Weighted Strategy for Error-Correcting Output Codes

Sergio Escalera, Oriol Pujol, and Petia Radeva
Computer Vision Center, Universitat Autònoma de Barcelona, Campus UAB, Edifici O, 08193, Bellaterra, Spain.
Dept. Matemàtica Aplicada i Anàlisi, Universitat de Barcelona, Gran Via 585, 08007, Barcelona, Spain.
{*sergio,oriol,petia*}*@maia.ub.es*

Abstract

Error Correcting Output Codes technique (ECOC) represents a general framework capable to extend any binary classification process to the multi-class case. In this work, we present a novel decoding strategy that takes advantage of the ECOC coding to outperform the up to now existing decoding strategies. The results show that the presented methodology considerably increases the performance of the state-of-the-art ECOC designs.
Keywords: Ensemble Methods and Boosting, Learning, Classification.

1 Introduction

Multi-class categorization in a Machine Learning is based on assigning labels to instances that belong to a finite set of object classes N $(N > 2)$. Nevertheless, designing a multi-classification technique is a difficult task. In this sense, it is common to conceive algorithms that distinguish between two classes and combine them following a special criterion. Pairwise (one-versus-one) voting scheme [6] or one-versus-all [8] grouping strategy are the procedures most frequently used. Error Correcting Output Codes were born as a framework for handling multi-class problems using binary classifiers [3]. ECOC has shown to dramatically improve the classification accuracy of supervised learning algorithms in the multi-class case by reducing the variance of the learning algorithm and correcting errors caused by the bias of the learners [4]. Furthermore, ECOC has been successfully applied to a wide range of applications, such as face recognition, text recognition or manuscript digit classification.

The ECOC framework consists of two steps: a coding step, where a codeword[1] is assigned to each class, and a decoding technique, where given a test sample the method looks for the most similar class codeword. One of the first designed binary coding strategies is the one-versus-all approach, where each class is discriminated against the rest. However, it was not until Allwein et al. [1] introduced a third symbol (the zero symbol) in the coding process that the coding step received special attention. The ternary ECOC gives more expressivity to the ternary ECOC framework by allowing some classes to be ignored by the binary classifiers. Thanks to this, strategies such as one-versus-one [6] and random sparse coding [1] are possible. However, these predefined codes are independent of the problem domain, and recently, new approaches involving heuristics for the design of problem-dependent output codes have been proposed [10][5] with successful results.

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 [3]. In this sense, the Hamming and the Euclidean distances were the first tentative for decoding [3]. Still very few alternative de-

[1]The codeword is a sequence of bits (called code) representing each class, where each bit identifies the class membership by a given binary classifier.

coding strategies have been proposed in the literature. In [11], Inverse Hamming Distance (IHD) and Centroid distance (CEN) for binary problems are introduced. Other decoding strategies for nominal, discrete and heterogeneous attributes have been proposed in [7]. With the introduction of the zero symbol, Allwein et al. [1] show the advantage of using a loss based function of the margin of the base classifier on the ternary ECOC. Also, there have been several attempts to introduce probabilities in the ECOC decoding process [9][2]. In [9], the authors use conditional probabilities to estimate the class membership in a kernel machine approach. An alternative probabilistic design of the coding and decoding strategies is proposed in [2]. Nevertheless, none of the few proposed decoding strategies in the literature takes into account the effect of the third (0) symbol during the decoding step, leaving this fact as an open issue worthy of exploring.

In this paper, we present a novel decoding technique, that we call Loss-Weighted decoding strategy (LW). LW is based on a combination of probabilities that adjusts the importance of each coded position in a ternary ECOC matrix given the performance of a classifier. The formulation of our decoding process allows the use of discrete output of the classifier as well as the margin when it is available. The traditional Euclidean distance, the Loss-based decoding, the probabilistic model presented in [9], and the proposed LW decoding are compared with 5 state-of-the-art coding strategies, showing the high performance of the presented strategy in public databases.

2 Error Correcting Output Codes

Given a set of N_c classes to be learned, n different bi-partitions (groups of classes) are formed, and n binary problems (dichotomies) are trained. As a result, a codeword of length n is obtained for each class, where each bin of the code corresponds to a response of a given dichotomy. 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. 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 class 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 by the current dichotomy).

During the decoding process, applying the n trained binary classifiers, a code x is obtained for each data point in the test set. This code is compared to the base codewords of each class $\{y_1, ..., y_4\}$ defined in the matrix M, and the data point is assigned to the class with the "closest" codeword [1][11]. Although different distances can be applied, the most frequently used are the Hamming (*HD*) and the Euclidean distances (*ED*). In fig.1, a new test input x is evaluated by all the classifiers and the method assigns label c_1 with the closest decoding distances.

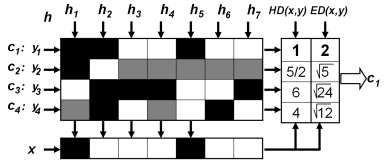


Figure 1: Example of ternary matrix M for a 4-class problem. A new test codeword is classified by class c_1 when using the HD and ED decoding strategies.

2.1 Coding designs

Different coding strategies were proposed for the design of the ECOC matrix M. One-versus-all [8] was the first ECOC design, where each learner is trained to distinguish one class from the rest of classes. The one-versus-one [6] strategy considers all pairs of classes. The Dense and Sparse random strategies were proposed for the binary and ternary ECOC framework, respectively [10]. These strategies must assure that the randomly generated matrix rows and columns are as different as possible in terms of the Hamming distance. The Dense

random strategy generates a random coding matrix M, where the values $\{+1, -1\}$ have a certain probability to appear. The sparse random strategy is similar to the dense case, but includes the third symbol 0 with another probability of appearance.

Due to the great number of bits involved in the traditional coding strategies and the low robustness of the one-versus-all strategy in comparison with one-versus-one, new coding approaches have been proposed [5][10]. The techniques take into account the knowledge of the problem domain by selecting the representative binary classifiers that increase the generalization performance while keeping the code length small. The DECOC method proposed in [10] is based on the embedding of discriminant tree structures derived from the problem domain. In the work of [5], the authors propose the ECOC-ONE coding strategy as an extension of any initial ECOC configuration. The method uses a coding process that learns relevant binary problems guided by a validation subset.

2.2 Decoding designs

The decoding step decides the final category of an input test by comparing the codewords. In this way, a robust decoding strategy is required to obtain accurate results. Several techniques for the binary decoding step have been proposed in the literature [11][7][9][2], though the most common ones are the Hamming and the Euclidean approaches [11]. In the work of [10], authors showed that usually the Euclidean distance was more suitable than the traditional Hamming distance in both the binary and the ternary cases. Nevertheless, little attention has been paid to the ternary decoding approaches.

In [1], the authors propose a Loss-based technique when a confidence on the classifier output is available. For each row of M and each data sample $\wp$, the authors compute the similarity between $f^j(\wp)$ and $M(i,j)$, where f^j is the j^{th} dichotomy of the set of hypothesis F, considering a loss estimation on their scalar product, as follows:

$$D(\wp, y_i) = \sum_{j=1}^{n} L(M(i,j) \cdot f^j(\wp)) \quad (1)$$

where L is a loss function that depends on the nature of the binary classifier. The most common loss functions are the linear and the exponential one. The final decision is achieved by assigning a label to example $\wp$ according to the class c_i with the minimal distance.

Recently, the authors of [9] proposed a probabilistic decoding strategy based on the margin of the output of the classifier to deal with the ternary decoding. The decoding measure is given by:

$$D(y_i, F) = -log\left(\prod_{j\in[1,\ldots,n]:M(i,j)\neq 0} P(x^j = M(i,j)|f^j) + \alpha\right) \quad (2)$$

where α is a constant factor that collects the probability mass dispersed on the invalid codes, and the probability $P(x^j = M(i,j)|f^j)$ is estimated by means of:

$$P(x^j = y_i^j|f^j) = \frac{1}{1 + exp(y_i^j(A^j f^j + B^j))} \quad (3)$$

Vectors A and B are obtained by solving an optimization problem [9].

3 Loss-Weighted decoding (*LW*)

As mentioned above, the 0 symbol allows to increase the number of bi-partitions of classes (thus, the number of possible binary classifiers), resulting in a higher number of binary problems to be learned. However, the effect of the ternary symbol is still an open issue. Since a zero symbol means that the corresponding classifier is not trained on a certain class, to consider the "decision" of this classifier on those zero coded position does not make sense. Moreover, the response of the classifier on a test sample will always be different to 0, so obligatory an error will be registered. Let return to fig. 1, where an example about the effect of the 0 symbol is shown. The classification result using the Hamming distance as well as the Euclidean distance is class c_1. Note that class c_2 has only coded

first both positions, thus it is the only information provided about class c_2. The first two coded locations of the codeword x correspond exactly to these positions. Thus, the correct classification should be class c_2 instead of c_1. The use of standard decoding techniques that do not consider the effect of the third symbol (zero) frequently fails. In the figure, the HD and ED strategies accumulate an error value proportional to the number of zero symbols by row, and finally miss-classify the sample x.

Given a coding matrix M,

1) Calculate the matrix of hypothesis H:

$$H(i,j) = \frac{1}{m_i} \sum_{k=1}^{m_i} \gamma(h_j(\wp_k^i), i, j) \quad (4)$$

based on $\gamma(x_j, i, j) = \begin{cases} 1, & \text{if } x_j = M(i,j) \\ 0, & \text{otherwise.} \end{cases}$ (5)

2) Normalize H so that $\sum_{j=1}^{n} M_W(i,j) = 1, \forall i = 1, ..., N_c$:

$$M_W(i,j) = \frac{H(i,j)}{\sum_{j=1}^{n} H(i,j)},$$
$$\forall i \in [1, ..., N_c], \quad \forall j \in [1, ..., n]$$

Given a test input $\wp$, decode based on:

$$d(\wp, i) = \sum_{j=1}^{n} M_W(i,j) L(M(i,j) \cdot f(\wp, j)) \quad (6)$$

Table 1: Loss-Weighted algorithm.

To solve the commented problems, we propose a Loss-Weighted decoding. The main objective is to find a weighting matrix M_W that weights a loss function to adjust the decisions of the classifiers, either in the binary and in the ternary ECOC. To obtain the weighting matrix M_W, we assign to each position (i, j) of the matrix of hypothesis H a continuous value that corresponds to the accuracy of the dichotomy h_j classifying the samples of class i (4). We make H to have zero probability at those positions corresponding to unconsidered classes (5), since these positions do not have representative information. Next step is to normalize each row of the matrix H so that M_W can be considered as a discrete probability density function (6). This step is very important since we assume that the probability of considering each class for the final classification is the same (independently of number of zero symbols) in the case of not having *a priori* information ($P(c_1) = ... = P(c_{N_c})$). In fig. 2 a weighting matrix M_W for a 3-class problem with four hypothesis is estimated. Figure 2(a) shows the coding matrix M. The matrix H of fig. 2(b) represents the accuracy of the hypothesis classifying the instances of the training set. The normalization of H results in the weighting matrix M_W of fig. 2(c)[2].

$$M = \begin{bmatrix} 1 & 1 & -1 & 0 \\ 1 & -1 & 0 & 0 \\ 1 & 1 & 1 & -1 \end{bmatrix} \quad H = \begin{bmatrix} 0.955 & 0.955 & 1.000 & 0.000 \\ 0.900 & 0.800 & 0.000 & 0.000 \\ 1.000 & 0.905 & 0.805 & 0.805 \end{bmatrix}$$

(a) (b)

$$M_W = \begin{bmatrix} 0.328 & 0.328 & 0.344 & 0.000 \\ 0.529 & 0.471 & 0.000 & 0.000 \\ 0.285 & 0.257 & 0.229 & 0.229 \end{bmatrix}$$

(c)

Figure 2: (a) Coding matrix M of four hypotheses for a 3-class problem. (b) Matrix H of hypothesis accuracy. (c) Weighting matrix M_W.

The Loss-weighted algorithm is shown in table 1. As commented before, the loss functions applied in equation (6) can be the linear or the exponential ones. The linear function is defined by $L(\theta) = \theta$, and the exponential loss function by $L(\theta) = e^{-\theta}$, where in our case θ corresponds to $M(i,j) \cdot f^j(\wp)$. Function $f^j(\wp)$ may return either the binary label or the confidence value of applying the j^{th} ECOC classifier to the sample $\wp$.

4 Results

The methodology of the validation affects the data, applications, strategies of the comparative, and measurements.

a) *Data and applications*: We consider the UCI classification using 13 datasets from the public UCI Machine Learning repository database.

b) *Strategies and measurements*: The strategies used to validate the classification are 40 runs of

[2]Note that the presented Weighting Matrix M_W can also be applied over any decoding strategy.

Discrete Adaboost with decision stumps, and the OSU implementation of SVM with RBF kernel ($\gamma = 1$)[3]. These two classifiers generate the set of binary problems to embed in the ECOC configurations: one-versus-one, one-versus-all, dense-random, DECOC, and ECOC-ONE. Each of the ECOC strategies is evaluated using different decoding strategies: the Euclidean distance, Loss-based decoding with exponential loss function, the probabilistic model of [9], and four variants of the Loss-Weighted decoding strategy: linear LW with output label of the classifier, linear LW with output margin of the classifier, exponential LW with output label of the classifier, and exponential LW with output margin of the classifier. The number of classifiers used for each methodology is the predefined or the provided by the authors in the case of problem-dependent designs, except for the dense random case, where we selected n binary classifiers. The classification tests are performed using stratified ten-fold cross-validation with two-tailed t-test at 95% for the confidence interval.

4.1 UCI Repository Database

The characteristics of the data set are shown in table 2. The classification ranking results for Discrete Adaboost and RBF SVM are shown in fig. 3. The ranking for Discrete Adaboost in fig. 3(a) shows that the label approaches of our LW decoding tend to outperform the rest of the decoding strategies for all databases and coding strategies. The Loss-based decoding strategy and the probabilistic model show similar behavior, and the Euclidean strategy obtains the lower performance. Observe that one-versus-one and ECOC-ONE coding strategies show the best accuracy. On the other hand, the output margin provided by Adaboost seems to be not robust enough to increase the performance of the LW decoding strategies. In the ranking of SVM, one-versus-one and ECOC-ONE codings also attain the best accuracy, and the label variants of *LW* increase the performance of the Euclidean, Loss-based and probabilistic decodings. Besides, in this case the LW output margin outperforms in most cases the label approaches. In particular, the exponential LW variant is clearly superior to the linear approach in this case, which supports the use of the prediction obtained by the margin of SVM.

Problem	#Train	#Test	#Attributes	#Classes
Dermathology	366	-	34	6
Iris	150	-	4	3
Ecoli	336	-	8	8
Wine	178	-	13	3
Glass	214	-	9	7
Thyroid	215	-	5	3
Vowel	990	-	10	11
Balance	625	-	4	3
Yeast	1484	-	8	10
Satimage	4435	2000	36	7
Letter	20000	-	16	26
Pendigits	7494	3498	16	10
Segmentation	2310	-	19	7

Table 2: UCI repository databases characteristics.

5 Conclusions

In this paper, we presented the Loss-Weighted decoding strategy, that obtains a very high performance either in the binary and in the ternary ECOC framework. The Loss-Weighted algorithm shows higher robustness and better performance than the state-of-the-art decoding strategies. The validation of the results is performed using the state-of-the-art coding and decoding strategies with Adaboost and SVM as base classifiers, categorizing a wide set of datasets from the UCI Machine Learning repository. The high success of the experiments shows the suitability of the present methodology to be applied over any type of Machine Learning and Computer Vision multi-class classification problems.

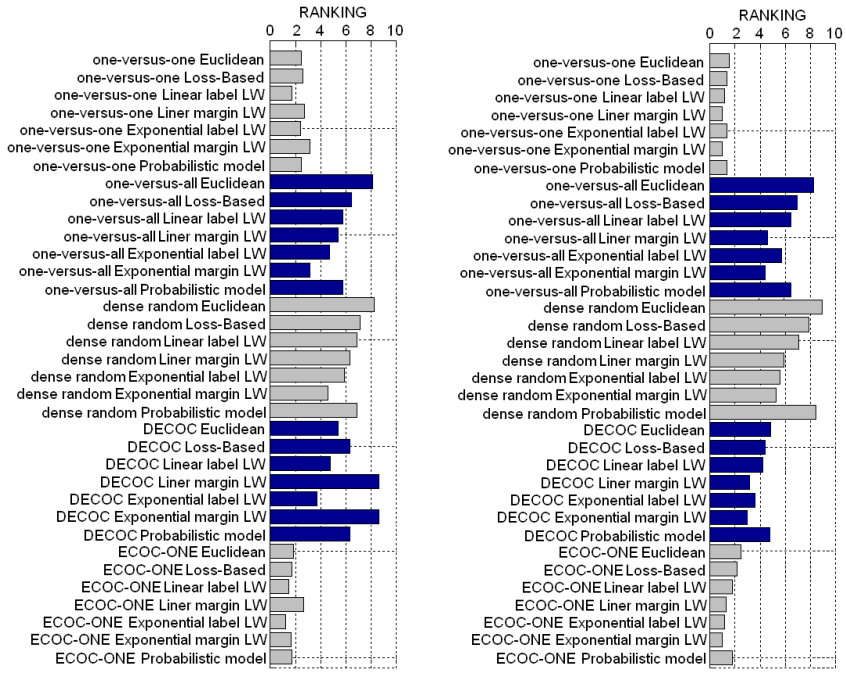
6 Acknowledgements

This work has been supported in part by TIN2006-15308-C02-01 and FIS ref. PI061290.

References

[1] E. Allwein, R. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying ap-

[3] We decided to keep the parameter fixed for sake of simplicity, though we are aware that this parameter might not be optimal for all data sets. Since the parameters are the same for all compared methods any weakness in the results will also be shared.



(a) (b)

Figure 3: Mean ranking on UCI data sets using Discrete Adaboost (a) and RBF SVM (b).

proach for margin classifiers. In *JMLR*, volume 1, pages 113–141, 2002.

[2] O. Dekel and Y. Singer. Multiclass learning by probabilistic embeddings. In *NIPS*, volume 15, 2002.

[3] T. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. In *J. of Artificial Intelligence Research*, volume 2, pages 263–286, 1995.

[4] T. Dietterich and E. Kong. Error-correcting output codes corrects bias and variance. In *ICML*, pages 313–321, 1995.

[5] S. Escalera, O. Pujol, and P. Radeva. Ecoc-one: A novel coding and decoding strategy. In *ICPR*, 2006.

[6] T. Hastie and R. Tibshirani. Classification by pairwise grouping. In *The annals of statistics*, volume 26, pages 451–471, 1998.

[7] N. Ishii, E. Tsuchiya, Y. Bao, and N. Yamaguchi. Combining classification improvements by ensemble processing. In *Int. proc. in conf. ACIS*, pages 240–246, 2005.

[8] N. Nilsson. In *Learning Machines, McGraw-Hill*, 1965.

[9] A. Passerini, M. Pontil, and P. Frasconi. New results on Error Correcting Output Codes of kernel machines. In *IEEE Trans. Neural Networks*, volume 15, pages 45–54, 2004.

[10] O. Pujol, P. Radeva, and J. Vitrià. Discriminant ECOC. In *IEEE Trans. of PAMI*, volume 28, pages 1007–1012, 2006.

[11] T. Windeatt and R. Ghaderi. Coding and decoding for multiclass learning problems. In *Information Fusion*, volume 1, pages 11–21, 2003.