# Effectiveness of Error Correcting Output Codes in Multiclass Learning Problems

Francesco Masulli[1,2] and Giorgio Valentini[1,2]

[1] Istituto Nazionale per la Fisica della Materia
via Dodecaneso 33, 16146 Genova, Italy
[2] DISI - Dipartimento di Informatica e Scienze dell'Informazione
Università di Genova
via Dodecaneso 35, 16146 Genova, Italy
{masulli, valenti}@disi.unige.it

**Abstract.** In the framework of decomposition methods for multiclass classification problems, error correcting output codes (ECOC) can be fruitfully used as codewords for coding classes in order to enhance the generalization capability of learning machines. The effectiveness of error correcting output codes depends mainly on the independence of codeword bits and on the accuracy by which each dichotomy is learned. Separated and non-linear dichotomizers can improve the independence among computed codeword bits, thus fully exploiting the error recovering capabilities of ECOC. In the experimentation presented in this paper we compare ECOC decomposition methods implemented through monolithic multi-layer perceptrons and sets of linear and non-linear independent dichotomizers. The most effectiveness of ECOC decomposition scheme is obtained by *Parallel Non-linear Dichotomizers* (*PND*), a learning machine based on decomposition of polychotomies into dichotomies, using non linear independent dichotomizers.

## 1 Introduction

Error correcting output codes (ECOC) [3] can be used in the framework of decomposition methods for multiclass classification problems to enhance the generalization capability of learning machines.

In [5,6], Dietterich and Bakiri applied ECOC to multiclass learning problems. Their work demonstrated that ECOC can be useful used not only in digital transmission problems [12], but also can improve the performances of generalization of classification methods based on distributed output codes [20]. In fact, using *codewords* for coding classes leads to classifiers with error recovering abilities. The learning machines they proposed are multi-layer perceptrons (MLP) [19] or decision trees [10] using error correcting output codes and with implicit dichotomizers learning in a way dependent on the others. We will call classifiers of this kind as *monolithic classifiers*.

In this paper we outline that on one hand the approach based on monolithic classifiers reduces the accuracy of the dichotomizers, and on the other hand

J. Kittler and F. Roli (Eds.): MCS 2000, LNCS 1857, pp. 107–116, 2000.

the dependency among codeword bits limits the effectiveness of error correcting output codes [18]. On the contrary, we show that the correlation among codeword bits can be lowered using separated and independent learning machines. In fact, the error recovering capabilities of ECOC can be used in the framework of the decomposition of polychotomies into dichotomies, associating each codeword bit to a separated dichotomizer and coming back to the original multiclassification problem in the reconstruction stage [15,13]. However, in real applications, the decomposition of a polychotomy gives rise to complex dichotomies that in turn need complex dichotomizers. Moreover, decompositions based on error correcting output codes can sometimes produce very complex dichotomies.

For these reasons, in this paper we propose to implement decomposition schemes generated via error correcting output codes using *Parallel Non-linear Dichotomizers* (*PND*) model [21,14] that is a learning machine based on decomposition of polychotomies into dichotomies making use of dichotomizers non-linear and independent on each other. In this way we can combine the error recovering capabilities of ECOC codes with a high accurate dichotomizers.

In the next section we introduce the application of ECOC to polychotomy problems. In Sect.s 3 and 4, an experimental comparison of monolithic and decomposition based classifiers is reported and discussed. Conclusions are given in Sect. 5.

## 2 ECOC for Multiclass Learning Problems

In classification problems based on decomposition methods[1], usually we code classes trough binary strings, or codewords. ECOC coding methods can improve performances of the classification system, as they can recover errors produced by the classification system [3].

Let be a $K$ classes polychotomy(or *K-polychotomy*) $\mathcal{P} : \mathbf{X} \rightarrow \{C_1, \ldots, C_k\}$, where $\mathbf{X}$ is the multidimensional space of attributes and $C_1, \ldots, C_k$ are the labels of the classes. The decomposition of the K-polychotomy generates a set of $L$ dichotomizers $f_1, ..., f_L$. Each dichotomizer $f_i$ subdivides the input patterns in two complementary superclasses $\mathcal{C}_i^+$ and $\mathcal{C}_i^-$, each of them grouping one or more classes of the $K$-polychotomy. Let be also a *decomposition matrix* $D = [d_{ik}]$ of dimension $L \times K$ represents the decomposition, connecting classes $C_1, \ldots, C_k$ to the superclasses $\mathcal{C}_i^+$ and $\mathcal{C}_i^-$ identified by each dichotomizer. An element of $D$ is defined as:

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

When a polychotomy is decomposed into dichotomies, the task of each dichotomizer $f_i : \mathbf{X} \rightarrow \{-1, 1\}$ consists in labeling some classes with $+1$ and others with $-1$. Each dichotomizer $f_i$ is trained to associate patterns belonging to class $C_k$ with values $d_{ik}$ of the decomposition matrix $D$. In the decomposition matrix,

[1] A more detailed discussion of decomposition methods for classification is presented in [13].

rows correspond to dichotomizers tasks and columns to classes. In this way, each class is univocally determined by its specific codeword. Using ECOC codes as codewords we can achieve a so-called ECOC decomposition (Fig. 1).

$$\begin{pmatrix} +1 & +1 & +1 & -1 \\ +1 & +1 & -1 & +1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & -1 & -1 & +1 \\ +1 & -1 & -1 & -1 \end{pmatrix}$$

**Fig. 1.** ECOC decomposition matrix for a 4 classes classification problem.

After the a-priori decomposition, the dichotomizers $f_i$ are trained to associate patterns belonging to class $C_k$ with values $d_{ik}$ of the decomposition matrix $D$, their outputs are used to reconstruct the polychotomy in order to determine the class $C_i \in \{C_1, \ldots, C_k\}$ of the input patterns, using a suitable measure of similarity. The polychotomizer then chooses the class whose codeword is the *nearest* to that computed by the set of dichotomizers:

$$class_{out} = \arg \max_{1 \leq i \leq K} Sim(F, c_i) \tag{1}$$

where $class_{out}$ is the class computed by the polychotomizer, $c_i$ is the codeword of class $C_i$, the vector $F$ is the codeword computed by the set of dichotomizers, and $Sim(x, y)$ is a general similarity measure between two vectors $x$ and $y$, e.g. Hamming distance or $L_1$ or $L_2$ norm distances for dichotomizers with are continuous outputs.

It is worth noting that classifiers based on decomposition methods and classifiers based on ensemble averaging methods [17,9] share the idea of using a set of learning machines acting on the same input and recombining their outputs in order to make decisions; the main difference lies in the fact that in classifiers based on decomposition methods the task of each learning machine is specific and different from that of the others.

There are two main approaches to the design of a classifier using ECOC codes:

– The first codes directly the outputs of a monolithic classifier, such us a MLP, using ECOC [5,6].

– The second is based on the usage of ECOC in the framework of decomposition of polychotomies into dichotomies, and leads to the distribution of the learning task among separated and independent dichotomizers. 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 [1]), or *Parallel Non-linear Dichotomizers* (*PND* ) if the dichotomizers are non-linear [21,14].

*Parallel Non-linear Dichotomizers* (*PND*) are multiclassifiers based on the decomposition of polychotomies into dichotomies, using dichotomizers solving their classification tasks independently from each other [21,14]. Each dichotomizer is implemented by a separate *non-linear* learning machine, and learns a different and specific dichotomic task using a training set common to all the dichotomizers. In the reconstruction stage a $L_1$ norm or another similarity measure between codewords is used to predict classes of unlabeled patterns.

*Parallel Linear Dichotomizers* (*PLD*) are also multiclassifiers based on decomposition of polychotomies into dichotomies, but each dichotomizer is implemented by a separate *linear* learning machine (see, e.g., [1]).

Error correcting codes are effective if errors induced by channel noise on single code bits are independent. In [18], Peterson showed that if errors on different code bits are correlated, the effectiveness of error correcting code is reduced. Moreover, 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.

Monolithic ECOC classifiers implemented on MLPs show an higher correlation among codeword bits compared with classifiers implemented using parallel dichotomizers. In fact, outputs of monolithic ECOC classifiers share the same hidden layer of the MLP, while *PND* dichotomizers, implemented with a separated MLP for each codeword bit, have their own layer of hidden units, specialized for a specific dichotomic task.

Moreover, concerning decomposition methods implemented as PLD [1], we point out that this approach reduces the correlation among codeword bits, but error recovering capabilities induced by ECOC are counter-balanced by higher error rates of linear dichotomizers.

In next section, we will experimentally test the following hypotheses about the effectiveness of ECOC:

**Hypothesis 1** *Error correcting output codes are more effective for PND classifiers rather than monolithic MLP classifiers.*

**Hypothesis 2** *In PLD error recovering induced by ECOC is counter-balanced by the higher error rate of the dichotomizers.*

**Table 1.** Data sets general features. The data sets *glass*, *letter* and *optdigits* data sets are from the *UCI repository* [16].

| Data set | Number of attributes | Number of classes | Number of training samples | Number of testing samples |
|---|---|---|---|---|
| *p6* | 3 | 6 | 1200 | 1200 |
| *p9* | 5 | 9 | 1800 | 5-fold cross-val |
| *glass* | 9 | 6 | 214 | 10-fold cross-val |
| *letter* | 16 | 26 | 16000 | 4000 |
| *optdigits* | 64 | 10 | 3823 | 1797 |

## 3 Experimental Results

In order to verify the hypotheses stated above, we have compared classification performances of *Parallel Non-linear Dichotomizers* (*PND*), *Parallel Linear Dichotomizers* (*PLD*) and monolithic classifiers implemented by MLP, using both ECOC and one-per-class (OPC)[2] decomposition methods.

*PND* are implemented by a set of multi-layer perceptrons with a single hidden layer, acting as dichotomizers, and *PLD* are implemented by a set of single layer perceptrons.

Monolithic MLP are built using a single hidden layer and sigmoidal activation functions, both in hidden and output neurons. The number of neurons of the hidden layer amounts roughly from ten to one hundred according to the complexity of the data set to be learned.

The programs used in our experiments have been developed using *NEURObjects* [22], a C++ library for neural networks development. We have used different data sets, both real and synthetic, as shown in Tab. 1. The data sets *p6* and *p9*, are synthetic and composed by normal distributed clusters associated. *p6* contains 6 class with connected regions, while the regions of the 9 classes of *p9* are not connected. *glass*, *letter* and *optdigits* data sets are from the *UCI repository* [16].

In the experimentation we used resampling methods, using a single pair of training and testing data set or the *k-fold cross validation* [4]. In particular the first (an simpler) form has been used for the data sets *p6, letter, optdigits*, and cross validation for the data sets *p9* and *glass*. For testing the significance of differences in performances of two different classification systems applied to the same data set, we have used *Mc Nemar*'s test [8] and the *k-fold cross validated paired t test* [7].

[2] In One-Per-Class (OPC) decomposition scheme (see, e.g., [2]), each dichotomizer $f_i$ have to separate a single class from all the others. As a consequence, if we have $K$ classes, we will use $K$ dichotomizers.

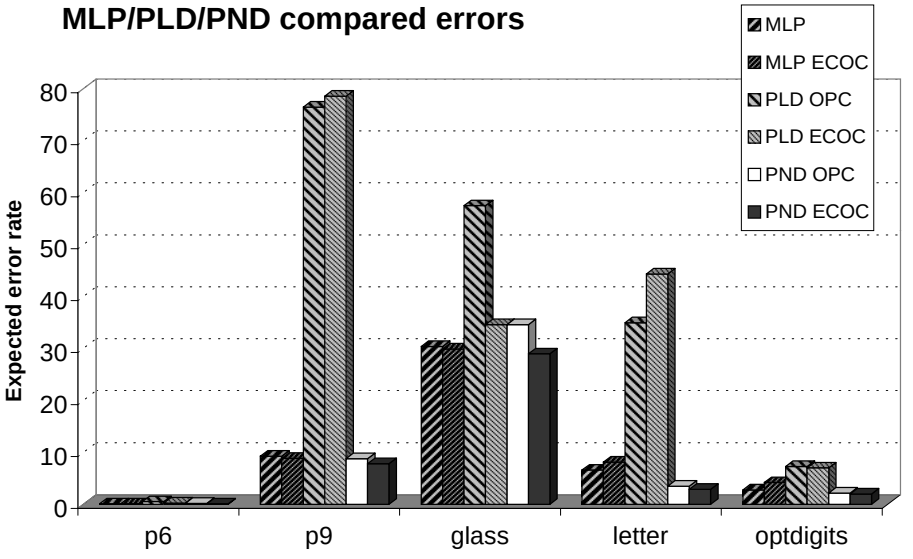


**Fig. 2.** Comparisons of classification expected error estimates over different data sets.

Fig. 2, shows the comparison the performances in classification of MLP, PLD and PND over the considered data sets.

Concerning monolithic MLP standard (OPC) and ECOC MLP, over data sets *p6, p9* and *glass* does not exist statistically significant difference between, but over *letter* and *optdigits* standard MLP performs better. In other words, *ECOC MLP monolithic classifiers do not outperform standard MLP*. This result is in contrast with Dietterich and Bakiri's thesis [6] stating that ECOC MLP outperform standard MLP. Note that, however, Dietterich and Bakiri themselves, in the experimentation over the same data set *letter* we have used, obtains better performances for standard MLP.

Concerning *PLD*, over data sets *p6*, *p9*, and *optdigits* there is no significant statistical difference among OPC and ECOC decomposition, while over *glass PLD* ECOC outperforms all other types of polychotomizers, but with *letter PLD* OPC achieve better results.

Considering *PND*, for data sets *p6* and *optdigits* no significant differences among OPC and ECOC *PND* can be noticed. Over the *p9* data set, ECOC shows expected errors significantly smaller than OPC. Expected errors over *glass* and *letter* data sets are significantly smaller for ECOC compared with OPC. So

we can see that *ECOC PND show expected error rates significantly lower than OPC PND*.

We can remark that, on the whole, expected errors are significantly smaller for *PND* compared with direct monolithic MLP classifiers and *PLD*. Moreover, *PLD* shows higher errors over all data sets, and in particular it fails over *p9* that is an hard non-linearly separable synthetic data set.

We have seen that ECOC MLP classifiers do not outperform standard MLP; moreover ECOC *PND* show expected error rates significantly lower than OPC *PND*. Also, ECOC *PND* largely outperform ECOC *PLD*. It follows that *Error correcting output codes are more effective for PND classifiers rather than direct MLP and PLD classifiers*. Then hypotheses 1 and 2 have been validated by the shown experiments.

## 4 Discussion

In [11], on the basis of geometrical arguments, it has been shown that, using ECOC codes, decision boundaries among classes are learned several times, and however at least a number of times equal to the minimal Hamming distance among codeword of the classes, while standard classifiers learn decision boundaries only two times. In this way ECOC classifiers can recover errors made by some dichotomizers. Moreover, in [5,11] it has been stated that ECOC classifiers should be preferred to directs standard classifiers, as they reduce error bias and variance more than standard classifiers and present experimental results confirming these hypotheses, with the exception of some cases over complex data sets (such us *letter* from UCI repository) where standard MLP classifiers perform better than ECOC MLP.

Our experimentation has pointed out that not always ECOC MLP outperform standard MLP classifiers, while we found a significant difference between ECOC and OPC *PND* performances (fig. 2).

ECOC codes have been originally used to recover errors in serial transmission of messages coded as bits sequences [3], supposing that channel noise induces errors in random and not correlated positions of the sequence. On the contrary, in a classification problem, each codeword bit corresponds to a particular dichotomy, and then similar dichotomizers can induce correlations among codeword bits. As shown by Peterson [18], the effectiveness of error correcting output codes decreases, if the errors on different codeword bits are correlated. ECOC algorithms used to recover errors in serial data transmission do not care about any correlations among codeword bits, and then a transformation of these algorithms for classification problems must at least provide for a control to avoid the generation of identical dichotomizers. More specifically, effectiveness of ECOC codes applied to classification systems depends mainly on the following elements:

1. Error recovering capabilities of ECOC codes.
2. Codeword bits correlation.
3. Accuracy of dichotomizers.

Error recovering capabilities of ECOC codes depends on the minimal Hamming distance among codeword of classes, and it is a property of the ECOC algorithm used. Accuracy of dichotomizers depends on the difficulty of the dichotomization problems (for example if the dichotomy is linearly separable or not). Accuracy depends also on the structure and properties of the dichotomizer and on the cardinality of the data set: A dichotomizer with too parameters with respect to the data set size will be subjected to overfitting and an high error variance. Correlation among computed ECOC codeword bits is less for *PND* compared to MLP classifiers: in *PND* each codeword bit is learned and computed by its own MLP, specialized for its particular dichotomy, while in monolithic classifiers each codeword bit is learned and computed by linear combinations of hidden layer outputs pertaining to one and only shared multi-layer perceptron. Hence, interdependence among MLP ECOC outputs lowers the effectiveness of ECOC codes for this kind of classifiers. Moreover, we point out that a "blind" ECOC decomposition can in some cases generate complex dichotomies, counter-balancing error recovering capabilities of error correcting output codes, especially if dichotomizers are too simple for their dichotomization task (with respect to the data set cardinality), as in the case of *PLD*. *PND*, instead, join independence of dichotomizers (low correlation among codeword bits) with a good accuracy of their non linear dichotomizers. These conditions are both necessary for the effectiveness of ECOC codes in complex classification tasks.

## 5 Conclusions

Decomposition methods for multiclass classification problems constitute a powerful framework to improve generalization capabilities of a large set of learning machines. Moreover, a successful technique to improve generalization capabilities of classification systems is based on Error correcting output codes (ECOC) [5, 6].

Our experimental results show that ECOC is more effective if used in the framework of decomposition of polychotomies into dichotomies, especially if non linear dichotomizers, such us multi-layer perceptrons implementing *Parallel Nonlinear Dichotomizers* [21,14] are used for the individual and separated learning of each codeword bit coding a class. Moreover, monolithic classifiers does not fully exploit the potentialities of error correcting output codes, because of the correlation among codeword bits, while *Parallel Linear Dichotomizers* (see, e.g., [1]), even though implementing non linear classifiers starting from linear ones, do not show good performances in case of complex problems, due to the linearity of their dichotomizers.

Effectiveness of error correcting output codes depends on codeword bits correlation, dichotomizers structure, properties and accuracy, and on the complexity of the multiclass learning problem.

On the basis of the experimental results and theoretical arguments reported in this paper we can claim that the most effectiveness of ECOC decomposition scheme can be obtained with *PND*, a learning machine based on decomposi-

tion of polychotomies into dichotomies, that are in turn solved using non linear independent classifiers implemented by MLP.

## Acknowledgments

This work was partially supported by INFM, Università of Genova, Madess II CNR. We thank Eddy Mayoraz for his suggestions and helpful discussions.

## References

1. E. Alpaydin and E. Mayoraz. Combining linear dichotomizers to construct nonlinear polychotomizers. Technical report, IDIAP-RR 98-05 - Dalle Molle Institute for Perceptual Artificial Intelligence, Martigny (Switzerland) 1998. ftp://ftp.idiap.ch/pub/reports/1998/rr98-05.ps.gz.
2. R. Anand, G. Mehrotra, C.K. Mohan and S. Ranka. Efficient classification for multiclass problems using modular neural networks. *IEEE Transactions on Neural Networks*, 6:117–124, 1995.
3. R.C. Bose and D.K. Ray-Chauduri. On a class of error correcting binary group codes. *Information and Control*, (3):68–79, 1960.
4. V. N. Cherkassky and F. Mulier. *Learning from data: Concepts, Theory and Methods*. Wiley & Sons, New York, 1998.
5. T. 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.
6. T. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, (2):263–286, 1995.
7. T.G. Dietterich. Approximate statistical test for comparing supervised classification learning algorithms. *Neural Computation*, 7 (10):1895–1924, 1998.
8. B.S. Everitt. *The analysis of contingency tables*. Chapman and Hall, London, 1977.
9. S. Hashem. Optimal linear combinations of neural networks. *Neural Computation*, 10:599–614, 1997.
10. J.R. Quinlan. *C4.5 Programs for Machine Learning*. Morgan Kauffman, 1993.
11. E. Kong and T. Dietterich. Error - correcting output coding correct bias and variance. In *The XII International Conference on Machine Learning*, pages 313–321, San Francisco, CA, 1995. Morgan Kauffman.
12. S. Lin and D.J.Jr. Costello. *Error Control Coding: Fundamentals and Applications*. Prentice-Hall, Englewood Cliffs, 1983.
13. F. Masulli and G. Valentini. Comparing decomposition methods for classification. In *KES'2000, Fourth International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies*, Brighton, England. (in press).
14. F. Masulli and G. Valentini. Parallel Non linear Dichotomizers. In *IJCNN2000, The IEEE-INNS-ENNS International Joint Conference on Neural Networks*, Como, Italy. (in press).
15. E. Mayoraz and M. Moreira. On the decomposition of polychotomies into dichotomies. In *The XIV International Conference on Machine Learning*, pages 219–226, Nashville, TN, July 1997.

16. C.J. Merz and P.M. Murphy. UCI repository of machine learning databases, 1998. www.ics.uci.edu/mlearn/MLRepository.html.
17. M.P. Perrone and L.N. Cooper. When networks disagree: ensemble methods for hybrid neural networks. In Mammone R.J., editor, *Artificial Neural Networks for Speech and Vision*, pages 126–142. Chapman & Hall, London, 1993.
18. W.W. Peterson and E.J. Jr. Weldon. *Error correcting codes*. MIT Press, Cambridge, MA, 1972.
19. D.E. Rumelhart , G.E. Hinton and R.J. Williams. Learning internal reperesentations by error propagation. In Rumelhart D.E., McClelland J.L., editor, *Parallel Distributed Processing: Explorations in the Microstructure of Conition*, volume 1, chapter 8. MIT Press, Cambridge, MA, 1986.
20. T. J. Sejnowski and C. R. Rosenberg. Parallel networks that learn to pronounce english text. *Journal of Artificial Intelligence Research*, (1):145–168, 1987.
21. G. Valentini. Metodi scompositivi per la classificazione. Master's thesis, Dipartimento di Informatica e Scienze Informazione - Università di Genova, Genova, Italy, 1999.
22. 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.