Analyzing the Separability Matrix for ECOC coding

Miguel Ángel Bautista[*+], Sergio Escalera[*+], Xavier Baró[*+†], Oriol Pujol[*+]

[*] *Applied Math and Analisis, University of Barcelona, Gran via de les Corts Catalanes, Barcelona, Spain*
[+] *Computer Vision Center, Autonomous University of Barcelona, Bellaterra, Spain*
† *Telecommunications and Multimedia Dept. , Open University of Catalonia, Barcelona, Spain*
E-mail:mbautista@cvc.uab.es,{sergio,xevi,oriol}@maia.ub.es

Abstract

Error Correcting Output Codes (ECOC) have demonstrate to be a powerful tool for treating multi-class problems. Nevertheless, predefined ECOC designs may not benefit from Error-correcting principles for particular multi-class data. In this paper, we introduce the Separability matrix as a tool to study and enhance designs for ECOC coding.

Keywords: Error Correcting Output Codes, Problem-dependent designs, Separability matrix.

1 Introduction

Multi-class classification tasks are problems in which a set of N classes, categories or namely brands are categorized. In the ensemble learning field, Error Correcting Output Codes (ECOC) have demonstrated to be a powerful tool to solve multi-class classification problems [DB95]. This methodology divides the original problem of N classes in n binary problems (2-class problems). Commonly, the step of defining n binary partitions of the N classes is known as *coding*. At this step, a coding matrix $M_{N \times n} \in \{-1, +1\}$ is generated. The columns of M denote the n bi-partitions of the original problem, and the rows of M, known as *codewords*, identify each one of the N classes of the problem uniquely. Once M is defined, a set of n base classifiers $\{h_1, \ldots, h_n\}$ learn the n binary problems coded in M.

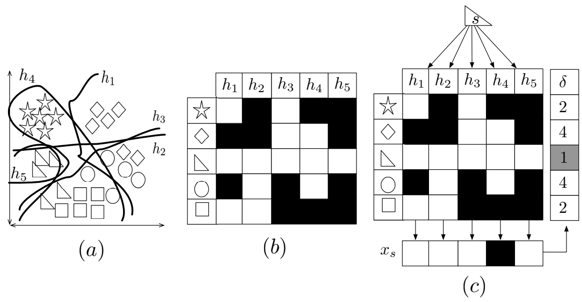

Figure 1: (a) Feature space and trained boundaries of base classifiers. (b) Coding matrix M, where black and white cells correspond to $\{+1, -1\}$. (c) Decoding step, where the predictions of classifiers, $\{h_1, \ldots, h_5\}$ for sample s are compared to the codewords $\{y_1, \ldots, y_N\}$.

At the *decoding* step, a new sample s is tested by each base classifier $\{h_1, \ldots, h_n\}$, obtaining a set of label predictions. The set of predictions x_s is compared to M using a decoding measure δ and sample s is labelled as the class code at minimum distance. In Figure 1, an example for coding and decoding steps is shown for a 5−class toy problem.

The coding step has been widely studied in literature [RK04], proposing either predefined or random coding designs. Nevertheless, predefined strategies may not be suitable for a given problem because they do not take into account the underlying distribution of the classes. In this scope, one can roughly find works on problem-dependent strategies for coding designs. In this paper we introduce the Separability matrix as a way to analyse and study the properties of a certain ECOC coding matrix. The Separability matrix explicitly shows the pairwise separation between the codes of all pairs of classes. With this tool in mind, we also propose a new compact problem-dependent coding design that shows the benefits of applying the separability criteria in a problem-dependent manner.

This paper is organized as follows: Section 2 introduces the Separability matrix, in Section 3 the novel problem-dependent coding design is proposed and, Section 4 shows the experimental results and concludes the paper.

2 The Separability matrix

One of the main concerns of the ECOC framework is to correct as many base classifiers errors as possible. In literature, the correction capability ρ of a coding matrix M is defined as $\rho = \frac{\min(\delta(y_i, y_j))-1}{2}$, $\forall i, j \in \{1, \ldots, N\}$, $i \neq j$. Therefore, distance between codewords and correction capability are directly related. Given this close relationship between distance and correction capability, we define the Separability matrix S, as follows:

Given an ECOC coding matrix $M_{N \times n}$, the Separability matrix $S_{N \times N}$ contains the distances between all pairs of codes in M. Let $\{y_i, y_j\}$ be two codewords, the Separability matrix S at position (i, j), defined as $S_{i,j}$, contains the distance between the codewords $\{y_i, y_j\}$, defined as $\delta(y_i, y_j)$. An example of Separability matrix estimation for two coding designs is shown in Figure 2.

Usually, the increment in the correcting capability problem has been tackled by enlarging the codeword length, and thus, the distance between codewords. However, Rifkin et al. show in [RK04] that if a classifier with high capacity is well optimized, small codes such as *One vs. All* are also suitable for solving the problem. Recently, following the same principle as Rifkin et al., in [BEB10] the authors propose to use a Compact ECOC matrix, with a code length of $\lceil \log_2(N) \rceil$, where $\lceil . \rceil$ round to the upper integer, which is optimized by a Genetic Algorithm in a problem-dependent manner.

If we analyse the Separability matrix S of predefined ECOC coding designs [RK04], we find that $S_{i,j} = \varsigma \; \forall i, j \in \{1, \ldots, N\}, i \neq j$, where ς is a constant separation value. This means that codewords are equidistant, as shown in Figure 2(d). Nevertheless, in problem-dependent coding strategies the Separability matrix acquires a great value, since it shows which codewords are prone to have more errors due to the lack of error correction capability. For example, if we analyse the Compact ECOC coding matrix M we find that codewords are not equidistant and the distribution of separability is not constant. An example of Compact ECOC coding and its Separability is shown in Figure 2(a) and 2(b), respectively.

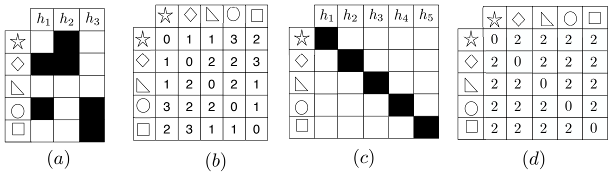

Figure 2: (a) Compact ECOC coding matrix. (b) Separability Matrix of a Compact ECOC. (c) *One vs. All* coding matrix. (d) Separability matrix of *One vs. All* coding.

3 Application of Separability matrix for Extension coding

Problem-dependent coding strategies have not been thoroughly studied in literature. In contrast to classical approaches [RK04], problem-dependent coding designs combine the error correcting principles with a guided coding procedure which takes into account the distribution of the data. In this work we define a problem-dependent coding design based on the Separability matrix to enhance the error correcting capabilities of the design.

In [BEB10] the authors propose a problem-dependent Compact ECOC coding matrix of length $\lceil \log_2 N \rceil$. However, the resultant matrix M has null correction capability since $\rho = 0$. On the other hand, one would like to have at least $\min(S) \geq 3$, to correct one error. This could be done by extending the codewords $\{y_1, \dots, y_N\}$ of the coding matrix M until $S_{i,j} = 3 \; \forall i, j \in \{1, \dots, N\}, i \neq j$. Therefore, we propose to extend the codewords of a non-optimized Compact ECOC coding (Binary ECOC), which is the binary representation of the N classes of our problem. This means that the codeword y_i of class c_i is the binary representation of a decimal value i $\forall i \in \{1, \dots, N\}$. This extension is calculated in order to increase the distance δ between the most confused codes, computing a problem-dependent extension still with a reduced code length. The proposed algorithm uses both Separability $S_{N\times N}$ and Confusion $C_{N\times N}$ matrices of a Binary ECOC to compute an extension of its coding matrix M, defined as $E_{N\times k}$ where k is the number of columns (base classifiers) of the extension.

The Confusion-Separability-Extension (CSE) coding algorithm is an iterative algorithm that looks for the most confused classes in C, i.e $\{c_i, c_j\}$ and codes an Extension matrix E that increases its separability $S_{i,j}$ until a certain user-defined separability value ϱ is achieved. In addition, the Extension matrix E also increments the separability for all the classes confused with c_i or c_j. This extension is performed in order to increase the correction with all the classes that are prone to confuse with classes c_i or c_j. When no classes are confused with $\{c_i, c_j\}$ the coding is performed taking into account the overall confusion with all classes $\{c_1, \dots, c_N\}$. Once E is completely coded, the algorithm checks if any column in E was previously on M. In that case, the algorithm changes specific codewords. Let t be an iteration of the algorithm, which codes E_t, then at iteration $t+1$, $M_{t+1} = M_t \cup E_t$, the algorithm will stop when in M, $n \geq N$, this stop condition is defined to upper bound the code length of the design to N. In addition, we consider that if $\delta(y_i, y_j) \geq \varrho$, then $C_{i,j} = 0$. Therefore, another stop condition for the algorithm is that $\forall i, j C_{i,j} = 0$, because that means that no confusion is left to treat. The CSE algorithm is illustrated in the toy example of Figure 3.

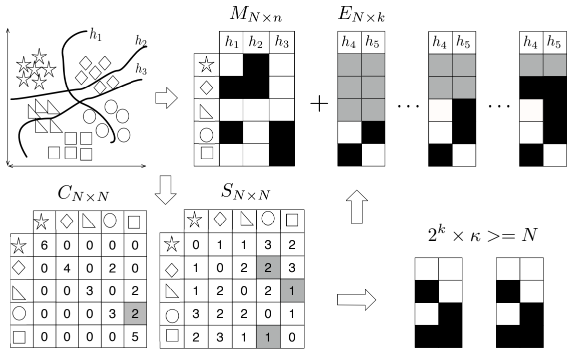

Figure 3: CSE example in a 5−class toy problem.

4 Experimental results

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

- **Data**: The first bench of experiments consists of seven muti-class problems extracted from the UCI Machine Learning Repository, showed in Table 2. In addition, we test our methodology over 3 challenging Computer Vision multi-class problems. First, we classify 70 visual object categories from the MPEG dataset. Then, 50 classes of the AR-Face database are classified. Finally, we test our method in a real traffic sign categorization problem consisting of 36 traffic sign classes.

Table 1: UCI classification results with SVM as base classifier.

	One vs. All ECOC		CSE ECOC $\varrho = 3$		CSE ECOC $\varrho = 5$		Dense Random ECOC	
Data set	Perf.	Classif.	Perf.	Classif.	Perf.	Classif.	Perf.	Classif.
Vowel	55.0±10.5	11	66.9±7.8	9.2	**69.8±6.3**	10.6	67.9±8.3	11
Yeast	41.0±7.3	10	54.7±11.8	5.7	53.0±9.3	9.5	**54.9±6.4**	10
Ecoli	**78.9±3.5**	8	76.4±4.4	7	78.6±3.9	7.4	72.1±2.7	8
Glass	51.6±10.2	7	**55.5±7.6**	6	52.7±8.4	3	42.8±11.02	7
Segment	**97.3±0.7**	7	96.9±0.8	6.6	96.6±1.0	6.2	96.6±1.3	7
Derma	97.1±1.2	6	**97.1±0.9**	5.2	95.9±1.2	3	95.7±0.8	6
Vehicle	80.1±4.0	4	**81.1±3.5**	3	70.6±3.4	3	81.1±3.6	4
MPEG7	83.2±5.1	70	88.5±4.5	15	89.6±4.9	20.4	**90.0±6.4**	70
ARFaces	76.0±7.22	50	80.7±5.2	13.8	84.6±5.3	20.2	**85.0±6.3**	50
Traffic	91.3±1.1	36	95.7±0.92	12.2	**96.6±0.8**	19	93.3±1.0	36
Rank & #	3.0	20.8	**2.2**	**8.8**	2.3	10.3	2.5	20.8

Table 2: UCI repository data sets characteristics.

Problem	#Training samples	#Features	#Classes
Dermathology	366	34	6
Ecoli	336	8	8
Vehicle	846	18	4
Segmentation	2310	19	7
Glass	214	9	7
Vowel	990	10	11
Yeast	1484	8	10

- **Methods**: We compare the *One vs. All* [RK04] ECOC approach with the CSE coding design with separability value $\varrho = \{3, 5\}$. In addition, we also compare our results with the Dense Random coding scheme using N classifiers. The ECOC base classifier is the libsvm implementation of a SVM with Radial Basis Function kernel. The SVM ζ and γ parameters are tuned via Genetic Algorithms for all the methods, minimizing the classification error of a two-fold evaluation over the training sub-set.

- **Evaluation Measurements**: The classification performance is obtained by means of a stratified ten-fold cross-validation.

  The classification results obtained for all the data sets considering the different ECOC configurations are shown in Table 1. In order to compare the performances provided for each strategy, the table also shows the mean rank of each ECOC design considering the twelve different experiments. The rankings are obtained estimating each particular ranking r_i^j for each problem i and each ECOC configuration j, and computing the mean ranking R for each design as $R_j = \frac{1}{N}\sum_i r_i^j$, where N is the total number of data sets. We also show the mean number of classifiers (#) required for each strategy.

To conclude, results show that the proposed method outperforms the *One vs. All* standard coding design in most cases, using far less number of dichotomizers. This is caused by the fact that the proposed algorithm focus the correcting capability in those classes more prone to be confused, and thus, less redundancy is needed. Nevertheless, when comparing Dense Random coding with our method in terms of performance, no significance is found since both methods have a comparable rank.

References

[DB95] T. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. In *JAIR*, volume 2, pages 263–286, 1995.

[BEB10] M.A. Bautista, S. Escalera, and X. Baro. Compact Evolutive Design of Error-Correcting Output Codes. In *SUEMA-ECML*, 119–128, 2010.

[RK04] R. Rifkin and A. Klautau. In defense of one-vs-all classification. In *JMLR*, 5:101–141, 2004.