Journal of Machine Learning Research 4 (2003) 1-15 Submitted 10/02; Published 4/03

On Nearest-Neighbor Error-Correcting Output Codes with Application to All-Pairs Multiclass Support Vector Machines

Aldebaro Klautau A.KLAUTAU@IEEE.ORG
Nikola Jevtić NIKOLA@CWC.UCSD.EDU
Alon Orlitsky ALON@ECE.UCSD.EDU
ECE Department, UCSD
9500 Gilman Drive
La Jolla, CA 92093-0407, USA

Editor: Yoram Singer

Abstract

A common way of constructing a multiclass classifier is by combining the outputs of several binary ones, according to an error-correcting output code (ECOC) scheme. The combination is typically done via a simple nearest-neighbor rule that finds the class that is closest in some sense to the outputs of the binary classifiers. For these nearest-neighbor ECOCs, we improve existing bounds on the error rate of the multiclass classifier given the average binary distance. The new bounds provide insight into the one-versus-rest and all-pairs matrices, which are compared through experiments with standard datasets. The results also show why *elimination* (also known as DAGSVM) and Hamming decoding often achieve the same accuracy.

Keywords: Error-correcting output codes, all-pairs ECOC matrix, multiclass support vector machines

1. Introduction

Several techniques for constructing binary classifiers with good generalization capabilities were developed in recent years, e.g., support vector machines (SVM) (Cortes and Vapnik, 1995). However, in many applications the number of classes is larger than two. While multiclass versions of most classification algorithms exist (e.g., Crammer and Singer, 2002), they tend to be complex (Hsu and Lin, 2002). A more common approach is to construct the multiclass classifier by combining the outputs of several binary ones (Dietterich and Bakiri, 1995, Allwein et al., 2000). Typically, the combination is done via a simple nearest-neighbor rule, which finds the class that is closest in some sense to the outputs of the binary classifiers.

The most traditional scheme for solving a multiclass problem with binary classifiers is based on the so-called one-versus-rest matrix. However, the popularity of an alternative scheme based on the all-pairs matrix (also known as *1 versus 1*, *round-robin* and *pairwise decomposition*) has recently increased (see, e.g., Fürnkranz, 2002). All-pairs with Hamming decoding is related to well-known methods of paired comparisons in statistics (David, 1963), and it was first applied to classification problems by Friedman (1996).

There are theoretical results that compare some aspects of the all-pairs and one-versus-rest (among other) matrices. These results also suggest guidelines for constructing accurate multiclass classifiers. For example, recent work has used the error incurred by the binary classifiers to up-

per bound the error committed by the combined nearest-neighbor classifier (Guruswami and Sahai, 1999, Allwein et al., 2000). These results are reviewed and expanded here.

We present theoretical and experimental contributions. We strengthen the bounds by Allwein et al. (2000) and extend the class of decoders to which they apply. These improved bounds provide insight into the properties of certain ECOC matrices when the number of classes is large. We also conduct detailed experiments directly comparing ECOC schemes that use the all-pairs and one-versus-rest matrices for solving multiclass problems with SVM, complementing previous work (e.g., Allwein et al., 2000, Hsu and Lin, 2002). Our results show that Hamming decoding is very effective for all-pairs. Additionally, our experimental results explain why *elimination* (Kreßel, 1999) (also known as DAGSVM) and Hamming decoding often achieve similar accuracy.

The paper is organized as follows. Section 2 provides a brief review about the construction of multiclass classifiers from binary ones and establishes the notation. Theoretical bounds for the multiclass error are presented in Section 3. Experimental results are presented in Section 4, followed by conclusions in Section 5.

2. Background on ECOC

In supervised classification problems, one is given a *training set* $\{(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_N,y_N)\}$ containing *N examples*. Each example $(\mathbf{x},y)$ consists of an instance $\mathbf{x}\in X$ and a label $y\in\{1,\ldots,K\}$, where X is the *instance space* and $K\geq 2$ is the number of *classes*. A *classifier* is a mapping $F:X\to\{1,\ldots,K\}$ from instances to labels. For binary problems ($K=2$ classes) the examples are labeled -1 and $+1$, for convenience. We assume the base learner is *class-symmetric*, i.e., the learning problem is equivalent if we exchange the labels -1 and $+1$, and we are especially interested on *confidence-valued* binary classifiers $f:X\to\mathbb{R}$ that return a *score*.

One of the most successful methods for constructing multiclass classifiers is to combine the outputs of several binary classifiers. First, a collection $f_1,\ldots,f_B$ of *B binary classifiers* is constructed, where each classifier is trained to distinguish between two subsets of classes. The classes involved in the training of the binary classifiers are typically specified by a matrix $\mathbf{M}\in\{-1,1\}^{K\times B}$ (Dietterich and Bakiri, 1995) or $\mathbf{M}\in\{-1,0,1\}^{K\times B}$ (Allwein et al., 2000), and classifier f_b is trained according to column $\mathbf{M}(\cdot,b)$.

The K-ary classifier F takes the scores $f(\mathbf{x})=(f_1(\mathbf{x}),\ldots,f_B(\mathbf{x}))$ and combines them using a function $g:\mathbb{R}^B\to\{1,\ldots,K\}$ to obtain $F(\mathbf{x})=g(f(\mathbf{x}))$. One can view the rows of the matrix $\mathbf{M}$ as codewords and the function g as decoding the output $f(\mathbf{x})$ of the binary classifiers. By analogy to coding,[1] $\mathbf{M}$ is refereed to as an *ECOC matrix* and the function g is called the *decoder*. We call the combination of a matrix $\mathbf{M}$ and a decoder g, an *ECOC scheme* or simply ECOC.

In spite of the appeal of matrices inspired by coding, the most popular ECOC matrices are obtained by simply taking all combinations of α versus (vs.) β classes, $\alpha+\beta\leq K$, where each binary classifier is trained to distinguish α positive from β negative classes. We will be specially interested in the 1 vs. 1 (*all-pairs*) and 1 vs. $K-1$ (*one-vs-rest*) matrices. The one-vs-rest matrix induces $B=K$ binary classifiers $f_1,\ldots,f_K$. The all-pairs matrix induces $B=\binom{K}{2}$ binary classifiers $f_{i,j},1\leq i<j\leq K$.

1. Many results from coding can be promptly used for ECOCs, specially when $\mathbf{M}\in\{-1,+1\}^{K\times B}$. For example, Theorem 2 by Berger (1999) corresponds to the well-known Plotkin's bound, which states that $\rho\leq(0.5BK)/(K-1)$, where ρ is the minimum Hamming distance between two distinct rows of $\mathbf{M}$.

ECOC schemes often adopt a *nearest-neighbor* decoder. These decoders use a *distance* measure $d : \mathbb{R}^B \times \{-1,0,1\}^B \to \mathbb{R}$, and select the class $F(\mathbf{x}) = \arg\min_k d(f(\mathbf{x}), \mathbf{M}(k,\cdot))$ that minimizes the distance between scores $f(\mathbf{x})$ and row $\mathbf{M}(k,\cdot)$. Of special interest are *loss-based* distances (Allwein et al., 2000), which are defined by

$$d(f(\mathbf{x}), \mathbf{M}(k,\cdot)) = \sum_{b=1}^{B} L(z_{k,b}), \tag{1}$$

where $L : \mathbb{R} \to \mathbb{R}$ is a *loss* function,[2] and $z_{k,b} = f_b(\mathbf{x})\mathbf{M}(k,b)$ would be the *margin* under classifier f_b if $\mathbf{M}(k,b)$ were the label of instance $\mathbf{x}$.

It is easy to show that, for any ECOC with loss-based decoding, all linear loss functions $L(z) = c_1 + c_2 z$ with negative (positive) c_2 lead to the same classification result. If the binary learner is an SVM, decoding with $L(z) = (1-z)_+$, where $(x)_+ = \max\{x, 0\}$, has the appeal of matching the criterion used to maximize the margin when training the SVMs (Allwein et al., 2000).

The natural decoding method for an ECOC with the one-vs-rest matrix is to select the class k that maximizes score $f_k(\mathbf{x})$. This decoder is called *max-wins*. It can be shown that for the one-vs-rest matrix, several choices of L lead to the same classification result as max-wins, such as $L(z) = (1-z)_+$ or when L is a strictly decreasing function.

Instead of scores, the binary classifiers may return *hard decisions* $h(\mathbf{x}) \in \{-1,1\}^B$, or the results of the binary classifiers may be quantized to $\{-1,1\}$ to overcome unreliability. A natural decoder in these cases is the *Hamming decoder* (also known as *voting*), the nearest-neighbor decoder that minimizes the *Hamming distance* (modified to allow for $\mathbf{M}(k,b) = 0$):

$$d_H(h(\mathbf{x}), \mathbf{M}(k,\cdot)) = 0.5 \sum_{b=1}^{B} (1 - h_b(\mathbf{x})\mathbf{M}(k,b)). \tag{2}$$

Hamming distance is a special case of a loss-based distance where $L(z) = (1 - \text{sign}(z))/2$ (Allwein et al., 2000).

We are mostly concerned with nearest-neighbor decoders, for which theoretical results are presented in Section 3. In general however, the decoder g can be any mapping, such as the one obtained with a stacked artificial neural network (Klautau et al., 2002). Another example of a non-nearest-neighbor decoder is the one proposed by Moreira and Mayoraz (1998). They adopted the all-pairs matrix and a decoding method equivalent to using Equation (1) with $L(z') = -z'$, where z' is a weighted margin $z'_{k,b} = \omega_b f_b(\mathbf{x})\mathbf{M}(k,b)$. The weight ω_b of classifier f_b was obtained with an additional ECOC based on a 2 vs. $K-2$ matrix.

3. Bounds on the K-ary Error

Previous work (Guruswami and Sahai, 1999, Allwein et al., 2000) used the error, and more generally, distance, incurred by the binary classifiers, to upper bound the error committed by the K-ary classifier with nearest-neighbor decoding. This section strengthens these bounds, extends the distance measures to which they apply and provides some insight into the properties of α vs. β ECOC matrices when K is large. We begin by discussing results for any distance d. Then we specialize the bounds for the cases where d is the Hamming distance, and later for ECOCs with Hamming distance and α vs. β matrix.

2. Allwein et al. (2000) defined $L : \mathbb{R} \to [0,\infty)$, but here we extend the range of L to allow for, e.g., $L(z) = -z$, used by Zadrozny (2002).

3.1 Bounds for Nearest-Neighbor with General Distance

The number of errors the K-ary classifier F commits on a given set (e.g., training or held-out set) with N examples is $\varepsilon_K \stackrel{\text{def}}{=} |\{n : F(\mathbf{x}_n) \neq y_n\}|$ and its *error rate* is

$$\overline{\varepsilon}_K \stackrel{\text{def}}{=} \frac{\varepsilon_K}{N}.$$

The accumulated distance between the outputs of the binary classifiers and the correct codeword over this set is $D \stackrel{\text{def}}{=} \sum_{n=1}^{N} d(f(\mathbf{x}_n), \mathbf{M}(y_n, \cdot))$ and their average distance is

$$\overline{D} \stackrel{\text{def}}{=} \frac{D}{N}.$$

To relate $\overline{\varepsilon}_K$ and $\overline{D}$, the minimum Hamming distance between any two rows was defined by Allwein et al. (2000) to be

$$\rho \stackrel{\text{def}}{=} \min_{k,k'} \{d_H(\mathbf{M}(k,\cdot), \mathbf{M}(k',\cdot)) : k \neq k'\},$$

where d_H is defined in Equation (2). For example, for one-vs-rest $\rho = 2$, and for all-pairs $\rho = (B+1)/2$. Allwein et al. (2000) also used

$$L^* \stackrel{\text{def}}{=} \min_{z \in \mathbb{R}} \left\{ \frac{L(z) + L(-z)}{2} \right\}$$

to prove some of their results.

Here we use the following two definitions related to distances between scores $f(\mathbf{x})$ and rows of $\mathbf{M}$. Given an ECOC matrix $\mathbf{M}$, a distance measure d, and a vector $\mathbf{f} \in \mathbb{R}^B$, let $d_1(\mathbf{f})$ be the smallest distance between $\mathbf{f}$ and any row of $\mathbf{M}$, and let $d_2(\mathbf{f}) \geq d_1(\mathbf{f})$ be the smallest distance between $\mathbf{f}$ and the remaining rows of $\mathbf{M}$. Define

$$d_1 = \min_{\mathbf{f} \in \mathbb{R}^B} d_1(\mathbf{f}) \quad \text{and} \quad d_2 = \min_{\mathbf{f} \in \mathbb{R}^B} d_2(\mathbf{f}).$$

For example, for an ECOC with the one-vs-rest matrix and Hamming decoding, $d_1 = 0$ (achieved when $f(\mathbf{x}) = h(\mathbf{x})$ matches a codeword) and $d_2 = 1$ (achieved when $f(\mathbf{x}) = h(\mathbf{x})$ contains two elements $+1$ while the others are -1). And for an ECOC with the all-pairs matrix and Hamming decoding, $d_1 = 0.5\binom{K-1}{2}$ and $d_2 = d_1 + 1$. The following result was originally presented by Allwein et al. (2000), and is restated here using d_2.

Lemma 1 (Implicit by Theorem 1 published by Allwein et al., 2000) For any ECOC with loss-based decoding using $L : \mathbb{R} \to [0, \infty)$,

$$d_2 \geq \rho L^*. \qquad \square$$

According to Lemma 1, whenever an example $(\mathbf{x}, y)$ leads to an error, the total error ε_K is incremented by 1 and at least ρL^* is added to the distance D. This reasoning can be used to interpret the bound

$$\overline{\varepsilon}_K \leq \frac{\overline{D}}{\rho L^*}, \tag{3}$$

which is proved by Allwein et al. (2000) for any ECOC with loss-based decoding using $L : \mathbb{R} \to [0, \infty)$. Using the definitions of d_1 and d_2, Equation (3) can be strengthened as follows.

Theorem 2 For any ECOC with nearest-neighbor decoding,

$$\overline{\varepsilon}_K \leq \frac{\overline{D} - d_1}{d_2 - d_1}.$$

Proof Split the set of instances into

$$C \stackrel{\text{def}}{=} \{(\mathbf{x}_n, y_n) : F(\mathbf{x}_n) = y_n\} \quad \text{and} \quad W \stackrel{\text{def}}{=} \{(\mathbf{x}_n, y_n) : F(\mathbf{x}_n) \neq y_n\},$$

containing the correctly and wrongly classified examples, respectively. The accumulated distance D can then be written as

$$D = \sum_{(\mathbf{x}_n, y_n) \in C} d(f(\mathbf{x}_n), \mathbf{M}(y_n, \cdot)) + \sum_{(\mathbf{x}_n, y_n) \in W} d(f(\mathbf{x}_n), \mathbf{M}(y_n, \cdot)).$$

The total number of errors is $\varepsilon_K = |W|$, hence the first part is at least $|C| d_1 = (N - \varepsilon_K) d_1$, and the second is at least $|W| d_2 = \varepsilon_K d_2$. Therefore $D \geq (N - \varepsilon_K) d_1 + \varepsilon_K d_2$. Normalizing by N and solving for $\overline{\varepsilon}_K$, we obtain the theorem. □

We note that Theorem 2 applies to all distance measures, not just the loss-based ones with $L(z) \geq 0$ and $L^* > 0$, as required for Equation (3). Also, for loss-based distances, and when Equation (3) is applicable and not trivial (i.e., $\overline{D} < \rho L^*$), Theorem 2 is always at least as strong as Equation (3) because $\overline{D} < \rho L^* \leq d_2$, and hence, $\forall d_1 \geq 0$,

$$\frac{\overline{D} - d_1}{d_2 - d_1} \leq \frac{\overline{D}}{d_2} \leq \frac{\overline{D}}{\rho L^*}.$$

A special case of interest is when the distance d obeys the triangle inequality. If d_{min} is the minimum distance between two rows of $\mathbf{M}$, by the triangle inequality $d_2 \geq d_{min}/2$. For example, for any ECOC with Hamming decoding ($d_{min} = \rho$) and a matrix $\mathbf{M}$ without zero entries ($d_1 = 0$), $\overline{\varepsilon}_K \leq 2\overline{D}/\rho$, because the Hamming distance obeys the triangle inequality. For Hamming decoding it is also possible to write $\overline{D}$ in terms of the binary error rate, as discussed in the next subsection.

3.2 Bounds for Hamming Decoding

For Hamming decoding, Allwein et al. (2000) presented a more natural form of Equation (3), which relates the K-ary classifier's error $\overline{\varepsilon}_K$ to that committed by the binary classifiers. Using Theorem 2, we strengthen this bound as well.

Let $T \stackrel{\text{def}}{=} \{(n,b) : \mathbf{M}(y_n, b) = 0\}$ be the set of pairs (n,b) corresponding to examples and binary classifiers not used when designing the K-ary classifier, and $T^c \stackrel{\text{def}}{=} \{(n,b) : \mathbf{M}(y_n, b) \neq 0\}$ be its complement. Clearly, $|T| + |T^c| = NB$. The number of examples misclassified by the binary classifiers is then $\varepsilon_b \stackrel{\text{def}}{=} |\{(n,b) \in T^c : h_b(\mathbf{x}_n) \neq \mathbf{M}(y_n, b)\}|$, and the *error rate* of the binary classifiers is

$$\overline{\varepsilon}_b \stackrel{\text{def}}{=} \frac{\varepsilon_b}{|T^c|}.$$

We need d_1 and d_2 to apply Theorem 2 for ECOCs with Hamming decoding. In this case, $d_1 = 0.5 B_0^{min}$, where B_0^{min} is the minimum number of zero entries in a row. In order to conveniently

express d_2, let $O_{k,k'} \stackrel{\text{def}}{=} \{b : \mathbf{M}(k,b) \neq 0 \wedge \mathbf{M}(k',b) \neq 0\}$ be the set of columns where both codewords $\mathbf{M}(k,\cdot)$ and $\mathbf{M}(k',\cdot)$ have non-zero elements. Assume a partial Hamming distance that takes in account only columns in $O_{k,k'}$ and let

$$\rho_1 \stackrel{\text{def}}{=} \min_{k,k'} \{0.5 \sum_{b \in O_{k,k'}} (1 - \mathbf{M}(k,b)\mathbf{M}(k',b)) : k \neq k'\}$$

be the minimum of such partial distances. Note that for ECOC matrices without zero entries, like the one-vs-rest, $\rho_1 = \rho$. The reason for using ρ_1 is to isolate the influence of zero entries in matrix $\mathbf{M}$. The following result can then be proved.

Lemma 3 For any ECOC with Hamming decoding,

$$d_2 \geq d_1 + \lceil \rho_1/2 \rceil.$$

Proof We are after the classifiers' output $\mathbf{h} \in \{-1,+1\}^B$ that minimizes $d_2(\mathbf{h})$. Let codewords $\mathbf{M}(r,\cdot)$ and $\mathbf{M}(s,\cdot)$ achieve $d_1(\mathbf{h})$ and $d_2(\mathbf{h})$, respectively. Define the following sets of columns: $S_{00} = \{b : \mathbf{M}(r,b) = 0 \wedge \mathbf{M}(s,b) = 0\}$, $S_{01} = \{b : (\mathbf{M}(r,b) = 0 \wedge \mathbf{M}(s,b) \neq 0)\}$ and $S_{10} = \{b : (\mathbf{M}(r,b) \neq 0 \wedge \mathbf{M}(s,b) = 0)\}$. For the entries corresponding to columns $b \in S_{00}$, $\mathbf{h}$ can assume any value, and for $b \in \{S_{01} \cup S_{10}\}$, $\mathbf{h}$ will match the non-zero entry. Hence, $\forall \mathbf{h}$,

$$d_1(\mathbf{h}) + d_2(\mathbf{h}) \geq |S_{00}| + 0.5(|S_{01}| + |S_{10}|) + \rho_1.$$

The distance $d_1(\mathbf{h})$ cannot be larger than $d_2(\mathbf{h})$, so

$$2d_2(\mathbf{h}) \geq d_1(\mathbf{h}) + d_2(\mathbf{h}) \geq B_0^{min} + \rho_1 = 2d_1 + \rho_1.$$

To properly take in account the case where ρ_1 is odd,

$$d_2(\mathbf{h}) \geq d_1 + \lceil \rho_1/2 \rceil. \qquad \square$$

Let $\overline{B_1} = |T^c|/N$ be the average number of non-zero entries in each codeword. Applying Theorem 2 leads to the following result.

Lemma 4 For any ECOC with Hamming decoding,

$$\overline{\varepsilon}_K \leq \frac{0.5(B - B_0^{min} - \overline{B_1}) + \overline{B_1}\overline{\varepsilon}_b}{\lceil \rho_1/2 \rceil}.$$

Proof For Hamming decoding,

$$D = 0.5|T| + \varepsilon_b = 0.5(NB - |T^c|) + |T^c|\overline{\varepsilon}_b.$$

So, $\overline{D} = 0.5(B - \overline{B_1}) + \overline{B_1}\overline{\varepsilon}_b$. From Theorem 2,

$$\begin{aligned} \overline{\varepsilon}_K &\leq \frac{0.5(B - \overline{B_1}) + \overline{B_1}\overline{\varepsilon}_b - d_1}{d_2 - d_1} \\ &\leq \frac{0.5(B - B_0^{min} - \overline{B_1}) + \overline{B_1}\overline{\varepsilon}_b}{\lceil \rho_1/2 \rceil}, \end{aligned}$$

where the last step follows from Lemma 3. $\square$

3.3 Bounds for Hamming Decoding and α vs. β Matrix

The bound in Lemma 4 becomes simpler when applied to ECOCs with α vs. β matrices. For these matrices the number B of binary classifiers is $B_0 + B_1$, where $B_0 = B_0^{min}$ and $B_1 = \overline{B_1}$ are the number of zero and non-zero elements in each row, respectively. Applying Lemma 4 leads to

$$\overline{\varepsilon}_K \leq \frac{B_1}{\lceil \rho_1/2 \rceil} \overline{\varepsilon}_b. \tag{4}$$

For example, for one-vs-rest, Equation (4) implies

$$\overline{\varepsilon}_K \leq K \overline{\varepsilon}_b, \tag{5}$$

which was originally presented by Guruswami and Sahai (1999). And for all-pairs

$$\overline{\varepsilon}_K \leq (K-1) \overline{\varepsilon}_b. \tag{6}$$

We note that for α vs. β matrices it can be proved that d_2 achieves the lower-bound in Lemma 3, namely $d_2 = d_1 + \lceil \rho_1/2 \rceil$.

To apply Equation (4) for a general α vs. β matrix $\mathbf{M}$, it is convenient to have expressions for B_1 and ρ_1. These can be written in terms of B_1^* and ρ_1^*, which are parameters obtained from the *base* matrix $\mathbf{M}^*$ of $\mathbf{M}$, defined as follows.

We construct an α vs. β matrix $\mathbf{M}$ using matrices

$$\mathbf{M}^* \in \{-1,+1\}^{(\alpha+\beta)\times B^*} \quad \text{and} \quad \mathbf{P} \in \{0,1,\ldots,\alpha+\beta\}^{K \times \binom{K}{\alpha+\beta}}.$$

Given the values K, α and β of $\mathbf{M}$, the associated $\mathbf{M}^*$ is simply an α vs. β ECOC matrix with the same values $\alpha^* = \alpha$ and $\beta^* = \beta$, but with the number of classes $K^* = \alpha + \beta$. Clearly, if $\alpha + \beta = K$ (i.e., if there are no zero entries in $\mathbf{M}$), then $\mathbf{M}^* = \mathbf{M}$. The matrix $\mathbf{P}$ is used to expand $\mathbf{M}^*$ into $\mathbf{M}$, taking in account all $\binom{K}{\alpha+\beta}$ ways of choosing $\alpha+\beta$ of the K classes. Each entry $\mathbf{P}(m,n) = i$, $i \neq 0$, is replaced by the i-th row of $\mathbf{M}^*$. If $\mathbf{P}(m,n) = 0$, the entry is substituted by B^* zeros. For example, for a 1 vs. 2 ECOC matrix with $K = 4$, the matrices $\mathbf{M}^*$, $\mathbf{P}$ and $\mathbf{M}$ are, respectively,

$$\begin{bmatrix} +1 & -1 & -1 \\ -1 & +1 & -1 \\ -1 & -1 & +1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 0 \\ 2 & 2 & 0 & 1 \\ 3 & 0 & 2 & 2 \\ 0 & 3 & 3 & 3 \end{bmatrix} \text{ and } \left[\begin{array}{ccc|ccc|ccc|ccc} +1 & -1 & -1 & +1 & -1 & -1 & +1 & -1 & -1 & 0 & 0 & 0 \\ -1 & +1 & -1 & -1 & +1 & -1 & 0 & 0 & 0 & +1 & -1 & -1 \\ -1 & -1 & +1 & 0 & 0 & 0 & -1 & +1 & -1 & -1 & +1 & -1 \\ 0 & 0 & 0 & -1 & -1 & +1 & -1 & -1 & +1 & -1 & -1 & +1 \end{array}\right].$$

The number of columns in the base matrix $\mathbf{M}^*$ is

$$B^* = \frac{1}{1 + I(\alpha = \beta)} \binom{\alpha+\beta}{\beta},$$

where the indicator function $I(\cdot)$ (which is 1 if its argument is true and zero otherwise) takes in account that, when $\alpha = \beta$, half of the $\binom{\alpha+\beta}{\beta}$ binary problems are effectively the same. And the minimum Hamming distance for $\mathbf{M}^*$ is

$$\rho^* = B^* - \binom{\alpha+\beta-2}{\beta-2},$$

where we assumed $\beta \geq \alpha$. For example, for all-pairs $\mathbf{M}^* = \begin{bmatrix} +1 \\ -1 \end{bmatrix}$, $B^* = 1$ and $\rho_1 = 1$. Having B^* and ρ^* for the base matrix $\mathbf{M}^*$, one can obtain

$$B_1 = \binom{K-1}{\alpha+\beta-1} B^* \quad \text{and} \quad \rho_1 = \binom{K-2}{\alpha+\beta-2} \rho^*,$$

which allow to use Equation (4) for any α vs. β ECOC.

Based on this result, we now briefly discuss the behavior of α vs. β ECOCs when the number K of classes is large. Allwein et al. (2000) stated Equation (3) as $\overline{\varepsilon}_K \leq \frac{B\xi}{\rho L^*}$, where $\xi = \overline{D}/B$ is the average distance per binary classifier (note that ξ does not explicitly take in account the influence of zero entries in $\mathbf{M}$). We note that for all-pairs, as K grows, the proportion B_0/B of zeros in each row goes to 1, and ξ goes to L^*, which makes the bound trivial. This can be easily seen for Hamming decoding. In this case,

$$\overline{\varepsilon}_K \leq \frac{B\xi}{\rho L^*} = \frac{2B\xi_H}{\rho},$$

where ξ_H is the average Hamming distance per binary classifier (Corollary 2 in Allwein et al., 2000). For all-pairs and large enough K, $\overline{\varepsilon}_K \leq 4\xi_H$, but also $\xi_H = L^* = 0.5$.

Alternatively, we can look at the behavior for large K of ECOCs with α vs. β and Hamming decoding using Equation (4), i.e., using $B_1/\lceil \rho_1/2 \rceil$. We note that, in spite of not being achieved by all-pairs, there are ECOCs with α vs. β matrix and Hamming decoding for which $B_1/\lceil \rho_1/2 \rceil \to 4$, as $K \to \infty$. When $\alpha = \beta = K/2$, Equation (4) leads to $\overline{\varepsilon}_K \leq 4(K-1)\overline{\varepsilon}_b/K$. This is the same asymptotic behavior achieved by Hadamard matrices (Guruswami and Sahai, 1999), but α vs. β matrices may correspond to a much larger number B of classifiers.

4. Experimental Results

In this section we investigate the individual performance of binary classifiers for different ECOCs. We are interested on evaluating if the bounds in Equations (5) and (6) are tight, and in using them to get insight into the multiclass performance. Previous work (e.g., Allwein et al., 2000, Hsu and Lin, 2002) compared the all-pairs and one-vs-rest matrices in terms of multiclass error, and here we concentrate attention on the performance of the binary classifiers. Our experimental setup is also propitious to explain why *elimination* (Kreßel, 1999), which is described below, and Hamming decoding are often equivalent in terms of accuracy.

4.1 The *Elimination* Decoding Method for All-Pairs

The *elimination* decoding method applies only to ECOCs with the all-pairs matrix and quantized scores $h(\mathbf{x})$. This decoder was originally described by Kreßel (1999) and independently reintroduced by Platt et al. (2000), where it was called *directed acyclic graph SVM* (DAGSVM) when SVM is the binary learner. It operates iteratively and, at each iteration $n = 1, 2, \ldots, K-2$, the size of the set $A_n = \{h_{i,j}\}$ of active binary classifiers h is decreased. The set A_1 contains all binary classifiers, namely $|A_1| = B$. At iteration n, the output of a classifier $h_{l,m} \in A_n$ is computed, the loosing class $t \in \{l, m\}$ is eliminated, and so are all classifiers related to it, namely $A_{n+1} = A_n - \{h_{i,j} : i = t \vee j = t\}$. The set A_{K-1} contains only one binary classifier, which determines the winner class. When compared to Hamming, *elimination* decoding can lead to substantial

Name	# train	# test	# classes	# attributes	average # training examples per class	minimum # training examples per class
soybean-large	307	376	19	35	16.2	1
vowel	528	462	11	10	48.0	48
vowel-lsf	528	462	11	9	48.0	48
pbvoweluF1-2	599	600	10	2	59.9	42
pbvowelF0-3	760	760	10	4	76.0	76
isolet	6238	1559	26	617	239.9	238
e-set	2160	540	9	617	240.0	240
letter	16000	4000	26	16	615.4	576
satimage	4435	2000	6	36	739.2	409
pendigits	7494	3498	10	16	794.4	719
timit-plp40	138839	7142	39	40	3560.0	304

Table 1: Datasets used for the experiments.

savings given that $K-1$ binary classifiers are consulted, instead of $\binom{K}{2}$. Platt et al. (2000) found the ordering of classifiers $h_{i,j}$ to be not important and adopted: $(i,j) = (1,2),(1,3),\ldots,(1,K),(2,3),\ldots,(K-1,K)$, which was also used here.

It is clear that Hamming and *elimination* decoding can in general lead to different results. For example, it may happen in *elimination* decoding that the class with smallest Hamming distance is prematurely eliminated, and the class with the largest Hamming distance is declared the winner. We note that, if there is a class that wins all other $K-1$ classes, Hamming and *elimination* decoding lead to the same classification result.

4.2 Datasets and Experimental Setup

We evaluated the performance of different ECOCs using the eleven standard datasets listed in Table 1. The datasets *soybean-large*, *vowel*, *isolet*, *letter*, *satimage* and *pendigits* are available at the UCI repository, with associated documentation. The other five datasets are related to speech recognition. In order to facilitate reproducing our results, these datasets and their descriptions were made available on the Web,[3] and here we present only a brief summary. The *vowel-lsf* is a version of vowel, obtained by a non-linear transformation (log-area ratios to line spectral frequencies) that is standard in speech coding. The *e-set* is a subset of isolet consisting of the confusable letters {B, C, D, E, G, P, T, V, Z}. The two versions of the Peterson and Barney's vowel data, namely *pbvowelF0-3* and *pbvoweluF1-2*, are described by Klautau (2002). The *timit-plp40* dataset is a version of TIMIT,[4] a speech database with phonetic transcriptions. We used 12 perceptual linear prediction (PLP) coefficients and energy to represent each frame (10 milliseconds). As phones have different durations, we linearly warped them into three regions, and took the average of each region to obtain a vector with fixed-length (Ganapathiraju et al., 1998). After adding the phone duration (number of frames), we obtained $3\times 13+1=40$ features. We collapsed the 61 TIMIT labels into the standard 39 classes proposed by Kai-Fu Lee.

All datasets have a standard partition into training and test sets, which were used throughout the experiments. For each binary training set, the attributes were normalized to the range $[0,1]$ based on their minimum and maximum values, and the same normalization factors were used for the test set.

3. *http://speech.ucsd.edu/aldebaro/repository*.
4. *http://www.ldc.upenn.edu/Catalog/top_ten.html*.

Dataset	ECOC matrix	SVM parameters (polynomial kernel, unless noted)	K-ary error $\overline{\varepsilon}_K$ (%)		
			$(1-z)_+$	Hamming	*elimination*
soybean-large	all-pairs	$\delta=1, E=2, C=0.1$	10.1	10.1	10.6
	one-vs-rest	linear, $C=1$	6.6 (+)	8.5	-
vowel	all-pairs	RBF, $\gamma=1, C=10$	37.7	34.6 (+)	33.1
	one-vs-rest	RBF, $\gamma=10, C=10$	41.3	68.6	-
vowel-lsf	all-pairs	RBF, $\gamma=1, C=1$	32.2	29.9 (+)	30.3
	one-vs-rest	RBF, $\gamma=10, C=10$	39.2	66.0	-
pbvowelFu1-2	all-pairs	RBF, $\gamma=10, C=1$	19.0	19.0	19.0
	one-vs-rest	RBF, $\gamma=10, C=10$	18.7	28.5	-
pbvowelF0-3	all-pairs	$\delta=1, E=4, C=1$	11.2	10.7	10.8
	one-vs-rest	RBF, $\gamma=10, C=10$	12.2	15.9	-
isolet	all-pairs	$\delta=0, E=3, C=10$	4.0	4.0	3.8
	one-vs-rest	$\delta=0, E=4, C=0.1$	3.8	8.1	-
e-set	all-pairs	$\delta=0, E=4, C=1$	5.6	6.3	5.9
	one-vs-rest	$\delta=1, E=4, C=1$	5.6	8.5	-
letter	all-pairs	RBF, $\gamma=10, C=10$	3.2	2.3	2.2
	one-vs-rest	RBF, $\gamma=10, C=10$	2.1	4.4	-
satimage	all-pairs	RBF, $\gamma=1, C=10$	8.4	8.2	8.3
	one-vs-rest	RBF, $\gamma=10, C=1$	8.2	13.1	-
pendigits	all-pairs	RBF, $\gamma=1, C=10$	2.1	1.6	1.6
	one-vs-rest	RBF, $\gamma=1, C=10$	1.1 (+)	2.1	-
timit-plp40	all-pairs	RBF, $\gamma=1, C=1$	31.3	25.9	26.0
	one-vs-rest	RBF, $\gamma=4, C=1$	27.0	42.9	-

Table 2: Comparison of one-vs-rest and all-pairs matrices in terms of accuracy. The all-pairs with Hamming and one-vs-rest with $L(z)=(1-z)_+$ (max-wins) decoding were compared through McNemar's test, with a symbol (+) indicating the two ECOCs are not equivalent.

The binary learner was the SVM with either the polynomial $K(\mathbf{x},\mathbf{y})=(\mathbf{x}\cdot\mathbf{y}+\delta)^E$ or Gaussian radial-basis function (RBF) $K(\mathbf{x},\mathbf{y})=e^{-\gamma||\mathbf{x}-\mathbf{y}||^2}$ kernel.[5] We used the same SVM parameters for all binary classifiers of a given ECOC matrix. Because we were interested on comparing the performance of binary classifiers using different ECOCs, we chose the *complexity parameter C* and kernel parameters according to performance on the test set. Therefore, our results should not be interpreted as indicating generalization error. More specifically, for each ECOC matrix we tested all decoders using the set of parameters that achieved the smallest error with any decoding method. If different sets of SVM parameters achieved the smallest error, we chose the parameters that minimized the number of distinct support vectors. This methodology differs from the one adopted by Platt et al. (2000), Hsu and Lin (2002), where different SVM parameters could be used for Hamming and *elimination*, making harder to identify the reason for their similar accuracy.

4.3 Results

Table 2 shows the results comparing the K-ary error of ECOCs with one-vs-rest and all-pairs. For the all-pairs matrix, *elimination* achieved accuracy similar to Hamming decoding, while $L(z)=(1-z)_+$ was slightly worse. For the one-vs-rest matrix, max-wins has much better performance than Hamming decoding, as expected.

5. Using $\delta=0$ and $E=1$ leads to a *linear* SVM, which can be converted to a perceptron to avoid storing the support vectors and save computations.

Our main goal is to evaluate the binary classifiers, but we note that Table 2 indicates that quantizing the scores may be beneficial when using all-pairs (or other matrices with zero entries). In contrast, Allwein et al. (2000) concluded that $L(z) = (1-z)_+$ often gives better results than Hamming decoding for the all-pairs matrix. For example, they reported that, for ECOCs using all-pairs and SVMs with polynomial kernel of order 4, decoding with $L(z) = (1-z)_+$ and Hamming led to 27.5% and 50.4% of error, respectively, for the satimage dataset. As we used the test set to perform model selection, their results should not be compared directly to Table 2. However, when we used all-pairs and Hamming with the SVM parameters $\delta = 1$, $E = 4$ and C=1, the error rate for satimage was 11.0%.

We attribute the fact that Hamming outperforms $L(z) = (1-z)_+$ to the large number of *unseen* classes for binary classifiers of all-pairs. If $\mathbf{M}(k,b) = 0$, we say that class k is *unseen* with respect to classifier f_b. During the test stage, all instances associated to an *unseen* class k lead f_b to make potentially erratic predictions.[6] All-pairs is the α vs. β matrix with the largest number of *unseen* classes per binary classifier. In this case, the scores are unreliable and quantizing them to $\{-1,+1\}$ can lead to higher accuracy.

At this point we assume Hamming and max-wins as the decoders for all-pairs and one-vs-rest, respectively, and compare the accuracy of these two ECOCs using McNemar's test (see Dietterich, 1998) (0.05 significance level). As shown in Table 2, McNemar's test indicated that the two classifiers were equivalent for 7 out of the 11 datasets. We now investigate the performance of the individual binary classifiers, trying to characterize the situations where one ECOC outperforms the other. This analysis is not required in order to understand the numbers for the soybean-large dataset though, which confirm that all-pairs may perform poorly if there is not enough training data for all classifiers.

Table 3 shows the performance on the test set of the binary classifiers corresponding to the ECOCs in Table 2. Besides some statistics of the binary error that we will discuss later, Table 3 presents histograms of Hamming distances $d_H(\mathbf{M}(k^*,\cdot), h(\mathbf{x}))$ between quantized scores $h(\mathbf{x})$ and codeword $\mathbf{M}(k^*,\cdot)$, where k^* is the winner class. For each dataset, the sum of the six right-most columns is equal to the total number of test instances. These six columns were split into two subsets, depending whether Hamming decoding led to a K-ary error or not. For example, for the soybean-large dataset and one-vs-rest matrix, 344 test instances were correctly classified (sum of 3 columns under "when match") and 32 were misclassified (columns "when error"), corresponding to the K-ary error of 8.5% in Table 2. In this case, all binary classifiers made the correct decision for 343 instances (column "when match / 0"). For one test instance, the Hamming distance was one, but the instance was correctly classified (column "when match / 1"). Among the instances that led to errors, there were 13 for which $d_H = 0$.

When $d_H = 0$ for one-vs-rest (only one binary classifier has a positive score), max-wins and Hamming decoding lead to the same decision. Table 3 shows that, for one-vs-rest, most of the K-ary errors occurred with the winner class leading to $d_H = 1$, while only 5 out of 21,399 test instances led to $d_H > 1$. Hence, in almost all cases, the results with max-wins differed from the ones with Hamming when the quantized scores $h(\mathbf{x})$ led to either a tie between two or among all K classes. In these cases, max-wins could use the magnitudes of scores as a tie-breaking rule, outperforming Hamming decoding as shown in Table 2. For the isolet dataset for example, among

6. We are using the term *unseen* classes to denote a problem that has been discussed in the literature related to all-pairs. For example, Hastie and Tibshirani (1998) conducted an experiment with artificial data to characterize it, and mentioned that Geoffrey Hinton originally pointed out the problem.

One-versus-rest										
	Binary error statistics				Occurrences of $d_H(\mathbf{M}(k^*,\cdot),h(\mathbf{x}))$					
					when match			when error		
Dataset	mean ($\overline{\varepsilon}_b$)	min.	max.	std.	0	1	>1	0	1	>1
soybean-large	0.006	0	0.043	0.013	343	1	0	13	19	0
vowel	0.074	0.054	0.087	0.011	120	25	0	35	282	0
vowel-lsf	0.076	0.032	0.132	0.029	141	16	0	64	241	0
pbvowelFu1-2	0.042	0.012	0.098	0.027	424	5	0	78	93	0
pbvowelF0-3	0.028	0.005	0.041	0.013	621	18	0	68	53	0
isolet	0.005	0	0.013	0.005	1391	40	2	25	102	0
e-set	0.017	0.002	0.026	0.008	475	18	1	18	28	0
letter	0.002	0	0.008	0.002	3807	15	0	39	139	0
satimage	0.032	0.012	0.056	0.017	1708	29	0	95	168	0
pendigits	0.003	2.86e-4	0.009	0.002	3412	11	0	23	52	0
timit-plp40	0.028	0.013	0.092	0.025	4011	65	2	603	2461	0
All-pairs										
	Binary error statistics				Occ. of $d_H(\mathbf{M}(k^*,\cdot),h(\mathbf{x}))-0.5\binom{K-1}{2}$					
					when match			when error		
Dataset	mean ($\overline{\varepsilon}_b$)	min.	max.	std.	0	1	>1	0	1	>1
soybean-large	0.010	0	0.789	0.079	334	1	3	23	3	12
vowel	0.054	0	0.286	0.073	297	4	1	133	26	1
vowel-lsf	0.046	0	0.238	0.068	318	5	1	125	13	0
pbvowelFu1-2	0.026	0	0.184	0.043	486	0	0	112	2	0
pbvowelF0-3	0.013	0	0.145	0.030	677	2	0	80	1	0
isolet	0.002	0	0.126	0.010	1496	1	0	51	11	0
e-set	0.014	0	0.042	0.014	505	1	0	27	7	0
letter	0.002	0	0.034	0.004	3907	0	1	86	6	0
satimage	0.019	0	0.092	0.029	1833	2	0	161	4	0
pendigits	0.004	0	0.022	0.005	3439	2	0	54	3	0
timit-plp40	0.015	0	0.220	0.027	5264	26	1	1799	51	1

Table 3: Performance of the binary classifiers associated to the ECOCs in Table 2. The six rightmost columns are histograms of Hamming distances d_H between quantized scores $h(\mathbf{x})$ and the codeword $\mathbf{M}(k^*,\cdot)$, where k^* is the winner class. For all-pairs, the constant $0.5\binom{K-1}{2}$ was subtracted from d_H.

the 144 instances that led to $d_H > 0$, 109 and 42 were correctly classified by max-wins and Hamming decoding, respectively. In this case, around half of the K-ary errors were associated to $h(\mathbf{x})$ with two positive entries. Assuming the correct class was among the two competing, a random guess had a 50% error rate. For the cases where $h_b(\mathbf{x}) = -1$, $\forall b$, Hamming decoding had to randomly break the tie among $K = 26$ classes.

For all-pairs, the constant $0.5\binom{K-1}{2}$ corresponding to $\binom{K-1}{2}$ zero entries in $\mathbf{M}$ was subtracted from d_H in Table 3. In this case, there was a class that won according to all of its $K-1$ binary classifiers for 99.1% of the test instances when considering all datasets.[7] If we look at each dataset individually, the percentage varies from 93.1% (vowel) to 99.9% (pendigits). This percentage of unanimous decisions can explain why *elimination* and Hamming decoding perform similarly in terms of accuracy (Platt et al., 2000, Hsu and Lin, 2002).

We now evaluate how tight are the bounds on the multiclass error $\overline{\varepsilon}_K$, and how they can help to understand the ECOC performance. It can be seen from Table 3 that $\overline{\varepsilon}_b$ is lower for one-vs-rest only for soybean-large and pendigits, which are the two datasets for which one-vs-rest outperformed all-

7. Kreßel (1999) noted this behavior in his experiments.

pairs. For vowel and vowel-lsf (for which all-pairs achieved higher accuracy), $\overline{\varepsilon}_b$ for one-vs-rest is higher than for all-pairs by a factor of 1.37 and 1.65, respectively. In spite of these facts, a careful evaluation indicates that only the binary error $\overline{\varepsilon}_b$ does not suffice to predict the K-ary error. We elaborate it as follows.

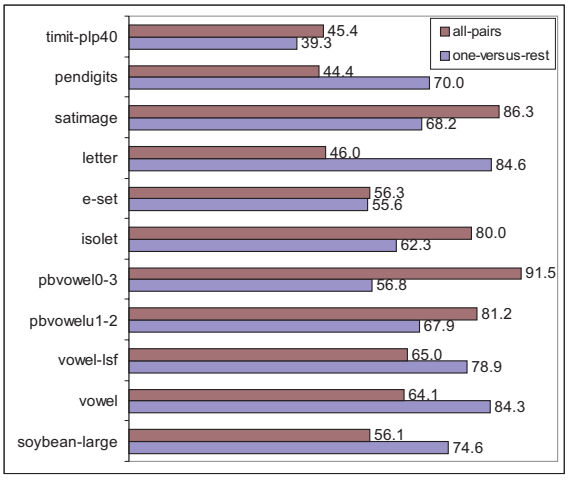

Figure 1: Comparison of upper bounds and empirical results for ECOCs with Hammming decoding. The numbers (in percentage) correspond to the division of results in Table 2 by the upper bounds on average number of K-ary errors $\overline{\varepsilon}_K$ for Equations (5) and (6). A result of 100% would correspond to an ECOC achieving in practice the upper bound on $\overline{\varepsilon}_K$.

Figure 1 shows that the bounds (5) and (6) on the K-ary error $\overline{\varepsilon}_K$ for Hamming decoding are close to experimental results. These bounds, and consequently the binary error $\overline{\varepsilon}_b$, can be effectively used to predict $\overline{\varepsilon}_K$ when using the Hamming decoder. In practice however, we want to use max-wins decoding for one-vs-rest, for which $\overline{\varepsilon}_b$ alone cannot predict performance. For example, for isolet, one-vs-rest has a binary error $\overline{\varepsilon}_b$ that is 2.5 times higher than $\overline{\varepsilon}_b$ for all-pairs, but still achieves slightly better K-ary error $\overline{\varepsilon}_K$.

5. Conclusions

We presented new bounds on the K-ary error of ECOCs with nearest-neighbor decoding. We then specialized the bounds for Hamming decoding and α vs. β matrices. We showed that for large enough K, α vs. β matrices with $\alpha = \beta = K/2$, have the same behavior as Hadamard matrices. We also conducted simulations to evaluate the bounds and compare ECOCs based on one-vs-rest and all-pairs matrices.

The conclusions of these experiments can be summarized as follows. The bounds are relatively tight, and accurately predict the multiclass error based on the performance of the binary classifiers when using Hamming decoding. Quantizing the scores (as in Hamming decoding) can be beneficial for ECOCs with the all-pairs matrix, and we attribute this to the influence of *unseen* classes, for which the scores are unreliable. Hamming and *elimination* decoding achieved equivalent performance for all datasets, and we explained that these two decoders lead to the same classification

result when one class wins according to all of its $K-1$ binary classifiers, which is the case for 99.1% of our test instances.

Acknowledgments

We are grateful to the reviewers and editor for comments that improved the paper. Aldebaro would like to acknowledge support from CAPES, Brazilian Government.

References

E. Allwein, R. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1(2):113–141, 2000.

A. Berger. Error-correcting output coding for text classification. In *International Joint Conference on Artificial Intelligence: Workshop on machine learning for information filtering*, 1999.

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

K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2(2):265–292, 2002.

H. David. *The method of paired comparisons*. Charles Griffin, 1963.

T. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, 1998.

T. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–86, 1995.

J. Friedman. Another approach to polychotomous classification. Technical report, Stanford University, 1996. URL `http://www-stat.stanford.edu/~jhf`.

J. Fürnkranz. Round robin classification. *Journal of Machine Learning Research*, 2:721–747, 2002.

A. Ganapathiraju, J. Hamaker, and J. Picone. Support vector machines for speech recognition. In *Third International Conference on Spoken Language Processing*, 1998.

V. Guruswami and A. Sahai. Multiclass learning, boosting, and error-correcting codes. In *12th Annual Conference on Computational Learning Theory*, pages 145–155, 1999.

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

C.-W. Hsu and C.-J. Lin. A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.

A. Klautau. Classification of Peterson & Barney's vowels using Weka. Technical report, UFPA, 2002. URL `http://citeseer.nj.nec.com/klautau02classification.html`.

A. Klautau, N. Jevtić, and A. Orlitsky. Combined binary classifiers with applications to speech recognition. In *Seventh International Conference on Spoken Language Processing*, pages 2469–2472, 2002.

U. Kreßel. *Advances in Kernel Methods - Support Vector Learning*, chapter 15 - Pairwise Classification and Support Vector Machines. MIT Press, Cambridge, 1999.

M. Moreira and E. Mayoraz. Improved pairwise coupling classification with correcting classifiers. In *Proceedings of the Tenth European Conference on Machine Learning*, pages 160–71, 1998.

J. Platt, N. Cristianini, and J. Shawe-Taylor. Large margin DAGs for multiclass classification. In *Advances in Neural Information Processing Systems 12*, pages 547–553, 2000.

B. Zadrozny. Reducing multiclass to binary by coupling probability estimates. In *Advances in Neural Information Processing Systems 14*, pages 1041–1048, 2002.