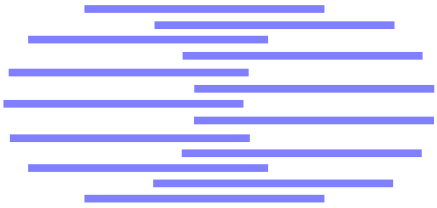
IDIAP
Martigny - Valais - Suisse



On the Decomposition of Polychotomies into Dichotomies

Eddy Mayoraz [†] Miguel Moreira [†]

IDIAP–RR 96-08

December 96

published in
Proceedings of The Fourteenth International Conference on Machine Learning, Nashville, TN, July 1997, 219–226

Dalle Molle Institute for Perceptive Artificial Intelligence • P.O.Box 592 • Martigny • Valais • Switzerland

phone +41 – 27 – 721 77 11
fax +41 – 27 – 721 77 12
e-mail secretariat@idiap.ch
internet http://www.idiap.ch

[†] IDIAP—Dalle Molle Institute for Perceptive Artificial Intelligence, P.O.Box 592, CH-1920 Martigny, Valais, Switzerland

IDIAP Research Report 96-08

On the Decomposition of Polychotomies into Dichotomies

Eddy Mayoraz Miguel Moreira

December 96

published in
Proceedings of The Fourteenth International Conference on Machine Learning, Nashville, TN, July 1997, 219–226

Abstract. Many important classification problems are *polychotomies*, *i.e.* the data are organized into K classes with $K > 2$. Given an unknown function $F : \Omega \rightarrow \{1, \ldots, K\}$ representing a polychotomy, an algorithm aimed at "learning" this polychotomy will produce an approximation of F, based on the knowledge of a set of pairs $\{(\boldsymbol{x}^p, F(\boldsymbol{x}^p))\}_{p=1}^{P}$. Although in the wide variety of learning tools there exist some learning algorithms capable of handling polychotomies, many of the interesting tools were designed by nature for dichotomies ($K = 2$). Therefore, many researchers are compelled to use techniques to decompose a polychotomy into a series of dichotomies in order to apply their favorite algorithms to the resolution of a general problem. A decomposition method based on error-correcting codes has been lately proposed and shown to be very efficient. However, this decomposition is designed only on the basis of K without taking the data into account. In this paper, we explore alternatives to this method, still based on the fruitful idea of error-correcting codes, but where the decomposition is inspired by the data at hand. The efficiency of this approach, both for the simplicity of the model and for the generalization, is illustrated by some numerical experiments.

Acknowledgements: The support of the Swiss National Science Foundation (Grant SNSF 21-46974.96) is gratefully acknowledged.

1 Introduction

Automated learning addresses the general problem of finding an approximation $\hat{F}$ of an unknown function F defined from an *input space* Ω onto an *output space* Σ, given a *training set* : $\{(\boldsymbol{x}^p, F(\boldsymbol{x}^p))\}_{p=1}^{P} \subset \Omega \times \Sigma$. *Classification* problems are characterized by a discrete unordered output space $\Sigma = \{1, \ldots, K\}$ (set of diseases, set of digits, set of letters, etc). F is called a *polychotomy* of Ω if $K > 2$ and a *dichotomy* if $K = 2$. A polychotomy or a dichotomy F of Ω provides a partition of the input space, and for each $k = 1, \ldots, K$, the set $F^{-1}(k)$ is called the *class* k.

There is a wide variety of algorithms available in the literature to handle classification problems. They originate in different domains such as : statistics (*e.g.* Bayesian classifiers, see [DH73]), logic (*e.g.* logical analysis of data [CHI88, BHI+96]), neural networks (*e.g.* perceptron algorithm [Ros58], back-propagation [Wer74]), artificial intelligence (*e.g.* decision trees [BOS84, Qui86]). While some of them are suitable to learn polychotomies (multi-layered perceptrons, decision trees), others are designed only for dichotomies (Bayesian classifiers, perceptrons or logical analysis of data). Since algorithms for learning dichotomies have stronger theoretical roots and are usually better understood, it is tempting to apply them to learn polychotomies by first decomposing the multi-class problem into a series of problems of 2 classes each.

Several *decomposition schemes* have been imagined, associating with a K classes polychotomy F of Ω, a series of L dichotomies of Ω, $f_1, \ldots, f_L$. A *reconstruction method* is coupled with each decomposition scheme, its purpose being the selection of one of the K classes, given the answers of all dichotomies for a particular input data. Among the simplest decomposition schemes frequently used, let us mention the *one-per-class* and the *paiwise coupling*. A polychotomy of K classes is decomposed by the former method into K dichotomies, each of which separating one class from all the others, while the latter will use $K(K-1)/2$ dichotomies, one for each pair of classes, the dichotomy for the pair (k, k') focuses on the separation of classes k and k', ignoring all other classes.

It is worth observing that a similar principle of decomposition/reconstruction is used by most of the methods suited to learn polychotomies, however, this decomposition is implicit and is elaborated during the learning phase, *i.e. a posteriori*. On the contrary, the one-per-class or the paiwise coupling decompositions are explicit and are designed *a priori*, *i.e.* without considering the training set.

In this study, a decomposition scheme, explicit and *a posteriori* (based on the training set) is developed and experimented. The method is strongly inspired by the decomposition based on error-correcting codes (ECOC) proposed by T. Dietterich and G. Bakiri [DB91, DB95, KD95], which is explicit and *a priori*. The latter has been proved to be very efficient, since not only does it allow the use of powerful algorithms specialized for dichotomies to learn polychotomies, but also a significant improvement of the overall generalization is observed when the same type of algorithm (*e.g.* C4.5) is used (i) to learn each dichotomy of the decomposition versus (ii) to learn directly the polychotomy.

In Section 2, a reflexion on the common form of the function computed by the most common learning methods is suggested and the notions of decomposition and reconstruction are formally defined. The interest of an *a posteriori* decomposition is introduced in Section 4 and a simple algorithm is proposed as well as a reconstruction technique. This method is experimented and the numerical results are reported in Section 5. Finally, the main advantages of an *a posteriori* elaboration of an explicit decomposition scheme are summarized in the last section and few lines of further research are enumerated.

2 Decomposition and reconstruction

In order to have a better insight of what is meant by decomposition scheme and by reconstruction, let us analyze the general form of the function $\hat{F}$ produced by typical learning algorithms handling polychotomies.

2.1 General form of $\hat{F}$

The approximation $\hat{F}$ of the unknown function $F : \Omega \rightarrow \{1, \ldots, K\}$ constructed by a learning algorithm can in general be decomposed as follows:

$$\hat{F} = \arg\max_{k \leq K} \circ \quad (\sigma \circ) \; m \circ \hat{\boldsymbol{f}} \tag{1}$$

with

$\hat{\boldsymbol{f}} = (\hat{f}_1, \ldots, \hat{f}_L)$, $\hat{f}_l : \Omega \rightarrow \mathbb{R}$, l^{th} element of the decomposition,

$m : \mathbb{R}^L \rightarrow \mathbb{R}^K$, linear mapping of matrix $\boldsymbol{M} \in \mathbb{R}^L \times \mathbb{R}^K$,

$\sigma : \mathbb{R}^K \rightarrow \mathbb{R}^K$, non linear mapping (optional).

This is obviously the form of the function yielded by a multi-layered perceptron with output coded by "grand-mother cells". The $\hat{f}_l$s are computed by the units of the last hidden layer; $\boldsymbol{M}$ is the matrix of connections between the last hidden layer and the output layer; and σ corresponds to the component-wise application of the transfer function (if any) of the output units.

In the case of decision tree algorithms, one $\hat{f}_l$ is associated with each leaf; $\boldsymbol{M}$ is a $\{0-1\}$-matrix so that $M_{lk} = 1$ if and only if leaf l is labeled by class k; and σ is omitted. One might argue that this Form (1) is not natural for the function of a decision tree, since $\hat{\boldsymbol{f}}$ will always result in a vector with one component set to 1 and all the others set to 0 and consequently, the role of $\arg\max_k$ is trivial. In practice however, widely used decision trees algorithms such as C4.5 are implemented so that for any input $\boldsymbol{x} \in \Omega$, a probability is computed for each leaf (probability that input $\boldsymbol{x}$ will lead to that leaf) and these probabilities might take values other than 0 and 1 (*e.g.* in the case that a test associated with one node of the tree does not apply for $\boldsymbol{x}$) and therefore, Form (1) really suits the function provided by a decision tree.

2.2 Decomposition scheme

The *decomposition scheme* is given by the L functions $f_l : \Omega \rightarrow \mathbb{R}$, $l = 1, \ldots, L$, such that

$$F = \arg\max_{k} \circ \sigma \circ m \circ \boldsymbol{f}.$$

To be *valid*, a decomposition scheme should allow reconstruction, *i.e.* there should not be two points $\boldsymbol{x}, \boldsymbol{y} \in \Omega$ belonging to two different classes such that $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{y})$. Learning methods based on neural networks or decision trees build their decomposition scheme implicitly. In the case of an explicit decomposition, tasks are assigned to f_l, $l = 1, \ldots, L$, and each of them is approximated by a learning algorithm yielding $\hat{f}_l$. When a polychotomy is decomposed into dichotomies, the task of one particular $f_l : \Omega_l \subset \Omega \rightarrow \{-1, +1\}$ consists in associating some classes with $+1$ and some others with -1. Therefore, the overall decomposition scheme can be specified by a *decomposition matrix* $\boldsymbol{D} \in \mathbb{R}^L \times \mathbb{R}^K$ such that

$$D_{lk} = \begin{cases} +1 & \text{if class } k \text{ is associated with } +1 \text{ by } f_l \\ -1 & \text{if class } k \text{ is associated with } -1 \text{ by } f_l \\ 0 & \text{if class } k \text{ does not belong to the task of } f_l \end{cases}$$

For example, the decomposition of the one-per-class and of the paiwise coupling schemes are given by the decomposition matrices of Figure 1 for the case $K = 4$.

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

(a) (b)

Figure 1: Classical decomposition matrices $\boldsymbol{D}$.
(a) illustrates the decomposition matrix of the one-per-class scheme; while the matrix in (b) corresponds to the paiwise coupling scheme.
Each row corresponds to one dichotomy and each column to one class.

2.3 Reconstruction

The *reconstruction* consists in the function $\arg\max_k \circ\sigma\circ m$. Although it is not the only possible choice, the decomposition matrix is clearly a good candidate for the linear mapping $\boldsymbol{M}$. Indeed, if for example $f_{l'}$ associates class k with $+1$, and $f_{l''}$ associates the same class with -1 ($D_{l'k} = +1$ and $D_{l''k} = -1$), a linear recombination of the $\hat{f}_l$s which should produce a value proportional to the probability that an input belongs to class k would naturally use a positive coefficient for $\hat{f}_{l'}$ and a negative coefficient for $\hat{f}_{l''}$ ($M_{l'k} > 0$ and $M_{l''k} < 0$). Unless otherwise specified, in what follows, $\boldsymbol{M}$ will be assumed to be $\boldsymbol{D}$.

Note that most training methods designed to learn dichotomies will produce for each $f_l : \Omega \to \{-1, +1\}$ an approximation of the form $\hat{f}_l = \text{sgn} \circ g_l$, where $g_l : \Omega \to \mathbb{R}$ and sgn is the sign function ($\text{sgn}(a) = 1$ if $a \geq 0$, $\text{sgn}(a) = -1$ otherwise). In this case, it is common to replace $\hat{\boldsymbol{f}}$ by $\boldsymbol{g}$ in (1), since the latter carries more information. For example, when a one-per-class scheme is used with $\boldsymbol{M} = \boldsymbol{D}$ (note that $\boldsymbol{M} = \boldsymbol{I}$, the identity matrix, leads to the same $\hat{F}$) and σ is omitted, the usage of $\hat{\boldsymbol{f}}$ produces a poor $\hat{F}$, since a single error $\hat{f}_l(\boldsymbol{x}) \neq f_l(\boldsymbol{x})$ produces an erroneous $\hat{F}(\boldsymbol{x})$, while with $\boldsymbol{g}$, the function $\hat{F} = \arg\max_k \circ \boldsymbol{g}$ is likely to be more robust.

3 A priori decompositions

There are numerous decomposition schemes for a polychotomy of K classes into dichotomies, among which the one-per-class and the paiwise coupling are the most natural ones. If a compact decomposition is sought, with few dichotomies, there is always a scheme with $L = \lceil \log_2(K) \rceil$ (Figure 2(a)). On the other hand, Figure 2(b) illustrates a decomposition containing all possible dichotomies based on 4 classes and including all of them simultaneously. Note that from the learning point of view, dichotomies given by $f : \Omega \to \{-1, +1\}$ and $f' = -f$ are equivalent, and trivial dichotomies such that $f^{-1}(-1) = \emptyset$ are not interesting. Consequently, it can be easily verified that a polychotomy of K classes can be decomposed into $\frac{1}{2}(3^K + 1) - 2^K$ different dichotomies, $2^{K-1} - 1$ of which include all the K classes simultaneously.

3.1 The error-correcting code concept

In case the size of the model produced is not critical, the idea of T. Dietterich and G. Bakiri is that larger decomposition schemes can be created in such a way that their redundancy is exploited to improve the quality of the approximation. For example, with the decomposition $\boldsymbol{D}$ of Figure 2(b), every two columns are at Hamming distance 4 from each other. Therefore, if for a given $\boldsymbol{x} \in \Omega$ of class

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

(a)

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

(b)

Figure 2: Minimal and maximal decomposition matrices $\boldsymbol{D}$.

k, not more than one $\hat{f}_l$ provides an erroneous answer, the vector $\hat{\boldsymbol{f}}(\boldsymbol{x}) \in \{+1,-1\}^L$ is at Hamming distance at most 1 from the k^{th}column of $\boldsymbol{D}$ and at least 3 from any other column, and consequently, $\hat{F}$ can output the correct answer.

Thus, some robustness is added to $\hat{F}$ by using a larger amount of dichotomies. This error-correcting ability depends on the minimal Hamming distance between two columns of the decomposition matrix. When the latter is a $\{+1,-1\}$-matrix, the number of errors that can be corrected is given by

$$\left\lfloor \frac{\Delta_{\boldsymbol{D}} - 1}{2} \right\rfloor,$$

where $\Delta_{\boldsymbol{D}}$ is the minimal Hamming distance between any two columns of $\boldsymbol{D}$. It can be verified that this error correcting property is achieved when $\boldsymbol{M} = \boldsymbol{D}$ and without σ. As already mentioned, some robustness can be added by replacing $\hat{\boldsymbol{f}}$ by $\boldsymbol{g}$ as long as the g_ls are normalized with each other.

3.2 ECOC reconstruction

In [DB91, DB95, KD95], the authors use functions g_ls with values in the range $[0,1]$ (expressing probabilities), and the reconstruction is done in such a way that the class produced by $\hat{F}$ is the one that minimizes, among all the classes k, the distance —in the L_1 norm— between $\boldsymbol{g}(\boldsymbol{x}) \in [0,1]^L$ and the so-called codeword of class k. This codeword is defined as a 0-1 vector corresponding to the k^{th}column of $\boldsymbol{D}$ where the -1s are replaced by 0s. Note that such an $\hat{F}$ is identical to the one given in (1) with g_ls centered around 0, with $\boldsymbol{M} = \boldsymbol{D}$ and without σ, since

$$\arg\min_k \sum_l \left| g_l - \frac{D_{lk}+1}{2} \right| \quad = \quad \arg\max_k \sum_l (2g_l - 1) D_{lk}$$

holds for any $g_l \in [0,1]$ and $D_{lk} \in \{-1,+1\}$.

3.3 A priori decomposition in ECOC

In their approach, the authors of the ECOC method propose an elaboration explicit and *a priori* of the matrix $\boldsymbol{D}$. They elaborate several algorithms (some of them borrowed from error correcting code theory) for the construction of $-1,+1$ matrices of K columns and a reasonable amount of rows such that pairwise Hamming distance between columns is as large as possible. Simultaneously, for each pair of rows $D_{l'}$ and $D_{l''}$, they maintain a reasonably large Hamming distance both between $D_{l'}$ and $D_{l''}$ and between $D_{l'}$ and $-D_{l''}$. This is necessary since using too similar dichotomies $f_{l'}$ and $f_{l''}$ would lead to two approximations $\hat{f}_{l'}$ and $\hat{f}_{l''}$ strongly correlated which would not significantly increase the amount of information available for the reconstruction.

4 A posteriori decompositions

The one-per-class, the paiwise coupling and the ECOC decomposition schemes are constructed *a priori*, requiring only the knowledge of the number of classes K. With these methods, two classes are gathered or opposed in a dichotomy in a systematic way, which does not depend on the "proximities" of these classes in the input space. If this sounds natural for one-per-class (respectively for paiwise coupling), since each dichotomy is isolating one class from all others (resp. from another one), it is more problematic for the ECOC method in which each dichotomy is defined with all the classes so that half of them selected at random are gathered and opposed to the other half. Consequently, solving each one of the dichotomies is usually not an easy task and indeed, in [DB95] where C4.5 is used to solve the dichotomies, the authors mentioned that the trees generated for each dichotomy were conserably large, in the same order of magnitude as the single tree generated by C4.5 for resolving the polychotomy directly.

The following part of this paper is devoted to the investigation of solutions for the generation of a $\{-1, 0, +1\}$-decomposition matrix $\boldsymbol{D}$ which try to simultaneously ensure that

(i) the minimal *distance* between every two columns $\boldsymbol{D}_{\cdot k'}$ and $\boldsymbol{D}_{\cdot k''}$ is large,

(ii) the minimal *variation* between every two rows $\boldsymbol{D}_{l'\cdot}$ and $\boldsymbol{D}_{l''\cdot}$ is large,

(iii) each row $\boldsymbol{D}_{l\cdot}$ corresponds to one dichotomy which is pertinent according to the relative positions of the classes in Ω.

It should be added that the *distance* between columns or rows is computed with a modified Hamming distance d_H where a 0 coefficient never contributes to the distance :

$$\begin{aligned} &d_H : \{-1,0,+1\}^n \times \{-1,0,+1\}^n \to \mathbb{R}, \\ &d_H(\boldsymbol{a},\boldsymbol{b}) = \sum_{i=1}^{n} d_H(a_i, b_i), \\ &\qquad\qquad d_H(+1,-1) = 1 \quad \text{and} \quad d_H(0,\pm 1) = 0. \end{aligned}$$

Moreover, the *variation* between two rows $\boldsymbol{D}_{l'\cdot}$ and $\boldsymbol{D}_{l''\cdot}$ is defined as the minimum between $d_H(\boldsymbol{D}_{l'\cdot}, \boldsymbol{D}_{l''\cdot})$ and $d_H(\boldsymbol{D}_{l'\cdot}, -\boldsymbol{D}_{l''\cdot})$.

4.1 Pertinent dichotomies

This notion of *pertinent* dichotomy is very intuitive and can be defined as a dichotomy which is "simple" to learn. However, this concept is not rigorous and highly depends on the learning method used to solve the dichotomy. A more formal definition based on some metrics of Ω could be considered, but here again, the choice of the metric should ideally depend on the learning method. With Bayesian classifiers in mind, two points of Ω might be considered as close if their are close according to the Euclidian distance. However, two points, even far from each other according to the Euclidian distance, can be considered as close according to networks based on perceptrons, or to decision trees with multivariate splits [UB90], if they are linearly separable from almost all other points in Ω.

4.2 Elaboration of the decomposition

In the present research a quite simple approach has been chosen, which does not depend on the learning method used for the dichotomies. Matrix $\boldsymbol{D}$ is constructed row by row in a "greedy" way according to the following algorithm.

1. $\boldsymbol{D}$ is a matrix of 0 rows and K columns, and S is a set of pairs of classes, initialized to $\emptyset$.
2. If there exist some pairs of classes (k', k'') not in S and with a minimal distance $d_H(\boldsymbol{D}_{\cdot k'}, \boldsymbol{D}_{\cdot k''})$, select one of them at random, otherwise STOP.
3. Set $S = S \cup \{(k', k'')\}$ and construct a dichotomy h associating class k' with $+1$, class k'' with -1 and ignoring all other classes.

4. Based on the training samples, use any algorithm to learn h and produce an approximation $\hat{h}$.
5. Using the training samples, test $\hat{h}$ for all classes other than k' and k'' and modify h as follows. If the rate of points of a class associated with $+1$ (resp. -1) is larger than τ (*e.g.* $\tau = 0.9$), add this class as a positive (resp. negative class) to the dichotomy h.
6. If the variation between the modified dichotomy h and any other dichotomy associated with a row of $\boldsymbol{D}$ is less than δ, go to 2.
7. Introduce the $\{-1, 0, +1\}$-vector defining the modified dichotomy h as a new row in $\boldsymbol{D}$.
8. If the minimal distance d_H between any two columns of $\boldsymbol{D}$ is at least Δ, STOP, otherwise go to 2.

As such, this algorithm contains 3 parameters τ, δ and Δ which still have to be specified. The parameter $\tau \in [\frac{1}{2}, 1]$ is influencing the rate of 0s in $\boldsymbol{D}$. The second parameter δ in Step 6 is a lower bound on the minimal variation tolerated between two dichotomies. The last parameter Δ in Step 8 specifies the targeted minimal distance d_H between two columns of $\boldsymbol{D}$. The last two parameters are typically small integers. The exact choices made for these three parameters are detailed in Section 5.

If the number of classes is small (≤ 10) it might happen that the above algorithm terminates at the STOP of Step 2 and that Δ is not reached, especially if δ is too big. However, it should be observed that if $\delta = 1$ and if no zeros are allowed in $\boldsymbol{D}$ ($\tau = \frac{1}{2}$), a Δ of 1 can always be reached. In order to ensure that $\Delta = 1$ can be reached for arbitrary τ, the distance d_H used to measure the variation between two rows of $\boldsymbol{D}$ must be replaced by another distance (*e.g.* the distance d_1 induced by the norm L_1) accepting more dichotomies in Step 6. It is worth observing that if d_1 is chosen to compare the rows and if $\delta = \Delta = \tau = 1$, the above algorithm will generate the decomposition matrix of the paiwise coupling scheme.

At this point it should be emphasized that after the class distribution has been found for each dichotomy h in matrix $\boldsymbol{D}$, a new approximation $\hat{h}$ is produced by retraining the associated classifier using the data of all the participating classes.

Clearly, this method does not depend on the algorithm used to learn each dichotomy since the latter is designed according to the method at hand. As a by-product of this key property, the method can handle any combination of learning algorithms of various nature. Indeed, at each iteration of the above algorithm a different learning method can be considered, as long as for any row $D_{l.}$, the learning method used for the computation of $\hat{f}_l$ will be the same as the one involved in the construction of the corresponding dichotomy. The method will even take advantage of such a mixture of learning algorithms since this is likely to increase the diversity of the dichotomies selected.

Moreover, if two different learning methods involved in this construction are such that they will very likely produce two quite different approximations $\hat{f}$ of an arbitrary function f, it is harmless to have two similar dichotomies in the final decomposition as long as they are associated with each of these two methods. The test in Step 6 can then be modified so that the variation between a pair of dichotomies based on these two methods will be considered in a less discriminative way. The obvious benefit of such a variation of the method is that it gives access to decompositions involving a larger amount of pertinent dichotomies, thus allowing larger Δs.

4.3 Reconstruction

In the present study, the learning algorithm used to approximate the dichotomies were yielded $\hat{f}_l$ of the form $\mathrm{sgn} \circ g_l$ and normalized such that $g : \Omega \rightarrow [-1, +1]$. The function $\hat{F}$ was thus based on $\boldsymbol{g}$ instead of $\hat{\boldsymbol{f}}$,

$$\hat{F} \;=\; \arg\max_{k \leq K} \circ m \circ \boldsymbol{g}.$$

As mentioned in Section 2.3, $\boldsymbol{D}$ is a reasonable choice for $\boldsymbol{M}$. However, whenever $\boldsymbol{D}$ contains some zeros and if their cardinalities in each column are not balanced, the choice $\boldsymbol{M} = \boldsymbol{D}$ produces a bias towards the classes corresponding to columns with few zeros, since their scalar products with $\boldsymbol{g}(\boldsymbol{x})$ will tend to be larger than the others. Therefore, the following $\boldsymbol{M}$ has been preferred :

$$M_{lk} = \frac{D_{lk}}{\sum_{l'} |D_{l'k}|}.$$

5 Numerical experiments

The main purpose of this bench of experiments was to demonstrate that the decomposition resulting from the ECOC approach can be replaced by other decompositions into the same number of dichotomies, but with easier ones (approximated by simpler models), without a significant degradation of the overall generalization. Since the aim was not to push the generalization performance as far as possible, only decompositions involving a reasonable number of dichotomies were studied.

The experiments were carried out on some standard polychotomic databases available at the Irvine repository of machine learning databases [MA94], namely `glass`, `letter`, `audiology` and `soybean`.

Database	#examples (P)	#classes (K)
`glass`	214	6
`letter`	20000	26
`audiology`	226	24
`soybean`	683	19

Table 1: Database characteristics.

All the experiments were based on release 8 of C4.5 [Qui86] and the generalization results obtained are also compared with the ones given by C4.5 applied directly to the polychotomy and with the results of the ECOC method. This same method was used for all experiments reported in [DB95], moreover, the number of nodes of a decision tree provides a relevant measure of the complexity of the function computed by the tree. Each time the C4.5 classifier was used, either for polychotomy or for dichotomy solving, only the pruned trees were considered. In order to make comparisons with the algorithm proposed in [DB95] as fair as possible, not only the numbers of dichotomies were always identical, unless specified differently, but also τ was set to $\frac{1}{2}$ so that $\boldsymbol{D}$ has no zeros.

The tests involved 25 replications. In each replication, the random 3-folding technique was applied to the total number of examples available in the database, with class distribution being respected in each of the 3 folds. Generalization results are presented on Table 2. The *3-fold cross-validated paired t test* described in [Die96] was used to check for significant differences of means. The symbols ($\star$) in the table mean that the two concerned error rates are equal with probability ≥ 0.95. All other differences are significant.

	Pertinent dichotomies		ECOC		C4.5 multiclass	
Database	mean	std	mean	std	mean	std
`glass`	31.3	4.9	29.2	4.4	33.6	2.9
`letter`	11.4	0.7	9.7	0.3	13.6	0.2
`audiology`	$\star$ 24.2	4.8	$\star$ 24.4	5.2	22.0	1.7
`soybean`	7.8	1.8	8.2	1.7	10.5	0.8

Table 2: Average percentage and standard deviation of the misclassification on the test set.

The reduction in the total decision tree sizes is illustrated in Table 3.
Table 4 shows the average number of dichotomies generated over the 75 experiments as well as the averages of the minimal dichotomy variations and class differences.

Database	Pertinent dichotomies	ECOC	C4.5 multiclass
`glass`	126.5	508.1	36.9
`letter`	12689.2	20838.4	1937.1
`audiology`	116.8	558.3	40.9
`soybean`	432.5	961.5	112.7

Table 3: Average total decision tree sizes.

	Pertinent dichotomies			ECOC		
Database	L	δ	Δ	L	δ	Δ
`glass`	12.0	1.4	1.0	31.0	1.0	16.0
`letter`	19.9	1.0	1.6	20.0	6.0	5.0
`audiology`	27.5	1.0	0.5	28.0	5.0	8.0
`soybean`	18.2	1.0	1.3	19.0	4.0	5.0

Table 4: Characteristics of the decompositions.

The `glass` database contains 7 classes, but one of them is empty and was thus not considered. The number of dichotomies in the ECOC method is a user-defined parameter in case the number of classes is ≥ 8. For a small number of classes the algorithm automatically sets $L = 2^{K-1} - 1$, as can be seen in Table 4 for the case of `glass`. Note that for this database, the Euclidian distance was used instead of d_H, hence the larger δ.

The `audiology` database is unbalanced, since some of its classes contain a single example, which causes the training set to occasionally have 0 examples of certain of those classes. This explains the Δ values below 1 in Table 4. Indeed, if two classes have no representatives in the training set, the decomposition matrix $\boldsymbol{D}$ will have two columns of 1s, which cause $\Delta_{\boldsymbol{D}} = 0$.

It is clear from the results that the pertinent code algorithm produces low values for Δ and δ. The explanation for this is as follows: since the generated codes are pertinent, and assuming that the same classification method is used for every dichotomy, each time a new code is obtained in Step 7 it will be based on the same distribution as the previously produced codes, and will thus carry information that has partially been presented before. This constrains both the class and code separability, as well as the amount of generated codes.

6 Conclusion and further research

This paper is twofold. On one part, it analyzes the general form of a function which is computed by the most common learning methods for polychotomies, such as decision trees, neural networks, one-per-class or paiwise coupling decompositions, etc. Within this form, the concepts of decomposition and reconstruction are highlighted and clearly dissociated from each other. Several ideas for improving the classical decomposition methods arise from this analysis, such as the exploitation of the information available in the training set for the determination of appropriate decomposition and reconstruction.

In a second part, one of these possible improvements is explored and an algorithm is proposed,

inspired by the best known decomposition method which involves error-correcting codes. The main idea is to use the training samples in order to design a scheme of decomposition of the polychotomy into dichotomies better suiting the data. This method does not depend on the learning algorithm used to solve the dichotomies and interestingly, it can easily handle —and even take advantage of— the combination of various algorithms of different nature. This characteristic is not the least one, considering the amount of research devoted lately to the conception of hybrid systems. Some numerical experiments illustrate essentially that when a polychotomy is decomposed into dichotomies, the complexity of the models built to learn each of these dichotomies can be drastically reduced when this method is used instead of a decomposition based on blind error-correcting codes, and this is achieved without deteriorating the overall generalization.

This research is currently expended mainly along three lines. First, the experiments reported here aimed at demonstrating that the models built by the error-correcting method can be simplified without deteriorating the generalization. Experiments whose goal is to push the generalization as far as possible are now being performed. For that, decompositions involving much more dichotomies are produced, and in a near future, the combination of different learning methods will be explored. In a second line of research, the greedy approach for the construction of $\boldsymbol{D}$ is replaced by a method in which a large number of pertinent dichotomies is first generated and then, an optimization problem is solved in order to extract a subset of dichotomies maximizing Δ and δ. A third line of research consists in exploring other reconstruction methods. In particular, some linear methods whose matrix $\boldsymbol{M}$ is elaborated on the basis of the training set look promising, although such approaches have to struggle against overfitting.

References

[BHI+96] E. Boros, P. L. Hammer, Toshihide Ibaraki, A. Kogan, E. Mayoraz, and I. Muchnik. An implementation of logical analysis of data. RRR 22-96, RUTCOR–Rutgers University's Center For Operations Research, `http://rutcor.rutgers.edu:80/~rrr/`, Submitted, July 1996.

[BOS84] L. Breiman, J. Olshen, and C. Stone. *Classification and Regression Trees.* Wadsworth International Group, 1984.

[CHI88] Yves Crama, Peter L. Hammer, and Toshihide Ibaraki. Cause-effect relationships and partially defined boolean functions. *Annals of Operations Research*, 16:299–326, 1988.

[DB91] 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.

[DB95] Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

[DH73] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, New York, 1973.

[Die96] Thomas G. Dietterich. Statistical tests for comparing supervised classification learning algorithms. OR 97331, Department of Computer Science, Oregon State University,, 1996.

[KD95] E. B. Kong and T. G. Dietterich. Error-correcting output coding corrects bias and variance. In *The XII International Conference on Machine Learning*, pages 313–321, San Francisco, CA, 1995. Morgan Kaufmann.

[MA94] P. M. Murphy and D. W. Aha. UCI repository of machine learning databases. Machine-readable data repository, Irvine, CA: University of California, Department of Information and Computer Science, 1994.

[Qui86] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[Ros58] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 63:386–408, 1958.

[UB90] P. E. Utgoff and C. E. Brodley. An incremental method for finding multivariate splits for decision trees. In *Proceedings of the Seventh International Conference on Machine Learning*, pages 58–65, Los Altos, CA, 1990. Morgan Kaufmann.

[Wer74] P. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University, 1974.