Quantitative Evaluation of Dependence among Outputs in ECOC Classifiers Using Mutual Information Based Measures

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

Abstract

In previous work, it has been experimentally shown that the implementation of Error Correcting Output Coding (ECOC) classification methods with an ensemble of parallel and independent non linear dichotomizers (ECOC PND) outperforms the implementation with a single monolithic multi layer perceptron (ECOC MLP). The low dependence of the errors on different codeword bits was qualitatively indicated as one of the main factors affecting this result. In this paper, we quantitatively evaluate the dependence of output errors in ECOC learning machines using mutual information based measures, and we study the relation between dependence of output errors and classification performances.

1 Introduction

The evaluation of the statistical dependence among the outputs of learning machines can provide us with information about their nature and behavior and can help us to select well-suited models for solving a specific learning problem.

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* [5] that are both χ^2 based, the covariance and the correlation coefficient statistics, the *Q-statistic* [6], or also non parametric correlation coefficients as the *Spearman rank-order correlation coefficient* or the *Kendall's tau* [7].

In this paper we use some mutual information based measures for the evaluation of dependence among outputs errors in a learning machine proposed in [10]. The main idea behind the application of those measures of dependence consists in interpreting the dependence among the outputs as the common information shared among them. Mutual information measures have been already applied to different problems in machine learning, e.g., to the modeling of self organizing systems [1], to image processing [15], and to feature transformation and selection [14]).

In a previous work [8], we have qualitatively identified the dependence among output errors as one of the factors affecting the effectiveness of ECOC decomposition methods [4]. Error Correcting Output Coding (ECOC) are classification methods based on a decomposition of a multiclass problem in a set of two-class subproblems and on a successive reconstruction of the original polychotomy, exploiting the error recovering capabilities of error correcting output codes. We have experimentally showed that the implementation of these methods with an ensemble of parallel and independent non linear dichotomizers (ECOC PND) outperforms the implementation with a single monolithic multi layer perceptron (ECOC MLP). This result can be ascribed to the better effectiveness of error correcting output coding methods when the errors on different code bits are low dependent.

In this paper, we apply the proposed mutual information measures for quantitatively evaluating the dependence among output errors in ECOC classifiers, analyzing the relations between this dependence and the performances of ECOC methods through an extensive experimentation.

The paper is structured as follows. In the next section we summarize the main characteristics of the measures

based on mutual information for evaluating the dependence among output errors. Sect. 3 presents the experimental setup, the results and the discussion, and, then, the section of conclusions summarizes the main results and the incoming developments of this work.

2 Mutual Information and Dependence between Output Errors

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

Let us represent the correct outputs as $\boldsymbol{c} = [c_1, c_2, \ldots, c_l]$ and the computed outputs of a learning machine as $\hat{\boldsymbol{c}} = [\hat{c_1}, \hat{c_2}, \ldots, \hat{c_l}]$. Then we define the corresponding output errors as $\boldsymbol{e} = [e_1, e_2, \ldots, e_l]$, where e_i expresses the error on the i^{th} output of the learning machine. Defining e_i as the absolute error ($e_i = |c_i - \hat{c_i}|, \forall i = 1, \ldots, l$), we can reduce the weight of outliers.

The outputs of learning machines can be considered correct with respect to an assigned tolerance $\delta > 0$ if $\forall i, e_i < \delta$. 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. Analogously, in a regression problem using the ϵ insensitive loss function [17], usually used with support vector machines, it is natural to associate δ with ϵ itself.

In order to compute the mutual information among the output errors of a learning machine, we have to discretize its outputs. Representing the output errors as a vector $\boldsymbol{e} = [e_1, e_2, \ldots, e_l]$, 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. The width of each interval $bin(j)$ is equal except possibly the first one.

We define $e_k^{(i)}$ as the output error of the i^{th} pattern on the k^{th} output and e_{kj} as the number of the $e_k^{(i)}$ values falling in the interval $bin(j)$:

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

where N is the cardinality of the data set. The *discrete probability function* $p(e_{kj})$ is defined as:

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

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

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

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

We define the *mutual information error* I_E as the mutual information of the l output errors:

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

The mutual information error (eq. 1) 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 the similarity of the probability distributions of the output errors.

Using the notion of correctness of the outputs defined 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}:

$$I_{SE}(e_1, \ldots, e_l) = \sum_{\mathcal{J}} p(e_{1j_1}, \ldots, e_{lj_l}) \log \left(\frac{p(e_{1j_1}, \ldots, e_{lj_l})}{p(e_{1j_1}) \ldots p(e_{lj_l})} \right) \quad (2)$$

where

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

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

The mutual information specific error 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.

It is worth noting that, in the computation of the mutual information error, the *curse of dimensionality* [2] problem can arise, as the computation of $I_E(e_1, \ldots, e_l)$ 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.

These problems can be tackled by 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:

$$\Phi_R = \sum_{i=1}^{l} \sum_{j=1}^{l} I_E(e_i, e_j) \tag{3}$$

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:

$$\Phi_S = \sum_{i=1}^{l} \sum_{j=1}^{l} I_{SE}(e_i, e_j) \tag{4}$$

These indices can be used as substitutes 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. 3 and 4) are not equivalent to the corresponding equations 1 and 2 of the mutual information among all output errors. Recall that eq. 3 and 4 consider only the mutual information between pairs of output errors, while eq. 1 and 2 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.

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. 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 behavior of different architectures of learning machines.

3 Estimating the Dependence between Output Errors in ECOC Learning Machines

In this section we analyze the dependence among output errors of *monolithic Error correcting Output Coding* [4, 8] (ECOC *monolithic* for short) and *ECOC Parallel Non linear Dichotomizers* [9] (ECOC *PND* for short) learning machines, using the proposed mutual information based measures.

3.1 The problem

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 [12], 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.

We know that the effectiveness of ECOC classification methods depends on the dependence among codeword bits errors [12, 8]. In our experimentation we evaluate this dependence in ECOC *monolithic* and ECOC *PND* learning machines.

3.2 The data

In our experiments we have used data sets from the UCI repository of Irvine (*glass, letter, optdigits*) [11] and a synthetic data set (*d5*) made up by five three-dimensional classes, each composed by two normal distributed disjoint clusters of data [1].

We have used, both for training the learning machines and for evaluating the dependence among the output errors, *NEURObjects* [16], a set of C++ library classes

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

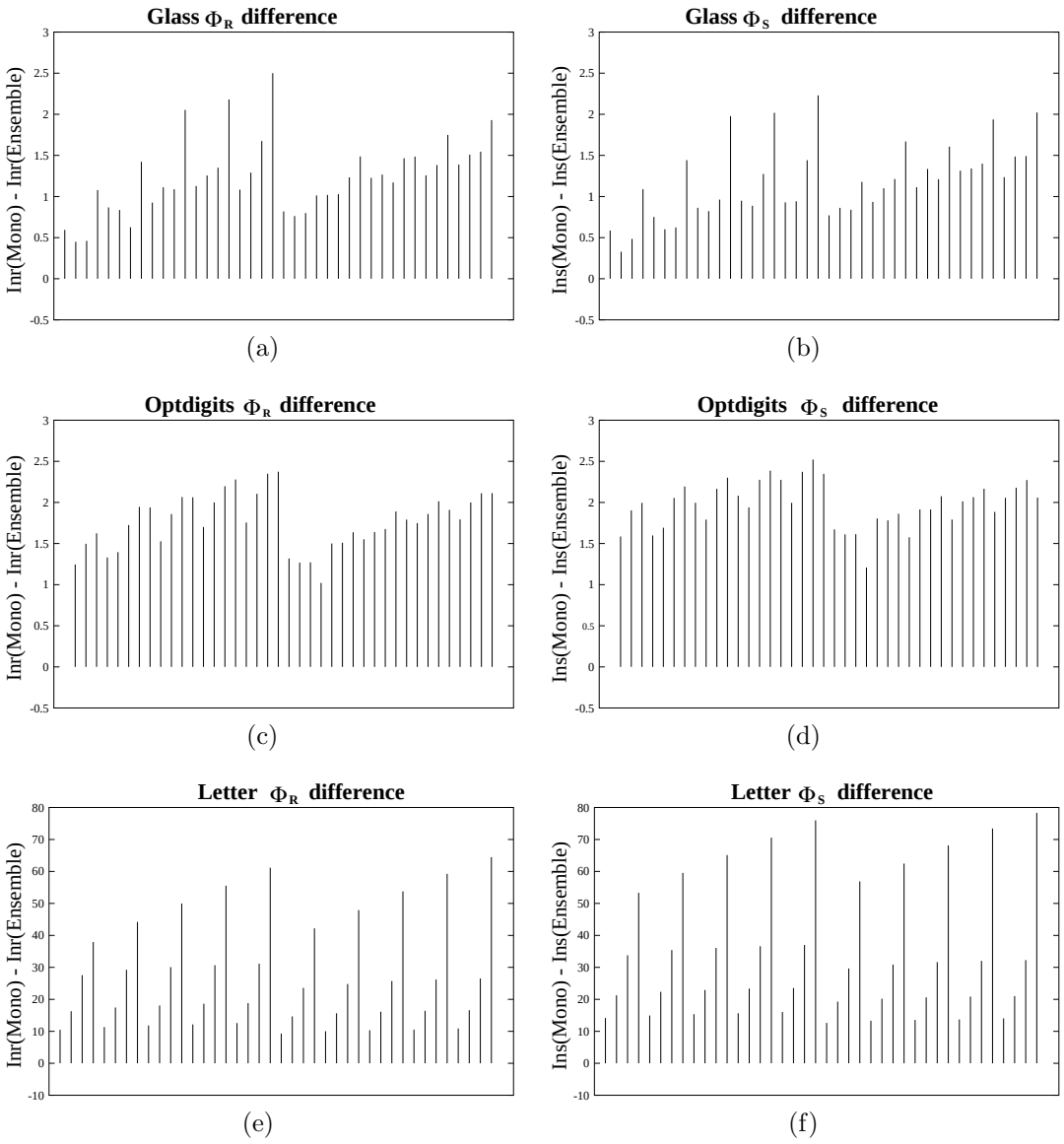

Figure 1: Difference of the pairwise mutual information error index Φ_R and of the pairwise mutual information specific error index Φ_S between ECOC *monolithic* and *PND* learning machines on the data sets glass (a,b), optdigits (c,d) and letter (e,f).

for neural networks development [2].

3.3 Results and discussion

The first results concerned the comparison of performances of ECOC *monolithic* and ECOC *PND* learning machines obtained with multiple runs of different random initialization of the weights and cross-validation techniques (see Tab. 1).

Then, we have compared the dependence among output errors of ECOC *monolithic* and ECOC *PND* learning machines varying the number of hidden units, the number of discretization intervals (bins) of the output errors, and the values of the output error tolerance δ. 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 discretization intervals $bins \in \{2, 3, 4, 5, 6\}$, for a total of 20 pairs of $(\delta, bins)$.

[2] *NEURObjects* is on line available at http://www.disi.unige.it/person/ValentiniG/NEURObjects.

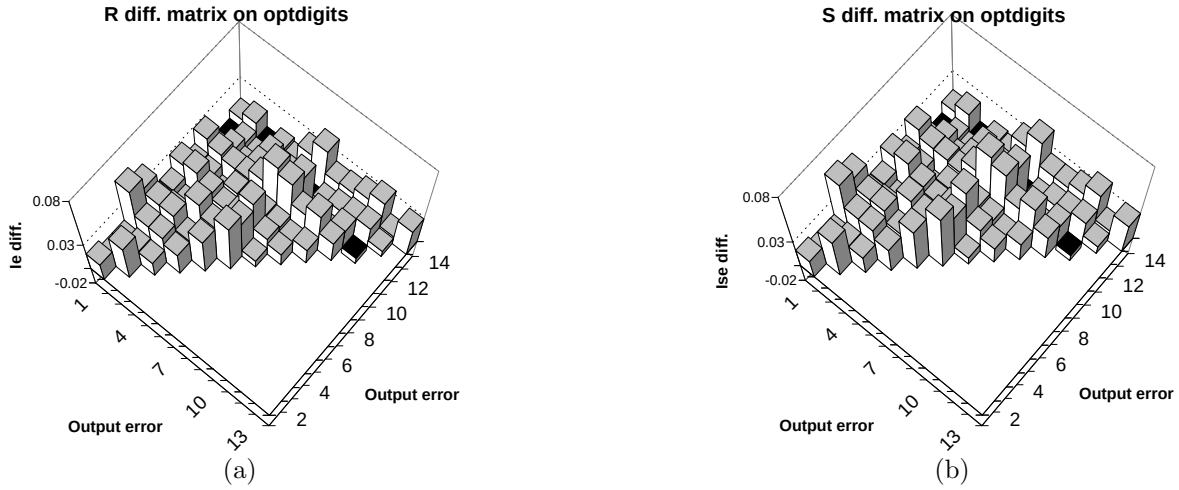

Figure 2: Differences of Pairwise mutual information matrices on the optdigits data set between ECOC *monolithic* and ECOC PND learning machines. R difference matrix (a) and S difference matrix (b).

I_E and I_{SE} among all output errors are greater for ECOC *monolithic* respect to ECOC *PND* learning machines for both the data sets *d5* and *glass*, no matter the structure, the number of intervals and the δ values used. Due to the dimensional problems described Sect. 2, I_E and I_{SE} values have not been computed on *letter* and *optdigits*. For these data sets we evaluated only the pairwise global indices Φ_R and Φ_S.

The differences between the pairwise mutual information error index Φ_R and the pairwise mutual information specific error index Φ_S between ECOC *monolithic* and ECOC *PND* are always positive on all UCI data sets (Fig. 1). In the graphs of Fig. 1, each line corresponds to a different triplet number of hidden units, number of intervals and values of δ. Considering the data set *d5*, only in 2 of the 220 cases we have negative values, showing that Φ_R and Φ_S are higher for ECOC *monolithic*.

Table 1: Performances of MLP ECOC monolithic and *PND* ECOC ensembles.

	MLP ECOC monolithic		**PND ECOC ensemble**	
Data set	**mean**	**stdev**	**mean**	**stdev**
d5	18.31	6.44	12.34	0.74
glass	36.17	4.54	32.05	1.77
letter	6.55	1.91	3.24	0.24
optdigits	3.08	0.47	1.95	0.10

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 $\delta = 0.4$ and a number of intervals equal to 6 for the data set *optdigits* (Fig. 2). This figure represents the differences of R (Fig. 2 a) and S (Fig. 2 b) matrices between ECOC *monolithic* and ECOC *PND* learning machines. Each three-dimensional 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 on the diagonal are the entropy of output errors. Gray bars stand for positive values, and black for negative ones. Only the output error pairs $(1, 13), (3, 14)$ and $(11, 12)$ show negative values of the I_E (Fig. 2 a) and I_{SE} (Fig. 2 b) differences. Similar results are obtained also for the other considered data sets.

We have seen that all the results about I_E, I_{SE}, Φ_R and Φ_S, together with the analysis of the pairwise mutual information matrices R and S show higher values for ECOC *monolithic* learning machines. Moreover, applying the *mutual information error t-test* [10] for evaluating the significance of the differences between the I_E

and I_{SE} values of the two ECOC learning machines, we have verified that in almost all the comparisons we have registered a significant difference with a degree of confidence of 95%.

Consequently, we can state 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* MLP.

4 Conclusions

In this paper we have presented an extensive experimentation for quantitatively evaluating the dependence among codeword bits errors in ECOC learning machines. In particular, we have used measures based on mutual information proposed in [10] for comparing the dependence among output errors between ECOC *monolithic* and ECOC *PND* learning machines.

Our experimentation shows that ECOC *PND* are affected by a lower dependence among the output errors of their decomposition unit compared with the output errors of the corresponding ECOC *monolithic* MLP, suggesting that a low dependence can be achieved implementing the decomposition unit through an ensemble of parallel and independent dichotomizers, such as dichotomic MLP or decision trees [13] or support vector machines [3].

Future developments of this work should consist in quantitatively studying the dependence among output errors in ECOC learning machines architectures that can improve the diversity between the dichotomizers implementing the decision unit.

Acknowledgments

This work has been partially funded by *Progetto finalizzato CNR-MADESS II*, INFM, and University of Genova.

References

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

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

[3] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

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

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

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

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

[8] 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.

[9] 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.

[10] F. Masulli and G. Valentini. Mutual information methods for evaluating dependence among outputs in learning machines. *TR-01-02*, DISI, Università di Genova, 2001. ftp:// ftp.disi.unige.it/ person/ ValentiniG/ papers/ TR-01-02.ps.gz.

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

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

[13] J.R. Quinlan. *C4.5 Programs for Machine Learning*. Morgan Kauffman, 1993.

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

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

[16] G. Valentini and F. Masulli. NEURObjects, a set of library classes for neural networks development. In *Proceedings of IIA'99 and SOCO'99*, pages 184–190, Millet, Canada, 1999. ICSC Academic Press.

[17] V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.