ECOC-ONE: A novel coding and decoding strategy

Sergio Escalera
CVC, Computer Science Department, UAB
Campus UAB, 08193 Bellaterra, Spain
sescalera@cvc.uab.es

Oriol Pujol
Dept. Matemàtica Aplicada i Anàlisi
UB, Gran Via 585, 08007, Barcelona, Spain
oriol@cvc.uab.es

Petia Radeva
CVC, Computer Science Department, UAB
Campus UAB, 08193 Bellaterra, Spain
petia@cvc.uab.es

Abstract

Error correcting output codes (ECOC) represent a classification technique that allows a successful extension of binary classifiers to address the multiclass problem. In this paper, we propose a novel technique called ECOC-ONE to improve an initial ECOC configuration by including new dichotomies guided by the confusion matrix over exclusive training subsets. In this way, the initial coding represented by an optimal decision tree is extended adding binary classifiers forming a network. Since not all dichotomies have the same relevance, a weighted methodology is included. Moreover, to decode we introduce a new distance to attenuate the error accumulated by zeros in the ECOC-ONE matrix. We compare our strategy to other well-known ECOC coding strategies on the UCI data set achieving very promising results.

1. Introduction

Error correcting output codes were born as a general framework to handle multiclass problems using binary classifiers [2]. It is well-known that ECOC, when applied to multiclass learning problems, can improve the generalization performance [6][1]. One of the reasons for this improvement is its property to decompose the original problem into a set of complementary two-class problems in the ECOC matrix that allow sharing of classifiers across the original classes. Another well-known technique for extending binary classifiers to the multiclass problem is the nested dichotomies strategy in form of independent binary trees to vote [3], where the generation of the tree depends on the nature of the problem we are solving. However, there are usually many candidate trees for a given problem and in the standard approach the choice of a particular tree is based on a priori domain knowledge that may not be available in practice. Recently, the embedding of tree structures in the ECOC framework has been proposed [5] and shown to obtain high accuracy with a very small number of binary classifiers. In this work, the discrete ECOC coding step proposed is application-dependent but independent of the classification strategy.

The goal of this article is twofold: firstly, we introduce a new coding strategy. Its starting point comes from an embedding of an optimal tree according to the classification score. This tree is extended by selective greedy optimization based on the confusion matrix of exclusive training subsets. The candidate dichotomies to be included come from previously generated optimal subsets of binary classifiers if they help the ECOC convergence. Our procedure creates an ECOC code that splits optimally the classes while reducing the number of classifiers used, increasing the Hamming distance between the difficult to discriminate classes. Similar to the Adaboost technique, we propose a weighting strategy that gives more attention to the most discriminant dichotomies. Moreover, our procedure uses a new decoding technique based on a weigthed Euclidean distance that attenuates the error due to the zeros in the ECOC matrix.

The article layout is as follows: section 2 introduces the general procedure of ECOC techniques used in the literature. Section 3 describes the new approach ECOC-ONE. Section 4 shows the experiment results and section 5 concludes the paper.

2. ECOC

For some classification problems, it is known that the lowest error rate is not always reliably achieved by trying to design a single classifier. An alternative approach is to employ a set of relatively simple sub-optimal classifiers and to determine a combination strategy that pools together the results. The basis of the ECOC framework is to create a codeword for each class of N_c classes (up to N_c codewords). Arranging the codewords as rows of a matrix we define the "coding matrix" M, where $M \in \{-1, 1\}^{N_c \times n}$, being n the code length. From the point of view of learning, the matrix M is represented as n binary learning problems (dichotomies from now on), each corresponding to a column of the ECOC matrix. Each dichotomy defines a partition of classes (coded by +1,-1 according to their class membership). As a result of the outputs of the n binary classifiers, a code is obtained for each data point in the test set. This code is compared with the base codewords of each class defined in the matrix M, and the data point is assigned to the class with the "closest" codeword. When we use a larger set, $M \in \{-1, 0, 1\}^{N_c \times n}$, some entries in the matrix M can be zero, indicating that a particular class is not considered for a given dichotomy. To design an ECOC system, we need a coding and a decoding strategy. 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 generalize well. A good error-correcting output code for a k-class problem should satisfy that rows and columns are well-separated from the rest in terms of Hamming distance (avoiding complementaries).

Most of the discrete coding strategies up to now are predesigned problem-independent codewords satisfying the former row and column properties. These strategies include one-versus-all, random techniques [1], and one-versus-one [4]. The last one mentioned has obtained high popularity showing a better accuracy in comparison with the other commented strategies. Recently, the embedding of tree structures [5] allows us to generate discrete application-dependent ECOC independent of the classification strategy.

Concerning the decoding strategies, one of the most standard technique is the Hamming decoding distance, $d_j = \sum_{i=1}^{n} |(x_i - y_i^j)|/2$, where d_j is the distance to the row j, n is the number of dichotomies, and x and y are the values of the input vector codes and base class codeword, respectively.

3. ECOC-ONE

Given a multiclass recognition problem, our procedure starts with the generation of an optimal tree that is included in the ECOC matrix, and allows to share the information across classes. We increase this ECOC matrix, in an iterative way, adding dichotomies that correspond to different spatial partitions of subsets of classes. These partitions are found using a greedy optimization of the confusion matrix so that the ECOC accuracy improves on both exclusive training subsets. Our training set is partitioned in 2 training subsets: the training subset, that guides the convergence process, and the validation subset, that leads the optimization process in order to avoid classification overfitting. Since not all problems require the same dichotomies, our optimum node embedding approach (ECOC-ONE), generates an optimal ECOC-ONE matrix dependent of our domain, forming a network with a reduced number of dichotomies.

To explain our procedure, we divide the algorithm in three steps: optimal tree generation, weights estimation, and optimum node embedding. The resumed algorithm is shown in fig. 1.

```
Given $N_c$ classes,
-Generate the optimal tree,
-Include the network nodes in the ECOC-ONE Matrix M,
for  t= 1 to T iterations,
    -Calculate node $h_t$:
        - Test accuracy on the validation subset.
        - Select  the pair of classes {$c_i$, $c_j$} with the highest
        error analyzing the confusion matrix $\vartheta$ on the
        validation subset.
        - Calculate $h_t$ with the partition of classes
        according from the training subset that minimizes
        the validation error for {$c_i$, $c_j$}.
    -Update dichotomy weight $w_t$ using (2) and M
```

Figure 1. ECOC-ONE Coding algorithm.

3.1. Optimal tree generation

The first dichotomies included in our ECOC-ONE matrix are the binary classifiers of the optimum tree generated for our problem. Since the tree used in [5] uses the mutual information to form the nodes, the computational cost can be very high and it does not assure us the best partitions for a given classifier. We use the classification score to create the optimum discrimination tree associated to that classifier.

Each node of the tree is generated by an exhaustive search of partitions of the classes associated to the parent node. Once we have generated the optimal tree, we embed each internal node of the tree in the following way: Consider the set of classes associated to a node $C_i = \{C_{i1} \cup C_{i2} | C_{i1} \cap C_{i2} = \emptyset\}$, the column i of the ECOC-ONE matrix M and row r corresponding to class c_r is filled as follows:

$$M(r, i) = \begin{cases} 0 & \text{if } c_r \notin C_i \\ +1 & \text{if } c_r \in C_{i1} \\ -1 & \text{if } c_r \in C_{i2} \end{cases} \qquad (1)$$

3.2. Weights estimations

Similar to boosting algorithms, our approach uses a weight to adjust the importance of each dichotomy in the ensemble ECOC matrix. In particular, the weight associated to each column depends on the error when applying the ECOC to the validation subset in the following way,

$$w_i = 0.5 \log(\frac{1 - e_i}{e_i}) \qquad (2)$$

where w_i is the weight for the ith dichotomy, and e_i is the error produced by this dichotomy at the affected classes of the validation subset.

3.3. Optimum node embedding

Test reliability of the validation subset: To introduce each network node, first, we test the current M reliability with the validation subset. For this step, we find the resulting codeword $x \in \{-1, 1\}^n$ for each class sample of the validation subset, where n is the number of trained dichotomies, and we label it as follows:

$$x \in C_j \quad if \quad j = argmin_{N_c} d_j \tag{3}$$

where j is the class c_j, and d_j is the distance estimation between x and the base codeword y for class c_j. The distance d is calculated for the validation subset sample and each class codeword using the following weighed Euclidean distance:

$$d_j = \sqrt{\sum_{i=1}^{n} |y_i|(x_i - y_i^j)^2 w_i} \tag{4}$$

We introduce the weight $|y_i|$ to attenuate the error that can be accumulated by zeros in the ECOC-ONE matrix M, and the weight w_i to adjust the importance of each dichotomy.

Once we test the reliability on the validation subset, we estimate its confusion matrix $\wp$.

The confusion matrix $\wp$: The confusion matrix $\wp$ is of size N_c^2, and it has at position (i, j) the number of instances of class c_i classified as class c_j. We select the pair $c_i c_j$ that maximizes $\wp(i, j) + \wp(j, i) | i \neq j \forall (i, j) \in [1, ..., N_c]$ from the validation subset confusion matrix $\wp$.

Calculate h_t: To optimize classes $\{c_i, c_j\}$ we consider all possible partitions $(c_i C_1, c_j C_2) \subseteq C | C_1 \cap C_2 \cap c_i \cap c_j = \emptyset$ from the training subset. We select the dichotomy h_t at iteration t that considers the partition that minimizes the error of classes $c_i c_j$ at validation subset using the classification score.

Update dichotomy: Node h_t is included in M with weight w_t (2). This process is iterated while the error on the validation subset is greater than ε or the number of iterations $t < T$.

To classify a new input, a codeword x of length n is generated and labeled using (3).

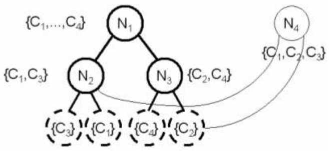

Figure 2. First optimal node embedded.

	N_1	N_2	N_3	N_4
C_1	1	-1	0	1
C_2	-1	0	-1	-1
C_3	1	1	0	1
C_4	-1	0	1	0

Figure 3. ECOC-ONE code matrix M for four dichotomies from the network of fig. 2.

Using the optimal nodes, the initial tree structure is upgraded to a network that is embedded in the ECOC-ONE matrix M. In fig. 2, the network for a toy problem formed by an initial optimum tree and the first optimal node is shown. Suppose that classes c_2, c_3 maximize the error in the confusion matrix $\wp$ for the validation subset. We search for the partition of classes using the training set so that the error c_2, c_3 is minimized. Suppose now that the new node, N_4, considers that the best partition is c_1, c_3 versus c_2. We can observe in fig. 2 that N_4 uses a class partition that is present in the tree. In this sense, this new node connects different branches of the tree creating the network. Note that using the previously included dichotomies, the partition c_1, c_3 is solved by N_2. In this way, the Hamming distance between c_2 and c_3 is increased by adding the node in the network. However, the distance among the rest of the classes is usually maintained or slightly modified.

As mentioned before, one of the desirable properties of the ECOC matrix is to have maximum distance between rows. In this sense, our procedure focuses on the relevant difficult partitions, increasing the distance between the classes. This fact increases the robustness of the method since difficult classes are likely to have a greater number of dichotomies focussed on them and, therefore, more error correction.

4. Results

To test our method, we compare it to the most well-known strategies used for ECOC coding: one-versus-all ECOC (1-vs-all), one-versus-one ECOC (1-vs-1), and Dense random ECOC. We have chosen dense random coding because it is more robust than the sparse technique when the number of colums is small [1]. The decoding strategy for all mentioned techniques is the standard Hamming decoding distance. We compare them with our ECOC-ONE strategy for coding and our weigthed Euclidean distance for decoding. We compute 10 iterations or dichotomies after the inclusion of the first optimal tree. In order to have reliable results we have used the same number of dichotomies for the generation of the Dense Random ECOC matrix columns. The Dense Random matrix is selected from an exhaustive search of 10000 iterations. We have used discriminant analysis as weak learner for all techniques. All tests are calculated using ten-fold cross-validation and a two-tailed

Problem	one-vs-one	one-vs-all	Dense random	ECOC-ONE
(a)	96.65±0.73	94.87±0.74	96.57±0.74	**98.48±0.49**
(b)	**82.40±1.46**	71.85±1.53	**81.15±1.55**	**83.90±1.23**
(c)	**76.76±1.16**	44.55±2.15	44.83±2.00	52.10±2.28
(d)	**85.24±0.57**	71.32±0.62	73.92±0.56	**85.44±0.50**
(e)	**71.20±1.27**	23.87±0.42	41.32±1.38	53.05±0.80
(f)	81.00±0.67	65.35±0.52	75.85±0.83	**82.85±0.54**
(g)	**52.21±0.80**	30.54±0.90	47.32±0.93	**51.21±0.70**
(h)	**93.18±0.43**	33.10±1.23	68.41±1.44	**91.21±0.78**

Table 1. ECOC Strategies hits for UCI databases using FLDA as a base classifier.

t-test for a 95% confidence interval. Finally, to compare, we have used a set of known databases from UCI repository. The results are shown in table 1. The description of each database is shown in table 2. We can see that our method is very competitive when compared to other standard ECOC coding techniques. And it attains a comparable accuracy to the 1-vs-1 ECOC coding strategy, which is known to usually obtain the best results. In some cases, 1-vs-1 improves our results for a certain database. For example, at Pendigits database, 1-vs-1 obtains a two percent of improvement over our method. However, one must note that 1-vs-1 requires 45 dichotomies in that database, but we only use 15. These results are easily explained by the fact that our method chooses at each step the most discriminable dichotomy. This procedure allows us to classify classes depending of their difficulty, reducing the number of binary classifiers to be selected. This is demonstrated observing the results of Dense Random ECOC and our procedure. Both cases have the same number of dichotomies, and although Random ECOC has a higher distance between rows, our procedure always obtains a higher hit ratio because the dichotomies are selected in an optimal way depending on the domain of the problem.

To accelerate our coding method, we store all the classifiers trained in previous iterations. In fig. 4 we can see the error evolution for our procedure for an iteration of Dermathology UCI database.

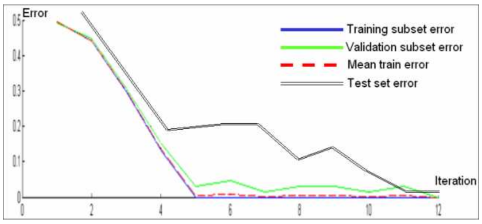

Figure 4. Error evolution for an iteration of Dermathology database using ECOC-ONE.

Problem	Database	#Train	#Test	#Attributes	#Classes
(a)	Dermathology	366	-	34	6
(b)	Ecoli	336	-	8	8
(c)	Glass	214	-	9	7
(d)	Segmentation	2310	-	19	7
(e)	Vowel	990	-	10	11
(f)	Satimage	4435	2000	36	6
(g)	Yeast	1484	-	8	10
(h)	Pendigits	7494	3498	16	10

Table 2. UCI databases characteristics.

5. Acknowledgements

This work was supported in part by the projects TIC2003-00654, FIS-G03/1085, FIS-PI031488, and MI-1509/2005.

6. Conclusions

In most of the ECOC coding strategies, the ECOC matrix is pre-designed, using the same dichotomies in any type of problem. We introduced a new coding and decoding strategy called ECOC-ONE and Weighted Euclidean Distance, respectively. The idea is based on the embedding of a graph in the ECOC matrix formed by an initial optimal tree and upgraded with a set of optimal dichotomies (nodes) to form a network. The procedure shares classifiers among classes in the ECOC-ONE matrix, and selects the best partitions weighed by their relevance. In this way, it reduces the overall error for a given problem. We show that this technique improves the Random ECOCS results. We compete with the 1-vs-1 ECOC strategy using a smaller number of dichotomies. As a result, a compact multiclass recognition technique with improved accuracy is presented with very promising results.

References

[1] E. Allwein, R. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, vol. 1:113–141, 2002.

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

[3] E. Frank and S. Kramer. Ensembles of nested dichotomies for multiclass problems. In *Proceedings of* 21^{st} *International Conference on Machine Learning*, pages 305–312, 2004.

[4] T. Hastie and R. Tibshirani. Classification by pairwise grouping. *The annals of statistics*, vol. 26(5):451–471, 1998.

[5] O. Pujol, P. Radeva, and J. Vitri. Discriminant ecoc: A heuristic method for application dependent design of error correcting output codes. *Transactions on PAMI*, 28(6):1001–1007, 2006.

[6] T. Windeatt and R. Ghaderi. Coding and decoding for multiclass learning problems. *Information Fusion*, vol. 4(1):11–21, 2003.