Available online at www.sciencedirect.com

SCIENCE DIRECT®

Neurocomputing 57 (2004) 189–214

NEUROCOMPUTING

www.elsevier.com/locate/neucom

An experimental analysis of the dependence among codeword bit errors in ECOC learning machines

Francesco Masulli[a,c], Giorgio Valentini[b,c,*]

[a]Dipartimento di Informatica, Università di Pisa, Via F. Buonarroti 2, Pisa 56127, Italy
[b]DSI - Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano, Via Comelico 39, Milano 20135, Italy
[c]Istituto Nazionale per la Fisica della Materia, Via Dodecaneso 33, Genova 16146, Italy

Received 12 February 2002; received in revised form 29 June 2003; accepted 25 September 2003

Abstract

One of the main factors affecting the effectiveness of error correcting output coding (ECOC) methods for classification is the dependence among the errors of the computed codeword bits. We present an extensive experimental work for evaluating the dependence among output errors of the decomposition unit in ECOC learning machines. In particular, we apply measures based on mutual information to compare the dependence of ECOC multi-layer perceptron (ECOC MLP), made up by a single multi-input multi-output MLP, and ECOC ensembles made up by a set of independent and parallel dichotomizers (ECOC PND). Moreover, the experimentation analyzes the relationship between the architecture, the dependence among output errors and the performances of ECOC learning machines. The results show that the dependence among computed codeword bits is significantly smaller for ECOC PND, pointing out that ensembles of independent parallel dichotomizers are better suited for implementing ECOC classification methods. The experimental results suggest new architectures of ECOC learning machines to improve the independence among output errors and the diversity between base learners.

Keywords: Error correcting output coding; ECOC ensembles of learning machines; Multiple classifier systems; Dependence among output errors in learning machines; Mutual information

[*] Corresponding author. Tel.:+39-010-3536709; fax: +39-010-3536699.
E-mail addresses: masulli@di.unipi.it (F. Masulli), valenti@disi.unige.it (G. Valentini).

0925-2312/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2003.09.011

1. Introduction

Error correcting output coding (ECOC) [19] is a an output coding (OC) decomposition method [30,26] that has been successfully applied to several classification problems [1,11,20,6,33]. OC methods decompose a multiclass-classification problem in a set of two-class subproblems, and then recompose the original problem combining them to achieve the class label.

ECOC methods present several open problems such as the tradeoff between error recovering capabilities and learnability of the dichotomies induced by the decomposition scheme [2,43]. A connected problem is the analysis of the relationship between codeword length and performances [20], while the selection of optimal dichotomic learning machines and the design of optimal codes for a given multiclass problem are open questions that are still subject to active research [16].

Another issue tackled by several works [24,21] is the relationship between performances of ECOC and dependence among output errors. In the framework of coding theory, Peterson [34] has shown that the error recovering capabilities of ECOC codes hold if there is a low correlation among codeword bits. We qualitatively identified the dependence among output errors as one of the factors which influences the effectiveness of ECOC decomposition methods [27]. In that work we hypothesized a higher dependence among codeword bit errors in *monolithic error correcting output coding* (ECOC *monolithic*) [19,27] compared with *ECOC parallel non-linear Dichotomizer* (ECOC *PND*) [28] learning machines, considering that ECOC *monolithic* share the same hidden layer of a single MLP, while *PND* dichotomizers, implemented by a separate MLP for each codeword bit, have their own layer of hidden units, specialized for a specific dichotomic task.

The main goal of the paper consists in understanding the relationships between the performances and the dependence between codeword bit errors in ECOC learning machines. In particular our aim is to analyze and to unravel the relationships between the architecture of the ECOC learning machines, the dependence among the errors of the decomposition unit and the resulting performances. This analysis requires a quantitative evaluation of the dependence among codeword bit errors. To this purpose we apply mutual information–based measures of dependence among output errors proposed in [29], interpreting the dependence among the output errors as the common information shared among them. These measures assess the dependence among the output errors considering their probability distributions, and in this sense they are more refined measures of dependence compared with the standard index of correlation or the rank order correlation coefficient. In particular, they can offer insights into the dependence and the probability distribution of the errors and can also be used to compare the dependence among output errors between different learning machines in order to select a model well-suited to a particular learning problem.

The proposed analysis of the dependence among codeword bit errors represents a novel application of mutual information to a machine learning problem. Information theory and in particular mutual information had been applied to several machine learning problems, such as modeling of self organized systems and feature maps [25,9], feature transformation and selection [7,40], image processing [8,41], independent component analysis [14], evaluation of the relations between output independence and complementariness in multiple classifier decision systems [36].

In this paper an extensive experimental comparison between different architectures of ECOC learning machines is accomplished, evaluating the relationships between the accuracy of the predictions and the dependence between codeword bit errors. In particular, ECOC *monolithic* [19,27] and ECOC *PND* [28] are compared using synthetic and UCI data sets [31], and a specific test of hypothesis [29] is applied to evaluate whether a significant statistical difference in the dependence among the codeword bit errors between the two ECOC learning machines does exist.

The paper is organized as follows. In the next section we outline ECOC methods and their main related open problems. In Section 3 we summarize the main characteristics of the measures based on mutual information to evaluate the dependence among output errors. Section 4 presents the experimental setup, the results and the discussion. The conclusions summarize the main results and the incoming developments of this work.

2. ECOC methods for classification

In this section we summarize the main characteristics and open problems of ECOC methods for classification.

2.1. ECOC methods overview

A k classes classification problem, (or *K-polychotomy*) consists in evaluating an unknown function $\mathrm{P}: \mathbf{X} \to \mathscr{C}$, where $\mathbf{X} \subseteq \mathbb{R}^d$ is the multidimensional space of the features and $\mathscr{C} = \{c_1, \ldots, c_k\}$ are the labels of the classes, using only a limited data set $\mathscr{D} = \{\mathbf{x}_i, c_i\}_{i=1}^N, \ \mathbf{x}_i \in \mathbb{R}^d, c_i \in \mathscr{C}$.

2.1.1. Output coding methods

ECOC is an output coding (OC) decomposition method [17,26] for classification problems. OC methods code classes through binary strings. A coding process is a mapping $\mathrm{M}: \mathscr{C} \to \mathscr{S}$ from the set of the classes to the set of binary strings $\mathscr{S} = \{s_1, \ldots, s_k\}$, where the $s_i \in \{-1, 1\}^l$ are named *codewords*, i.e. binary strings of length l. Each string s_i must univocally determine its corresponding class, i.e. $\forall i, j$ such that $i \neq j, \ 1 \leqslant i, j \leqslant k$ we have $s_i \neq s_j$.

The class coding implicitly generates a decomposition of the k-polychotomy into a set of l dichotomies $f_1, \ldots, f_l$, where l is the length of the codeword coding a class. Each dichotomy f_i subdivides the input patterns into two complementary superclasses $\mathscr{C}_i^+$ and $\mathscr{C}_i^-$, each of them grouping one or more classes of the k-polychotomy. Given a *decomposition matrix* $D = [d_{ik}]$ of dimension $l \times k$ that represents the decomposition, connecting classes $C_1, \ldots, C_k$ to the superclasses $\mathscr{C}_i^+$ and $\mathscr{C}_i^-$ identified by each dichotomy, an element of D is defined as:

$$d_{ik} = \begin{cases} +1 & \text{if } C_k \subseteq \mathscr{C}_i^+, \\ -1 & \text{if } C_k \subseteq \mathscr{C}_i^- \end{cases}$$

Table 1
Decomposition matrix example

Dichotomizers tasks	Columns: class codewords			
	C_1	C_2	C_3	C_4
f_1	+1	−1	+1	−1
f_2	+1	+1	−1	+1
f_3	+1	−1	+1	+1
f_4	−1	−1	+1	+1
f_5	+1	+1	+1	−1
f_6	+1	−1	−1	+1
f_7	−1	+1	+1	+1

In a decomposition matrix, rows correspond to dichotomizer tasks and columns to classes and each class is univocally determined by its *codeword*. For instance, considering a decomposition matrix for a 4 class classification problem with 7-bit class coding (Table 1), the task of the sixth dichotomizer, namely f_6, consists in separating the patterns belonging to classes C_1 and C_4 from the patterns of class C_2 and C_3. The third column of the decomposition matrix represents the codeword $[+1, -1, +1, +1, +1, -1, +1]$ associated to the class C_3.

2.1.2. Decomposition schemes and ECOC

Different *decomposition schemes* have been proposed in literature: In the one-per-class (OPC) decomposition [5], each dichotomizer f_i has to separate a single class from all others; in the *pairwise coupling* (PWC) decomposition [22], the task of each dichotomizer f_i consists in separating a class C_i from class C_j, ignoring all other classes; the *correcting classifiers* (CC) and the *pairwise coupling correcting classifiers* (PWC-CC) are variants of the PWC decomposition scheme, that reduce the noise originated in the PWC scheme due to the processing of non-pertinent information performed by the PWC dichotomizers [32].

Our work focuses on ECOC decomposition methods [19]. These OC decomposition methods try to improve the error correcting capabilities of the codes generated by the decomposition through the maximization of the minimum distance between each couple of codewords [24,26]. Dietterich and Bakiri [18,19] proposed the *error correcting output coding* (ECOC) decomposition scheme with the aim of improving the generalization capabilities of NETtalk classifier systems based on distributed output codes [39]: Coding the classes by codewords suggests the idea of adding *error recovering* capabilities to decomposition methods in order to obtain classifiers less sensitive to noise [24,27]. This goal is achieved by means of the redundancy of the coding scheme, as shown by coding theory [45].

The error-recovering capabilities of ECOC codes depend mainly on column separation, i.e. a codeword must be distanced from the other codewords of the decomposition matrix, according to an assigned measure. For binary strings we can use the Hamming distance. The maximal number of errors $\mathrm{Max}_{\mathrm{err}}$ that can be corrected in an ECOC based decomposition is [24]:

$$\mathrm{Max}_{\mathrm{err}} = \left\lfloor \frac{\Delta_D - 1}{2} \right\rfloor \tag{1}$$

where Δ_D is the minimal Hamming distance between each pair of columns of the decomposition matrix D.

2.1.3. Decomposition and reconstruction

ECOC is a two-stage classification method: After the *decomposition stage*, a *reconstruction stage* performs the decoding of the codeword computed during the decomposition stage in order to output the class label.

In fact learning machines constructed by ECOC are composed of two units:

- *Decomposition unit* analyzes the input patterns and calculates the codewords using an assigned decomposition scheme generated by an appropriate algorithm. This unit computes a function $\mathrm{F} : \mathbb{R}^d \to \mathbb{R}^l$:

  $$\mathrm{F}(x) = [f_1(x), f_2(x), \ldots, f_l(x)] \tag{2}$$

  where $f_i : \mathbb{R}^d \to \mathbb{R},\ 1 \leqslant i \leqslant l$.
- *Decision unit* decodes the computed codeword $\hat{s} = [f_1(x), f_2(x), \ldots, f_L(x)]$, mapping it to the associated class. This unit computes a function $\mathrm{G} : \mathbb{R}^l \to \mathscr{C}$:

  $$\mathrm{G}(\hat{s}) = \mathrm{G}[f_1(x), f_2(x), \ldots, f_L(x)] \tag{3}$$

  where $\mathscr{C}$ is the set of the classes, $f_i(x)$ are the hypotheses returned by the learning algorithm, and G is a suitable decoding function.

The decoding performed by the decision unit depends on the type of output of the decomposition unit. If the outputs are continuous the decision unit computes a function $\mathrm{G} : \mathbb{R}^l \to \mathscr{C}$; if the outputs are discrete, i.e, if the decomposition unit computes a function $\mathrm{F} : \mathbb{R}^d \to \mathbb{B}^l$, where $\mathbb{B} = \{-1, +1\}$, then the decoding unit computes a function $\mathrm{G} : \mathbb{B}^l \to \mathscr{C}$. The decoding function $\mathrm{G}(\hat{s})$ can be implemented by a maximization of a suitable similarity measure between the computed $\hat{s}$ codeword and the effective codewords $s_i,\ 1 \leqslant i \leqslant k$ associated to the classes:

$$\mathrm{G}(\hat{s}) = \arg\max_{1 \leqslant i \leqslant K} Sim(\hat{s}, s_i), \tag{4}$$

where s_i is the codeword of class C_i, the vector $\hat{s}$ is the codeword computed by the set of dichotomizers, and $Sim(x, y)$ is a general similarity measure between two vectors x and y. This similarity measure can be the Hamming distance for dichotomizers with discrete outputs, or are an inner product or one of the L_1 or L_2 norm distances for dichotomizers with continuous outputs.

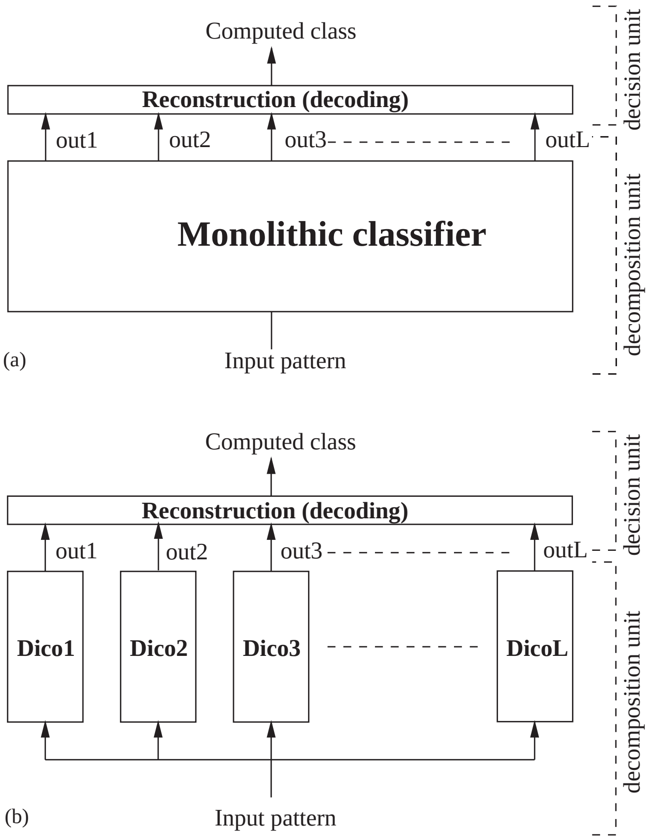

Fig. 1. Design of output coding learning machines: monolithic (a) and parallel ensemble (b).

2.1.4. Design of ECOC classifiers

There are two main approaches to the design of a classifier using OC methods, depending on the characteristics of the Decomposition unit (Fig. 1):

- *Monolithic classifier unit* is composed of a *monolithic* classifier with multiple outputs, exploiting the decomposition in an implicit way. Examples are multiple-input multiple-output (MIMO) learning machines, such as MIMO MLP or MIMO decision trees [18,19].
- *Parallel classifiers unit* is implemented by an ensemble of dichotomizers, assigning each dichotomy to a different dichotomizer. Consequently the learning task is distributed among separated and independent dichotomizers, each learning a different bit of the codeword coding a class. In this case, we call the resulting learning machines *parallel linear dichotomizers* (*PLD*) if the dichotomizers used for implementing the dichotomies are linear (as in [3,27]), or *parallel non-linear dichotomizers* (*PND*) if the dichotomizers are non-linear [19,28].

The good generalization achieved using ECOC methods have been explained through the reduction of both bias and variance [24,11] and by interpreting them as large margin classifiers [38,2]. Application of ECOC methods in several domains have shown improvements over standard k-way classification methods. For instance ECOC was successfully applied to classify cloud types [1], for text classification [11,20], for text-to-speech synthesis [6], to classify olive oils by means of electronic noses [33], and for the molecular diagnosis of multiple tumor types using gene expression data [42].

2.2. Open problems

ECOC methods present several open problems. The tradeoff between error recovering capabilities and complexity/learnability of the dichotomies induced by the decomposition scheme have been tackled in several works [2,43], but an extensive experimental evaluation of the tradeoff has to be performed in order to achieve a better comprehension of this phenomenon.

A related problem is the analysis of the relationship between codeword length and performances: some preliminary results seem to show that long codewords improve performance [20].

Another open problem, not sufficiently investigated in literature [20,27,11], is the selection of optimal dichotomic learning machines for the decomposition unit.

Several methods for generating ECOC codes have been proposed: exhaustive codes, randomized hill climbing [19], Hadamard and BCH codes [12,34], and random codes [23], but open problems are still the joint maximization of distances between rows and columns of the decomposition matrix.

Another open problem consists in designing codes for a given multiclass problem. An interesting greedy approach is proposed in [30], and a method based on soft weight sharing to learn error correcting codes from data is presented in [4]. In [16] it has been shown that given a set of dichotomizers the problem of finding an optimal decomposition matrix is NP-complete: by introducing continuous codes and casting the design problem of continuous codes as a constrained optimization problem, we can achieve an optimal continuous decomposition using standard optimization methods.

In [27] we have highlighted that the effectiveness of ECOC decomposition methods depends also on the design of learning machines implementing the decision unit, on the similarity of the ECOC codewords, on the accuracy of the dichotomizers, on the complexity of the multiclass learning problem and on the dependence of the codeword bits. Consequently, if a decomposition matrix contains very similar rows (dichotomies), each error of an assigned dichotomizer will be likely to appear in the most correlated dichotomizers, thus reducing the effectiveness of ECOC. In this paper we address the problem of quantitatively evaluating the dependence among output errors of the decomposition unit of ECOC learning machines, in order to analyze the relationship between dependence among output errors and performances. To achieve this goal a suitable measure of dependence among outputs and among output errors must be defined.

3. Mutual information based measures of dependence among output errors

In this section we introduce mutual information–based measures to evaluate the dependence among output errors in learning machines. A more detailed discussion can be found in [29].

3.1. Mutual information and dependence among output errors

Our goal consists in evaluating the independence among output errors of a learning machine. For instance, considering the output errors e_1 and e_2 of a two-output learning machine, we want to evaluate if $p(e_1, e_2) = p(e_1)p(e_2)$, where p is the density probability function associated to the random variables e_1 and e_2. Using standard statistical measures such as the covariance or the coefficient of correlation, we estimate only the linear relationship between e_1 and e_2. Conversely, a suitable measure of dependence must directly evaluate the probability distribution of the output errors in order to properly estimate the stochastic independence between random variables. Mutual information, being a special case of the Kullback-Leibler divergence between two distributions, measures the matching between the joint probability density distribution and the product of the marginal probability density distribution of the output errors. If we have a complete matching, the mutual information is 0 and the output errors are independent, otherwise the higher the value of the mutual information between output errors is, the higher the dependence between them will be.

Put another way, the main idea behind the evaluation of dependence among output errors of learning machines through mutual information based measures consists in interpreting the dependence among the outputs as the common information shared between them. Consequently, if the information conveyed by each output is similar to that of other outputs, a dependence can be checked through mutual information based measures.

In order to define a suitable measure, we have to define the error on the outputs. More precisely, consider the estimation of an unknown function $\mathbf{f}(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^l$ using a limited data set $\mathscr{D} = \{(\mathbf{x}^{(i)}, \mathbf{c}^{(i)})\}_{i=1}^N$, where $\mathbf{x}^{(i)} \in \mathbb{R}^d$ and $\mathbf{c}^{(i)} \in \mathbb{R}^l$. We represent the correct outputs as $\mathbf{c} = [c_1, c_2, \ldots, c_l]$ and the computed outputs of a learning machine as $\hat{\mathbf{c}} = [\hat{c_1}, \hat{c_2}, \ldots, \hat{c_l}]$. Then we define the corresponding output errors as $\mathbf{e} = [e_1, e_2, \ldots, e_l]$, where e_i expresses the error on the ith output of the learning machine, such as the absolute error $e_i = |c_i - \hat{c_i}|$, $\forall i = 1 \ldots l$ or the quadratic error.

Let us consider the overall output error $\mathbf{e} = [e_1, e_2, \ldots, e_l]$. In order to compute the dependence among the output errors, we have to divide the range of each e_i in b intervals $bin(j)$, $1 \leqslant j \leqslant b$:

$$bin = \{[k_0, k_1), [k_1, k_2), \ldots, [k_{b-1}, k_b]\}$$

with $0 = k_0 < k_1 < k_2 < \cdots < k_b = max$. The jth interval is selected by

$$bin(j) = [k_{j-1}, k_j) \quad j = 1 \ldots b, \quad k_{j-1}, k_j \in [0, max]$$

For instance, in the simplest case we have only two intervals: $bin = \{[k_0, k_1),\ [k_1, k_2]\}$. The intervals $bin(j)$ are of equal width, except for the first one which can have a different width.

We define $e_k^{(i)}$ as the error of the kth output relative to the ith input example, and e_{kj} as the number of examples whose values $e_k^{(i)}$ fall in the interval $bin(j)$:

$$e_{kj} = |\{i \in [1, N] \mid e_k^{(i)} \in bin(j)\}|,$$

where N is the cardinality of the data set and we define also the *discrete probability function* $p(e_{kj})$ of e_{kj}:

$$p(e_{kj}) = \frac{|\{i \in [1, N] \mid e_k^{(i)} \in bin(j)\}|}{N}$$

and the *discrete joint probability function* among all the output errors:

$$p(e_{1j_1}, e_{2j_2}, \ldots, e_{lj_l}) = \frac{|\{i \in [1, N] | \bigwedge_{1 \leqslant u \leqslant l} (e_u^{(i)} \in bin(j_u))\}|}{N}$$

where $j_u \in \{1, \ldots, b\}$.

3.2. Global measures based on mutual information

We can now evaluate the mutual information among the output errors. If we have l outputs, we define the *mutual information error* I_{E} as

$$I_{\mathrm{E}}(e_1, \ldots, e_l) = \sum_{j_1=1}^{b} \cdots \sum_{j_l=1}^{b} p(e_{1j_1}, \ldots, e_{lj_l}) \log\left(\frac{p(e_{1j_1}, \ldots, e_{lj_l})}{p(e_{1j_1}) \ldots p(e_{lj_l})}\right). \tag{5}$$

The mutual information error (Eq. (5)) expresses the dependence among all output errors of a learning machine. If it is equal to 0 then the distributions of the output errors are statistically independent. It expresses also how similar the probability distribution of the output errors are.

The outputs of a learning machine can be considered correct if their errors are below a certain threshold, i.e if $\forall i,\ e_i < \delta, \delta > 0$. As a consequence, the first interval $bin(1) = [0, k_1)$ defines the range of tolerance for the correct output, where $k_1 = \delta$. Therefore an output affected by an error lower than δ is interpreted as a correct one. Considering the output errors only when two or more errors spring from the outputs, and disregarding all cases with no errors or with only one error, we can also define the *mutual information specific error* I_{SE}:

$$I_{\mathrm{SE}}(e_1, \ldots, e_l) = \sum_{\mathscr{J}} p(e_{1j_1}, \ldots, e_{lj_l}) \log\left(\frac{p(e_{1j_1}, \ldots, e_{lj_l})}{p(e_{1j_1}) \ldots p(e_{lj_l})}\right), \tag{6}$$

where

$$\mathscr{J} = \{[j_1, \ldots j_l] \,|\, \exists (j_v, j_w) | \,(j_v \neq 1) \wedge (j_w \neq 1) \wedge (v \neq w)\}$$

with $v, w \in \{1, \ldots, l\}$.

Then, if we have l outputs, all cases with $l-2$ correct outputs or less are considered. It is worth noting that I_{SE} is not in a proper sense a mutual information among random variables according to the information theory, but it expresses the dependence among two or more output errors of a learning machine, disregarding the mutual information error due to only a single error or no errors on the outputs.

3.3. Pairwise measures based on mutual information

To evaluate the dependence among specific pairs of output errors, we introduce the *pairwise mutual information error matrix* R composed by the elements $I_{\mathrm{E}}(e_i, e_j) = [R_{ij}]$ and the *pairwise mutual information specific error matrix* S, composed by the elements $I_{\mathrm{SE}}(e_i, e_j) = [S_{ij}]$. We define also two other global indices: the *pairwise mutual information error matrix index* Φ_R:

$$\Phi_R = \sum_{i=1}^{l} \sum_{j=1}^{l} I_{\mathrm{E}}(e_i, e_j) \tag{7}$$

and the *pairwise mutual information specific error matrix index* Φ_S:

$$\Phi_S = \sum_{i=1}^{l} \sum_{j=1}^{l} I_{\mathrm{SE}}(e_i, e_j). \tag{8}$$

These indices measure the sum of the mutual information error and the mutual information specific error between all the output pairs of the learning machines, and in this sense can be regarded as global measures of dependence between output errors. Note that these indices (Eqs. (7) and (8)) are not equivalent to the corresponding Eqs. (5) and (6) of the mutual information among all output errors: Eqs. (7) and (8) consider only the mutual information between pairs of output errors, while Eqs. (5) and (6) consider the overall mutual information among all output errors.

These mutual-information related quantities can be used to compare the dependence of the output errors among different learning machines on the same learning problem, using, of course, the same data sets.

4. Experimental analysis

In this section we present an extended experimental work we performed in order to test the following hypothesis: *ECOC parallel non-linear dichotomizers show a lower dependence among the output errors of their decomposition unit compared with the output errors of the corresponding ECOC monolithic multi-layer perceptron.*

In order to verify this hypothesis we performed a quantitative comparison of the dependence among output errors of the decomposition unit of ECOC MLP and ECOC *PND* learning machines. We also analyzed the relationship between performances, architecture and dependence among output errors between these two ECOC learning machines.

Table 2
Data sets general features

Data set	Number of attributes	Number of classes	Number of training samples	Number of testing samples
d5	3	5	30000	30000
glass	9	6	214	10-fold cross-val
letter	16	26	16000	4000
optdigits	64	10	3823	1797

Table 3
Performance of MLP ECOC monolithic and *PND* ECOC ensemble on four data sets

	MLP ECOC monolithic			PND ECOC ensemble		
Data set	min	mean	stdev	min	mean	stdev
d5	13.27	18.31	6.44	11.91	12.34	0.74
glass	33.18	36.17	4.54	30.37	32.05	1.77
letter	4.95	6.55	1.91	3.05	3.24	0.24
optdigits	2.61	3.08	0.47	1.89	1.95	0.10

In particular we made an experimental comparison of the mutual information error I_{E}, the mutual information specific error I_{SE} and the pairwise indices Φ_R and Φ_R (Section 3) of the ECOC *monolithic* and *PND* learning machines using different data sets.

4.1. Experimental setup

We used four different data sets: the first one, d5,[1] was generated using NEURObjects, a C++ software library for neural networks development [44], and the other three, *glass*, *letter* and *optdigits* were taken from the UCI machine learning repository of Irvine [31]. The synthetic data set d5 is made up by five three-dimensional classes, each composed by two disjoint clusters of data: the data points for each class were drawn from two normal distributions with equal probability and different diagonal covariance matrix. The main characteristics of the data sets are shown in Table 2.

In order to perform training and testing of the considered learning machines, we applied multiple runs of different random initializations of the weights using a single pair of training and testing data sets and *k-fold cross validation* [13] methods. The results are summarized in Table 3: Errors on the test set are expressed as percent rates, and for each data set the minimum (min), average (mean), and standard deviation (stdev) of the error is given.

[1] d5 is available on-line at ftp://ftp.disi.unige.it/person/ValentiniG/Data.

After the training, we used only the outputs of the decomposition units of the learning machines. Then we computed the errors, obtaining the matrices of output errors (Section 3): their lines are the vectors of output errors on all outputs relative to a single sample, and their columns are the errors on a single output of the overall samples.

Using these error data we computed and compared the mutual information error I_{E} (Eq. (5)) and the mutual information specific error I_{SE} (Eq. (6)) among all the outputs of the learning machines. Then, we computed and compared the mutual information error matrices R, the mutual information specific error matrices S (Section 3), and the their relative global indices Φ_R and Φ_S (Eqs. (7) and (8)).

We used *NEURObjects* [44], both to train the learning machines and to evaluate the dependence among the output errors.

We compared the dependence among output errors of ECOC *monolithic* and ECOC *PND* learning machines varying the structure (number of hidden units), the number of intervals (bins) of the output errors, and the values of the output error tolerance δ (Section 3). For each data set and for a fixed number of hidden units we have considered all the combinations of $\delta \in \{0.1, 0.2, 0.3, 0.4\}$ with the number of intervals $bins \in \{2, 3, 4, 5, 6\}$, for a total of 20 pairs of $(\delta, bins)$.

For the data set d5 we used 11 different structures for the learning machine, varying the number of hidden units between 5 to 50, yielding to $11 \times 20 = 220$ evaluations of I_{E}, I_{SE}, Φ_R and Φ_S both for ECOC *monolithic* and ECOC *PND* learning machines. For the UCI data sets *glass*, *letter* and *optdigits* we used only 2 different structures, using, respectively, 5 and 9, 120 and 140, 60 and 70 hidden units, yielding to $2 \times 20 = 40$ evaluations of the mutual information error based quantities both for ECOC *monolithic* and ECOC *PND* learning machines.

The ECOC codes generated for *letter* and *optdigits* data sets lead up to learning machines with respectively 30 and 14 outputs. The computation of I_{E} and I_{SE} requires the storage of l-dimensional matrices composed by $(bins)^l$ elements. Note that selecting only 2 intervals would result in having huge joint probability matrices with 2^{30} and 2^{14} elements, requiring a large amount of data, not available for these data sets, in addition to an intractable amount of space and time computational resources. It is worth noting that this problem is a form of *curse of dimensionality* [10], that can be avoided computing only the global pairwise indices Φ_R and Φ_S. To evaluate the dependence between specific pairs of output errors, we also compared R and S matrices between ECOC *monolithic* and *PND* considering a single triplet structure, numbers of bins and δ.

4.2. Results and discussion

Now we present the results of the comparison of I_{E} and I_{SE} among all outputs, of the Φ_R and Φ_S pairwise indices and the comparison of R and S matrices considering a particular triplet for each data set.

4.2.1. Comparing I_{E} and I_{SE} among codeword bit errors

In Figs. 2(a) and (b) we compare I_{E} among all output errors of the *monolithic* and ECOC *PND* learning machines on the data sets d5 (a) and *glass* (b). On the axes

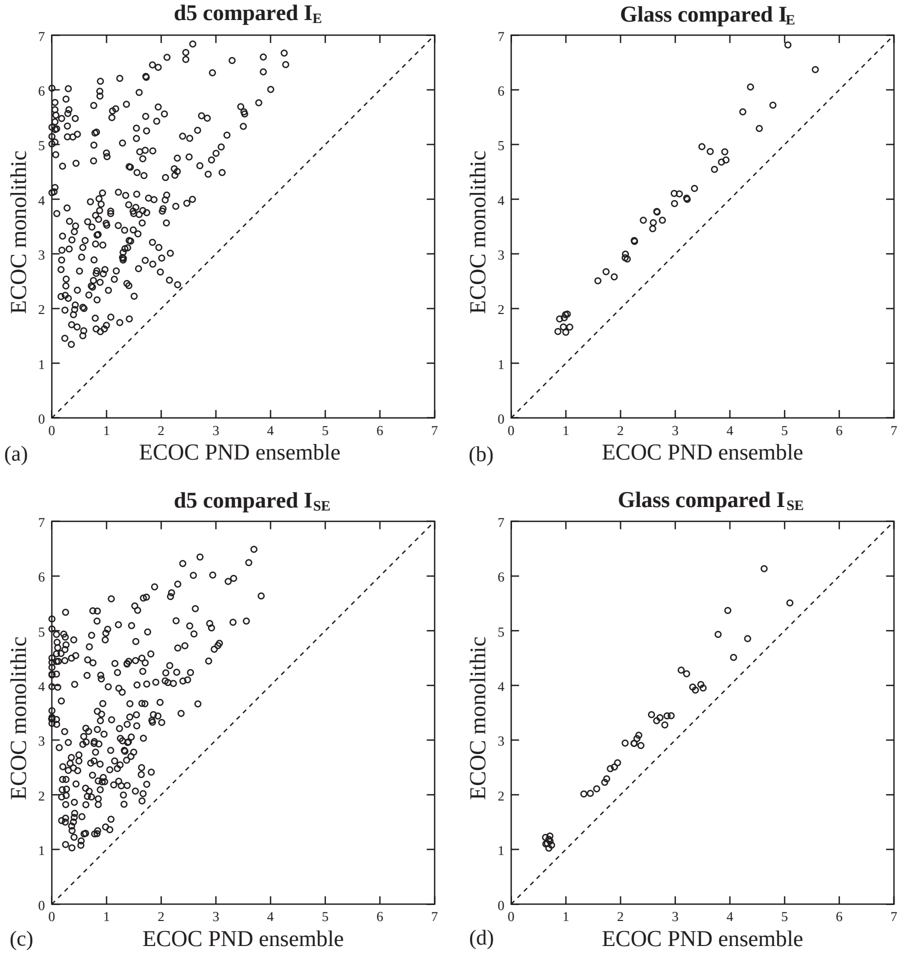

Fig. 2. Compared mutual information error I_E ((a) and (b)), and mutual information specific error I_{SE} ((c) and (d)), for d5 and glass data sets.

are represented the computed I_E values. Each point corresponds to a different triplet of hidden units, number of intervals and values of δ. All points are above the dotted line, showing that the mutual information error I_E is greater for ECOC *monolithic* with respect to ECOC *PND*, no matter the structure, the number of intervals and the δ values used.

These results are confirmed in Figs. 2(c) and (d), representing the comparison of the mutual information specific error I_{SE} among all the outputs on the same data sets d5 and *glass*. In all the 220 comparisons on the data set d5 (Fig. 2(c)) and the 40 comparisons on the data set *glass* (Fig. 2(d)), I_{SE} is greater for ECOC *monolithic* with respect to ECOC *PND* learning machines.

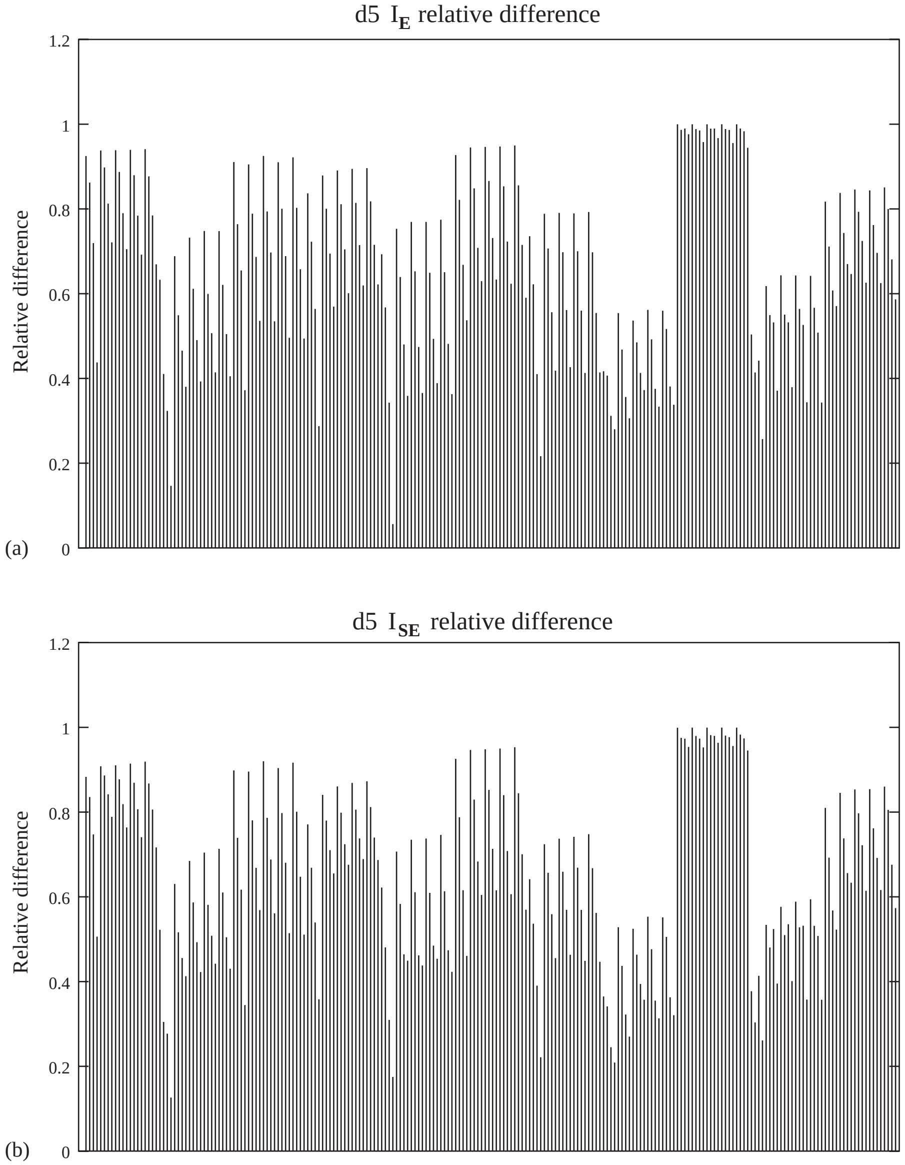

Fig. 3. Relative difference of the mutual information error I_E (a) and of the mutual information specific error I_{SE} (b) among all outputs between ECOC *monolithic* and *PND* learning machines for the d5 data set.

Fig. 3 shows the relative difference of the mutual information error I_E (a) and of the mutual information specific error I_{SE} (b) among all outputs between *monolithic* and *PND* ECOC learning machines on the d5 data set. More precisely, each line represents relative differences I_E^{rel} or I_{SE}^{rel} of I_E and I_{SE} between ECOC *monolithic* and ECOC *PND* with respect to the I_{E} and I_{SE} of the ECOC *monolithic* learning machine:

$$I_{\mathrm{E}}^{\mathrm{rel}} = \frac{I_{\mathrm{E}}(monolithic) - I_{\mathrm{E}}(PND)}{I_{\mathrm{E}}(monolithic)}$$

$$I_{\mathrm{SE}}^{\mathrm{rel}} = \frac{I_{\mathrm{SE}}(monolithic) - I_{\mathrm{SE}}(PND)}{I_{\mathrm{SE}}(monolithic)}. \tag{9}$$

Each vertical line corresponds to a different triplet number of hidden units, number of intervals and value of δ. The 11 "spires" (Figs. 3(a) and (b)) correspond to 11 different structures (i.e. number of hidden units) of the learning machines. Inside each group the values of the number of intervals and the values of δ are varied in an ordered way, respectively from 2 to 6 and from 0.4 down to 0.1. The most significant fact is that all the values are positive, showing that the correlation among all the output errors is greater for ECOC *monolithic* with respect to ECOC *PND*.

Similar results are obtained for the *glass* data set, as also in this case all the I_{E} and I_{SE} differences are positive.

Due to the dimensional problems described above in this section, I_{E} and I_{SE} values have not been computed on *letter* and *optdigits*. For these data sets we evaluated only the pairwise global indices Φ_R and Φ_S.

4.2.2. Comparing pairwise mutual information indices

Let us consider now the pairwise mutual information error matrices R and S and in particular their associated pairwise mutual information error indices Φ_R and Φ_S (Eqs. (7) and (8)). These matrices can be computed element by element, considering a different pair of outputs each time; in this way the 30×30 and 14×14 matrices for the data sets *letter* and *optdigits* can also be considered and the corresponding indices Φ_R and Φ_S can be used as surrogate of the I_{E} and I_{SE} values among all output errors.

Fig. 4 shows the compared mutual information error-matrix indices Φ_R between *monolithic* and *PND* ECOC learning machines considering 4 different data sets. On the axes are represented the Φ_R values of ECOC *monolithic* and ECOC *PND* learning machines. Each point corresponds to a triplet of hidden units, number of intervals and values of δ. On all the data sets about all the points are above the dotted line, i.e. all the values of Φ_R are greater for ECOC *monolithic* compared with ECOC *PND*. Concerning the compared Φ_S indices, we can outline that almost all the points are above the equality line on all the data sets (Fig. 5). In particular, considering the relative differences Φ_R^{rel} and Φ_S^{rel} of the pairwise mutual information error index Φ_R and of the pairwise mutual information specific error index Φ_S:

$$\Phi_R^{\mathrm{rel}} = \frac{\Phi_R(monolithic) - \Phi_R(PND)}{\Phi_R(monolithic)}$$

$$\Phi_S^{\mathrm{rel}} = \frac{\Phi_S(monolithic) - \Phi_S(PND)}{\Phi_S(monolithic)} \tag{10}$$

we note that only 2 of the 220 cases give negative values for the data set d5, while for all the remaining data sets all their values are positive.

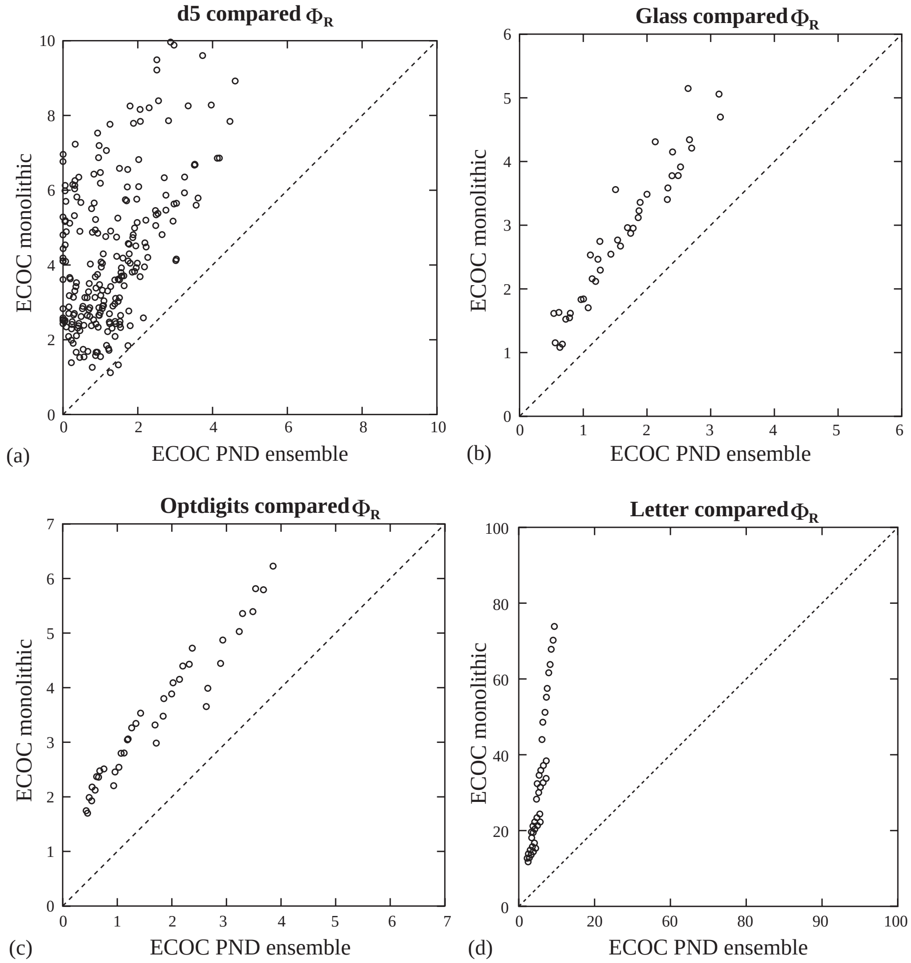

Fig. 4. Compared mutual information error matrix indices Φ_R between ECOC *monolithic* and *PND* learning machines for d5 (a), glass (b), optdigits (c) and letter (d) data sets.

Coming back to Figs. 4 and 5, clusters of points can be observed, especially in the *optdigits* (Figs. 4(c) and 5(c)) and *letter* (Figs. 4(d) and 5(d)) plots, and in a less noticeable way also in *glass* (Figs. 4(b) and 5(b)). Focusing on *optdigits*, Figs. 6(a) and (b) show that the clusters depend mainly on δ values, with increasing values of Φ_R and Φ_S when δ decreases. Moreover, the spread of the points inside each cluster depends on the number of bins, showing an increment of the Φ_R and Φ_S values when the number of bins increase. The 10 points in each cluster corresponds to 5 different number of bins and 2 different number of hidden units. Note that the spread of ECOC *PND* inside each cluster is lower compared to ECOC *monolithic*, revealing in such a way a lower sensibility to the interval partition. The role of the number of intervals in

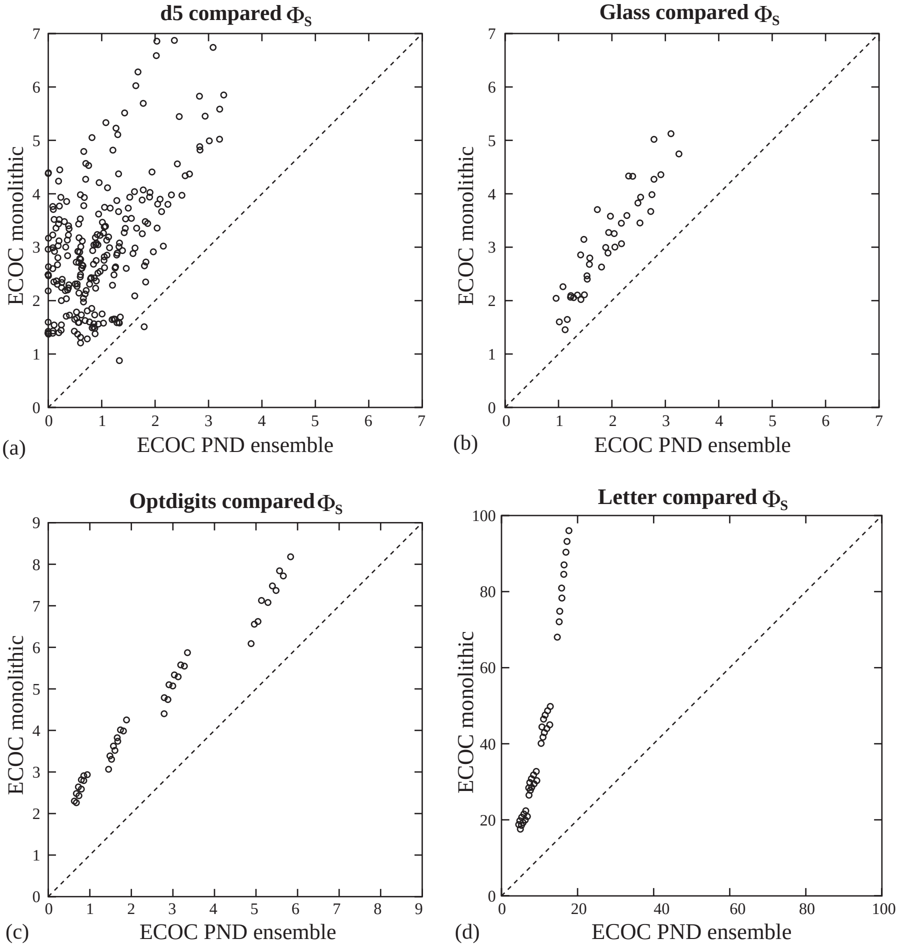

Fig. 5. Compared mutual information specific error matrix indices Φ_S between ECOC *monolithic* and *PND* learning machines for d5 (a), glass (b), optdigits (c) and letter (d) data sets.

determining the values of the pairwise global indices is outlined in Fig. 6(c) concerning the *letter* data set. This figure highlights how the structure, the number of bins and the values of δ affect the pairwise mutual specific error index Φ_S considering its difference between ECOC *monolithic* and ECOC *PND* learning machines.

In the optdigits data set, the width δ affects the relative differences of Φ_R and Φ_S between ECOC *monolithic* and ECOC *PND* learning machines (Fig. 6(c)). On the contrary, even if in the data sets *letter* and *glass* we can observe slightly higher relative differences for δ=0.1, neither δ nor the number of intervals affect the relative differences in a significant way (data not shown).

However, ECOC *monolithic* learning machines show greater Φ_R and Φ_S values, no matter what the number of hidden units, intervals and values of δ are selected.

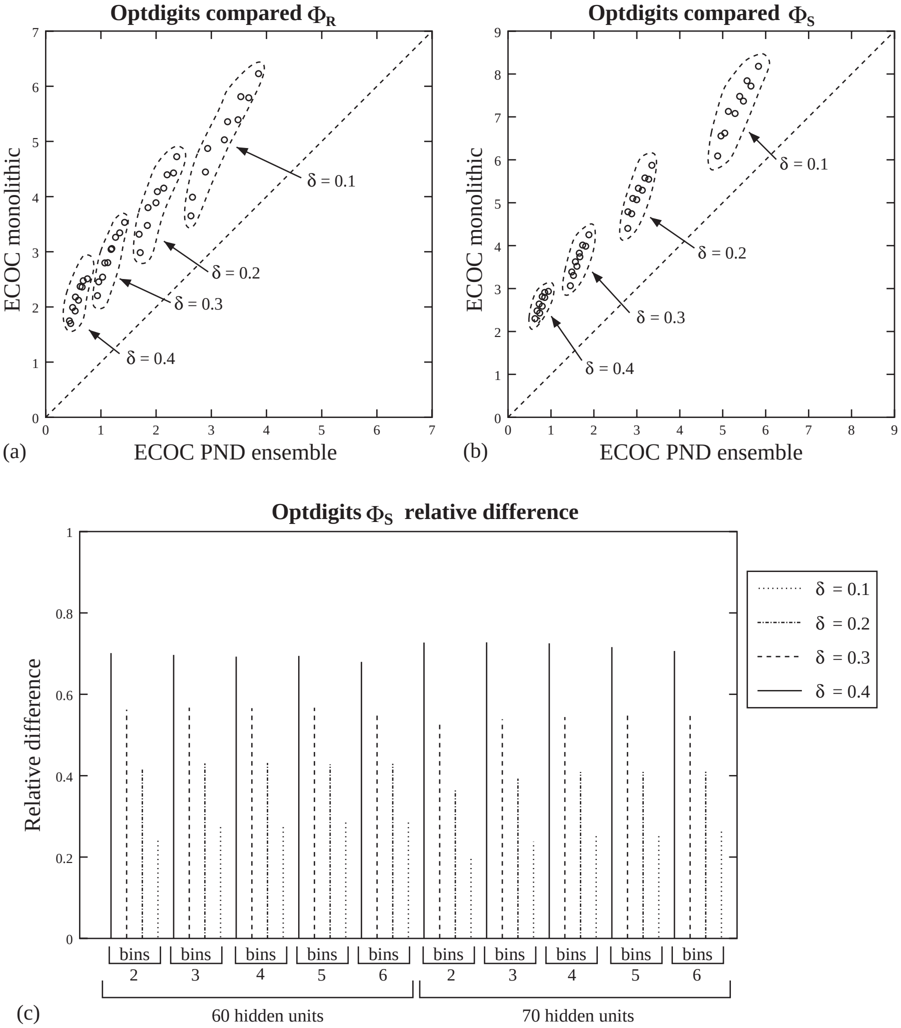

Fig. 6. Cluster of points depending on δ values of the compared pairwise mutual information error index Φ_R (a) and of the compared pairwise mutual information specific error index Φ_S (b) on the optdigits data set. Relative differences of the pairwise mutual information specific error index Φ_S between ECOC *monolithic* and *PND* learning machines for the optdigits data set (c).

4.2.3. Comparing mutual information error matrices

The examination of the pairwise mutual information error matrices can provide us with information about the dependence of specific pairs of output errors. In addition we can also directly compare the matrices of different learning machines to synthetically evaluate the dependence among all the output pairs. As an example, we consider the matrices R and S, selecting a triplet with $\delta = 0.4$, a number of intervals equal to 6 for all the data sets used in the experimentation and with a fixed number of hidden units for each data set. In particular we shall study one of the points in Figs. 4 and 5, corresponding to a pair of matrices relative to the ECOC *monolithic* and ECOC *PND* learning machines.

Fig. 7 represents the mutual information matrices for the d5 data set. On the left column the R matrices for ECOC *monolithic* (a), ECOC *PND* (b) and their difference (c) are shown. On the right column are represented the S matrices for ECOC *monolithic* (d), ECOC *PND* (e) and their difference (f). Each tridimensional bar matches a pair of output errors and corresponds to their mutual information error I_{E} or their mutual information specific error I_{SE}. The S and R matrices are represented as triangular matrices, without the diagonal, since they are symmetric and the elements on the diagonal are the entropy of the output errors. Gray bars stand for positive values, and black for negative ones. We can observe that all the values of the R difference matrix are positive (Fig. 7(c)), and in the S difference matrix only on the pair of outputs 2 and 3 we have a negative value (Fig. 7(f)).

Comparing the R and S matrices of ECOC *monolithic* and ECOC *PND* learning machines on the UCI data sets *glass* and *optdigits* (Fig. 8) we obtain similar results. For instance, considering the pairwise mutual information matrices for the *optdigits* data set, only the output error pairs $(1, 13), (3, 14)$ and $(11, 12)$ show negative values for the I_{E} (Fig. 8(c)) and I_{SE} (Fig. 8(f)) differences.

Learning machines with 30 outputs are generated by the ECOC decomposition of the classification problem on the *letter* data set. Considering the differences between R and S matrices, we point out that in all the 435 comparisons of the pairwise I_{E} and I_{SE} the values for ECOC *monolithic* are higher (data not shown).

4.2.4. Dependence among codeword bit errors, performances and design of ECOC classifiers

Fig. 9 shows the relations between error rates and mutual information based measures I_{E} and I_{SE} considering the d5 data set. Both I_{E} and I_{SE} curves of ECOC *PND* ensemble lie below the corresponding curves of ECOC *monolithic* learning machines: These figures confirm that the dependence among output errors is lower for ECOC *PND*. It is worth noting that, as expected, I_{E} and I_{SE} grow with error rates, but their values are mostly related to a specific learning machine architecture. We have seen that all the results relative to the mutual information error I_{E} and the mutual information specific error I_{SE} among all the outputs on the data sets d5 and *glass* show greater values for ECOC *monolithic* respect to ECOC *PND* (Figs. 2 and 3). These results are confirmed by the evaluation of the mutual information error matrix indices Φ_R and Φ_S (Figs. 4 and 5), concerning also the *optdigits* and *letter* data sets. The analysis of the pairwise mutual information matrices R and S converges on showing that nearly all the I_{E} and I_{SE} values between each pair of output errors are greater for ECOC *monolithic* learning machines (Figs. 7 and 8). Moreover, applying the *mutual information error t-test* [29] for evaluating the significance of the differences between the I_{E} and I_{SE} values of the two ECOC learning machines, we verified that in almost all the comparisons we registered a significant difference with a degree of confidence of 95%.

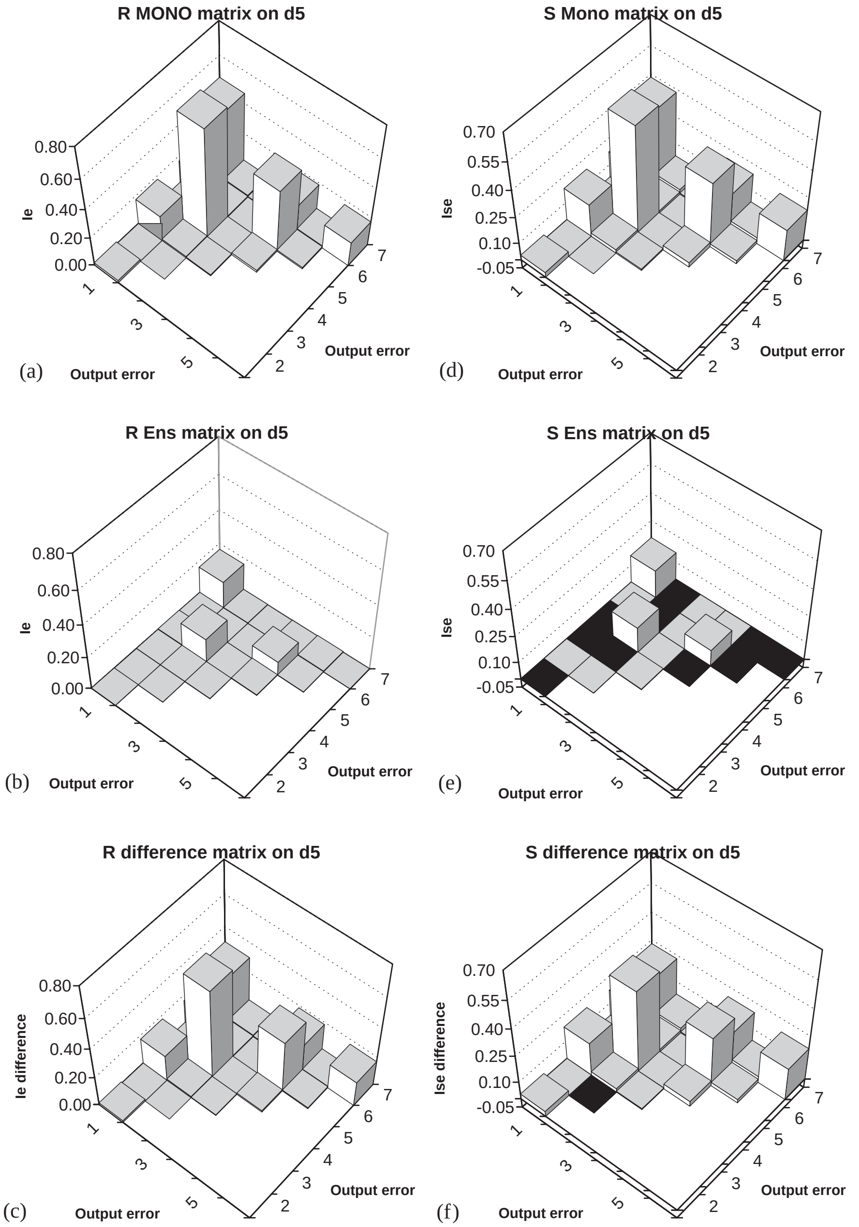

Fig. 7. Pairwise mutual information matrices for the d5 data set. *R matrix* of the ECOC *monolithic* (Mono) learning machine (a), of the ECOC *PND* Ensemble (Ens) learning machine (b), and their difference (c); *S matrix* of the Mono (d) and the Ens (e) learning machines, and their difference (f).

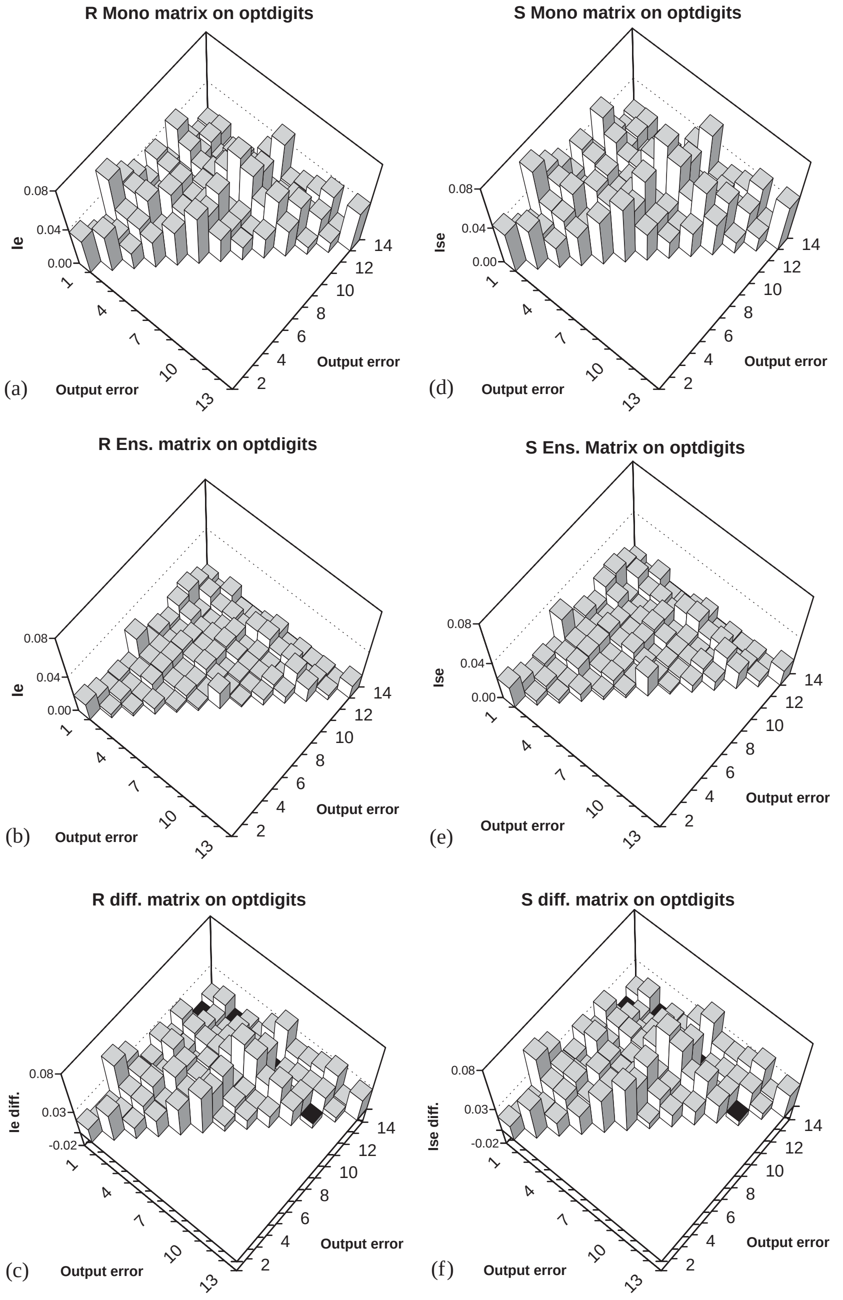

Fig. 8. Pairwise mutual information matrices for the optdigits data set. *R matrix* of the ECOC *monolithic* (Mono) learning machine (a), of the ECOC PND Ensemble (Ens) learning machine (b), and their difference (c); *S matrix* of the Mono (d) and the Ens (e) learning machines, and their difference (f).

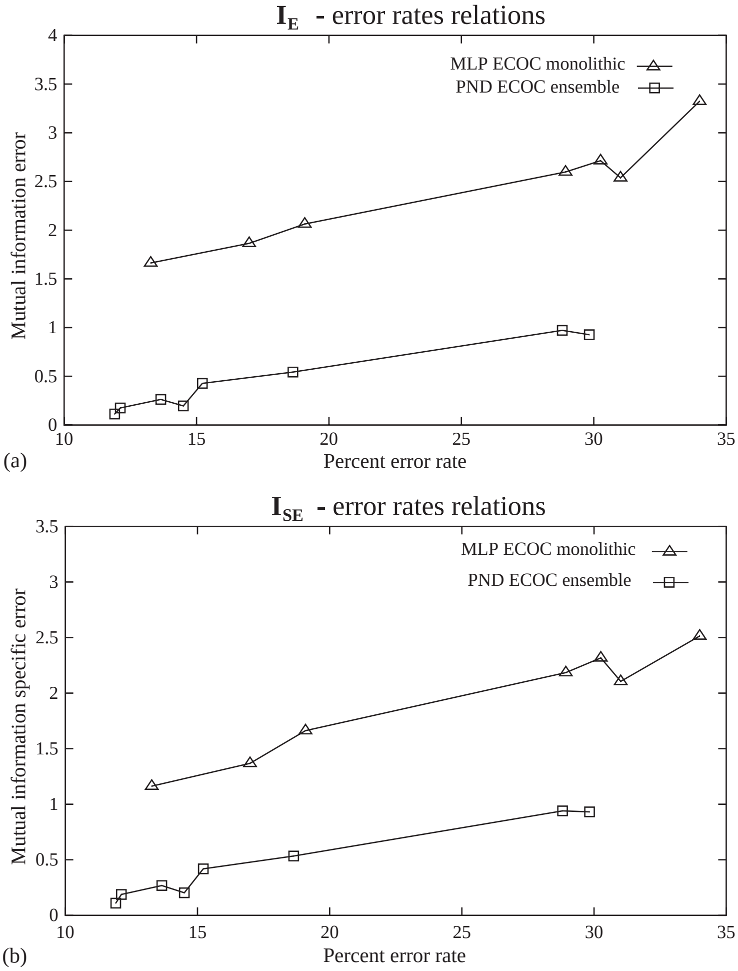

Fig. 9. Relations error rates—mutual information error I_E (a) and error rates—mutual information specific error I_{SE} (b) in ECOC *monolithic* and *PND* learning machines for the d5 data set.

Consequently the hypothesis proposed in Section 4, stating that *ECOC Parallel Non-linear Dichotomizers* show a lower dependence among the output errors of their decomposition unit compared with the output errors of the corresponding ECOC monolithic multi-layer perceptron cannot be rejected by the experimental results on the selected data sets.

The observed difference in the dependence among output errors is related to the different architecture of the two learning machines and in particular to the design of the decomposition unit. Our experimentation shows quantitatively that one of the main factors affecting the effectiveness of ECOC decomposition methods is the dependence among output errors of the decomposition unit. A low dependence can be achieved implementing the decomposition unit through an ensemble of parallel and independent dichotomizers, such as the dichotomic MLP proposed in our experimentation, or other suitable non-linear dichotomizers.

5. Conclusions

The effectiveness of ECOC decomposition methods depends on many factors, including the similarity of the ECOC codewords, the accuracy of the dichotomizers, the complexity of the multiclass learning problem, the design of learning machines implementing the decision units, and the dependence among codeword bits.

While some of these problems have been tackled elsewhere [24,11,16,28,2,43], the proper design of ECOC learning machines and the quantitative evaluation of the dependence among codeword bits have not been adequately addressed.

In this paper we have presented an extensive experimental work to evaluate quantitatively the dependence among codeword bit errors in ECOC learning machines. In particular, we have proposed and used measures based on mutual information to compare the dependence among output errors between ECOC *monolithic* and ECOC *PND* learning machines.

The measurements of the mutual information error I_{E}, the mutual information specific error I_{SE} and the mutual information error matrix indices Φ_R and Φ_S show that ECOC *PND* present a lower dependence among the output errors of their decomposition unit compared with the output errors of the corresponding ECOC *monolithic* MLP.

We have also analyzed the relationship between performance and dependence among output errors, showing that the design of ECOC learning machines affects this relationship. In fact, the results show that *monolithic* architectures are affected by a higher dependence among codeword bit errors leading to a higher generalization error. Our experimental work suggests that a low dependence can be achieved implementing the decomposition unit through an ensemble of parallel and independent dichotomizers, such as the dichotomic MLP proposed in our experiments, or other suitable dichotomizers such as decision trees [35] or support vector machines [15].

Future developments of this work should consist in studying quantitatively the dependence among output errors in ECOC learning machines architectures that can improve the diversity between the dichotomizers implementing the decision unit. In particular, we shall quantitatively study how boosting methods [37] can increase the diversity among the dichotomizers and the independence among output errors in ECOC learning machines.

Acknowledgements

This work has been partially funded by *Progetto Finalizzato* CNR-MADESS II, INFM and University of Genova. We thank the anonymous reviewers for their comments and suggestions.

References

[1] D. Aha, R. Bankert, Cloud classification using error-correcting output codes, in: Artificial Intelligence Applications: Natural Science, Agriculture and Environmental Science, Vol. 11, 1997, pp. 13–28.

[2] E.L. Allwein, R.E. Schapire, Y. Singer, Reducing multiclass to binary: a unifying approach for margin classifiers, J. Mach. Learn. Res. 1 (2000) 113–141.

[3] E. Alpaydin, E. Mayoraz, Combining linear dichotomizers to construct nonlinear polychotomizers, Technical Report IDIAP-RR 98-05, IDIAP - Dalle Molle Institute for Perceptual Artificial Intelligence, Martigny, 1998.

[4] E. Alpaydin, E. Mayoraz, Learning error-correcting output codes from data, In ICANN'99, Edinburgh, UK, 1999, pp. 743–748.

[5] R. Anand, G. Mehrotra, C.K. Mohan, S. Ranka, Efficient classification for multiclass problems using modular neural networks, IEEE Trans. Neural Networks 6 (1995) 117–124.

[6] G. Bakiri, T.G. Dietterich, Achieving high accuracy text-to-speech with machine learning, in: B. Damper (Ed.), Data Mining in Speech Synthesis, Kluwer Academic Publishers, Boston, MA, 2001.

[7] R. Battiti, Using mutual information for selecting features in supervised neural net learning, IEEE Trans. Neural Networks 5 (4) (1994) 537–550.

[8] S. Becker, G.E. Hinton, A self organizing neural network that discovers surfaces in random-dot stereograms, Nature 355 (1992) 161–163.

[9] A.J. Bell, T.J. Sejnowski, An information-maximization approach to blind separation and blind deconvolution, Neural Comput. 6 (1995) 1129–1159.

[10] R. Bellman, Adaptive Control Processes: a Guided Tour, Princeton University Press, New Jersey, 1961.

[11] A. Berger, Error correcting output coding for text classification, in: IJCAI'99: Workshop on Machine Learning for Information Filtering, Stockholm, Sweden, 1999.

[12] R.C. Bose, D.K. Ray-Chauduri, On a class of error correcting binary group codes, Inf. Control (3) (1960) 68–79.

[13] V.N. Cherkassky, F. Mulier, Learning From Data: Concepts, Theory and Methods, Wiley, New York, 1998.

[14] P. Comon, Independent component analysis: a new concept? Signal Process. 36 (1994) 287–314.

[15] C. Cortes, V. Vapnik, Support vector networks, Mach. Learn. 20 (1995) 273–297.

[16] K. Crammer, Y. Singer, On the learnability and design of output codes for multiclass problems, in: Proceedings of the 13th Annual Conference on Computational Learning Theory, Stanford, CA, 2000, pp. 35–46.

[17] T.G. Dietterich, Ensemble methods in machine learning, in: Multiple Classifier Systems, First International Workshop, MCS2000, Cagliari, Italy, Springer, Berlin, 2000, pp. 1–15.

[18] T.G. Dietterich, G. Bakiri, Error-correcting output codes: a general method for improving multiclass inductive learning programs, in: Proceedings of AAAI-91, AAAI Press/MIT Press, Cambridge, MA, 1991, pp. 572–577.

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

[20] R. Ghani, Using error correcting output codes for text classification, in: ICML 2000: Proceedings of the 17th International Conference on Machine Learning, San Francisco, US, Morgan Kaufmann Publishers, Los Altos, CA, 2000, pp. 303–310.

[21] V. Guruswami, A. Sahai, Multiclass learning, boosting, and error-correcting codes, in: Proceedings of the 12th Annual Conference on Computational Learning Theory, ACM Press, New York, 1999, pp. 145–155.

[22] T. Hastie, R. Tibshirani, Classification by pairwise coupling, Ann. Statist. 26 (1) (1998) 451–471.

[23] G. James, Majority vote classifiers: theory and applications, Ph.D. Thesis, Department of Statistics, Stanford University, Stanford, CA, 1998.

[24] E. Kong, T.G. Dietterich, Error-correcting output coding correct bias and variance, in: The XII International Conference on Machine Learning, San Francisco, CA, Morgan Kauffman, Los Altos, CA, 1995, pp. 313–321.

[25] R. Linsker, An application of the principle of maximum information preservation to linear systems, in: D.S. Touretzky (Ed.), Advances in Neural Information Processing Systems, Vol. 1, Morgan Kauffman, San Mateo, CA, 1989, pp. 186–194.

[26] F. Masulli, G. Valentini, Comparing decomposition methods for classification, in: R.J. Howlett, L.C. Jain (Eds.), KES'2000, Fourth International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies, Piscataway, NJ, 2000, pp. 788–791.

[27] F. Masulli, G. Valentini, Effectiveness of error correcting output codes in multiclass learning problems, in: J. Kittler, F. Roli (Eds.), Lecture Notes in Computer Science, Vol. 1857, Springer, Berlin, Heidelberg, 2000, pp. 107–116.

[28] F. Masulli, G. Valentini, Parallel non linear dichotomizers, in: IJCNN2000, The IEEE-INNS-ENNS International Joint Conference on Neural Networks, Vol. 2, Como, Italy, 2000, pp. 29–33.

[29] F. Masulli, G. Valentini, Mutual information methods for evaluating dependence among outputs in learning machines, Technical Report TR-01-02, DISI - Dipartimento di Informatica e Scienze dell' Informazione-Università di Genova, 2001. ftp://ftp.disi.unige.it/person/ValentiniG/papers/TR-01-02.ps.gz.

[30] E. Mayoraz, M. Moreira, On the decomposition of polychotomies into dichotomies, in: The XIV International Conference on Machine Learning, Nashville, TN, July 1997, pp. 219–226.

[31] C.J. Merz, P.M. Murphy, UCI repository of machine learning databases, 1998. www.ics.uci.edu/mlearn/MLRepository.html.

[32] M. Moreira, E. Mayoraz, Improved pairwise coupling classifiers with correcting classifiers, in: C. Nedellec, C. Rouveirol (Eds.), Lecture Notes in Artificial Intelligence, Vol. 1398, Springer, Berlin, Heidelberg, New York, 1998, pp. 160–171.

[33] M. Pardo, G. Sberveglieri, F. Masulli, G. Valentini, Decompositive classification models for electronic noses, Anal. Chim. Acta 446 (2001) 223–232.

[34] W.W. Peterson, E.J. Weldon Jr., Error Correcting Codes, MIT Press, Cambridge, MA, 1972.

[35] J.R. Quinlan, C4.5 Programs for Machine Learning, Morgan Kauffman, Los Altos, CA, 1993.

[36] A. Saranli, M. Demirekler, On output independence and complementariness in rank-based multiple classifier decision systems, Patter Recognition 34 (2001) 2319–2330.

[37] R.E. Schapire, Using output codes to boost multiclass learning problems, in: Proceedings of the 14th International Conference on Machine Learning, San Francisco, CA, Morgan Kauffman, Los Altos, CA, 1997, pp. 313–321.

[38] R.E. Schapire, Y. Freund, P. Bartlett, W. Lee, Boosting the margin: a new explanation for the effectiveness of voting methods, Ann. Statist. 26 (5) (1998) 1651–1686.

[39] T.J. Sejnowski, C.R. Rosenberg, Parallel networks that learn to pronounce English text, J. Artif. Intell. Res. (1) (1987) 145–168.

[40] K. Torkkola, W.M. Campbell, Mutual information in learning feature transformations, in: Proceedings of the ICML'2000, The 17th International Conference on Machine Learning, 2000.

[41] A.M. Ukrainec, S. Haykin, A modular neural network for enhancement of cross-polar radar targets, Neural Networks 9 (1996) 143–168.

[42] G. Valentini, Gene expression data analysis of human lymphoma using support vector machines and output coding ensembles, Artif. Intell. Med. 26 (3) (2002) 283–306.

[43] G. Valentini, Upper bounds on the training error of ECOC-SVM ensembles, Technical Report TR-00-17, DISI - Dipartimento di Informatica e Scienze dell' Informazione - Università di Genova, 2000. ftp://ftp.disi.unige.it/person/ValentiniG/papers/TR-00-17.ps.gz.

[44] G. Valentini, F. Masulli, NEURObjects: an object-oriented library for neural network development, Neurocomputing 48 (1–4) (2002) 623–646.

[45] J. Van Lint, Coding Theory, Springer, Berlin, 1971.

Francesco Masulli is an Associate Professor of Computer Science with the University of Pisa (Italy). He received the Laurea degree in Physics from the University of Genova in 1976. After the military service, he was a researcher with the Italian National Institute for Nuclear Physics (1978–1979), and with the Ansaldo Automazione Co. (1979–1983), and an Assistant Professor with the University of Genova (1983–2001). He was also in leave as a visiting scientist at the University of Nijmegen (Holland) in 1983, and at the International Computer Science Institute in Berkeley, California in 1991, 1993, and 1994. He authored or co-authored more than 100 scientific papers on Machine Learning, Neural Networks, Fuzzy Systems and Ensemble Methods and co-edited three books and two special issues of scientific journals on those subjects. He serves as an Associate Editor the international journal “Intelligent Automation and Soft Computing”. His previous duties include the chairing of the Conference of the International Graphomomics Society (IGS) in 1997, and of the Symposium on Soft Computing SOCO, in 1999. He is member of the IEEE-Neural Network Council (Italian R.I.G.), and a Board Member of the Italian Neural Network Society (SIREN) and of the SIG Italy of the International Neural Network Society (INNS).

Giorgio Valentini is research assistant at the Computer Science Department of the University of Milano. He received the “laurea degree” in Biological Sciences and in Computer Science from the University of Genova, and the Ph.D. in Computer Science from DISI (Dipartimento di Informatica e Scienze dell’ Informazione), University of Genova. He is member of the International Neural Network Society (INNS), of the International Society of Computational Biology (ISCB), of the Italian Association of Artificial Intelligence (AIIA) and of the Italian Neural Network Society (SIREN). His main research interests concern with ensembles of learning machines and bioinformatics.