Introducing Separability Matrix for Error Correcting Output Codes coding

Miguel Ángel Bautista[1,2], Oriol Pujol[1,2], Xavier Baró[1,2,3]
, and Sergio Escalera[1,2]

[1]Applied Math and Analisis Dept, University of Barcelona, Gran Via de les Corts Catalanes. 585, 08007 Barcelona, Spain
[2] Computer Vision Center, Campus UAB, Edifici O, 08193, Bellaterra, Spain
[3]Computer Science, Multimedia, and Telecommunications Dept, Universitat Oberta de Catalunya. Rambla del Poblenou 156, 08018 Barcelona
mbautista@cvc.uab.es,oriol@maia.ub.es,xbaro@uoc.edu,sergio@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. In addition, a novel problem-dependent coding design based on the Separability matrix is tested over a wide set of challenging multi-class problems, obtaining very satisfactory results.

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

1 Introduction

Multi-class classification tasks are problems in which a set of N classes, categories or namely brands are categorized. Most of state-of-the-art multi-class methodologies need to deal with the categorization of each class either by modelling its probability density function, or by learning a classification boundary and using some kind of aggregation/selection function to obtain a final decision. Another way to deal with multi-class problems is to use a divide-and-conquer approach. Instead of extending a method to cope with the multi-class case, one can divide the multi-class problem into smaller binary problems and then combine their responses using some kind of strategy, such as voting.

In the ensemble learning field, Error Correcting Output Codes (ECOC) have demonstrated to be a powerful tool to solve multi-class classification problems [CS02,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.

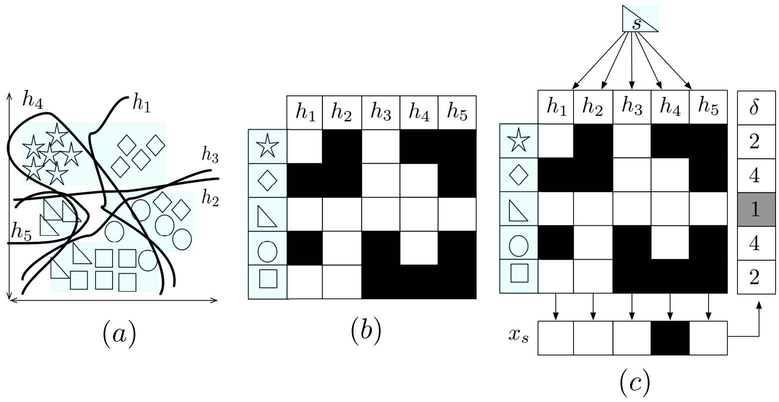

Fig. 1. (a) Feature space and trained boundaries of base classifiers. (b) Coding matrix M, where black and white cells correspond to $\{+1, -1\}$, denoting the two partitions to be learnt by each base classifier (white cells vs. black cells). (c) Decoding step, where the predictions of classifiers, $\{h_1, \ldots, h_5\}$ for sample s are compared to the codewords $\{y_1, \ldots, y_N\}$ and s is labelled as the class codeword at minimum distance.

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 each codeword of M using a decoding measure δ and sample s is labelled as the class c_i with codeword y_i at minimum distance (i-th row of M). In Figure 1, an example for coding and decoding steps is shown for a 5−class toy problem. Note that though classifier h_4 fails its prediction, s is correctly classified.

The coding step has been widely studied in literature [TR98,RK04,ASS02], proposing either predefined [TR98,RK04] or random [ASS02] coding designs always following the trend of reducing the number of used dichotomizers. 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 [EOR08,PRV06].

In this paper we introduce the Separability matrix as a way to analyse and study the properties of a certain ECOC coding matrix. Although the concept of separability has always been in the heart of all ECOC studies, up to this moment there has not been the need of defining explicitly a matrix of this kind. This is mainly due to the fact that predefined strategies assume that the coding matrix must have equidistant codewords. However, with the introduction of problem-dependent and sub-lineal coding designs this general assumption does not hold and more concise tools are needed for their analysis. The Separability matrix explicitly shows the pairwise separation between 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. Finally, Section 5 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 [TR98]. 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 [TR98,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). In fact, when dealing with predefined codings, the Separability matrix makes little sense and has been overlooked since all non-diagonal values are constant. 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.

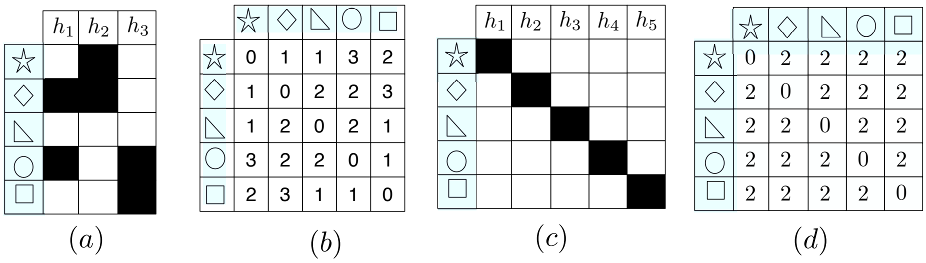

Fig. 2. (a) Compact ECOC coding matrix. (b) Separability Matrix of a Compact ECOC. (c) *One vs. All* coding matrix. (d) Separability matrix of *One vs. All* coding.

3 Application of Separability matrix for Extension coding

Problem-dependent coding strategies have not been thoroughly studied in literature [EOR08,PRV06]. In contrast to classical approaches [RK04,TR98,ASS02], 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. Moreover, we also take profit of the Confusion matrix to define the partitions of classes of each binary classifier.

In [BEB10] the authors propose a problem-dependent Compact ECOC coding matrix of length $\lceil \log_2 N \rceil$. However, the computational cost of optimizing this coding matrix is very expensive and in every case 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 $\{y_1, \ldots, y_N\}$ of the coding matrix M until $S_{i,j} = 3$ $\forall i, j \in \{1, \ldots, N\}, i \neq j$. However, we have to take into account that confusion is not equally distributed among all the classes, and thus separability might not have to be also equally distributed. Let $\{c_i, c_j, c_k, c_l\}$ be four classes of our N-class problem, then, if $(C_{i,j} + C_{j,k}) > (C_{k,l} + C_{l,k})$ (where $C_{i,j}$ is the number of samples of class c_i classified as class c_j), it will be more probable to misclassify a sample between classes c_i and c_j than between classes c_k and c_l. Thus, it will be more efficient to increase $\delta(y_i, y_j)$ than $\delta(y_k, y_l)$.

Therefore, following the idea of Compact ECOC coding, 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 $\forall i \in \{1, \ldots, N\}$. 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 separability 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 $\{c_1, \ldots, c_N\}$. 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, though smaller codes may be suitable. 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. Note that CSE coding algorithm only requires the C and S matrices generated by a Binary ECOC. In addition, no retraining or testing of classifiers is needed trough the extension process. Algorithm 1 shows the CSE coding algorithm, which is illustrated in the toy example of Figure 3.

Data: $M_{N\times n}$, $C_{N\times N}$, $S_{N\times N}$, ϱ
Result: $E_{N\times k}$
k `// separability increment needed`
$Y^E_{N\times k} \in \{-1, +1\}$`// set of unused generated codewords`
$S^m_{1\times 1} \in \{0, \ldots, \infty\}$`// minimum separability value`
$S^c_{p\times q} \in \{0, 1\}$,$p \leq N$,$q = 2$`// classes at minimum separability with` $\{c_i, c_j\}$
while $k + n < N$ *and* $\exists\ i, j\ C_{i,j} \geq 0$ **do**
  $(i, j) := \arg\max_{i,j}(C)$ `// look for the pair of classes` $\{c_i, c_j\}$ `with maximum confusion in` C
  $k := \varrho - S_{i,j}$;
  $Y^E :=$generateCodes(k,N) `// generate` 2^k `codes` κ `times until` N `codes are generated`
  $y^E_i := Y^E_1$ `// assign random code to one of the classes with maximum confusion`
  $(E, Y^E) :=$findCode(y^E_i, k, Y^E) `// find a code at` $\delta = k$ `with the code` Y^E_i
  while $S_m < \varrho$ **do**
    $(S^c, S^m) :=$findMinSepClasses(E, S, C);
    $(E, Y^E) :=$codifyMinSep(S^c, E, Y^E) `// look for a suitable code for` S^c
    $S^m = S^m + 1$;
  end
  if $\exists\{i, j\} : E_{i,j} = 0$ **then**
    $E :=$codifyZero(E, S, C, Y^E) `// codify the undefined codes in` E `taking into account confusion with` $\{c_1, \ldots, c_N\}$
  end
  $E :=$checkExtension(M, E) `// check if some column in` E `was previously in` M
  $(C, S, M) :=$updateMatrices(M, E, S, C) `// update confusion, separability and coding matrices`
end

Algorithm 1: CSE coding algorithm.

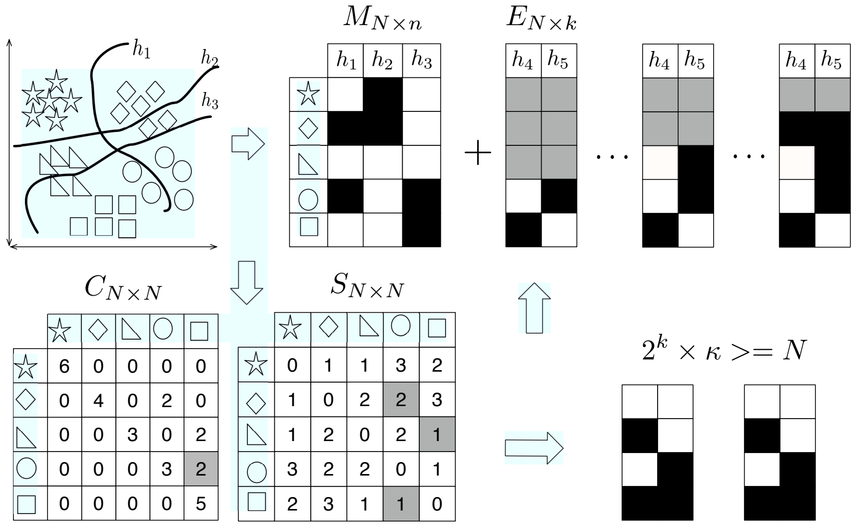

Fig. 3. CSE example in a 5−class toy problem.

The CSE coding algorithm codifies an Extension matrix E based on the Separability and Confusion matrices of a certain coding matrix M. Note that though in this paper this Extension matrix is applied over a Binary ECOC, the CSE coding algorithm is independent of the initial coding matrix M, and thus it could be applied to extend any coding design.

The confusion matrix C of Figure 3 has it maximum confusion value at $C_{4,5}$ (circle and square classes). Therefore, in the first iteration, an extension to split those classes and increment its separability will be coded. For this example, let the user-defined value of ϱ be 3. Thus, the length k of the Extension matrix at the current iteration E is $k = \varrho - S_{4,5}$. To increment the distance $\delta(y_4, y_5)$ to ϱ, we have to find two codes $\{y_4^E, y_5^E\}$ so that $\delta(y_4^E, y_5^E) = k$. In fact, the algorithm generates the 2^k codes κ times until N codewords are generated, and then, searches for two codes at $\delta = k$. Once this codes are defined in E, the algorithm looks for all the classes with $\min S_{i,j}$, $i \in \{4,5\}$, $j \in \{1,\dots,N\}$ and $\max C_{i,j}$, $i \in \{4,5\}$, $j \in \{1,\dots,N\}$ in order to increment its distance δ. If no confusion positions are found and the codes in E are left empty, then the algorithm applies $\min(S_{i,j})$, $\forall i,j \in \{1,\dots,N\}$ and $\max(C_{i,j})$, $\forall i,j \in \{1,\dots,N\}$.

Once the Extension matrix E is coded the algorithm checks if $E \cap M = \emptyset$ column-wise, if not, then the codeword corresponding to the class with $\min C_{i,j}$, $i \in \{4,5\}$, $j \in \{1,\dots,N\}$ and an opposite with $\min S_{i,j}$, $i \in \{4,5\}$, $j \in \{1,\dots,N\}$ are interchanged, and E is checked again. When E is completely coded and checked, M, S, and C are updated. That means that for the next iteration $M = M \cup E$. In addition, S is re-estimated with the new M.

3.1 Training the base classifiers

In [RK04] the author concludes that if the base classifier is properly tuned, the *One vs. All* may be used without loss of generalization capability. Following this idea, our coding design is upper bounded by N classifiers and thus, we need to use powerful dichotomizers in order to reduce possible misclassifications.

In literature, Support Vector Machines with a RBF-Gaussian kernel have demonstrated to be powerful dichotomizers. Nevertheless, they need some parameters to be optimized. In this case, parameters ζ, which is the regularizer, and γ, which has a relationship with the smoothness of the boundary, have to be optimized. A common way to optimize this parameters is to perform a grid search with cross-validation. Recently, in [BEB10] the authors have shown that Genetic Algorithms (GA) can be introduced in this optimization problem with good results.

For each binary problem, defined by a column of M, we use Genetic Algorithms in order to estimate values for ζ and γ. For this task, we use the same settings than in [LdC08], where individuals correspond to a pairs of genes, and each gene corresponds to the binary codification of a floating point value. This parameter estimation is performed under a 2-fold cross-validation measurement in order to avoid over-fitting bias and improve generalization.

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 multi-class problems extracted from the UCI Machine Learning Repository [AN07], showed in Table 1. 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 [MP]. Then, 50 classes of the ARFace database [MB98] are classified. Finally, we test our method in a real traffic sign categorization problem consisting of 36 traffic sign classes [CMP$^+$04].

Table 1. 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 [ASS02] using N classifiers. The ECOC base classifier is the libsvm implementation of a SVM with Radial Basis Function kernel [CC01a]. 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. Furthermore, the same experiments were run with Real AdaBoost as base classifier [FS95].
- **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 2 and Table 3, with SVM an Adaboost as base classifier, respectively. 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.

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 focuses the correcting capability in those classes more prone to be confused, and thus, less redundancy is needed. However, one has to notice that if designing a coding matrix with $n = N$ classifiers, Dense Random coding seems to be a suitable choice that also outperforms the standard *One vs. All* coding.

Nevertheless, when comparing Dense Random coding with our method in terms of performance, no statistical significant difference is found since both methods have a comparable rank. However, Dense Random coding seems to perform better than our proposal in the Computer Vision problems, where the number of classes is large. This situation is expected since Dense Random coding uses N dichotomies, and thus, it has a higher correction capability. In fact, we can approximate its correction capability by dividing the number of classes between the minimum number of classifiers needed to increase, at least, one unit the distance between codes ($\rho_{est} = \frac{N}{\lceil \log_2(N) \rceil}$). For example, in the MPEG7 experiment, the estimation of the correction capability of Dense Random coding tends to be$\rho_{est} = \frac{70}{\lceil \log_2(70) \rceil} = 10$. While for the CSE algorithm proposed with $\varrho = 5$ the estimated correcting capability is $\rho = 2$. Note however, that the number of classifiers used is approximately 1/4 of the number of classifiers used by *One vs. All* or Dense Random coding.

5 Conclusions

In this paper, we introduce the Separability matrix as a tool to enhance and analyse ECOC coding designs. Although separability issues have been always in the core of all ECOC coding proposals, until now there was no explicit need to define such a matrix. Nevertheless, in problem-dependent strategies and in sub-linear coding designs, it acquires great value since it shows which codes are prone to be confused due to the lack of correction capability. As a result, more precise and compact codes can be defined. Moreover, a novel ECOC coding design based on the Separability matrix is proposed

Results show that the proposed coding design obtains comparable or even better results than predefined compact coding designs using far less number of dichotomizers.

Future lines of work include, between others, to formalize a framework to analyse ECOC coding designs. In addition, comparatives with other compact problem-dependent coding designs might be performed.

Acknowledgments This work has been supported by projects TIN2009-14404-C02 and CONSOLIDER-INGENIO CSD 2007-00018.

Table 2. 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 3. UCI classification results with Real AdaBoost 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	40.6±1.3	11	44.7±0.8	10	46.5±1.2	10.6	**47.0±1.2**	11
Yeast	36.8±1.1	10	**45.6±0.4**	9.6	42.9±1.0	9.5	40.8±1.3	10
Ecoli	71.5±10.9	8	68.1±8.3	7.4	63.3±9.2	7.4	**75.0±7.8**	8
Glass	**53.8±12.1**	7	52.8±13.5	6	44.5±10.8	6	49.5±10.9	7
Segment	**96.4±0.7**	7	95.0±0.3	6.8	94.8±0.9	6.2	95.3±1.0	7
Derma	**89.3±4.9**	6	77.6±6.3	5.4	76.0±5.3	3	76.7±5.3	6
Vehicle	**73.6±1.3**	4	72.7±1.9	4	62.9±1.4	3	72.7±1.5	4
MPEG7	54.4±7.2	70	65.5±9.5	15	73.7±8.3	24.3	**86.5±6.4**	70
ARFaces	36.3±7.2	50	53.8±5.2	13.8	62.8±8.3	20.4	**81.5±6.3**	50
Traffic	80.6±6.2	36	81.3±8.1	12.2	87.4±7.9	20.6	**91.2±5.3**	36
Rank & #	2.6	20.8	2.4	**9.16**	3.0	10.89	**1.9**	20.8

References

[AN07] A. Asuncion and D.J. Newman. UCI machine learning repository. http://www.ics.uci.edu/∼mlearn/MLRepository.html, 2007. University of California, Irvine, School of Information and Computer Sciences.

[ASS02] E. Allwein, R. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. In *JMLR*, volume 1, pages 113–141, 2002.

[CMP+04] 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.

[CS02] K. Crammer and Y. Singer. On the learnability and design of output codes for multi-class problems. In *Machine Learning*, volume 47, pages 201–233, 2002.

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

[Dem06] J. Demsar. Statistical comparisons of classifiers over multiple data sets. *JMLR*, 7:1–30, 2006.

[DK95] T. Dietterich and E. Kong. Error-correcting output codes corrects bias and variance. In ICML, editor, *S. Prieditis and S. Russell*, pages 313–321, 1995.

[Hol75] J.H. Holland. *Adaptation in natural and artificial systems: An analysis with applications to biology, control, and artificial intelligence.* University of Michigan Press, 1975.

[BEB10] M.A. Bautista, S. Escalera, and X. Baro. Compact Evolutive Design of Error-Correcting Output Codes. *Supervised and Unsupervised Ensemble methods and applications - European Conference on Machine Learning*, 119–128, 2010.

[LdC08] Ana Carolina Lorena and Andr C.P.L.F. de Carvalho. Evolutionary tuning of svm parameter values in multiclass problems. *Neurocomputing*, 71(16-18):3326 – 3334, 2008.

[MB98] A. Martinez and R. Benavente. The AR face database. In *Computer Vision Center Technical Report #24*, 1998.

[MP] http://www.cis.temple.edu/latecki/research.html.

[PRV06] O. Pujol, P. Radeva, and J. Vitrià. Discriminant ECOC: A heuristic method for application dependent design of error correcting output codes. In *Trans. on PAMI*, volume 28, pages 1001–1007, 2006.

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

[TR98] T.Hastie and R.Tibshirani. Classification by pairwise grouping. *NIPS*, 26:451–471, 1998.

[EOR08] S. Escalera and O. Pujol and P. Radeva. Sub-class error-correcting output codes. In Proceedings of the 6th international conference on Computer vision systems, pages 494–504, 2008.

[BEV09] Baro, X.; Escalera, S.; Vitria, J.; Pujol, O.; Radeva, P.; , Traffic Sign Recognition Using Evolutionary Adaboost Detection and Forest-ECOC Classification Intelligent Transportation Systems, IEEE Transactions on , vol.10, no.1, pp.113-126, March 2009

[BGV92] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik A training algorithm for optimal margin classifiers In Proceedings of the fifth annual workshop on Computational learning theory (COLT '92). ACM, New York, NY, USA,144-152,1992.

[FS95] Yoav Freund and Robert E. Schapire A decision-theoretic generalization of on-line learning and an application to boosting In Proceedings of the Second European Conference on Computational Learning Theory, London, UK, 23-37, 1995.

[CC01a] Chih-Chung Chang and Chih-Jen Lin LIBSVM: a library for support vector machines `http://www.csie.ntu.edu.tw/~cjlin/libsvm`