On the Design of an ECOC-Compliant Genetic Algorithm

Miguel Ángel Bautista [1,2], Sergio Escalera[1,2], Xavier Baró[1,2,3], Oriol Pujol[1,2]

[1] *Applied Math and Analisis Dept, University of Barcelona, Gran Via de les Corts Catalanes. 585, 08007 Barcelona, Spain*
[2] *Computer Vision Center, Campus UAB, Edifici O, 08193, Bellaterra, Spain*
[3] *Computer Science, Multimedia, and Telecommunications Dept, Universitat Oberta de Catalunya. Rambla del Poblenou 156, 08018 Barcelona*
mbautista@cvc.uab.es,oriol@maia.ub.es,xbaro@uoc.edu,sergio@maia.ub.es

Abstract

Genetic Algorithms (GA) have been previously applied to Error-Correcting Output Codes (ECOC) in state-of-the-art works in order to find a suitable coding matrix. Nevertheless, none of the presented techniques directly take into account the properties of the ECOC matrix. As a result the considered search space is unnecessarily large. In this paper, a novel Genetic strategy to optimize the ECOC coding step is presented. This novel strategy redefines the usual crossover and mutation operators in order to take into account the theoretical properties of the ECOC framework. Thus, it reduces the search space and lets the algorithm to converge faster. In addition, a novel operator that is able to enlarge the code in a smart way is introduced. The novel methodology is tested on several UCI datasets and four challenging computer vision problems. Furthermore, the analysis of the results done in terms of performance, code length and number of Support Vectors shows that the optimization process is able to find very efficient codes, in terms of the trade-off between classification performance and the number of classifiers. Finally, classification performance per dichotomizer results shows that the novel proposal is able to obtain similar or even better results while defining a more compact number of dichotomies and SVs compared to state-of-the-art approaches.

Keywords: ECOC, Genetic Algorithms, Multi-class classification

1. Introduction

In classification problems the goal is to find a function $f : S \rightarrow K$, where S is the set of observations and K the set of possible mutually exclusive labels ($|K| > 2$ for the multi-class context). The goal of f is to map any possible observation $s \in S$ to a label $k \in K$. There are many possible strategies for estimating f, nevertheless, literature has shown that the complexity for estimating a unique f for the whole multi-class problem grows with the cardinality of the

label set. In this sense, most of the strategies aim to either model the probability density function of each category. Moreover, lazy learning methods like Nearest Neighbours which try to estimate k by a local search of the most proximate observations. Another noticeable procedure is known as the divide and conquer approach, in which instead of developing a method to cope with the multi-class case, the classification task is divided into a set of n binary problems which are treated separately. Once the responses of binary problems are obtained one can use some kind of committee strategy to obtain the final output. In this trend one can find three main lines of research: flat strategies, hierarchical classification, and Error Correcting Output Codes. Flat strategies like One vs. One and One vs. All are those that use a predefined problem partition scheme followed by some kind of committee strategy. On the other hand, hierarchical classification relies on some similarity metric distance among classes to build a binary tree which nodes correspond to different problem partitions. Finally, the ECOC framework consists of two steps: In the *coding* step, a set of binary partitions of the original problem are encoded in a matrix of codewords (univocally defined, one code per class). At the *decoding* step a final decision is obtained by comparing the test codeword with every class code and choosing the class with the code at minimum distance [1, 2]. In this sense, the ECOC approach can be considered as a generalization of the former strategies since it allows the inclusion of both flat and hierarchical strategies as shown in [3, 4]. Moreover, [5] proved that the use of ECOC reduces bias and variance errors produced by algorithms that learn the binary problem. However, note that *predefined* ECOC strategies need between N and $\binom{N}{2}$ classifiers to deal with an $N-$class problem. Although this is acceptable for a small number of categories, it becomes a great scalability problem when the number of classes is arbitrarily large. This number has been reduced in literature in the Dense and Sparse Random designs to a length $10 \cdot \log_2 N$ and $15 \cdot \log_2 N$, respectively [3]. Nevertheless, predefined or random problem partition approaches like One vs. One, One vs. All, Dense or Sparse Random may not be suitable for a given problem since they clearly overlook the underlying distribution of the data. However, in order to take into account the data distribution, researchers have developed some problem-dependent techniques. For example, the DECOC approach which defines $N-1$ dichotomizers [6]. In particular, [7] demonstrated that finding the optimum ECOC matrix for a given problem (the optimal set of binary partitions) was an *NP-Complete* problem. Thus, in order to obtain compact problem-dependent designs some works have applied GA to the ECOC coding matrix. In [8] the authors defined an ECOC code with $log_2 N$ dichotomies and the approach in [9] defined a code in the range $[logN, N]$ dichotomies. Finally, Pedrajas et. al [10] applied a GA to obtain the ECOC coding matrix, though in the experimental setting they fix the code to a length of 30, 50 or 200 dichotomies. Nevertheless, those works lacked of a theoretical basis when taking into account certain crucial aspects of ECOC matrices that have to be carefully revisited. Figure 2 shows the code length yielded by each method in relationship with the number of classes.

Genetic Algorithms are a family of optimization processes based on Darwin's evolution theory. In these processes, points in the search space are seen as in-

dividuals and evaluated by means of a fitness function. This function provides the value of its adaptation to the environment, and thus, more adapted individuals represent better solutions of the optimization problem. These optimization processes converge to a population (set of individuals) which accomplish a certain optimization criteria. In order to achieve this goal, individuals are mapped into binary vectors (in standard GAs), which suffer transformations due to either random changes (also known as mutations) or exchanges of information between individuals (also known as recombinations or crossovers) along generations. The application of GAs in the ECOC coding step has been the focus of some recent works [8, 9, 10].

[...][8] proposed a standard GA to optimize an ECOC matrix, known as Minimal ECOC matrix, which is the theoretical lower-bound in terms of the number of classifiers $\lceil \log_2 N \rceil$. In this work the evaluation of each individual (ECOC matrix) is obtained by means of its classification error over the validation set. In addition, [10] proposed the use of the CHC Genetic Algorithm [11] to optimize a Sparse Random ECOC matrix. In this work, the code length is fixed in the interval $[30, 50]$ independently of the number of classes. Finally, [9] used a Genetic Algorithm to optimize a Sparse Random coding matrix of length in the interval $[\log_2(N), N]$. The evaluation of each individual (ECOC coding matrix) is performed as the classification error over a validation set.

The main trend of previous works was to map the ECOC matrix into a binary vector and then use standard genetic operators to optimize such matrix. However, note that using standard GAs to optimize problems in which individuals have a certain degree of complexity in their representation or are constrained to fulfil some properties (i.e ECOC matrices) raises certain issues. The first one is the uncontrolled generation of non-valid individuals during the optimization process. A clear example in the ECOC framework would be the situation in which a matrix with two equivalent codes is constructed. The second issue is how the optimization process is guided to portions of the search space that optimize the fitness of the individuals. Note that when using standard GAs no heuristic of how binary vectors should recombine, and thus, convergence to a population that performs accurately can be misguided.

With the previous issues in mind, we propose a novel Genetic framework for treating the optimization process of an ECOC matrix. These ECOC matrix is based on the Support Vector Machine (SVM) as the base classifier, since it has shown powerful results in recent literature. In this framework the genetic operators have been carefully redefined in order to avoid non-valid individual generation, and thus, minimize the search space with relation to previous works that used Genetic Algorithms to optimize an ECOC coding matrix. Moreover, special effort has been put in designing operators that smartly guide the optimization process in order to converge in few generations. In addition, the ECOC code length is reduced to be sub-linear in the number of categories, building both reduced and high-performance codes. This novel procedure is tested on a wide set of publicly available datasets obtaining very promising results. Summarizing, our contributions in this paper are the following:

- The **crossover and mutation operators are redefined** taking into account the ECOC properties in order to generate valid individuals.
- A new **operator that is able to extend the ECOC code** length taking into account the idiosyncrasies of the data is developed.
- We introduce a novel **regularization parameter** that is able to control the number of dichotomies produced by each individual, and as consequence, the learning capacity of the ECOC matrix.
- A **complexity and performance analysis** taking into account the various factors that affect the performance measures (number of SVs, number of dichotomies, generations in the optimization procedure, etc.) is performed.

The rest of the paper is organized as follows: Section 2 overviews the background of ECOC and GA applied in the ECOC framework. In Section 3, we present the novel ECOC-Compliant Genetic Algorithm. Section 4 is devoted to present the experimental results. Finally, Section 5 concludes the paper.

2. Background Research

In this section, we overview the background research on ECOC in terms of coding and decoding designs, ECOC properties, ECOC code length and problem-dependent designs, and the use of genetic algorithms within the ECOC framework, motivating the basis for our ECOC-Compliant genetic algorithm presented in next sections. The notation used in the paper is presented in Table 1.

2.1. The ECOC Framework

ECOC is a general multi-class framework built on the basis of the error-correcting principles of communication theory [12]. This framework is composed of two different steps: *coding* [12, 13] and *decoding* [1, 2]. At the coding step an ECOC coding matrix $M_{N\times n} \in \{-1, +1\}$ is constructed, where N denotes the number of classes in the problem and n the number of bi-partitions defined to discriminate the N classes. In this matrix, the rows (also known as *code-words*) are univocally defined, since these are the identifiers of each category in the multi-class categorization problem. On the other hand, the columns of M denote the set of bi-partitions, dichotomies, or meta-classes to be learnt by each base classifier h^j (also known as dichotomizer) $H = \{h^1, \ldots, h^n\}$. In this sense, classifier h^j is responsible of learning the bi-partition denoted on the $j-$th column of M. Therefore, each dichotomizer learns the classes with value $+1$ against the classes with value -1 in a certain column. Note that the ECOC framework is independent of the base classifier applied. For notation purposes in further sections we will refer to the entry of M at the i-th row and the j-th column as $M_{i,j}$. Following this notation the i-th row (codeword of class c^i) will

Table 1: Paper notation

Abbreviation	**Meaning**
AHD	Attenuated Hamming Distance
CD	Critical difference
DECOC	Discriminant Error-Correcting Output Codes
ECOC	Error-Correcting Output Codes
GA	Genetic Algorithms
HD	Hamming Distance
LWD	Loss-Weighted Decoding
RBF	Radial Basis Function
SVM	Support Vector Machines
C	Confusion matrix
c^k	$k-th$ class
cp	Current parent for the ECOC-Compliant crossover algorithm
d^j	$j-th$ dichotomy, ($j-th$ column of the coding matrix)
d_i^j	$i-th$ bit of the $j-th$ dichotomy, (entry of the coding matrix)
E	Error rate
f	Found flag for the ECOC-Compliant crossover algorithm
G	Number of generations of the ECOC-Compliant GA
H	Set of classifiers
h^i	$i-th$ classifier
I	Set of ECOC individuals of the ECOC-Compliant GA
L	Loss function for the Loss-Weighted decoding
M	ECOC coding matrix
mt_c	Mutation control value for the ECOC-Compliant mutation algorithm
M_w	Matrix of weights for the Loss-Weighted decoding
N	Number of classes
n	Number of dichotomies
P	Performance per classifier vector
p^i	Performance of the $i-th$ classifier
R_t	Set of repeated codewords in a matrix $M_{N\times t}$
r_t	Cardinality of R_t
s	Testing sample
sp_c	Sparsity control value for the Sparsity controlled extension algorithm
x^s	Predicted codeword for sample s
y^i	$i-th$ code, ($i-th$ row of the coding matrix)
y_k^i	$k-th$ bit of the $i-th$ code, (entry of the coding matrix)
Y	Set of codewords
δ	Decoding function
μ	Set of selected position for mutation
τ	Selection order for the ECOC-Compliant crossover algorithm

be referred as y^i and, the j-th column (j-th bi-partition or dichotomy) will be referred as d^j.

Originally, the coding matrix was binary valued ($M \in \{-1, +1\}$). However, [13] introduced a third value, and thus, $M_{N \times n} \in \{-1, +1, 0\}$, defining ternary valued coding matrices. In this case, for a given dichotomy categories can be valued as $+1$ or -1 depending on the meta-class they belong to, or 0 if they are ignored by the dichotomizer. This new value allows the inclusion of well-known decomposition techniques into the ECOC framework, such has One vs. One [14] or Sparse [3] decompositions.

At the decoding step a new sample s is classified among the N possible categories. In order to perform the classification task, each dichotomizer in H predicts a binary value for s whether it belongs to one of the bi-partitions defined by the correspondent dichotomy. Once the set of predictions $x(s) \in \mathcal{R}^n$ is obtained, it is compared to the codewords of M using a distance metric δ, known as the decoding function. The usual decoding techniques are based on well-known distance measures such as the Hamming or Euclidean distances. This measures were proven to be effective in binary valued ECOC matrices $\{+1, -1\}$. Nevertheless, it was not until the work of [1] that decoding functions took into account the meaning of the 0 value at the decoding step. Generally, the final prediction for s is given by the class c^i, where $\underset{i}{\operatorname{argmin}}\ \delta(y^i, x(s))$, $i \in \{1, \ldots, N\}$. An example of coding and decoding steps for a 5-class toy problem is shown in Figure 1.

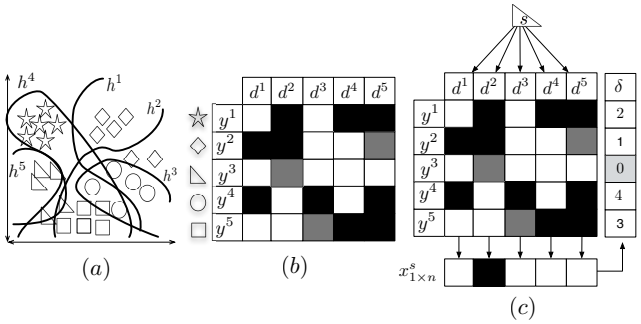


Figure 1: (a) Feature space and trained boundaries of dichotomizers. (b) Coding matrix M, where black and white cells correspond to $\{-1, +1\}$, denoting the two partitions to be learnt by each base classifier (white cells vs. black cells) while grey cells correspond to 0 (ignored classes). (c) Decoding step, where the predictions of classifiers, $\{h^1, \ldots, h^5\}$ for sample s are compared to the codewords $\{y^1, \ldots, y^5\}$ and s is labelled as the class codeword at minimum distance.

2.2. ECOC Coding Matrix Properties

[12] defined the properties of a valid ECOC coding matrix. These properties concern the repetitions of rows in M. Since a repetition of rows will define two different categories with the same codeword, an ambiguous coding matrix

M will be constructed. Moreover, error-correcting principles are based on the assumption that errors introduced by each dichotomizer are uncorrelated [12]. In this sense, similar dichotomies will output correlated outputs, and thus, this situation has to be avoided. In the limit case, equivalent dichotomies will have equivalent outputs, and thus, no correction capability will be added. However, this is more a suggestion than a constraint of the ECOC coding matrix, since breaking it does not drastically affect the generalization capability but introduces redundant computation. Nevertheless, when aiming for sound designs all these suggestions must be taken into account. Therefore, we define an ECOC coding matrix $M_{N\times n} \in \{-1, +1, 0\}$ to be constrained by:

$$\min(\delta_{AHD}(y^i, y^k)) \geq 1,\ \forall i, k : i \neq k,\ i, k \in [1, \ldots, N], \tag{1}$$

$$\min(\delta_{HD}(d^j, d^l)) \geq 1,\ \forall j, l : j \neq l,\ j, l \in [1, \ldots, n], \tag{2}$$

$$\min(\delta_{HD}(d^j, -1 \cdot d^l)) \geq 1,\ \forall j, l : j \neq l,\ j, l \in [1, \ldots, n], \tag{3}$$

where δ_{AHD} and δ_{HD} are the Attenuated Hamming Distance (AHD) and the Hamming Distance (HD), respectively, defined as follows:

$$\delta_{AHD}(y^i, y^j) = \sum_{k=1}^{n} |y_k^j||y_k^i| I(y_k^i, y_k^j), \tag{4}$$

$$\delta_{HD}(y^i, y^j) = \sum_{k=1}^{n} I(y_k^j, y_k^i),\ I(i, j) \begin{cases} 1 \text{ if } i \neq j \\ 0 \text{ otherwise.} \end{cases} \tag{5}$$

The motivation of using AHD to measure the distance between rows and HD to measure distance between columns is motivated by the different influences of the value 0 in columns and rows of M. Thus, a position valued as 0 in a codeword means that a certain dichotomy is not taken into account in the definition of the class code, while for a dichotomy a position d_j^i valued as 0 means that class c^j is ignored in the training step.

Equation 1 defines the minimum AHD between the codewords of M to be greater or equal than one. In other words, there can not exist two identical codewords. In addition, Equations 2 and 3 refer to the column suggestion. Note that, when interchanging all values $+1$ and -1 of a column the same meta-classes are defined. In addition, note that maximizing Equation 1 implies an increment of the error correction capabilities of M. Analogously, the maximization of Equations 2 and 3 implies an increment of dichotomy diversity, and thus, a reduction of the bias and variance errors [5].

2.3. Towards Reducing the ECOC Code Length

The coding step of the ECOC framework has been widely studied in literature, obtaining either predefined [15, 14], random [13] or problem-dependent [6, 16, 17, 18, 19] coding designs. The most well-known coding schemes are the predefined ones, such as, One vs. One [14] and One vs. All [15] designs, in which $\binom{N}{2}$ and N dichotomies are defined, respectively. In the One vs. One scheme all the possible pair-wise groups of the N categories are defined, while

in the One vs. All scheme each dichotomy is responsible of discriminating one class against the rest of the classes. In contra-position, some works in literature have stated that random designs [3], with code length of $\{10, 15\} \cdot \log_2(N)$ can perform as well as predefined codes. Note that, predefined or random designs do not exploit the problem-domain information.

Some works in literature have proposed the use of problem-dependent ECOC coding matrices. Problem-dependent coding designs lay on the assumption that predefined and random codes may not be suitable to solve a given problem since they do not take into account the underlying distribution of the classes in the multi-class problem. In [6] the authors proposed the DECOC coding design of length $N-1$ in which a tree structure is embedded in the ECOC coding matrix, where nodes correspond to classes that maximize a split criterion. In the trend of the previous works, [16] proposed the same tree embedding where the nodes correspond to the most difficult meta-classes to be learnt. Other works aim to treat the problem either by soft weight sharing methods [20] or by using EM algorithm to find the optimal decomposition of the multi-class problem [21]. In [15] Rifkin et. al stated that when using high capacity dichotomizers the code length can be reduced with no loss of generalization, and test their hypothesis in the One vs. All coding design of N dichotomies. Nevertheless, few are the works that aim to reduce the code length by using problem-dependent designs [6].

Recently, [8] proposed the use of a Minimal ECOC coding matrix of length $\lceil \log_2(N) \rceil$, where $\lceil \cdot \rceil$ rounds to the upper integer. This coding matrix M is the theoretical lower-bound in terms of the numbers of classifiers. The authors showed that if this coding matrix M is properly tuned and the dichotomizers are high capacity classifiers, it can be used with no loss of generalization capability when compared to most state-of-the-art approaches.

In general, classification performance has always been the core of all ECOC evaluation, regardless of its length. Nevertheless, following the Occam's razor principle, in equal conditions, simpler models tend to be more suitable. In this sense, we can consider that in the ECOC framework the number of classifiers has a direct relationship to the complexity of the model. For instance, when using SVM as the base classifier, the number of classifiers has a direct relationship to the overall number of Support Vectors (SVs) of the ECOC matrix. At the same time, the number of SVs is directly proportional to the complexity in the ECOC decoding step. Thus, a trade off between generalization performance and code length has to be taken into account in order to perform a fair analysis of ECOC capabilities.

In Figure 2 we show the number of classifiers defined for some of the state-of-the-art coding designs with respect to the number of classes of the multi-class problem. The coding designs taken into account are the One vs. One, One vs. All, Sparse and Dense Random, DECOC and Minimal ECOC [14, 15, 3, 6, 8].

Note the great difference between the number of dichotomies defined by state-of-the-art strategies. In this case we can see that the Minimal ECOC approach defines the most reduced code length in contra-position with the One vs. All and One vs. One strategies, which have a linear and quadratic growth

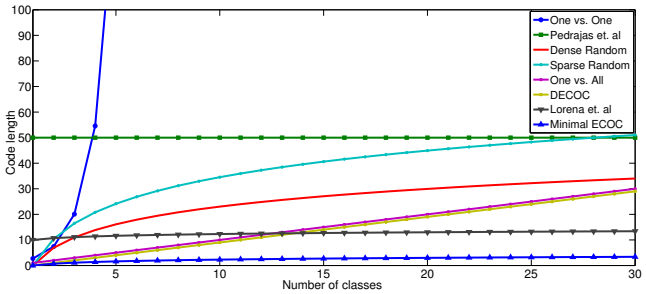


Figure 2: Number of classifiers per coding design and number of classes.

with the number of classes, respectively. This fact encourages the use of sub-linear ECOC strategies (with respect to the number of classifiers used), since the scalability problem that is present when using other strategies can be easier to tackle.

2.4. Genetic Algorithms in the ECOC Framework

Genetic Algorithms are stochastic optimization processes based on Darwin's evolution theory. These processes start with a set of individuals which represent a set of random points in the search space. Each individual has a fitness value, which denotes the adaptation to the environment and in most cases it is the value to optimize. Individuals are transformed through crossover and mutation operators along generations, improving their adaptation to the environment. Commonly, the crossover operator guides the optimization process to parts of a search space which optimize the fitness value. On the other hand, the mutation operator is responsible of not letting the algorithm converge to local minima. Recent literature on applying GAs to the ECOC framework is summarized in the following paragraphs.

- **Minimal Design of Error-Correcting Output Codes [8]:** In this work the authors propose a standard GA to optimize an ECOC coding matrix $M_{N\times n} \in \{-1, +1\}$, where $n = \lceil log_2(N) \rceil$. In addition, the evaluation of each individual is obtained by means of its classification error over the validation set. In this work, the scattered crossover operator is used. Mutation is implemented using a Gaussian distortion over the selected gene. The achieved results are comparable with most of the state-of-the-art ECOC approaches with a reduced code length.

- **Evolving Output Codes for Multiclass Problems [10]:** In addition, [10] proposed the use of the **C**ross Generational Elitist election, **H**eterogeneous Recombination and **C**ataclysmic Mutation (CHC) Genetic Algorithm [11] to optimize a Sparse Random ECOC matrix [3] $M_{N\times n} \in \{-1, +1, 0\}$ where $n \in [30, 50]$. In this case, the code length is fixed in the interval $[30, 50]$ independently of the number of classes, which seems

to be a non well founded approach since it disagrees with a large body of literature [14, 15, 3, 6, 8]. It is interesting to note that the evaluation of individuals is the aggregation of different aspects of the ECOC coding matrix such as distance between rows and columns or dichotomizer performances.

- **Evolutionary Design of Multiclass Support Vector Machines [9]:** In this work the authors propose a Genetic Algorithm to optimize a Sparse Random [3] coding matrix $M_{N\times n} \in \{-1, +1, 0\}$, where $n \in [\log_2(N), N]$. The evaluation of each individual (ECOC coding matrix) is performed using the classification error over the validation sets. The crossover operator considered is the exchange of a set of $k \in [1, n_{ex} < n]$ dichotomies between individuals. The considered mutation operator has four variants which depend on how values in the coding matrix are changed. The most interesting point is that operator variants are chosen based on an historic of its performance in previous generations.

Despite these works, the problem of optimizing an ECOC coding matrix M present some issues that must be carefully revised.

The first one is the uncontrolled generation of non-valid individuals (see Equations 1, 2, and 3). This issue has been treated in state-of-the-art works either by automatically setting the fitness value of a non-valid individuals to be lower than the worst valid individual value, and thus, letting the algorithm converge to valid solutions, or by simply rejecting non-valid individuals. Definitely, both are valid options used in most evolutionary frameworks. Nevertheless, it is easy to see that when tackling the problem of minimizing the search space of ECOC solutions, the mentioned approximation is inappropriate. The generation of new individuals along the optimization process is performed by crossover and mutation operators. Therefore, one may argue that operators used by state-of-the-art approaches are not suitable for the problem of optimizing an ECOC coding matrix M, and thus, they have to be redefined.

The second issue is how the optimization process is guided to parts of the search space that optimize the fitness of the individuals. In this sense, not only the constraints of individuals have to be taken into account when designing crossover and mutation operators but also how these individuals improve their adaptation through those operators, allowing the process to converge in fewer generations. On the other hand, designing operators that dramatically reduce the stochastic search may imply premature convergence to local minima.

When making use of optimization processes in the ECOC framework the first step to perform is the estimation of the ECOC search space cardinality. Assume an N-class problem to be treated, then the ECOC framework will construct a matrix $M_{N\times n}$ in which N codewords will be chosen from the 3^n codes available. Following Newton's binomial this could be expressed as $\binom{3^n}{N}$. Nevertheless, taking into account the constraints defined in Equations 2 and 3 a matrix M and its opposite (swapping all 1 by -1 and vice-versa) are equivalent since they define the exact same partitions of the data. In this sense, the number of possible ECOC coding matrices is shown in Equation 6.

$$\#M = \frac{\binom{3^n}{N}}{2} = \frac{3^n!}{2 \cdot N! \cdot (3^n - N)!} \tag{6}$$

In addition to the huge cardinality of the search space, [7] showed that the computation of an optimum ECOC matrix given the set of all possible dichotomizers H is *NP-Complete*. Finally, this search space is non-continuous since a change in a single position of the ECOC matrix M can break the ECOC matrix constraints.

In the next section we describe the method which is able to deal with all these issues by taking into account the properties of the ECOC framework within the definition of the GA.

3. ECOC-Compliant Genetic Algorithm

In this section the novel ECOC-Compliant Genetic Optimization is presented. In order to provide a complete description of the method in Figure 3 we show a scheme of the procedure.

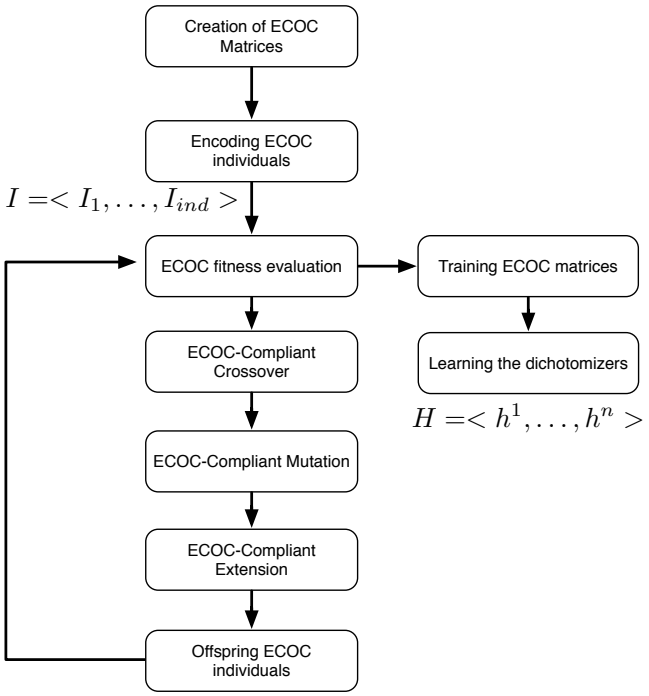


Figure 3: Diagram of the ECOC-Compliant Genetic Algortihm.

3.0.1. Problem Encoding

Since our proposal redefines the standard crossover and mutation operators there is no need to be tied to standard encoding schemes. In this sense, our individuals are encoded as structures $I_{ECOC} =< M, C, H, P, E, \delta >$, where the fields are defined as follows.

- The **coding matrix**, $M_{N\times n} \in \{-1, +1, 0\}$ where $n \geq \lceil \log_2 N \rceil$. Note that for the initial population $n = \lceil \log_2 N \rceil$, where n can grow along generations.
- The **confusion matrix**, $C_{N\times N}$, over the validation subset. Let c^i and c^j be two classes of our problem, then the entry of C at the i-th row and the j-th column, defined as $C_{i,j}$, contains the number of examples of class c^i classified as examples of class c^j.
- The **set of dichotomizers**, $H =< h^1, \ldots, h^n >$.
- The **performance of each dichotomizer**, $P \in \mathcal{R}^n$. This vector contains the proportion of correctly classified examples over the validation subset for each dichotomizer in H. Note that this measure is not the performance of the overall multi-class problem but the one of the dichotomizer over the meta-classes defined by the correspondent dichotomy.
- The **error rate**, E, over the validation subset. This scalar is the proportion of incorrectly classified examples in the multi-class problem over the validation subset. Let the set of samples in the validation subset be $V =< (s_1, l(s_1)), \ldots, (s_v, l(s_v)) >$, then the calculus is shown in Equations 7 and 8.

$$E = \frac{\sum_{j=1}^{v} \sigma(\Delta(M, x^{s_j}), l(s_j))}{v}, \ \sigma(i,j) \begin{cases} 1 \text{ if } i = j \\ 0 \text{ otherwise,} \end{cases} \tag{7}$$

$$\Delta(M, x) = \underset{i}{\operatorname{argmin}}\ \delta(y_i, x), \ i \in \{1, \ldots, N\}. \tag{8}$$

- The **decoding function**, δ. We use the Loss-Weighted decoding [1] of Equation 9, where M_w is a matrix of weights and L is a loss function ($L(\theta) = \exp^{-\theta}$ in our case).

$$\delta_{LW}(x(s), i) = \sum_{j=1}^{n} M_w(i,j) L(y_j^i \cdot I((x(s)), j)) \tag{9}$$

3.0.2. Fitness Function

The fitness function measures the environmental adaptation of each individual, and thus, is the one to be optimized. In this sense, the most common approach in state-of-the-art literature has been to evaluate an ECOC individual

as the performance it obtains on the validation subset. Nevertheless, following Occam's razor principles, in the hypothetical situation in which two individuals obtain the same performance on the field the one showing a simpler model (which very often implies a smaller code length) tends to be the most suitable choice.

This general assumption can be redefined and used in the fitness function in order to penalize those individuals with a large code length. Let $I =< I_1, \ldots, I_{ind} >$ be a a set of individuals and I_k a ECOC individual encoded as shown in Section 3.0.1, then our fitness function is defined as shown in Equation 10.

$$F_f(I_k) = E_{I_k} + \lambda n_{I_k} \tag{10}$$

This expression (similar to the one showed by regularized classifiers), serve us to control the learning capacity and avoid over-fitting.

There exists several ways of defining complexity in the ECOC framework. Nevertheless, the code length has been always in the core of this definition. Thus, we have adopted the term complexity as the number of dichotomies defined in the coding matrix M, that is n.

3.0.3. ECOC-Compliant Crossover and Mutation Operators

In this section we introduce the novel ECOC-compliant crossover and mutation operators. These operators do not only take into account the restrictions of the ECOC framework (shown in Equations 1, 2 and 3) but also are carefully designed in order to avoid a premature convergence to local minima without generating non-valid individuals, and thus, converging to satisfying results in fewer generations.

In our proposal, when performing the genetic optimization we have to take into account the effect of the operators used. In many Genetic Algorithms a trade-off between exploring a satisfying portion of the search space and converging quickly to a final population is desirable [22], thus avoiding the convergence to local minima. To achieve this goal in our proposal two versions of each operator where designed, the generic and the specific. On one hand, the generic version of each operator builds valid individuals with a random seed, which aims to accomplish the exploration of a satisfying portion of the search space. On the other hand, the specific version takes into account certain factors (i.e dichotomizer performances, confusion matrices, etc.) that imply the guidance of the optimization procedure to promising regions of the search space, where individuals may obtain a better fitness value.

• **ECOC-Compliant Crossover Algorithm**

In GAs one of the most important issues is how individuals are recombined in order to produce a more adapted offspring. In this sense, one would like to find a smart recombination method (known as crossover operator) that take profit of problem domain information in order to allow a faster convergence. These recombinations are completely defined for standard encodings schemes

(such as binary encoding) and have been deeply studied in literature. Nevertheless, when facing problems in which individuals are constrained and standard encoding designs have to be redefined, we consider also the redefinition of the recombination procedure to be an unavoidable task. This consideration is given by the fact that using standard GAs in problems where individuals are constrained can lead to situations where the search space is enlarged due to the generation of non-valid individuals.

Picture a N-class problem and let I_F and I_M be two individuals encoded as shown in Section 3.0.1. Then the crossover algorithm will generate a new individual I_S which coding matrix $M^{I_S}_{N\times n},\ n = \min(M^{I_F}_{N\times n}, M^{I_M}_{N\times n})$ contains dichotomies of each parent. Therefore, the key aspect of this recombination is the selection of which dichotomies of each parent are more suitable to be combined. Taking into account the aim of avoiding the generation of non-valid individuals, we introduce a dichotomy selection algorithm that chooses those n dichotomies that fulfil the constraints shown in Equations 1, 2, and 3.

The dichotomy selection algorithm generates a dichotomy selection order $\tau \in \mathcal{R}^n$ for each parent, where τ^I is the selection order of parent I and τ^I_k is the value at the k-th position. However, this selection order might lead to a situation in which the n dichotomies chosen define an incongruence in the coding matrix, such as defining two classes with the same codeword. In such case, the dichotomy election algorithm checks if the separation between codewords is congruent with the number of dichotomies left to add.

Theorem 3.1 describes the number of equivalent codes allowed to appear on a matrix that is being built to fulfil the ECOC properties in terms of rows (Equation 1). In this sense, when the final length of the ECOC matrix in terms of columns is known, we can determine the maximum number of equivalent codes allowed to appear when each extension dichotomy is appended to build the ECOC matrix.

Theorem 3.1. *Should $M_{N\times t} \in \{-1, +1, 0\}$ be a randomly distributed matrix. Then, the extension of $M_{N\times t}$ to an ECOC coding matrix $M_{N\times(t+k)}$ with N unequivocally defined rows, will be possible if and only if when including the i-th ($0 \leq i \leq k$) extension dichotomy in M, $2^{(k-i)}$ repeated codewords of length $t+i$ are obtained at most.*

Proof Let a matrix $M_{N\times t}$ define R_t repeated codes (two codes y^a, y^b are equivalent if $\delta_{AHD}(y^a, y^b) = 0$). Assume $M_{N\times t}$ is to be extended to $M_{N\times(t+k)}$ so that it fulfils Equation 1: $\min(\delta_{AHD}(y^i, y^k)) \geq 1,\ \forall i, k : i \neq k,\ i, k \in [1, \ldots, N]$.

Then, from Information theory $\lceil \log_2(R_t) \rceil$ is known to be the minimal number of extension bits needed to unequivocally split R_t codes. Therefore, if $M_{N\times t}$ is extended with k dichotomies, then $\lceil \log_2(R_t) \rceil \leq k$ dichotomies are needed to assure that Equation 1 holds. When the first of the k dichotomies is added, then $k-1$ dichotomies will be used to split the remaining set of repeated codes (R_{t+1}). As in the former case, $\lceil \log_2(R_{t+1}) \rceil \leq k-1$ are needed. Accordingly, when the second dichotomy is appended $\lceil \log_2(R_{t+2}) \rceil \leq k-2$. Generalizing, $M_{N\times t}$ will only be extendible to a valid ECOC matrix $M_{N\times t+k}$ if when adding

the i-th dichotomy $\lceil \log_2(R_{t+i}) \rceil \leq k - i$. Thus, 2^{k-i} repeated codewords are obtained at most when adding the i-th extension dichotomy.

Following Theorem 3.1 the i-th dichotomy will be only added if it splits the existing codewords to define $R_{t+i} \leq 2^{(k-i)}$ different codes. However, in a certain iteration it may happen that there are no existing dichotomies in both parents that accomplish the split criteria. In this situation, a new dichotomy is generated in order to ensure the ECOC properties. We define the ECOC-compliant crossover algorithm as shown in Algorithm 1.

```
Data: I_F, I_M
Result: I_S
n := min(M^{I_F}, M^{I_M}) // Minimum code length among parents
τ^{I_F} ∈ ℝ^n = selorder(I_F) // Dichotomy selection order of I_F
τ^{I_M} ∈ ℝ^n = selorder(I_M);
cp := I_F // Current parent to be used
M^{I_S} := ∅ // Coding matrix of the offspring
for i ∈ {1, ..., n} do
    for j ∈ {1, ..., n_cp} : τ_j^{cp} ≠ ∅ do
        f := 0 // Valid dichotomy search flag
        if calcRepetitions (M^{I_S}, d^{τ_j^{cp}}) ≤ 2^{(k-i)} then
            d^i := d^{τ_j^{cp}} // Inheritance of dichotomies
            h^i := h^{τ_j^{cp}} // Inheritance of dichotomizer
            p^i := p^{τ_j^{cp}} // Inheritance of performance
            τ_j^{cp} := ∅ // Avoid using a dichotomy twice
            f := 1 // Valid dichotomy found
            break;
        end
    end
    if !f then
        d^i := generateCol(M^{I_S}) // If non ECOC matrix can be built
        h^i := ∅;
        p^i := ∅;
    end
    if cp = I_F then
        cp := I_M // Dichotomy inheritance parent switch
    else
        cp := I_F;
    end
end
```

Algorithm 1: ECOC Crossover Algorithm.

The ECOC crossover algorithm variants have an equal probability of being executed. In the first one, which is the generic version, the dichotomy selection order is randomly generated, and thus, it generates a random ECOC individual that ensures to fulfil Equations 1,2, and 3. In the second one, the specific version, the dichotomy selection order is based on dichotomizer performance, and thus, dichotomizers that show a higher performance have more chances of being selected. These two variants of crossover provide us a trade-off between covering an enough portion of the search space and guiding the optimization process to a population with minimal values of the fitness function. An example of the ECOC-compliant crossover operator is shown in Figure 4.

In the crossover example shown in Figure 4 two individuals I_M and I_F are

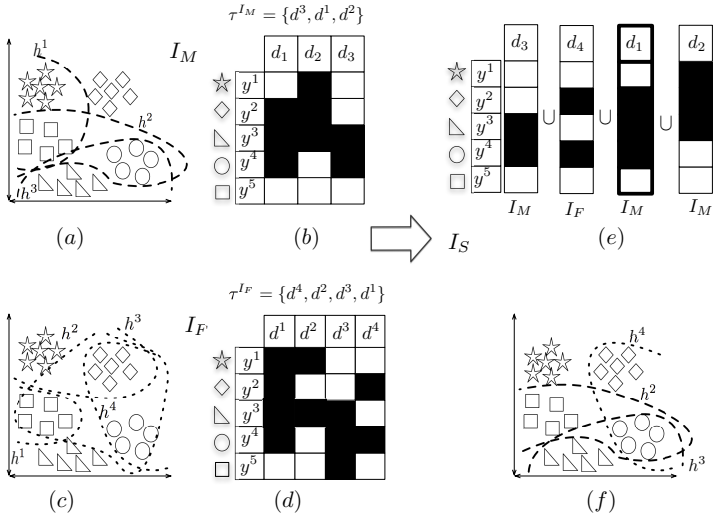


Figure 4: Crossover example for a 5-class toy problem. (a) Feature space and trained classifiers for parent I_M. (b) ECOC coding matrix of parent I_M. (c) Feature representation and boundaries for parent I_F. (d) Coding matrix of I_F. (e) ECOC coding matrix composition steps for the offspring I_S. (f) Feature space and inherited classifiers for I_S.

combined to produce a new offspring I_S[1]. The crossover algorithm generates a dichotomy selection order τ for each parent. The first parent from which a dichotomy is taken is I_M, and d_3 is valid since $r \leq 2^{(3-1)} = 4$, and it only defines three codes without separation (y^1, y^2, and y^5). Once this step is performed, the parent is changed, and the following dichotomy will be extracted from I_F based on its selection order τ^{I_F}. In this case, d_4 is valid since $r \leq 2^{(3-2)} = 2$ and d_3 of I_M together with d_4 of I_F define only two equivalent codewords (y^1 and y^5). In the following iteration, the parent is changed again, and thus, I_M is used. Following τ^{I_M} the dichotomy to use is d_1, but if we apply Theorem 3.1 we find that $r \leq 2^{(3-3)} = 1$, and thus, d_1 is unuseful. Since $\delta_{AHD}(y^1, y^5) = 0$, d^1 can not be considered as an extension dichotomy, and therefore, the next dichotomy to use is d_2, which satisfies Equation 1 defining a valid ECOC coding matrix.

• ECOC-Compliant Mutation Algorithm

Historically, mutation operators have been responsible of not letting the algorithm converge to local minima. In literature, these operators have been defined for standard encoding designs (such as binary encoding). Nevertheless, when individuals are not encoded following standard schemes these operators have to be redefined in order to completely fulfil their purpose.

Picture an individual I encoded as shown in Section 3.0.1 to be transformed by means of the mutation operator. This operator will select a set of positions $\mu =< M_{i,j}, \ldots, M_{k,l} >, i, k \in \{1, \ldots, N\}, j, l \in \{1, \ldots, n\}$ of M^I to be mutated.

[1]Note that for applying Theorem 3.1 in this example we consider $k = 3$.

The value of these positions is changed constrained to the set $\{-1, +1, 0\}$. Two variants of this algorithm are implemented depending on how the positions in μ are chosen and how the matrix is recoded. The first is defined as the Generic ECOC mutation algorithm shown in Algorithm 2. In this version the set of positions μ is randomly chosen. Once μ is defined, the positions are randomly recoded to one of the three possible values in $\{-1, +1, 0\}$. However, note that the mutation of values may lead to a situation in which the matrix M does not fulfil the ECOC constraints. To avoid this effect, we check the ECOC matrix at each bit mutation in order to ensure that a valid ECOC individual is generated, if a certain bit mutation generates a non-valid individual this particular bit mutation is obviated.

Data: I_T, mt_c
// Individual and mutation control value
Result: I_X
$\mu :=< M_{i,j}, \ldots, M_{k,l} >,\ i, k \in \{1, \ldots, N\},\ j, l \in \{1, \ldots, n\}$;
$\mu = getRandomPositions(M^{I_T}, mt_c)$ // Select the position in the M^{I_T} for mutation
;
for $M_{i,j} \in \mu$ **do**
  switch $M_{i,j}$ **do**
    // If the value selected for mutation is 0 it might turn +1 or −1
    case $M_{i,j} = 0$
      $r =$*Random(0,1)*; **if** $r > 0.5$ **then**
        $M_{i,j} := +1$;
      else
        $M_{i,j} := -1$;
      end
    endsw
    case $M_{i,j} = -1$
      $r =$*Random(0,1)*// Obtain a random value in $[0, 1]$
      if $r > 0.5$ **then**
        // Equiprobablity of selecting the remaning values
        $M_{i,j} := +1$;
      else
        $M_{i,j} := 0$;
      end
    endsw
    case $M_{i,j} = +1$
      $r =$*Random(0,1)*;
      if $r > 0.5$ **then**
        $M_{i,j} := 0$;
      else
        $M_{i,j} := -1$;
      end
    endsw
  endsw
end
$M^{I_X} = M$

Algorithm 2: Generic ECOC-Compliant Mutation Algorithm.

In the second, defined as the Specific ECOC-Compliant mutation algorithm, the set of positions μ is chosen taking into account the confusion matrix C. In this sense, the mutation algorithm will iteratively look for the most confused categories in the confusion matrix $(c^i, c^j) = \underset{i,j}{\text{argmax}}(C_{i,j} + C_{j,i})$. Once these classes are obtained, the algorithm will transform the bits valued 0 of those classes codewords $\{y^i, y^j\}$ in order to increment the distance $\delta_{AHD}(y_i, y_j)$, and

thus, increasing their correction capability, while keeping a valid ECOC matrix. The specific ECOC mutation algorithm is shown in Algorithm 3.

Data: I_T, mt_c
// Individual and mutation control value
Result: I_X
$C^{I_T}_{N\times N}$ // Confusion matrix of I_T
$k := 0$// Number of recoded bits of M^{I_T}
while $k < mt_c$ **do**
  $(c^i, c^j) := \underset{i,j}{\text{argmax}}(C_{i,j} + C_{j,i})\ \forall i, j : i \neq j;$
  for $b \in \{1, \ldots, n\}$ **do**
    if $|y^i_b| + |y^j_b| \leq 1$ **and** $k < mt_c$ **then**
      if $y^i_b = 0$ **and** $y^j_b = 0$ **then**
        $y^i_b := +1$ // Invert both bits valued 0
        $y^j_b := -1;$
      else
        if $y^i_b = 0$ **then**
          $y^i_b := -y^j_b$ // Invert bit valued 0
        else
          $y^j_b := -y^i_b;$
        end
      end
      $k := k + 1;$
    end
  end
  $C^{I_T}_{i,j} := 0,\ C^{I_T}_{j,i} := 0;$
end

Algorithm 3: Specific ECOC-Compliant Mutation Algorithm.

In Figure 5 an example of the specific mutation algorithm is shown. Let I_T be an individual encoded as shown in Section 3.0.1. The confusion matrix C_{I_T} has its non-diagonal maximum at $C_{4,3} + C_{3,4}$. Then codewords y^4 and y^3 are going to be mutated. The 0 valued bits of this codewords are changed in order to increment $\delta_{AHD}(y^4, y^3)$, and thus, incrementing also the correction capability between them. At the following iteration $C_{4,3}$ is not taken into consideration and the procedure will be repeated with y^5 and y^4 which are the following classes that show confusion in C.

3.0.4. Problem-Dependent Extension of ECOCs

In related works that used a GA to optimize the ECOC matrices the length was fixed in a certain interval and the crossover operators where the ones responsible for obtaining reduced or large codes [9, 10]. Nevertheless, we consider that from the ECOC point of view the length of the code is a crucial factor that has to be separately addressed, since the length of the code matrix has a direct relationship to its correction capability. In this sense, as stated in Section 3.0.1 our initial population is based on the coding scheme proposed by [8], that is, the use of a Minimal ECOC coding matrix $M_{N\times\lceil\log_2(N)\rceil}$. Nevertheless, when analyzing the Minimal ECOC matrix a lost of correction capability is found. Let M be a ECOC coding matrix, then:

$$\rho = \min\left(\frac{\delta(y^i, y^j) - 1}{2}\right),\ \forall i, j \in \{1, \ldots, N\},\ i \neq j. \tag{11}$$

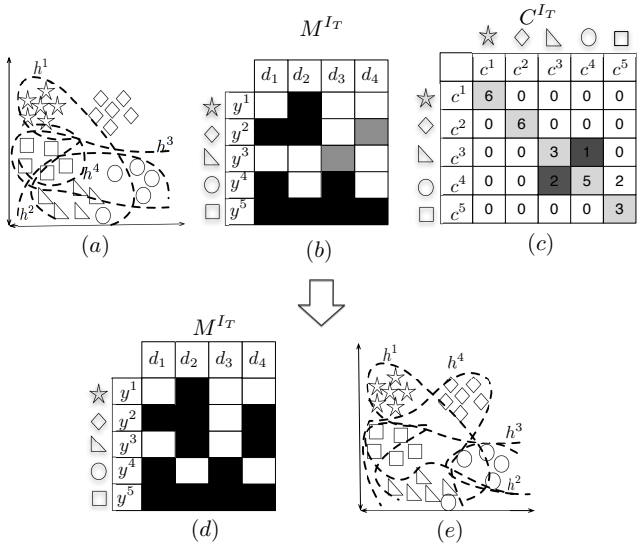


Figure 5: Mutation example for a 5-class toy problem. (a) Feature space and trained dichotomizers for and individual I_T. (b) ECOC coding matrix of I_T. (c) Confusion matrix of I_T. (d) Mutated coding matrix. (e) Mutated feature space with trained dichotomizers.

Therefore, we obtain a null correction capability $\rho = 0$ for the Minimal ECOC design, since for this ECOC matrices:

$$\min(\delta_{AHD}(y^i, y^k)_{\forall i,k:i\neq k}) = 1,\ i,k \in [1,\ldots,N]. \tag{12}$$

This means that in Minimal ECOC coding schemes, a sample s will be misclassified if just a dichotomizer $h^i \in H$ misses its prediction. Although this coding design has proved to be fairly effective when its properly tuned, we believe that an extension of such is needed to properly benefit from Error-Correcting principles. However, this extension is not only motivated by the null correction capability issue. The confusion between categories is also a determinant factor when extending ECOC designs [23], since one would like to focus dichotomies in those categories which are more difficult to be learnt.

We propose a novel methodology to extend ECOC designs based on the confusion matrix, aiming to focus the extension dichotomies in those categories which are more difficult to be learnt, and thus, show a greater confusion. This methodology defines to types of extensions, the One vs. One extension and the Sparse extension, which have the same probability of being executed along the optimization process. In the former, the ECOC coding matrix $M_{N\times n}$ will be extended with a dichotomy d^{n+1} which will have 0 values except for those two positions d^i and d^j which correspond to the categories $c^i,\ c^j$ that maximize the confusion $(c^i, c^j) = \underset{i,j}{\text{argmax}}(C_{i,j} + C_{j,i})$. In the second, the Sparsity Controlled Extension shown in Algorithm 4 follows the scheme in which two categories

(c^i, c^j) that maximize the confusion are discriminated. Nevertheless, as high effort to obtain both reduced and powerful codes is performed, one may want to extend M controlling the sparsity of d^{n+1}. Hence, generating a dichotomy that is focused on certain categories but also increments the correction capabilities of M. Picture the case in which a dichotomy $d_k^{n+1} = 0,\ \forall k \in \{1, \ldots, n\} \setminus \{i, j\}$. The Sparse extension algorithm will set d_k^{n+1} to $\{-1, +1\}$, based the confusion of class c^k with c^i and c^j. In this case, a d_k^{n+1} will be valued $\upsilon \in \{-1, +1\}$ if the category of lowest confusion between $\{c^i, c^j\}$ is valued υ. An example of Sparsity extension procedure is shown in Figure 6.

Data: I_T, sp_c
// Individual and sparsity control value
Result: I_X
$C^{I_T}_{N\times N}$ // Confusion matrix of I_T
$(c^i, c^j) := \underset{i,j}{\operatorname{argmax}}(C_{i,j} + C_{j,i})\ \forall i, j : i \neq j;$
$k := 0$ // Recoded bit counter of M^{I_T}
$d_i^{n+1} = \omega$ // Where $\omega \in \{+1, -1\}$
$d_j^{n+1} = -\omega;$
for $b \in \{1, \ldots, N_{I_T}\} \setminus \{i, j\} : \underset{b}{\operatorname{argmin}}(C_{b,i} + C_{i,b} + C_{b,j} + C_{j,b})$ **and** $k < sp_c$ **do**
  if $C_{b,i} > C_{b,j}$ **and** $d_j^{n+1} = \omega$ **then**
    // Give an inverse value to the bit of the class which is most confused with c^i or c^j
    $d_b^{n+1} = \omega;$
  else
    $d_b^{n+1} = -\omega;$
  end
  $k = k + 1;$
end

Algorithm 4: Sparsity Controlled Extension Algorithm.

3.1. Implementation Details

[15] stated that if dichotomizers are high capacity classifiers and are properly tuned, the code length can be reduced to obtain simpler models. Following this idea, we adopted Support Vector Machines with a Gaussian RBF Kernel (SVM-RBF) as our dichotomizer, since it proved to be a very powerful classifier in literature. Typically, training a SVM implies the selection of certain data points (Support Vectors) to build the boundaries. In the specific case of the SVM-RBF two parameters have to be tuned in order to reach for good performances. This parameters are the regularization parameter C and the kernel parameter γ, which are closely related to the data distribution. In literature, the main approach to choose these parameters is the use of cross-validation processes to find the best $\{C, \gamma\}$ pair over a discretization of the parameter space. However, some works have shown that GA's can be applied to this problem, since it can be seen as an optimization process [8, 24]. In this sense, we use a GA to determine the value of the $\{C, \gamma\}$ pair for every dichotomizer in H.

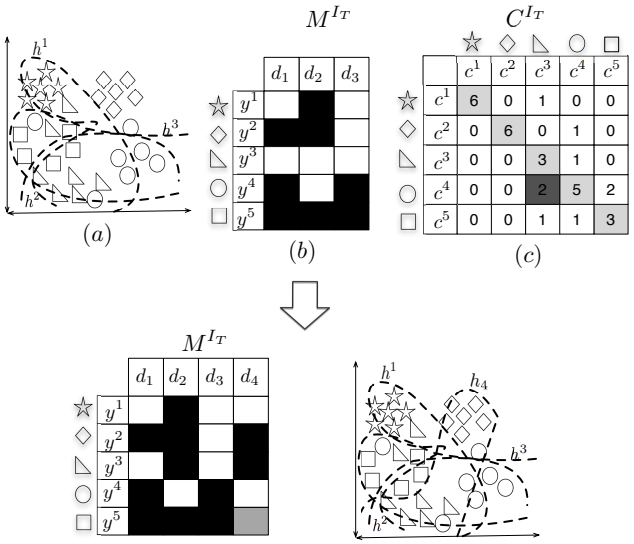


Figure 6: Sparsity extension example for a 5-class toy problem. (a) Feature space and trained dichotomizers for I_T. (b) ECOC coding matrix of I_T. (c) Confusion matrix of I_T. (d) Extended coding matrix. (e) Extended feature space with trained dichotomizers.

4. Experimental Results

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

4.1. Data

The first data used for the experiments consists of nine multi-class datasets from the UCI Machine Learning Repository database [25]. The number of training samples, features, and classes per dataset are shown in Table 2.

Moreover, we apply the classification methodology in four challenging computer vision categorization problems. [2]

- **Traffic sign categorization:** For this computer vision experiment, we use the video sequences obtained from the Mobile Mapping System of [26] to test the ECOC methodology on a real traffic sign categorization problem. In this system, the position and orientation of the different traffic signs are measured with video cameras fixed on a moving vehicle. The system has a stereo pair of calibrated cameras, which are synchronized with

[2] First, we use the video sequences obtained from a Mobile Mapping System [26] to test the methods in a real traffic sign categorization problem consisting of 36 traffic sign classes. Second, 20 classes from the ARFaces [27] dataset are classified using the present methodology. Third, we classify seven symbols from old scanned music scores, and finally, we classify the 70 visual object categories from the public MPEG7 dataset [28]. These datasets are public upon request to the authors.

Table 2: UCI repository datasets characteristics.

Problem	#Training samples	#Features	#Classes
Vowel	990	10	11
Yeast	1484	8	10
Ecoli	336	8	8
Glass	214	9	7
Segmentation	2310	19	7
Dermathology	366	34	6
Shuttle	14500	9	7
Vehicle	846	18	3
Satimage	4435	36	6

a GPS/INS system. The result of the acquisition step is a set of stereo-pairs of images with their position and orientation information. From this system, a set of 36 circular and triangular traffic sign classes are obtained. Some categories from this data set are shown in Figure 7. The dataset contains a total of 3481 samples of size 32×32, filtered using the Weickert anisotropic filter, masked to exclude the background pixels, and equalized to prevent the effects of illumination changes. These feature vectors are then projected into a 100 feature vector by means of PCA.

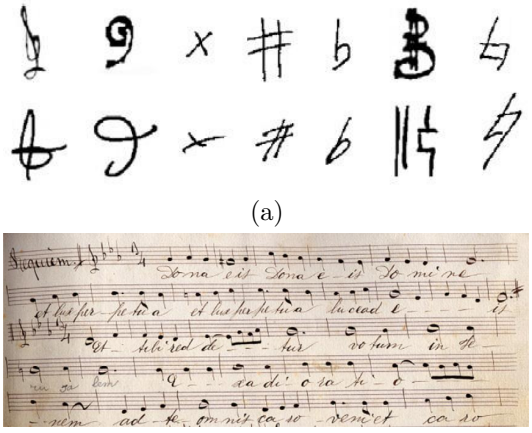
Figure 7: Traffic sign classes.

- **ARFaces classification:** The ARFace database [27] is composed of 26 face images for each one of the 126 different subjects (70 men and 56 women). The images have uniform white background. The database has two sets of images from each person, acquired in two different sessions, with the following structure: one sample of neutral frontal images, three samples with strong changes in the illumination, two samples with occlusions (scarf and glasses), four images combining occlusions and illumination changes, and three samples with gesture effects. One example of each type is plotted in Figure 8. For this experiment, we selected all the samples from 20 different categories (persons).

- **Clefs and accidental dataset:** The dataset of clefs and accidental is obtained from a collection of modern and old musical scores (19th century) of the Archive of the Seminar of Barcelona. The dataset contains a total of 4098 samples among seven different types of clefs and accidental from 24 different authors. The images have been obtained from original image documents using a semi-supervised segmentation approach [29]. The main

Figure 8: ARFaces dataset classes. Examples from a category with neutral, smile, anger, scream expressions, wearing sun glasses, wearing sunglasses and left light on, wearing sun glasses and right light on, wearing scarf, wearing scarf and left light on, and wearing scarf and right light on.

difficulty of this dataset is the lack of a clear class separability because of the variation of writer styles and the absence of a standard notation. A pair of segmented samples for each of the seven classes showing the high variability of clefs and accidental appearance from different authors can be observed in Figure 4.1 (a). An example of an old musical score used to obtain the data samples are shown in Figure 4.1(b). The object images are described using the Blurred Shape Model descriptor [30].



(a)

(b)

Figure 9: (a) Object samples, (b) Old music score.

- **MPEG7 categorization:** The MPEG7 dataset contains 70 classes with 20 instances per class, which represents a total of 1400 object images. All

samples are described using the Blurred Shape Model descriptor [30]. A couple of samples for some categories of this dataset are shown in Figure 10.

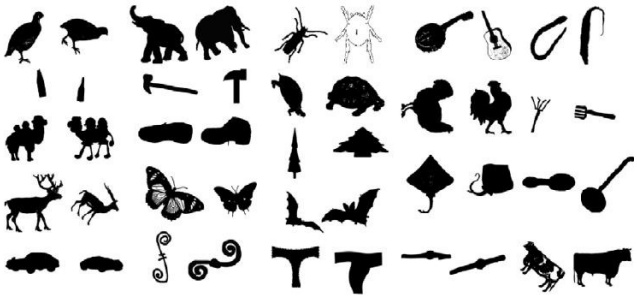


Figure 10: MPEG7 samples.

4.2. Methods

We compare the One vs. One [14] and One vs. All [15] ECOC, DECOC [6] and Forest-ECOC [16] approaches with the novel ECOC-compliant genetic approach. Moreover, the approaches of [9] and [10] have been replicated in order to obtain a fair comparison with state-of-the-art ECOC GA methods. The Loss-Weighted decoding is applied at the decoding step [1]. The ECOC base classifier is the libsvm implementation of SVM with Radial Basis Function kernel [31].

4.3. Experimental Settings

For all experiments the base classifier used is an SVM with an RBF kernel. The optimization of its parameters is performed with a GA using a population of 60 individuals, using the operators defined by [8]. In addition for all evolutionary methods ([10], [9] and our proposal), the number of ECOC individuals in the initial population is set to $5N$, where N is the number of classes of the problem. The elite individuals is set to 10% of the population size. The stopping criteria is a stall activity of performance results during five generations.

4.4. Evaluation Measurements

The classification performance is obtained by means of a stratified five-fold cross-validation, and tested for the confidence interval with a two-tailed t-test. We also apply the Friedman and Nemenyi tests [32] in order to look for statistical significance among the obtained performances.

4.5. Experimental Classification Results

The classification results obtained for all the datasets considering the different ECOC configurations are shown in Table 3. The main trend of experimental results is that the One vs. One coding is the most successful coding in terms of

performance, obtaining very good results in most of the datasets. Nevertheless, in certain situations coding designs with far less number of dichotomizers can achieve similar or even better results (i.e E.coli, Yeast, and CLEAFS results). Moreover, taking into account the number of classifiers yielded per each coding design we can see how those codings that were optimized with a GA are far more efficient than the predefined ones. In addition, in order to compare the performances provided for each strategy, the table also shows the mean rank of each ECOC design considering the 26 different experiments (13 classification accuracies and 13 coding lengths). The rankings are obtained estimating each particular ranking r_i^j for each problem i and each ECOC configuration j, and computing the mean ranking R for each design as $R_j = \frac{1}{N}\sum_i r_i^j$, where N is the total number of datasets. We also show the mean number of classifiers (#) required for each strategy. Furthermore, Table 4 shows the mean performance ranking and the mean performance per classifier ranking, which is computed as the rank of $PC = \frac{\sum_{i=1}^{\mathcal{N}} 1-E^i}{\sum_{i=1}^{\mathcal{N}} n^i}$, where $1 - E_i$ is the performance obtained in the i-th problem and n^i is the length of the code in the i-th problem.

In order to reject the null hypothesis that the measured performance ranks differ from the mean performance rank, and that the performance ranks are affected by randomness in the results, we use the Friedman test. The Friedman statistic value is computed as follows:

$$X_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right]. \tag{13}$$

In our case, with $k = 8$ ECOC designs to compare, $X_F^2 = 11.26$. Since this value is undesirable conservative, Iman and Davenport proposed a corrected statistic:

$$F_F = \frac{(N-1)X_F^2}{N(k-1) - X_F^2}. \tag{14}$$

Applying this correction we obtain $F_F = 1.1$. With eight methods and thirteen experiments, F_F is distributed according to the F distribution with 7 and 175 degrees of freedom. The critical value of $F(7, 175)$ for 0.05 is 0.31. As the value of F_F is higher than 0.31 we can reject the null hypothesis.

Furthermore, we perform a Nemenyi test in order to check if any of these methods can be singled out [32], the Nemenyi statistic is obtained as follows:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}. \tag{15}$$

In our case, for $k = 8$ ECOC designs to compare and $N = 26$ experiments the critical value for a 90% of confidence is $CD = 2.780 \cdot \sqrt{\frac{56}{156}} = 1.8$. In this case, since our approach is the best in rank but it intersects with Minimal ECOC, Pedrajas et. al, DECOC, and, Lorena et al. approaches, we can state that there is no statistically significant difference among these five approaches. However, since our approach uses less dichotomizers than any of the other approaches

(except for the Minimal ECOC which obtains the lowest classification accuracy ranking) it can be considered as the most suitable choice.

Moreover, we have to take into account that although Minimal ECOC, Pedrajas et. al, DECOC, and, Lorena et al. approaches intersect with our proposal, the number of SVs defined by these approaches are generally bigger than the number of SVs defined by our proposal. Therefore, their testing complexity is larger than the one showed by our method (see Section 4.6.2 for further details). As for the Minimal ECOC method, although it is close in rank to our method we can see in Table 3 that the randomness of its optimization leads to a much lower classification performance.

This results support the fact that using far less number of dichotomizers than standard techniques our proposal is able to find ECOC matrices with a extremely high efficiency. In addition, the general trend of the experiments shows that the proposed method improves the classification accuracy of methods at the same complexity level and reduces the computational complexity of methods with similar accuracy.

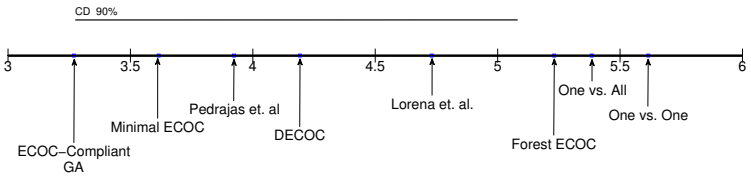


Figure 11: Critical difference for the Nemenyi test and the performance per classifier ranking values.

4.6. Discussion

4.6.1. Regularization Analysis

In section 3.0.2 we defined the Fitness function of an ECOC individual as follows:

$$F_f(I_k) = E_{I_k} + \lambda n_{I_k}, \tag{16}$$

where λ is a user defined value that plays a regularization role for the ECOC matrix, similar to the control of learning capacity in regularized classifiers. In our proposal the value of λ is estimated by a cross-validation procedure. In Figure 12 a cross-validation procedure to determine the λ value is shown for the *Vowel* dataset. In this procedure, the values of λ follow a logarithmic progression (from 0.01 to 1). It can be seen how the number of classifiers

Table 3: Classification results and number of classifiers per coding design.

	ECOC-Compliant GA		Minimal ECOC		Lorena et al.	
Dataset	Perf.	#Class.	Perf.	#Class.	Perf.	#Class.
Vowel	64.7±13.4	5.6	57.7±22.2	4.0	69.3±11.3	9.2
Yeast	55.6±12.2	5.0	50.2±17.3	4.0	46.9±15.3	6.4
E.coli	**84.5±10.2**	3.0	80.5±9.7	3.0	83.1±13.2	5.2
Glass	50.1±22.8	5.0	38.4±23.4	3.0	45.2±21.9	6.4
Segment	96.8±1.2	5.0	66.9±3.4	3.0	97.1±1.4	5.6
Dermatology	96.3±3.1	3.8	96.0±4.2	3.0	96.5±2.9	3.8
Shuttle	74.8±13.2	4.0	72.5±25.3	3.0	73.6±13.2	4.6
Vehicle	81.1±10.3	3.0	72.5±13.3	2.0	82.0±12.2	5.6
Satimage	84.3±3.1	4.0	79.2±4.2	3.0	54.7±6.5	6.6
MPEG	84.4±2.8	7.0	89.3±3.9	7.0	84.4±1.3	7.0
ARFACE	76.5±4.8	5.4	76.0±5.7	5.0	84.2±3.2	8.4
TRAFFIC	84.1±3.6	6.0	90.8±2.6	6.0	92.3±2.9	6.8
CLEAFS	96.3±6.9	3.0	81.2±8.7	3.0	96.3±7.8	7.0
Rank & #Class.	4.5	4.5	6.2	**3.7**	4.4	6.3

	Pedrajas et al.		One vs. All		One vs. One	
Dataset	Perf.	#Class.	Perf.	#Class.	Perf.	#Class.
Vowel	55.7±18.3	7.0	**80.7±11.0**	11.0	78.9±13.2	28.0
Yeast	53.5±18.2	5.0	51.1±16.7	10.0	52.4±12.3	45.0
E.coli	83.1±13.3	3.0	79.5±10.3	8.0	79.2±12.3	28.0
Glass	56.1±25.7	5.0	53.9±23.5	7.0	**60.5±21.3**	15.0
Segment	96.8±1.7	3.0	96.1±2.2	7.0	**97.2±1.7**	21.0
Dermatology	95.7±2.3	4.0	95.1±1.3	6.0	94.7±2.3	15.0
Shuttle	68.5±17.2	4.0	**90.6±13.2**	7.0	86.3±14.2	21.0
Vehicle	79.6±15.7	3.8	74.2±11.2	3.0	83.6±9.6	6.0
Satimage	83.5±5.2	3.0	83.9±5.6	6.0	**85.2±7.9**	15.0
MPEG	84.7±2.6	7.0	87.8±3.4	70.0	**92.8±2.3**	2415
ARFACE	77.7±6.7	5.8	84.0±3.9	20.0	**96.0±2.8**	190.0
TRAFFIC	93.8±3.2	6.0	91.8±3.4	36.0	90.6±4.2	630.0
CLEAFS	94.9±6.3	3.0	80.8±4.8	7.0	84.2±6.7	21.0
Rank & #Class.	4.8	4.6	4.7	15.2	**3.2**	265.3

	DECOC		Forest ECOC	
Dataset	Perf.	#Class.	Perf.	#Class.
Vowel	66.8±12.8	8.2	70.2±10.3	30.0
Yeast	55.8±15.4	5.4	**56.1±13.2**	27.0
E.coli	69.5±9.7	4.2	75.2±8.9	21.0
Glass	55.0±21.3	5.4	46.9±21.8	15.0
Segment	97.0±2.3	4.6	97.1±1.3	18.0
Dermatology	**97.1±4.3**	2.8	96.0±2.1	15.0
Shuttle	77.1±13.2	3.6	84.4±12.1	18.0
Vehicle	**84.1±10.3**	4.6	81.7±13.3	9.0
Satimage	52.3±5.3	5.0	51.9±4.7	21.0
MPEG	83.4±2.5	69.0	88.9±4.3	207.0
ARFACE	82.7±4.3	19.0	85.6±4.2	147.0
TRAFFIC	86.2±2.9	35.0	**96.7±2.5**	105.0
CLEAFS	96.9±5.3	6.0	**97.1±4.2**	18.0
Rank & #Class.	4.3	14.2	3.4	50.0

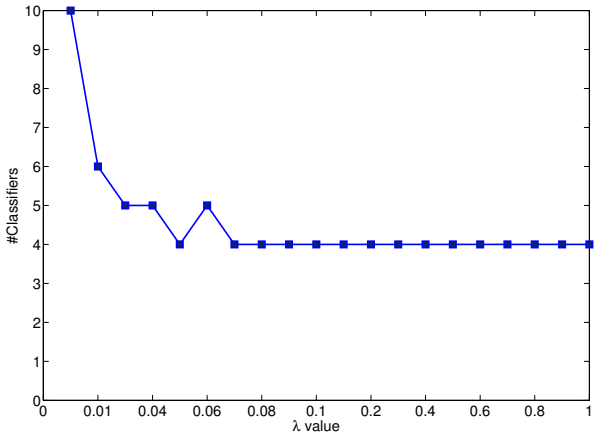
yielded by the proposal diminishes when λ increases (particularly between 0.01 and 0.05). Finding its minimum at $\lambda = 0.06$ which corresponds to the Minimal ECOC length [8].

In this sense, in our experimental settings the lambda value was set to be in the middle point of the interval $[\lambda_{min}, \lambda_{max}]$, where λ_{min} is the smallest value of λ that yielded the smallest ECOC code length, and respectively for λ_{max}. Therefore, by performing this cross-validation setting of λ our proposal is able

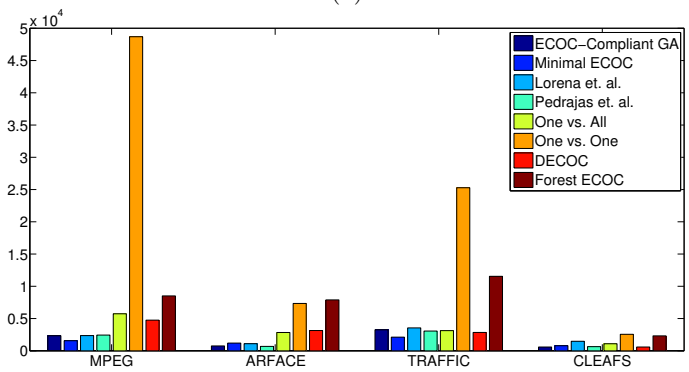
Table 4: Mean rank per coding design.

	ECOC-Compliant GA	Minimal ECOC	Lorena et al.	Pedrajas et al.
Perf. rank	4.5	6.2	4.4	4.8
#Class. rank	2	**1**	5	3
PC rank	**3.2**	3.6	4.7	3.9

	One vs.All ECOC	One vs. One ECOC	DECOC	Forest ECOC
Perf. rank	4.7	**3.2**	4.3	3.4
#Class. rank	6	8	4	7
PC rank	5.3	5.6	4.1	5.2



(a)



(b)

Figure 12: (a) Number of classifiers per λ value. (b) Number of SVs per coding design in the Computer Vision problems.

to find accurate models without a extremely high complexity in terms of the number of classifiers.

4.6.2. Complexity Analysis

In order to provide a complete description of the method, in this section we analyze its computational training and testing cost.

Let us consider the average computational cost for optimizing a dense dichotomizer (a dichotomizer in which all classes are taken into account) as a constant value. Then we find that in the case of One vs. All coding, we have n dichotomizers to be optimized, and thus, its computational cost is $O(n)$. In the case of evolutionary codings (Pedrajas et al., Lorena et al. and our proposal) we have $O(G \cdot I \cdot log_2(n))$, where I represents the number of individuals in the genetic optimization (which is constant along generations) and G represents the number of generations of the Genetic Algorithm. Undoubtedly, this cost may be greater than the cost of the other non evolutionary techniques. Nevertheless, with the introduction of an historic of optimized dichotomizers each dichotomizer is optimized once and its parameters are stored for future usage. In this sense, since the number of partitions of the classes is finite, as the algorithm progresses, the number of dichotomizers to be optimized exponentially decreases. Thus, with this approximation procedure the value of G tends to one. In consequence, in an optimal case the computational complexity becomes $O(I \cdot log_2(n))$.

In addition to the usual performance measures we also provide the number of Support Vectors (SVs) per coding scheme. Since this number is proportional to the complexity in the test step we can perform an analysis of what strategies show less complexity while still obtaining high performance. Figure 12(b) shows the number of SVs per coding design for the UCI data and Figure 12(b) shows the number of SVs for the Computer Vision problems. We can see that, in most cases, the four first columns of each dataset (corresponding to the Binary Minimal and all the evolutionary strategies) yield a lower number of SVs per model. Analyzing the empirical results shown in Figures 13 and 12(b), we find that the number of SVs defined by all three evolutionary strategies and the Binary Minimal approach are significantly smaller than the number of SVs defined by other strategies. This is due to the fact that all initial populations of evolutionary methods were set to follow a Minimal design [8], and thus, they are all expected to yield a similar number of SVs. However, the new proposal defines a more reduced number of SVs than other standard predefined or even problem-dependent strategies. Defining a compact but yet discriminate enough number of SVs, obtaining comparable or even better results than other coding designs while dramatically reducing the testing time.

4.6.3. Convergence Analysis

This section is devoted to perform an analysis of convergence for the methods that use a GA to optimize the ECOC matrix ([9, 10] and our proposal). To properly perform this analysis we ran experiments for three UCI datasets (*Glass*, *Vowel* and *Yeast*), fixing the initial population to avoid the random initialization point issue (although equivalent ECOC matrices may yield different results due to the Genetic tuning of the SVM parameters). The number of generations was

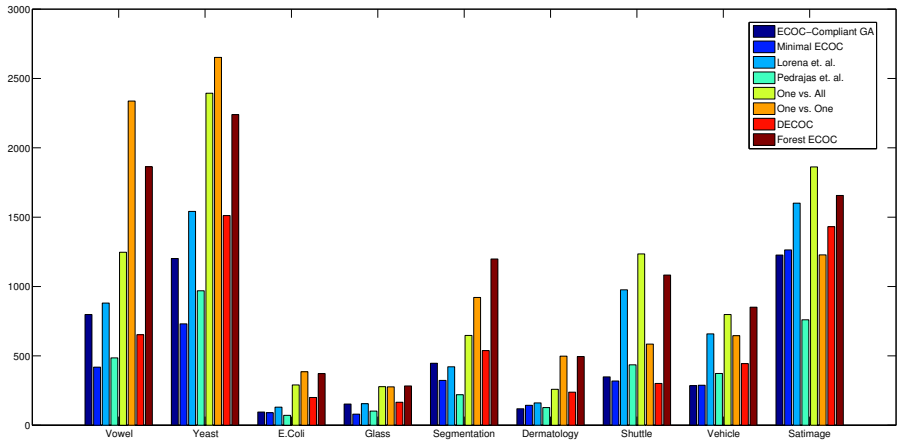


Figure 13: Number of SVs per coding design in the UCI datasets.

set to 50 and the rest of the experimental settings were the same as the ones described in Section 4.3.

In Figure 14 we show the evolution of the classification error, as well as the evolution of the performance per classifier rate for three UCI datasets. In Figures 14(a), 14(c) and 14(e) we can see how most of the times our proposal converges faster to better results than the state-of-the-art GA approaches. This fact is motivated by the redefinition of the operators that allows a fast exploration of the search space, without generating non-valid individuals. In addition, Figures 14(b), 14(d) and 14(f) show the evolution of the performance per classifier rate along generations. Figures 14(b) and 14(f) clearly show that our proposal yields models that are more efficient since we get a higher performance per classifier rate. Nevertheless, in Figure 14(d) the proposal of Pedrajas et al. obtains a higher rate. This is motivated by the fact that in the calculus of such rate both the classification error and the produced code length have the same weight. However, Figure 14(c) clearly shows that our method obtains better classification results.

Experimental results show that our proposal is able to converge faster to better results. In addition, the models yielded by the novel proposal are more efficient than the ones obtained by similar approaches. These results are obtained because of the redefinition of the crossover and mutation operators taking into account the theoretical properties of the ECOC framework. In this sense, our operators have a higher probability of finding good ECOC matrices than others since our search space is more reduced and the operators can guide the optimization procedure to promising regions of the search space.

5. Conclusions

In this paper we presented a novel ECOC-Compliant Genetic Algorithm for the coding step in the ECOC framework. The novel methodology redefines the

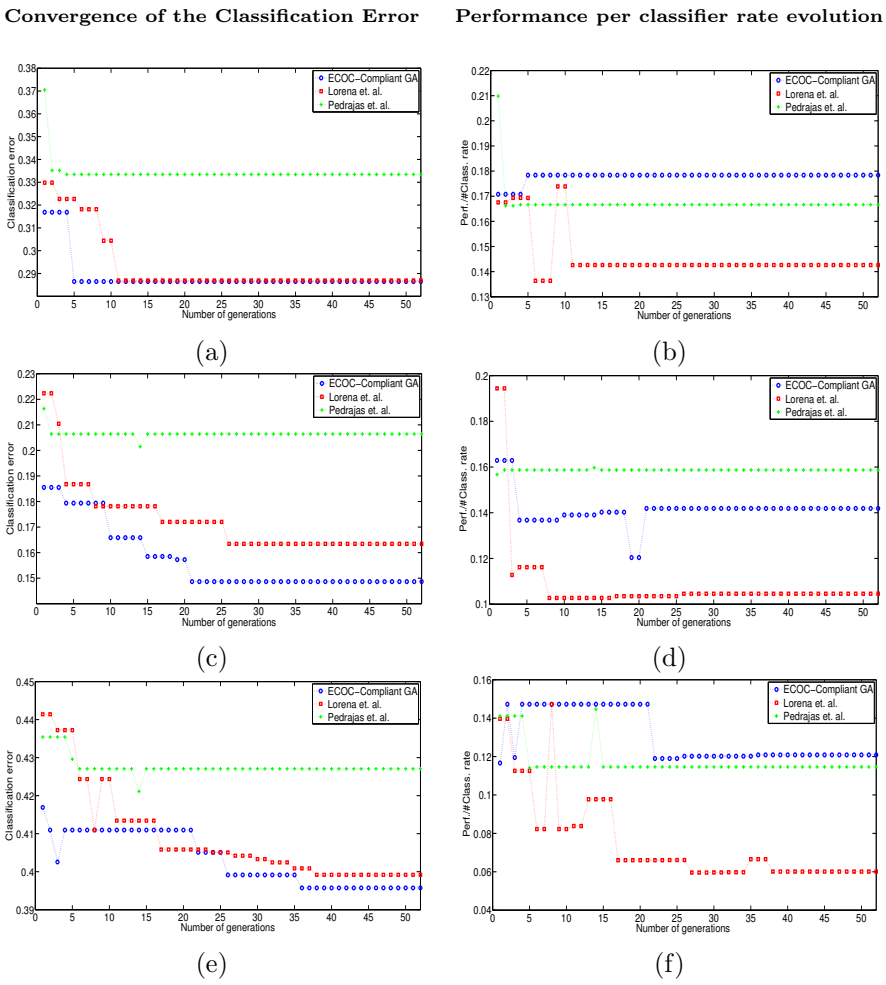


Figure 14: (a) Classification error evolution for the *Glass* dataset. (b) Evolution of the Performance per classifier rate for the *Glass* dataset. (c)Classification error evolution for the *Vowel* dataset.(d) Evolution of the Performance per classifier rate for the *Vowel* dataset. (e)Classification error evolution for the *Yeast* dataset. (f) Evolution of the Performance per classifier rate for the *Yeast* dataset.

usual crossover and mutation operators taking into account the properties of ECOC matrices. As a result, the search space is cropped, which causes the optimization to converge faster. Furthermore, a new operator which is able to increment the code length in a smart way was also introduced. The initial ECOC population followed a Minimal coding scheme, in which only $\lceil log_2(N) \rceil$ classifiers are needed to discriminate N classes, and as consequence, the methodology is able to find very efficient codes.

The proposal was tested on a wide set of datasets of the UCI Machine Learn-

ing Repository and over four challenging Computer Vision problems. For comparison purposes state-of-the-art ECOC GA schemes were replicated, as well as standard ECOC coding techniques. All the experiments were carried out using Support Vector Machines with an RBF kernel. Experimental results showed that the new proposal obtains significant improvements in comparison to state-of-the-art techniques.

In particular, we analyzed the performance in terms of code length, and training and testing complexity. Those analysis showed that our proposal is able to find ECOC matrices with a high efficiency based on a trade-off optimization between performance and complexity obtained along the GA ECOC-Compliant optimization process.

Acknowledgements

This work is partly supported by projects IMSERSO-Ministerio de Sanidad 2011 Ref. MEDIMINDER and RECERCAIXA 2011 Ref. REMEDI, and SURDEC of the Generalitat de Catalunya and FSE.

References

[1] S. Escalera, O. Pujol, P.Radeva, On the decoding process in ternary error-correcting output codes, Transactions in Pattern Analysis and Machine Intelligence 99 (1).

[2] J. D. Zhou, X. D. Wang, H. J. Zhou, J. M. Zhang, N. Jia, Decoding design based on posterior probabilities in ternary error-correcting output codes, Pattern Recognition 45 (4) (2012) 1802 – 1818.

[3] E. L. Allwein, R. E. Schapire, Y. Singer, P. Kaelbling, Reducing multiclass to binary: A unifying approach for margin classifiers, Journal of Machine Learning Research 1 (2000) 113–141.

[4] S. Escalera, D. Tax, O. Pujol, P. Radeva, R. Duin, Subclass problem-dependent design of error-correcting output codes, in: IEEE Transactions in Pattern Analysis and Machine Intelligence, Vol. 30, 2008, pp. 1–14.

[5] T. Dietterich, E. Kong, Error-correcting output codes corrects bias and variance, in: ICML (Ed.), S. Prieditis and S. Russell, 1995, pp. 313–321.

[6] O. Pujol, P. Radeva, J. Vitrià, Discriminant ECOC: A heuristic method for application dependent design of error correcting output codes, in: Trans. on PAMI, Vol. 28, 2006, pp. 1001–1007.

[7] K. Crammer, Y. Singer, On the learnability and design of output codes for multi-class problems, in: Machine Learning, Vol. 47, 2002, pp. 201–233.

[8] M. Bautista, S. Escalera, X. Baró, P. Radeva, J. Vitriá, O. Pujol, Minimal design of error-correcting output codes, Pattern Recognition Letters. In press.

[9] A. C. Lorena, A. C. P. L. F. Carvalho, Evolutionary design of multiclass support vector machines, J. Intell. Fuzzy Syst. 18 (2007) 445–454.

[10] N. Garcia-Pedrajas, C. Fyfe, Evolving output codes for multiclass problems, Evolutionary Computation, IEEE Transactions on 12 (1) (2008) 93 –106.

[11] L. J. Eshelman, The chc adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination, in: FOGA, 1990, pp. 265–283.

[12] T. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, in: JAIR, Vol. 2, 1995, pp. 263–286.

[13] E. Allwein, R. Schapire, Y. Singer, Reducing multiclass to binary: A unifying approach for margin classifiers, in: JMLR, Vol. 1, 2002, pp. 113–141.

[14] T.Hastie, R.Tibshirani, Classification by pairwise grouping, NIPS 26 (1998) 451–471.

[15] R. Rifkin, A. Klautau, In defense of one-vs-all classification, JMLR 5 (2004) 101–141.

[16] S. Escalera, O. Pujol, P. Radeva, Boosted landmarks and forest ECOC: A novel framework to detect and classify objects in clutter scenes, in: Pattern Recognition Letters, Vol. 28, 2007, pp. 1759–1768.

[17] O. Pujol, S. Escalera, P. Radeva, An incremental node embedding technique for error correcting output codes, Pattern Recognition 41 (2) (2008) 713 – 725.

[18] J. Zhou, H. Peng, C. Y. Suen, Data-driven decomposition for multi-class classification, Pattern Recognition 41 (1) (2008) 67 – 76.

[19] P. Simeone, C. Marrocco, F. Tortorella, Design of reject rules for ecoc classification systems, Pattern Recognition 45 (2) (2012) 863 – 875.

[20] E. Alpaydin, E. Mayoraz, Learning error-correcting output codes from data, in: Artificial Neural Networks, 1999. ICANN 99. Ninth International Conference on (Conf. Publ. No. 470), Vol. 2, 1999, pp. 743 –748 vol.2.

[21] W. Utschick, W. Weichselberger, Stochastic organization of output codes in multiclass learning problems, in: Neural Computation, Vol. 13, 2004, pp. 1065–1102.

[22] J. Holland, Adaptation in natural and artificial systems: An analysis with applications to biology, control, and artificial intelligence, University of Michigan Press, 1975.

[23] S. Escalera, O. Pujol, P. Radeva, Ecoc-one: A novel coding and decoding strategy, in: Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, Vol. 3, 2006, pp. 578 –581.

[24] A. C. Lorena, A. C. de Carvalho, Evolutionary tuning of svm parameter values in multiclass problems, Neurocomputing 71 (16-18) (2008) 3326 – 3334.

[25] A. Asuncion, D. Newman, UCI machine learning repository, University of California, Irvine, School of Information and Computer Sciences, http://www.ics.uci.edu/∼mlearn/MLRepository.html, 2007.

[26] J. Casacuberta, J. Miranda, M. Pla, S. Sanchez, A.Serra, J.Talaya, On the accuracy and performance of the GeoMobil system, in: International Society for Photogrammetry and Remote Sensing, 2004.

[27] A. Martinez, R. Benavente, The AR face database, in: Computer Vision Center Technical Report #24, 1998.

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

[29] A. Fornés, J. Lladós, G. Sánchez, Primitive segmentation in old handwritten music scores, Graphics Recognition 3926 (2006) 279–290.

[30] S. Escalera, A. Fornes, O. Pujol, P. Radeva, G. Sanchez, J. Llados, Blurred shape model for binary and grey-level symbol recognition, Pattern Recognition Letters 30 (2009) 1424–1433.

[31] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, ACM Transactions on Intelligent Systems and Technology 2 (2011) 27:1–27:27, software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[32] J. Demsar, Statistical comparisons of classifiers over multiple data sets, JMLR 7 (2006) 1–30.