Decoding of Ternary Error Correcting Output Codes

Sergio Escalera[1], Oriol Pujol[2] and Petia Radeva[1]

[1] Computer Vision Center, Dept. Computer Science, UAB, 08193 Bellaterra, Spain
[2] Dept. Matemàtica Aplicada i Anàlisi, UB, Gran Via 585, 08007, Barcelona, Spain

Abstract. Error correcting output codes (ECOC) represent a successful extension of binary classifiers to address the multiclass problem. Lately, the ECOC framework was extended from the binary to the ternary case to allow classes to be ignored by a certain classifier, allowing in this way to increase the number of possible dichotomies to be selected. Nevertheless, the effect of the zero symbol by which dichotomies exclude certain classes from consideration has not been previously enough considered in the definition of the decoding strategies. In this paper, we show that by a special treatment procedure of zeros, and adjusting the weights at the rest of coded positions, the accuracy of the system can be increased. Besides, we extend the main state-of-art decoding strategies from the binary to the ternary case, and we propose two novel approaches: Laplacian and Pessimistic Beta Density Probability approaches. Tests on UCI database repository (with different sparse matrices containing different percentages of zero symbol) show that the ternary decoding techniques proposed outperform the standard decoding strategies.

1 Introduction

Machine learning studies automatic techniques for learning to make accurate predictions based on past observations. There are plenty of classification techniques reported in literature: Support Vector Machines [1][2], decision trees [3], nearest neighbors rules, etc. It is known that for some classification problems, the lowest error rate is not always reliably achieved by trying to design a single classifier. An alternative approach is to use a set of relatively simple sub-optimal classifiers and to determine a combination strategy that pools together the results. Different types of systems of multiple classifiers have been proposed in the literature, most of them use similar constituent classifiers, which are often called base classifiers (dichotomies from now on). Adaboost [4], for example, uses weak classifiers as predictions that showed to be slightly better than random guessing and combines them in an ensemble classifier.

Although binary classification is a well-studied problem, building a highly accurate multiclass prediction rule is certainly a difficult task. In those situations, the usual way to proceed is to reduce the complexity of the problem by dividing it into a set of multiple simpler binary classification subproblems. One-versus-one pairwise [5] or one-versus-all techniques are some of the most frequently used

schemes. In the line of the aforementioned techniques Error Correcting Output Codes [6] were born. ECOC is a general framework based on coding and decoding (ensemble strategy) techniques to handle multiclass problems. One of the most well-known properties of the ECOC is that it improves the generalization performance of the base classifiers [7][5].

In this technique the multiclass to binary division is handled by a coding matrix. Each row of the coding matrix represents a codeword assigned to each class. On the other hand, each column of the matrix (each bit of the codeword) defines a partition of the classes in two sets. The ECOC strategy is divided in two parts: the coding part, where the binary problems to be solved have to be designed, and the decoding technique, that given a test sample, looks for the most similar codewords. For the coding strategies, the three most well-known strategies are one-versus-all, all-pairs (one-versus-one) and random coding.

The decoding step was originally based on error-correcting principles under the assumption that the learning task can be modelled as a communication problem, in which class information is transmitted over a channel [8]. The decoding strategy corresponds to the problem of distance estimation between the test codeword and the codewords of the classes. Concerning the decoding strategies, two of the most standard techniques are the Euclidean distance and the Hamming decoding distance. If the minimum Hamming distance between any pair of class codewords is d, then any $[(d-1)/2]$ errors in the individual dichotomies result can be corrected, since the nearest codeword will be the correct one. The original two-symbol coding matrix M was extended to the ternary case $M \in \{-1, 0, 1\}^{N_c \times n}$ by Allwein et. al [5]. The new zero symbol indicates that a particular class is not considered by a given dichotomy. This fact allows to obtain a higher number of possible dichotomies that create different decision boundaries, allowing more accurate results for multiclass classification problems. Nevertheless, the effect of increasing the sparseness of the coding matrix has not been previously analyzed enough.

The goal of this article is twofold: firstly, we extend the standard state-of-art decoding strategies to the ternary case. We analyze the effect of the zero symbol in the ECOC matrix M. We show how this symbol affects to the decoding strategy, and we take into account the two main properties than define the problem: the zero symbol may not introduce decoding errors, and the coded positions have different relevance depending on the number of zeros contained on each coding matrix M row. We compare the evolution results for standard decoding strategies as Hamming (HD), inverse Hamming (IHD) or Euclidean distance (ED) when the number of zeros is increased. Secondly, we extend the state-of-art coding strategies to the ternary case: Attenuated Euclidean distance (AED), and Loss-based decoding (LB). In this context, we propose two new decoding techniques to solve the exposed problem: Laplacian decoding (LAP), and Beta Density Distribution Pessimistic score (β-DEN).

The paper is organized as follows: section 2 explains the ECOC framework, section 3 reviews the state-of-art decoding strategies, shows the ternary adapta-

tion and the new decoding approaches. Section 4 contains the experiments and results, and section 5 concludes the paper.

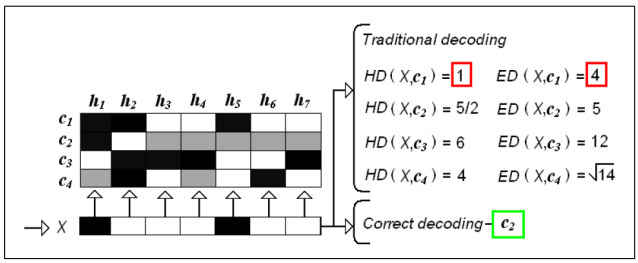

Fig. 1. Example of ternary matrix M for a 4-class problem. A new test codeword is missclassified due to the confussion of using the traditional decoding strategies.

2 ECOC

The basis of the ECOC framework is to create a codeword for each of the N_c classes. Arranging the codewords as rows of a matrix, we define a "coding matrix" M, where $M \in \{-1, 0, 1\}^{N_c \times n}$ in the ternary case, being n the code length. From point of view of learning, M is constructed by considering n binary problems (dichotomies), each corresponding to a matrix column. Joining classes in sets, each dichotomy defines a partition of classes (coded by +1, 0 or -1, according to their class set membership). In fig. 1 we show an example of a ternary matrix M. The matrix is coded using 7 dichotomies $h_1, ..., h_7$ for a four multiclass problem (c_1, c_2, c_3, and c_4). The white regions are coded by 1 (considered as positive for its respective dichotomy, h_i), the dark regions by -1 (considered as negative), and the grey regions correspond to the zero symbol (not considered classes for the current dichotomy). For example, the first classifier is trained to discriminate c_3 versus c_1 and c_2, the second one classifies c_2 versus c_1, c_3 and c_4, and so on. Applying the n trained binary classifiers, a code is obtained for each data point in the test set. This code is compared to the base codewords of each class defined in the matrix M, and the data point is assigned to the class with the "closest" codeword.

To design an ECOC system, we apply a coding and a decoding strategy. The most well-known decoding strategies are Hamming and Euclidean distance. The Hamming distance is estimated by $d(x, y^i) = \sum_{j=1}^{n} \mid (x_j - y_j^i) \mid /2$, where $d(x, y^i)$ is the distance of the codeword x to the class i, n is the number of dichotomies (and thus, the components of the codeword), and x and y are the values of the input vector codeword and the base class codeword, respectively. For the Euclidean distance, the measure is based on minimizing the distance $d(x, y^i) = \sqrt{\sum_{j=1}^{n}(x_j - y_j^i)^2}$. To classify a new input $x = [-1, 1, 1, 1, -1, 1, 1]$ in fig. 1, the traditional Hamming or Euclidean distances are applied, obtaining in both cases the minimum distance corresponding to class one. Note that the correct decoding corresponds to c_2 since both first dichotomies trained on c_2 classify the new example correctly.

Most of the discrete coding strategies up to now are based on predesigned problem-independent codewords. When the ECOC technique was first developed it was designed to have certain properties to enable them to generalize well. A good error-correcting output code for a k-class problem should satisfy that rows, columns (and their complementaries) are well-separated from the rest in terms of Hamming distance. These strategies are one-versus-all, dense and sparse random techniques [5], and one- versus-one [9]. Crammer et. al [10] were the first authors reporting improvement in the design of the ECOC problem-dependent codes. However, the results were rather pessimistic since they proved that the problem of finding the optimal discrete codes is computationally unfeasible since it is NP-complete [10]. Specifically, they proposed a method to heuristically find the optimal coding matrix by changing its representation from discrete to continuous values. Recently, new improvements in the problem-dependent coding techniques have been presented by Pujol et. al. [11]. They propose embedding of discriminant tree structures in the ECOC framework showing high accuracy with a very small number of binary classifiers. Escalera et. al [12][13] propose a multiple tree structures embedding to form a Forest-ECOC and design of a problem-dependent ECOC-ONE coding strategy. The procedure is based on generating a code matrix by searching for the dichotomies that best split the difficult classes in the training procedure guided by a validation subset.

Many decoding strategies have been proposed in the ECOC framework. Nevertheless, very few attention has been given to the ternary case. Often techniques add errors due to the zeros, while other approaches do not consider the effect of this symbol for the decoding strategy. In the next chapter, we address the ternary case of the decoding strategies in depth.

3 Ternary ECOC Decoding

The zero symbol allows to ignore some classes for a certain dichotomy. Although the binary matrix M is extended with the zero symbol, the decoding strategies are not adapted to the influence of that symbol. The use of standard decoding techniques that do not consider the effect of this symbol frequently fail (as shown in fig. 1). To understand the extension to the ternary case, first we define the reasons why the zero symbol needs special attention. As shown in fig. 1, the error accumulated by the zero symbol has to be non-significative in comparison with the failures at coded positions. Another important aspect is that if a codeword of length n has k zeros, the rest of the positions $(n - k)$ not containing zeros must have more importance either in case of coincidence or failure. For example, if we consider two codewords y_1 and y_2, we can not consider the same error for the codeword y_1 if it has one fail and two coded positions than if there are ten coded positions in y_2. Therefore, the large difference in the number of coded positions between codewords is an important issue that must be taken into account. Allwein et. al [5] studied numerically the effect of the symbol zero and they proposed the Loss-based decoding technique in order to take this symbol into account.

3.1 Traditional decoding strategies

Analyzing the Hamming distance in the ternary case, we can observe that it introduces a high error for the zero values (ignored classes by certain dichotomies) and all positions obtain the same importance at the decoding step. Euclidean distance accumulate half of the error estimated by Hamming distance. Equally, it still assigns a considerable error to the symbol zero and does not increase the relevance of the rest of the coded codeword positions. Another traditional strategy for decoding is the Inverse Hamming distance.

Inverse Hamming distance Let $D(x) = [d(x, y^1), d(x, y^2), ..., d(x, y^{N_c})]$ be define as the set of estimated distances from a test codeword to the N_c classes codewords. Let us define Δ as the matrix composed by the Hamming distances between the codewords of M. Each position of Δ is defined by $\Delta(i, j) = d(y^i, y^j)$, where $d(y^i, y^j)$ defines the Hamming distance between codeword i and j. If the set D is evaluated using the Hamming distance, Δ can be inverted to find the vector $Q = [q_1, q_2, ..., q_{N_c}]$ containing the N_c individual class probabilities by means of $Q = \Delta^{-1} D^T$. This approach is based on the Hamming minimization theory, hence its properties are the same for the ternary case.

3.2 Extended decoding strategies

The following techniques are adaptations of some traditional decoding strategies to the ternary case.

Attenuated Euclidean decoding This technique is an adaptation of the Euclidean distance to take into account the symbol zero. To solve the previously commented problem of the Euclidean distance, we redefine the decoding as $d(x, y^i) = \sqrt{\sum_{j=1}^{n} \mid y_j^i \mid (x_j - y_j^i)^2}$, where the factor $\mid y_j^i \mid$ rejects the errors accumulated by the zero symbol at codeword of class i. Using this technique, we consider that the relevant information is only represented by the coded positions, though the rest of coded positions still obtains the same relevance in the decoding process. Extending this discrete idea of the importance of zeros to the probabilistic case, we find the Loss-based decoding strategy.

Loss-based decoding The loss-based decoding method [5] requires that the output of the binary classifier is a margin score satisfying two requirements. First, the score should be positive if the example is classified as positive, and negative if the example is classified as negative. Second, the magnitude of the score should be a measure of confidence in the prediction.

Let $f(\ell, j)$ be the margin score for example ℓ predicted by the classifier corresponding to column j of the code matrix M. For each row i of M and for each example ℓ, we compute the distance between $f(\ell, j)$ and $y^i = M(i, j)$ $\forall j \in \{1, ..., n\}$,

$$d^i(\ell, i) = \sum_{j=1}^{n} L(M(i, j) \cdot f(\ell, j)) \tag{1}$$

where L is a loss function that depends on the nature of the binary classifier. The two most common loss functions are $L(\hbar) = -\hbar$ and $L(\hbar) = e^{-\hbar}$, where $\hbar = M(i,j) \cdot f(\ell, j)$. We label each example x with the label that minimizes d_L. Note that this technique attenuates the error for the zero symbol while maintaining the weight for all the coded positions independently of the number of zeros from each codeword. This technique attenuates the errors introduced by zeros in the same way that the discrete Attenuated Euclidean distance strategy extending the measure estimation to an additive probabilistic model.

3.3 Novel decoding strategies

The previous methods attenuate the errors from the zero symbol in a discrete and probabilistic way. The following novel approaches are based on considering the distance and probability conditions to decode the coding matrices depending on their structure, adding new conditions on coded positions to adjust the analysis of the ternary case.

Laplacian strategy We propose a Laplacian decoding strategy to give to each class a score according to the number of coincidences between the input codeword and the class codeword, normalized by the errors without considering the zero symbol. In this way, the coded positions of the codewords with more zero symbols attain more importance. The decoding score is estimated by:

$$d(x, y^i) = \frac{C_i + 1}{C_i + E_i + K} \tag{2}$$

where C_i is the number of coincidences from the test codeword and the codeword for class i, E_i is the number of failures from the test codeword and the codeword for class i, and K is an integer value that codifies the number of classes considered by the classifier, in this case 2, due to the binary partitions of the base classiers. The offset $1/K$ is the default value (bias) in case that the coincidences and failures tend to zero. Note that when the number of C and E are sufficiently high, the factor $1/K$ does not contribute:

$$\lim_{C \to 0, E \to 0} d(x, y^i) = \frac{1}{K} \qquad \lim_{C \to \infty, E \to \infty} d(x, y^i) = \frac{C}{C + E} \tag{3}$$

Beta Density Distribution Pessimistic Strategy The method is based on estimating the probability density functions between two codewords, extending the Laplacian ternary properties from the discrete to the probabilistic case. The main issue of this strategy is to model at the same time the accuracy and uncertainty based on a pessimistic score to obtain more reliable predictions. We use an extension of the continuous binomial distribution, the Beta distribution defined as:

$$\psi(z, \alpha, \beta) = \frac{1}{K} z^{\alpha} (1 - z)^{\beta} \tag{4}$$

where ψ_i is the Beta Density Distribution between a codeword x and a class codeword y^i for class i, α and β are the number of coincidences and failures respectively, and $z \in [0, 1]$. The expectation $E(\psi_i)$ and the variance $var(\psi_i)$of the distribution are:

$$E(\psi_i) = \frac{\alpha}{\alpha + \beta} \qquad var(\psi_i) = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)} \tag{5}$$

where the expectation tends to the Laplacian estimation when $C \to \infty, E \to \infty$ in (2).

Let Z_i be the value defined as $Z_i = argmax_z(\psi_i(z))$. To classify an input codeword x given the set of functions $\psi(z) = [\psi_1(z), \psi_2(z), ..., \psi_{N_c}(z)]$, we select the class i with the highest score $(Z_i - a_i)$, where a_i is defined as the pessimistic score satisfying the following equivalency:

$$a_i \ : \quad \int_{Z_i - a_i}^{Z_i} \psi_i(z) = \frac{1}{3} \tag{6}$$

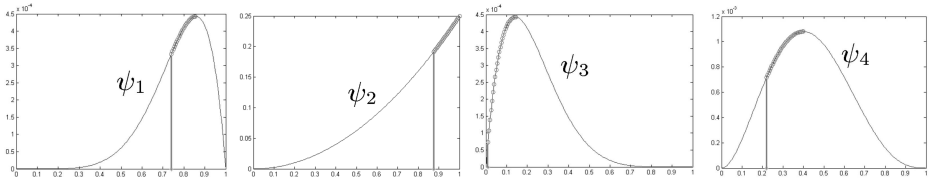

Fig. 2. Pessimistic Density Probability estimations for the test codeword x and the matrix M for the four classes of fig. 1. The probability for the second class allows a successful classification in this case.

In fig. 2 the density functions $[\psi_1, \psi_2, \psi_3, \psi_4]$ of fig. 1 for the input test codeword x are shown. Fig. 2(b) corresponds to the correct class c_2, well-classified by the method with the highest pessimistic score. One can observe that the Beta Density Probability decreases faster in c_1 compared to c_2 due to the failure of one code position for the codeword of class 1 compared to the pessimistic score of the second codeword with five zeros and two code coincidences.

It can be shown that when a function ψ_i is estimated by a combination of sets α and β of z and (1-z) respectively, the sharpness is higher than when it is generated by a majority of one of the two types. Besides, this sharpness depends on the number of code positions different to zero and the balance between the number of coincidences and failures.

4 Results

To test the different decoding strategies, we used the UCI repository databases. The characteristics of the 5 used databases are shown in table 1. As our main

goal is to analyze the effect of the ternary matrix M, we have generated a set of matrices with different percentages of zeros. Once generated the coding matrices, the dichotomies are trained. The generated set of experiments is composed by 6 sets of matrices for each database, each one containing 10 different random sparse matrices of different percentage of zeros. We increase the number of zeros by 10% starting from the previously generated matrices to obtain more realistic analysis. Besides, each matrix from this set is evaluated with a ten-fold cross-validation. The decoding strategies used in the comparative are: Hamming distance (HD), Euclidean distance (ED), Inverse Hamming distance (IHD), Attenuated Euclidean Distance (AED), Loss-based decoding with exponential loss-function (ELB), Loss-based decoding with linear loss-function (LLB), Laplacian decoding (LAP), and Beta Pessimistic Density Probability (β-DEN).

Table 1. UCI repository databases characteristics.

Problem	#Train	#Test	#Attributes	#Classes
Dermathology	366	-	34	6
Ecoli	336	-	8	8
Glass	214	-	9	7
Vowel	990	-	10	11
Yeast	1484	-	8	10

The tests for the five databases are shown graphically in fig. 3(a)-(e). The graphics show the error evolution for all the decoding strategies at each database. In table 2 and fig. 3(f) the ranking of each method at each percentage step of zeros is shown. The ranking values of the table correspond to the average performance position for each method for all runs on all databases. One can observe that some methods obtain reasonable well-positions at the ranking in all percentages of sparseness, as our proposed Laplacian and Beta Pessimistic Density Probability decoding. Euclidean distance also can contribute to reduce the error of zeros better than techniques as loss-based function, although the last one shows the best accuracy with dense matrices (0% of zeros). However, its performance is reduced as the number of zeros increases. Observing the global rank of table 2, the first position is for Beta Pessimistic Density Probability followed by Laplacian decoding.

Table 2. Mean ranking evolution for the methods on the UCI databases tests when the number of zeros is increased.

Strategy	0% zeros	10% zeros	20% zeros	30% zeros	40% zeros	50% zeros	Global rank
HD	3.2	3.2	4.4	4.2	4.6	4.0	3.9
ED	3.2	3.2	2.4	2.2	2.6	3.2	2.8
AED	3.2	3.6	4.6	3.8	2.4	4.0	3.6
IHD	3.4	4.0	5.8	4.0	6.0	5.2	4.7
LLB	**1.6**	6.8	7.0	6.8	6.6	7.2	6.0
ELB	**1.6**	4.2	6.8	5.2	5.8	5.6	4.9
LAP	2.4	**2.2**	2.2	2.0	**1.8**	1.6	**2.0**
β-DEN	2.4	2.4	**1.8**	**1.0**	2.4	**1.2**	**1.9**

Table 2 shows that Loss based decoding is the best option for the dense matrix case, and Beta Pessimistic Density Probability and Laplacian decoding

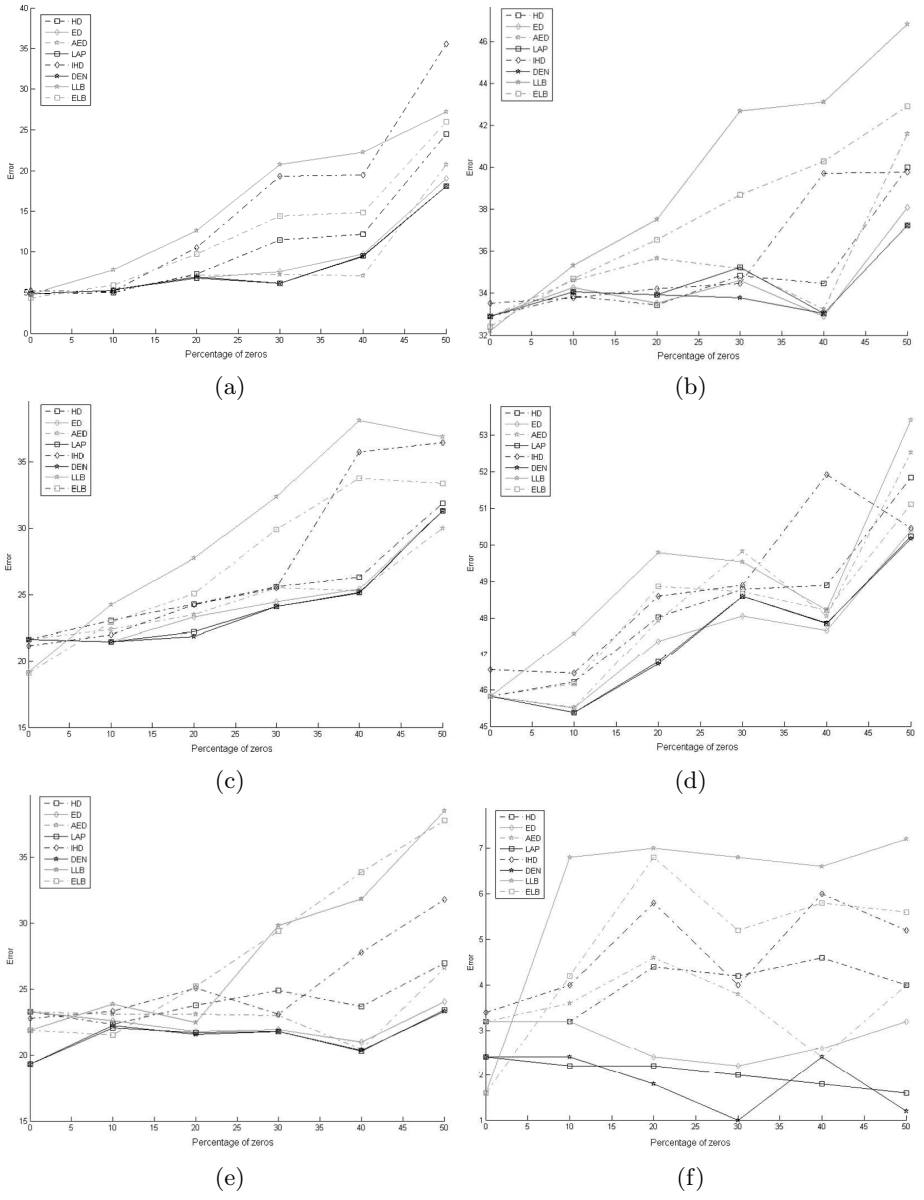

Fig. 3. Error evolution for decoding strategies on Dermathology (a), Glass (b), Ecoli (c), Yeast (d), and Vowel (e) UCI databases. (f) Mean ranking evolution for the methods on the UCI databases tests. The x-axis correspond to the percentage of ceros (increased 10% by step) of 10 sparse matrices M.

are the best choices when we have an increase of the sparseness degree. If we do not have information about the composition of the code matrix M, we can use the general rank of table 2, being the Beta Pessimistic Density Probability and Laplacian strategies the more suitable for each case.

5 Conclusions

The ternary ECOC when applying a decoding strategy has not been previously enough analyzed. In this paper, we show the effect on reliability reduction when the number of zeros (non considered class by a given dichotomy) is increased. We analyzed the state-of-art ECOC decoding strategies, adapting them to the ternary case, taking into account the effect of the ternary symbol and the weights of the code positions depending on the number of containing zeros. We propose two new decoding strategies that outperform the traditional decoding strategies when the percentage of zeros is increased. The validation of the decoding strategies at UCI repository databases gives an idea about the techniques that are more useful depending of the sparseness of the ECOC matrix M, where our proposed Pessimistic Density Probability and Laplacian strategies obtain the best ranking in the general case. We are planning to extend the proposed decoding strategies to the continuous case.

6 Acknowledgements

This work was supported in part by the projects TIC2003-00654, FIS-G03/1085, FIS-PI031488, and MI-1509/2005.

References

1. V. Vapnik, Estimation of dependences based on empirical data, Springer, 1982.
2. V. Vapnik, The nature of statistical learning theory, Springer, 1995.
3. L. Breiman, J. Friedman, Classification and Regression Trees, Wadsworth, 1984.
4. J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting, (38), 1998, pp. 337-374.
5. E. Allwein, R. Schapire, Y. Singer, Reducing multiclass to binary: A unifying approach for margin classifiers, (1), 2002, pp. 113-141.
6. T. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, (2), 1995, pp. 263-286.
7. T. Windeatt, R. Ghaderi, Coding and decoding for multi-class learning problems,(4), 2003, pp. 11-21.
8. T. Dietterich, G. Bakiri, Error-correcting output codes: A general method for improving multiclass inductive learning programs, in: A. Press (Ed.), Ninth National Conference on Artificial Intelligence, 1991, pp. 572-577.
9. T.Hastie, R.Tibshirani, Classification by pairwise grouping, (26), 1998, pp. 451-471.
10. K. Crammer, Y. Singer, On the learnability and design of output codes for multiclass problems, (47), 2002, pp. 201-233.
11. O. Pujol, P. Radeva, J. Vitrià, Discriminant ECOC: A heuristic method for application dependent design of error correcting output codes, (28), 2006, pp. 1001-1007.
12. S. Escalera, O. Pujol, P. Radeva, ECOC-ONE: A novel coding and decoding strategy, ICPR, Hong Kong, China, 2006, (in press).
13. S. Escalera, O. Pujol, P. Radeva, Forest extension of error correcting output codes and boosted landmarks, ICPR, Hong Kong, China, 2006, (in press).