Compact Design of ECOC for Multi-class Object Categorization

Miguel Angel Bautista[*+], Sergio Escalera[*+], Xavier Baro[−]
Oriol Pujol[*+], Petia Radeva[*+] and Jordi Vitria[*+]

[*] *Applied Math and Analysis Dept., University of Barcelona, Gran Via de les Corts 585, Barcelona, Spain.*
E-mail: miguelangelbautistamartin@gmail.com, {sergio, oriol, petia, jordi}@maia.ub.es

[+] *Computer Vision Center, Autonomous University of Barcelona, Edifici O Campus UAB , Cerdanyola, Spain.*
E-mail: {sergio, oriol, petia, jordi}@cvc.uab.es

[−] *Department of Computer Science, Universitat Oberta de Catalunya, Rambla del Poblenou 158, Barcelona, Spain.*
E-mail: xbaro@uoc.edu

Abstract

In this paper, we propose a Compact design of Error Correcting Output Codes (ECOC) in terms of the number of dichotomizers. Evolutionary computation is used for tuning the parameters of the classifiers and looking for the best Compact ECOC code configuration. The results over several challenging multi-class Computer Vision problems show comparable and even better results than state-of-the-art ECOC methodologies with far less cost.

Keywords: Ensemble Learning, Error Correcting Output Codes, Evolutionary Optimization, Multi-class Object Categorization.

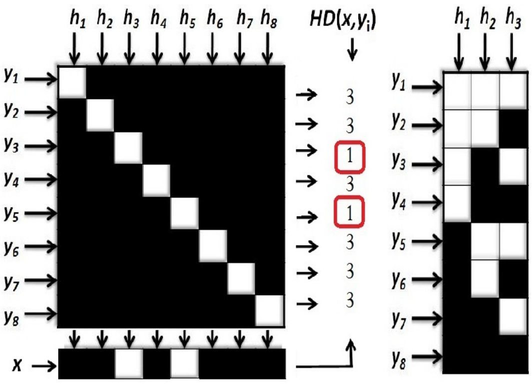
1 Introduction

A common way to deal with Multi-class Object Categorization problems is by means of a divide-and-conquer approach. In this scope, ECOC have been applied with successful results. ECOC encodes different partitions of the problem in a matrix of codewords1(a)(one codeword per class) and the final decision is obtained by looking at the most similar codeword at the test step. Given N different classes classical ECOC designs require between N and N^2 classifiers, being prohibitive when the number of classes becomes large. The proposal of this work relies on the ECOC framework, reducing the number of binary classifiers of the ensemble. We propose a Compact ECOC design of size $log_2(N)$ in terms of the number of classifiers. An evolutionary approximation is proposed for tuning the parameters of the classifiers and looking for a Compact design with high generalization capability. The novel Compact ECOC is compared with the state-of-the-art ECOC approaches, obtaining comparable and even better results with far less cost. The paper is organized as follows: Section 2 presents the Compact ECOC design. Section 3 evaluates the novel methodology on different Computer Vision datasets. Finally, Section 4 concludes the paper.

2 Compact ECOC

2.1 Error-Correcting Output Codes

Given a set of N classes to be learnt in an ECOC framework, n different bi-partitions (two groups of classes) are formed, and n binary problems (di-



(a) (b)

Figure 1: Example for Hamming Decoding (a) and Compact Coding (b)

chotomizers) over the partitions are trained. As a result, a codeword of length n is obtained for each class, where each position (bit) of the code corresponds to a response of a given dichotomizer (coded by +1 or -1 according to their class set membership). Arranging the codewords as rows of a matrix, we define a *coding matrix* M, where $M \in \{-1,+1\}^{N\times n}$ in the binary case. In the case of the ternary symbol-based ECOC, the coding matrix becomes $M \in \{-1,0,+1\}^{N\times n}$ where the symbol zero means that a particular class is not considered for a given classifier. Details of classical binary and ternary designs can be found in [2]. During the decoding process, applying n binary classifiers, a code x is obtained for each data sample ρ in the test set. This code is compared to the base codewords ($y_i, i \in [1,..,N]$) of each class defined in the matrix M, and the data sample is assigned to the class with the *closest* codeword.

2.2 Compact ECOC Coding

The one-versus-all ECOC coding has been widely applied in the binary ECOC framework (see Figure 1(a)). Given N classes to be coded, the one-versus-all codification is of length N. Recently, several problem-dependent designs have been also proposed [7]. Those new techniques are based on exploiting the problem domain by selecting the representative binary problems that increase the generalization performance while keeping the code length "relatively" small. All previous designs consider a large number of dichotomizers.

However, we can take advantage of the information theory principles to obtain a more compact definition of the codewords. Having a N-class problem, the minimum number of bits necessary to codify and univocally distinguish N codes is $B = \lceil \log_2 N \rceil$, where $\lceil . \rceil$ rounds to the upper integer, an example can be seen in figure 1(b).

Moreover, instead of using a predefined Compact coding matrix, we also propose the design of a different compact codification of M based on the distribution of the data and the characteristics of the applied base classifier, which can increase the discrimination capabilities of the system. However, finding a suitable *Compact ECOC* matrix for a $N-$class problem requires to explore all the possible $N \times B$ binary matrices, where B is the minimum codeword length in order to define a valid ECOC matrix. Because of this reason, we also propose an evolutionary parametrization of the Compact ECOC design.

2.2.1 Evolutionary Compact Parametrization

Given N classes, the number of ECOC matrices that we can build is $\sharp M = \frac{V^N_{2^B}}{2} = \frac{2^B!}{2*(2^B-N)!}$. In these type of scenarios, where the search space is huge, evolutionary approaches, in special Genetic Algorithms, are often introduced with good results [1].

Problem encoding: The first step in order to use an evolutionary algorithm is to define the problem encoding, which consists of the representation of a certain solution or point in the search space by means of a *genotype* or alternatively a *chromosome*. In binary encoding, every chromosome is a string of bits 0 or 1. Although this encoding is often not natural for many problems and sometimes corrections must be performed after crossover and/or mutation, in our case, the chromosomes represent binary ECOC matrices, and therefore, this encoding perfectly adapts to the problem. Each ECOC is encoded as a binary chromosome $\zeta =< h_1^{c_1}, \ldots, h_B^{c_1}, h_1^{c_2}, \ldots, h_B^{c_N} >$, where $h_i^{c_j} \in \{0,1\}$ is the expected value of the $i-th$ classifier for the class c_j, which corresponds to the $i-th$ bit of the class c_j codeword.

Adaptation function: Once the encoding is defined, we need to define the adaptation function, which associates to each individual its adaptation value to the environment. In the case of the ECOC framework, the adaptation value must be related to the classification error. Given a chromosome $\zeta =< \zeta_0, \zeta_1, \ldots, \zeta_L >$ with $\zeta_i \in \{0,1\}$, the first step is to recover the ECOC matrix M codified in this chromosome. The values of M allows to create binary classification problems from the original multi-class problem, following the partitions defined by the ECOC columns. Each binary problem is addressed by means of a binary classifier, which is trained in order to separate both partitions of classes.

Evolutionary process: We use the standard Genetic Algorithm in order to evolve the Compact ECOC matrices. During the evolutionary process, we use a scattered crossover operator, in which, we generate a random binary vector, with a binary value assigned to each gene. The first child is created using all the genes from the first parent in those positions with a value of one, and the genes of the second parent with positions with the value zero. The second child is created as the complementary of the first one. In order to introduce variations to the individuals, we use mutation operator that adds a unit Gaussian distributed random value to the chosen gene.

Finally, we adopt an *Island Model* evolution scheme in order to exploit a more coarse grain parallel model. The main idea is to split a population of K individuals into S sub-populations of K/S individuals. By introducing migration, the *Island Model* is able to exploit differences in the various sub-populations.

Learning the binary classifiers: In this paper we adopt the Support Vector Machines with Gaussian Radial Basis Functions as kernel (SVM-RBF). In the specific case of Gaussian RBF kernels, we need to learn the kernel parameters C and γ, which have a close relation to the data distribution. In our case, for each binary problem, we use Genetic Algorithms in order to find good values for C and γ parameters.

3 Results

In order to present the results, first, we discuss the data, methods, and evaluation measurements of the experiments.

- **Data**: We apply the methodology in five challenging computer vision categorization problems. Labeled Faces in the Wild [5] (184 face categories). We use a real traffic sign categorization problem [3] (36 traffic sign classes). Third, the ARFaces dataset [6] (20 classes). Fourth, we classify old scanned music scores [4] (7 score categories), and fifth, we classify the MPEG7 dataset [1] (70 object categories).
- **Methods**: We compare the one-versus-one and one-versus-all ECOC approaches with the binary and evolutionary compact approaches. The Hamming decoding is applied at the decoding step. The ECOC base classifier is the OSU implementation of SVM with Radial Basis Function kernel. The SVM C and γ parameters are tuned via Genetic Algorithms for all the methods, minimizing the classification error of a two-fold evaluation over the training sub-set.

The results are shown in Table 1.

Analyzing globally the results of the Computer Vision classification problems, which ranks are shown in the last row of Table 1, one can see that globally, the one-versus-one is the first choice, followed by the evolutionary proposal. The last two positions are for the binary and one-versus-all coding designs. This result encourages the use of

[1] http://www.cis.temple.edu/latecki/research.html

Figure 2: Examples of the 5 Computer Vision datasets

	Binary Compact ECOC		Evol. Compact ECOC		one-vs-all ECOC		one-vs-one ECOC	
Data set	Perf.	Classifiers	Perf.	Classifiers	Perf.	Classifiers	Perf.	Classifiers
FacesWild	26.4±2.1	10	30.7±2.3	10	25.0±3.1	184	-	16836
Traffic	90.8±4.1	6	90.6±3.4	6	91.8±4.6	36	90.6±4.1	630
ARFaces	76.0±7.2	5	85.84±5.2	5	84.0±6.3	20	96.0±2.5	190
Clefs	81.2±4.2	3	81.8±9.3	3	80.8±11.2	7	84.2±6.8	21
MPEG7	89.29±5.1	7	90.4±4.5	7	87.8±6.4	6.170	92.8±3.7	2415
Rank & # Classifiers	3.0	**5.2**	2.2	**5.2**	3.0	33.2	**1.5**	814.0

Table 1: Computer Vision data sets classification results.

the Compact approach since there is no evidential significance in using any of the four methods. Moreover, note the reduced number of dichotomizers required by Compact approaches diminishing in some cases a 400% the number of classifiers needed by classical approaches.

4 Conclusions

We presented a general methodology for the classification of several object categories which only requires $\lceil \log_2 N \rceil$ classifiers for a N-class problem. The methodology is defined in the ECOC framework, designing a Compact coding matrix which univocally distinguish N codes. Moreover, in order to speed up the design of the coding matrix and the tuning of the classifiers, evolutionary computation is also applied.

Results over different Computer Vision problems show comparable ever better results than traditional ECOC designs with far less number of dichotomizers.

References

[1] K. DeJong. *An analysis of the behavior of a class of genetic adaptive systems.* PhD thesis, Ann Arbor, MI, USA, 1975.

[2] S. Escalera, O. Pujol, and P.Radeva. On the decoding process in ternary error-correcting output codes. *IEEE PAMI*, 99(1), 2009.

[3] S. Escalera, O. Pujol, and P. Radeva. Traffic sign recognition system with β-correction. *MVA*, 21(2):99–, 2010.

[4] A. Fornés, J. Lladós, and G. Sánchez. Primitive segmentation in old handwritten music scores. *Graphics Recognition*, 3926:279–290, 2006.

[5] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[6] A. Martinez and R. Benavente. The AR face database. In *Computer Vision Center Technical Report #24*, 1998.

[7] O. Pujol, S. Escalera, and P. Radeva. An incremental node embedding technique for error correcting output codes. In *Pattern Recognition*, volume 14, pages 713–725, 2008.