Adding Classes Online in Error Correcting Output Codes Framework

Sergio Escalera[⋆‡], David Masip[†‡], Eloi Puertas[⋆], Petia Radeva[⋆‡] and Oriol Pujol[⋆‡]
⋆ *Dept. Matemàtica Aplicada i Anàlisi, Universitat de Barcelona 585, Edifici Històric, 08007, Barcelona, Spain.*
† *Universitat Oberta de Catalunya, Rambla del Poblenou 156, 08018, Barcelona, Spain.*
‡*Computer Vision Center (CVC), Edifici O, Campus Universitat Autònoma de Barcelona, 08193, Barcelona, Spain.*
sergio@maia.ub.es, dmasipr@uoc.edu, {eloi,petia,oriol}@maia.ub.es

***Abstract*—This article proposes a general extension of the Error Correcting Output Codes (ECOC) framework to the online learning scenario. As a result, the final classifier handles the addition of new classes independently of the base classifier used. Validation on UCI database and two real machine vision applications show that the online problem-dependent ECOC proposal provides a feasible and robust way for handling new classes using any base classifier.**

I. Introduction

Machine vision applications are constantly evolving, a fact leading to the development of strong online classifiers that can deal with the variability of the data. Given a classification task, the goal of online learning is to model the classifiers parameters using an initial training set, being able to incrementally evolve this model parameters as new data or classes become available.

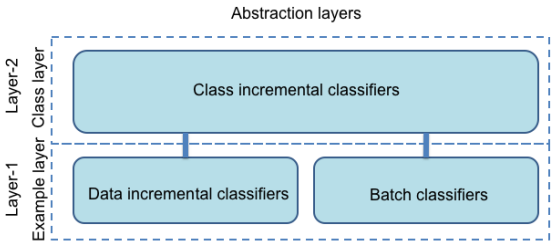

Figure 1. Abstraction layers of the online behavior.

The online learning objectives are clearly differentiable in comparison to classic batch learning. However, the "online" term involves different levels of behavior associated to the classifier. One can clearly distinguish two behaviors at different level of abstraction: the first one, from the point of view of data examples - one expects the classifier to adapt to new data from previously seen classes. The second abstraction layer, from the point of view of classes - one expects the classifier to adapt to new classes without retraining the complete classifier. Figure 1 shows the relationship between these two layers. The bottom layer deals only with examples; in this layer, we can distinguish between data incremental/decremental classifiers and batch classifiers, depending on the un/capability of the method to adapt to new data examples. In this layer, most online learning approaches are based on extending classical binary classifiers to the online case, like decision trees, online Support Vector Machines (SVM), or online ensemble of classifiers [1], [2]. Online feature extraction has also been studied in the prototype based classification leading to methods like Incremental Principal Component Analysis (iPCA) or the extension of the Fisher Linear Discriminant Analysis criterion [3].

On a higher level, we find the class incremental/decremental behavior. This layer deals with the addition/removal of classes. Up to now, literature has considered the first and second layers as one. However, few are the methods that allow both behaviors at the same time, and most of literature is focussed on the first layer. This second level of abstraction is where our proposal, online ECOC, is defined. As a meta-learning strategy it can accommodate either example incremental online classifiers of the first layer or batch classifiers.

In this paper, we study the suitability of the ECOC [4] framework to adapt to the online learning scenario. In particular, we focus on layer-2 using ECOC coding schemes, which incrementally allow new classes to be added to the original problem independently of the base classifier. The addition of unseen samples in the online ECOC becomes straightforward using an online (layer-1) base binary classifier. In particular, we study, propose, and evaluate a problem dependent matrix generation algorithm, and validate it over the UCI repository and two computer vision problems.

The paper is organized as follows: Section 2 overviews the ECOC framework and proposes its extension to the layer-2 online case. Section 3 explains the presented method. Section 4 shows the validation, and finally, section 5 concludes this work.

II. Online Error Correcting Output Codes

ECOC technique is a metalearning strategy that allows to extend any binary classifier to the multiclass case. The classic ECOC meta learning algorithm [4] has two phases: in the learning step, an ECOC encoding matrix is constructed in order to define the combination of the M binary classifiers that allow full multi-class classification. In the testing (decoding) phase, the new sample $\mathbf{x}$ is classified according to the M binary classifiers set. The decoding algorithm finds

the most suitable class label for the test sample using the output of this binary set of classifiers.

Briefly, given a set of N training samples $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, where each $\mathbf{x}_i$ belongs to the class $C_i \in \{C_1, \ldots, C_K\}$, an ECOC encoding consists on constructing M binary problems (called dichotomizers, h_j) from the original K classes. At each dichotomizer, the class set is split into the binary classes $\{+1, -1\}$, forming a $K \times M$ encoding ECOC matrix $\mathbf{T}$. When a new sample must be classified, the outputs of the dichotomizers (columns of the matrix $\mathbf{T}$) are used to construct the codeword that is compared with each row of the matrix $\mathbf{T}$. The class with the minimum distance is selected as the classifier output. In this binary grouping setting, the total number of possible splits is $2^{K-1} - 1$, being the efficient construction of the ECOC matrix $\mathbf{T}$ the key issue in the training step. The encoding step was extended by Alwein et al [5] to include a new symbol - symbol 0 - so that, a class can be omitted in the training of a particular dichotomizer. Classical multi-class classification strategies, such as the one-vs-all or one-vs-one (when ternary representations are used) can be easily represented as an ECOC matrix. Nevertheless, more sophisticated problem dependent encodings have been shown to outperform classical approaches [6], without a significant increment of the codeword length.

Possible decoding strategies are the Hamming distance between the output of the dichotomizers on the new test sample and each codeword (row) of the encoding matrix as well as Euclidean decoding, Probabilistic decoding or Loss-based decoding.

The recent Weighted decoding [6] has shown to be generally better than most of the state of the art decoding measures. The weighting methodology is designed to fulfill two properties that allow a better behavior of the binary and ternary decoding, being able to decode matrices with any sparseness degree.

III. Online ECOC coding

The new class addition in the ECOC matrix reshapes it in two ways: First, a new code must be defined for the new class. Second, new dichotomies could be needed to discriminate the new class. Additionally, any knowledge in the ECOC matrix should be preserved, i.e. it is undesirable to retrain previously learnt dichotomizers. In this section, we propose a problem dependent method to extend online the ECOC matrix to consider the addition of new classes.

A. General dependent online ECOC (PDo)

Traditionally, the grouping properties of the encoding matrix $\mathbf{T}$ are predefined at design time. The input $K \times M$ encoding ECOC matrix $\mathbf{T}$, where each $T(i,j) \in \{+1, -1\}$ represents the binary metaclass membership of the class i in the dichotomizer j, can be statically extended with a new dichotomizer h_{M+1} and a new codeword C_{K+1}. Nevertheless, it seems that more efficient grouping decisions could be performed when the specific nature of the problem is taken into account. A problem dependent approach can select the proper values of $T(i,j)$ using a data driven criterion, such as the training error on a validation subset. We take advantage of the weighted decoding, which allows to take into account the metaclass relative accuracy (r-value), defined as follows:

Definition: The metaclass relative accuracy (r-value) of class k on the set S given the definition of the metaclasses ρ is defined as follows,

$$r_k(S, \rho, i) = \frac{\#\text{elements of class } k \text{ classified as metaclass } i \text{ in the set } S}{\#\text{elements belonging to class } k \text{ in the set } S}, \quad (1)$$

where ρ defines classes belonging to metaclasses.

The **PDo** coding is built in three steps: (formulated in Algorithm 1):

1) *Vertical extension:* This step consists of creating a new codeword for the new class. In Fig.2, the first step tests dichotomizers (h_1, h_2, h_3) with the samples of the new class, assuming that they belong to the metaclasses $\{+1, -1\}$. The metaclass value with higher r-value is assigned to the new code. In the example, the new code corresponds to $\{-1, 0, +1\}$. As the r-value obtained for the new class by h_2 is not higher than ϵ, the result of the dichotomizer is not considered in the decoding.
2) *Base horizontal extension:* A new dichotomizer specialized on the new class is included. In Fig.2, the second step sets class 5 to $+1$ and the rest to -1. The r-value for each class must be stored in the weighting matrix W.
3) *Problem-dependent horizontal refinement:* This step proposes an r-value driven variable codeword expansion. In Fig.2, C_2 is the class with the highest error in the confusion vector for the new class. We set the value corresponding to that class to 0 in h_4 and create a new dichotomizer specialized on distinguishing C_2 from C_5.

Note that we get independence of the base classifier.

IV. Results

First, we discuss the data, methods, measurements, and experimental settings of the experiments.

• *Data*: We use eleven multi-class data sets from the UCI Machine Learning Repository database [7]. Then, we apply the online classification methodology in a 36-class computer vision Mobile Mapping System [8] and a 30-class problem from the ARFaces [9] data set.

• *Methods*: We test the multi-class OSU implementation of RBF-Support Vector Machines as batch classifier. In the case of the online classifiers, we compare the $iLDA$ with one-nearest neighbor classifier and our ECOC online methodology with and without online base classifiers: *PDo* and the batch *PDo*. The different classifiers parameters are

Input: Set of data points $S = \{(x_i, C_i)|x_i \in \mathbf{X} \wedge C_i \in \mathbf{C}\}$ divided in a training set $S_t \subset S$ and a validation set $S_v \subset S$ so that, $S_t \bigcup S_v = S$ and $S_t \bigcap S_v = \varnothing$
Input: ECOC matrix T of size $K \times M$
Input: Set of new training instances $S_o = \{x_{N+1}, \ldots, x_{N+U}\}$ from a new class C_{K+1}
Input: Parameters ϵ and α
Output: Expanded ECOC matrix $\tilde{T}$

begin step 1: *Vertical expansion*
  foreach *column/dichotomy* $j \in \{1 \ldots M\}$ **do**
    /* Find the weight associated to that class for dichotomy j as the maximum metaclass relative accuracy for all possible codes */
    $W(K+1, j) = max(\alpha r_{K+1}(S_t, T(\cdot, j), l) + (1 - \alpha) r_{K+1}(S_v, T(\cdot, j), l)) \quad \forall l \in \{1, -1\}$
    if $W(K+1, j) < \epsilon$ **then**
      $W(K+1, j) = 0$; $\tilde{T}(K+1, j) = 0$
    else
      /* Fill the ECOC matrix with the code value that maximized the weight */
      $\tilde{T}(K+1, j) = argmax_l(\alpha r_{K+1}(S_t, T(\cdot, j), l) + (1 - \alpha) r_{K+1}(S_v, T(\cdot, j), l)) \quad \forall\, l \in \{1, -1\}$
end

begin step 2: *Base horizontal expansion*
  $\tilde{T}(K+1, M+1) = -1$
  $\tilde{T}(j, M+1) = 1 \quad \forall j \in \{1 \ldots K\}$
  $w(K+1, M+1) =$
  $\alpha r_{K+1}(S_t, T(\cdot, M+1), -1) + (1-\alpha) r_{K+1}(S_v, T(\cdot, M+1), -1)$
end

begin step 3: *Problem dependent horizontal expansion*
  while $w(K+1, M+1) \leq \epsilon$ **and** $|\{\tilde{T}(j, M+1) = 1, \quad \forall j \in \{1 \ldots K\}\}| > 1$ **do**
    Calculate the confusion vector with respect to class C_{K+1}
    Select the class C_e with maximum error
    $\tilde{T}(e, M+1) = 0$
    Add a new column at position $s = length(\tilde{T}) + 1$ so that,
    $$\tilde{T}(j, s) = \begin{cases} 1 & j = e \\ -1 & j = K+1 \\ 0 & otherwise \end{cases}$$
    Find the new weights according to the new dichotomy definitions
    $w(j, M+1) = \alpha r_j(S_t, T(\cdot, M+1), T(j, M+1)) + (1 - \alpha) r_j(S_v, T(\cdot, M+1), T(j, M+1))$ and
    $w(j, s) = \alpha r_j(S_t, T(\cdot, s), T(j, s)) + (1 - \alpha) r_j(S_v, T(\cdot, s), T(j, s)) \quad \forall j \in \{1 \ldots K+1\}$
end

Algorithm 1: General algorithm for the creation of the problem dependent layer-2 online ECOC matrix.

tuned via cross-validation of the training set. All online multi-class experiments are solved by considering an initial 2-class problem and progressively increasing the number of classes by one.

• *Measurements*: We apply stratified ten-fold cross-validation and test for the confidence interval with a two-tailed t-test. We also use the Friedman and Nemenyi tests to look for significant statistical differences between the methods' performances [10].

A. Validation over UCI data sets

The average accuracy and rankings are shown in table I. The asterisks mark the best performance and the values in bold correspond to the methods which fall within the 95% confidence interval of the best result. The rankings are obtained estimating each relative rank r_i^j for each data set i and each classification strategy j, and computing the mean ranking R for each classifier as $R_j = \frac{1}{J}\sum_i r_i^j$, where J is the total number of data sets. For comparison

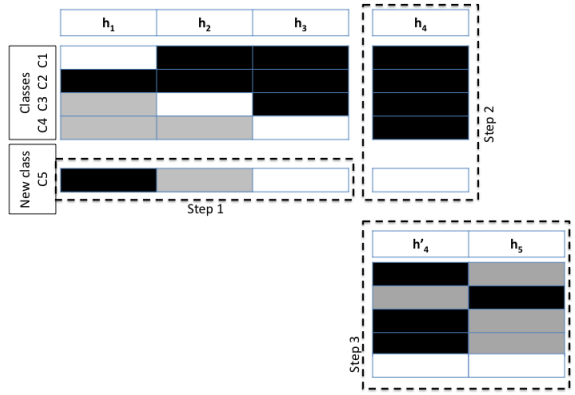

Figure 2. Problem-Dependent Online strategy. White, black, and grey regions are coded by +1, -1, and 0, respectively.

Table I
UCI CLASSIFICATION RESULTS.

DB	iLDA	PDo	batch PDo	SVM
Balance	0.84 (0.03)	**0.97 (0.02)** *	**0.97 (0.02)** *	0.97 (0.02)
Wine	**0.74 (0.06)** *	**0.65 (0.05)**	0.64 (0.04)	0.61 (0.04)
Thyroid	0.72 (0.03)	**0.96 (0.03)** *	**0.96 (0.03)** *	0.95 (0.03)
IRIS	**0.97 (0.03)** *	**0.95 (0.04)**	**0.95 (0.04)**	0.97 (0.04)
Glass	**0.53 (0.08)** *	**0.50 (0.04)**	**0.49 (0.04)**	0.46 (0.04)
Ecoli	**0.82 (0.04)**	**0.84 (0.03)**	**0.86 (0.02)**	0.85 (0.02)
Yeast	0.52 (0.02)	**0.58 (0.03)**	**0.57 (0.03)**	0.59 (0.02)
Vowel	**0.74 (0.04)** *	0.54 (0.05)	0.50 (0.03)	0.53 (0.04)
Derma.	0.88 (0.03)	**0.96 (0.01)** *	**0.96 (0.01)** *	0.96 (0.01)
Vehicle	0.38 (0.04)	**0.73 (0.02)** *	**0.73 (0.03)** *	0.72 (0.02)
Segmen.	0.60 (0.01)	**0.95 (0.02)** *	**0.91 (0.03)**	0.95 (0.02)
Rank	2.73	1.63	2.00	

purposes, the last column in the table shows the SVM results trained as a multi-class off-line classifier and it is not used in the computation of the ranking and confidence interval. Notice that the best online method is the PDo, followed by the batchPDo. In addition, we test for statistical significance applying the Nemenyi test [10], founding the PDo statistically outperform iLDA.

1) Traffic sign categorization: For this first computer vision experiment, we use the video sequences obtained from the Mobile Mapping System of [8] to test the online ECOC methodology on a real traffic sign categorization problem. A set of 36 circular and triangular traffic sign classes are obtained (Fig. 3). 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 [11].

The classification results of the traffic sign data sets are shown in table II. The ranks are computed taking into account each iteration of the 10-fold evaluation as a different experiment. On can see that the PDo approach outperforms the results of the rest of strategies, and obtains comparable results to the off-line SVM.

In order to analyze if the difference between methods ranks is statistically significant, we applied the Friedman

Figure 3. Examples of Traffic sign and face classes.

Table II
TRAFFIC AND FACE DATA SETS CLASSIFICATION.

Traffic	SVM	iLDA	PDo	batch PDo
Performance	**0.97 (0.01)**	0.90 (0.01)	**0.95 (0.02)**	0.93 (0.03)
Rank	2.6	7.4	3.4	4.9
Face	SVM	iLDA	PDo	batch PDo
Performance	**0.88 (0.06)** *	0.49 (0.09)	**0.83 (0.07)**	**0.74 (0.09)**
Rank	1.2	7.2	2.4	3.9

test, rejecting the null hypothesis. Once we have checked for the non-randomness of the results, we use the Nemenyi test to compare 4 methods with a confidence value $\alpha = 0.10$. In this case, we obtain a critical difference that single out SVM and PDo as the best methods in the traffic sign recognition experiment.

2) Face classification: The AR Face database [9] is composed of 26 uniform white background face images from 126 subjects. The database has two sets of images for each person, acquired in two different sessions (Fig. 3). We selected all the samples from 30 different persons. The classification results and ranks (Table II) show that the differences among strategies are similar to the previous cases. The best results are obtained by the batch SVM approach, followed by the PDo and batch PDo strategies, respectively. Finally, the $iLDA$ approach offers the less performance. The Friedman test also rejects the null hypothesis, and Nemenyi critical difference single out SVM and PDo approaches as the best methods for the ARFace categorization.

V. CONCLUSIONS

In this paper, we proposed a general problem-dependent methodology for the design of online ECOC matrices for both online and batch base classifiers. We have shown different applications where the addition of classes online can be applied by considering an initial 2-class problem and progressively increasing the number of classes. The online results have been compared with multi-class batch classifiers, showing encouraging results.

REFERENCES

[1] N. C. Oza, "Online ensemble learning," in *AAAI/IAAI*. AAAI Press / The MIT Press, 2000, p. 1109.

[2] S. Katagiri and S. Abe, "Incremental training of support vector machines using hyperspheres," *Pattern Recognition Letters*, vol. 27, no. 13, pp. 1495–1507, 2006.

[3] M. Artac, M. Jogan, and A. Leonardis, "Incremental pca or on-line visual learning and recognition." in *ICPR (3)*, 2002, pp. 781–784.

[4] T. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," in *JAIR*, vol. 2, 1995, pp. 263–282.

[5] E. Allwein, R. Schapire, and Y. Singer, "Reducing multiclass to binary: A unifying approach for margin classifiers," in *JMLR*, vol. 1, 2002, pp. 113–141.

[6] O. P. Sergio Escalera and P. Radeva, "On the decoding process in ternary error-correcting output codes," *PAMI*, vol. 99, no. 1, 2009.

[7] A. Asuncion and D. Newman, "UCI machine learning repository," in *University of California, Irvine, School of Information and Computer Sciences*, 2007. [Online]. Available: http://mlearn.ics.uci.edu/MLRepository.html

[8] J. Casacuberta, J. Miranda, M. Pla, S. Sanchez, A.Serra, and J.Talaya, "On the accuracy and performance of the GeoMobil system," in *International Society for Photogrammetry and Remote Sensing*, 2004.

[9] A. Martinez and R. Benavente, "The ar face database," in *Computer Vision Center Technical Report #24*, 1998.

[10] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," in *JMLR*, vol. 7, 2006, pp. 1–30.

[11] S. Escalera, O. Pujol, and P. Radeva, "Traffic sign recognition system with β-correction," *MVA*, 2008.