Recoding Error-Correcting Output Codes

Sergio Escalera, Oriol Pujol, and Petia Radeva

Computer Vision Center, Campus UAB, Edifici O, 08193, Bellaterra, Spain.
Dept. Matemàtica Aplicada i Anàlisi, UB, Gran Via 585, 08007, Barcelona.

Abstract. One of the most widely applied techniques to deal with multi-class categorization problems is the pairwise voting procedure. Recently, this classical approach has been embedded in the Error-Correcting Output Codes framework (ECOC). This framework is based on a coding step, where a set of binary problems are learnt and coded in a matrix, and a decoding step, where a new sample is tested and classified according to a comparison with the positions of the coded matrix. In this paper, we present a novel approach to redefine without retraining, in a problem-dependent way, the one-versus-one coding matrix so that the new coded information increases the generalization capability of the system. Moreover, the final classification can be tuned with the inclusion of a weighting matrix in the decoding step. The approach has been validated over several UCI Machine Learning repository data sets and two real multi-class problems: traffic sign and face categorization. The results show that performance improvements are obtained when comparing the new approach to one of the best ECOC designs (one-versus-one). Furthermore, the novel methodology obtains at least the same performance than the one-versus-one ECOC design.

1 Introduction

Recently, significant amount of robust binary classifiers have been proposed in the bibliography with very high performance, such as Support Vector Machines, Neural Networks, Adaboost [1], etc. However, the extension of many binary classifiers to the multi-class case, where N possible categories appear, is a hard task. In this sense, a common strategy consists of defining a set of binary problems, which are combined in a Multiple Classifier system.

Error-Correcting Output Codes (ECOC) were defined as a framework to combine binary problems in order to deal with the multi-class case [2]. This framework is based on two main steps. At the first step, named coding, a set of binary problems (dichotomizers) are defined based on the learning of different sub-partitions of classes by means of a base classifier. Then, each of the partitions is embedded as a column of a coding matrix M, which rows correspond to the codewords codifying each class. At the second step, named decoding, a new data sample that arrives to the system is tested, and a codeword formed as a result of the output of the binary problems is obtained. This test codeword is compared with each class codeword based on a given decoding measure, and a classification prediction is obtained for the new object. Unlike the voting procedure, the information provided by the ECOC dichotomizers are shared among classes in order to obtain a precise classification decision, being able to reduce either the variance as the bias produced by the learners [3].

When Dietterich et. al. defined the binary ECOC framework in [2], all positions from the coding matrix M belonged to the $\{+1, -1\}$ symbols. It makes all classes to be considered by each dichotomizer as a member of one of both possible partitions of classes that define each binary problem. In this case, the one-versus-all and dense random ECOC approaches were defined [2]. Afterwards, Allwein et. al. in [4] defined the ternary ECOC, where the positions of the coding matrix M can be either $+1$, -1 or 0, and the sparse random and one-versus-one (pairwise voting) designs could be defined in the ECOC framework. In this case, the zero symbol means that a given class is not considered in the learning process of a particular dichotomizer. The huge set of possible bi-partitions of classes from this ternary ECOC framework has recently suggested the use of problem-dependent designs as well as new decoding strategies[5][6][7][8][9].

Concerning the one-versus-one ECOC strategy, it codifies the splitting of each possible pair of classes as a dichotomizer, which results in $N(N-1)/2$ binary problems for an N-class problem. This number is usually larger in comparison with the linear tendency of the rest of ECOC designs. Although this suggests larger training times, the individual problems that we need to train on are significantly smaller, and if the training algorithm scales superlinearly with the training set size, it is actually possible to save time. Moreover, the problems to be learnt are usually easier, since the classes have less overlapping. For all these reasons, the one-versus-one ECOC design tends to obtain better results than the rest of ECOC designs in real multi-class problems[5][7].

In this paper, we focus on the one-versus-one coding matrix design. Our goal is to look for a better coding of the matrix without retraining the classifiers involved. Training data are used in a problem-dependent way for updating the zero positions to $+1$ or -1 symbols if a higher classification performance can be achieved. Observe the 4-classes problem shown in Fig. 1(a). A decision boundary of a non-linear classifier has been obtained in the learning process of the dichotomizer h_1 that splits classes c_1 and c_2. The point of this article is that without the necessity of retraining the classifier, the same decision boundary can be used to give a prediction hypothesis about class c_3. On the other hand, note that the use of this decision boundary to classify class c_4 may result in a random decision function. Using this information, we recode the classical problem-independent one-versus-one into a problem-dependent one-versus-one design extending the trained classifier on new classes for the binary classifier for which the dichotomizer is relevant. The design is possible thanks to a new weighting procedure that takes into account the performance of the dichotimizers at the decoding step [7]. Moreover, the approach requires almost the same training and testing computational complexity than the classical one (since retraining of classifiers is not required).

The paper is organized as follows: Section 2 describes the recoded problem-dependent one-versus-one approach. Section 3 evaluates the methodology over a set of UCI data sets and two real multi-class problems: traffic sign and faces categorization. Finally, section 4 concludes the paper.

2 Recoded one-versus-one ECOC

In this section, we present a problem-dependent redefinition of the classical one-versus-one ECOC design. The one-versus-one ECOC technique is defined in the ternary ECOC framework $M^{N\times M} \in \{-1, 0, +1\}$, being M a coding matrix of N rows (as the number of classes), M the number of columns (dichotomizers to be learnt, where $M = N(N-1)/2$ in the case of the one-versus-one design), $\{-1, +1\}$ symbols codify the class membership, and the zero symbol ignores a particular class for a given dichotomizer. Each column of the matrix M corresponds to the ith binary problem h_i, which splits a pair of classes using a given base classifier. Figure 1(b) codifies a coding matrix M for a 4-class problem. The white positions correspond to the symbol +1, the black positions to the symbol -1, and the grey positions to the zero symbol. Note that this design is independent from the problem-domain. Once the set of binary problems $h = \{h_1, .., h_M\}$ is learnt, a new test sample ρ that arrives to the system is tested applying the set h, and a test codeword $x^{1\times M} \in \{-1, +1\}$ is obtained. Afterwards, a decoding function $d(x, y_j)$ is used to compare the test codeword x with each codeword y_j (jth row from M) codifying class c_j. Finally, the classification prediction corresponds to the class c_j which corresponding codeword y_j minimizes d.

In the one-versus-one ECOC design, only $2M$ from the NM possible positions are coded to $\{-1, +1\}$ symbols, which corresponds to a $(1 - 2/N) \cdot 100$ percentage of positions coded to zero. Note that the zero symbol does not give class membership information for its corresponding dichotomizer. Then, it could happen that if some of these positions coded to zero are re-coded to +1 or -1 without the need of re-training the dichotomizers, the final performance could be improved almost without increasing the training cost.

2.1 RECOC coding

Given the training data $C = \{C_1, .., C_N\}$, where C_i is the data belonging to class c_i, and M the one-versus-one coding matrix, the set of dichotomizers $h = \{h_1, .., h_M\}$ is learnt applying a base classifier over the corresponding subsets of C, obtaining the classical one-versus-one ECOC design. In order to update the coding matrix in a problem-dependent way, for each position $M(i, j) = 0$, the corresponding data C_i, $i \in \{1, .., N\}$, $i \notin (k, l)$, where c_k and c_l are the classes considered by the jth dichotomizer, are tested using h_j under the hypothesis that their membership should be +1. Then, a classification accuracy β is obtained. If the magnitude of β or $(1 - \beta)$ is greater than a performance threshold $\alpha \in (0.5, 1]$, then that position of the coding matrix M is set (recoded) to +1 (or -1), respectively. Otherwise, the value of $M(i, j)$ is kept to zero.

Since we use the training data to modify the positions of M, the one-versus-one design mutates in a problem-dependent way. Moreover, since the modification of the positions of M does not require to retrain the set h, the computational cost of the coding process is not significantly increased. Table 1 shows the algorithm for training the Recoding ECOC (RECOC) design. The algorithm codifies the classical one-versus-one design at the same time that modifies the positions of M based on the input value of α. Note that in the algorithm, a matrix of weights W saving the accuracy values β is defined. This matrix will be used at the decoding process in order to weight the final classification.

```
Input: $\alpha$, $C = \{C_1, .., C_N\}$ // Accuracy value and multi-class data
Output: $M$, $W$, and set of dichotomizers $h = \{h_1, .., h_M\}$
$W^{N \times M} := 0$, $M^{N \times M} := 0$, $cont := 1$
for $i \in \{1, .., N-1\}$
    for $j \in \{i+1, .., N\}$
        Given a base classifier, learn dichotomizer $h_{cont}$ to split $(C_i, C_j)$
        // Update membership and accuracy
        $M(i, cont) := +1$, $W(i, cont) := h_{cont}(C_i, +1)$
        // Update membership and accuracy
        $M(j, cont) := -1$, $W(j, cont) := h_{cont}(C_j, -1)$
        for $k \in \{i, .., N\}$
            if $k \notin \{i, j\}$
                // Accuracy for class $c_k$ considered as class $c_i$ (label $+1$)
                $\beta := h_{cont}(C_k, +1)$
                // Consider the coding matrix position $k$ as $+1$
                if $\beta \geq \alpha$ then
                    // Update membership and accuracy
                    $M(k, cont) := +1$, $W(k, cont) := \beta$
                // Consider coding matrix position $k$ as -1
                elseif $1 - \beta \geq \alpha$ then
                    // Update membership and accuracy
                    $M(k, cont) := -1$, $W(k, cont) := 1 - \beta$
                endif
            endif
        endfor
        $cont := cont + 1$
    endfor
endfor
```

Table 1. RECOC learning algorithm.

In order to obtain more precise classification results, we need to know which values of α are useful to increase the generalization capability of the system, since some values of α may result in wrong classification predictions. In order to look for the values of α, cross-validation is applied. For this task, the training data C is split into a training C^T and a validation C^V subsets, so that $C = C^T \cup C^V$. The use of a validation subset helps the system to increase generalization. Thus, for a set of values $\alpha = \{\alpha_1, .., \alpha_k\}$, algorithm 1 is called. However, the set h is only learnt once over C at the beginning. At each round, the set C^T is used to mutate the positions of M, and the validation set C^V will be used to test the performance of each M for a particular α. For this last task, a decoding procedure using the weighting matrix W is proposed next. This step is required to obtain a successful classification. Finally, the matrix M for which value of α maximizes the classification performance over C^V is selected.

Figure 1 shows an example of a training process for a 4-class problem. Figure 1(a) shows the non-linear decision boundaries that splits all possible pairs of classes. Figure 1(b) shows the classical one-versus-one design. Figure 1(c) shows the problem-dependent coding matrix M for $\alpha = 0.9$. Note that several positions previously coded to zero are now set to +1 or -1 values since they achieve an accuracy upon 90% over the training data. Finally, Fig. 1(d) shows the same process for $\alpha = 1.0$. Now, less positions satisfy the performance restrictions. Note that if the testing of the validation data C^V does not take benefits from the values of α, then, the classical one-versus-one design is selected, and thus, in the worst case, the recoded problem-dependent approach attains the same performance than the classical approach.

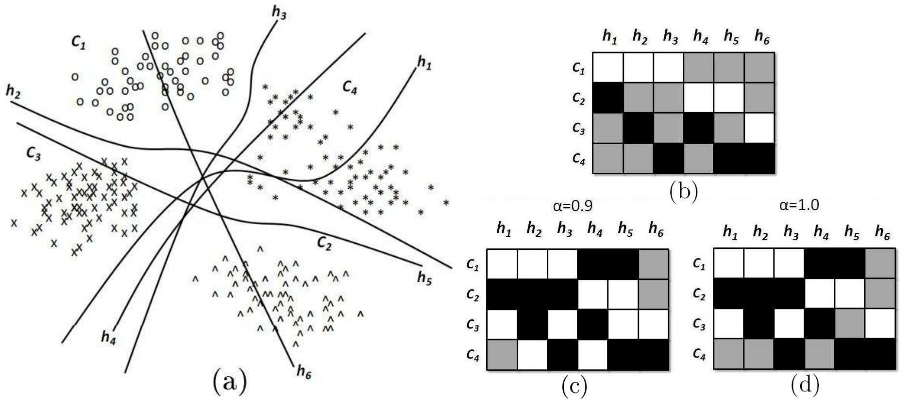


Fig. 1. ECOC codification for a 4-class problem: (a) Non-linear decision boundaries for the 4-class problem, (b) initial one-versus-one ECOC codification, (c) RECOC codification with $\alpha = 0.9$, and (d) RECOC codification with $\alpha = 1.0$.

2.2 RECOC decoding

In [7], the authors show that to properly decode a ternary ECOC matrix two biases must be avoided at the decoding step. First, classical decoding strategies introduce a bias when comparing positions that contain the zero symbol, which do not give information about meta-class membership. On the other hand, the addition of the bias produced by the comparison with the zero symbol makes the codewords to take values from different ranges, which makes the measures among codewords non-comparable. In this sense, the authors present how to robustly decode sparse coding matrices where codewords may contain different number of positions coded to $\{-1, +1\}$ symbols. This is done by weighting the final decision so that it avoids the influence of the zero symbol at the same time that all classes codewords have the same probability of being predicted.

Due to the previous properties, we use a Loss-based decoding [4] weighted by the weighting matrix W computed at the RECOC coding step to decode the RECOC matrix M. The approach uses a Loss-function to penalize the miss-classifications produced by the set of dichotomizers h.

First, we normalize each row of the weighting matrix W obtained at the coding step so that M_W can be considered as a discrete probability density function $M_W(i,j) = \frac{W(i,j)}{\sum_{j=1}^{M} W(i,j)}$, $\forall i \in [1, ..., N]$, $\forall j \in [1, ..., M]$. Once we obtain the normalized weighting matrix M_W, we introduce it in a Loss-based decoding [4]. In this approach, the decoding estimation is obtained by means of a Loss-based model with a Loss-function $L(\theta)$ weighted by M_W, where $L(\theta) = -\theta$

and θ corresponds to $y_i^j \cdot h^j(\rho)$: $LW(\rho, i) = \sum_{j=1}^{n} M_W(i,j) L(y_i^j \cdot h_j(\rho))$. The final classification decision is done by the class c_i which corresponding codeword y_i that minimizes the LW function.

3 Results

In order to present the results, first, we discuss the data, methods, measurements, and experimental settings of the experiments.

• *Data*: The data used for the experiments consist of eleven multi-class data sets from the UCI Machine Learning Repository database [10]. We also categorize two real Computer Vision classification problems. First, we use the video sequences obtained from a Mobile Mapping System [11] to test the methods in a real traffic sign categorization problem consisting of 36 traffic sign classes. Second, 30 classes from the ARFaces [12] data set are classified using the present methodology.

• *Methods*: We compare the classical one-versus-one ECOC design with the RECOC strategy for three base classifiers: Gentle Adaboost [1], Linear Support Vector Machines [13], and Support Vector Machines with Radial Basis Function kernel ($RBF\ SVM$) [13]. In order to compare the methods at same conditions, we use a linear Loss-Weighted decoding in both one-versus-one and RECOC strategies.

• *Measurements*: To measure the performance of the different strategies, we apply stratified ten-fold cross-validation and test for confidence interval with a two-tailed t-test.

• *Experimental settings*: 50 decision stumps are considered for the Gentle Adaboost algorithm. The $RBF\ SVM$ classifier is tuned via cross-validation, where the σ and regularization parameters are tested from 0.05 increasing per 0.05 up to 1 and from one increasing per 5 up to 150, respectively. For the RECOC strategy cross-validation is applied, where α is tested from 0.7 increasing per 0.05 up to 1, and 10% of the training data are used as a validation subset.

3.1 UCI classification

Table 2 shows the performance results of the one-versus-one ECOC and RECOC algorithms for the different ECOC base classifiers. For each UCI data set, the performance obtained by each method is shown. In the cases where RECOC improves the one-versus-one ECOC results, the selected values of α are shown. The number of wins, losses, and draws considering the ten experiments of the ten-fold cross-validation for each data set are also shown in the table. Note that in several data sets, RECOC obtains performance improvement for the three base classifiers. The table shows that the more classes there are, the more significant the results are. The highest performances are achieved for high values of α (about 0.90-0.95 in most cases). Note that in the worst case, RECOC becomes the one-versus-one ECOC designs, and it achieves the same performance. Moreover, looking at the wins and losses of each experiment, one can see that though in some case the performance improvements of RECOC are no significant, the number of wins of the ten-fold experiments are statistically significant.

Gentle Adaboost	one-versus-one	RECOC	α	Wins	Losses	Draws
Balance	**87.46**	**87.46**	-	0	0	10
Wine	**94.38**	**94.38**	-	0	0	10
Thyroid	**95.37**	**95.37**	-	0	0	10
Iris	**95.33**	**95.33**	-	0	0	10
Glass	63.10	**68.65**	0.95	10	0	0
Ecoli	81.29	**83.36**	0.75	8	2	0
Dermatology	91.76	**92.52**	0.85	5	0	5
Vowel	57.88	**62.73**	0.95	9	1	0
Vehicle	57.81	**63.57**	0.95	9	1	0
Yeast	55.46	**56.67**	0.95	5	2	3
Segmentation	**97.45**	**97.45**	-	0	0	10
Linear *SVM*	one-versus-one	RECOC	α	Wins	Losses	Draws
Balance	**91.64**	**91.64**	-	0	0	10
Wine	**95.55**	**95.55**	-	0	0	10
Thyroid	**96.71**	**96.71**	-	0	0	10
Iris	**98.67**	**98.67**	-	0	0	10
Glass	28.74	**37.58**	1.00	5	1	4
Ecoli	**74.63**	**74.63**	-	0	0	10
Dermatology	94.79	**95.07**	0.95	1	0	9
Vowel	63.33	**64.44**	0.95	8	2	0
Vehicle	**80.24**	**80.24**	-	0	0	10
Yeast	26.11	**37.81**	0.95	9	1	0
Segmentation	96.02	**96.32**	1.00	6	2	2
RBF SVM	one-versus-one	RECOC	α	Wins	Losses	Draws
Balance	97.25	**97.41**	0.95	1	0	9
Wine	61.31	**61.84**	1.00	1	0	9
Thyroid	**95.35**	**95.35**	-	0	0	10
Iris	**96.67**	**96.67**	-	0	0	10
Glass	**46.41**	**46.41**	-	0	0	10
Ecoli	**86.74**	**86.74**	-	0	0	10
Dermatology	88.80	**89.05**	0.85	3	0	7
Vowel	54.95	**55.76**	0.90	4	1	5
Vehicle	72.00	**72.12**	0.90	1	0	9
Yeast	**56.68**	**56.68**	-	0	0	10
Segmentation	95.14	**95.25**	0.90	2	0	8

Table 2. UCI performances for the different ECOC base classifiers.

Now, we compare the results obtained by the RECOC approach on the UCI data sets with the results obtained with the same strategy retraining classifiers. In Fig. 3 one can see the performance obtained by both classification strategies for the three different base classifiers. Note that there are no significant differences among the obtained performances. Moreover, the RECOC strategy obtains better performance in more cases than using the same coding matrix retraining classifiers, with far less computational complexity.

3.2 Traffic sign categorization

For this experiment, we use the video sequences obtained from the Mobile Mapping System [11] to test the classification methodology on a real traffic sign categorization problem. In this system, the position and orientation of the different traffic signs are measured with video cameras fixed on a moving vehicle. From this system, a set of 36 circular and triangular traffic sign classes are obtained. Some categories from this data set are shown in Fig. 2(a). The data set contains a total of 3481 samples of size 32×32, filtered using the Weickert anisotropic filter, masked to exclude the background pixels, and equalized to prevent the effects of illumination changes. These feature vectors are then projected into a 100 feature vector by means of PCA. The classification results of the one-versus-one ECOC and RECOC strategies for the three base classifiers are shown in Table 3. In this experiment, for all base classifiers, the RECOC design obtains performance improvements for high values of α.

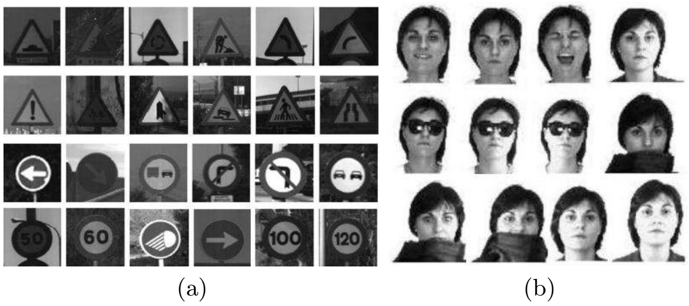
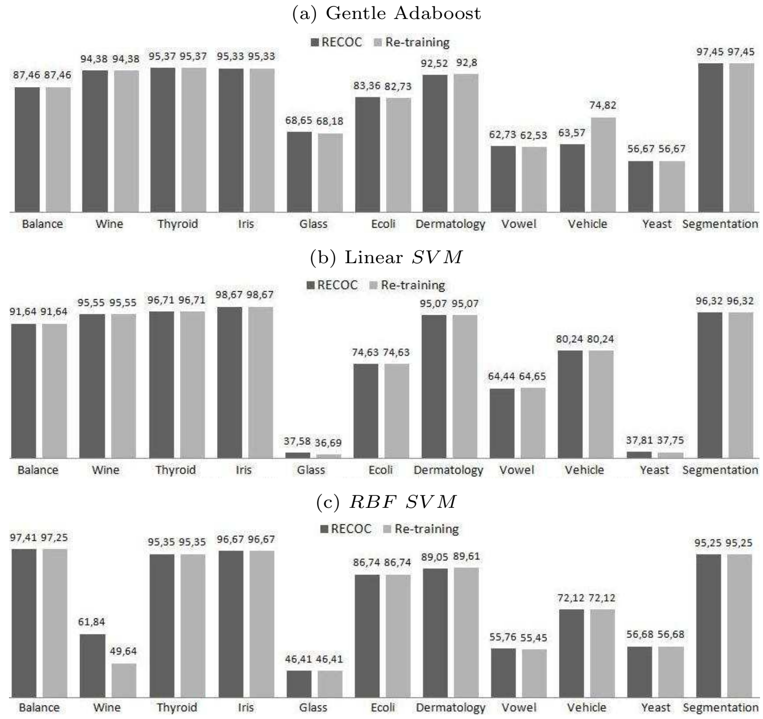


Fig. 2. (a) Traffic sign classes. (b) ARFaces data set classes. Examples from a category with neutral, smile, anger, scream expressions, wearing sun glasses, wearing sunglasses and left light on, wearing sun glasses and right light on, wearing scarf, wearing scarf and left light on, and wearing scarf and right light on.



(a) Gentle Adaboost

(b) Linear SVM

(c) RBF SVM

Fig. 3. UCI data sets performance using the recoded matrix with and without retraining.

Problem	one-versus-one	RECOC	α	Wins	Losses	Draws
Gentle Adaboost	88.70	88.95	0.95	3	1	6
Linear SVM	88.02	91.23	1.00	4	0	6
RBF SVM	97.44	97.85	0.95	1	0	9

Table 3. Traffic data set performances.

3.3 ARFaces classification

The AR Face database [12] is composed of 26 face images from 126 different subjects (70 men and 56 women). The images have uniform white background. The database has two sets of images from each person, acquired in two different sessions, with the following structure: one sample of neutral frontal images, three samples with strong changes in the illumination, two samples with occlusions (scarf and glasses), four images combining occlusions and illumination changes, and three samples with gesture effects. One example of each type is plotted in Fig. 2(b). For this experiment, we selected all the samples from 30 different categories (persons).

The classification results of the one-versus-one ECOC and RECOC strategies for the three base classifiers are shown in Table 4. As in the previous experiments, all base classifiers obtain performance improvements using the RECOC strategy for high values of α ($\alpha = 0.95$).

Problem	one-versus-one	RECOC	α	Wins	Losses	Draws
Gentle Adaboost	65.50	70.06	0.95	6	1	3
Linear SVM	39.41	43.92	0.95	9	1	0
$RBFSVM$	88.33	88.75	0.95	2	0	8

Table 4. ARFaces data set performances.

3.4 Discussion

As a final conclusion of the results, we can state that performance improvements are obtained using the RECOC approach instead of the one-versus-one ECOC. Note that none of the RECOC experiments for any base classifier obtains inferior results to the one-versus-one performances.

Concerning the computational complexity of the strategy, the classifiers learnt at the coding step are not retrained during the RECOC recodification. Thus, though cross-validation of α should be applied to assure the better performance, the training cost is not significantly increased. On the other hand, the testing time remains the same than in the classical one-versus-one approach since all classifiers should be applied on the test sample. Moreover, we show that we obtain similar (even superior) results with the recoded RECOC matrix M than using the same procedure but retraining classifiers (that is, using the re-coded positions to re-train again the dichotomizers).

Finally, it is important to bring up that though the recoding strategy has been performed on the one-versus-one coding matrix, this strategy is directly applicable to any kind of ternary ECOC design where the symbol zero may appear.

4 Conclusion

In this paper, we presented a problem-dependent design of Error-Correcting Output Codes to deal with multi-class categorization problems. The method is based on redefining the classical one-versus-one ECOC design so that the generalization of the system is increased. For this task, the training data are analyzed using the previously learnt binary problems, and the coding matrix is recoded without the need of retraining classifiers. A weighting matrix is also included in order to weight the final classification and obtain more precise results. The experimental evaluation over several UCI Machine Learning repository data sets and two real multi-class problems: traffic sign and face categorization, show that significant performance improvements can be obtained. Moreover, our new methodology is guaranteed by design to achieve at least the one-versus-one performance.

Acknowledgments

This work has been supported in part by projects TIN2006-15308-C02, FIS PI061290, and CONSOLIDER-INGENIO CSD 2007-00018.

References

1. J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting, in: The annals of statistics, Vol. 38, 1998, pp. 337–374.
2. T. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, in: Journal of Artificial Intelligence Research, Vol. 2, 1995, pp. 263–282.
3. E. B. Kong, T. G. Dietterich, Error-correcting output coding corrects bias and variance, in: ICML, 1995, pp. 313–321.
4. E. Allwein, R. Schapire, Y. Singer, Reducing multiclass to binary: A unifying approach for margin classifiers, in: JMLR, Vol. 1, 2002, pp. 113–141.
5. S. Escalera, D. Tax, O. Pujol, P. Radeva, R. Duin, Subclass problem-dependent design of error-correcting output codes, in: Transactions in Pattern Analysis and Machine Intelligence, Vol. 30, 2008, pp. 1041–1054.
6. O. Pujol, P. Radeva, , J. Vitrià, Discriminant ECOC: A heuristic method for application dependent design of error correcting output codes, in: PAMI, Vol. 28, 2006, pp. 1001–1007.
7. S. Escalera, O. Pujol, P. Radeva, On the decoding process in ternary error-correcting output codes, in: Transactions in Pattern Analysis and Machine I Intelligence, in press.
8. S. Escalera, O. Pujol, P. Radeva, Boosted landmarks of contextual descriptors and Forest-ECOC: A novel framework to detect and classify objects in clutter scenes, in: Pattern Recognition Letters, Vol. 28(13), 2007, pp. 1759–1768.
9. O. Pujol, S. Escalera, P. Radeva, An incremental node embedding technique for error correcting output codes, in: Pattern Recognition, Vol. 41, 2008, pp. 713–725.
10. A. Asuncion, D. Newman, UCI machine learning repository, in: University of California, Irvine, School of Information and Computer Sciences, 2007.
    URL http://mlearn.ics.uci.edu/MLRepository.html
11. J. Casacuberta, J. Miranda, M. Pla, S. Sanchez, A.Serra, J.Talaya, On the accuracy and performance of the GeoMobil system, in: International Society for Photogrammetry and Remote Sensing, 2004.
12. A. Martinez, R. Benavente, The AR Face database, in: Computer Vision Center Technical Report #24, 1998.
13. Osu-svm-toolbox.
    URL http://svm.sourceforge.net