Binary labelling and decision-level fusion

Terry Windeatt, Reza Ghaderi

Univ. of Surrey, Guildford, Surrey GU2 5XH

Address for correspondence

Centre for Vision, Speech and Signal Processing

School of Elec. Eng., IT and Maths

University of Surrey,

Guildford, Surrey, GU2 5XH

United Kingdom

Email: t.windeatt@surrey.ac.uk

Fax: +44(0) 1483 259554

Phone: +44(0) 1483 259286

for submission to Information Fusion

Word2000 on PC

Binary labelling and decision-level fusion

Abstract

Two binary labelling techniques for decision-level fusion are considered for reducing correlation in the context of multiple classifier systems. First, we describe a method based on error correcting coding that uses binary code words to decompose a multi-class problem into a set of complementary two-class problems. We look at the conditions necessary for reduction of error, and introduce a modified version that is less sensitive to code word selection. Second, we describe a partitioning method for two-class problems that transforms each training pattern into a vertex of the binary hypercube. A constructive algorithm for binary-to-binary mappings identifies a set of inconsistently classified patterns, random subsets of which are used to perturb base classifier training sets. Experimental results on artificial and real data, using a combination of simple neural network classifiers, demonstrate improvement in performance for these techniques, the first suitable for k-class problems, k > 2 and the second for k = 2.

Keywords: decision-level fusion, multiple classifiers, partitioning, error-correcting, bagging, boosting

1 Introduction

Various fusion strategies and associated architectures have been reported for the identification stage of data fusion systems. In [1], fusion strategies are characterised as data-level, feature-level and decision-level, and it is customary to divide decision-level strategies into soft-level and hard-level. By hard-level we mean that the combining mechanism operates on single-hypothesis decisions, in contrast with soft-level which implies a measure of confidence associated with the decision. Designing optimal strategies for decision-level fusion has been of interest to researchers in the fields of pattern recognition, machine learning, neural networks and more recently in data mining, knowledge discovery and data fusion. If the strategy is based on learning from a set of training patterns whose category label is known, it is referred to as a supervised learning problem. The classification task is to assign a test pattern, not previously used in training, to one of several possible classes.

What makes the classification problem challenging is that learning predictive relationships from data is in general ill-posed [2], which means that certain mathematical properties associated with the mapping (uniqueness, continuity, existence) may be violated. For a practical application, this implies that a learning machine must be designed to appropriately fit the training data; otherwise it will not perform well on a separate set of test data. For classification problems, there are many methods for building assumptions into the machine to enable it to generalise well. One approach, which has become popular across many disciplines, is based upon the combination of multiple classifiers, also referred to as an ensemble, committee or expert fusion. Various combining frameworks have been proposed but most systems use similar processing units as individual (base) classifiers. However, if the results of all the base classifiers are too well correlated there is little gain in combining, as discussed in [3].

It is therefore desirable to reduce correlation between classifiers in order to obtain optimal performance. In practice, it may not be feasible to obtain additional training samples, in which case an alternative is to use existing patterns but arrange it so that each classifier sees a slightly different problem. Some design methods aim to reduce correlation by, for example, using different base classifier parameters, feature sets or training sets [4]. A popular approach is to perturb the training set as in Bagging [5] and Boosting [6]. These methods appear to work well for unstable classifiers, such as neural networks and decision trees, in which a small perturbation in the training set may lead to a significant change in constructed classifier. Bagging and the original form of Boosting are voting algorithms, that is they use a combining strategy based on hard-level classifications. Bagging (from Bootstrap Aggregating) forms replicate training sets by sampling with replacement, and combines the resultant classifications with a majority vote. Boosting, which combines with a weighted vote is more complex than Bagging in that the distribution of the training set is adaptively changed based upon the performance of sequentially constructed classifiers. Training set perturbation methods were generally developed with classification trees as base classifiers, and do not necessarily improve performance with neural network base classifiers which often come with their own built-in perturbation methods (such as random weight initialisation). It also needs to be remembered that even if the chosen perturbation method successfully de-correlates the classifiers there is still the need to design a suitable combining strategy.

We distinguish two main types of correlation reduction methods in a multiple classifier framework. The perturbation methods discussed above are from the first category. They are distinguished by the fact that each base classifier handles the same problem in the sense that the class labelling is identical. In the second category are correlation reduction techniques in which each classifier is trying to solve a different sub-problem of the original problem. In a method like ECOC (Error-Correcting Output Coding) [10] each sub-problem uses a different class labelling. Techniques like binary decision clustering [7], pairwise coupling [8] may also be considered in this category.

In this paper we present two techniques for correlation reduction. The first, described in Section 2, is aimed at multi-class problems and is a modified version of ECOC. In Section 3, we present a two-class constructive technique applied to multiple classifier feature space for partitioning and perturbing the training set. Both techniques make no assumption about underlying probability distributions, and are particularly suitable if there is little prior information and possibly different types of classifier outputs. In Section 4 we use real benchmark data from [24], which contains problems from a variety of domains including medical diagnosis, prediction, image segmentation, various taxonomies. In fact any task that can be formulated as a supervised learning problem is a potential beneficiary of the proposed techniques. In Section 4.1 we also consider a more complex problem with a large feature set.

2 Correlation reduction by error-correcting coding

There are several motivations for decomposing a multi-class problem into complementary two-class problems. Such a decomposition means that attention can be focused on developing an effective technique for the two-class classifier, without having to consider explicitly the design and automation of the multi-class case. This is useful since some accurate and efficient two-class classifiers do not naturally scale up to multi-class. Also, it is hoped that the parameters of a simple parallel machine run several times may be easier to determine than a complex machine run once, and perhaps facilitate faster and more efficient solutions. Finally, as we will demonstrate, solving different 2-class sub-problems, perhaps repeatedly with random perturbation, may help to reduce error in the original problem.

The ECOC method, described in Section 2.1, is an example of distributed output coding [9], in which a test pattern is assigned to the class that is closest to a corresponding code word. Rows of the ECOC matrix act as the code words, and are designed using error-correcting principles to provide some error insensitivity with respect to individual classification errors. The motivation for encoding multiple classifiers using an error-correcting code is based on the idea of modelling the prediction task as a communication problem, in which class information is transmitted over a channel [10]. Errors introduced into the process arise from various stages of the learning algorithm, including features selected and finite training sample. For a 2-class problem, we assume that propagated errors can be of two kinds, either predicted class1 for target class2 or predicted class2 for target class1. From error-correcting theory, we know that a matrix designed to have *d* bits error-correcting capability implies that there is 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 fewer than *d* bits in error, by assigning the pattern to the code word closest in Hamming distance. Clearly, from this perspective it is desirable to use a matrix containing code words having high minimum Hamming distance between any pair. The ability to detect and possibly correct errors is dependent on the assumption that each error is independently produced. While in practice some errors will be correlated, the experimental evidence reported in [10] is that application of ECOC principles does lead to reduced error.

2.1 Summary of the original ECOC algorithm

To solve a multi-class problem in the ECOC framework we need a set of codes to decompose the original problem, a suitable two-class base classifier, and a decision-making framework. For a *k*-class problem $\{\omega_1, _, \ldots , \omega_{k},\}$, each row of the *k x b* binary ECOC matrix acts as a code word or label for each class. Each of the b columns of the matrix partitions the training data into two super-classes according to the value of the corresponding binary element (e.g. 1 or 0). A test pattern is presented to the b trained base classifiers giving a soft-level b-dimensional binary vector, and combining is based on identifying the closest code word.

Summary of Training: *for i = 1 : b*

- re-label training patterns into two classes (super-classes) according to binary element corresponding to each class for column *i*
- train a 2-class base classifier using the re-labelled training set

Summary of Testing:

- apply pattern to the b trained base classifiers forming vector $[\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_b}]^T$ where y_j is the output of the *jth* base classifier
- compute distance between output vector and code word for each class

$$L_i = \sum_{j=1}^{b} \left| Z_{ij} - y_j \right| \qquad (1)$$

- assign pattern according to class ω_m corresponding to closest code word

$$m = \underset{i}{ArgMin}(L_i) \qquad (2)$$

2.2 Analysis of ECOC

Assuming that our goal is to minimise classification error rate, we can identify three different kinds of errors:

- Bayesian error, which is the result of overlapped classes, and cannot be removed by combining, error-correcting or any other technique
- Base classifier error, which is derived from a mismatch between the base classifier and the particular re-labelled training set, and may be partially removed by error-correcting capability and by repeating parts of the two-class sub-problems
- Combining error, which comes from imperfect combining strategy and is present even if the base classifiers are perfect i.e. Bayesian.

To understand the concept of combining error, consider the relationship between the combining rule used by ECOC and the Bayesian decision rule. Let the posterior probabilities of each class be given by $[q_1, q_2, ..., q_k]^T$. According to Bayes rule, a pattern is assigned to class ω_m given by

$$m = \underset{\mathbf{i}}{ArgMax}\,(q_i) \qquad (3)$$

Each base classifier is trained on a re-labelled training set, but let us assume that base classifiers provide posterior probability of super-class membership. The matrix equation can be expressed as

$$[\mathbf{y_1, y_2, ..., y_b}]^T = \mathbf{Z}^T \bullet [\mathbf{q_1, q_2, ..., q_k}]^T$$

so that

$$y_j = \sum_{l=1}^{k} q_l Z_{lj} \qquad l = 1,2, \dots k \qquad (4)$$

From equations 1 and 4

$$L_i = \sum_{j=1}^{b} \left| \left(\sum_{l=1}^{k} q_l Z_{lj} \right) - Z_{ij} \right| \qquad (5)$$

From equation 5, separating the case *l* = *i*, and using the fact that $\sum_{l=1}^{k} q_l = 1$

$$L_i = \sum_{j=1}^{b} \left| q_i Z_{ij} - Z_{ij} + \sum_{l \neq i} q_l Z_{lj} \right|$$

$$= \sum_{j=1}^{b} \left| \sum_{l \neq i} q_l (Z_{ij} - Z_{lj}) \right|$$

$$= (1 - q_i) \sum_{j=1}^{b} \left| Z_{ij} - Z_{lj} \right| \qquad (6)$$

But $\sum_{j=1}^{b} \left| Z_{ij} - Z_{lj} \right|$ is just the Hamming distance between *lth* and *ith* rows

Equation 6 tells us that L_i is a function of the posterior probability and Hamming distance of *ith* row to all other rows. Therefore, when all pairs of code words have equal distance between them, equation 2 and 3 represent the same decision rule

$$\underset{i}{\mathbf{ArgMax}}\,(\mathbf{q_i}) \quad = \quad \underset{i}{\mathbf{ArgMin}}\,(\mathbf{L_i})$$

or in other words ECOC uses Bayes decision rule

2.3 ECOC matrix design

The main constraints in designing ECOC matrices are as follows:

1. distance between rows, which determines error-correcting capability
2. repetition of different parts of sub-problems defined by b base classifiers i.e. length of code words, which affects error variance
3. variation of distance between code words, which changes the effectiveness of the decision making strategy
4. distance between columns, which determines independence of base classifiers

To satisfy the first three constraints we need to find a matrix with long code words and having maximum and equal distance between all pairs, but we know from the theory of error-correcting codes that this is complex. Reference [11] explains the complexities involved and describes several techniques for designing such matrices including BCH codes, which we employ in Section 4. Additionally we need to satisfy the fourth constraint, which requires that Hamming Distance between pairs of columns should also be maximum [10] and which therefore makes the problem even more complex. However, by making a few assumptions we can analyse the expected error-correcting capability as length of code word increases, and justify the practice of using random codes. Consider an imaginary k-class problem in which code matrix Z has guaranteed distance *2d+1* between any pair of code words, or in other words Z has *d* bits error correction capability. For simplicity also assume that all classes have the same overlap with each other, all the rows of Z are equi-distant and base classifier errors are equal and independently produced. We can find a formula [25] giving the probability of assigning test pattern to class ω_j when it correctly belongs to class ω_i, which shows that error reduction capability asymptotes at approximately 20, 30, 40 bits for error probability of 0.25, 0.5, 0.75 respectively. This leads us to expect that longer random codes might perform almost as well as optimal codes, since Hamming Distance between pairs of random code words will tend to increase on average as length is increased. In Section 4, results are presented for several different optimal and random codes, demonstrating that the performance difference between them is less significant as length of code word increases.

2.4 Circular ECOC

There is however another issue, beyond those mentioned in Section 2.3, that may affect performance and which is not addressed by random codes. Since in practice the overlap between classes can be different, we might suspect that certain code words are suited to some super-classes more than others. Although it would seem a good idea to use a labelling in which distance between labels is based on distance between classes (e.g. Mahalonobis distance), the additional constraint is likely to be difficult to implement. Taking a different approach that we call Circular ECOC, we suggest the following algorithm in which all code words are used for each class to reduce sensitivity to code word selection. Design considerations for the *k x b* ECOC matrix are the same for both methods, and as before BCH or random codes are suitable.

Summary of Training:

```
For n = 1...k
    - Circularly shift one row of ECOC matrix
    - Train b base classifiers on shifted ECOC
    - For each test pattern, compute the distance $L_{ni}$ for class $\omega_i$ between b-dimensional base classifier outputs and code word corresponding to class $\omega_i$
```

Summary of Testing:

Assign pattern to class ω_m by averaging the final distance measurement

$$\mathbf{m = \underset{i}{ArgMin}} \sum_{n=1}^{k} L_{ni} \qquad (7)$$

3 Correlation reduction by constructive partitioning

The second correlation reduction technique that we discuss is aimed at multiple classifier systems for 2-class problems. The assumption is that a hard-level decision is taken in intermediate multiple classifier feature space, and the partitioning method is based on applying a constructive approach to the mapping defined by 2-class target vector with respect to individual classifier decisions.

3.1 Constructive algorithms

Constructive methods applied to binary-to-binary mappings were popular over a decade ago for implementing TLU (Threshold Logic Unit) networks, and a summary of these algorithms can be found in [12]. The idea is to grow a network a node at a time as needed, thus overcoming the requirement of knowing the network architecture in advance. These constructive procedures are built around the concept of single perceptron training, and the partitioning step can be characterised as finding a halfspace consistent with all patterns from one class and a maximal subset of the other class [13]. Since finding the maximum subset is NP-hard, an approximation algorithm is required. One such method is the pocket variant of classical perceptron algorithm [14], which updates weights only after obtaining a longer run of correct classifications. Although it can handle noisy, incomplete and contradictory data and is guaranteed to converge, the pocket algorithm has no upper bound on number of iterations. Alternate methods of partitioning used in constructive algorithms include: (i) quality function to achieve faithful layers [15] (ii) thermal perceptron that gradually reduces the size of the sensitive zone around a hyperplane [16] (iii) linear programming approach to include negative examples in the separable subset [13] (iv) necessary checks for separability from threshold logic theory [17] and (v) topological strategy for visiting vertices of the binary hypercube [18].

In contrast to previous methods, we apply the constructive approach to multiple classifier feature space rather than to binary-valued raw data. For this purpose we select a modified version of the Sequential Learning model of recruiting neurons, described in [19], which builds hidden units by a partitioning method that sequentially excludes clusters of patterns of the same target. The original goal of Sequential Learning was to regularly partition the input hypercube so that (i) vertices not separated by a hyperplane are of the same classification, and (ii) each hyperplane separates vertices of different classification. We use the irregular partitioning version that can choose either of the two classes, and in our implementation we incorporate a check for separability from threshold logic theory as described in Section 3.2 and 3.3.

3.2 Check for separability

By assigning one of two classes to each base classifier and repeating *b* times, each training pattern may be regarded as a vertex in the *b*-dimensional binary hypercube:

$$X_m = (x_{m1}, x_{m2}, \dots, x_{mb}) \qquad x_{mj} \text{ and } f(X_m) \in \{0,1\} \qquad \textbf{(8)}$$

For a particular pattern, we want to determine its contribution to separability by considering collections of vertices, and a measure of correlation (σ) is assigned to each binary component x_j according to the following rule:

$$\begin{array}{ll} & \text{For all } \boldsymbol{X_1}, X_2 \text{ such that } f(X_1) \neq f(X_2) \qquad \textbf{(9)} \\ & \text{Assign } |\sigma_j| = |x_{1j} - x_{2j}| |X_1 \oplus X_2|^{-1} \\ \text{where} & \sigma_j \text{ is excitatory } = \sigma_j^{\,+} \quad \text{if } x_{1j} = f(X_1) \\ & \sigma_j \text{ is inhibitory} = \sigma_j^{\,-} \quad \text{if } x_{1j} \neq f(X_1) \end{array}$$

Equation 9 states that $|\sigma_j|$ is inversely proportional to Hamming Distance, with $\sigma_j^{\,+}$ *and* $\sigma_j^{\,-}$ indicating excitatory and inhibitory contributions. For a completely specified Boolean function (all 2^b rows of the truth table), this rule reduces to the first stage of logic minimisation [20] when only nearest neighbours (unit Hamming Distance) are compared. Furthermore summing $\sigma_j^{\,+}$ *and* $\sigma_j^{\,-}$ over all patterns is identical to spectral summation [21], with $\sum_X \sigma_j^{\,+}$ and $\sum_X \sigma_j^{\,-}$ giving the first order spectral coefficients decomposed into excitatory and inhibitory contributions [22]. The existence of $\sum_X \sigma_j^{\,+} > 0$ and $\sum_X \sigma_j^{\,-} > 0$ provides evidence that the set of patterns is not *1*-monotonic in the *j*th component and therefore non-separable. A discussion of *k*-monotonicity as necessary and increasingly sufficient conditions for separability is given in [21] [23].

3.3 Extracting Subsets

After applying the rule defined by 9, each pattern component x_j has associated correlation measures σ_j^+ *and* σ_j^-, representing positive and negative correlation. In order to define a pattern's contribution to separability we look at the relative contribution with respect to $\sum_X \sigma_j^+$ and $\sum_X \sigma_j^-$ and define the total correlation as sum of the relative contributions.

$$\sigma_{TOT} \quad = \quad \sum_{j=1}^{b}\left[signum(\sum_X \sigma_j^+ - \sum_X \sigma_j^-)\left(\frac{\sigma_j^+}{\sum_X \sigma_j^+} - \frac{\sigma_j^-}{\sum_X \sigma_j^-} \right)\right] \qquad \mathbf{(10)}$$

In equation (10) the sign of the *j*th contribution is based on the larger of $\sum_X \sigma_j^+$ and $\sum_X \sigma_j^-$.

Patterns are then separated into two subsets, depending on whether σ_{TOT} is positive or negative. The larger subset is extracted and the partitioning repeated for the remaining patterns [28]. For experiments reported in Section 4.2, four separable subsets (two class 1 and two class 2) are extracted and the remaining patterns are called the inconsistently classified set (*ICS*). The first two class 1 (or class 2) extracted subsets contain unambiguously correctly and incorrectly classified patterns respectively, and we were able to observe this for the two-dimensional artificial data in Section 4.2.

Patterns in the *ICS* are split into approximately κ equal subsets, each subset being left out of a base classifier training set to obtain the ICS estimate for the next recursion:

ICS(1) = *ICS* estimate after one recursion using empty *ICS* (i.e. no patterns left out)

ICS(m) = *ICS* estimate after one recursion using *ICS(m-1), m* is recursion number

4 Experimental Results

In order to understand the partitioning performed by correlation reduction techniques it is desirable to visualise different parts of input space. While average test error rate is usually what interests designers, it is not particularly helpful in evaluating different designs. Artificial data has the further advantage that we can calculate the best possible error rate i.e. Bayes rate. On the other hand artificial data is of necessity low dimension since we want to visualise it, and is often not very representative of real high dimensional data, so we also include tests on real benchmark data from [24] [29].

4.1 Original and Circular ECOC

An artificial five-class overlapping Gaussian problem is defined by five groups of two-dimensional vectors having normal distribution, as shown in Table 1. Base classifiers are not trained, but using the parameters from Table 1, we find the posterior probability of super-class membership. Increasing levels of (zero mean) Gaussian noise are added to the soft-level (real-valued) output of base classifiers to simulate noisy imperfect classifiers. Table 3 gives percentage match with Bayes rate for all *k x b* code matrices defined in Table 2, and figure 2 and figure 3 plot results for Z2, Z4, Z5, Z6. Comparison of figure 3 with figure 2 demonstrates that the circular method is more robust as noise is increased. The following points can be observed for both original and circular ECOC methods:

- code matrices of similar length perform better if error-correction capability is increased
- longer codes perform better even if error-correction capability is reduced
- beneficial effect of optimal compared with random codes has been reduced as length of code words increases (as predicted Section 2.4)

By visualising decision boundaries [25] we were also able to observe that

- each individual classifier concentrates on different part of the input space, and therefore has a less complex problem to handle.
- same part of a decision boundary appears in more than one classifier
- to achieve robust performance more classifiers are required as noise is increased, as shown in figure 3
- repeatedly learning the same part of the decision boundary leads to reduction in error variance

.

.

For experiments on real benchmark data from [24], a single hidden layer MLP trained by Levenberg-Marquardt algorithm is used as base classifier with fixed number of epochs. The number of hidden nodes is specified in Table 4, which also shows the size of the random ECOC code matrix used in the experiments. Table 4 gives percentage improvement for mean and std of Circular over Original ECOC, for ten independent runs repeated over random training/testing splits with the number of training samples specified. The average improvement shown in the final two columns in Table 4 is 5% and 30% mean and std respectively. Although both algorithms are effective, Circular ECOC generally exhibits higher classification rate with lower variance. In [25] we also showed that these results compared favourably with those obtained for a single multi-output MLP.

To demonstrate the effectiveness of circular ECOC approach we applied it to a more complex problem with a large feature set - the Olivetti Research face recognition database [26]. It consists of 400 images, 10 each of 40 different subjects, shot at different times and with different lighting conditions, and manually cropped to resolution 92 x 112 8-bit gray-levels. . The only pre-processing we performed was a resampling based on nearest neighbour interpolation to reduce the size of the images. To further reduce complexity, we divided each image into four (non-overlapping) approximately equal vertical stripes, the gray-level values from each stripe being input directly as features to one of four single-hidden layer feedforward sub-networks. The outputs of the four sub-networks are combined using a single node, so that the four sub-networks plus output node form a binary classifier which can be trained by Back-Propagation. Five images for each subject were randomly selected for training and the remaining five used for testing, giving a random 50/50 training/testing split and repeated ten times. Figure 4 shows test error rate for 12 x 12 and 32 x 32 images, using ECOC with a binary classifier containing four 25 hidden-node feedforward sub-networks. The result of boosting the binary classifier with ten iterations of Adaboost can also be seen in figure 4. As number of ECOC columns is increased, the difference in performance due to boosting is reduced, demonstrating a degree of insensitivity to base classifier accuracy. The final classification rate for 32 x 32 images in figure 4 is comparable to that found in [26]. For comparison with a single multi-output network, we modified the binary classifier by adding 40 output nodes, one for each of 40 classes. By trial and error variation of number of hidden nodes and learning parameters, the best that we could achieve with Back-Propagation applied to the single multi-class network was 31% (free guess 4%). Of course, we are unable to say that no single network exists with an architecture that outperforms the final error rate shown in figure 4, but finding the parameters would be difficult.

4.2 Constructive partitioning

An artificial two-class problem from [27] is employed (mean (0,0), variance 1 and mean (2,0), variance 4) with 400 training & 30,000 test patterns. The base classifier is a 3-hidden node MLP trained by Levenberg-Marquardt algorithm with fixed number (50) epochs, and the base classifier is run fifty times (b = 50) with random initial weights. The Bayes boundary is circular for this problem with Bayes error rate of 18.49%.

Fifty independent *ICS(1)* estimates are summed and normalised, to give a value between 0 and 1 that represents the confidence that a particular pattern belongs to *ICS(1)*. By varying a confidence threshold, τ we can change the number of patterns in *ICS(1)*. Figure 5 shows patterns in *ICS(1)* clustering around the Bayes boundary when $\tau = 0.9$. The second estimate *ICS(2)* is recursively made by repeating *50x50* runs, but leaving out a random fraction $1/\kappa$ of *ICS(1)* from each base classifier, with the patterns in *ICS(1)* defined by τ. Figure 6 shows the combined (majority vote) decision boundary for the second recursive estimate m = 2, $\kappa = 2$, $\tau = 0.7$. It is clear that the Bayes boundary is closely approximated except in the low-density region, and we were able to observe that the decision boundary is quite sensitive to the value of κ [28]. Figure 7 shows test error rate variation with τ for m = 2, $\kappa = 2$ when the experiment is repeated ten times for different random 400 pattern training set. Table 5 compares mean base classifier error rate and majority vote combination for the first two recursions. Since ICS(1) uses base classifiers operating on the full training set, improvement due to combining for m = 1 is the result of correlation reduction by random weight initialisation alone. The combined test error with respect to Bayes rate has been reduced from 0.91% (m=1) to 0.56 % (m = 2).

To determine sensitivity to base classifier complexity, we used this strategy with MLP networks trained by Levenberg-Marquardt algorithm on real benchmark data from [29] (Cancer, Diabetes). Each experiment was repeated ten times with random 50/50 training testing splits while number of hidden nodes *h* and number of epochs *nepochs* was systematically varied [30]. As *nepochs* is reduced, a single ICS estimate at one value is used to determine patterns left out at the next lower value. Figure 8 shows majority vote training and test error rates for Diabetes data at $\kappa = 2$ for three different values of h. Figure 9 shows error rates for Cancer data at h = 1 for three different values of κ. We see that the test error rate for $\kappa = \infty$ (no patterns left out) is more sensitive to the value of *nepochs* than $\kappa = 2$ *or* $\kappa = 3$. Leaving out ICS patterns from MLP base classifier training sets appears to decrease sensitivity of overfitting to number of training epochs.

Conclusion

We considered two examples of techniques based upon binary labelling for reducing correlation among individual classifiers for decision-level fusion in a multiple classifier framework. In the first technique, which is based on error-correcting coding, the existing ECOC method was analysed and modified to make it less sensitive to the binary label assigned to each class. In the second technique, which is based on a constructive algorithm with separability check, a set of inconsistently classified patterns was identified and used to perturb base classifier training sets. From these examples, we conclude that techniques based on binary labels may be usefully employed for reducing correlation and enhancing performance of two-class as well as multi-class supervised learning problems.

Class	class1	Class2	Class3	Class4	Class5
Mean	(0,0)	(3,0)	(0,5)	(7,0)	(0,9)
Variance	1	4	9	25	64

Table 1

Artificial five-class Gaussian problem defined by mean and variance for each class

ID	Type	equ?	b	e-c
Z1	identity	no	5	0
Z2	Random	no	7	1
Z3	BCH	no	7	3
Z4	BCH	yes	7	3
Z5	BCH	yes	15	7
Z6	Random	yes	15	2
Z7	BCH	no	31	7

Table 2
Code matrices (*5 x b*) showing number of base classifiers (b), error-correcting capability in bits (e-c), and whether code words are equi-distant (equ). Z1 is the one-per-class matrix.

noise	Z1	Z2	Z3	Z4	Z5	Z6	Z7
σ	Orig./Circ	Orig./Circ	Orig./Circ	Orig./Circ	Orig./Circ	Orig./Circ	Orig./Circ
0	100/100	96.1/99.2	98.9/99.7	100/100	100/100	97.8/99.4	100/100
0.2	88.6/96.0	89.4/96.8	91.7/97.1	93.2/97.1	95.5/97.8	94.4/97.6	96.7/98.4
0.4	66.0/89.8	76.0/93.9	76.5/93.6	78.7/93.9	88.4/96.0	86.1/95.6	93.0/97.0
0.6	51.3/78.7	56.3/85.1	60.2/87.0	62.6/89.1	76.0/92.7	71.8/92.2	86.1/95.3
0.8	42.7/68.1	48.9/79.7	49.1/78.3	52.1/80.8	63.2/89.0	60.5/88.0	76.9/93.4

Table 3
Rate of matching (%) with Bayes rate for original and circular ECOC with different code matrices, as level of Gaussian noise added to base classifiers is increased

database	# hidden nodes	#columns code matrix	#rows (classes)	#training samples	% improve Mean	% improve std
car	1	15	4	100	1.5	13
cmc	1	15	3	100	4	4
gauss5	2	15	5	200	1	10
iris	1	15	3	50	12	84
segment	1	50	7	200	8	50
vehicle	5	15	4	350	5	15
zoo	1	15	7	51	2.5	30

Table 4
Improvement (%) in mean and std of Circular over Original ECOC method for various datasets. Number of hidden nodes of MLP base classifier is shown, along with number of rows and columns of random code matrix.

	Base classifiers m = 1	Combined maj. vote m = 1	Base classifiers m = 2	Combined maj. vote m = 2
mean	21.98	19.4	22.6	19.05
std	0.9	0.06	0.58	0.06

Table 5
test error rates (%) for first two recursive estimates with τ=0.7, κ = 2

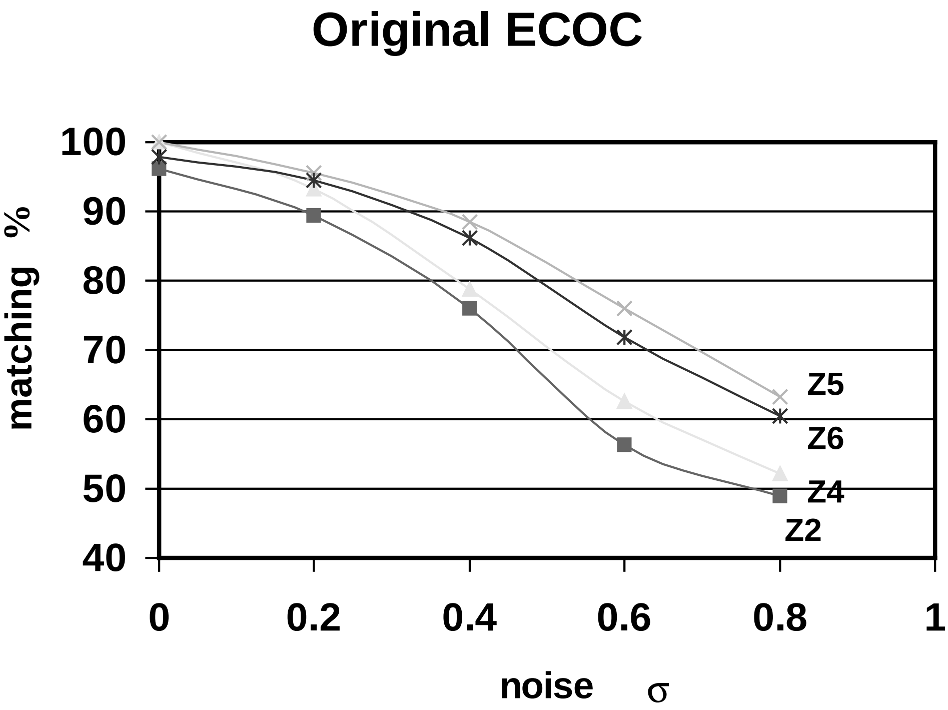

Original ECOC
100
90
80
70
60
50
40
matching %
0
0.2
0.4
0.6
0.8
1
noise σ
Z5
Z6
Z4
Z2

Figure 1

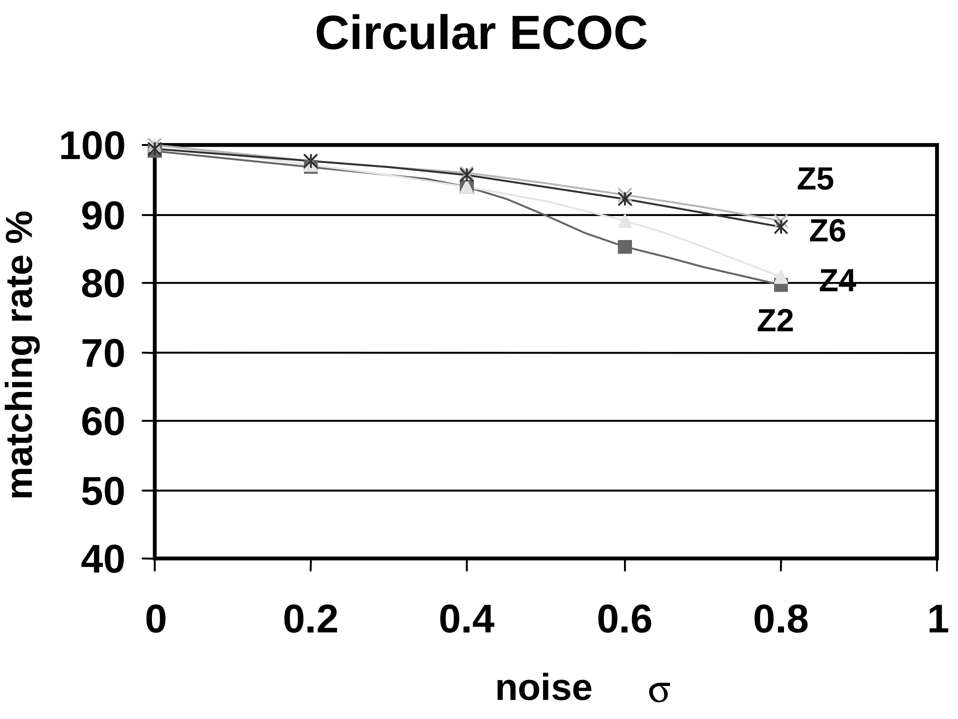
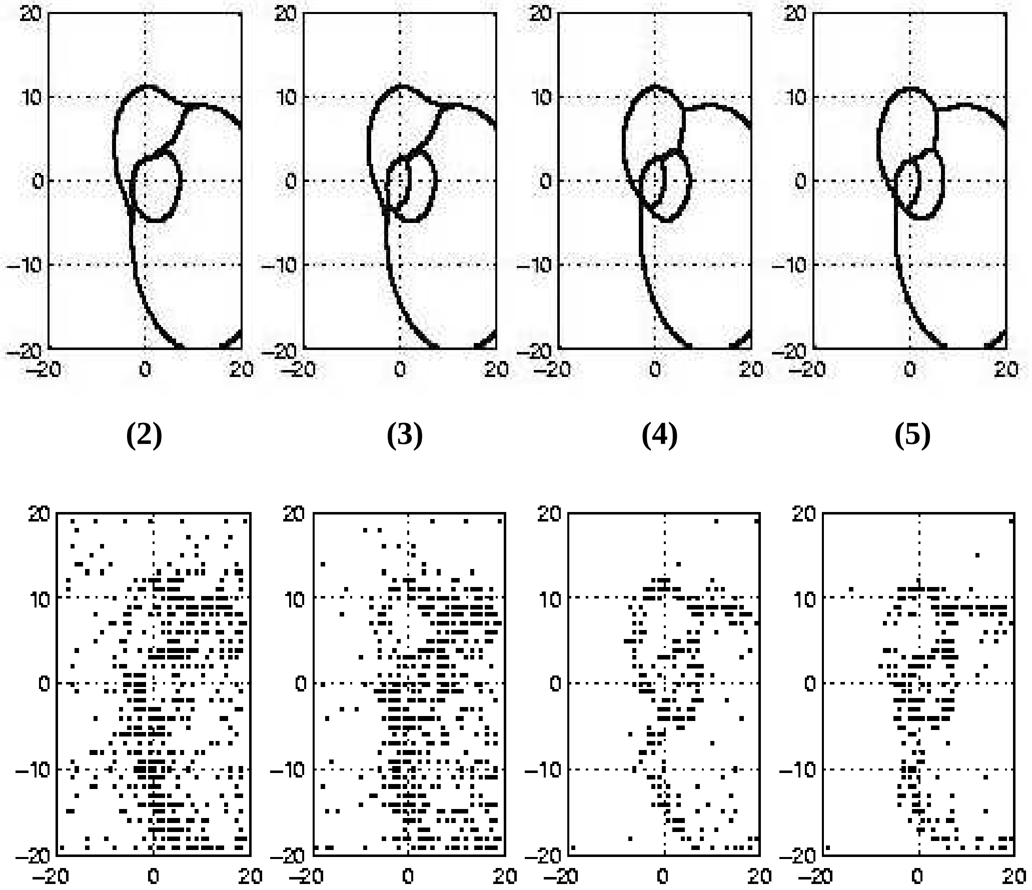

Circular ECOC
matching rate %
100
90
80
70
60
50
40
0
0.2
0.4
0.6
0.8
1
noise σ
Z5
Z6
Z4
Z2

Figure 2



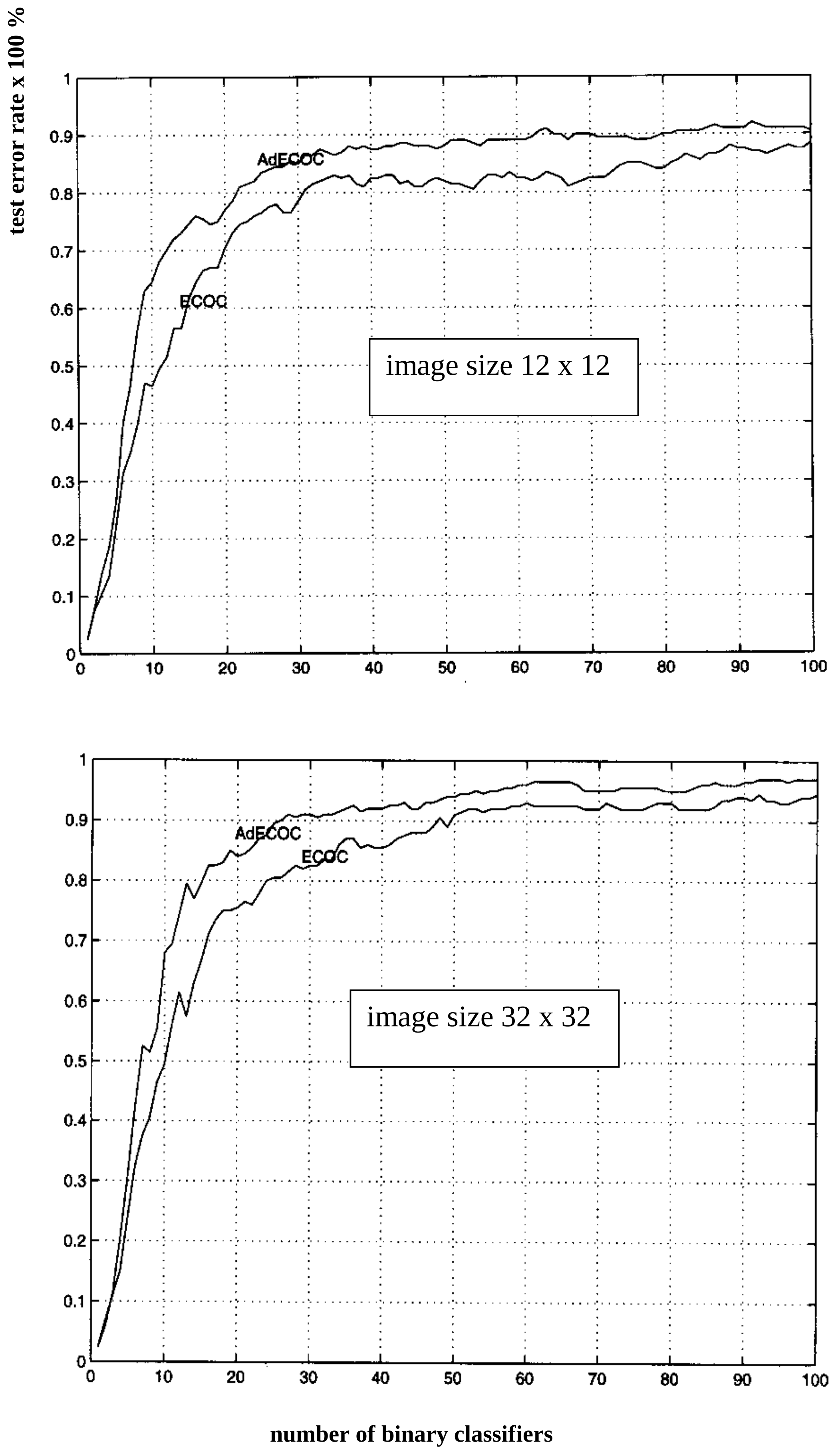
(2)

(3)

(4)

(5)

Figure 3:


test error rate x 100 %
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
10
20
30
40
50
60
70
80
90
100
AdECOC
ECOC
image size 12 x 12
AdECOC
ECOC
image size 32 x 32
number of binary classifiers

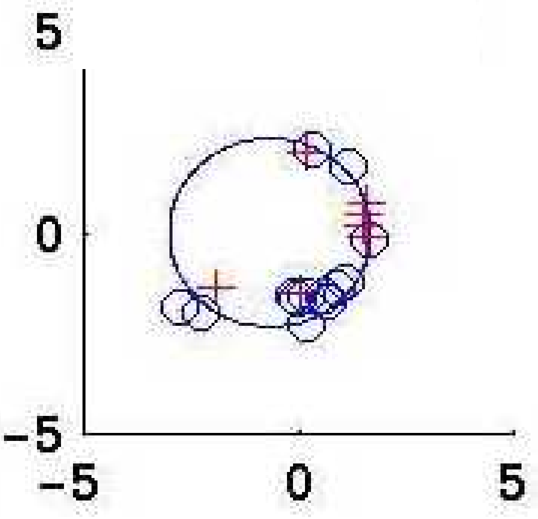


Figure 5

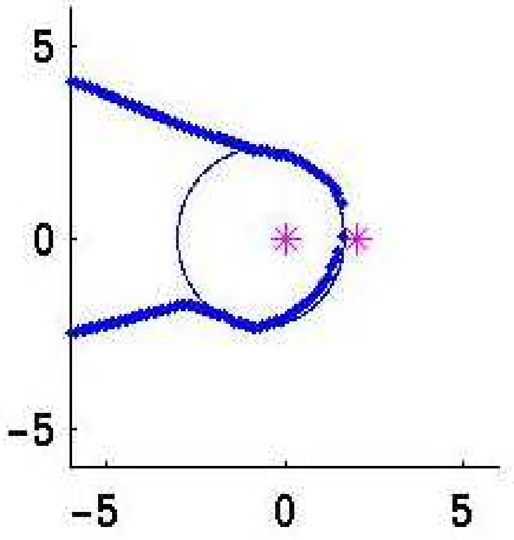


Figure 6

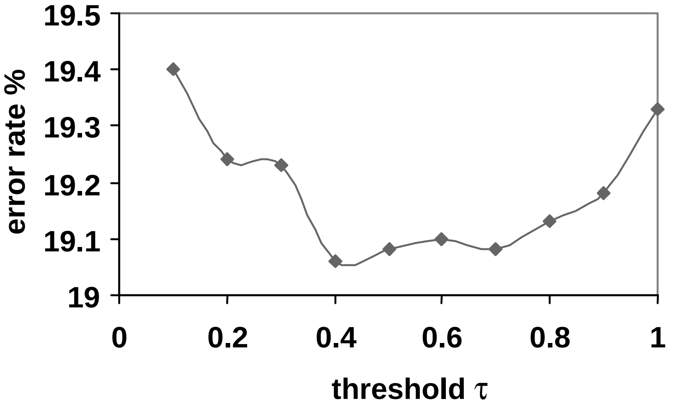


Figure 7

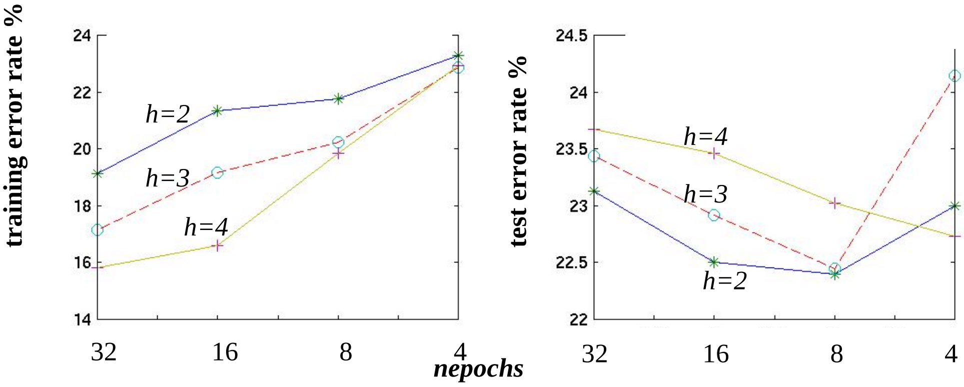

training error rate %
24
22
20
18
16
14
h=2
h=3
h=4
32
16
8
4
nepochs
test error rate %
24.5
24
23.5
23
22.5
22
h=4
h=3
h=2
32
16
8
4

Figure 8

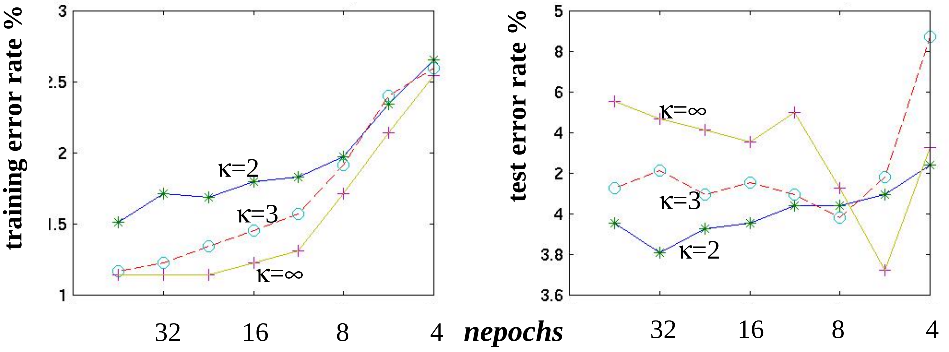

training error rate %
3
2.5
2
1.5
1
κ=2
κ=3
κ=∞
32
16
8
4
nepochs
test error rate %
5
8
6
4
2
4
3.8
3.6
κ=∞
κ=3
κ=2
32
16
8
4

Figure 9

.

Figure 1

Matching (%) with Bayes rate for increasing levels of Gaussian noise (std) and different code matrices for original ECOC method

Figure 2

Matching (%) with Bayes rate for increasing levels of Gaussian noise (std) and different code matrices for circular ECOC method

Figure 3

Decision boundary of composite classifier for code matrix Z4, as number of base classifiers is increased from 2 to 5, noise $\sigma = 0$ (top) and $\sigma = 0.25$ (bottom)

Figure 4

Performance of ECOC and ECOC + Adaboost (AdECOC) on face recognition database as number of columns of the ECOC matrix is increased

Figure 5

ICS(1) patterns (+ and o representing the two classes), shown with circular Bayes boundary when threshold $\tau = 0.9$

Figure 6

Combined (majority vote) decision boundary for $m = 2$, $\kappa = 2$, $\tau = 0.7$. (Gaussian centres marked *)

Figure 7

Combined (majority vote) test error rate variation with ICS(1) threshold τ , $\kappa = 2$, $m = 2$

Figure 8

Combined (majority vote) training and test error rates for Diabetes data, as number of epochs is reduced when $\kappa = 2$ and $h = 2,3, 4$

Figure 9

Combined (majority vote) training and test error rate for Cancer data, as number of epochs is reduced when $h = 1$ and $\kappa = 2,3, \infty$

1 D. L. Hall, Mathematical techniques in multisensor data fusion, Artech House, Norwood Mass, USA, 1992.

2 A. N. Tikhonov and V. A. Arsenin, Solutions of ill-posed problems, Winston & Sons, Washington, 1977

3 K. Tumer and J. Ghosh, Error correlation and error reduction in ensemble classifiers, Connection Science, Special Issue on Combining Artificial Neural Networks: Ensemble Approaches, 8(3-4), pp 385-404, 1996.

4 J. Kittler, A. Hojjatoleslami and T. Windeatt, Strategies for combining classifiers employing shared and distinct pattern representations, Pattern Recognition Letters, Vol.18(11-13), pp.1373-1377, 1997.

5 L. Breiman, Bagging Predictors, Machine Learning, 24(2), pp123-40, 1997.

6 Y. Freund and R.E. Schapire. A decision-theoretic generalisation of on-line learning and an application to boosting, J. of Computer and System Science, 55(1), pp119-139, 1997.

7 C. Wilson, P. Grother and C. Barnes, Binary decision clustering for neural network-based optical character recognition. Pattern Recognition 29(3), pp 425-437, 1996.

8 T. Hastie and R. Tibshirani, Classification by pairwise coupling, tech report 94305, Dept Statistics, Stanford Univ., 1996.

9 T.J. Sejnowski and C.R. Rosenberg, Parallel networks that learn to pronounce english text, Journal of Complex Systems, 1(1), pp 145-168, 1987.

10 T. G. Dietterich and G. Bakiri, Solving multiclass learning problems via error-correcting output codes, J. of Artificial Intelligence Research 2, pp263-286, 1995.

11 W Peterson and J. Weldon, Error-correcting codes, MIT Press, Cambridge, Mass, 1972.

12 Muselli, M. (1995). On sequential construction of binary neural networks, IEEE Trans. Neural Networks 6(3), 678-690.

13 Marchand, M. and M. Golea (1993). On learning simple neural concepts: from halfspace intersections to neural decision lists, Network Computation in Neural Systems 4 (1), 67-85.

14 S. I. Gallant, Perceptron-based learning algorithms, IEEE Trans. on Neural Networks 1(2), 1990, pp 179-192.

15 G. T Barkema, H. Andree and A. Taal, The patch algorithm: fast design of binary feedforward neural networks, Network Computation in Neural Systems 4 (3), 1993,393-407.

16 M. Frean, ‘Thermal’ perceptron learning rule, Neural Computation 4, 1992, 946-957.

17 S.A.J Keibek, G.T. Barkema, H.M.A. Andreee, M.H.F. Savenlie and A. Taal, A fast partitioning algorithm and a comparison of binary feedforward neural networks, Europhys. Lett., 18 (6), 1992, 555-559.

18 F. M Mascioli,. and G. Martinelli, A constructive algorithm for binary neural networks: The Oil-Spot Algorithm, IEEE Trans. Neural Networks. 6(3), 1995,794-797.

19 M Marchand, M. Golea and P. Rujan, A convergence theorem for sequential learning in two-layer perceptrons, Europhys. Lett. 11 (6), 1990, 487-492.

19 E. J. McCluskey, Minimisation of boolean functions, Bell Syst. Tech. J., Vol 35(5), pp1417-1444, 1956.

21 Hurst, S. L., Miller, D. M. & Muzio J. C. (1985). Spectral Techniques in Digital Logic, Academic Press.

22 T. Windeatt and R. Tebbs, Spectral technique for hidden layer neural network training, Pattern Recognition Letters, Vol.18(8) pp723-731, 1997.

22 S. Muroga, Threshold Logic & its Applications, Wiley, 1971.

24 C.J. Merz and P. M. Murphy, UCI repository of machine learning databases, 1998

25 T. Windeatt and R. Ghaderi, Binary codes for multi-class decision combining, Aerosense 2000, Sensor fusion:Architectures Algorithms & Applications, SPIE, Orlando, pp 23-34, 2000

26 F.S. Samaria and A.C. Harter, Parameterisation of a stochastic model for human face identification, In Proc. Second IEEE Workshop on application of computer vision, Sarasota, Florida, 1994. http://mambo.ucscs.edu/ps/olivetti.htm

27 P. Yee, Classification requirements involving Backpropagation and RBF networks, Tech Report 249, McMasters Univ, Ontario, 1992.

28 T. Windeatt, Recursive partitioning for combining multiple classifiers, Neural Processing Letters, 13(3) 2001.

29 L. Prechelt, Proben1, Tech Report 21/94, Univ. Karlsruhe, Germany, 1994.

30 T. Windeatt, Classifier instability and partitioning, Proc. of Int. Conf. on Multiple Classifier Systems, Jun 20-23, 2000, Cagliari, Italy, Springer-Verlag, to appear.