COMBINED BINARY CLASSIFIERS WITH APPLICATIONS TO SPEECH RECOGNITION

Aldebaro Klautau[*] *Nikola Jevtić*[†] *Alon Orlitsky*[†]

ECE Department, UCSD
9500 Gilman Drive
La Jolla, CA 92093, USA

ABSTRACT

Many applications require classification of examples into one of several classes. A common way of designing such classifiers is to determine the class based on the outputs of several binary classifiers. We consider some of the most popular methods for combining the decisions of the binary classifiers, and improve existing bounds on the error rates of the combined classifier over the training set. We also describe a new method for combining binary classifiers. The method is based on stacking a neural network and, when used with support vector machines as the binary learners, substantially decreased the error rate in two vowel classification tasks.

1. INTRODUCTION

Many techniques for constructing binary classifiers with good generalization capabilities were developed in recent years. For example, support vector machines (SVM) [1] and AdaBoost [2]. However, in many situations the number of classes is larger than two, e.g., [3]. This is the case in most speech applications, where the classes are, for example, vowels [4] or HMM states [5]. While multiclass versions of most classification algorithms exist, e.g. [6] (SVM) and [2] (AdaBoost), they tend to be complex. For example, in [7, 8], a multiclass version of AdaBoost was applied to speech recognition, but the original algorithm had to be simplified due to a high computational cost.

A more common approach is to construct the multiclass classifier by combining the outputs of several binary ones [3, 9]. 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.

Recent work has used the error incurred by the binary classifiers to upper bound the error committed by the combined nearest-neighbor classifier [3, 10]. These results also suggest guidelines for constructing accurate multiclass classifiers.

We present two contributions. We strengthen the bounds in [3] and extend the class of decoders to which they apply. We also propose a decoding method based on a stacked neural network. The new decoding method substantially decreased the error rate in two vowel classification tasks with SVM as the binary learners.

The paper is organized as follows. Section 2 discusses the construction of multiclass classifiers from binary ones. Theoretical bounds for the training error are presented in Section 3. The new decoding technique is described and evaluated in Section 4, followed by conclusions in Section 5.

[*]Supported by CAPES, Brazil.
[†]Supported by NSF grant # 9815018.

2. COMBINING CLASSIFIERS

In supervised classification problems, one is given a training set $\{(x_1, y_1), \ldots, (x_N, y_N)\}$ containing N *examples*. Each example (x, y) consists of an instance $x \in \mathcal{X}$ and a label $y \in \{1, \ldots, K\}$, where $\mathcal{X}$ is the instance space and $K \geq 2$ is the number of classes. A *classifier* is a mapping $F : \mathcal{X} \to \{1, \ldots, K\}$ from instances to labels. One seeks classifiers that minimize the number of misclassified examples, or *errors*.

In recent years there has been considerable progress in the construction of classifiers for binary problems, consisting of $K = 2$ classes. Several of these constructions yield *confidence-valued* classifiers $f : \mathcal{X} \to \mathbb{R}$ which return a *score*. The sign of the score indicates the predicted class, and its magnitude reflects the confidence about the prediction. These classifiers can be converted to standard ones by taking the sign of their output.

In multiclass problems, the number K of classes is larger than two. 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 (possibly confidence-valued) *binary classifiers* is constructed, where each classifier is trained to distinguish between two subsets of classes. Then, the outputs of these binary classifiers are combined to produce a *K-ary classifier* which attempts to determine the correct label.

The classes involved in the training of the binary classifiers are typically [9, 3] specified by a matrix $M \in \{-1, 0, 1\}^{K \times B}$. Classifier f_b is trained according to column $M(\cdot, b)$. If $M(k, b) = +1$, all examples of class k are considered "positive", if $M(k, b) = -1$, all examples of class k are "negative", and if $M(k, b) = 0$, none of the examples of class k participates in the training of classifier f_b.

The K-ary classifier then takes the scores $f(x) = (f_1(x), \ldots, f_B(x))$ and combines them using a function $g : \mathbb{R}^B \to \{1, \ldots, K\}$ to obtain a K-ary classifier $F(x) = g(f(x))$.

One can view the rows of the matrix M as codewords and the function g as decoding the output $f(x)$ of the binary classifiers. By analogy to coding, M is refereed to as an *error-correcting output code* (ECOC), and the function g is called the *decoder*.

Two common ECOC constructions are the *all-pairs* and *one-against-all* codes. The all pairs ECOC consists of $B = \binom{K}{2}$ binary classifiers $f_{i,j}$, $1 \leq i < j \leq K$, where $f_{i,j}$ is trained to distinguish between classes i and j. The *one-against-all* ECOC consists of $B = K$ binary classifiers $f_1, \ldots, f_K$, where f_i is trained to distinguish between class i and all other classes. Their respective matrices for $K = 4$ are

$$\begin{bmatrix} +1 & +1 & +1 & 0 & 0 & 0 \\ -1 & 0 & 0 & +1 & +1 & 0 \\ 0 & -1 & 0 & -1 & 0 & +1 \\ 0 & 0 & -1 & 0 & -1 & -1 \end{bmatrix} \text{ and } \begin{bmatrix} +1 & -1 & -1 & -1 \\ -1 & +1 & -1 & -1 \\ -1 & -1 & +1 & -1 \\ -1 & -1 & -1 & +1 \end{bmatrix}.$$

The all-pairs code is related to well-known methods of paired comparisons in statistics [11] and was applied to classification problems in [12]. While there is empirical evidence that ECOC consisting of fewer binary classifiers may outperform all-pairs codes [3, 13], and ECOC design methods are under investigation [14, 15], several researchers have adopted the all-pairs with promising results [16, 17, 18, 19, 20]. In speech recognition, the all-pairs code was used for connected digits [5] and for vowel classification [4], both using SVM. It was also used with artificial neural networks for TIMIT vowel classification [21].

Some of the most natural decoders are *nearest-neighbor* decoders. They use some distortion measure $d : \mathbb{R}^B \times \{-1, 0, 1\}^B \to [0, \infty]$, and select the class $F(x) = \arg\min_k d(f(x), M(k, \cdot))$ that minimizes the distortion between $f(x)$ and the row $M(k, \cdot)$.

Of special interest are *margin-based* distortions [3, 13], which are defined by $d(f(x), M(k, \cdot)) = \sum_{b=1}^{B} L(z_b)$, where $L : \mathbb{R} \to [0, \infty]$, and $z_b = f_b(x)M(y, b)$ is the *margin* of example (x, y) under classifier f_b.

It can be shown that for the all-pairs ECOC, all decreasing linear functions L lead to the same classification result. Similarly, for the one-against-all ECOC, all decreasing functions L lead to the same classification result: they select the class k which maximizes $f_k(x)$. This decision rule is called *max-wins*.

The binary classifiers may of course return *hard decisions* $h(x) \in \{-1, 1\}$, 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*, the nearest-neighbor decoder that minimizes the *Hamming distance* (modified to allow for 0's):

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

It can be easily verified [3] that this is a special case of a margin-based distortion where $L(z) = (1 - z)/2$.

In general however, the decoder g can be any mapping. In Section 4 we will show that in some applications, more complex decoders may outperform nearest-neighbor ones.

3. BOUNDS ON THE TRAINING ERROR

Previous work [3, 10] used the error, and more generally, distortion, incurred by the binary classifiers, to upper bound the error committed by the K-ary classifier. This section strengthens these bounds and extends the distortion measures to which they apply. We begin by defining the quantities involved.

The number of errors the K-ary classifier F commits on the training set is $\varepsilon_K \stackrel{\text{def}}{=} |\{n : F(x_n) \neq y_n\}|$ and its *error rate* is $\overline{\varepsilon}_K \stackrel{\text{def}}{=} \varepsilon_K / N$. The distortion between the outputs of the binary classifiers and the correct codeword over the training set is

$$D \stackrel{\text{def}}{=} \sum_{n=1}^{N} d(f(x_n), M(y_n, \cdot)) \tag{1}$$

and their *average distortion* is $\overline{D} \stackrel{\text{def}}{=} D/N$.

To relate $\overline{\varepsilon}_K$ and $\overline{D}$, the minimum Hamming distance between any two rows was defined in [3] to be $\rho \stackrel{\text{def}}{=} \min\{d_H(M(k, \cdot), M(k', \cdot)) : k \neq k'\}$. For example, in one-against-all $\rho = 2$, and for all-pairs $\rho = (B + 1)/2$. They showed that for all nearest-neighbor decoders with margin-based distortions,

$$\overline{\varepsilon}_K \leq \overline{D}/(\rho L^*) \tag{2}$$

where $L^* \stackrel{\text{def}}{=} 0.5 \min_z \{L(z) + L(-z)\}$.

For Hamming decoding, [3] presented a more natural form of the bound which relates the K-ary classifier's error $\overline{\varepsilon}_K$ to that committed by the binary classifiers. Let $T \stackrel{\text{def}}{=} \{(n, b) : M(y_n, b) \neq 0\}$ be the set of pairs (n, b) corresponding to examples and binary classifiers used when designing F, and let $|T|$ be the cardinality of T. For example, in the one-against-all ECOC, every row of the matrix M contains $B = K$ nonzero entries, hence $|T| = NK$, and for the all-pairs ECOC, each row of M has $K - 1$ nonzero entries, hence $|T| = N(K - 1)$. The number of training examples misclassified by the binary classifiers is then

$$\varepsilon_b \stackrel{\text{def}}{=} |\{(n, b) \in T : h_b(x_n) \neq M(y_n, b)\}|, \tag{3}$$

and the *error rate* of the binary classifiers is $\overline{\varepsilon}_b \stackrel{\text{def}}{=} \varepsilon_b / |T|$.

For Hamming decoding, $D = (NB - |T|)/2 + \varepsilon_b$ and $L^* = 1/2$, hence (2) can be used to bound $\overline{\varepsilon}_K$ in terms of $\overline{\varepsilon}_b$. This was done in [3], and when their bound is applied to the all-pairs ECOC, it yields

$$\overline{\varepsilon}_K \leq \frac{2(K - 1)(K - 2 + 4\overline{\varepsilon}_b)}{K(K - 1) + 2}. \tag{4}$$

We strengthen the bound (2) and extend it to non margin-based distortions. To illustrate the new bound, we first discuss a bound originally presented in [10].

Example 1 [10] For one-against-all with Hamming decoding,

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

Proof Every K-ary classification error can be attributed to at least one binary classification error, hence $\varepsilon_K \leq \varepsilon_b$. Since, in this case, $|T| = NK$,

$$\overline{\varepsilon}_K = \frac{\varepsilon_K}{N} \leq \frac{\varepsilon_b}{N} = \frac{|T|\overline{\varepsilon}_b}{N} = K\overline{\varepsilon}_b. \qquad \square$$

This bound can be shown to be tight as, in the worst-case, each binary error can indeed lead to one K-ary classification error. A similar reasoning can be applied to the all-pairs ECOC.

Example 2 For all-pairs with Hamming decoding,

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

Proof Every K-ary classification error can be attributed to at least one binary classification error, hence $\varepsilon_K \leq \varepsilon_b$. For the all-pairs, $|T| = N(K - 1)$, hence

$$\overline{\varepsilon}_K = \frac{\varepsilon_K}{N} \leq \frac{\varepsilon_b}{N} = \frac{|T|\overline{\varepsilon}_b}{N} = (K - 1)\overline{\varepsilon}_b. \qquad \square$$

This bound is both simpler and tighter than (4). It can be shown that whenever (4) is less than one, the new bound is lower. Figure 1 illustrates the difference between the two bounds for a range of values of K and $\overline{\varepsilon}_b$.

The new bound also better correlates with empirical evidence showing that all-pairs usually outperforms one-against-all. Since each all-pairs binary classifier is trained on just two classes, one would expect the all-pairs binary classification error rate $\overline{\varepsilon}_b$ to be lower than the corresponding error rate for one-against-all. This, and the fact that the bound (6) is lower than (5), suggest that the K-ary classification error of all-pairs should be lower than that of one-against-all, agreeing with experimental results in [3]. By contrast,

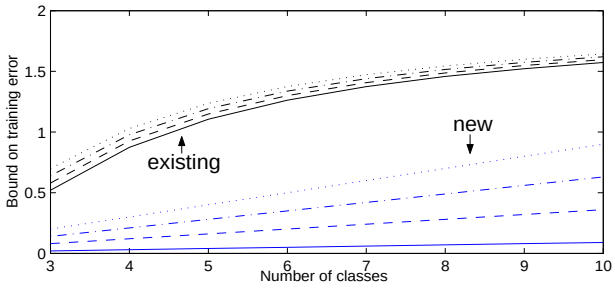

Fig. 1. Existing and new bounds on training error rate of all pairs with Hamming decoding for $\overline{\varepsilon}_b$ =0.01 (bottom curves), 0.04, 0.07, and 0.1 (top curves).

for some values of K and $\overline{\varepsilon}_b$, the bound (4) is higher than (5), and therefore does not necessarily lead to the same conclusion.

We now address the general bound. Given an ECOC matrix M, a distortion measure d, and a vector $f \in \mathbb{R}^B$, let $d_1(f)$ be the smallest distortion between f and any row of M, and let $d_2(f)$ be the smallest distortion between f and the remaining rows of M. Define $d_1 = \min_f d_1(f)$ and $d_2 = \min_f d_2(f)$.

Theorem 1 For any ECOC, the K-ary error of nearest-neighbor decoding over the training set satisfies

$$\overline{\varepsilon}_K \leq (\overline{D} - d_1)/(d_2 - d_1).$$

Proof Split the training set into $C \stackrel{\text{def}}{=} \{(x_n, y_n) : F(x_n) = y_n\}$ and $W \stackrel{\text{def}}{=} \{(x_n, y_n) : F(x_n) \neq y_n\}$, containing the correctly and wrongly classified examples. Equation (1) can then be written as

$$D = \sum_{(x_n,y_n)\in C} d(f(x_n), M(y_n, \cdot)) + \sum_{(x_n,y_n)\in W} d(f(x_n), M(y_n, \cdot)). \quad (7)$$

Note that $\varepsilon_K = |W|$, 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 $\varepsilon_K \leq (D - N \cdot d_1)/(d_2 - d_1)$ and the theorem follows. □

We note that this bound applies to all distortion measures, not just margin-based ones, and that for margin-based distortions, this bound is always at least as strong as (2). Also, when applied to the one-against-all and the all-pairs ECOC's with Hamming decoding, it yields bounds (5) and (6).

4. EXPERIMENTAL RESULTS

To evaluate the performance of different ECOC's and decoding methods, we tested them on several popular datasets. This section presents the results obtained.

The binary classifiers were constructed using SVM. We chose SVM because they have shown promising results in classification tasks in the speech domain [4, 22], and in noisy digits [5] and large-vocabulary speech recognition [23].

We used the SVM implementation available in [24] with $C = 1$ and a nonhomogeneous polynomial kernel with degree 4. The main goal was to evaluate decoding methods, so no attempts were made to tune the SVM parameters.

The ECOC's used were the one-against-all and the all-pairs. Though not possessing good error correcting capabilities, they are the most used schemes due to their simplicity.

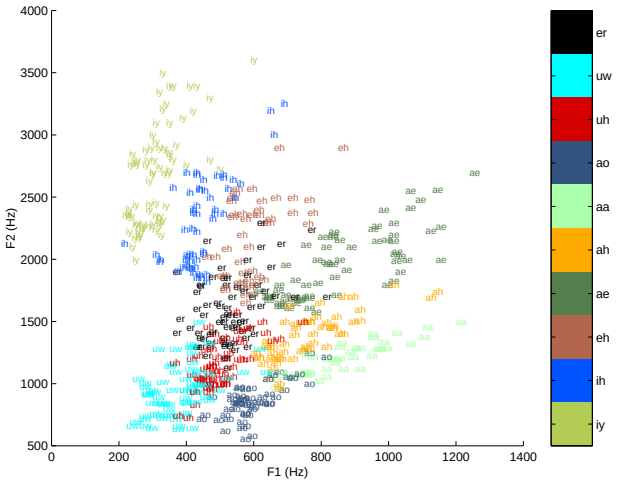

Fig. 2. Training set with the ten vowels of pbvoweluF1-2.

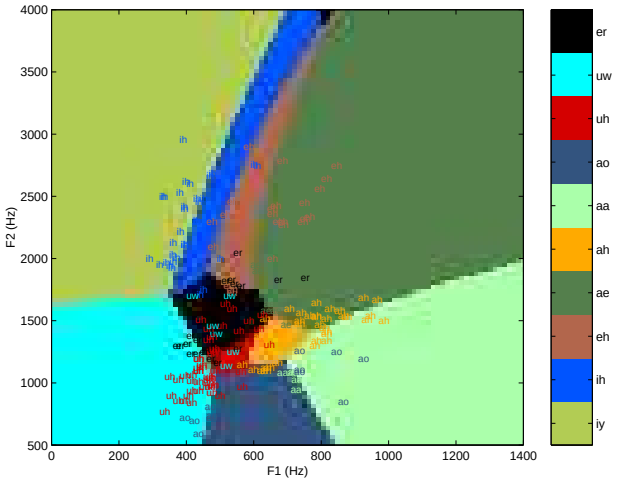

Fig. 3. Decision regions for pbvoweluF1-2 with SVM combined through one-against-all ECOC and max-wins decoding.

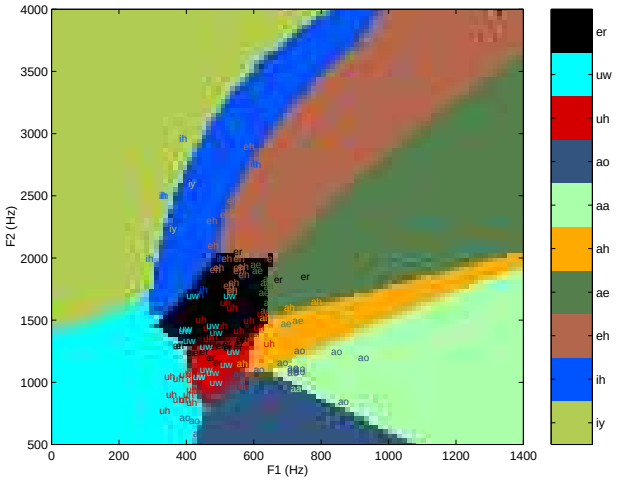

Fig. 4. Decision regions for pbvoweluF1-2 with SVM combined through all-pairs ECOC and Hamming decoding.

Table 1. Error (%) for different decoding methods with all-pairs ECOC and max-wins with one-against-all ECOC.

	one-against-all	all-pairs					
Dataset	max-wins	Hamming	$-z$	$(1-z)_+$	$(0.5-z)_+$	e^{-z}	ANN
vowel	46.97	40.26	66.45	38.96	38.09	44.16	**34.85**
e-set	**5.37**	6.30	9.07	**5.37**	**5.37**	6.30	5.93
pbvowel0-3	17.63	**10.66**	62.50	11.18	10.79	13.68	11.97
pbvowelu1-2	31.50	22.17	70.00	24.50	22.50	28.00	**18.00**
soybean	**7.18**	9.84	12.50	12.23	10.90	11.70	9.31
pendigits	**2.32**	2.49	13.46	2.54	2.52	2.63	2.49
satimage	10.70	11.05	37.45	10.40	10.65	10.40	**10.35**

In the one-against-all ECOC, the binary classifiers were combined using the max-wins rule. In the all-pairs ECOC, the binary classifiers were combined using Hamming decoding, four margin-based distortions corresponding to $L(z)$ given by $-z$, $(1-z)_+$, $(0.5-z)_+$ and e^{-z}, where $(z)_+ = \max\{z, 0\}$, and a new method based on stacking [25] an artificial neural network (ANN) to ECOC.

The ANN was a multi-layer perceptron trained with backpropagation. More details can be found in [24]. The ANN was trained on features consisting of the scores $f(x)$, which were generated using 10-fold cross-validation.

We note that alternatives to nearest-neighbor decoding have been tested before [17, 26]. In particular, [27] investigated stacking classifiers to the one-against-all ECOC.

The datasets used were: *vowel*, *e-set*, *pbvowelF0-3*, *pbvowelF1-2*, *soybean*, *pendigits* and *satimage*. The last three datasets are not related to speech and their documentation can be found on the Web. A detailed description of the four speech-related datasets was made available in [28]. In summary, vowel is the data collected by Deterding. The *e-set* is a subset of ISOLET consisting of the confusable letters {B, C, D, E, G, P, T, V, Z}. The Peterson and Barney's vowel data *pbvowel* was organized into two versions: *pbvowelF0-3*, with fundamental frequency (F0) and three formants (F1-F3), and *pbvoweluF1-2*, a subset of *pbvowelF0-3* that contains only the instances unanimously identified by listeners and F1-F2.

Table 1 shows the results. The training data for pbvoweluF1-2 is shown in Figure 2, and the decision regions for the one-against-all and all-pairs with Hamming decoding are shown in Figures 3 and 4, respectively.

It can be seen that all-pairs combined with stacked ANN gave the best results for three out of the seven datasets. In two of the three, it provided significant improvements over all other classifiers. Of the remaining four datasets, two were best classified by one-against-all, one by all-pairs with Hamming decoding, and one was equally well classified by several decoders. In all these cases the improvements over the stacked ANN were relatively small. Table 1 also shows the importance of bounding $f(x)$. This was not crucial in [13], which used boosted naive Bayes learners.

5. CONCLUSIONS

A new bound on the training error of the combined classifiers with nearest-neighbor decoding was presented. A new method for decoding, which is based on stacking a neural network to ECOC was tested and led to considerable improvements in two vowel classification tasks.

6. REFERENCES

[1] R. Herbrich. *Learning Kernel Classifiers: Theory and Algorithms*. MIT Press, 2002.

[2] R. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. In *COLT*, 1998.

[3] E. Allwein, R. Schapire, and Y. Singer. Reducing multiclass to binary: a unifying approach for margin classifiers. *Journal of Machine Learning Research*, pages 113–141, 2000.

[4] P. Clarkson and P. Moreno. On the use of support vector machines for phonetic classification. In *ICASSP*, pages 585–8 vol.2, 1999.

[5] S. Fine, G. Saon, and R. Gopinath. Digits recognition in a noisy environment via a sequential GMM/SVM system. In *ICASSP (submitted)*, 2002.

[6] J. Weston and C. Watkins. Support vector machines for multi-class pattern recognition. In *ESANN*, pages 219–24, 1999.

[7] H Schwenk. Using boosting to improve a hybrid HMM/neural network speech recognizer. In *ICASSP*, pages 1009–12, 1999.

[8] G. Zweig. Boosting gaussian mixtures in an LVCSR system. In *ICASSP*, pages 1527–30, 2000.

[9] T. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artif. Intellig. Research*, 2:263–86, 1995.

[10] V. Guruswami and A. Sahai. Multiclass learning, boosting, and error-correcting codes. In *COLT*, pages 145–155, 1999.

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

[12] J. Friedman. Another approach to polychotomous classification. Technical report, Stanford University, 1996.

[13] B. Zadrozny. Reducing multiclass to binary by coupling probability estimates. In *NIPS*, 2001.

[14] K. Crammer and Y. Singer. On the learnability and design of output codes for multiclass problems. In *COLT*, 2000.

[15] W. Utschick and W. Weichselberger. Stochastic organization of output codes in multiclass learning problems. *Neural Computation*, 13(5):1065–1102, 2001.

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

[17] M. Moreira and E. Mayoraz. Improved pairwise coupling classification with correcting classifiers. In *10th European Conference on Machine Learning*, pages 160–71, 1998.

[18] E. N. Mayoraz. Multi-class classification with pairwise coupled neural networks or support vector machines. In *ICANN*, pages 314–21, 2001.

[19] V. Roth and K. Tsuda. Pairwise coupling for machine recognition of hand-printed japanese characters. In *CVPR*, pages 1120–1125, 2001.

[20] Y. Yao, G. Marcialis, M. Pontil, P. Frasconi, and F. Roli. A new machine learning approach to fingerprint classification. In *7th Congress of the Italian Association for Artificial Intelligence*, pages 57–63, 2001.

[21] S. Zahorian and Z. Nossair. A partitioned neural network approach for vowel classification using smoothed time/frequency features. *IEEE Trans. on Speech and Audio Processing*, 7(4):414–25, 1999.

[22] A. Ganapathiraju, J. Hamaker, and J. Picone. Support vector machines for speech recognition. In *ICSLP*, 1998.

[23] A. Ganapathiraju. *Support Vector Machines for Speech Recognition*. PhD thesis, Mississippi State University, 2002.

[24] http://www.cs.waikato.ac.nz/ml/weka.

[25] D. Wolpert. Stacked generalization. *Neural Networks*, pages 241–259, 1992.

[26] J. Platt, N. Cristianini, and J. Shawe-Taylor. Large margin DAGs for multiclass classification. In *NIPS*, pages 547–553, 2000.

[27] E. Mayoraz and E. Alpaydin. Support vector machines for multi-class classification. In *IWANN*, pages 833–42, 1999.

[28] http://speech.ucsd.edu/aldebaro/repository.