Comparing Decomposition Methods for Classification

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

Abstract

Decomposition methods for multiclass classification problems constitute a powerful framework to improve generalization capabilities of a large set of learning machines, including support vector machines and multi-layer perceptrons. We present a review of the main decomposition approach to classification and an experimental comparison of One-Per-Class (OPC), Correcting Classifiers (CC) and Error Correcting Output Codes (ECOC) decomposition methods implemented using multi-layer perceptrons as dichotomizers. The results show that CC and ECOC outperform OPC over the considered data sets.

1 Introduction

Decomposition methods for classification split a multiclass problem, or *polychotomy*, in a series of independent twoclass problems (*dichotomies*) and recompose them using the outputs of dichotomizers, in order to reconstruct the original polychotomy [10, 12, 8].

Learning machines implementing decomposition methods are composed by two main units:

- *Decomposition Unit* that analyzes the input pattern and calculates the codeword using an assigned decomposition scheme.
- *Decision Unit* that associates the computed codeword with a class.

In the next section we discuss the decomposition approach to classification and present the main decomposition methods proposed in the recent machine learning literature. In Sect. 3 we present an experimental comparison of three decomposition methods using two real world data bases. Conclusions are given in Sect. 4.

2 Decomposition approaches to classification

In the framework of *decomposition methods* for classification, the problem complexity is reduced through the decomposition of the polychotomy in less complex subproblems. The resultant dichotomies are implemented by learning machines whose assignments concern with partitioning of patterns in two superclasses aggregating disjoint groups of classes [10, 12, 1].

We shall analyze now the core of this approach to supervised classification, namely the decomposition of polychotomies and the reconstruction of polychotomies (or decision) stages.

2.1 Decomposition stage

Let be $\mathbf{X}$ a multidimensional space of attributes and $C_1, \ldots, C_k$ the labels of classes. Then, the decomposition of a K classes polychotomy (or K-*polychotomy*), that we denote $\mathcal{P} : \mathbf{X} \rightarrow \{C_1, \ldots, C_k\}$, generates a set of L dichotomizers $f_1, ..., f_L$.

The dichotomizer f_i is a discriminating function that subdivides the input patterns in two separated superclasses $\mathcal{C}_i^+$ and $\mathcal{C}_i^-$, each of them grouping a subset of classes of the K-polychotomy. A *decomposition matrix* $D = [d_{ik}]$ of dimension $L \times K$ represents the decomposition and connects classes $C_1, \ldots, C_k$ to the superclasses $\mathcal{C}_i^+$ and $\mathcal{C}_i^-$. Its elements are defined as:

$$d_{ik} = \begin{cases} +1 & \text{if } C_k \subset \mathcal{C}_i^+ \\ -1 & \text{if } C_k \subset \mathcal{C}_i^- \\ 0 & \text{if } C_k \cap (\mathcal{C}_i^+ \cup \mathcal{C}_i^-) = \emptyset \end{cases} \quad (1)$$

In this way, when a polychotomy is decomposed into dichotomies, the task of each dichotomizer $f : \mathbf{X} \rightarrow \{-1, 0, 1\}$ consists in labeling some classes with $+1$ and others with -1, and in ignoring the remaining classes, labeling them with 0. Each dichotomizer f_i is trained to associate patterns belonging to class C_k with values d_{ik} of the decomposition matrix D.

In a decomposition matrix, rows correspond to dichotomizers tasks and columns to classes and each class is univocally determined by its *codeword.* For instance, in the decomposition matrix for a four classes classification problem shown in Tab. 1, the task of the second dichotomizer, namely f_2, consists in separating the patterns belonging to classes

C_1 and C_4 from the patterns of class C_3, while patterns belonging to class C_2 are ignored. The third column of the decomposition matrix represents the codeword $[0, -1, +1, +1, 0, -1, +1]$ associated to the class C_3.

Table 1: Decomposition matrix.

Dichotomizers tasks	Columns: class codewords			
	C_1	C_2	C_3	C_4
f_1	+1	-1	0	-1
f_2	+1	0	-1	+1
f_3	+1	-1	+1	0
f_4	-1	0	+1	+1
f_5	+1	+1	0	-1
f_6	+1	-1	-1	+1
f_7	-1	+1	+1	0

2.2 Reconstruction stage

In the reconstruction or decision stage, the outputs of dichotomizers, trained as previously explained, are used to reconstruct the polychotomy in order to determine the class $C_i \in \{C_1, \ldots, C_k\}$ of the unlabeled patterns. The vector $F = (f_1, f_2, \ldots, f_L)$ of the outputs of the L dichotomizers is compared with the codewords $c_i, 1 \leq i \leq K$ of the classes (that are the columns of the decomposition matrix), using a suitable measure of similarity. The polychotomizer then chooses the class whose codeword is the *nearest* to F:

$$class_{out} = \arg \max_{1 \leq i \leq K} Sim(F, c_i) \tag{2}$$

where $class_{out}$ is the class computed by the polychotomizer, c_i is the codeword of class C_i, and $Sim(x, y)$ is a general similarity measure between two vectors x and y.

According to the characteristics of the of the dichotomizers outputs, we can use the Hamming distance or its variations as similarity measures if the outputs of the dichotomizers are discrete, and the inner product or the L_1 or L_2 norm distances, if the dichotomizers outputs are continuous.

2.3 Decomposition schemes

There are many possibilities for decomposing a polychotomies into dichotomies. In the following of this subsection we will depict the more popular approaches.

2.3.1 Minimal and maximal decompositions

The most compact decomposition of a K-polychotomy into dichotomies is based on a scheme with $L = \lceil \log_2(K) \rceil$ different dichotomic tasks [10].

Instead, the decomposition of a K-polychotomy containing all the possible dichotomies has a cardinality of $L = \frac{1}{2}(3^K + 1) - 2^K$, but only $2^{K-1} - 1$ of the resultant dichotomies include all the classes simultaneously. In fact, dichotomies f' and f such that $f' = -f$ are equivalent, and, moreover, trivial dichotomies like $f^{-1}(-1) = \emptyset$ are not useful for classification tasks.

An example of a minimal and maximal decomposition matrix for a 4 classes polychotomy is shown in Tab. 2.

Table 2: Minimal (left) and maximal (right) decomposition matrices. Lines correspond to dichotomies, columns correspond to classes.

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

2.3.2 One-Per-Class

According to the One-Per-Class (OPC) decomposition scheme [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.

An example of a one-per-class decomposition matrix for a 4 classes polychotomy is shown in Tab. 3.

Table 3: One per class decomposition matrix.

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

The standard approach to this method involves the separate training of K different dichotomizers $f_1, \ldots, f_K$.

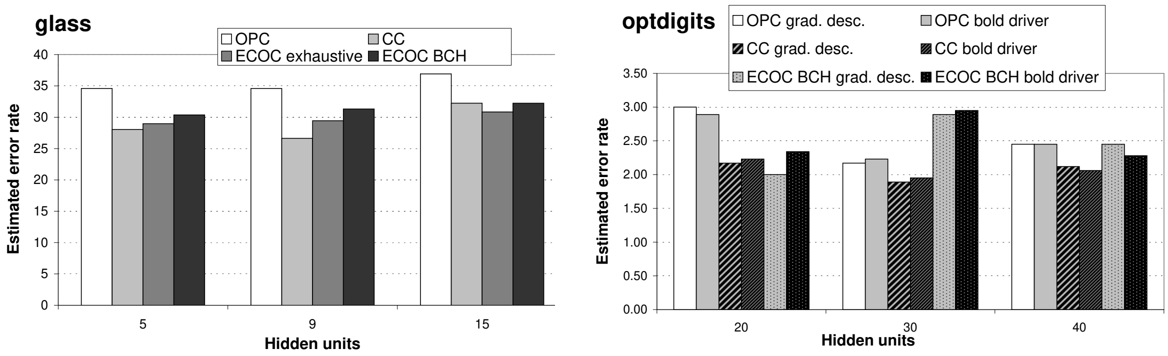
The assignment of a new input $\mathbf{x}$ to a certain class can be performed using similarity measures or feeding all f_i on $\mathbf{x}$, and assigning to $\mathbf{x}$ the class j if f_j has provided the higher value.

2.3.3 PairWise Coupling

In the *PairWise Coupling* (PWC) classification scheme [12], the task of each dichotomizer f_i is to separate a class c_i form class c_j, ignoring all other classes, leading to $\binom{K}{2}$ pairwise dichotomies.

Variants of the PWC scheme are the *Correcting Classifiers* (CC) and the *PairWise Coupling Correcting Classifiers* (PWC-CC) that can reduce the noise originated in the PWC scheme by the processing of not pertinent information performed by the PWC dichotomizers [12].

Examples of a Pairwise Coupling and a CC decomposition matrices for a 4 classes polychotomy are shown in Tab. 4.



(a) (b)

Figure 1: Comparison of performances of different decomposition methods on glass (a) and optdigits (b) data sets of the UCI.

Table 4: Standard PWC (left) and CC (right) decomposition matrices.

$$\begin{pmatrix} +1 & -1 & 0 & 0 \\ +1 & 0 & -1 & 0 \\ +1 & 0 & 0 & -1 \\ 0 & +1 & -1 & 0 \\ 0 & +1 & 0 & -1 \\ 0 & 0 & +1 & -1 \end{pmatrix} \quad \begin{pmatrix} +1 & +1 & -1 & -1 \\ +1 & -1 & +1 & -1 \\ +1 & -1 & -1 & +1 \\ -1 & +1 & +1 & -1 \\ -1 & +1 & -1 & +1 \\ -1 & -1 & +1 & +1 \end{pmatrix}$$

2.3.4 Error Correcting Output Codes

Classification based on *Error Correcting Output Codes* (ECOC) is a decomposition method borrowed from coding theory [3].

Dietterich and Bakiri [4, 5] proposed this decomposition scheme with the aim of improving the generalization capabilities of NETtalk classifiers based on distributed output codes [13].

It is worth noting that the notion of *codeword* used for class labeling suggests *ipso facto* the idea of adding *error recovering* capabilities to decomposition methods in order to obtain classifiers less sensitive to noise [6, 8].

An example of an ECOC decomposition matrix for a 4 classes polychotomy is shown in Tab. 5.

Table 5: ECOC decomposition matrix.

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

An ECOC allows a correct classification even if a subset of dichotomizers gives wrong classification results. A good ECOC is characterized by:

1. *Columns separation.* A codeword must be far from the other codewords of the decomposition matrix, according to an assigned distance measure. For binary strings we can use the Hamming distance.

2. *Row separation.* Each dichotomizer f_i, computing the i^{th} bit of the output code, should not be correlated with the remaining dichotomizers $f_j, \quad i \neq j$. Then, we should also maximize the Hamming distance between each row of the decomposition matrix and between each row and the complements of the others.

The maximal number of errors (MaxNE) that can be corrected in an ECOC based decomposition is [6]:

$$\text{MaxNE} = \left\lfloor \frac{\Delta_{\mathbf{D}} - 1}{2} \right\rfloor \tag{3}$$

where $\Delta_{\mathbf{D}}$ is the minimal Hamming distance between each pair of columns of the decomposition matrix $\mathbf{D}$.

3 Experimental results and discussion

In this section, we present an experimental comparison of OPC, CC, and ECOC decomposition methods for classification using *Parallel Non-linear Dichotomizers* (*PND*) [9]. In *PND*, each dichotomizer is implemented by a multi-layer perceptron with a single hidden layer, and the reconstruction stage uses a L_1 norm distance between codewords. The results we show in this section have been obtained on two data sets from the UCI repository [11].

ECOC have been generated through Bose-Chauduri-Hocquenghem (BCH) [3] and exhaustive [5] algorithms. The comparison of the different classification decomposition methods was performed using resampling and cross-validation methods [7] for estimating the expected misclassification risk.

As shown in Fig. 1a, on the first data base we have considered, constituted by the *glass* data set, the decomposition methods based on CC, ECOC BCH and exhaustive ECOC perform better than those based on OPC. No significant differences can be noticed using ECOC BCH and exhaustive ECOC, and, moreover, classifiers based on CC *PND* show error rates significantly lower then those using ECOC only for nets with 9 hidden units.

The performances obtained on the second data base, constituted by the *optdigits* data set, are good for all considered decomposition methods (Fig. 1b). In this case CC outperform the other decomposition methods, even if often the differences are not significant. Moreover, ECOC BCH have error rates significantly higher compared with OPC only in case of 30 neurons in the hidden layer.

In summary, CC and ECOC outperform OPC over all considered data sets, and not sharp difference in performances is noticed between CC and ECOC decomposition methods.

We outline, moreover, the high error correcting capabilities of ECOC and CC codes. This is consistent with the fact that the Hamming distance between class codewords is greater for ECOC and CC than for OPC, and then, according to equation (3), the maximal number of recoverable errors is increased.

4 Conclusions

The theoretical development and the comparative study of the decomposition methods for classification show that these methods can improve generalization capabilities of a large set of learning machines [10, 12, 9, 8].

In the first part of this paper we have discussed the decomposition approach to classification that is a constructive methodology based on the splitting of multiclass problems, or *polychotomies*, in a series of independent two-class problems (*dichotomies*) and on the succeeding recomposition of the original polychotomy using the outputs of the dichotomizers.

Then, we have presented an experimental comparison on two data sets from UCI repository [11] of One-Per-Class (OPC), Correcting Classifiers (CC) and Error Correcting Output Codes (ECOC) decomposition methods implemented using as dichotomizers multi-layer perceptrons (*Parallel Nonlinear Dichotomizers*, *PND*) [9].

Our results shown that CC and ECOC outperform OPC over all considered data sets. Moreover, not evident difference in performances is noticed between CC and ECOC decomposition methods, and ECOC and CC codes show high error correcting capabilities.

References

[1] E. Alpaydin and E. Mayoraz Combining linear dichotomizers to construct nonlinear polychotomizers", institution. Technical report IDIAP-RR 98-05, IDIAP - Dalle Molle Institute for Perceptual Artificial Intelligence, Martigny (Switzerland), 1998

[2] Anand R., Mehrotra G., Mohan C.K. and Ranka S. 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] 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.

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

[6] 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.

[7] A. Krogh, J. Vedelsby. Neural networks ensembles, cross validation and active learning. In Touretzky D. S., Tesauro G., Leen T.K., editors, *Advances in Neural Information Processing Systems*, volume 7, pages 107–115. MIT Press, Cambridge, MA, 1995.

[8] F. Masulli and G. Valentini. Effectiveness of error correcting output codes in multiclass learning problems. In *Proceedings of MCS'2000, First International Workshop on Multiple Classifier Systems*, Cagliari, Italy. (in press).

[9] F. Masulli and G. Valentini. Decomposition methods for classification tasks using Parallel Non linear Dichotomizers. Technical report, DISI - Università di Genova, 2000.

[10] 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.

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

[12] M. Moreira and E. Mayoraz. Improved pairwise coupling classifiers with correcting classifiers. In *Tenth European Conference on Machine Learning*, Chemnitz, Germany, April 1998.

[13] T. J. Sejnowski and C. R. Rosenberg. Parallel networks that learn to pronounce English text. *Journal of Artificial Intelligence Research*, (1):145–168, 1987.