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

Francesco Masulli\textsuperscript{a,c}, Giorgio Valentini\textsuperscript{b,c,}\textsuperscript{*}

\textsuperscript{a}Dipartimento di Informatica, Università di Pisa, Via F. Buonarroti 2, Pisa 56127, Italy
\textsuperscript{b}DSI - Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano, Via Comelico 39, Milano 20135, Italy
\textsuperscript{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.

© 2003 Elsevier B.V. All rights reserved.

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.
1. Introduction

Error correcting output coding (ECOC) [19] is 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 $P : X \rightarrow C$, where $X \subseteq \mathbb{R}^d$ is the multidimensional space of the features and $C = \{c_1, \ldots, c_k\}$ are the labels of the classes, using only a limited data set $\mathcal{D} = \{(x_i, c_i)\}_{i=1}^N$, $x_i \in \mathbb{R}^d$, $c_i \in C$.

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 $M : C \rightarrow S$ from the set of the classes to the set of binary strings $S = \{s_1, \ldots, s_k\}$, 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 $i \neq j$, $1 \leq i, j \leq k$ 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 $C^+_i$ and $C^-_i$, 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 $C^+_i$ and $C^-_i$ identified by each dichotomy, an element of $D$ is defined as:

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

<table>
<thead>
<tr>
<th>Dichotomizers tasks</th>
<th>Columns: class codewords</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$C_1$</td>
</tr>
<tr>
<td>$f_1$</td>
<td>+1</td>
</tr>
<tr>
<td>$f_2$</td>
<td>+1</td>
</tr>
<tr>
<td>$f_3$</td>
<td>+1</td>
</tr>
<tr>
<td>$f_4$</td>
<td>-1</td>
</tr>
<tr>
<td>$f_5$</td>
<td>+1</td>
</tr>
<tr>
<td>$f_6$</td>
<td>+1</td>
</tr>
<tr>
<td>$f_7$</td>
<td>-1</td>
</tr>
</tbody>
</table>

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 \( \text{Maxerr} \) that can be corrected in an ECOC based decomposition is [24]:

\[
\text{Maxerr} = \left\lfloor \frac{\Delta D - 1}{2} \right\rfloor
\]

(1)

where \( \Delta D \) 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 \( F : \mathbb{R}^d \rightarrow \mathbb{R}^l \):

\[
F(x) = [f_1(x), f_2(x), \ldots, f_l(x)]
\]

(2)

where \( f_i : \mathbb{R}^d \rightarrow \mathbb{R}, \ 1 \leq i \leq l \).

- **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 \( G : \mathbb{R}^l \rightarrow \mathcal{C} \):

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

(3)

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

\[
G(\hat{s}) = \arg \max_{1 \leq i \leq k} \text{Sim}(\hat{s}, s_i),
\]

(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 \( \text{Sim}(x,y) \) 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.
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 $f(x) : \mathbb{R}^d \rightarrow \mathbb{R}^l$ using a limited data set $\mathcal{D} = \{(x^{(i)}, c^{(i)})\}_{i=1}^N$, where $x^{(i)} \in \mathbb{R}^d$ and $c^{(i)} \in \mathbb{R}^l$. We represent the correct outputs as $c = [c_1, c_2, \ldots, c_l]$ and the computed outputs of a learning machine as $\hat{c} = [\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_l]$. Then we define the corresponding output errors as $e = [e_1, e_2, \ldots, e_l]$, where $e_i$ expresses the error on the $i$th 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 $e = [e_1, e_2, \ldots, e_l]$. In order to compute the dependence among the output errors, we have to divide the range of each $e_i$ in $b$ intervals $bin(j)$, $1 \leq j \leq b$:

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

with $0 = k_0 < k_1 < k_2 < \cdots < k_b = \text{max}$. The $j$th interval is selected by

$$bin(j) = [k_{j-1}, k_j) \quad j = 1 \ldots b, \quad k_{j-1}, k_j \in [0, \text{max}]$$
For instance, in the simplest case we have only two intervals: \( \text{bin} = \{[k_0, k_1), [k_1, k_2]\} \). The intervals \( \text{bin}(j) \) 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 \( k \)th output relative to the \( i \)th input example, and \( e_{kj} \) as the number of examples whose values \( e_k^{(i)} \) fall in the interval \( \text{bin}(j) \):

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

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

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

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

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

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

\[
I_E(e_1, \ldots, e_l) = \sum_{j_1=1}^{b} \ldots \sum_{j_l=1}^{b} p(e_{1j_1}, \ldots, e_{lj_l}) \log \left( \frac{p(e_{1j_1}, \ldots, e_{lj_l})}{p(e_{1j_1}) \ldots p(e_{lj_l})} \right).
\]

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 \( \text{bin}(1) = [0, k_1) \) defines the range of tolerance for the correct output, where \( k_1 = \delta \). Therefore an output affected by an error lower than \( \delta \) 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 \( I_{SE} \):

\[
I_{SE}(e_1, \ldots, e_l) = \sum_{j} p(e_{1j_1}, \ldots, e_{lj_l}) \log \left( \frac{p(e_{1j_1}, \ldots, e_{lj_l})}{p(e_{1j_1}) \ldots p(e_{lj_l})} \right),
\]

where

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

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 \( I_{E}(e_i, e_j) = [R_{ij}] \) and the *pairwise mutual information specific error matrix* \( S \), composed by the elements \( I_{SE}(e_i, e_j) = [S_{ij}] \). We define also two other global indices: the *pairwise mutual information error matrix index* \( \Phi_R \):

\[
\Phi_R = \sum_{i=1}^{l} \sum_{j=1}^{l} I_{E}(e_i, e_j)
\]  

and the *pairwise mutual information specific error matrix index* \( \Phi_S \):

\[
\Phi_S = \sum_{i=1}^{l} \sum_{j=1}^{l} I_{SE}(e_i, e_j).
\]

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.
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 $\Phi_R$ and $\Phi_R$ (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 NEUROb-jects, 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 \textit{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 $\Phi_R$ and $\Phi_S$ (Eqs. (7) and (8)).

We used NEUROObjects [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 $\delta$ (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 $\text{bins} \in \{2, 3, 4, 5, 6\}$, for a total of 20 pairs of $(\delta, \text{bins})$.

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 $I_E$, $I_{SE}$, $\Phi_R$ and $\Phi_S$ 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 $(\text{bins})^l$ 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 $\Phi_R$ and $\Phi_S$. 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 $\delta$.

4.2. Results and discussion

Now we present the results of the comparison of $I_E$ and $I_{SE}$ among all outputs, of the $\Phi_R$ and $\Phi_S$ 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
are represented the computed $I_E$ values. Each point corresponds to a different triplet of hidden units, number of intervals and values of $\delta$. 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 $\delta$ 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 $I_{rel}^E$ or $I_{rel}^{SE}$ 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:

$$I_{rel}^E = \frac{I_E(\text{monolithic}) - I_E(\text{PND})}{I_E(\text{monolithic})},$$

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

Each vertical line corresponds to a different triplet number of hidden units, number of intervals and value of $\delta$. 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 $\delta$ 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 $\Phi_R$ and $\Phi_S$.

### 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 $\Phi_R$ and $\Phi_S$ (Eqs. (7) and (8)). These matrices can be computed element by element, considering a different pair of outputs each time; in this way the $30 \times 30$ and $14 \times 14$ matrices for the data sets *letter* and *optdigits* can also be considered and the corresponding indices $\Phi_R$ and $\Phi_S$ 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 $\Phi_R$ between *monolithic* and *PND* ECOC learning machines considering 4 different data sets. On the axes are represented the $\Phi_R$ values of ECOC *monolithic* and ECOC *PND* learning machines. Each point corresponds to a triplet of hidden units, number of intervals and values of $\delta$. On all the data sets about all the points are above the dotted line, i.e. all the values of $\Phi_R$ are greater for ECOC *monolithic* compared with ECOC *PND*. Concerning the compared $\Phi_S$ 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 $\Phi_{rel}^R$ and $\Phi_{rel}^S$ of the pairwise mutual information error index $\Phi_R$ and of the pairwise mutual information specific error index $\Phi_S$:

$$\Phi_{rel}^R = \frac{\Phi_R(\text{monolithic}) - \Phi_R(\text{PND})}{\Phi_R(\text{monolithic})},$$

$$\Phi_{rel}^S = \frac{\Phi_S(\text{monolithic}) - \Phi_S(\text{PND})}{\Phi_S(\text{monolithic})}. \tag{10}$$

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.
Coming back to Figs. 4 and 5, clusters of points can be observed, especially in the \textit{optdigits} (Figs. 4(c) and 5(c)) and \textit{letter} (Figs. 4(d) and 5(d)) plots, and in a less noticeable way also in \textit{glass} (Figs. 4(b) and 5(b)). Focusing on \textit{optdigits}, Figs. 6(a) and (b) show that the clusters depend mainly on \( \delta \) values, with increasing values of \( \Phi_R \) and \( \Phi_S \) when \( \delta \) decreases. Moreover, the spread of the points inside each cluster depends on the number of bins, showing an increment of the \( \Phi_R \) and \( \Phi_S \) 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 EOC \textit{PND} inside each cluster is lower compared to EOC \textit{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 $\Phi_S$ 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 $\delta$ affect the pairwise mutual specific error index $\Phi_S$ considering its difference between ECOC monolithic and ECOC $PND$ learning machines.

In the optdigits data set, the width $\delta$ affects the relative differences of $\Phi_R$ and $\Phi_S$ 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 $\delta=0.1$, neither $\delta$ nor the number of intervals affect the relative differences in a significant way (data not shown).

However, ECOC monolithic learning machines show greater $\Phi_R$ and $\Phi_S$ values, no matter what the number of hidden units, intervals and values of $\delta$ are selected.
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 $\Phi_R$ and $\Phi_S$ (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).
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 $\Phi_B$ and $\Phi_S$ 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


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.