D.I.S.I.
Dipartimento di Informatica e
Scienze dell' Informazione
Università di Genova

Francesco Masulli
Giorgio Valentini

Mutual information methods for evaluating dependence among outputs in learning machines

D.I.S.I. Technical Report TR-01-02

Mutual information methods for evaluating dependence among outputs in learning machines

Francesco Masulli, Giorgio Valentini
INFM - Istituto Nazionale per la Fisica della Materia,
DISI - Dipartimento di Informatica e Scienze dell'Informazione,
Università di Genova, via Dodecaneso 35, 16146 Genova
E-mail: {masulli, valenti}@disi.unige.it

January 2001

Abstract

The evaluation of dependence among output errors of multi-input multi-output learning machines can help us in designing well-behaved systems, highlighting hidden interactions among their internal components that can add noise to the learning process. By estimating the relations between performances and dependence among output errors, we can compare different models of learning machines in order to select the ones best suited to a particular problem. We distinguish between dependence among outputs and dependence among output errors and we propose measures based on mutual information for evaluating both these types of dependence. Global measures of dependence between outputs and output errors, together with mutual information error matrices for evaluating specific dependences between each pair of outputs are presented. We propose a statistical test of hyphotesis for

evaluating the difference of the dependence among outputs and output errors between different learning machines, and we present also some numerical experiments to exemplify a practical application of the proposed measures.

Keywords: Measures of dependence among outputs in learning machines, dependence between errors and accuracy of learning machines, mutual information.

1 Introduction

The evaluation of the statistical dependence among the outputs of learning machines can provide us with information about their nature and behaviour and can suggest the selection of a well-suited model for solving a particular learning machine problem.

The analysis of the dependence among the outputs and among the output errors can highlight how the design of learning machines affects their performances: in fact their internal components can interact adding noise to the learning process. For example, the units of the hidden layers of a Multi Input Multi Output (MIMO) Multi Layer Perceptron (MLP) are shared by different outputs; each output in general computes a different function, as in the One Per Class decomposition scheme for classification [27], where each output tries to discriminate one class against all others. As a consequence the hidden units are not specialized for the task of a specific output unit, and they must take into account substantially different learning tasks. Conversely, ensemble of learning machines [8, 17] can achieve their best performances if their base components are accurate and diverse [15]. Kuncheva and Whitaker [20] have shown that the dependency between classifiers in majority vote ensembles [21, 18] is related to their classification accuracy, and Masulli and Valentini [25] have shown that effectiveness of ECOC ensemble methods [10] depends on the accuracy and independence among the base dichotomic classifiers.

The analysis of the dependence among output errors of different learning machines can suggest new architectures in order to lower the interdependence

of the errors, yielding to an improvement of the generalization capabilities of automatic learning systems.

The dependence among outputs is connected with the characteristics of the learning machine and the data set being used, and we can experimentally study the relations between performances and dependence among outputs or output errors of a learning machine only if we dispose of suitable measures of dependence.

We can measure the dependence among the outputs of a learning machine using different statistical tools such as *Cramer's V* or the *contingency coefficient C* [11] that are both χ^2 based, the covariance and the correlation coefficient statistics, the *Q-statistic* [20], or also non parametric correlation coefficients as the *Spearman rank-order correlation coefficient* or the *Kendall's tau* [22].

In this paper we propose measures based on mutual information for evaluating the dependence among the outputs and the output errors of learning machines.

Some of the main applications of mutual information to machine learning problems concern modeling of self organized systems and feature maps [23, 3], feature transformation and selection [13, 1, 5, 34, 30], image processing [2, 31], independent component analysis [6]. We extend the application of mutual information to the evaluation of the dependence among outputs and among output errors in learning machines.

The main idea behind the evaluation of dependence among outputs of learning machines through mutual information based measures consists in interpreting the dependence among the outputs as the common information shared among 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.

These measures assess the dependence among the outputs 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. Mutual information takes into account the marginal and joint probability distributions of the outputs, measuring in a sense the information shared among them. Moreover, if we define a precise notion of error on the outputs, we can also use the mutual information of these errors to evaluate their dependence. In particular, mutual information based measures 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.

This paper is organized as follows: In the next section the basic concepts about mutual information are revised. Then mutual information based measures for evaluating the dependence among the outputs and among the output errors of a learning machine are presented, and a statistical test of hypothesis for comparing mutual information based measures of different learning machines are proposed. In Sect. 3 some numerical experiments show how to use the proposed mutual information based measures to evaluate the dependence among output errors in different learning machines. The conclusions summarize the main results and the incoming developments of this work.

2 Mutual information and dependence among the outputs of a learning machine

We can distinguish two main problems related to the correlation among the outputs of a learning machine:

1. Estimating the dependence among the outputs of a learning machine.
2. Estimating the dependence among the output errors of a learning machine.

Both can be addressed using information theory methods based on the mutual information. In this section we recall the basic concepts about mutual

information and we present measures based on mutual information for estimating the dependence among the outputs and among the output errors of a learning machine.

2.1 Mutual information

Let us consider two *discrete random variables* X and Y having values in two alphabets $\mathcal{X} = \{x_k | k = 0, \pm 1, \ldots, \pm K\}$ and $\mathcal{Y} = \{y_j | j = 0, \pm 1, \ldots, \pm J\}$ where x_k and y_j are discrete numbers and $(2K + 1)$ and $(2J + 1)$ are respectively the total number of discrete levels, and let be $p(x_k) = P(X = x_k)$, $p(y_j) = P(Y = y_j)$ the probabilities that the random variables X and Y assume the values x_k and y_j. Then the *mutual information* $I(X, Y)$ between the random variables X and Y is:

$$I(X,Y) = \sum_{k=-K}^{+K} \sum_{j=-J}^{+J} p(x_k, y_j) \log \left(\frac{p(x_k, y_j)}{p(x_k) p(y_j)} \right) \tag{1}$$

where $p(x_k, y_j)$ is the joint probability mass function of discrete random variables X and Y.

We can consider $I(X, Y)$ as the information relative to the random variable X conveyed by observing the random variable Y.

The mutual information can be characterized by the following properties [7, 14]:

- *Symmetry* of the mutual information between X and Y:

$$I(X,Y) = I(Y,X)$$

- *Nonnegativity*: $I(X, Y) \geq 0$

- The mutual information can be expressed in terms of the entropy of the random variables:

$$I(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

where

$$H(X) = \sum_{k=-K}^{+K} p(x_k) \log\Big(\frac{1}{p(x_k)}\Big)$$

is the *entropy*, representing the average amount of information conveyed by the random variable X and

$$H(X|Y) = H(X,Y) - H(Y)$$

is the *conditional entropy* of X given Y, representing the amount of the uncertainty remaining about the random variable X after observing the random variable Y, where

$$H(X,Y) = \sum_{k=-K}^{+K} \sum_{j=-J}^{+J} p(x_k, y_j) \log\Big(\frac{1}{p(x_k)p(y_j)}\Big)$$

is the *joint entropy* of X and Y, representing the joint average amount of information conveyed by the random variables X and Y.

2.2 Estimating the dependence among the outputs of a learning machine

In a typical machine learning problem a learning algorithm outputs an hypothesis $\hat{\mathbf{f}}(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^l$ of the unknown function $\mathbf{f}(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^l$ using a limited data set $\mathcal{D} = \overset{N}{_{i=1}}\{(\mathbf{x}^{(i)}, \mathbf{c}^{(i)})\}$, where $\mathbf{x}^{(i)} \in \mathbb{R}^d$ and $\mathbf{c}^{(i)} \in \mathbb{R}^l$.

In order to compute the mutual information among the outputs of a learning machine, we have to discretize its outputs. Considering a vector of outputs $\mathbf{c} = [c_1, c_2, \ldots, c_l]$, we can discretize each c_i in b intervals $bin(j), 1 \leq j \leq b$, defining it in the following way:

$$bin(j) = \Big[min + (j-1)\frac{(max - min)}{b}, min + j\frac{(max - min)}{b}\Big]$$

where min and max are minimum and maximum values of the outputs c_i. Let be $c_k^{(i)}$ the k^{th} output of the i^{th} sample; we define c_{kj} as the number of $c_k^{(i)}$ values falling in the interval $bin(j)$:

$$c_{kj} = \Big|\{i \in [1, N] | c_k^{(i)} \in bin(j)\}\Big|$$

where N is the cardinality of the data set. Consequently, the *discrete probability mass function* of c_{kj} is:

$$p(c_{kj}) = \frac{\left|\{i \in [1,N] | c_k^{(i)} \in bin(j)\}\right|}{N}$$

and the *discrete joint probability mass function* of c_{kj} and $c_{k'j'}$ is:

$$p(c_{kj}, c_{k'j'}) = \frac{\left|\{i \in [1,N] | (c_k^{(i)} \in bin(j)) \wedge (c_{k'}^{(i)} \in bin(j'))\}\right|}{N}$$

The *discrete joint probability mass function among all the outputs* can be defined in the following way:

$$p(c_{1j_1}, c_{2j_2}, \ldots, c_{lj_l}) = \frac{\left|\{i \in [1,N] | \bigwedge_{1 \leq u \leq l} (c_u^{(i)} \in bin(j_u))\}\right|}{N}$$

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

The *mutual information* $I(c_1, c_2, \ldots, c_l)$ among all the outputs $\mathbf{c} = [c_1, c_2, \ldots, c_l]$ of a learning machine is defined as:

$$I(c_1, \ldots, c_l) = \sum_{j_1=1}^{b} \ldots \sum_{j_l=1}^{b} p(c_{1j_1}, \ldots, c_{lj_l}) \log \left(\frac{p(c_{1j_1}, \ldots, c_{lj_l})}{p(c_{1j_1}) \ldots, p(c_{lj_l})} \right) \quad (2)$$

It is worth noting that, in the computation of the mutual information, a form of the *curse of dimensionality* problem can arise, as the computation of $I(c_1, \ldots, c_l)$ requires the sum of b^l elements and the memorization of matrices l dimensional composed by b^l elements. For instance, with 10 outputs and 8 intervals we would have joint probability matrices with 8^{10} elements, and also disregarding the space and time computational complexity involved, we need anyway billions of data samples to fill so huge matrices.

To overcome this problem we can evaluate the mutual information between all the output pairs, that is, $I(c_i, c_j),\ \forall i, j$, collecting the values in *pairwise mutual information matrices* $\mathbf{M}$, composed by elements $[\mathbf{M}_{ij}] = I(c_i, c_j)$. We define also a *pairwise mutual information matrix index* Φ_M:

$$\Phi_M = \sum_{i=1}^{l} \sum_{j=1}^{l} I(c_i, c_j) \quad (3)$$

It extracts a global value from the pairwise mutual information matrix and can be used as a "surrogate" of the mutual information among all the outputs (eq. 2); it is worth noting that in general eq. 2 is not equivalent to 3, because it considers only the mutual information between pairs of outputs and not the overall mutual information among all the outputs.

Mutual information is always positive and it is zero if and only if the outputs are statistically independent; the mutual information expresses not only the dependence among the outputs but also how are similar the probability distributions of the different outputs. In this sense the mutual information is stronger than the simple correlation coefficients among the outputs.

Pairwise mutual information matrices provide information about the dependence among all the outputs, but can provide insights into the similarity of the probability distributions between specific pairs of outputs. The pairwise mutual information matrix index (eq. 3) does not express precisely the dependence among all the outputs, but can be used in conjunction with the mutual information among all outputs or when the direct computation of the overall mutual information is not computationally feasible.

It is worth noting that a basic correlation among the outputs of learning machines always exists, because their significant task consists in learning specific output patterns, that in general, are internally correlated and not random.

2.3 Estimating the dependence among the output errors of a learning machine

In this section we deal with the problem of the dependence among the output errors of a learning machine.

As a consequence, we have to evaluate the mutual information not among the outputs but among the errors of the outputs. We call this quantity *mutual information error*, I_E for short. In particular, defining an explicit notion of correctness of the outputs, we can introduce a "mutual information" generated by two or more errors on the outputs, that is, without considering

the mutual information error generated by correct outputs and by errors only on a single output. We define this quantity *mutual information specific error* I_{SE}.

More precisely, let us 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 $e_i = |c_i - \hat{c_i}|, \forall i = 1 \ldots l$ expresses the error on the i^{th} output of the learning machine.

The outputs of a learning machine can be considered correct if $\forall i,\ e_i \sim 0$, or, better, if $e_i < \delta,\ \delta \geq 0$. For instance, in a classification problem a threshold usually separates the assignment of a class from another and so it is natural to associate δ with this threshold. In the same way in a regression problem using the ϵ insensitive loss function [33] usually used with support vector machines is natural to associate δ with ϵ itself.

In order to compute the mutual information error and the mutual information specific error we have to discretize the outputs of the learning machine. Representing the output errors as a vector $\mathbf{e} = [e_1, e_2, \ldots, e_l]$, we can discretize each e_i in b intervals, defining the set of the intervals $bin(j), 1 \leq j \leq b$ as an ordered list:

$$bin = \{[k_0, k_1), [k_1, k2), \ldots, [k_{b-1}, k_b]\}$$

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

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

The $bin(1)$ is the correct interval and the others are intervals corresponding to errors. The first interval $bin(1) = [0, k_1)$ is such that $k_1 = \delta$, that is an error lower than δ is interpreted as a correct output. For instance, in the simplest case we have two intervals: $bin = \{[k_0, k_1),\ [k_1, k2]\}$ and $bin(1) = [k_0, k_1), k_1 = \delta$ is the correct interval.

Having a data set $\mathcal{D} =_{i=1}^{N} \{(\mathbf{x}^{(i)}, \mathbf{c}^{(i)})\}$, where $\mathbf{x}^{(i)} \in \mathbb{R}^d$ and $\mathbf{c}^{(i)} \in \mathbb{R}^l$, in a way similar to the previous section 2.2 we define e_{kj} as the number of the

$e_k^{(i)}$ values falling in the interval $bin(j)$, that is, e_{kj} counts the values that fall in j^{th} interval of the k^{th} output error:

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

where N is the cardinality of the data set. Analogously to the previous section 2.2 we define the *discrete probability function* $p(e_{kj})$ of e_{kj}:

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

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

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

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

Let us assess the dependence among output errors in the simplest case, that is when we have only two outputs. In this case, the mutual information among output errors, (*mutual information error* I_E) is:

$$I_E(e_1, e_2) = \sum_{j_1=1}^{b} \sum_{j_2=1}^{b} p(e_{1j_1}, e_{2j_2}) \log \left(\frac{p(e_{1j_1}, e_{2j_2})}{p(e_{1j_1}) p(e_{2j_2})} \right) \tag{4}$$

and the *mutual information specific error* I_{SE} is:

$$I_{SE}(e_1, e_2) = \sum_{j_1=2}^{b} \sum_{j_2=2}^{b} p(e_{1j_1}, e_{2j_2}) \log \left(\frac{p(e_{1j_1}, e_{2j_2})}{p(e_{1j_1}) p(e_{2j_2})} \right) \tag{5}$$

If we have only two intervals (i.e. $b = 2$) than I_{SE} is reduced to:

$$I_{SE}(e_1, e_2) = p(e_{12}, e_{22}) \log \left(\frac{p(e_{12}, e_{22})}{p(e_{12}) p(e_{22})} \right)$$

In equation 5 we do not consider the intervals corresponding to both correct outputs or when an output is correct but the other one is not. I_{SE} evaluates only the mutual information error due to the coupling of the errors on the outputs of the learning machine. To this purpose recall that the first

interval in the ordered list (corresponding to the index $j_1 = 1$ or $j_2 = 1$ in eq. 5) represents a correct output (see above in this section).

Considering the mutual information error among an arbitrary number, say l, of outputs, then the mutual information among output errors, (*mutual information error* I_E) becomes:

$$I_E(e_1, \ldots, e_l) = \sum_{j_1=1}^{b} \ldots \sum_{j_l=1}^{b} p(e_{1j_1}, \ldots, e_{lj_l}) \log \left(\frac{p(e_{1j_1}, \ldots, e_{lj_l})}{p(e_{1j_1}) \ldots p(e_{lj_l})} \right) \qquad (6)$$

The mutual information error (eq. 6) 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 are similar the probability distribution of the output errors.

In most learning machine problem we have a "relaxed" notion of error. For instance, in a classification problem with neural networks a class is selected if the output is above or below a predeterminated threshold value, or if it is "near" a respecified value [16, 4]. Also in regression problems the ϵ insensitive loss function [33] introduces a more relaxed notion of error. Defining an explicit notion of correctness of the outputs (i.e. introducing δ, see above in this section), we can introduce a "mutual information" generated by two or more errors on the outputs, that is, without considering the mutual information error generated by correct outputs and by errors only on a single output. We define this quantity *mutual information specific error* I_{SE}:

$$I_{SE}(e_1, \ldots, e_l) = \sum_{\mathcal{J}} p(e_{1j_1}, \ldots, e_{lj_l}) \log \left(\frac{p(e_{1j_1}, \ldots, e_{lj_l})}{p(e_{1j_1}) \ldots p(e_{lj_l})} \right) \qquad (7)$$

where

$$\mathcal{J} = \left\{ [j_1, j_2, \ldots j_l] | \exists (j_v, j_w) | (j_v \neq 1) \wedge (j_w \neq 1) \wedge (v \neq w),\ v, w \in \{1 \ldots l\} \right\}$$

The mutual information specific error (eq. 7) takes into account the output errors when two or more errors spring from the output, disregarding all cases with no errors or with only one error. Then, if we have l outputs, are considered all cases with $l - 2$ correct outputs, $l - 3, l - 4$, until 0 correct

outputs and l errors. In a proper sense it is not a mutual information among random variables according to the information theory, but it expresses the dependence among two or more errors on the outputs of a a learning machine, disregarding the mutual information error due to a single error or no errors on the outputs.

Due to the same reasons we have outlined in Sect. 2.2, the *curse of dimensionality* problem can arise also with the mutual information error and the mutual information specific error.

These problems can be tackled evaluating the mutual information error between all the output pairs. We define a *pairwise mutual information error matrix* R composed by the elements $I_E(e_i, e_j) = [R_{ij}]$. It can be defined also a *pairwise mutual information error matrix index* Φ_R:

$$\Phi_R = \sum_{i=1}^{l} \sum_{j=1}^{l} I_E(e_i, e_j) \tag{8}$$

In the same way can be defined a *pairwise mutual information specific error matrix* S, composed by the elements $I_{SE}(e_i, e_j) = [S_{ij}]$ and a *pairwise mutual information specific error matrix index* Φ_S:

$$\Phi_S = \sum_{i=1}^{l} \sum_{j=1}^{l} I_{SE}(e_i, e_j) \tag{9}$$

These indices can be used as a substitute of the mutual information error and the mutual information specific errors among all output errors, because these values express the total pairwise dependence between all the couples of output errors. However these indices (eq. 8 and 9) are not equivalent to the corresponding equations 6 and 7 of the mutual information among all output errors. Recall that eq. 8 and 9 consider only the mutual information between pairs of output errors, while eq. 6 and 7 consider the overall mutual information among all output errors.

It is worth noting that the absolute values of I_E, I_SE, Φ_R and Φ_S depend on the number of outputs and on the selected number of discretization intervals. Consequently we can consider some kind of normalization, such as dividing by the number of outputs and intervals.

These mutual information related quantities can be used to compare the correlation of the output errors among different learning machines on the same learning problem, using, of course, the same data sets. The mutual information error and the mutual information specific error can offer insights into the dependence and the probability distribution of the errors, especially when we want to compare the behaviour of different architectures of learning machines. Moreover we can also apply these measures to evaluate the diversity of classifiers ensemble. I_E and I_{SE} express the global diversity of the ensemble, while R and S matrices can give insight into the dependence between specific pairs of base classifiers.

2.4 Mutual information error t-test

In this section we propose a *mutual information error t-test* for evaluating if a statistically significant difference between the mutual information error I_E or the mutual information specific error I_{SE} of two different learning machines does exist. More specifically, consider two different learning machines, say A and B, trained to solve a specific classification or regression problem using the same data set T. Let I_E^A and I_E^B the mutual information error associated respectively with the learning machine A and B on the same data set T [1].

Our aim consists in testing the following null hyphotesis H_o:

Null hypothesis: *I_E^A and I_E^B evaluated on the same data set are equal.*

We consider I_E^A and I_E^B as arithmetic means of multiple evaluations of I_E, i.e.:

$$I_E^A = \frac{1}{N}\sum_{i=1}^{N} I_{E_i}^A, \qquad I_E^B = \frac{1}{N}\sum_{i=1}^{N} I_{E_i}^B \tag{10}$$

where N is the number of evaluations. These can be attained running several times the learning algorithm with different initial conditions, or using resampling techniques, such as k-fold cross validation [19].

We assume that both $I_{E_i}^A$ and $I_{E_i}^B$, $i \in \{1, \ldots, N\}$ are randomly drawn from a normal distribution with means μ_A and μ_B respectively and unknown

[1] Here we consider I_E, but the same argumentations can be applied to I_{SE}.

variance. Hence I_E^A and I_E^B (eq. 10) are normal distributed with means μ_A and μ_B. Now, our null hyphotesis H_o becomes:

$$H_o : \mu_A - \mu_B = 0 \tag{11}$$

Let us consider the paired differences between $I_{E_i}^A$ and $I_{E_i}^B$:

$$D_i = I_{E_i}^A - I_{E_i}^B, \quad i = 1, \ldots, N \tag{12}$$

Considering that $I_{E_i}^A$ and $I_{E_i}^B$ are normal distributed with means μ_A and μ_B, the paired differences are normal distributed with mean $\mu_A - \mu_B$, and also the corresponding mean:

$$\overline{D} = \frac{1}{N} \sum_{i=1}^{N} D_i, \quad i = 1, \ldots, N \tag{13}$$

is normal distributed with mean $\mu_A - \mu_B = \mu$.

Let be σ^2 the variance of the paired differences D_i; the *sample variance* s^2 of the normal distributed D_i is:

$$s^2 = \frac{\sum_{i=1}^{N} (D_i - \overline{D})^2}{N - 1} \tag{14}$$

Let us define two statistics S_1 and S_2:

$$S_1 = \frac{\overline{D} - \mu}{\frac{\sigma}{\sqrt{N}}} \tag{15}$$

$$S_2 = \frac{s^2}{\frac{\sigma^2}{N-1}} \tag{16}$$

S_1 has a standard normal distribution and S_2 has a χ^2 distribution with $N - 1$ degrees of freedom. A well-known result from statistics (see, for example, [12]) is that the following variable t has a t distribution with $N - 1$ degrees of freedom:

$$t = \frac{S_1}{\sqrt{S_2/(N-1)}} \tag{17}$$

Using eq. 17, 15, 16 and 14 we have:

$$\begin{aligned} t &= \frac{S_1}{\sqrt{S_2/(N-1)}} \\ &= \frac{\sqrt{N}(\overline{D}-\mu)}{\sigma}\sqrt{\frac{\sigma^2}{s^2}} \\ &= \frac{\sqrt{N}(\overline{D}-\mu)}{\sqrt{s^2}} \\ &= \frac{\sqrt{N(N-1)}(\overline{D}-\mu)}{\sqrt{\sum_{i=1}^{N}(D_i-\overline{D})^2}} \end{aligned} \tag{18}$$

The t statistic (eq.18) has a t-Student distribution with $N-1$ degrees of freedom. According to our null hypothesis (11), $\mu_A - \mu_B = 0$ and hence $\mu = \mu_A - \mu_B = 0$.

We can now define a statistical test of hypothesis using the above t statistic and the *t-Student distribution*:

Mutual information error t-test : An appropriate critical region of size α for testing the null hypothesis $Ho : \mu_A - \mu_B = 0$ is defined by $|t| > t_{\alpha/2}$, where $t_{\alpha/2}$ is defined in such a way that

$$\int_{t_{\alpha/2}}^{\infty} f(t)dt = \alpha/2$$

where $f(t)$ is a t-Student distribution.

Consequently the null hyphotesis can be rejected with a degree of confidence of $1-\alpha$ if the computed t statistic falls into the tails of the t-Student distribution, i.e if $|t| > t_{\alpha/2}$.

In practice we have to compute the t-statistic of eq. 18; then, after choosing a desired size α of the critical region or equivalently a $1-\alpha$ degree of confidence, we can find the requested value $t_{\alpha/2}$ with $N-1$ degrees of freedom using tabulated values for the t-Student distribution or computing it through the *gamma* distribution [29].

3 Numerical experiments

In this section we exemplify an application of the mutual information based methods described in the previous sections to the evaluation of the dependence among output errors of *Error correcting Output Coding monolithic* [9, 10, 25, 26] (ECOC *monolithic* for short) and *ECOC Parallel Non linear Dichotomizers* [25, 26, 24] (ECOC *PND* for short) learning machines, using a synthetic data set.

3.1 Experimental setup

ECOC is a two-stage classification method, that consists in decomposing a multiclass problem in a number of two-class (dichotomic) subproblems and then combining them to achieve the class label. Both *monolithic* and *PND* ECOC learning machines code their outputs through error correcting output codes [28], in order to exploit their error correcting capabilities. They differ in their design: ECOC *monolithic* are implemented by a single multilayer perceptron (MLP) with one hidden layer, while ECOC *PND* are implemented by an ensemble of dichotomic MLPs, one for each different dichotomy generated by the ECOC decomposition.

The synthetic data set *d5* is made up by five three-dimensional classes, each composed by two normal distributed disjoint clusters of data [2].

We have used, both for training the learning machines and for evaluating the dependence among the output errors, *NEURObjects* [32], a set of C++ library classes for neural networks development.

In order to perform training and testing of the two ECOC learning machines, we have applied multiple runs of different random initialization of the weights using a training and test set both composed by 30000 samples. MLP ECOC achieved an error on the test set equal to $18.31\% \pm 6.44$, while *PND* ECOC ensemble an error equal to $12.34\% \pm 0.74$.

After the training we have collected the outputs of the decomposition

[2] The synthetic data set *d5* is available at ftp://ftp.disi.unige.it/person/ValentiniG/Data.

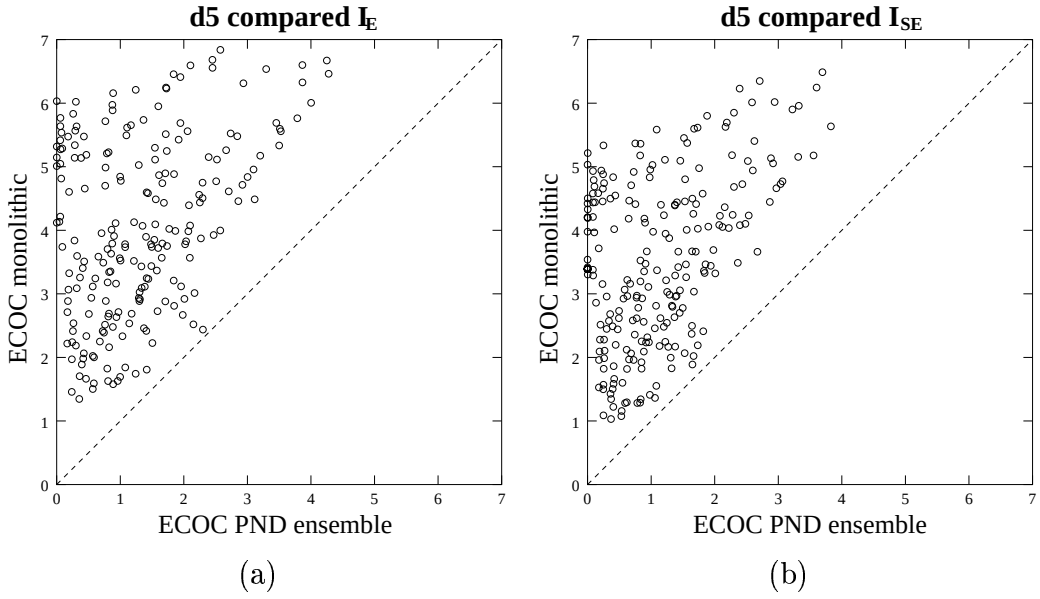


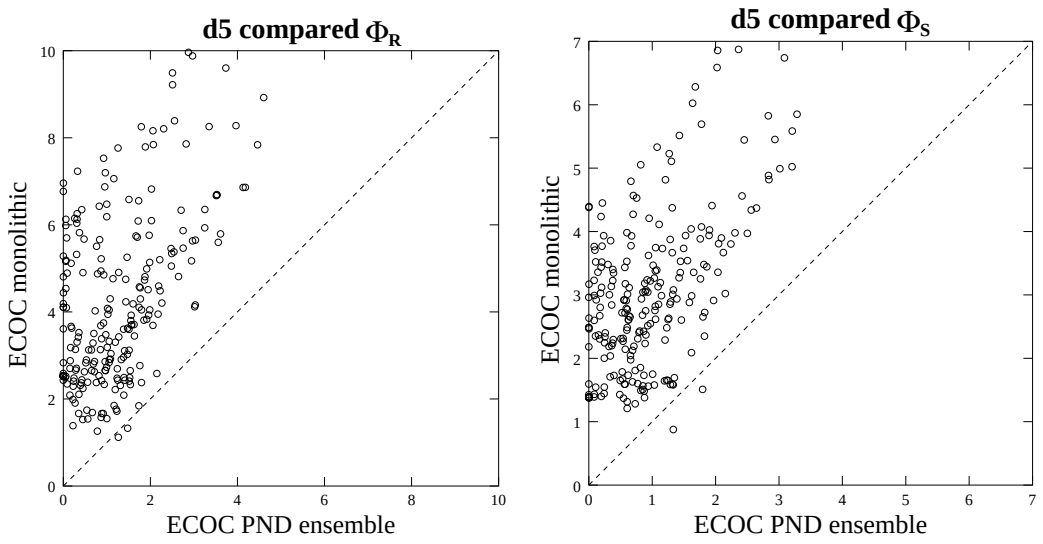
Figure 1: Compared mutual information error I_E (a) and mutual information specific error I_{SE} (b) among all outputs between ECOC *monolithic* and *PND* learning machines on the d5 data set.

units of the learning machines on the test sets. Then we have computed the errors (see Sect. 2.3), obtaining matrices of output errors: their lines are the vectors of output errors relative to a single sample, and their columns are the errors on a single output of the overall samples.

Using these error data we have computed and then compared the mutual information error I_E (eq.6) and the mutual information specific error I_{SE} (eq.7) among all the outputs of the considered learning machines (Fig. 1).

3.2 Results

In Fig. 1(a) we compare I_E among all output errors of the *monolithic* MLP and ECOC *PND* learning machines. On the axes are represented the computed I_E values. Each point corresponds to a different triplet number of hidden units, number of intervals and values of δ. If a point is on the dotted diagonal then the values of I_E are equal for both the learning machines; if a point is above or below the dotted line then the I_E of the ECOC *monolithic* is



(a) (b)

Figure 2: Compared mutual information error matrix indices Φ_R (a) and mutual information specific error matrix indices Φ_S (b) between ECOC *monolithic* and *PND* learning machines on the d5 data set.

respectively greater or smaller than the corresponding I_E value of the ECOC *PND* learning machine. All points are above the dotted line, showing that the mutual information error I_E is greater for ECOC *monolithic* respect to ECOC *PND*, no matter the structure, the number of intervals and the δ values used. Considering the mutual information specific error I_{SE} among all the outputs (Fig. 1(b)), similar results are obtained: In all the 220 comparisons on the data set *d5*, I_{SE} is greater for ECOC *monolithic* with respect to ECOC *PND* learning machines.

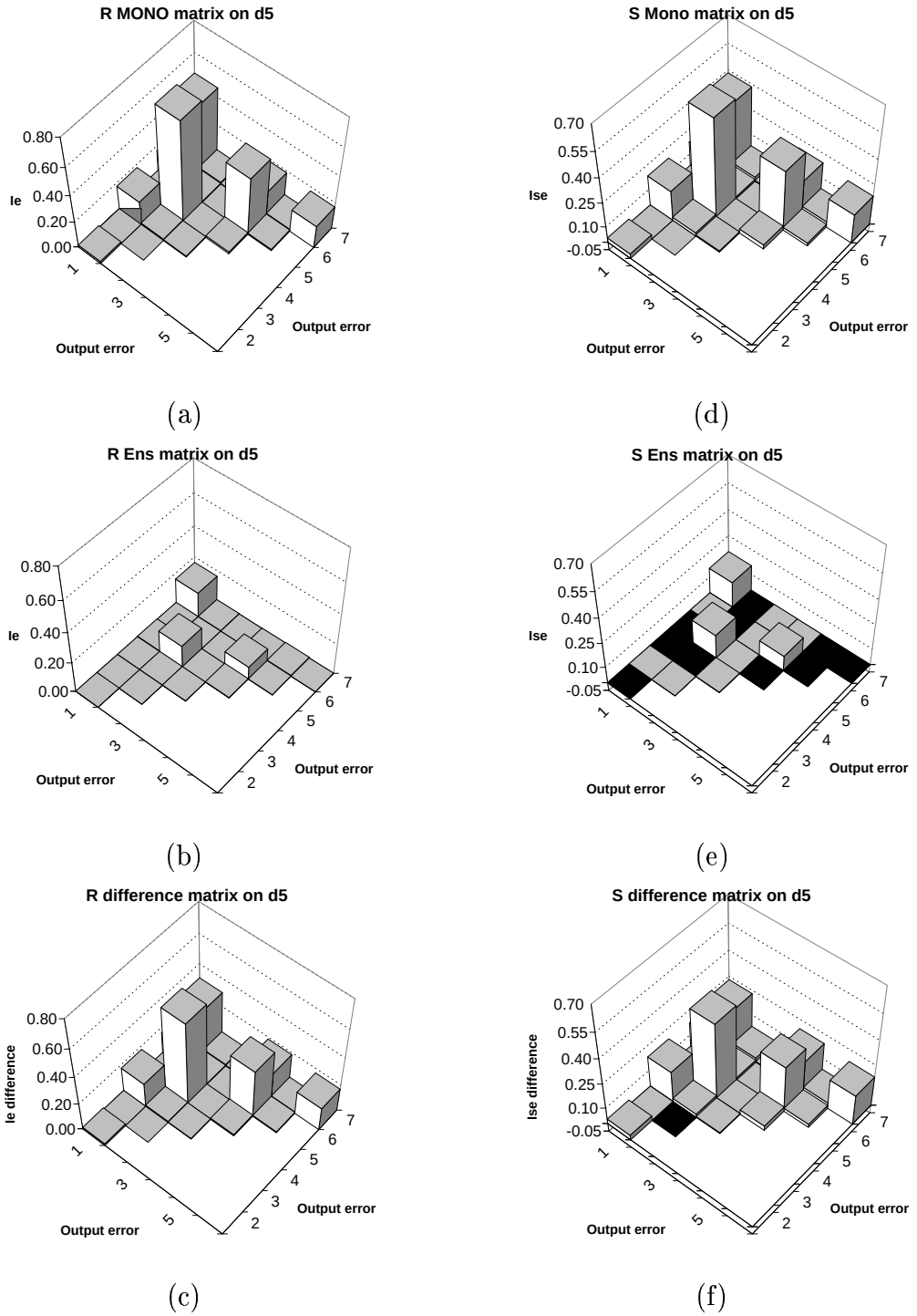
Then we have computed and compared the mutual information error matrices R and the mutual information specific error matrices S (see Sect. 2.3) and the their relative global indices Φ_R and Φ_S (eq. 8 and 9).

Examining the differences of the pairwise mutual information error index Φ_R (Fig. 2a), and of the pairwise mutual information specific error index Φ_S (Fig.2b), we can see that only in 2 up to 220 cases we have higher values for ECOC *PND* compared to ECOC MLP. Note that in these two cases the

values correspond to a triplet with 2 intervals of discretization and a value of $\delta = 0.1$; this kind of raw discretization joined to a strong asymmetry of the intervals with a low value of δ can introduce noise in the computation of Φ_R and Φ_S.

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 (Sect. 2.3) selecting a triplet with $\delta = 0.4$ and a number of intervals equal to 6 and with a fixed number of hidden units. Fig. 3 represents the mutual information matrices on the *d5* data set. On the left column the R matrices of ECOC *monolithic* (a), ECOC *PND* (b) and their difference (c) are shown. On the right column are represented the S matrices of 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, because they are symmetric and the elements of 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. 3c), and in the S difference matrix only on the pair of outputs 2 and 3 we have a negative value (Fig. 3f).

All the considered mutual information based measures agree that ECOC MLP shows an higher dependence among outputs compared with ECOC *PND*. Using the proposed *mutual information error t-test* (Sect. 2.4) for evaluating if a statistically significant difference between the mutual information error I_E of MLP and *PND* ECOC does exist, in almost all the 220 comparisons we have registered a significant difference with a degree of confidence of 95%.



(a) (d)

(b) (e)

(c) (f)

Figure 3: Pairwise mutual information matrices on 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), of the Ens (e) learning machines, and their difference (f).

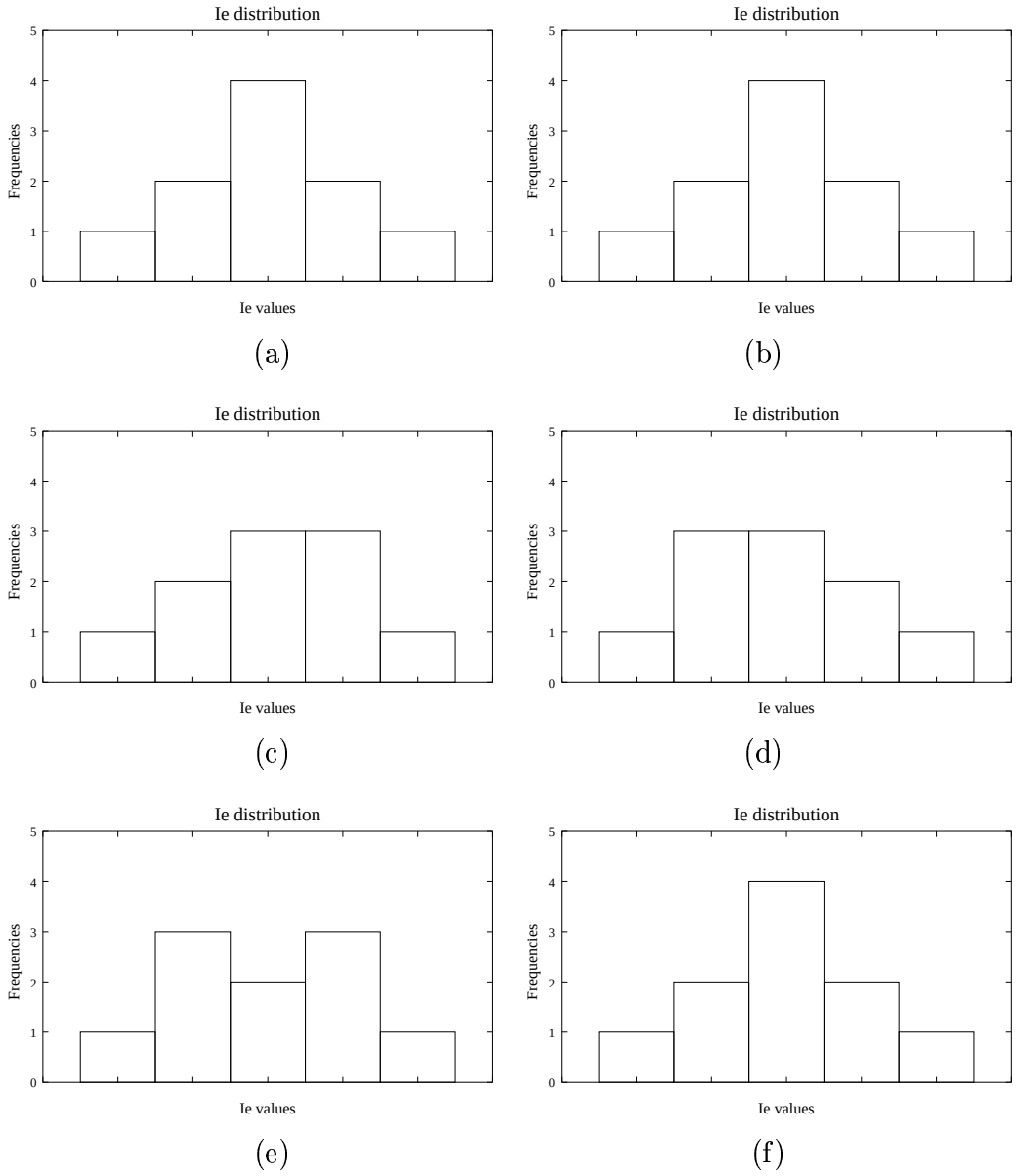


Figure 4: Distribution of I_E on the synthetic *d5* data set considering three triplets (number of hidden units, number of intervals, δ), one different for each line. Each histogram show the distribution of the I_E generated by 10 different pseudorandom initialization of two different learning machines. On the left column is evaluated the I_E distribution of the ECOC *monolithic* MLP, on the right the corresponding distribution of the ECOC *PND*.

Recall that dealing with this test in Sect. 2.4 we assumed the I_{E_i} were normally distributed. The empirical distributions, computed using the synthetic data set *d5* used in our experimentation, show a normal-like shape (Fig. 4), confirming that we can safely apply the proposed test.

4 Conclusions

We have presented measures based on mutual information for evaluating the dependence among the outputs and among the output errors of a learning machine.

These measures assess the dependence among the outputs considering also 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. Mutual information based quantities can provide us with useful insights into the internal behaviour of learning machines, supplying also information for a proper selection of well-designed systems: learning machines with low dependent output errors should be preferred in order to achieve better generalization capabilities.

We can assess the dependence among outputs evaluating directly the mutual information, as the outputs are statistically independent if and only if their mutual information is zero.

The mutual information error I_E evaluates the dependence among all output errors, considering as error any deviation from the correct output. Introducing the notion of correct output through a tolerance threshold δ, the mutual information specific error I_{SE} assesses the dependence among output errors considering an output correct if its deviation from the true value is below a threshold δ. Moreover I_{SE} provides us with a more specific test to evaluate the dependence among output errors, joining the notion of correct output to the notion of two ore more errors on the outputs.

The evaluation of the statistical significance of the difference of I_E and I_{SE} between different learning machines can be performed using classical test of

hypotheses: To this aim we have proposed a *mutual information error t-test* that is straightforward to implement.

If we have an high number of outputs and if we select an high number of discretization intervals, I_E and I_{SE} among all output errors can be too computationally expensive to be evaluated, as a *curse of dimensionality* problem can arise. Introducing the pairwise mutual information matrices R and S and their associated global indices Φ_R and Φ_S we can overcome this problem considering a less expensive evaluation of the mutual information error between pairs of outputs, even if the values of these indices are not equivalent to the global I_E and I_{SE}. In fact Φ_R and Φ_S estimates the sum of the dependencies among all output error pairs, and I_E and I_{SE} directly the dependence among all output errors.

R and S matrices supply also information about dependence between specific pairs of output errors, highlighting how much a specific output error is associated with another one. The overall matrices can be also used for comparing the dependence among outputs of different learning machines, as shown in the proposed numerical experiments on the *d5* data set.

Some straightforward enhancements concerning the proposed mutual information based measures may consist in explicitly considering the number of outputs and the selected number of discretization intervals. In fact, as pointed out in Sect. 2.3, the absolute values of I_E, I_{SE}, Φ_R and Φ_S depend on the number of outputs and on the selected number of discretization intervals. In order to overcome this problem we can consider some kind of normalization, such as dividing by the number of outputs and intervals. For evaluating the differences of these measures between different learning machines, we can also consider their relative differences, in order to obtain homogeneous values ranging from -1 to 1.

A planned development of this work will be the extension of the proposed measures based on mutual information to the evaluation of the diversity of base learners, a fundamental problem for a proper design of ensembles of learning machines.

Acknowledgments

This work has been partially funded by CNR - Progetto finalizzato MADESS II. We would like to thank Franco Fagnola for his suggestions and helpful discussions.

References

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

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

[3] A.J. Bell and T.J. Sejnowski. An information-maximazation approach to blind separation and blind deconvolution. *Neural Computation*, 6:1129–1159, 1995.

[4] C. M. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford, 1995.

[5] B. Bonnlander and A.S. Weigend. Selecting input variables using mutual information and non parametric density estimation. In *Proceedings of the 1994 International Symposium on Artificial Neural Networks*, pages 42–50, Taiwan, 1994.

[6] P. Comon. Independent component analysis: A new concept? *Signal Processing*, 36:287–314, 1994.

[7] T.M. Cover and J.A. Thomas. *Elements of Information Theory.* Wiley, New York, 1991.

[8] T.G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems. First International Workshop, MCS2000, Cagliari, Italy*, pages 1–15. Springer-Verlag, 2000.

[9] T.G. Dietterich and G. Bakiri. Error - correcting output codes: A general method for improving multiclass inductive learning programs. In *Proceedings of AAAI-91*, pages 572–577. AAAI Press / MIT Press, 1991.

[10] T.G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, (2):263–286, 1995.

[11] O.J. Dunn and V.A. Clark. *Applied Statistics: Analysis of Variance and Regression.* Wiley, New York, 1974.

[12] J.R. Freund. *Mathematical Statistics.* Prentice-Hall, Englewood Cliffs, NJ, 1962.

[13] K. Fukunaga. *Introduction to statistical pattern recognition.* Academic Press, New York, 1990.

[14] R.M. Gray. *Entropy and Information Theory.* Springer-Verlag, New York, 1990.

[15] L. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990.

[16] S. Haykin. *Neural Networks: a comprehensive foundation.* Mc Millan, New York, 1999.

[17] A. Jain, R. Duin, and J. Mao. Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:4–37, 2000.

[18] G. James. *Majority vote classifiers: theory and applications.* PhD thesis, Department of Statistics - Stanford University, Stanford, CA, 1998.

[19] A. Krogh and J. Vedelsby. Neural networks ensembles, cross validation and active learning. In Touretzky D. S., Tesauro G., Leen T.K., editor,

Advances in Neural Information Processing Systems, volume 7, pages 107–115. MIT Press, Cambridge, MA, 1995.

[20] L. Kuncheva and C. Whitaker. Measures of diversity in classifier ensembles. 2001. (to appear).

[21] L. Lam and C. Sue. Application of majority voting to pattern recognition: an analysis of its behavior and performance. *IEEE Transactions on Systems, Man and Cybernetics*, 27(5):553–568, 1997.

[22] E.L. Lehmann. *Nonparametrics: Statistical Methods based on Ranks*. Holden-Day, S.Francisco, 1975.

[23] R. Linsker. An aplication of the principle of maximum information preservation to linear systems. In *Advances in Neural Information Processing Systems*, volume 1, pages 186–194. Morgan Kauffman, San Mateo, CA, 1989.

[24] F. Masulli and G. Valentini. Comparing decomposition methods for classification. In R.J. Howlett and L.C. Jain, editors, *KES'2000, Fourth International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies*, pages 788–791, Piscataway, NJ, 2000. IEEE.

[25] F. Masulli and G. Valentini. Effectiveness of error correcting output codes in multiclass learning problems. In *Lecture Notes in Computer Science*, volume 1857, pages 107–116. Springer-Verlag, Berlin, Heidelberg, 2000.

[26] F. Masulli and G. Valentini. Parallel Non linear Dichotomizers. In *IJCNN2000, The IEEE-INNS-ENNS International Joint Conference on Neural Networks*, volume 2, pages 29–33, Como, Italy, 2000.

[27] N.J. Nilsson. *Learning Machines*. Mc Graw Hill, New York, 1965.

[28] W.W. Peterson and E.J.Jr. Weldon. *Error correcting codes.* MIT Press, Cambridge, MA, 1972.

[29] W.H. Press, S.A. Teukolski, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 1992.

[30] K. Torkkola and W. M. Campbell. Mutual information in learning feature transformations. In *Proc. ICML'2000, The Seventeenth International Conference on Machine Learning*, 2000.

[31] A.M. Ukrainec and S. Haykin. A modular neural network for enhancement of cross-polar radar targets. *Neural Networks*, 9:143–168, 1996.

[32] G. Valentini and F. Masulli. NEURObjects, a set of library classes for neural networks development. In *Proceedings of the third International ICSC Symposia on Intelligent Industrial Automation (IIA'99) and Soft Computing (SOCO'99)*, pages 184–190, Millet, Canada, 1999. ICSC Academic Press.

[33] V. N. Vapnik. *The nature of Statistical Learning Theory.* Springer, New York, 1995.

[34] H. Yang and J. Moody. Feature selection based on joint mutual information. In *Advances in Intelligent Data Analysis (AIDA), Computational Intelligence Methods and Applications (CIMA), ICSC*, Rochester, New York, 1999.