z^T\mathbf{q}) = \hat{\mathbf{p}}^\mathbf{T}\hat{\mathbf{p}} - \mathbf{2}\hat{\mathbf{p}}^\mathbf{T}\mathbf{p} + \mathbf{p}^\mathbf{T}\mathbf{p} \tag{12}$$

Therefore the solution is given by

$$\hat{\mathbf{q}} = (CC^T)^{-1}C\hat{\mathbf{p}} \tag{13}$$

From equation (12) and (13) we may write a cost function for q

$$R_q = \hat{\mathbf{q}}^T CC^T\hat{\mathbf{q}} - 2\hat{\mathbf{q}}^T CC^T\mathbf{q} + \mathbf{q}^T CC^T\mathbf{q} \tag{14}$$

Using properties (11) and (10)

$$CC^T = \sum_{l=1}^{b} C_{il} C_{lj}^T = \sum_{l=1}^{b} C_{il} C_{jl} = \begin{cases} n & \text{if i=j} \\ m & \text{otherwise} \end{cases} \tag{15}$$

after substituting equation (15) into (14) and using (12)

$$R_p = (n - m) R_q \tag{16}$$

Therefore an equidistant code has the desirable property that it simultaneously minimises the cost function for $\mathbf{q}$ (R_q) as well as for $\mathbf{p}$ (R_p) [13].

4.2 Inverse Hamming Distance approach to recovering probabilities (IHD-ECOC)

From equation (2) we may write the matrix equation

$$\mathbf{L^1} = \Delta \mathbf{q} \tag{17}$$

where $\mathbf{L^1} = [L_1^1, L_2^1, ...L_k^1]$ and Δ is a matrix whose element Δ_{ij} represents the Hamming Distance between row i and row j. Since the elements of Δ are non-negative, Δ can be inverted to find an estimate for $\mathbf{q}$.

4.3 Linear estimation approach to recovering probabilities (Lin-ECOC)

The assumption here is that there is a linear relationship between outputs of the base classifiers and individual class probabilities.

$$\hat{q}_i = \sum_{j=1}^{b} W_{ij} y_j \qquad i = 1...k \tag{18}$$

where

$$\sum_{j=1}^{b} W_{ij} = 1 \qquad i = 1...k \tag{19}$$

Although there are several ways to estimate the weights, a simple approach is to count, for each column, the number of times that a training pattern belongs both to class w_i and to superclass Ω_1. The count is repeated for all classes ($i = 1, ...k$) and normalised as in (19).

4.4 Centroid of classes (Cen-ECOC)

In L1-ECOC the assignment of a test pattern is based on minimising L^1 distance to a target vertex of the multi-dimensional hypercube. However, if a class can be considered to cluster around a point the centroid of the class might be a more suitable target since each class would then define its own representative. In Cen-ECOC, the jth element of the b-dimensional centroid for each class is computed by taking the average over j base classifier outputs for all patterns of a class. A pattern is classified according to the closest centroid in the sense of Euclidean distance

$$Argmin_i ||\mathbf{y} - \mathbf{c_i}|| \text{ where } \mathbf{c_i} \text{ is the centroid for class } w_i$$

4.5 Dempster-Shafer (DS-ECOC)

The Dempster-Shafer approach to combining provides a mathematical framework which may be regarded as a generalised form of Bayesian statistics [27]. It is based upon the concept of probability mass function (basic probability assignment) which assigns a value to a subset of propositions. We assume that there are k elemental propositions, one each for the k classes to which a pattern can be assigned. In the context of ECOC each column of the code matrix C defines a partitioning into two super-classes as described in Section 3.1. A superclass is a subset of the k classes and represents one of 2^k

possible subsets in Dempster-Shafer theory. Therefore, each column can be regarded as providing evidence for class membership, which can be combined with evidence from previous columns.

The mass function of the jth base classifier is set to $m_j(A_j)$ = $[y_j \quad 1-y_j]$ for the two super-classes Ω_1 and Ω_2, where y_j is the jth base classifier output (defined in (1)), and A_j represents the decomposition defined by the jth column of C. The evidence for the $(j+1)th$ base classifier can be combined with evidence from the jth by invoking Dempster's rule of combination to calculate the orthogonal sum $m_j \oplus m_{j+1}$

$$[m_j \oplus m_{j+1}] = 1/K \sum_{A_j \cap A_{j+1}} m_j(A_j) m_{j+1}(A_{j+1}) \tag{20}$$

where normalisation factor K is given by $1 - \sum_{A_j \cap A_{j+1}=\{\}} m_j(A_j) m_{j+1}(A_{j+1})$. Initially with $j = 1$ equation (20) gives $m_1 \oplus m_2$, and equation (20) is applied recursively for $j = 2, ...b-1$ to find the mass function for the combination of b classifiers. The decision strategy is to assign the class corresponding to the maximum probability mass function.

5 Experimental Results

5.1 Code Matrices

To find equidistant codes we have used the BCH method described in section 3.4. The following code matrices are used in these experiments:

C1: a $k \times k$ unitary code OPC (one-per class)

C2: a $k \times 7$ matrix with randomly chosen binary elements

C3: a $k \times 7$ BCH code (minimum distance of 3, non-equal)

C4: a $k \times 7$ BCH code with equal distance of 4

C5: a $k \times 15$ matrix with randomly chosen binary elements

C6: a $k \times 15$ BCH code with equal distance of 8

C7: a $k \times 31$ BCH code with equal distance of 16

5.2 Artificial Benchmark

We use two dimensional data so that decision boundaries can be visualised. Table 1 shows parameters for five equal size groups of random vectors having normal distribution: $p(\mathbf{x}|w_i) = \frac{1}{2\pi\sigma_i^2} exp[\frac{||\mathbf{x}-\mu_i||^2}{-2\sigma_i^2}] \quad for \quad i = 1, 2, ..., 5.$

For this problem, the base classifiers are not trained, but using the parameters from table 1 we find the posterior probability of super-class membership. Since the prior probabilities are known, we can use Bayes' formula to re-interpret the decision rule for assigning pattern $\mathbf{x}$ to class w_i, so that

$ArgMax_i(p(w_i|\mathbf{x})) \longrightarrow ArgMax_i(p(\mathbf{x}|w_i))$

The Bayes rate of correct classification for this five class problem is 74.28%, and the Bayes (optimum) decision boundary is shown in figure 1.

To determine the robustness of the ECOC framework, Gaussian random noise with different mean (μ_n) and variance (σ_n^2) is added to the super-class probabilities. A 'perfect' base classifier would achieve Bayesian performance as calculated using Bayes' formula, so Gaussian noise is intended to simulate noisy imperfect base classifiers. To ensure that the comparison of performance for different code matrices is fair, the noise is generated once and used with each code matrix.

5.3 L1-ECOC and coding strategies

To demonstrate the difference between one-per-class (OPC) and the ECOC concept, decision boundaries are plotted in figure 2(a) (code C1) and figure 2(b) (code C4) for several individual classifiers. By comparing with the Bayes boundary shown in figure 1, we can see that each classifier concentrates on different part of the input space. So each classifier has a less complex problem to handle compared with the complete boundary that would, for example, be handled by a single multiclass classifier. Figure 2(b) also shows how some parts of the boundary appear in more than one classifier, and so are learned repeatedly, as investigated in [20, 31]. For added noise ($\mu_n = 0$, $\sigma_n^2 = 0.25$) figure 3(a) and figure 3(b) show respectively the C1 and C4 decision boundaries of the ensemble as the number of individual classifiers is increased. A comparison shows that code C4, by virtue of repeated learning of parts of decision boundaries, more effectively reduces variance.

For all codes specified in Section 5.1, the percentage match with Bayes correct classification rate (c.c.r./Bayes) as noise variance is increased is shown in Table 2 ($\mu_n = 0$ and $\mu_n = 0.5$). The OPC code C1 is just shown for reference, and it has been demonstrated elsewhere that ECOC is superior to OPC [11]. For code matrices with equidistant rows (C4, C6, C7) and no added noise, error due to decoding is zero (100% match) as expected. For 7-bit (C2,C3,C4) and 15-bit (C5, C6) codes, increased error-correction capability leads to better performance, with equidistant codes (C4, C6) the best. For longer code words, improved performance is evident even if error-correction capability is reduced; for example the random code matrix C6, found to have only 2 bits error-correction capability performs better than C4 having 3 bits error-correction capability.

Finally, repeating the same sub-problems appears to enhance performance, for example C7 is more robust than C5 even though the number of unique classification problems is the same (due to $k = 5$ and complementary column labelling giving same problem).

5.4 Comparison of decoding strategies: Synthetic data

The six decoding strategies L1-ECOC, Cen-ECOC, LS-ECOC, IHD-ECOC, Lin-ECOC, DS-ECOC which were described in section 4, were tested for various levels of added noise. The percentage match with respective Bayes rates is given for no added noise in table 3, for ($\mu_n = 0.5$, $\sigma_n^2 = 0$) in table 4 and for ($\mu_n = 0$, $\sigma_n^2 = 0.5$) in table 5. The Bayes rate is 72.08% for ($\mu_n = 0.5, \sigma_n^2 = 0$) and 71.82% for ($\mu_n = 0, \sigma_n^2 = 0.5$).

Table 3 shows that LS-ECOC and IHD-ECOC with no added noise have zero decoding error for all codes. In addition, there is zero decoding error for equidistant codes with L1-ECOC and Lin-ECOC. However for non-equidistant codes Lin-ECOC performs better than L1-ECOC, which we believe is due to the normalisation step (Section 4.3).

For added bias ($\mu_n > 0$), table 4 shows that equidistant codes give very low decoding error for all codes except Cen-ECOC and DS-ECOC. For non-equidistant codes Lin-ECOC is least sensitive to bias and Cen-ECOC is most sensitive. This is expected since the distance-based strategy in L1-ECOC, IHD-ECOC and Cen-ECOC will be adversely affected by bias, particularly the calculation of the centroid.

For added variance ($\sigma_n^2 > 0$) table 5 shows that equidistant codes give very similar performance for L1-ECOC, IHD-ECOC and Lin-ECOC. For longer codes there is little difference in performance, except for DS-ECOC which is adversely affected by high noise variance.

In all three tables 3, 4 and 5 we can observe that for the same number of columns (7 and 15) equidistant codes perform best. However for the longer code the difference in sensitivity to bias and variance is reduced.

5.5 Comparison of decoding strategies: Natural data

To determine if the results obtained on artificial data hold for higher dimensional real data we have experimented with some popular multiclass classification benchmarks. For the base classifier we have chosen a feedforward neural network which is capable of making a global approximation to the classification boundary. We tested Codes C1-C6 on five databases from [5], using a conventional single hidden layer MLP trained by Levenberg-Marquardt algorithm, with default parameters and fifty epochs of training. The number of hidden nodes of MLP, numbers of training and test patterns and number of classes for the problems are shown in table 6. The mean and standard deviation of classification rates for ten independent runs are given for 'satellite' database in table 7, and a similar table was produced for each dataset. It may be observed that equidistant codes C4, C6 exhibit lower variance across all decoding strategies.

Rather than show significant differences of decoding strategies for each particular dataset (with the exception of LS, IHD, Lin for code C2 the differences do not show up on significance tests), in Table 8 we give a comparison over all five datasets for codes C1 to C6. In Table 8 each entry denotes the ratio of correct classification of the respective code/decode strategy to the best (mean over ten runs) classification rate for that problem over all codes. The best (mean) classification rates (%) achieved for the five databases were zoo/95, car/75, vehicle/76, glass/61, satellite/81. From table 8, we can observe that longer codes generally perform better but decoding strategy

appears to have little impact on performance, except for code C2. For 7-bit code, equidistant performs best (C2 and C3 compared with C4). However for the 15-bit code, the individual results show that random is better for two datasets, while equidistant is better for the other three, but there is no difference in the average ratio. It can also be observed that no decoding strategy outperforms DS-ECOC for any code.

6 Discussion and Conclusion

Three types of binary problem-independent codes have been tested with six soft-level ECOC decoding strategies; random, equidistant BCH, and non-equidistant BCH were compared for various code lengths. The six decoding strategies used in the comparison were L1-ECOC (L^1 norm), Cen-ECOC (Centroid of classes), LS-ECOC (Least squares) IHD-ECOC (Inverse Hamming Distance), Lin-ECOC (Linear estimation) and DS-ECOC (Dempster-Shafer).

By using synthetic data with Bayesian base classifiers, we were able to measure the error due to the decoding strategy. It was evident that there was no decoding error for LS-ECOC and IHD-ECOC with any code, nor for L1-ECOC and Lin-ECOC with equidistant codes. In the presence of added Gaussian noise, the decoding strategies exhibited different sensitivity to bias and variance as described in Section 5.4. In particular, the distance-based decoding strategies (L1-ECOC, IHD-ECOC, Cen-ECOC) were adversely affected by bias for non-equidistant codes. However sensitivity to added noise was reduced as length of code was increased. For the five natural data sets tested, there was little difference in performance of decoding strategies except with 7-bit random code. For longer codes it appeared that the choice of random versus equidistant

code may be problem-dependent, but that equidistant codes reduce variance.

Overall we have demonstrated using synthetic and natural data that equidistant codes are superior, at least for shorter codes; but as length of code word is increased it appears that, as others have found, the coding/decoding strategy becomes less significant. Further work is aimed at validating these conclusions as the probability estimation error distribution of base classifiers varies, for both soft and hard-level decoding strategies.

In this paper we showed that there are five alternatives to L1-ECOC that can provide comparable classification performance. For certain applications it may be desirable to recover individual class probabilities from super-class probabilities, and we described four methods LS-ECOC, Lin-ECOC, IHD-ECOC and DS-ECOC that can accomplish this.

Acknowledgement Reza Ghaderi is grateful to the Ministry of Culture and Higher Education of Iran for its financial support during his Ph.D studies.

References

[1] D.W. Aha and R. L. Bankert. Cloud classification using error-correcting output codes. *Artificial Intelligence Applications: Natural Resources, Agriculture, and Environmental Science*, 11(1):13–28, 1997.

[2] E.L. Allwein, R.E. Schapire, and Y. Singer. Reducing multi-class to binary: A unifying approach for margin classifiers. *Machine learning research*, 1:113–141, 2000.

[3] E. Alpaydin and E. Mayoraz. Learning error-correcting output codes from data. In *Proceeding of Int. Conf. Artificial Neural Networks ICANN'99*, pages 743–748, Edinburgh, U.K., September 1999.

[4] A. Berger. Error-correcting output coding for text classification. In *Proccedings of Int. Joint Conf. Artificial Intelligence, IJCAI'99*, Stockholm, Sweden, 1999.

[5] C.L. Blake and C.J. Merz. UCI repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Sciences, 1998. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[6] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1997.

[7] L. Breiman. Some infinity theory for predictor ensembles. Technical Report 577, Statistic Dept. university of California, Berkeley CA., 2000.

[8] K. Crammer and Y. Singer. On the learnability and design of output codes for multiclass problems. *Machine Learning*, to appear.

[9] T.G. Dietterich. Ensemble methods in machine learning. In J.Kittler and F.Roli, editors, *Multiple Classifier Systems, MCS2000*, pages 1–15, Cagliari, Italy, 2000. Springer Lecture Notes in Computer Science.

[10] T.G Dietterich and G. Bakiri. Error-correcting output codes: A general method for improving multiclass inductive learning programs. In *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91)*, pages 572–577. AAAI Press, 1991.

[11] T.G. Dietterich and G Bakiri. Solving multi-class learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

[12] Y. Freund and R.E. Schapire. A decision-theoretic generalisation of on-line learning and application to boosting. *Journal of computer and system science*, 55:119–139, 1997.

[13] R. Ghaderi and T. Windeatt. Least squares and estimation measures via error correcting output code. In *2nd Int. Workshop Multiple Classifier Systems,Lecture notes in computer science, Springer-V erlag*, pages 148–157, Jul 2001.

[14] T. Hastie and R Tibshirani. Classification by pairwise coupling. *The annals of statistics*, 2:451–471, 1998.

[15] G. M. James. *Majority Vote Classifiers: Theory and Applications*. PhD thesis, Dept. of Statistics, Univ. of Stanford, Calif., May 1998.

[16] G. M. James. Variance and bias for general loss functions. *Machine Learning*, to appear.

[17] G. M. James and T. Hastie. The error coding method and PICT's. *Computational and Graphical Statistics*, 7:377–387, 1998.

[18] J. Kittler and F. M. Alkoot. Relationship of sum and vote fusion strategies. In *2nd Int. Workshop Multiple Classifier Systems,Lecture notes in computer science, Springer-V erlag*, pages 339–348, Jul 2001.

[19] J. Kittler, R. Ghaderi, T. Windeatt, and G. Matas. Face verification using error correcting output codes. In *Computer Vision and Pattern Recognition CVPR01*, Hawaii, December 2001. IEEE Press.

[20] E.B. Kong and T.G. Diettrich. Error-correcting output coding corrects bias and variance. In *12th Int. Conf. of Machine Learning*, pages 313–321, San Fransisco, 1995. Morgan Kaufmann.

[21] E.B. Kong and T.G. Diettrich. Probability estimation via error-correcting output coding. In *Int. Conf. of Artificial Inteligence and soft computing*, Banff,Canada, 1997.

[22] F. Leisch and K. Hornik. Combining neural networks voting classifiers and error correcting output codes. In I. Frolla and A. Plakove, editors, *MEASURMENT 97*, pages 266–269, Smolenice, Slovakia, May 1997.

[23] F. Masulli and G. Valentini. Effectiveness of error correcting output codes in multiclass learning problems. In J.Kittler and F.Roli, editors, *Multiple Classifier Systems, MCS2000*, pages 107–116, Cagliari, Italy, 2000. Springer Lecture Notes in Computer Science.

[24] W.W. Peterson and J.R. Weldon. *Error-Correcting Codes*. MIT press, Cambridge,MA, 1972.

[25] R.E. Schapire. Using output codes to boost multiclass learning problems. In *14th International Conf. on Machine Learning*, pages 313–321. Morgan Kaufman, 1997.

[26] T.J. Sejnowski and C.R. Rosenberg. Parallel networks that learn to pronounce english text. *Complex systems*, 1:145–168, 1987.

[27] G. Shafer. *A mathematical theory of evidence*. Princeton Univ Press, Princeton, New Jersey, 1976.

[28] E. Tapia, E.J.C. Gonzalez, and J. Garcia-Villalba. On the design of error adaptive ecoc algorithms. In *Integrating Aspects of Data Mining, Decision Support and Meta-Learning, IDDM*, pages 139–150, Freiburg, Sept 2001.

[29] K. Tumer and J. Ghosh. Error correlation and error reduction in ensemble classifiers. *Connection Science, special issue on combining artificial neural ne tworks: ensemble approaches*, 8(3):385–404, 1996.

[30] C.L. Wilson, P.J. Grother, and C.S. Barnes. Binary decision clustering for neural-network-based optical character recognition. *Pattern Recognition*, 29(3):425–437, 1996.

[31] T. Windeatt and R. Ghaderi. Binary codes for multi-class decision combining. In *14th Annual International Conference of Society of Photo-Optical Instrumentation Engineers (SPIE)*, volume 4051, pages 23–34, Florida,USA, April 2000.

[32] T. Windeatt and R. Ghaderi. Multi-class learning and error-correcting code sensitivity. *Electronics Letters*, 36(19):1630–1632, Sep 2000.

[33] T. Windeatt and R. Ghaderi. Binary labelling and decision level fusion. *Information Fusion*, 2(2):103–112, 2001.

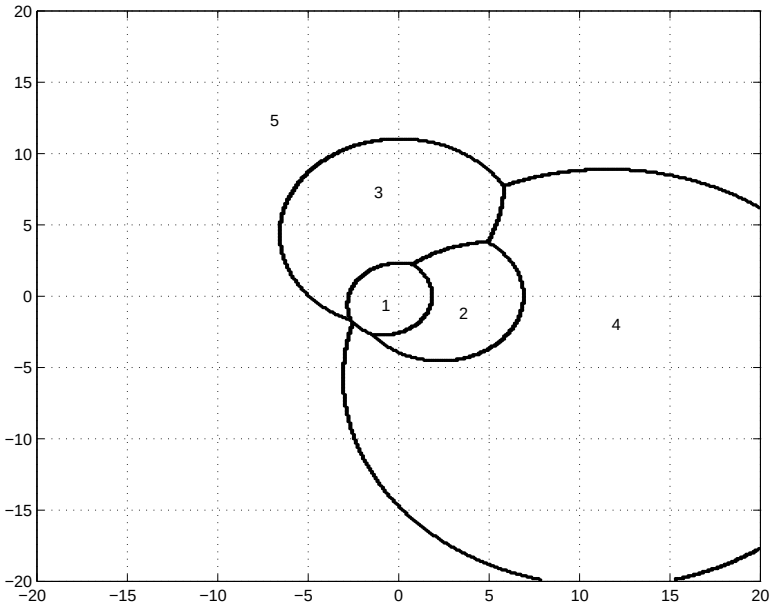
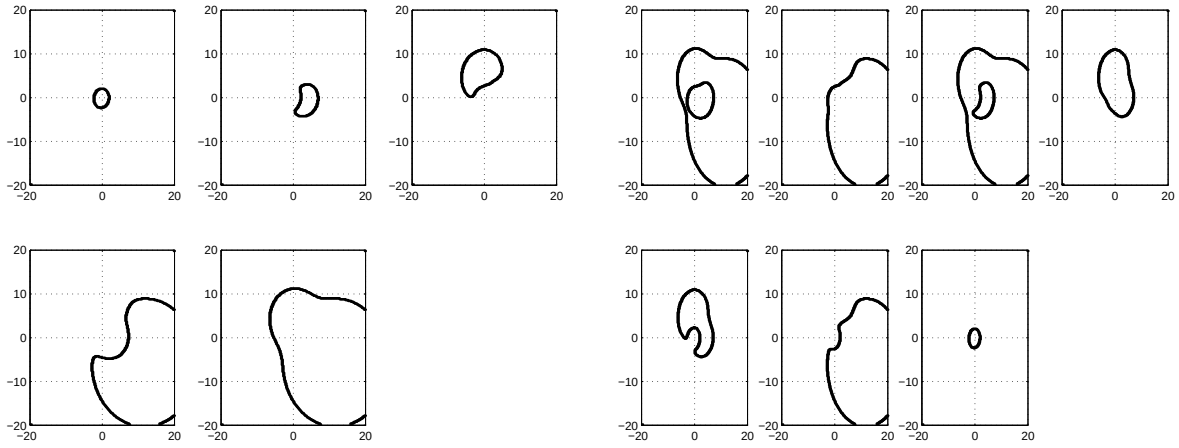


Figure 1: Optimum decision boundary for five class artificial data

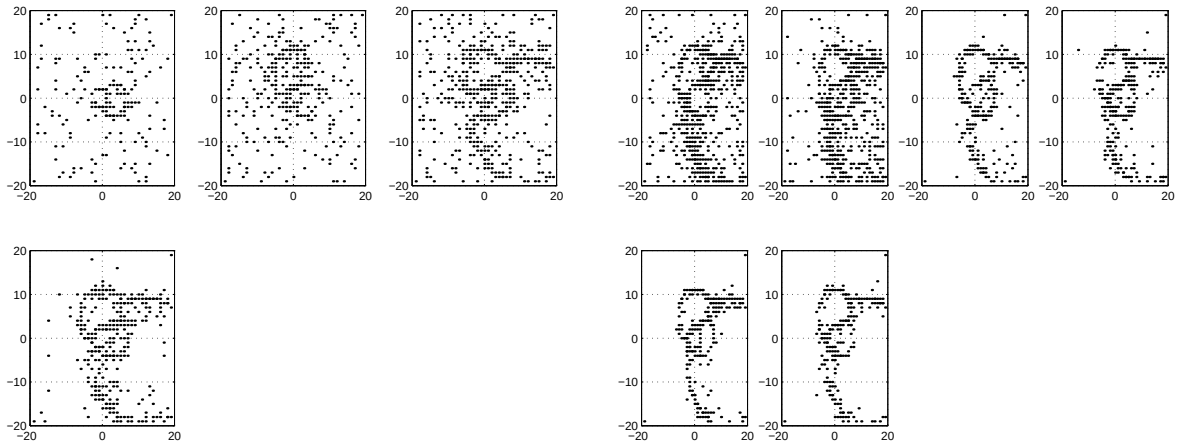
class	w_1	w_2	w_3	w_4	w_5
μ_i(mean)	[0,0]	[3,0]	[0,5]	[7,0]	[0,9]
σ_i^2(variance)	1	4	9	25	64

Table 1: Distribution parameters of data used in artificial benchmark



(a) (b)

Figure 2: Individual decision boundaries made by several base classifiers (a) C1 Code 5 classifiers (b) C4 code 7 classifiers



(a) (b)

Figure 3: Decision boundaries of ensemble as number of base classifiers is increased in presence of noise ($\mu_n = 0$, $\sigma_n^2 = 0.25$) (a) C1 code 2,3,4,5 classifiers (b) C4 code 2,3,4,5,6,7 classifiers (numbered left to right)

μ_n	σ_n^2	$C1*$	$C2$	$C3$	$C4*$	$C5$	$C6*$	$C7*$
0	0	100.0	96.11	98.93	100.0	100.0	97.79	100.0
0.5	0	100.0	53.17	74.04	100.0	98.09	100.0	100.0
0	0.2	88.59	89.40	91.71	93.22	95.49	94.44	96.67
0.5	0.2	87.25	57.19	65.98	91.54	84.64	94.69	96.34
0	0.4	65.96	75.99	76.51	78.71	88.38	86.14	92.92
0.5	0.4	62.83	56.71	59.50	76.16	76.46	85.73	91.72
0	0.6	51.32	56.26	60.16	62.58	75.96	71.81	86.13
0.5	0.6	46.91	49.52	51.19	59.98	67.18	72.32	83.72
0	0.8	42.68	48.87	49.07	52.13	63.21	60.52	76.86
0.5	0.8	39.79	44.80	45.36	50.29	57.42	61.73	74.47

Table 2: % match with Bayes rate for different codes (* equidistant) with Bayesian classifier in the presence of noise of specified mean (μ_n) and variance (σ_n^2)

Code	L1	Cen	LS	IHD	Lin	DS
C1*	100.00	98.46	100.00	100.00	100.00	100.00
C2	97.56	97.32	100.00	100.00	92.76	97.56
C3	97.30	97.32	100.00	100.00	94.86	97.30
C4*	100.00	98.40	100.00	100.00	100.00	100.00
C5	97.08	97.00	100.00	100.00	94.08	97.16
C6*	100.00	98.74	100.00	100.00	100.00	99.98
C7*	100.00	98.50	100.00	100.00	100.00	99.98

Table 3: % match with Bayes rate of various decoding/coding (* equidistant) strategies for artificial data and no added noise

Code	L1	Cen	LS	IHD	Lin	DS
C1*	100.00	58.96	100.00	100.00	100.00	100.00
C2	53.88	49.08	86.88	63.90	74.68	57.66
C3	83.06	49.74	31.52	51.30	97.24	96.14
C4*	98.64	76.72	98.64	98.64	98.64	94.40
C5	87.50	73.26	88.30	75.22	95.24	92.68
C6*	100.00	86.08	100.00	100.00	100.00	91.82
C7*	100.00	80.38	100.00	100.00	100.00	91.82

Table 4: % match with Bayes rate of various decoding/coding (* equidistant) strategies for artificial data and added noise ($\mu_n = 0.5, \sigma_n^2 = 0$)

Code	L1	Cen	LS	IHD	Lin	DS
C1*	57.66	52.10	57.74	57.48	57.74	56.52
C2	65.74	62.86	59.98	62.72	56.60	59.30
C3	66.40	62.66	60.30	61.12	64.90	61.74
C4*	69.04	66.46	69.04	69.02	69.04	63.02
C5	78.72	78.24	76.02	76.78	76.66	60.92
C6*	82.78	80.80	82.78	82.78	82.78	58.90
C7*	89.50	88.38	89.50	89.50	89.50	43.58

Table 5: % match with Bayes rate of various decoding/coding (* equidistant) strategies for artificial data and added noise ($\mu_n = 0, \sigma_n^2 = 0.5$)

Database	#Class	#Train	#Test	#Nodes
zoo	7	50	51	1
car	4	50	1678	1
vehicle	4	350	496	5
glass	6	100	114	2
satellite	6	1000	5435	2

Table 6: Specification of problems, showing number of problems, number of train and test patterns, and number of MLP hidden nodes

Code	L1	Cen	LS	IHD	Lin	DS
C1*	65.05	65.05	65.05	65.05	65.05	65.05
	17.29	17.29	17.29	17.29	17.29	17.29
C2	80.29	80.29	48.73	23.91	41.67	79.85
	6.915	6.915	4.88	2.30	7.77	1.27
C3	70.06	70.06	68.94	62.67	65.48	73.39
	10.42	10.42	10.92	6.96	10.71	8.08
C4*	69.48	69.48	69.48	69.48	69.48	72.33
	3.88	3.88	3.88	3.88	3.88	4.01
C5	77.74	77.74	77.56	77.74	76.94	78.17
	4.31	4.31	5.04	4.98	2.08	3.98
C6*	80.43	80.43	80.43	80.43	80.43	80.64
	1.73	1.73	1.73	1.73	1.73	1.67

Table 7: Mean and standard deviation of classification rate on 10 independent runs for various decoding/coding (* equidistant) strategies with “satellite” data

Code	L1	Cen	LS	IHD	Lin	DS
C1*	0.91	0.91	0.91	0.91	0.91	0.91
C2	0.89	0.89	0.83	0.61	0.58	0.90
C3	0.84	0.84	0.84	0.83	0.82	0.84
C4*	0.92	0.92	0.92	0.92	0.92	0.92
C5	0.97	0.97	0.97	0.97	0.94	0.97
C6*	0.97	0.97	0.97	0.97	0.97	0.99

Table 8: Mean value over five databases of ratio of classification rate of specified coding/decoding strategy to maximum classification rate over all decoding/coding (* equidistant) strategies for a particular database