`

Multi-class learning and error-correcting code sensitivity

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 Electronic Letters

Multi-class learning and error-correcting code sensitivity

Abstract

Properties of optimal error-correcting codes for decomposing a multi-class problem into a set of two-class problems are discussed. After analysing the effect of the code matrix on individual classification errors and on errors due to combining, we propose a modification that is less sensitive to code word selection.

1 Introduction

The use of error-correcting codes for decomposing a multi-class learning problem into a set of complementary two-class problems, known as the ECOC (Error-correcting output coding) method, is well established [1]. Generally, methods that have been developed to provide such a decomposition aim at simplicity of implementation which is intended to give improved efficiency and accuracy [2]. Some effective two-class classifiers are not easily modified to deal directly with the multi-class case, and the ECOC approach leads to a parallel combination of two-class classifiers whose parameters are normally easier to determine. In this Letter, we look at the conditions necessary for reduction of error in the ECOC framework, and introduce a modified version that is less sensitive to code word selection.

Error-correcting coding and classification

ECOC is an example of a distributed output code [3] in which a pattern is assigned to a class according to closest distance to a binary code word. The idea of using codes with error-correcting properties is based on modelling the prediction task as a communication problem in which class information is transmitted over a channel. The assumption for a

2-class problem is that errors introduced into the process can be of two kinds, either predicted class Ω_1 for target class Ω_2 or predicted Ω_2 for target Ω_1. The *k x b* code word matrix Z 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 (typically 0 or1) variable *x*, a training pattern with target class ω_i *(i = 1... k)* is re-labelled as class Ω_1 if $Z_{ij} = x$ and as class Ω_2 if $Z_{ij} = \overline{x}$. The two super-classes Ω_1 and Ω_2 represent, for each column, a different decomposition of the original problem. In the test phase a pattern is assigned to the class ω_i that is represented by the closest code word. In general the classifier outputs y_j *(j = 1... b)* are real numbers, and typically distance is defined as

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

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 [1] is that application of errror-correcting principles does lead to reduced test error rate.

By analysing errors in the ECOC framework we can discover requirements of Z for good generalisation. In addition to error caused by overlapped classes (which cannot of course be removed), two kinds of errors can be identified, that due to individual classifiers and that due to the combining strategy. Maximising Hamming Distance between any pair of code words is intended to improve error-correcting capability and remove individual classification errors on the re-labelled training sets, but even if classifiers are perfect (Bayesian) there will still be errors due to combining. The combining error can be categorised into (i) errors due to inability of sub-problems to represent the main problem, and (ii) errors due to the distance-decision rule used for combining. To address (i), note that sub-problems are more independent and likely to benefit from combining [4] if Hamming distance between columns is maximised. To understand (ii), consider the relationship betwen the distance-based combining strategy (equation 1) and Bayes decision rule. Let us assume that each classifier provides exactly the posterior probability of respective super-class membership, with posterior probability of class w_l represented as q_l ($l = 1 ... k$)

From equation 1

$$D_i = \sum_{j=1}^{b} \left| \left(\sum_{l=1}^{k} q_l Z_{lj} \right) - Z_{ij} \right| \tag{2}$$

$$\therefore D_i = (1 - q_i) \sum_{j=1}^{b} \left| Z_{ij} - Z_{lj} \right| \tag{3}$$

Equation 3 tells us that D_i is the product of $(1-q_i)$ and Hamming Distance between code words. When all pairs of code words are equi-distant, minimising D_i 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.

Finding optimal code matrices having maximum and equal distance between row pairs and maximum distance between column pairs is a complex problem. In lieu of optimal codes, longer random codes have been employed with almost as good performance [1]. However we might suspect that certain code words are suited to some super-classes more than others. To reduce this potential sensitivity to code word selection, we suggest a modified algorithm:

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 ω_i between b-dimensional base classifier outputs and code word corresponding to class ω_i

Assign pattern to class ω_m given by

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

Results

An artificial five-class overlapping Gaussian problem is defined by five groups of two-dimensional random vectors having normal distribution, shown in Table 1. The classifiers are not trained, but using the parameters from Table 1, the posterior probability of super-class membership is computed. Code matrices for comparison include BCH [5] and random codes, and are defined as follows: Z1 is 5x7 random, Z2 is 5x7equi-distant BCH, Z3 is 7x15 equidistant BCH and Z4 is 5x15 equi-distant random. The error-correcting capability for Z1, Z2, Z3, Z4 is 1,3,7,2 bits respectively. Increasing levels of (zero mean) Gaussian noise are added to real-valued output of individual classifiers to simulate noisy imperfect classifiers. Figure 1 shows rate of matching with Bayes rate as noise (σ) is increased and demonstrates the robustness of the modified algorithm. Also from Figure 1 the following points can be observed for both modified and original approach: (i) if no noise is added (σ = 0), the equi-distant matrices Z2, Z3 have zero error as expected, (ii) if the number of classifiers is the same, the optimal code performs slightly better than random code, shown by comparing compare Z4 with Z3 (b = 15) and Z2 with Z1 (b = 7), (iii) if longer random code is used, repetition of sub-problems gives improved performance even if error-correction capability is reduced, shown by comparing Z4 with Z2 .

Figure 2 compares original and modified decision boundaries when the same noise (σ = 0.25) is added to each individual classifier, and confirms that error variance has been reduced. To confirm superiority of the modified algorithm, real benchmark data-sets were chosen from [6], with conventional single hidden layer MLP as base classifier, fixed parameters and fifty epochs trained by Levenberg-Marquardt. Ten independent runs using 15-column random code matrix were repeated for random training/testing splits with numbers of training patterns and hidden nodes specified: car/100/1, cmc/100/1, gauss5/200/2, iris/50/1, segment/200/1, vehicle/350/5, zoo/50/1. The improvement of the modifed over original algorithm was 5% mean and 30% std averaged over the seven data-sets.

Conclusion

Errors associated with using error-correcting code words as distributed output codes were analysed, with particular reference to combining strategy and Bayes decision rule. While longer random codes perform almost as well as optimal codes, performance is improved if senstivity to code word assignment is reduced.

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

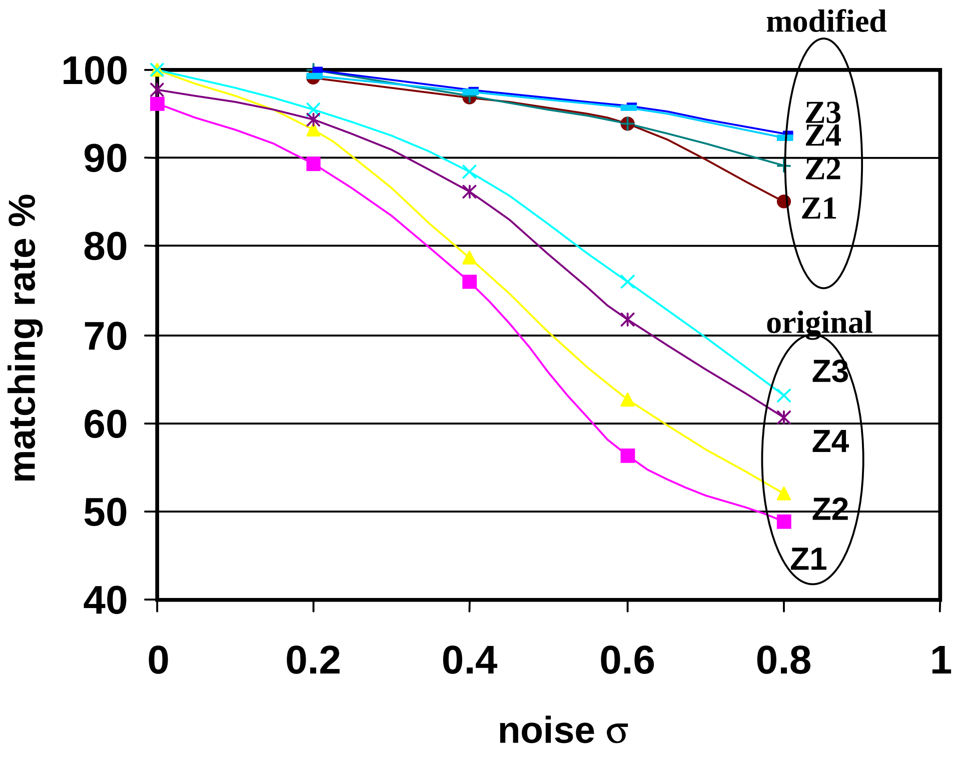

Figure 1

Matching (%) with Bayes rate for increasing levels of Gaussian noise (σ) and different code matrices for modified ECOC (top 4 curves) and original method

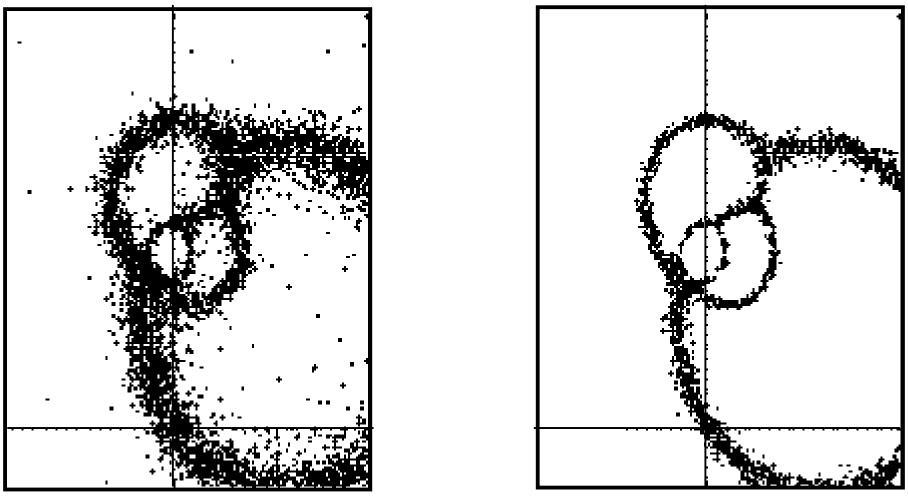

Figure 2

Comparison of original (left) and modified ECOC decision boundaries for random 5 x 15 code matrix Z4 when noise ($\sigma = 0.25$) is added to simulate imperfect classifiers

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

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

3 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.

4 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.

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

6 C.J. Merz and P. M. Murphy, UCI repository of machine learning databases, 1998 http://www.ics.uci.edu/~mlearn/MLRepository.html