Pursuing the Best ECOC Dimension for Multiclass Problems

Edgar Pimenta and **João Gama** *
LIACC, FEP - University of Porto;
Rua de Ceuta, 118-6; 4050 Porto, Portugal

André Carvalho
Dept. of Computer Science, ICMC-USP,
CP 668, CEP 13560-970, S. Carlos, SP, Brazil

Abstract

Recent work highlights advantages in decomposing multiclass decision problems into multiple binary problems. Several strategies have been proposed for this decomposition. The most frequently investigated are *All-vs-All*, *One-vs-All* and the *Error correction output codes* (ECOC). ECOC are binary words (codewords) and can be adapted to be used in classifications problems. They must, however, comply with some specific constraints. The codewords can have several dimensions for each number of classes to be represented. These dimensions grow exponentially with the number of classes of the multiclass problem. Two methods to choose the dimension of a ECOC, which assure a good trade-off between redundancy and error correction capacity, are proposed in this paper. The methods are evaluated in a set of benchmark classification problems. Experimental results show that they are competitive against conventional multiclass decomposition methods.

Introduction

Several Machine Learning (ML) techniques can only induce classifiers for 2-class problems. However, there are many real classification problems where the number of classes is larger than two. These problems are known as multiclass classification problems, here named multiclass problems. Two approaches have been followed in the literature to deal with multiclass problems using binary classifiers. In the first approach, the classification algorithm is internally adapted, by the modification of part of its internal operations. The second and most usual approach is the decomposition of the multiclass problem into a set of 2-class classification problems. This paper covers the second approach.

The investigation of techniques able to decompose multiclass problems into multiple binary problems is attracting growing attention. Most of the proposed approaches consist of two phases: a decomposition phase, which occurs before learning, and a reconstruction phase, which takes place after the binary predictions. These phases can be formalized as follows. Suppose a decision problem $D = \{\vec{x}, y\}^n$, where $\vec{x}$ is an input instance, y is its class label and n is the number of training instances. Assume further that $y \in \{y_1, ..., y_k\}$ where k ($k > 2$) is the number of classes. The decomposition phase consists in obtaining multiple binary problems in the form $B_i = \{\vec{x}, y'\}^n$, where $y' \in \{0, 1\}$. After the binary problems are defined, a learning algorithm induces a decision model for each problem B_i. Afterward, these decision models are used to classify test examples into one of two classes, producing a binary output value. In the reconstruction phase, the set of predictions carried out by the models define a binary vector, which is decoded into one of the k original classes. Many strategies have been proposed in the literature to deal with multiclass problems through binary classifiers. Among them, the most known are *One-vs-All* (OVA), where k binary problems are generated, each discriminating one class from the remaining classes, *All-vs-All* (AVA), where $k(k-1)/2$ binary problems are produced, discriminating between all pairs of classes and *Error-correcting-output codes (ECOC)*, where the classes are encoded by binary vectors. This paper will focus on ECOC.

There are many advantages in the decomposition-based approach for dealing with multiclass problems. First, several classification algorithms are restricted to two-class problems: e.g. Perceptron, Support Vector Machines (SVMs), etc. Even algorithms able to process multiclass problems, here named multiclass algorithms, contain internal procedures based on two classes. (e.g.: the twoing rule and the subset splitting of nominal attributes in CART (Breiman *et al.* 1984), etc.). Second, multiclass algorithms have difficulty incorporating misclassification costs. Different misclassifications may have different costs. A possible method to incorporate misclassification cost sensitivity is the employment of stratification. However, this method is only efficient when applied to binary problems. Third, it is easier to implement decomposition-based methods in parallel machines. Fourth, recent work (Furnkranz 2002) has shown that the AVA decomposition shows accuracy gains even when applied to a multiclass algorithm, which is confirmed by experiments carried out in this paper. Finally, since decomposition of multiclass problems is performed before learning, it can be applied to any learning algorithm. For instance, it has been applied to algorithms like Ripper (Furnkranz 2002), C4.5 (Quinlan 1993), CART (Breiman *et al.* 1984), and SVMs (Hsu & Lin 2002).

*Thanks to project ALES II (POSI/EIA/55340/2004), CNPq (200650/2005-0) and the FEDER Plurianual support attributed to LIACC.

The decomposition method investigated, ECOC, employs a distributed output code to encode the k classes of a multiclass problem (Dietterich & Bakiri 1995). For such, a codeword of length e is assigned to each class. However, the number of bits for each codeword is usually larger than necessary to represent each class uniquely. The main contribution of this work is to propose a new strategy to define the best ECOC dimension for a given multiclass problem. After describing alternatives investigated in the literature, we will show that the proposed strategy maximizes the trade-off between redundancy and error correction capacity.

The paper is organized as follows. The next Section describes ML methods previously investigated for the decomposition of multiclass problems into multiple binary problems, including the original ECOC. The new methods proposed here to select ECOC dimension are introduced in third Section. The fourth Section presents experimental evaluation of these methods for benchmark multiclass problems. The conclusions and future works are presented in the last section.

Related Work

We have already referred to the decomposition and reconstruction phases employed to deal with multiclass problems using a set of binary classifiers. In the following subsections, we summarize the methods most frequently employed for each of these phases.

Decomposition Phase. The decomposition phase divides a multiclass problem into several binary problems. Each binary problem defines a training set, which is employed by a learning algorithm to induce a binary decision model. To illustrate the methods employed for this phase, suppose a decision problem with k classes ($k > 2$). The most frequent methods in the literature are:

- *AVA*: This method, employed in (Hastie & Tibshirani 1998; Moreira 2000; Furnkranz 2002), generates a binary problem for each pair of original classes. The number of binary problems created is $k(k-1)/2$. For each decision problem, only the examples with the two corresponding class labels are considered. Therefore, each binary problem contains fewer examples than the original problem.
- *OVA*: This method, described in (Cortes & Vapnik 1995; Rifkin & Klautau 2004), produces k binary problems. Each problem discriminates one class from all the others. All the available training examples appear in all the binary problems.
- *Nested dichotomies*: The k classes are grouped into 2 groups. Each group is recursively divided into two smaller groups, till a group containing only one class is obtained (Moreira 2000).
- *ECOC*: Studied in (Dietterich & Bakiri 1995; Klautau, Jevti, & Orlitsky 2003), this method will be described in detail in the next subsection.

Error Correcting Output Codes Transmission of information through a noisy channel can involve information loss. With the development of the digital technology, the receiver of a message can detect and correct transmission errors. The basic idea is: instead of transmitting the message in its original form, the message is previously encoded. The encoding involves the introduction of some redundancy. The codified message is sent through a noisy channel that can change the message (errors). After receiving the message, the receiver decodes it. Due to the redundancy included, eventual errors might be detected and corrected. This is the context where the ECOC appeared. In 1948, Claude Shannon (Shannon 1948) showed how ECOC could be used to obtain a good trade-off between redundancy and capacity to recover from errors. Later, Hamming (Hamming 1950) presented the Hamming matrix, developed to codify 4 bits of information using 7 bits: $\mathbf{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$.
The function employed to encode a 4 bits message into 7 bits is: $C(d_1, d_2, d_3, d_4) = (d_1 + d_2 + d_4, d_1 + d_3 + d_4, d_1, d_2 + d_3 + d_4, d_2, d_3, d_4)$. For instance, the message 1001 would be encoded as 0011001. Suppose that the message $m = 0010001$ was received. The Hamming matrix is able to inform if there is an error and where it occurred. If there is no error, $H \oplus m = 0$. Otherwise, we can determine where the error occurred. By rotating $\mathbf{H} \oplus \mathbf{m} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ clockwise, we get the binary number 100, indicating the fourth bit as the error source.

In the context of classification, each class is codified into a binary *codeword* of size e and all codewords must be different. Since each column defines a binary learning problem, the number of binary decision models is e. However, the Hamming matrix cannot be directly used in classification problems: the last column will not define a decision problem, the first and sixth columns are complementary (correspond to the same decision problem), etc. In Section 3 we define the desirable ECOC properties for decomposing multiclass problems and propose new methods for their decoding design.

Reconstruction Phase After decomposition, a learning algorithm generates a decision model for each binary problem. When a test example is presented, each model makes a prediction. In the reconstruction phase, the set of predictions is combined to select one of the classes. Different approaches can be followed for this combination: *i) Direct-voting*. Applied to the AVA decomposition, it counts how many times each class was predicted by the binary classifiers. The most voted class is selected. *ii) Distributed-voting*. Usually applied to the OVA decomposition. If one class is predicted, this class receives 1 vote. If more than one class is predicted, each one of them receives $1/(k-1)$ votes, where k is the total number of classes. The most voted class becomes the predicted class. *iii) Hamming Distance*. Employed by ECOCs, it is based on the distance between binary vectors. The Hamming distance between two codewords cw_1 and cw_2 of size e is defined by: $hd(cw_1, cw_2) = \sum_{i=1}^{e}(|cw_{1,i} - cw_{2,i}|)$. The predicted class has the closest codeword to the vector of predictions made by the binary models.

For all these methods there are probabilistic variants, whose details can be seen in (Hastie & Tibshirani 1998).

Discussion In this section, we discuss the main advantages and disadvantages of each multiclass decomposition strategy when applied to a multiclass problem (k classes, $k > 2$): *i)* AVA: The number of decision problems is $k(k-1)/2$. When the number of classes is increased, the number of binary problems increases in quadratic order, while the number of examples per problem decreases. Moreover, most of these binary problems are irrelevant. For instance, suppose a problem with 10 classes. From the 45 binary problems, only 9 can correctly classify a test example (those comparing the correct class against one of the remaining classes). The other 36 problems would certainly wrongly classify this example. Therefore, $(k-1)(k-2)/2$ binary classifiers would misclassify any example and only $k-1$ could provide the correct classification. *ii)* OVA: This decomposition employs k binary decision problems. All the available training examples appear in all the binary problems. The number of binary problems grows linearly with the number of classes. Since each class has to face all the others, if one class has much fewer examples than the others (unbalanced), the prediction model may be weak. Therefore, we expect this method to perform better with balanced class distribution. *iii)* ECOC: An advantage of ECOC is that the number of binary decision problems depends on the size of the codeword. The size of the codeword is at least $\lceil log_2(k) \rceil$ and at most $2^{k-1}-1$. Although we can control the size of the codeword, the number of possible sizes grows exponentially with the number of classes. The number of examples in each binary problem is the same as in the original dataset.

By requiring smaller training sets, the classification models created for the AVA method generally are, individually, faster to train. On the other hand, by creating fewer binary problems (and consequently classifiers) than the AVA method, the OVA method usually presents memory and test time gains. Both these methods can be represented by ECOC using proper codewords.

Decomposition Using ECOC

Before ECOC can be applied to classification problems, a set of conditions should be satisfied: *i)* Maximize the minimum Hamming distance (d$_{min}$) between codewords; *ii)* Avoid equal (and complementary) columns, because they would generate equivalent decision models; *iii)* Do not allow any constant column (only 0 or 1), because it would not generate a decision problem.

It must be observed that compliance with these conditions does not avoid the generation of infeasible ECOCs. The design of a feasible ECOC for classification problems is an open issue. It is not addressed in this paper. This paper is concerned only with the selection of the best codeword dimension for an ECOC in a multiclass problem. The methods presented here for this selection independent of the strategy used to create the ECOC. The decomposition of a k-classes multiclass problem by ECOC produce between $\lceil log_2(k) \rceil$ and $2^{k-1}-1$ binary problems. Like the maximum dimension, the number of possible dimensions increases exponentially with k. In this paper, we propose alternatives to reduce this number.

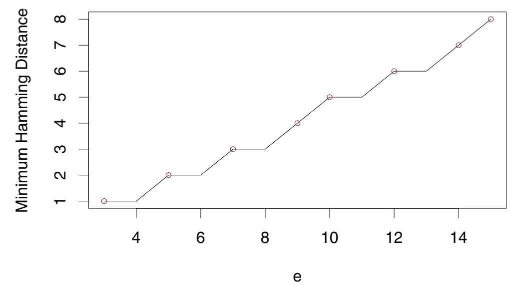

Figure 1: Support Function for k=5.

The Hamming distance selection. For a given value of k, it is possible to have several dimensions with the same Hamming distance. Since a lower dimension implies a smaller number of binary classifiers, the ECOC with the lowest dimension should be used. The highest possible d$_{min}$ for the minimum size ECOC (ECOC$_{min}$) and for the maximum size ECOC (ECOC$_{max}$) are 1 and 2^{k-2}, respectively. These values define two points, which can be used to define a straight line $y = m.e + b$, where $m = \frac{2^{k-2}-1}{(2^{k-1}-1)-\lceil log_2(k) \rceil}$ and $b = 1 - m\lceil log_2(k) \rceil$. We name this line y a *support line*, because it allows us to know the best d$_{min}$ for particular values of k and e. Since the Hamming distance is always an integer, we rounded down $y(k,e)$ to define the support function $s(k,e) = \lfloor \frac{2^{k-2}-1}{(2^{k-1}-1)-\lceil log_2(k) \rceil}(e - \lceil log_2(k) \rceil) + 1 \rfloor$. The support function $s(k,e)$ for k=5 is represented in Fig 1. The points represented by a dot should be preferred, since they have the lowest dimension among the points with the same Hamming distance. For instance, since the best Hamming distance for e=11 is equal to the best Hamming distance for e=10, one should use the lowest value (e=10). In this case ($k = 5$), we reduce the number of possible dimensions from 13 to 8. This method associated with the increase of k can reduce the number of possible dimensions for the ECOC by nearly 50%.

Maximum error correction selection. The maximum number of errors (me) that can be corrected for a certain Hamming distance is given by the expression: $me = \lfloor \frac{d_{min}(p,w_j)-1}{2} \rfloor$. This expression allows us to know in advance the number of errors that a certain ECOC is able to *correct*. Considering that a wrong classification is an error, the value given by me is the number of binary prediction models that can return a wrong prediction, provided that the final prediction will still be correct. There are several

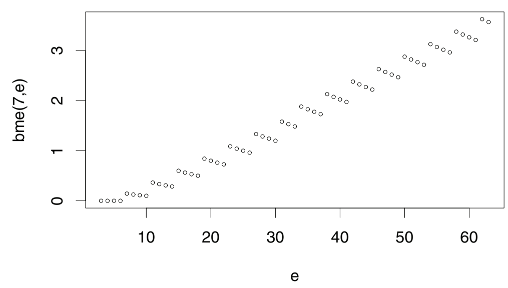

Figure 2: Maximum Error Correction bme for k=7.

Hamming distances with the same value for the maximum number of errors $me(d_{min}, e_i) = me(d_{min}, e_j), e_i \neq e_j$. The Hamming distance depends on the value of e. At the same time, there are many ECOC dimensions resulting on the same number of maximum errors that can be corrected. Therefore, we should select the lowest dimension, since it requires the creation of a smaller number of prediction models.

Evaluation function. As stated earlier, when using ECOC, we look for a trade-off between redundancy (to be minimized) and error correction capacity (to be maximized). More important than simply increasing the dimension e is to increase the d_{min} and, consequently, the me. For each value of me, there are several possible dimensions. We want the solution that presents the lowest dimension for the same me. Thus, we defined the function $bme(k, e) = me^2/e$, which is represented in Fig 2 for k=7. This function will be a continuously growing function, since each local maximum has a value higher than the previous one. In this function, the difference between the local maxima is almost constant. However, we also want to penalize the increase in the dimension e of the ECOC. Therefore, we created an evaluation function, defined as: $eval(k, e) = \frac{bme(k,e)}{k}$. The local maxima of this function have a logarithm behavior, as shown in Fig 3 for k=7. This function allows the reduction of the number of possible dimension to approximately a quarter of the initial value. However, there are still many dimensions to choose from. We will present two methods to select a dimension based on the evaluation function.

Tangent selection. The evaluation function can be seen as a function that has a benefit ($eval(k, e)$) with a cost (the dimension e of the ECOC). An increase in the benefit implies an increase in the cost. We want to find the point that has a good trade-off between the benefit and the cost. A possible approach to find a suitable cost/benefit balance is to select in the evaluation function the point whose tangent (derivative) is equal to 1 (angle of $45°$). Since our function is discrete, we can use this approach by selecting 3 consecutive local maxima and creating 2 straight lines: one connecting the first 2 points and another connecting the last 2 points.

The tangent we are looking for is the perpendicular to the bisector of the angle defined by these two lines. Other angles can be used. For example, to get a dimension more biased toward the reduction of e, we can use a $60°$ angle between $b1$ and $t1$. On other hand, to favor a better $aval(k, e)$, a $30°$ angle can be used. Thus, this method is very flexible. Table 1 ($ECOC_{tang}$) presents the dimension resulting from this method for several numbers of classes.

Classes	OVA	AVA	$ECOC_{min}$	$ECOC_{tang}$	$ECOC_{pa}$
5	5	10	3	10	8
10	10	45	4	64	33
15	15	105	4	360	33
20	20	190	5	2289	45

Table 1: Number of binary problems for several Decomposition Methods: OVA, AVA, and ECOCs (Minimum, Tangent, and Pareto).

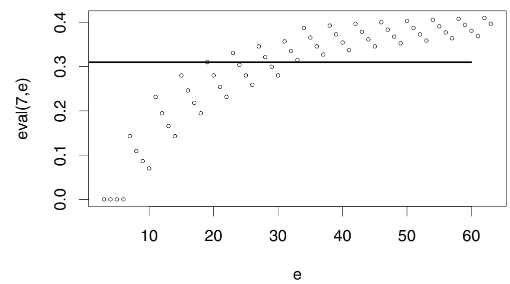

Figure 3: Evaluation function for $k = 7$ and the threshold defined by the Pareto principle.

Pareto selection. We applied the Pareto principle [1] to our evaluation function. Our assumption is that the first 80% values of the evaluation function are too low to generate good prediction results. Therefore, we decided to focus on the highest 20% (Fig 3). This will be the area where the evaluation function has good values, i. e., has good capacity to correct errors. At the same time, we want to reduce the dimension of the ECOC to the lowest possible. For such, we look for the local maximum with the lowest of those dimensions for which the evaluation functions is in the top 20% values. Table 1 ($ECOC_{pa}$) represents the dimensions obtained by this method for different values of k.

Discussion. With the increase in the number of classes k, the number of possible binary problems using ECOC increases rapidly. The tangent selection method can be used to select a suitable dimension. However, with the increase

[1]This principle states that few is vital (20%) and many are trivial (80%).

of k, this dimension will be much higher than those of the OVA and AVA methods (Table 1). By using the dimension given by the Pareto method, we get less binary problems than AVA and this difference increases with k. Thus, the adaptation of the Pareto method to the evaluation function for the design of ECOC allows the decomposition of a multiclass problem with a large number of classes into a reasonable number of binary problems, being a good alternative to the AVA method.

Experimental Evaluation

Experiments were carried out using a set of benchmark problems to compare the results obtained by a classification algorithm using the original multiclass problem and the previously discussed decomposition methods: AVA, OVA and ECOC (for several dimensions). For these comparisons, we selected 6 datasets from the UCI repository. All datasets are multiclass problems. The number of classes varies from 4 to 10. We employed the 10-fold cross validation evaluation procedure. The C4.5 and SVMs learning algorithms, as implemented in R, were used. We should observe that C4.5, in opposite to SVM, can directly process multiclass problems.

We evaluated ECOCs of several dimensions. $ECOC_{min}$ is the ECOC with minimum possible size, $ECOC_{oa}$ is the ECOC with the same size of the OVA method, $ECOC_{aa}$ is the ECOC with the size used by the AVA strategy, $ECOC_{tang}$ is the ECOC with the size given by the tangent selection criterion defined in section and $ECOC_{pa}$ is the ECOC whose size is given by the Pareto method, presented in section . The ECOCs used in these experiments were created using the persecution algorithm (Pimenta & J.Gama 2005).

Error rates were compared using the Wilcoxon test with a confidence level of 95%. In the case of C4.5, the reference for comparison is the default mode of C4.5 (multiclass). In the case of SVMs, the reference for comparisons is the AVA method, following the suggestion in (Rifkin & Klautau 2004) and because this is the default method in the implementation used. The experimental results are illustrated in Tables 2 and 3. For each dataset, we present two rows of values. The first row shows the mean of the percentage of correct predictions and the standard deviation. The second row has the number of binary problems created [2]. A positive (negative) sign before the accuracy value implies that the decomposition method was significantly better (worse) than the reference algorithm (Multiclass in Table 2 and AVA in Table 3) . For each dataset, the best results are presented in bold [3]. Table 2 shows the correct classification rates using the C4.5 algorithm.

The main conclusions derived from these results are:

- The ECOCs (except $ECOC_{min}$) and AVA decomposition methods usually improved the results obtained by the multiclass approach. The superiority of the AVA decomposition regarding the multiclass C4.5 confirm the results presented in (Furnkranz 2002);
- $ECOC_{tang}$ outperformed AVA, at the cost of producing a larger number of binary problems for larger values of k;
- $ECOC_{tang}$ outperformed the other ECOC variations (except for the $ECOC_{oa}$ in the Cleveland dataset), at the cost of requiring a larger number of classifiers for higher values of k;
- $ECOC_{pa}$ outperformed AVA in 3 of 4 datasets (those with the highest number of classes), requiring, at the same time, fewer binary problems.
- In the same 4 datasets, $ECOC_{tang}$ was significantly better than the multiclass approach (C4.5);
- The $ECOC_{oa}$ method exhibited better classification results than the OVA decomposition, using the same number of classifiers;
- Usually, the best accuracy rates were obtained by the $ECOC_{tang}$ decomposition. This method used the largest number of classifiers.

The results of the car dataset for the OVA method are very poor when compared with all other methods. We can attribute them to the distribution of the car dataset. This dataset has 4 classes with unbalanced distribution. This is a weakness of the OVA method, already referred to in the related work section. Table 3 presents the classification results obtained by the SVM algorithm.

The main conclusions from these results are:

- The OVA method presented the worst performance for all datasets, contradicting the results of (Rifkin & Klautau 2004);
- Overall, the ECOCs and AVA methods shown similar results. The ECOCs presented the best results in 3 datasets (only one statistically significant). AVA also presented the best results in 3 datasets (two statistically significant);
- Results produced by $ECOC_{pa}$ are competitive with those produced by AVA in 3 of 4 datasets (the datasets with the largest number of classes), while requiring fewer binary classifiers. The results using the OVA method for the car dataset were, again, very poor.

Conclusions and Future Work

This paper investigates the decomposition of multiclass classification problems into multiple binary problems using ECOC's. One of the main issues related to the use of ECOC is the definition of the codeword dimension. We introduced a new approach to reduce the number of possible dimensions for a problem with k classes to a quarter of the initial value - the evaluation function. Moreover, we presented two new solutions to select one from a set of possible dimensions: the tangent selection and the Pareto method. Experimental results using 6 UCI datasets and two learning algorithms (C4.5 and SVM) show that the proposed methods are very competitive when compared with standard decomposition methods (AVA, OVA) and with the direct multiclass approach (C4.5), which is a good indication of their potential.

[2] For the Multiclass column, it has the number of classes in the original problem.

[3] There are no results in $ECOC_{tang}$ for the Car and Cleveland datasets and in $ECOC_{pa}$ for the Cleveland dataset because there are too few local maxima in the evaluation function.

Dataset	Multiclass	OVA	AVA	$ECOC_{min}$	$ECOC_{oa}$	$ECOC_{aa}$	$ECOC_{tang}$	$ECOC_{pa}$
Car	93.1±2.0	−20.5±2.6	+**94.3±1.3**	−90.6±2.1	−90.4±2.9	−92.1±2.3	NA	NA
	4	4	6	2	4	6		
Cleveland	52.5±6.3	54.1±9.6	51.1±6.8	54.8±9.0	**55.1±8.2**	54.8±9.9	54.8±9.9	NA
	5	5	10	3	5	10	10	
Glass	55.3±13.3	49.5±11.6	+63.1±11.4	57.1±11.3	50.4±8.9	58.8±13.1	+**70.6±4.6**	52.7±12.9
	6	6	15	3	6	15	18	13
Satimage	86.7±1.6	−83.0±1.8	87.4±1.6	−82.7±1.1	85.5±1.3	+89.7±1.4	+**90.2±1.5**	+88.9±1.1
	6	6	15	3	6	15	18	13
Pendigits	96.3±0.7	−94.2±1.0	+96.5±0.6	−93.3±0.6	+97.4±0.7	+99.1±0.1	+**99.2±0.2**	+99.1±0.2
	10	10	45	4	10	45	64	33
Optidigits	90.1±1.2	−88.6±1.2	+94.8±0.9	−85.5±1.7	+92.3±1.4	+97.7±0.8	+**98.2±0.5**	+97.4±0.5
	10	10	45	4	10	45	64	33

Table 2: Comparison between multiclass, OVA, AVA and ECOCs with several dimensions using C4.5.

Dataset	OVA	AVA	$ECOC_{min}$	$ECOC_{oa}$	$ECOC_{aa}$	$ECOC_{tang}$	$ECOC_{pa}$
Car	−21.8±2.0	83.3±22.7	**86.1±3.4**	−21.2±2.0	84.4±3.7	NA	NA
	4	6	2	4	6		
Cleveland	54.1±9.6	**59.4±11.9**	56.8±7.9	58.4±9.8	57.8±9.0	57.8±9.0	NA
	5	10	3	5	10	10	
Glass	48.2±9.5	51.4±8.8	**55.2±13.7**	48.6±9.9	50.4±9.8	54.2±10.5	50.4±9.2
	6	15	3	6	15	18	13
Satimage	−86.5±1.1	**90.9±1.2**	−87.5±1.3	−88.8±1.4	−88.1±1.0	−89.0±1.3	88.1±0,9
	6	15	3	6	15	18	13
Pendigits	−99.1±0.4	**99.6±0.3**	−98.9±0.5	−99.3±0.3	−99.3±0.3	−99.4±0.3	99.3±0,4
	10	45	4	10	45	64	33
Optidigits	−97.0±0.5	98.5±0.5	−97.4±0.9	−98.0±0.6	+**99.3±0.3**	+98.5±0.3	98.5±0.5
	10	45	4	10	45	64	33

Table 3: Comparison between OVA, AVA and ECOCs with several dimensions using SVM.

References

Breiman, L.; Friedman, J.; Olshen, R.; and Stone, C. 1984. *Classification and Regression Trees*. Wadsworth International Group., USA.

Cortes, C., and Vapnik, V. 1995. Support-vector networks. *Machine Learning* 20(3):273–297.

Dietterich, T., and Bakiri, G. 1995. Solving multiclass learning problems via error-correcting output codes. *Journal Artificial Inteligence Research* Vol. 2, 263–286.

Furnkranz, J. 2002. Round Robin Classification. *Journal of Machine Learning Research* 2:721–747.

Hamming, R. 1950. Error-detecting and error-correcting codes. *Bell System Technical Journal* 29:147–160.

Hastie, T., and Tibshirani, R. 1998. Classification by pairwise coupling. *The Annals of Statistics* 26(2):451–471.

Hsu, C.-W., and Lin, C.-J. 2002. A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks* 13:415–425.

Klautau, A.; Jevti, N.; and Orlitsky, A. 2003. On nearest-neighbor ECOC with application to all-pairs multiclass SVM. *Journal of Machine Learning Research* 4(1):1–15.

Moreira, M. 2000. *The use of Boolean concepts in general classification contexts*. Ph.D, Polytechnique de Lausanne.

Pimenta, E., and J.Gama. 2005. A study on error correcting output codes. In *Proc. of 2005 Portuguese Conference on Artificial Intelligence*, 218–223.

Quinlan, R. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, Inc.

Rifkin, R., and Klautau, A. 2004. In defense of one-vs-all classification. *Journal Machine Learning Reasearch* 5

Shannon, C. E. 1948. A mathematical theory of communication. *Bell System Technical Journal* 27:379–423.