Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition 41 (2008) 713–725
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/pr

An incremental node embedding technique for error correcting output codes

Oriol Pujol[a,b,*], Sergio Escalera[b], Petia Radeva[b]

[a]*Dept. Matemàtica Aplicada i Anàlisi, UB, Gran Via 585, 08007 Barcelona, Spain*
[b]*Centre de Visió per Computador and Computer Science Dept, Campus UAB, 08193 Bellaterra, Barcelona, Spain*

Received 20 June 2006; received in revised form 17 April 2007; accepted 20 April 2007

Abstract

The error correcting output codes (ECOC) technique is a useful way to extend any binary classifier to the multiclass case. The design of an ECOC matrix usually considers an *a priori* fixed number of dichotomizers. We argue that the selection and number of dichotomizers must depend on the performance of the ensemble code in relation to the problem domain. In this paper, we present a novel approach that improves the performance of any initial output coding by extending it in a sub-optimal way. The proposed strategy creates the new dichotomizers by minimizing the confusion matrix among classes guided by a validation subset. A weighted methodology is proposed to take into account the different relevance of each dichotomizer. As a result, overfitting is avoided and small codes with good generalization performance are obtained. In the decoding step, we introduce a new strategy that follows the principle that positions coded with the symbol zero should have small influence in the results. We compare our strategy to other well-known ECOC strategies on the UCI database, and the results show it represents a significant improvement.

Keywords: Multiclass classification; Error correcting output codes; Ensemble of dichotomizers; Codeword design; One-versus-one; One-versus-all

1. Introduction

Multiclass classification is the term applied to those machine learning problems that require assigning labels to instances where the labels are drawn from a set with at least three classes. Many examples of this problem can be found in real life applications: in the case of optical character recognition, the goal is to find the digit value or the character letter. In object recognition, a new instance is categorized according to the pool of trained objects (cars, motorbikes, horses, flowers, etc.). In medical imaging, for instance, a potential application would be the automatic classification of different kind of plaque tissues (lipidic, fibrous, calcified, necrotic, etc.).

However, in designing machine learning techniques, it is common to conceive algorithms for distinguishing between just two classes. Some of the well-known binary classification learning algorithms can be extended to handle multiclass problems, but for most algorithms this extension is very difficult. In such cases, the usual way to proceed is to reduce the complexity of the original problem by dividing it into a set of multiple simpler binary classification sub-problems. Pairwise (one-versus-one) [1] or one-versus-all [2] grouping techniques are the schemes most frequently used.

In the line of these techniques, Dietterich et al. [3] proposed a framework inspired in the signal processing coding and decoding techniques with error correction properties. This technique divides the problem into n binary problems (dichotomizers) that are combined forming a multiclass classifier ensemble. In the error correcting output codes (ECOC), the multiclass to binary classification process is handled by a coding matrix. Each column of the matrix shows a partition of the classes in two sets $\{-1, +1\}$. Alternatively, each row of the coding matrix represents a codeword assigned to each class. The decoding technique defines the strategy for assigning the "closest" codeword given a test sample. This binary strategy has been extended in several researches showing promising results [4,5]. Allwein et al. [6] improved the representability of the ECOC technique by adding a third symbol to the coding matrix. With this new symbol each element of the coding matrix is

* Corresponding author. Dept. Matemàtica Aplicada i Anàlisi, UB, Gran Via 585, 08007 Barcelona, Spain. Tel.: +34 636421147.
E-mail addresses: oriol@maia.ub.es, oriol@cvc.uab.es (O. Pujol), sescalera@cvc.uab.es (S. Escalera), petia@cvc.uab.es (P. Radeva).

0031-3203/$30.00 © 2007 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2007.04.008

chosen from $\{-1, 0, +1\}$, where classes with zero value are not considered for that particular dichotomy. This improved technique allows to express the classic pairwise and one-versus-all schemes in one common framework as well as to define new coding strategies such as random dense or random sparse output codes. Due to its simplicity and high accuracy performance, the output coding scheme has been widely applied to very different problems with great success: anti-spam filtering [7], text classification [8], face verification [9], object identification or traffic sign recognition [10], to mention just a few.

All these coding strategies are fixed in the ECOC design step, defined independently of the problem domain or the classification performance. In fact, very little attention has been paid in literatures to the coding process of the ECOC matrix. The first approach to ECOC coding design was proposed by Utschick et al. [11]. In their work, they optimize a maximum-likelihood objective function by means of the expectation maximization algorithm in order to improve the process of binary coding. As mentioned by the authors "the results of some experiments make us believe that for many polychotomous classification problems, the one-versus-all method is still the optimal choice for the output coding". Crammer et al. [12] also reported improvement in the design of the ECOC codes. However, their results were rather pessimistic since they proved that the problem of finding the optimal discrete codes is computationally unfeasible. As an alternative, they proposed a method for heuristically search of the optimal coding matrix by relaxing the representation of the code matrix from discrete to continuous values. Recently, new improvements in the problem dependent coding techniques have been presented by Pujol et al. [13]. In their work, the authors proposed the embedding of discriminant tree structures derived from the problem domain in the ECOC framework. As a result, they obtained a compact discrete coding matrix with a small number of dichotomizers and very high accuracy.

In the present article, we propose a method for extending any discrete ECOC coding matrix driven by the performance of the ensemble of dichotomizers. Our work is defined in the context of discrete matrix coding adapted to the problem domain. As a result, just adding a very few number of dichotomizers, we obtain an ensemble with increased generalization performance. We pay special attention to the extension of binary tree-based ECOC because of its ability to build compact codes and the high performance usually obtained by this technique [13]. ECOC-optimizing node embedding (ECOC-ONE) is based on a robust strategy of selective optimization process focused on the confusion matrices of two exclusive training data sets. The first set is used for standard training purposes. The second one is used for guiding the process and to avoid classification overfitting. In this way, the training process selects at each step the optimal hypothesis based on its discrimination performance. As a result, wrongly classified classes are given priority and are used for creating the candidate dichotomizer. As a consequence of the ECOC-ONE coding, the Hamming distance between *difficult*[1] classes is increased. In this way, the generalization performance is also increased [3]. This generalization is reinforced by a weighting strategy used to define the relevance of each dichotomizer. These weights are used in a double weighted Euclidean distance decoding step that takes into account the importance of each dichotomizer and the error due to the offset introduced by the zero symbol in the ECOC matrix.

The article layout is as follows: Section 2 introduces the general procedure of ECOC techniques used in the literatures. Section 3 describes the proposed extension using the ECOC-ONE strategy. In Section 4, the technique is evaluated on a four-class toy classification problem. Section 5 shows the experiment results, and Section 6 concludes the paper.

2. Error correcting output codes

The idea behind the ECOC framework is to design a codeword for each of the given N_c classes. Arranging these codewords as rows of a matrix, we define the "coding matrix" M, where $M \in \{-1, 1\}^{N_c \times n}$, n being the code length. From the point of view of learning, the matrix M can be seen as n independent binary learning problems, each corresponding to a column of the matrix (see the six hypothesis $\{h_1, \ldots, h_6\}$ from Fig. 1). Thus, classes are joined to form sub-partitions (sets of classes). Each dichotomy is trained on a sub-partition where a class is coded by $+1$ or -1 according to their partition membership. Applying the n trained binary classifiers to each data point in the test set, the test code is obtained. This code is compared with the base codeword of each class–rows of the coding matrix—and the data point is assigned to the class with the "closest" codeword.

The coding matrix was extended by Allwein et al. [6] by including a third symbol $M \in \{-1, 0, 1\}^{N_c \times n}$. With this new codification, classes that are not considered by a given dichotomy have 0 value. This gives to the ECOC more versatility allowing the representation of more codification strategies in the ECOC framework.

In general, the ECOC system is fully defined when a coding and decoding strategy is chosen. It has been shown that to obtain maximal distance between partition of classes, ECOC code matrices should be designed to have certain properties which enable them to generalize well [3]. A good error-correcting output code for a N_c-class problem should satisfy the condition that rows, columns (and their complementaries) are well-separated from the rest in terms of Hamming distance.

In Fig. 1 an example of ECOC coding and decoding for a classification example is shown. In this case we have four

	h_1	h_2	h_3	h_4	h_5	H_6
C_1	1	-1	-1	1	0	0
C_2	1	0	1	1	1	-1
C_3	-1	1	1	1	-1	1
C_4	1	1	-1	-1	-1	0

$$h_1(x) = \begin{cases} 1 & \text{if } x \in \{C_1, C_2, C_4\} \\ -1 & \text{if } x \in \{C_3\} \end{cases} \quad h_2(x) = \begin{cases} 1 & \text{if } x \in \{C_3, C_4\} \\ -1 & \text{if } x \in \{C_1\} \end{cases} \quad h_3(x) = \begin{cases} 1 & \text{if } x \in \{C_2, C_3\} \\ -1 & \text{if } x \in \{C_1, C_4\} \end{cases}$$

$$h_4(x) = \begin{cases} 1 & \text{if } x \in \{C_1, C_2, C_3\} \\ -1 & \text{if } x \in \{C_4\} \end{cases} \quad h_5(x) = \begin{cases} 1 & \text{if } x \in \{C_2\} \\ -1 & \text{if } x \in \{C_3, C_4\} \end{cases} \quad h_6(x) = \begin{cases} 1 & \text{if } x \in \{C_3\} \\ -1 & \text{if } x \in \{C_2\} \end{cases}$$

Fig. 1. ECOC example for four classes using Hamming distance as a decoding technique.

[1] Those that are most overlapped.

classes (c_1, c_2, c_3, and c_4). Six dichotomizers, $h_1, \ldots, h_6$, are generated randomly by selecting different sub-partitions of the set of classes. The dichotomizers are embedded in the ECOC matrix with 1 or -1 according to their sub-partition membership. For example, h_1 learns to discriminate between c_3 vs c_1, c_2 and c_4. The cells with zero value represent the classes that are not considered by a given dichotomy—e.g. for the second dichotomizer (h_2), class c_2 is not considered. Let $\mathbf{x} \in \{-\mathbf{1}, \mathbf{1}\}^{\mathbf{n}}$ be the codeword that results from applying all the dichotomizers to a new input. This test codeword is compared with the codeword of each class $c_j, \{y_1^j, \ldots, y_6^j\}$using some kind of decoding distance.

The matrix construction step codifies the different partitions of classes that are considered by each dichotomizer. Most of the discrete coding strategies up to now are based on pre-designed problem-independent codeword construction satisfying the requirement of high separability between rows and columns. These strategies include: *one-versus-all*, where each classifier is trained to discriminate a given class from the rest of classes using N_c dichotomizers; *random techniques* that can be divided in the *dense random strategy* that consists of a two-symbol matrix with high distance between rows with estimated length of $10\log_2(N_c)$ bits per code; and the *sparse random strategy* that includes the ternary symbol and the estimated optimal length is about $15\log_2(N_c)$. Finally, *one-versus-one* is one of the most well-known coding strategies with $N_c(N_c-1)/2$ dichotomizers including all the combinations of pairs of classes [1]. Note that in a 40-class problem, one-versus-all, dense random strategy, sparse random strategy, and one-versus-one strategy require 40, 53, 80, and 780 dichotomizers, respectively. One-versus-one has obtained high popularity showing a better accuracy in comparison to the other commented strategies in spite of its large code length.

The decoding step was originally based on error-correcting principles under the assumption that the learning task could be modelled as a communication problem, in which class information is transmitted over a channel [14]. The decoding strategy corresponds to the problem of distance estimation between the codeword of the new example and the codewords of the trained classes. Concerning the decoding strategies, two of the most common techniques are the Euclidean distance $d_i^j = \sqrt{\sum_{i=1}^n (x_i - y_i^j)^2}$ and the Hamming decoding distance $d_i^j = \sum_{i=1}^n (1 - \text{sign}(x_i \cdot y_i^j))/2$, where d_i^j is the distance to the class j, n is the number of dichotomizers (and thus, the components of the codeword), and x and y are the values of the input vector codeword and the base class codeword, respectively. If the minimum Hamming distance between any pair of class codewords is d, then any $[(d-1)/2]$ errors in the individual dichotomizers result can be corrected, since the nearest codeword will still be the correct one.

3. Basis of the node embedding process in the ECOC framework

Our work is motivated by the necessity of having fast algorithms with high discriminative power able to generate as much as necessary number of dichotomizers in order to obtain the desired performance. In this sense, the work of Pujol et al. [13] has shown that finding codewords with small length and high performance is feasible if the codeword is adapted to the problem domain. Moreover, in that work the authors show that trading optimality in the codewords separation for discrimination information may lead to a rise in the classifier performance. This work has motivated the look for techniques with small codeword length that provide high performance in general conditions. In this section, we propose a general procedure to increase the accuracy of any ECOC coding by adding very few optimal dichotomizers. In this sense, if the original coding has small length, the extension after the ECOC-ONE results in a still compact codewords but with increased performance. In particular, we apply this technique to optimize the initial embedded tree proposed in Ref. [13].

3.1. ECOC-ONE definition

ECOC-ONE defines a general procedure capable of extending any coding matrix by adding dichotomizers based on a discriminability criterion. In the case of a multiclass recognition problem, our procedure starts with a given ECOC coding matrix. We increase this ECOC matrix in an iterative way, adding dichotomizers that correspond to different sub-partitions of classes. These partitions are found using greedy optimization based on the confusion matrices so that the ECOC accuracy improves on both training and validation sets. The training set guides the convergence process, and the validation set is used to avoid overfitting and to select a configuration of the learning procedure that maximizes the generalization performance [15]. Since not all problems require the same dichotomizers structure—in the form of sub-partitions—our optimal node embedding approach generates an optimal ECOC-ONE matrix dependent on the hypothesis performance in a specific problem domain.

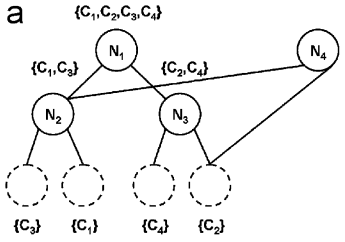
3.2. Optimizing node embedding

In order to explain our procedure, we divide the ECOC-ONE algorithm in six steps: optimal tree generation, weights estimation, accuracy estimate based on confusion matrix, defining the new optimal dichotomy, and ECOC matrix M construction.

Let us define the notation used in the following paragraphs: given a data pair (s, l), where s is a multidimensional data point and l is the label associated with that sample, we define $S = \{(\mathbf{s}, \mathbf{l})\} = \{(\mathbf{s_t}, \mathbf{l_t})\} \cup \{(\mathbf{s_v}, \mathbf{l_v})\}$, where $S_t = \{(\mathbf{s_t}, \mathbf{l_t})\}$ and $S_v = \{(\mathbf{s_v}, \mathbf{l_v})\}$ are the sets of data pairs associated with training and validation sets, respectively. In the same way, $e(h(\mathbf{s}), \mathbf{l})$ represents the empirical error over the data set $\mathbf{s}$ given an hypothesis $h(\cdot)$.

3.2.1. Optimal tree generation

We propose the use of a binary tree structure using accuracy as a sub-partition splitting criterion. This proposal differs from the one in Ref. [13] that uses the mutual information to



b

	h1	h2	h3	h4
C1	1	-1	0	1
C2	-1	0	-1	-1
C3	1	1	0	1
C4	-1	0	1	0

Fig. 2. (a) Optimal tree and first optimal node embedded, (b) ECOC-ONE code matrix M for four dichotomizers.

form the nodes, without taking into account the particularities of the current classification scheme. We initialize the root of the tree with the set containing all the classes. Afterwards, for the tree building, each node of the tree is generated by an exhaustive search[2] of the sub-partition of classes associated with the parent node, so that the classifier using that sub-partition of classes attains maximal accuracy on the training and validation subsets. In Fig. 2, the sub-partition of classes required at each node of the optimal tree is shown. For example, given the root node containing all classes, the optimal partition achieving the least error is given by $\{\{c_1 \cup c_3\}, \{c_2 \cup c_4\}\}$. Once we have generated the optimal tree, we embed each internal node of the tree into the coding matrix M in the following way: consider the partition of the set of classes associated with a node $C = \{C_1 \cup C_2 | C_1 \cap C_2 = \varnothing\}$. The element (i, r) of the ECOC-ONE matrix corresponding to class i and dichotomy r is given by:

$$M(i,r) = \begin{cases} 0 & \text{if } c_i \notin C, \\ +1 & \text{if } c_i \in C_1, \\ -1 & \text{if } c_i \in C_2. \end{cases} \quad (1)$$

Although, this strategy is the one chosen in this article for our initial coding, note that any coding could be used instead.[3]

3.2.2. Weight estimates

It is known that when a multiclass classification problem is decomposed into binary problems, not all of these base classifiers have the same importance. In this way, our approach introduces a weight to adjust the importance of each dichotomy in the ensemble ECOC matrix. In particular, the weight associated with each column depends on the error when applying the ECOC to both training sets (training and validation) in the following way,

$$w_i = 0.5 \log\left(\frac{1 - e(h_i(s), l)}{e(h_i(s), l)}\right), \quad (2)$$

where w_i is the weight for the ith dichotomy, and $e(h_i(s), l)$ is the error produced by this dichotomy at the affected classes on both sets of the partition. This equation is based on the weighted scheme of the additive logistic regression [16]. In the following section, we explain how we select the dichotomizers and how their weights affect the convergence of the algorithm.

3.2.3. Test accuracy of the training and validation sets

Once constructed the binary tree and its corresponding coding matrix, we look for additional dichotomizers in order to focus on the examples that are difficult to classify. To select the next optimal node, we test the current M accuracy on S_t and S_v resulting in a_t and a_v, respectively. We combine both accuracies in the following way:[4]

$$a_{total} = \tfrac{1}{2}(a_t + a_v).$$

In order to find each accuracy value, we obtain the resulting codeword $x \in \{-1, 1\}^n$ using the strong hypothesis $\mathscr{H} = \{h_1, \ldots, h_j\}$ for each sample of these sets, and we label it as follows:

$$\tilde{l} = \underset{j}{\operatorname{argmin}}(d(x, y_j)), \quad (3)$$

where $d(\cdot)$ is a distance estimation between codeword x and the codeword y_j. $\mathscr{H}(M, h, s)$ is the strong hypothesis resulted from the application of the set of learning algorithms $h(\cdot)$ on the problems defined by each column of the ECOC matrix M on a data point s. The result of $\mathscr{H}(M, h, s)$ is an estimated codeword x. We propose to use a double weighted Euclidean distance in the following way:

$$d(x, y^j) = \frac{1}{2}\sqrt{\sum_{i=1}^{n} w_i |x_i||y_i|(x_i - y_i^j)^2}, \quad (4)$$

where the modules of the codes $|x_i|$,$|y_i|$ act as attenuation factors of the errors that can be accumulated due to the zero values in the ECOC-ONE matrix M. The weight w_i estimated by means of Eq. (2) introduces the relevance of each dichotomy in the ensemble learning technique.

3.2.4. The training and validation confusion matrices

Once we test the accuracy of the strong hypothesis $\mathscr{H}$ on S_t and S_v, we estimate their respective confusion matrices $v_t(\mathbf{S_t})$ and $v_v(\mathbf{S_v})$. Both confusion matrices are of size $N_c \times N_c$, and have at position (i, j) the number of instances of class c_i classified as class c_j.

$$v_k(i, j) = |\{(s, l)_k : l = c_i, h(s) = c_j\}|, \quad k = \{t, v\}, \quad (5)$$

where l is the label estimation obtained using Eq. (4). Once the matrices have been obtained, we select the pair $\{c_i, c_j\}$ with maximal value according to the following expression:

$$\{c_i, c_j\} = \underset{\{c_i, c_j; i \neq j\}}{\operatorname{argmax}} (v_t(i, j) + v_t^T(i, j) + v_v(i, j) + v_v^T(i, j)), \quad (6)$$

[2] In the case that the number of classes makes the exhaustive computation unfeasible we can use SFFS as explained in Ref. [13].

[3] In the Discussions section, the reader can find the results of the application of our extension technique using the one-versus-all strategy as initial coding.

[4] Other combinations are possible, but we consider that the importance of the validation set must be very significant when compared to the training accuracy. Otherwise, the total accuracy will have a major influence of the training set and the benefit from the validation set will be minimal. Moreover, we have experimentally observed that this combination leads in general to slightly better results than other split criteria.

$\forall(i, j) \in [1, \ldots, N_c]$, where v^T is the transposed matrix of v. The resulting pair is the set of classes that are most easily confounded, and therefore they have the maximum partial empirical error.

3.2.5. Find the new dichotomy

Once the set of classes $\{c_i, c_j\}$ with maximal error has been obtained, we create a new column of the ECOC matrix. Each candidate column considers a possible sub-partition of classes $\wp = \{\{\{c_i\} \cup C_1\}, \{\{c_j\} \cup C_2\}\} \subseteq C$ so that $C_1 \cap C_2 \cap c_i \cap c_j = \emptyset$ and $C_i \subseteq C$. In particular, we are looking for the subset division of classes $\wp$ so that the dichotomy h_t associated with that division minimizes the empirical error defined by $e(\mathscr{H}(s), l)$:

$$\tilde{\wp} = \underset{\wp}{\operatorname{argmin}}(e(\mathscr{H}(s), l)). \tag{7}$$

Once defined the new sets of classes, the column components associated with the set $\{\{c_i\}, C_1\}$ are set to $+1$, the components of the set $\{\{c_j\}, C_2\}$ are set to -1 and the positions of the rest of classes are set to zero. In the case that multiple candidates obtain the same performance, the one involving more classes is preferred. Firstly, it reduces the number of uncertainty in the ECOC matrix by reducing the number of zeros in the dichotomy. Secondly, one can see that when more classes are involved, the generalization achieved is greater. Each dichotomy finds a more complex rule on a greater number of classes. This fact has also been observed in the work of Torralba et al. [17]. In their work, a multi-task scheme is presented that yields to a classifier with an improved generalization by aids of class grouping algorithms.

3.2.6. Update the matrix

The column m_i is added to the matrix M and its weight w_i is calculated using Eq. (2).

Table 1 shows the summarized steps for the ECOC-ONE approach. Note that, the process described is iterated while the error on the training subsets is greater than ε or the number of iterations $i \leqslant T$.[5]

Table 1
ECOC-ONE general algorithm

Given N_c classes and a coding matrix M (see Fig. 1):
while $error > \varepsilon$ or $error_t < error_{t-1}$, $t \in [1, T]$:
  Compute the optimal node t:
  1) Test accuracy on the training and validation sets S_t and S_v.
  2) Select the pair of classes $\{c_i, c_j\}$ with the highest error analyzing the confusion matrices from S_t and S_v.
  3) Find the partition $\wp_t = \{C_1, C_2\}$ that minimizes the error rate in S_t and S_v.
  4) Compute the weight for the dichotomy of partition $\wp_i$ based on its classification score.
Update the matrix M.

3.3. Sub-optimal embedding

When the number of classes is high enough, exhaustive search optimization is computationally unfeasible. In this case, the problem should be addressed using a modification of the sequential forward floating search.

Pudil et al. in Ref. [18] introduced a family of sub-optimal search algorithms called *floating search methods* effective in high dimensional problems involving non-monotonic search criteria. This method was proposed as a sub-optimal search method for alleviating the prohibitive computation cost of exhaustive search strategies in feature selection. This family of methods is directly related to the *plus-l take away-r* algorithm. However, the first differs from *plus-l take away-r* algorithm in the fact that the number of forward and backtracking steps is not decided beforehand. Floating search methods can be described as a dynamically changing number of forward and backward steps as long as the resulting subsets are better than the previously evaluated ones at that level. In this sense, this method avoids nesting effects typical of sequential forward and backward selection while equally being step-optimal since the best (worst) item is always added (discarded) to (from) the set. Since backtracking is controlled dynamically, no parameter setting is needed.

The algorithm used in this paper is a modified version of the top-down approach called sequential forward floating search (MSFFS, see Table 2). The most notable difference from the SFFS is that we work with three sets of elements: a pool of elements Y and the two searched sets X^1, X^2. In this case, both sets start empty $X_0^1 = X_0^2 = \emptyset$ and they are filled from the pool set while the search criterion J applied to both sets increases. The most beneficial item from the pool of elements is added to the corresponding set at each inclusion step. In the conditional exclusion step, the worst item from both sets is removed if the criterion keeps increasing. In our approach, the criterion used for designing this partition is the empirical error. In the context of our ECOC problem, the two sets $\{X^1, X^2\}$ are the sub-partition sets of classes $\wp_t = \{C_1, C_2\}$.

3.4. ECOC-ONE example

An example of an ECOC-ONE strategy applied to a four-class classification example can be found in Fig. 2. The initial optimal tree corresponds to the dichotomizers of optimal sub-partition of the classes. This tree has been generated using accuracy as a sub-partition splitting criterion. After testing the performance of the ensemble tree (composed by the columns $\{h_1, h_2, h_3\}$ of the ECOC matrix M of Fig. 2(b)), let us assume that classes $\{c_2, c_3\}$ get maximal error in the confusion matrices v_t and v_v. We search for the sub-partition of classes using the training and validation subsets so that the

[5] The stopping criterion of our method involves two cases: Firstly, the case in which the combined error is reduced to zero. If both training and validation errors go to zero the method should stop because we cannot obtain meaningful information from now on. Therefore, ε is usually set to zero unless some *a priori* knowledge about the acceptable error is considered. Second, since the sub-optimal node embedding tries to increase the accuracy of the ECOC coding increasing the number of bits per word, a certain number of bits should be decided to be the maximum allowable for our application. In our experiments, T is usually set to values in the range $[2, \ldots, N]$, where N is the number of classes. We selected this range of values in order to increase the global performance with very few additional dichotomizers.

Table 2
Modified sequential forward floating search algorithm

Input:

$Y = \{y_j | j = 1..D\}$//Pool of available items//

Output:

$X_k^1 = \{x_l | l = 1..|Y| (or\ D), x_l \in Y\}; \quad X_k^2 = \{x_m | m = 1..|Y|, x_m \in (Y/X_k^1)\}$

Initialization:

$X_0^1 = X_0^2 = \{\emptyset\}; \quad k = 0$

Termination:
Stop when $|J(X_k^1, X_k^2) - J(X_{k-1}^1, X_{k-1}^2)| \leqslant \varepsilon$

Step 1 (inclusion)

$$x'_+ = \underset{x \in Y/\{X_k^1 \cup X_k^2\}}{\operatorname{argmax}} J(X_k^1 \cup x, X_k^2); \quad x''_+ = \underset{x \in Y/\{X_k^1 \cup X_k^2\}}{\operatorname{argmax}} J(X_k^1, X_k^2 \cup x)$$

$$(X_{k+1}^1, X_{k+1}^2) = \begin{cases} (X_k^1 \cup x'_+, X_k^2) & \text{if } J(X_k^1 \cup x'_+, X_k^2) > J(X_k^1, X_k^2 \cup x''_+) \\ (X_k^1, X_k^2 \cup x''_+) & \text{if } J(X_k^1 \cup x'_+, X_k^2) < J(X_k^1, X_k^2 \cup x''_+) \end{cases}$$

$k = k + 1$

Step 2 (conditional exclusion)

$$x'_- = \underset{x \in Y/X_k^1}{\operatorname{argmax}} J(X_k^1/x, X_k^2); \quad x''_- = \underset{x \in Y/X_k^2}{\operatorname{argmax}} J(X_k^1, X_k^2/x)$$

$$(X_{k+1}^1, X_{k+1}^2) = \begin{cases} (X_k^1/x'_-, X_k^2) & \text{if } J(X_k^1/x'_-, X_k^2) > J(X_k^1, X_k^2) \text{and } J(X_k^1/x'_-, X_k^2) > J(X_k^1, X_k^2/x''_-) \\ (X_k^1, X_k^2/x''_-) & \text{if } J(X_k^1, X_k^2/x''_-) > J(X_k^1, X_k^2) \text{and } J(X_k^1/x'_-, X_k^2) < J(X_k^1, X_k^2/x''_-) \end{cases}$$

$k = k + 1$
if $J(X_k^1, X_k^2/x''_-) > J(X_k^1, X_k^2)$ **or** $J(X_k^1/x'_-, X_k^2) > J(X_k^1, X_k^2)$
then go to **Step 2**
else go to **Step 1**

error between $\{c_2, c_3\}$ and all previous misclassified samples is minimized. Suppose now that this sub-partition is$\{c_1, c_3\}$ versus $\{c_2\}$. As a result, a new node N_4 corresponding to dichotomy h_4 is created. We can observe in Fig. 2 that N_4 uses a class partition that is present in the tree. In this sense, this new node connects two different nodes of the tree. Note that using the previously included dichotomizers, the partition $\{c_1, c_3\}$ is solved by N_2. In this way, the Hamming distance between c_2 and c_3 is increased by adding the new dichotomy to the whole structure. At the same time, the distance among the rest of the classes is usually maintained or slightly modified.

As mentioned before, one of the desirable properties of the ECOC matrix is to have maximal distance between rows. Our procedure focuses on the relevant difficult partitions, increasing the distance between "close" classes. This fact improves the robustness of the method since difficult classes are likely to have a greater number of dichotomizers centered on them. In this sense, it creates different geometrical arrangements of decision boundaries and leads the dichotomizers to make different bias errors.

4. ECOC-ONE in a four-class toy problem

To analyze the properties of our proposed technique and compare it to the state-of-art approaches, we have designed the toy classification problem of Fig. 3(a). This multiclass problem has 50 samples for each of the four classes. The ideal boundaries are shown in Fig. 3(b). In this particular case, two of the classes are difficult to classify (triangles and dots). The number of dichotomizers used in this toy problem, for each ECOC technique, is: six for one-versus-one, four for one-versus-all, and five for dense random and ECOC-ONE. We select five dichotomizers for the ECOC-ONE and the dense-random technique because we want to show the performance when the number of hypothesis is smaller than the one-versus-one method. An illustration of the training evolution process for all the techniques is shown in Fig. 4(a) where the error is given as a function of the number of dichotomizers. One can observe a greater error reduction for ECOC-ONE with few dichotomizers compared to the rest of methods. The test evolution for the same problem is shown in Fig. 4(b), where the number of dichotomizers and the error rate are shown at x-axis and y-axis,

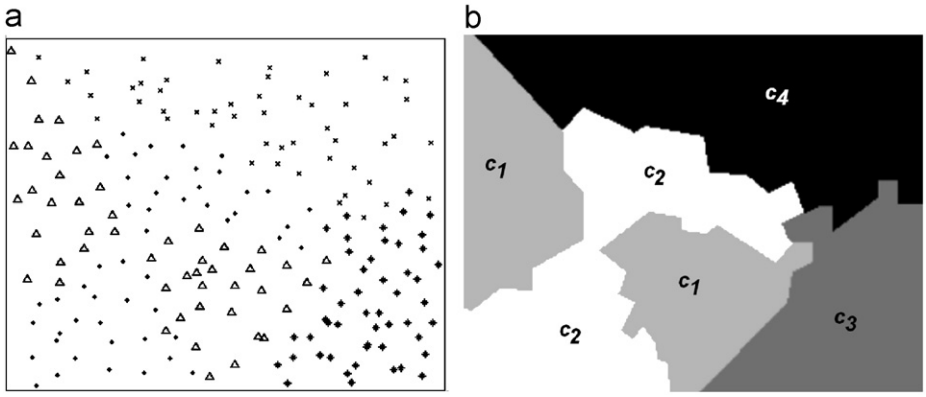


Fig. 3. (a) Four classes for a toy problem, (b) classes boundaries for the toy problem.

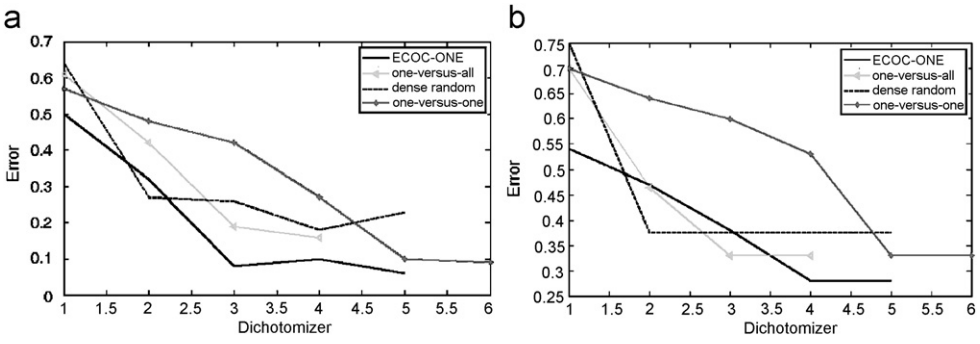


Fig. 4. (a) Train evolution for the toy problem, (b) test evolution for the toy problem.

Table 3
ECOC matrices and weights for ECOC-ONE and dense random strategy

$W_{one} = (2\ 2\ 2\ 0.9229\ 1.0271)$

(a)

$$M_{one} = \begin{pmatrix} -1 & -1 & 0 & 1 & -1 \\ -1 & 1 & 0 & -1 & 1 \\ 1 & 0 & -1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

(b)

$$M_{dense} = \begin{pmatrix} 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & 1 & 1 \end{pmatrix}$$

respectively. Table 3 displays two ECOC matrices used in this evaluation: ECOC-ONE (M_{one}) with column weights (W_{one}) and dense random (M_{dense}).

Table 4 shows the 10-fold cross-validation results for all the commented ECOC techniques. In this table, the accuracy, the confidence interval at 95%, and the number of dichotomizers used are displayed. The results on this toy classification problem show that our technique outperforms the others. An example of the trained boundaries for all the techniques at one iteration of cross-validation is shown in Fig. 5. The dark lines correspond to the real boundaries and the grey regions to the learning errors. We can observe that the regions of ECOC-ONE (Fig. 5(a)) are better defined. Note that two different dense random matrices with the same distance create different decision boundaries that do not approximate well the expected boundaries (Fig. 5(d) and (e)).

In order to analyze the fitting of the selected dichotomizers of the ECOC-ONE matrix to the classes boundaries, the volume of the errors for the one-versus-all and ECOC-ONE technique are shown in Fig. 6. The height corresponds to the number of times that one technique misclassifies a data sample for each spatial location. Observe that the volume of the one-versus-all technique (Fig. 6(b)) is in this case about 70% higher than the one generated by the ECOC-ONE strategy (Fig. 6(a)).

5. Experimental results

In order to validate the proposed method, we use the well-known UCI database [19]. The description of the selected data sets is shown in Table 5. We compare our technique with the following ECOC coding strategies: one-versus-all ECOC (one-vs-all), one-versus-one ECOC (one-vs-one), and dense

Table 4
ECOC strategy hits for a toy problem

One-versus-one ECOC		One-versus-all ECOC		Dense random ECOC		ECOC-ONE	
Hit	#D	Hit	#D	Hit	#D	Hit	#D
70.83 ± 1.17	6	66.67 ± 1.07	4	67.67 ± 1.91	5	$\mathbf{72.92 \pm 0.82}$	5

#D means number of dichotomizers.

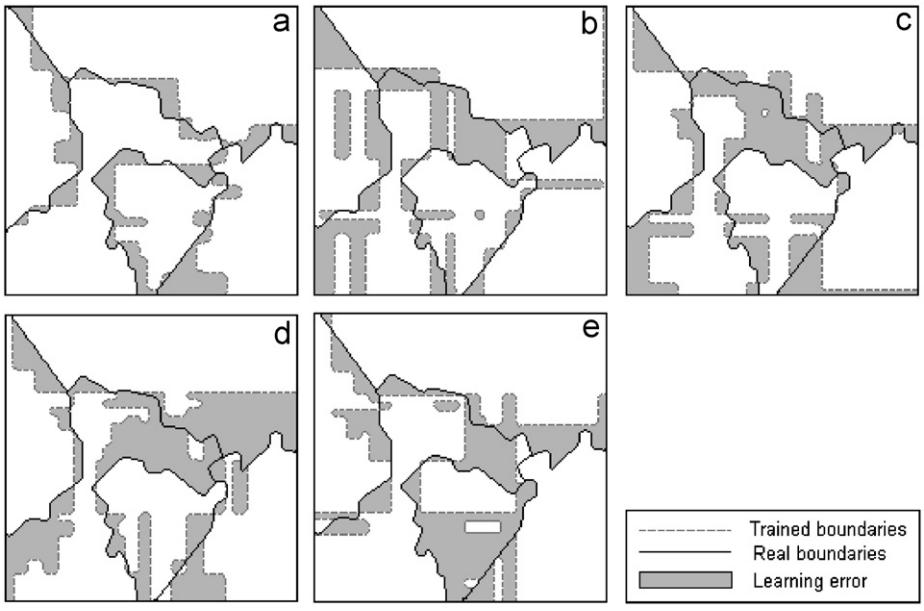


Fig. 5. Boundaries resulted after one iteration of training. (a) ECOC-ONE, (b) one-versus-one, (c) one-versus-all and, (d) and (e) two different matrices of dense random with the same minimal distance, respectively. Dark line corresponds to the real boundary and grey regions correspond to learning errors.

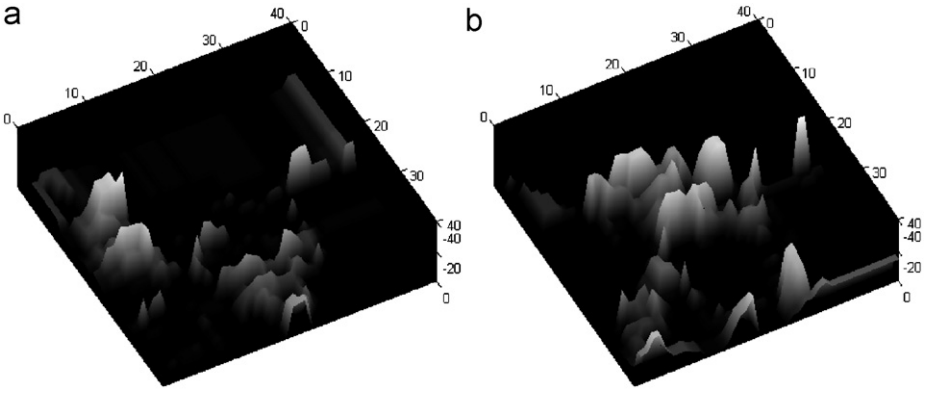


Fig. 6. Error surface comparison between ECOC-ONE and one-versus-all technique for the toy problem of Fig. 3.

random ECOC.[6] The decoding process for all mentioned techniques is the standard Euclidean distance because it shows the same behavior as the Hamming decoding, but it also tends to reduce the confusion due to the use of the zero values [13]. All these strategies are compared with our ECOC-ONE method extending a tree for coding and our weighted Euclidean distance for decoding. We also include the results obtained by the ECOC-ONE computed with the MSFFS. We use a maximum of 10 iterations or dichotomizers including the first optimal tree. In order to have a fair comparison, we used the same number of dichotomizers for the generation of the dense random ECOC matrix columns. The dense random matrix is selected from an exhaustive search of 10 000 iterations. We have used discriminant analysis, Discrete Adaboost with 50

[6] We choose dense random coding because it is more robust than the sparse technique when the number of columns is small [6].

Table 5
UCI repository databases characteristics

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

decision stumps, and linear support vector machines[7] as base learners for all techniques.[8] Nevertheless, note that our technique is generic in the sense that it only uses the classification score—it is independent of the particular base classifier. All the tests are calculated using stratified 10-fold cross-validation.

Tables 6–8 show the number of dichotomizers, accuracy rates and confidence intervals at 95%—we have tested for statistical significance using a two tailed t-test—for the FLDA, Adaboost and SVM techniques, respectively. The results in boldface are related to the first position in ranking of the methods which confidence interval overlaps with the one with the best accuracy—and therefore not statistically significant from the maximum mean accuracy. The rank shows the average position of each technique. For example, if a technique obtains the best accuracy in 8 of 10 validation sets and it has been chosen as a second option in the other two sets, its rank value is 1.20. Note that all strategies with results not statistically significant from the top one are considered also as the first choice. Observing the results, we can see that our method is very competitive when compared to the other standard ECOC coding techniques. Furthermore, it attains a comparable accuracy to the one-versus-one ECOC coding strategy, which is known to usually obtain the best results. In some cases, one-versus-one improves our results for a certain database. For example, at Pendigits database using FLDA, it obtains a two percent of improvement over our method. However, one must note that one-versus-one requires 45 dichotomizers in that database, and we use only 10. These results are easily explained by the fact that our method chooses at each step the most discriminable dichotomy compared to the one-versus-one strategies where all pairs of classifiers are considered. Thus, our procedure allows to classify classes depending on their difficulty. For example, two difficult classes will have a high Hamming distance between rows. But two easy classes, perhaps will not have a considerable Hamming or Euclidean distance between them, since it is not necessary to correct so many errors.In this way, we can reduce the number of binary classifiers to be selected. This effect can also be seen in the results of dense random ECOC and our procedure. Both cases have the same number of dichotomizers (or less in our case due to the fact that we analyze the training convergence), and although random ECOC has a higher distance between rows in most cases, our procedure usually obtains a higher hit ratio because the dichotomizers are selected in an optimal way depending on the domain of the problem. Note that the results obtained using MSFFS are usually very close to the ones obtained with the exhaustive approach. As expected, its performance is poorer than using exhaustive search. There is a trade-off between accuracy and computing time. If ECOC-ONE with exhaustive search and one-versus-one are the first choices, ECOC-ONE with MSFFS is a very close second choice. Note that there is a further trade-off in the exactitude of the MSFFS method between the optimality of the solution and the time complexity. This trade-off is governed by the number of iterations of the floating search procedure. A maximum of N iterations (where N is the number of items in the search) should suffice to obtain a good approximation [18].

5.1. Discussions

In order to provide more insight on the ECOC-ONE process, we show different experiments that address the following issues: Firstly, we discuss the use of the validation subset. Then, we show the optimality of our extension technique when it is compared with a random extension. We show an extension of the one-versus-all technique using ECOC-ONE. We compare a multiclass built-in SVM with the ECOC-ONE extension of a tree. The computational complexity of the ECOC-ONE is compared to the ECOC-ONE (MSFFS). And finally, We discuss the effect of the weights in the ECOC matrix.

In order to show the effect of the validation set, we focus on the results obtained on two data sets, the dermatology and the glass sets. Figs. 7(a) and (b) display the error evolution using our procedure. Observe that the training error is zero in both cases at iteration 5. At that point further learning using the training subset is futile. However, using the validation set we still have information for accuracy improvement. In fact, looking at test evolution we can see how the test error further decreases. In general, this behavior holds even if the training error does not achieve the zero error, since the validation subset is used as an external oracle. The oracle tries to capture the variability not observed in the training set. In this way, it reinforces the learning process, serving just as an observable test.

The second experiment is designed to show the optimality of our extension technique. We increase the initial one-versus-all with the embedding of only two extra dichotomizers. Discrete Adaboost is used as a base classifier for the comparative. We compare our extension to the one-versus-all including two dense random dichotomizers (one-versus-all-dense) that

[7] The regularization parameter C has been set to 1 for all the experiments. We have selected this parameter after a preliminary set of experiments. We decided to keep the parameter fixed for sake of simplicity and easiness of replication of the experiments, though we are aware that this parameter might not be optimal for all data sets. Nevertheless, since the parameters are the same for all the compared methods any weakness in the results will also be shared.

[8] The comparative with the multiclass adaboost has been omitted due to the fact that the Adaboost. MH algorithm, that has dominated other proposals in empirical studies [16], converts the N_c-class problem into that of estimating a two-class classifier on a training set N_c times as large. Thus, it is essentially the same that the one-versus-all scheme that we analize in the framework of error correction.

Table 6
ECOC strategy hits for UCI databases using FLDA as a base classifier

#	One-versus-one		One-versus-all		Dense random		ECOC-ONE		MSFFS	
	Hit	#D	Hit	#D	Hit	#D	Hit	#D	Hit	#D
(a)	96.65 ± 0.73	15	94.87 ± 0.74	6	96.57 ± 0.74	10	**98.48 ± 0.49**	7.8	96.03 ± 0.97	10
(b)	**82.40 ± 1.46**	28	71.85 ± 1.53	8	**81.15 ± 1.55**	10	**83.90 ± 1.23**	10	**81.73 ± 2.14**	10
(c)	**76.76 ± 1.16**	21	44.55 ± 2.15	7	44.83 ± 2.00	10	52.10 ± 2.28	10	51.65 ± 1.87	10
(d)	**85.24 ± 0.57**	21	71.32 ± 0.62	7	73.92 ± 0.56	10	**85.44 ± 0.50**	9.2	**84.65 ± 1.05**	10
(e)	**71.20 ± 1.27**	55	23.87 ± 0.42	11	41.32 ± 1.38	10	53.05 ± **0.80**	10	51.04 ± **1.42**	10
(f)	81.00 ± 0.67	15	65.35 ± **0.52**	6	75.85 ± 0.83	10	**82.85 ± 0.54**	9.4	80.48 ± 0.85	10
(g)	**52.21 ± 0.80**	45	30.54 ± 0.90	10	47.32 ± 0.93	10	**51.21 ± 0.70**	10	**50.67 ± 1.35**	10
(h)	**93.18 ± 0.43**	45	33.10 ± 1.23	10	68.41 ± 1.44	10	**91.21 ± 0.78**	10	**91.03 ± 1.23**	10
Rank	**1.25**		3.87		2.75		**1.25**		2.12	

Table 7
ECOC Strategy hits for UCI databases using Discrete Adaboost as a base classifier

#	One-versus-one		One-versus-all		Dense random		ECOC-ONE		MSFFS	
	Hit	#D	Hit	#D	Hit	#D	Hit	#D	Hit	#D
(a)	**96.30 ± 0.61**	15	92.65 ± 1.23	6	**95.26** ± 0.82	10	**95.17 ± 0.74**	8.2	**95.11 ± 0.71**	10
(b)	**78.05 ± 1.46**	28	**77.10 ± 1.19**	8	**77.65 ± 1.33**	10	**78.15 ± 1.84**	10	**77.14 ± 1.55**	10
(c)	**67.93 ± 1.66**	21	60.83 ± 2.34	7	63.69 ± 2.51	10	**67.03 ± 1.63**	10	**66.55 ± 1.76**	10
(d)	**97.01 ± 0.72**	21	92.89 ± 1.16	7	94.51 ± 1.22	10	**96.23 ± 1.52**	9.6	94.38 ± 1.84	10
(e)	**81.43 ± 1.12**	55	73.33 ± 1.40	11	74.50 ± 1.96	10	**81.50 ± 1.22**	10	**80.83 ± 2.53**	10
(f)	**86.23 ± 0.79**	15	81.99 ± 0.86	6	84.39 ± 0.76	10	**85.47 ± 1.00**	9.8	**84.67 ± 2.17**	10
(g)	**52.35 ± 1.05**	45	**51.48 ± 1.08**	10	**51.82 ± 1.47**	10	**52.87 ± 1.96**	10	**52.87 ± 1.96**	10
(h)	**98.01 ± 1.04**	45	93.98 ± 2.56	10	**95.54 ± 1.71**	10	**97.84 ± 1.13**	10	**97.09 ± 1.56**	10
Rank	**1.00**		2.37		1.50		**1.00**		**1.25**	

Table 8
ECOC Strategy hits for UCI databases using SVM as a base classifier

#	One-versus-one		One-versus-all		Dense random		ECOC-ONE		MSFFS	
	Hit	#D	Hit	#D	Hit	#D	Hit	#D	Hit	#D
(a)	**96.02 ± 0.95**	15	**94.83 ± 1.84**	6	**95.94 ± 1.22**	10	**95.83 ± 0.94**	8.7	**95.72 ± 1.01**	10
(b)	**76.11 ± 1.26**	28	63.97 ± 1.51	8	72.94 ± 1.37	10	**75.68 ± 1.28**	10	**74.75 ± 1.48**	10
(c)	**58.52 ± 2.63**	21	49.73 ± 2.45	7	54.13 ± 2.73	10	**57.83 ± 1.93**	10	**56.79 ± 1.21**	10
(d)	**98.36 ± 1.47**	21	94.36 ± 1.13	7	93.83 ± 1.43	10	**97.84 ± 1.12**	9.2	**96.84 ± 1.52**	10
(e)	**73.18 ± 1.15**	55	32.07 ± 1.62	11	46.00 ± 1.34	10	**69.14 ± 3.01**	10	67.65 ± 4.02	10
(f)	87.43 ± 0.80	15	85.85 ± 1.08	6	84.03 ± 1.49	10	**89.04 ± 0.63**	10	**88.01 ± 0.97**	10
(g)	**55.31 ± 1.47**	45	41.41 ± 1.79	10	51.07 ± 2.12	10	**52.58 ± 1.73**	10	**52.49 ± 2.13**	10
(h)	**98.53 ± 1.03**	45	95.04 ± 1.88	10	**96.44 ± 1.12**	10	**98.43 ± 0.99**	10	**96.05 ± 1.76**	10
Rank	1.13		2.62		2.25		**1.00**		1.13	

maximally increase the distance between rows and columns (and its complementaries). We can observe in Table 9 that with the reduced set of optimal extra dichotomizers, our proposed technique increases considerably the accuracy of the initial coding technique. Besides, the extension of ECOC-ONE dichotomizers seems to perform better than the extra dense dichotomizers of the comparative.

One-versus-all is considered, in general, one of the poorest choices for learning with ECOC. However, it is still used because of the small number of dichotomizers involved. The third experiment showed in this section compares the extension of the one-versus-all adding just two dichotomizers using our method with the one-versus-one approach—recall that one-versus-one is the standard technique with highest accuracy. In order to perform this comparison we have used Discrete Adaboost on the UCI repository. Table 10 shows the results of these experiments. Observe that both methods achieve the same performance considering the confidence interval at 95%. Note also that the number of dichotomizers involved in our extension is smaller than the one-versus-one approach.

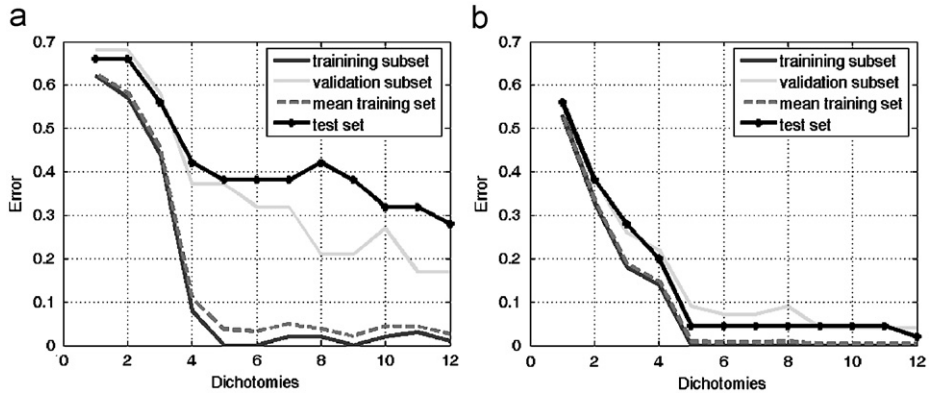


Fig. 7. Error evolution of dermatology database using ECOC-ONE with FLDA: (a) error evolution for the glass data set, (b) error evolution for the dermatology data set.

Table 9
UCI one-versus-all extension using Discrete Adaboost

Problem	One-versus-all ECOC	One-versus-all-dense ECOC	One-versus-all ECOC-ONE
Dermatology	92.65 ± 1.23	93.85 ± 1.02	**95.53 ± 0.89**
Ecoli	**77.10 ± 1.19**	**77.58 ± 1.54**	**78.43 ± 1.02**
Glass	60.83 ± 2.34	**65.59 ± 2.52**	**64.90 ± 2.39**
Segmentation	92.89 ± 1.16	**94.80 ± 1.21**	**95.90 ± 1.03**
Vowel	73.33 ± 1.40	74.97 ± 1.40	**79.34 ± 1.40**
Satimage	81.99 ± 0.86	**83.93 ± 1.11**	**84.83 ± 0.96**
Yeast	51.48 ± 1.08	51.48 ± 1.08	**53.52 ± 0.89**
Pendigits	**93.98 ± 2.56**	**95.64 ± 1.89**	**96.88 ± 2.01**
Rank	2.50	1.38	**1.00**

Table 10
UCI one-versus-one and one-versus-all-ECOC-ONE comparison

Problem	One-versus-one ECOC		One-versus-all ECOC-ONE	
	Hit	#D	Hit	#D
Dermatology	**96.30 ± 0.61**	15	**95.53 ± 0.89**	8
Ecoli	**78.05 ± 1.46**	28	**78.43 ± 1.02**	10
Glass	**67.93 ± 1.66**	21	**64.90 ± 2.39**	9
Segmentation	**97.01 ± 0.72**	21	**95.90 ± 1.03**	9
Vowel	**81.43 ± 1.12**	55	**79.34 ± 1.40**	13
Satimage	**86.23 ± 0.79**	15	**84.83 ± 0.96**	8
Yeast	**52.35 ± 1.05**	45	**53.52 ± 0.89**	12
Pendigits	**98.01 ± 1.04**	45	**96.88 ± 2.01**	12

Table 11
UCI ECOC-ONE with SVM and built-in multiclass SVM with lineal kernel comparative

Problem	ECOC-ONE	Multiclass SVM
Dermatology	**95.83 ± 0.94**	**96.52 ± 0.61**
Ecoli	**75.68 ± 1.28**	69.74 ± 0.76
Glass	**57.83 ± 1.93**	**59.93 ± 1.99**
Segmentation	**97.84 ± 1.12**	95.23 ± 0.59
Vowel	69.14 ± 3.01	**77.55 ± 0.96**
Satimage	**89.04 ± 0.63**	85.60 ± 0.40
Yeast	**52.58 ± 1.73**	**52.57 ± 0.92**
Pendigits	**98.43 ± 0.99**	**98.72 ± 0.17**
Rank	**1.12**	1.38

In order to further validate our approach, we provide a new experiment comparing the ECOC-ONE technique using support vector machines with linear kernels with a built-in multiclass SVM [20] with the same kernel. The results are shown in Table 11. Observe that our technique slightly improves the accuracy of the multiclass SVM using the same parametrization for both techniques.

In order to reduce the computational complexity of the exhaustive search when the number of classes is high, we propose the use of the modified sequential forward floating search (MSFFS). We have designed an experiment that shows the difference in complexity between the MSFFS and the exhaustive search. Using the Pendigits data set, we compute the time of finding a sub-optimal column of the ECOC matrix as the number of classes increases.

Fig. 8 illustrates the results of the experiment. Observe the exponential behavior of the exhaustive search and the quasi-linear tendency of the MSFFS. As we have shown in the former section, the results using this sub-optimal search technique are very similar to those obtained using the exhaustive search.

As commented in former sections, the dichotomizers are selected in an optimal way in order to ensure generalization of the proposed approach. Each of the selected dichotomizers corrects a certain partition of the subset of classes and has

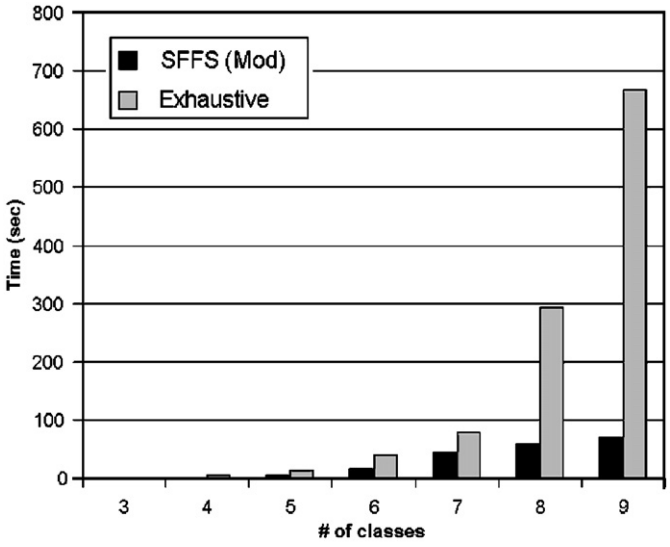


Fig. 8. Time consumed by the exhaustive search and MSFFS.

associated an error according to the training and validation subset of misclassified samples. We use the classification score to weight each dichotomy using the empiric error of classification for that dichotomy using Eq. (2). Fig. 9 shows the average and relative improvement of the weighted Euclidean distance referred to the error obtained using just the Euclidean distance. Besides, we present the figures that reflect the effect of the weighted distance (Table 12).The results show that the weighting scheme increases the accuracy in all cases, showing the absolute and relative improving percentages. Besides, we can observe that the variance is clearly reduced by the fact that in all cases—except for the Ecoli dataset—the confidence rate is smaller.

Table 12
Accuracy of the Euclidean and weighted Euclidean decoding at UCI databases using Discrete Adaboost and $N \times 2$ columns, N being the number of classes, and dense random coding

	Dermatology	Ecoli	Glass	Segmentation
Euclidean	96.74 ± 0.79	78.39 ± 1.43	62.59 ± 2.74	95.38 ± 1.51
Weighted	$\mathbf{96.85 \pm 0.73}$	$\mathbf{79.29 \pm 1.53}$	$\mathbf{64.48 \pm 2.60}$	$\mathbf{96.22 \pm 1.20}$
% Absolute	+0.11	+0.90	+1.89	+0.84
% Relative	+0.11	+1.15	+3.02	+0.88
	Vowel	Satimage	Yeast	Pendigits
Euclidean	78.10 ± 2.38	85.80 ± 1.49	54.73 ± 1.66	96.95 ± 1.05
Weighted	$\mathbf{78.53 \pm 2.32}$	$\mathbf{87.50 \pm 1.03}$	$\mathbf{55.00 \pm 1.46}$	$\mathbf{97.15 \pm 0.95}$
% Absolute	+0.43	+1.70	+0.27	+0.20
% Relative	+0.55	+1.98	+0.49	+0.21

6. Conclusion

In most of the ECOC coding strategies, the ECOC matrix is pre-designed, using a fixed number of dichotomizers independent on the considered domain. We introduced a new coding and decoding strategy called ECOC-ONE, based on the extension of an initial optimal tree upgraded with a set of optimal dichotomizers. Furthermore, the ECOC-ONE can be seen as a general extension strategy for any initial coding matrix. The procedure shares classifiers among classes in the ECOC-ONE matrix and selects the best partitions weighed by their relevance. In this way, it reduces the overall error for a given problem. Moreover, using the validation set, the generalization performance is increased and overfitting is avoided. We show that this technique improves in most cases the standard ECOC technique results, though it has a smaller value of Hamming distance among the coding matrix rows. This improvement is due to the fact that the optimal dichotomizers selected at each step of the method are locally focused on the difficult cases of classification. We compete with the one-versus-one ECOC strategy using far less number of dichotomizers. As a result, a compact—small number of classifiers—multiclass recognition technique with improved accuracy is presented with very promising results. We are currently extending the ECOC analysis focusing on the non-previously analyzed effect of the zero symbol in the ECOC framework. We are studying the way in which that symbol affects to the ternary decoding step, and adapting decoding strategies to avoid the problem of the confusion errors generated in these cases.

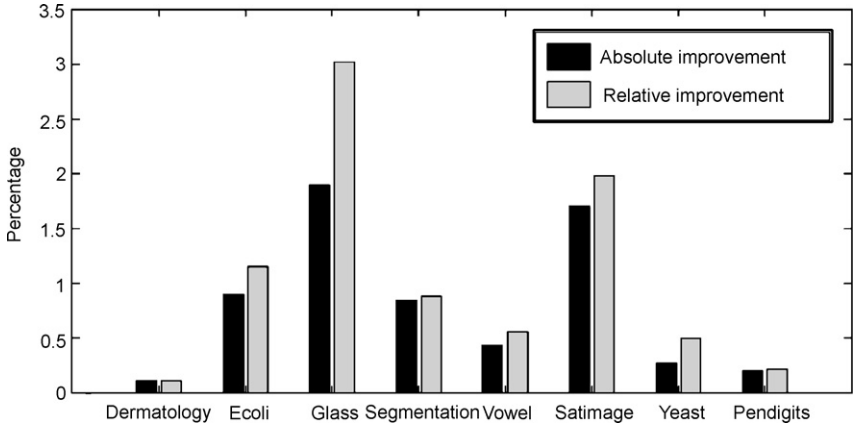


Fig. 9. Absolute and relative percentage improvement comparison between Euclidean distance and weighted Euclidean distance.

References

[1] T. Hastie, R. Tibshirani, Classification by pairwise grouping, The Annals of Stat. 26 (5) (1998) 451–471.

[2] N.J. Nilsson, Learning Machines, McGraw-Hill, New York, 1965.

[3] T. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, J. Arti. Intell. Res. 2 (1995) 263–286.

[4] T. Windeatt, T. Ghaderi, Coding and decoding strategies for multi-class learning problems, Inf. Fusion 4 (1) (2003) 11–21.

[5] J. Zhou, C. Suen, Unconstrained numeral pair recognition using enhanced error correcting output coding: a holistic approach, 1 (2005) 484–488, doi:10.1109/ICDAR.2005.246.

[6] E. Allwein, R. Schapire, Y. Singer, Reducing multiclass to binary: a unifying approach for margin classifiers, J. Mach. Learn. Res. 1 (2002) 113–141.

[7] J. Rennie, ifile: An application of machine learning to e-mail filtering, KDD-2000 workshop on Text Mining.

[8] R. Ghani, Using error-correcting codes for text classification, in: Proceedings of International Conference on Machine Learning'00, 2000, pp. 303–310.

[9] J. Kittler, R. Ghaderi, T. Windeatt, G. Matas, Face verification using error correcting output codes, in: International Conference on Computer Vision and Pattern Recognition'01, 2001 pp. 755–560.

[10] S. Escalera, O. Pujol, P. Radeva, Forest extension of error correcting output codes and boosted landmarks, in: Proceedings of International Conference on Pattern Recognition'06, vol. 3, 2006, pp. 104–107.

[11] W. Utschick, W. Weichselberger, Stochastic organization of output codes in multiclass learning problems, Neural Comput. 13 (5) (2001) 1065–1102.

[12] K. Crammer, Y. Singer, On the learnability and design of output codes for multiclass problems, Mach. Learn. 47 (2) (2002) 201–233.

[13] O. Pujol, P. Radeva, J. Vitrià, Discriminant ecoc: a heuristic method for application dependent design of error correcting output codes, IEEE Tran. Pattern Anal. Mach. Intell. 28 (6) (2006) 1001–1007.

[14] T. Dietterich, G. Bakiri, Error-correcting output codes: a general method for improving multiclass inductive learning programs, in: A. Press (Ed.), Ninth National Conference on Artificial Intelligence, 1991, 572–577.

[15] H. Madala, A. Ivakhnenko, Inductive Learning Algorithm for Complex Systems Modelling, CRC Press, Boca Raton, 1994.

[16] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting, Ann. Stat. 38 (2) (1998) 337–374.

[17] A. Torralba, K. Murphy, W. Freeman, Sharing visual features for multiclass and multiview object detection, in: International Conference on Computer Vision and Pattern Recognition, 2004, pp. 762–769.

[18] P. Pudil, J. Novovičová, J. Kittler, Floating search methods in feature-selection, Pattern Recognition Lett. 15 (11) (1994) 1119–1125.

[19] P. Murphy, D. Aha, Uci repository of machine learning databases, Irvine, CA: University of California, Department of Information and Computer Science.

[20] OSU-SVM-TOOLBOX, ⟨http://svm.sourceforge.net⟩.

About the author—ORIOL PUJOL received the Ph.D. degree in Computer Science from Universitat Autonoma de Barcelona in 2004. Currently, he is a lecturer at Universitat de Barcelona. His main research interest includes basic statistical machine learning techniques for object recognition and medical imaging analysis.

About the author—SERGIO ESCALERA received the BS and MS degrees from Universitat Autonoma de Barcelona in 2003 and 2005, respectively. He is currently working towared the Ph.D. degree in Computer Science. His research interests include machine learning and object recognition.

About the author—PETIA RADEVA has received her Ph.D. at UAB on Development of physics-based models applied to image analysis. Currently, Petia Radeva is an associate professor in the Computer Science Department of the Universitat Autónoma de Barcelona. Her present research interest is concentrated on Development of physics-based and statistical approaches for object recognition, medical image analysis and industrial vision.