Book Title
Book Editors
IOS Press, 2003

Optimal Extension of Error Correcting Output Codes

Sergio Escalera [a], Oriol Pujol [b], and Petia Radeva [a]
[a] *Centre de Visió per Computador, Campus UAB, 08193 Bellaterra (Barcelona), Spain*
[b] *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. In this paper, we propose a novel technique called ECOC-ONE (Optimal Node Embedding) to improve an initial ECOC configuration defining a strategy to create new dichotomies and improve optimally the performance. The process of searching for new dichotomies is guided by the confusion matrices over two exclusive training subsets. A weighted methodology is proposed to take into account the different relevance between dichotomies. We validate our extension technique on well-known UCI databases. The results show significant improvement to the traditional coding techniques with far few extra cost.

Keywords. Error Correcting Output Codes, Multiclass classification

1. Introduction

Machine learning studies automatic techniques to make accurate predictions based on past observations. There are several multiclass classification techniques: Support Vector Machines [1], multiclass Adaboost [2], decision trees [3], etc. Nevertheless, building a highly accurate multiclass prediction rule is certainly a difficult task. 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. Various 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).

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) [4] or one-versus-all grouping voting techniques or trees of nested dichotomies [5] are some of the most frequently used schemes. In the line of the aforementioned techniques Error Correcting Output Codes [6] were born. ECOC represents a general framework based on a coding and decoding (ensemble strategy) technique 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][4]. Moreover, the ECOC technique has demonstrated to be able to decrease the error caused by the bias and the variance of the base learning algorithm [8].

In the ECOC 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) shows 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. Very few attention has been paid in the literature to the coding part of the ECOC. The most known coding strategies are one-versus-all, all-pairs (one-versus-one) and random coding. Crammer et. al [9] were the first authors reporting improvement in the design of the ECOC 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. 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. [10]. They propose embedding of discriminant tree structures in the ECOC framework showing high accuracy with a very small number of binary classifiers, still the maximal number of dichotomies is bounded by the classes to be analyzed.

In this article, we introduce the ECOC Optimal Nodes Embedding (ECOC-ONE), that can be considered as a general methodology for increasing the performance of any given ECOC coding matrix. The ECOC-ONE is based on a selective greedy optimization based on the confusion matrices of two exclusive training data sets. The first set is used for standard training purposes and the second one for guiding and validation avoiding classification overfitting. As a result, wrongly classified classes are given priority and are used as candidate dichotomies to be included in the matrix in order to help the ECOC convergence. Our procedure creates an ECOC code that correctly splits the classes while keeping a reduced number of classifiers. Besides, we compare our optimal extension with another standard state-of-art coding strategies applied as coding extensions.

2. Error Correcting Output Codes

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 the "coding matrices" M, where $M \in \{-1, 1\}^{N_c \times n}$, being n the code length. From point of view of learning, matrix M representes n binary learning problems (dichotomies), each corresponding to a matrix column. Joining classes in sets, each dichotomy defines a partition of classes (coded by +1,-1 according to their class membership). 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. The matrix values can be extended to the trinary case $M \in \{-1, 0, 1\}^{N_c \times n}$, indicating that a particular class is not considered (gets 0 value) by a given dichotomy. To design an ECOC system, we need a coding and a decoding strategy. When the ECOC technique was first developed it was believed that the ECOC code matrices should be designed to have certain properties 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.

Most of the discrete coding strategies up to now are based on pre-designed problem-independent codeword construction satisfying the requirement of high separability between rows and columns. These strategies include one-versus-all that uses N_c dichotomies, random techniques, with estimated length of $10 \log_2(N_c)$ bits per code for Dense random and $15 \log_2(N_c)$ for Sparse random [4], and one-versus-one with $N_c(N_c - 1)/2$ dichotomies [11]. The last one mentioned has obtained high popularity

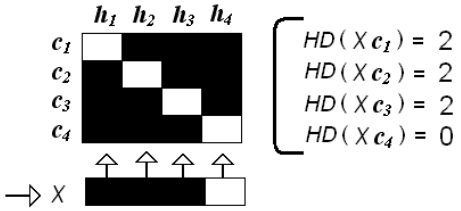

Figure 1. Coding matrix M for a four classes one-versus-all toy problem. New test sample with codeword X is classified to class c_4 of minimal distance using the Hamming distance.

showing a better accuracy in comparison to the other commented strategies. These traditional coding strategies are based on a prior division of subsets of classes independently of the problem to be used.

The decoding step was originally based on error-correcting principles under the assumption that the learning task can be modeled as a communication problem, in which class information is transmitted over a channel [12]. The decoding strategy corresponds to the problem of distance estimation between the codeword of the new example and the codewords of the trained classes. Concerning the decoding strategies, two of the most standard techniques are the Euclidean distance $d_j = \sqrt{\sum_{i=1}^{n}(x_i - y_i^j)^2}$ and the Hamming decoding distance $d_j = \sum_{i=1}^{n} |(x_i - y_i^j)|/2$, where d_j is the distance to the row class j, 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. 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.

In fig. 1 an example of a coding matrix M for an one-versus-all toy problem is shown. The problem has four classes, and each column represents its associated dichotomy. The dark and white regions are coded by -1 and 1, respectively. The first column h_1 represents the training of $\{c1\}$ vs $\{c2, c3, c4\}$, and so on. A new test input is evaluated using dichotomies $h_1, ..., h_4$, and its codeword X is decoded using the Hamming distance (HD) between each row of M and X. Finally, the new test input is classified by the class of minimum distance (c_4, in this case).

3. ECOC-ONE

ECOC-Optimal Node Embedding defines a general procedure capable of extending any coding matrix by adding dichotomies based on discriminability criteria.

Given a multiclass recognition problem, our procedure starts with a given ECOC coding matrix. The initial coding matrix can be one of the previously commented or one generated by the user. We increase this ECOC matrix in an iterative way, adding dichotomies that correspond to different spatial partitions of subsets of classes $\wp$. These partitions are found using a greedy optimization based on the confusion matrices so that the ECOC accuracy improves on both exclusive training subsets. Our training set is partitioned in 2 training subsets: a training subset of examples that guides the convergence process, and a validation subset, that leads the optimization process in order to avoid classification overfitting. Since not all problems require the same dichotomies, our opti-

mal node embedding approach generates new dichotomies just for classes not well separable yet. Thus, we construct an optimal ECOC-ONE matrix dependent of the concret domain.

To explain our procedure, we divide the ECOC-ONE algorithm into 3 steps: optimal node estimation, weights estimation, and ECOC-ONE matrix construction. The training process guided by the two training and validation subsets, ignores a significant amount of data from the training set, which can be redundant or harmful to the learning process, and avoid overfitting [13].

Let us define the notation used in the following paragraphs: given a data pair (x, l), where x is a multidimensional data point and l is the label associated to that sample, we define $\{\mathbf{x}, \mathbf{l}\} = \{\mathbf{x_t}, \mathbf{l_t}\} \bigcup \{\mathbf{x_v}, \mathbf{l_v}\}$, where $\{\mathbf{x_t}, \mathbf{l_t}\}$ and $\{\mathbf{x_v}, \mathbf{l_v}\}$ are the sets of data pairs associated to training and validation sets, respectively. In the same way, $e(h(\mathbf{x}), \mathbf{l})$ represents the empirical error over the data set $\mathbf{x}$ given an hypothesis $h(\cdot)$.

3.1. Optimal node estimation

Test accuracy of the training subsets: To introduce each network node, first, we test the current M accuracy on the training subsets. For this step, we find the resulting codeword $x \in \{-1, 1\}^n$ for each class sample of these subsets, and we label it as follows:

$$\tilde{\mathrm{l}} = argmin_j \quad \big(d(\mathcal{H}(M, h, x), y_j)\big) \tag{1}$$

where $d(\cdot)$ is a distance value between $\mathcal{H}(M, h, x)$ and the codeword y_j. $\mathcal{H}(M, h, x)$ is the strong hypothesis resulting in applying the set of learning algorithms $h(\cdot)$, parameterized with Θ on the problems defined by each column of the ECOC matrix M on a data point x. The result of $\mathcal{H}(M, h, x)$ is an estimated codeword. We propose the use of a weighed Euclidean distance in the following way:

$$d = \sqrt{\sum_{i=1}^{n} w_i (x_i - y_i^j)^2} \tag{2}$$

where the weight w_i introduces the relevance of each dichotomy in the learning ensemble technique.

The training and validation confusion matrices: Once we test the accuracy of the strong hypothesis $\mathcal{H}$ on the training and validation subsets, we estimate their respective confusion matrices ϑ_t and ϑ_v. Both confusion matrices are of size $N_c \times N_c$, and have at position (i, j) the number of instances of class c_i classified as class c_j.

$$\vartheta(i, j) = \big|\{(x, l) \,|\, h(x) = c_i \,,\, l = c_j\}\big| \tag{3}$$

where $h(x)$ is the label estimation obtained using equation (2) and l is the true label of example x. Once the matrices have been obtained, we select the pair $\{c_i, c_j\}$ with maximum value according to the following expression:

$$\{c_i, c_j\} = argmax_{\{C_i, C_j; i \neq j\}} \big(\vartheta_t(i, j) + \vartheta_t^T(i, j,) + \vartheta_v(i, j) + \vartheta_v^T(i, j)\big) \tag{4}$$

$\forall (i, j) \in [1, ..., N_c]$, where ϑ^T is the transposed matrix. The resulting pair is the set of classes that are more easily confounded, and therefore they have the maximum partial empirical error.

Find the new dichotomy: Once the set of classes with maximal error has been obtained, $\{c_i, c_j\}$, we create a new column of the ECOC matrix as follows: each candidate column considers a possible pair of subsets of classes $\wp = \{\{c_i \bigcup C^1\}, \{c_j \bigcup C^2\}\} \subseteq C$ so that $C^1 \cap C^2 \cap c_i \cap c_j = \oslash$ and $C^i \subseteq C$. In particular we are looking for the subset division of classes $\wp$ so that the dichotomy h_t associated to that division minimizes the empirical error defined by $e(\{\mathbf{x}, \mathbf{l}\})$.

$$\tilde{\wp} = argmin_{\wp} \quad \big(e(\mathcal{H}(M \cup m_i(\wp), h, \mathbf{x}), \mathbf{l})\big) \tag{5}$$

where $m_i(\wp)$ follows the rule in equation (7). The column components associated to the classes in the set $\{c_i, C^1\}$ are set to $+1$, the components of the set $\{c_j, C^2\}$ are set to -1 and the positions of the rest of classes are set to zero. In the case that multiple candidates obtain the same performance the one involving more classes is preferred. Firstly, it reduces the number of uncertainty in the ECOC matrix by reducing the number of zeros in the dichotomy. Secondly, one can see that when more classes are involved the generalization is greater. Each dichotomy finds a more complex rule on a greater number of classes. This fact has also been observed in the work of Torralba et al. [14]. In their work a multi-task scheme is presented that yields to an improved generalization classifier by aids of class grouping algorithm. This work shows that this kind of learners can increase generalization performance.

3.2. Weights estimates

It is known that when a multiclass problem is decomposed in binary problems, not all of these base classifiers have the same importance and generate the same decision boundaries. Our approach uses a weight to adjust the importance of each dichotomy in the ensemble ECOC matrix. In particular, the weight associated to each column depends on the error obtained when applying the ECOC to the training and validation subsets in the following way,

$$w_i = 0.5 \log(\frac{1 - e_i}{e_i}) \tag{6}$$

where w_i is the weight for the i^{th} dichotomy, and e_i is the error produced by this dichotomy at the affected classes of the two training subsets of classes. This equation is based on the weighed scheme of the additive logistic regression [2].

Update the matrix: The column m_i is added to the matrix M and the weight w_i is calculated using equation (6).

3.3. ECOC-ONE matrix construction

Once we have generated the optimal nodes, we embed each one in the following way: consider the set of classes associated to a node $C_i = \{C_{i1} \cup C_{i2} | C_{i1} \cap C_{i2} = \oslash\}$, the element (i, r) of the ECOC-ONE matrix corresponding to class i and dichotomy r is filled as (7). The summarized ECOC-ONE algorithm is shown in fig. 1.

$$M(r, i) = \begin{cases} 0 \quad \text{if } c_r \notin C_i \\ +1 \text{ if } c_r \in C_{i1} \\ -1 \text{ if } c_r \in C_{i2} \end{cases} \tag{7}$$

Given N_c classes and a coding matrix M (see fig. 1):

for $t = 1$ to T iterations:

1) Compute the optimal partition $\wp_i$ of the subset of classes
2) Test accuracy on the training and validation subsets.
3) Select the pair of classes $\{C_i, C_j\}$ with the highest error analyzing the confusion matrices from the training and validation subsets.
4) Find the partition $\wp_i$ containing $\{C_i, C_j\}$ that minimizes the error rate in the training and validation subsets.
5) Compute the weight for the dichotomy of partition $\wp_i$ based on the error.

Update the matrix M.

Table 1. ECOC-ONE extension algorithm

As mentioned before, one of the desirable properties of the ECOC matrix is to have maximal distance between rows. In this sense, our procedure focuses on the relevant difficult partitions, increasing the distance between the classes. This fact improves the robustness of the method since difficult classes are likely to have a greater number of dichotomies focussed on them. In this sense, it creates different geometrical arrangements of decision boundaries, and leads the dichotomies to make different bias errors.

4. Results

To test our proposed extension method, we extend the most well-known strategies used for ECOC coding: one-versus-all ECOC (one-vs-all), one-versus-one ECOC (one-vs-one), and Dense random ECOC. We have chosen dense random coding because it is more robust than the sparse technique for the same number of columns [4]. The decoding strategy for all mentioned techniques is the standard Euclidean distance because it shows the same behavior as the Hamming decoding but it also reduces the confusion due to the use of the zero values [10]. The number of dichotomies considered for Dense random is based on $10 \times log_2 n$, where n is the number of classes for a given database. The decoding strategy for our ECOC-ONE extension is the weighted euclidean distance. The weak classifier used for all the experiments is Gentle Adaboost. Nevertheless, note that our technique is generic in the sense that it only uses the classification score. In this sense it is independent of the particular base classifier. All tests are calculated using ten-fold cross-validation and a two-tailed t-test with a 95% confidence interval. In order to test ECOC-ONE coding extension, we have used a set of very well-known databases from UCI repository. The description of each database is shown in table 2.

To test our extension technique, we have extended the three commented coding strategies embedding 3 new dichotomies for all cases. The new 3 dichotomies embedded by dense random maximize the hamming distance between matrix rows. The results of extending one-versus-all, Dense random, and one-versus-one matrices in 5 UCI databases are shown in tables 3, 4 and 4 respectively. For each case we show the hit obtained and the number of dichotomies used for that experiment (#D). One can observe that adding just 3 extra dichotomies the accuracy increase considerably in comparison with the initial coding length. Besides, our problem-dependent ECOC-ONE coding extension

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

Table 2. UCI repository databases characteristics.

Problem	one-versus-all		one-versus-all-ONE		one-versus-all-dense	
	Hit	#D	Hit	#D	Hit	#D
Ecoli	77.00±1.14	8	**80.60±0.75**	11	77.75±1.02	11
Yeast	51.28±0.99	10	**55.84±1.08**	13	**54.76±1.06**	13
Glass	62.34±2.17	7	**65.17±1.80**	10	**65.52±2.07**	10
Dermathology	93.17±0.82	6	**95.43±0.72**	9	**94.70±0.69**	9
Vowel	73.97±1.73	11	**83.63±0.81**	14	78.43±1.41	14
Rank	4.00		**1.00**		1.40	

Table 3. Results of coding extensions of one-versus-all for UCI repository database.

outperform in all cases the Dense extension strategy due to the problem-dependent optimal selection of the extra dichotomies. One can observe that the confidence rates for our proposed technique is comparable and decreased in most cases in comparison with the results obtained by the dense extension strategy.

If we compare the initial differences between one-versus-all and one-versus-one for the initial codes, their results are considerable different. When the one-versus-all initial code is extended with 3 extra ECOC-ONE dichotomies, the results are comparable with the obtained using one-versus-one with far less cost.

Problem	Dense random		Dense random-ONE		Dense random-dense	
	Hit	#D	Hit	#D	Hit	#D
Ecoli	80.55±0.79	30	**82.90±0.84**	33	80.35±0.93	33
Yeast	55.33±1.12	33	**57.86±1.20**	36	**56.90±1.01**	36
Glass	65.52±1.80	28	**68.52±1.02**	31	66.34±1.88	31
Dermathology	96.13±0.73	26	**97.49±0.74**	29	**96.35±0.67**	29
Vowel	79.30±1.43	35	**83.53±1.29**	38	78.97±1.47	38
Rank	2.20		**1.00**		2.00	

Table 4. Results of coding extensions of Dense random for UCI repository database.

5. Conclusions

In most of the ECOC coding strategies, the ECOC matrix is pre-designed, using the same dichotomies in any type of problem. We introduced a new coding and decoding strategy called ECOC-ONE. The ECOC-ONE strategy can be seen as a general extension for any initial coding matrix. The procedure shares classifiers among classes in the ECOC-ONE matrix, and selects the best partitions weighed by their relevance. In this way, it reduces the overall error for a given problem. Moreover, using the validation subset the general-

Problem	one-versus-one		one-versus-one-ONE		one-versus-one-dense	
	Hit	#D	Hit	#D	Hit	#D
Ecoli	80.35±1.61	28	**80.65±1.59**	31	**81.20±1.29**	31
Yeast	54.58±1.10	45	**56.83±0.89**	48	54.48±0.94	48
Glass	67.38±1.98	21	**68.97±1.99**	24	**67.79±1.88**	24
Dermathology	95.48±0.80	15	**96.95±0.67**	18	**95.83±0.82**	18
Vowel	86.00±1.16	55	**88.96±1.07**	58	81.33±1.24	58
Rank	2.00		**1.00**		1.80	

Table 5. Results of coding extensions of one-versus-one for UCI repository database.

ization performance is increased and overfitting is avoided. We show that this technique improves in most cases the performance of any initial code with few extra cost better than other distance maximization extensions. Besides, ECOC-ONE can generate an initial small code by itself. As a result, a compact - small number of classifiers - multiclass recognition technique with improved accuracy is presented with very promising results.

6. Acknowledgements

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

References

[1] V. Vapnik, The nature of statistical learning theory, Springer-Verlag, 1995.
[2] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting, The Annals of Statistics vol. 38 (2) (1998) 337–374.
[3] L. Breiman, J. Friedman, R. Olshen, C. Stone, Classification and Regression Trees, Wadsworth, 1984.
[4] E. Allwein, R. Schapire, Y. Singer, Reducing multiclass to binary: A unifying approach for margin classifiers, Journal of Machine Learning Research vol. 1 (2002) 113–141.
[5] E. Frank, S. Kramer, Ensembles of nested dichotomies for multiclass problems, in: Proceedings of 21^{st} International Conference on Machine Learning, 2004, pp. 305–312.
[6] T. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, Journal of Artificial Intelligence Research vol. 2 (1995) 263–286.
[7] T. Windeatt, R. Ghaderi, Coding and decoding for multi-class learning problems, Information Fusion vol. 4 (1) (2003) 11–21.
[8] T. Dietterich, E. Kong, Error-correcting output codes corrects bias and variance, in: P. of the 21th ICML (Ed.), S. Prieditis and S. Russell, 1995, pp. 313–321.
[9] K. Crammer, Y. Singer, On the learnability and design of output codes for multiclass problems, Machine Learning 47 (2) (2002) 201–233.
[10] O. Pujol, P. Radeva, J. Vitrià, Discriminant (ecoc): A heuristic method for application dependent design of error correcting output codes, Transactions on PAMI 28 (6) (2006) 1001–1007.
[11] T. Hastie, R. Tibshirani, Classification by pairwise grouping, The annals of statistics vol. 26 (5) (1998) 451–471.
[12] T. Dietterich, G. Bakiri, Error-correcting output codes: A general method for improving multiclass inductive learning programs, in: A. Press (Ed.), 9th CAI, 1991, pp. 572–577.
[13] H. Madala, A. Ivakhnenko, Inductive Learning Algorithm for Complex Systems Modelling, CRC Press Inc, 1994.
[14] A. Torralba, K. Murphy, W. Freeman, Sharing visual features for multiclass and multiview object detection, MIT AIM.