Classification by Pairwise Coupling

TREVOR HASTIE *

Stanford University

and

ROBERT TIBSHIRANI †

University of Toronto

Abstract

We discuss a strategy for polychotomous classification that involves estimating class probabilities for each pair of classes, and then coupling the estimates together. The coupling model is similar to the Bradley-Terry method for paired comparisons. We study the nature of the class probability estimates that arise, and examine the performance of the procedure in real and simulated datasets. Classifiers used include linear discriminants, nearest neighbors, adaptive nonlinear methods, and the support vector machine.

*Department of Statistics, Sequoia Hall, Stanford University, Stanford California 94305; trevor@playfair.stanford.edu

†Department of Preventive Medicine and Biostatistics, and Department of Statistics; tibs@utstat.toronto.edu

1 Introduction

We consider the discrimination problem with K classes and N training observations. The training observations consist of predictor measurements $\mathbf{x} = (x_1, x_2, \ldots x_p)$ on p predictors and the known class memberships. Our goal is to predict the class membership of an observation with predictor vector $\mathbf{x}_0$

Typically K-class classification rules tend to be easier to learn for $K = 2$ than for $K > 2$ — only one decision boundary requires attention. Friedman (1996a) suggested the following approach for the the K-class problem: solve each of the two-class problems, and then for a test observation, combine all the pairwise decisions to form a K-class decision. Friedman's combination rule is quite intuitive: assign to the class that wins the most pairwise comparisons.

Friedman points out that this rule is equivalent to the Bayes rule when the class posterior probabilities p_i (at the test point) are known:

$$\text{argmax}_i[p_i] = \text{argmax}_i[\sum_{j \neq i} I\left(p_i/(p_i + p_j) > p_j/(p_i + p_j)\right)]$$

Note that Friedman's rule requires only an estimate of each pairwise decision. Many (pairwise) classifiers provide not only a rule, but estimated class probabilities as well. In this paper we argue that one can improve on Friedman's procedure by combining the pairwise class probability estimates into a joint probability estimate for all K classes.

This leads us to consider the following problem. Given a set of mutually exclusive events $A_1, A_2, \ldots A_K$, some experts give us pairwise probabilities $r_{ij} = \text{Prob}(A_i | A_i \text{ or } \text{A}_\text{j})$. Is there a set of probabilities $p_i = \text{Prob}(A_i)$ that are compatible with the r_{ij}?

In general, a solution satisfying these constraints may not exist. Since $\text{Prob}(A_i|A_i \text{ or } A_j) = p_j/(p_i + p_j)$ and $\sum p_i = 1$, we are requiring that $K-1$ free parameters satisfy $K(K-1)/2$ constraints and, this will not have a solution in general. For example, if the r_{ij} are the ijth entries in the matrix

$$\begin{pmatrix} \cdot & 0.9 & 0.4 \\ 0.1 & \cdot & 0.7 \\ 0.6 & 0.3 & \cdot \end{pmatrix} \tag{1}$$

then they are not compatible with any p_i's. This is clear since $r_{12} > .5$ and $r_{23} > .5$, but also $r_{31} > .5$.

The model $\text{Prob}(A_i|A_i \text{ or } A_j) = p_j/(p_i + p_j)$ forms the basis for the Bradley-Terry model for paired comparisons (Bradley & Terry 1952). In this paper we fit this model by maximizing a (negative) Kullback-Leibler distance criterion to find the best approximation $\hat{r}_{ij} = \hat{p}_i/(\hat{p}_i + \hat{p}_j)$ to a given set of r_{ij}'s. We carry this out at each predictor value $\mathbf{x}$, and use the estimated probabilities to predict class membership at $\mathbf{x}$.

In the example above, the solution is $\hat{\mathbf{p}} = (0.47, 0.25, 0.28)$. This solution makes qualitative sense since event A_1 "beats" A_2 by a larger margin than the winner of any of the other pairwise matches.

Figure 1 shows an example of these procedures in action. There are 600 data points in three classes, each class generated from a mixture of Gaussians. A linear discriminant model was fit to each pair of classes, giving pairwise probability estimates r_{ij} at each $\mathbf{x}$. The first panel shows Friedman's procedure applied to the pairwise rules. The shaded regions are areas of indecision, where each class wins one vote. The coupling procedure described in the next section was then applied, giving class probability estimates $\hat{\mathbf{p}}(\mathbf{x})$ at

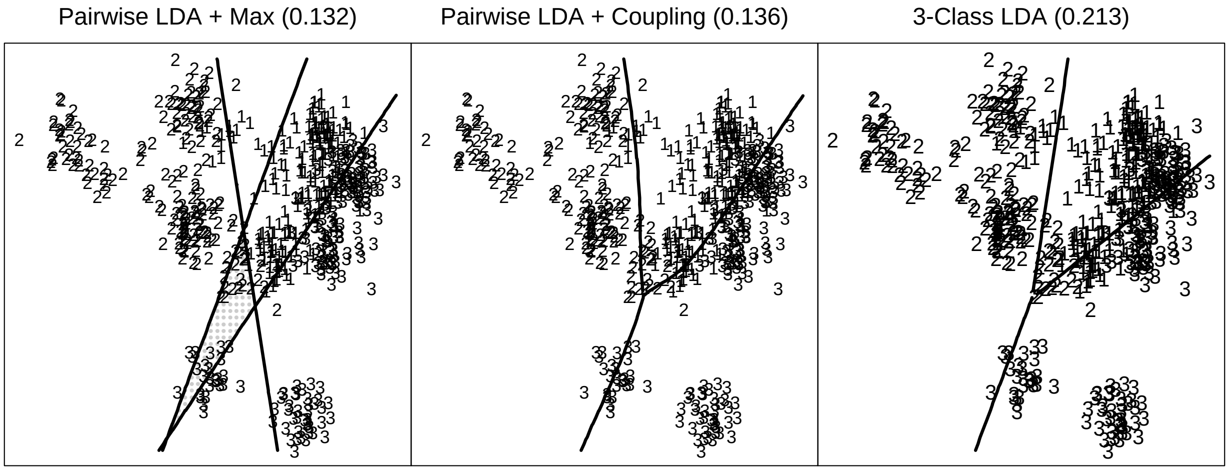

Figure 1: *A three class problem, with the data in each class generated from a mixture of Gaussians. The first panel shows the maximum-win procedure. The second panel shows the decision boundary from coupling of the pairwise linear discriminant rules based on $\hat{d}$ in (6). The third panel shows the three-class LDA boundaries. Test-error rates are shown in parentheses.*

each $\mathbf{x}$. The decision boundaries resulting from these probabilities are shown in the second panel. The procedure has done a reasonable job of resolving the confusion, in this case producing decision boundaries similar to the three-class LDA boundaries shown in panel 3. The numbers in parentheses above the plots are test-error rates based on a large test sample from the same population. Notice that despite the indeterminacy, the max-wins procedure performs no worse than the coupling procedure, and both perform better than LDA. Later we show an example where the coupling procedure does substantially better than max-wins.

Often the pairwise approach yields a more flexible class of models than a K-class method. For example, a stanadrd linear discriminant analysis (LDA) assumes that all classes have the same covariance. In the pairwise application of LDA, this assumption is used only for each pair of classes.

This paper is organized as follows. The coupling model and algorithm are given in section 2. Section 3 discusses the properties of the coupling solution, and its relation to the max-wins rule. Pairwise threshold optimization, a key advantage of the pairwise approach, is discussed in section 4. Section 5 examines the performance of the various methods on some real and simulated problems. In section 6 we apply coupling to an adaptive, nonlinear classifier based on additive modelling. Application to the support vector machine is described in section 7, while sections 8 and 9 look at coupling applied to nearest neighbour rules. The final section contains some discussion.

2 Coupling the probabilities

Let the probabilities at feature vector $\mathbf{x}$ be $\mathbf{p}(\mathbf{x}) = (p_1(\mathbf{x}), \ldots p_K(\mathbf{x}))$. In this section we drop the argument $\mathbf{x}$, since the calculations are done at each $\mathbf{x}$ separately.

We assume that for each $i \neq j$, there are n_{ij} observations in the training set and from these we have estimated conditional probabilities $r_{ij} = \mathrm{Prob}(i|i \text{ or } j)$.

Our model is

$$\mu_{ij} = \frac{p_i}{p_i + p_j} \tag{2}$$

or equivalently

$$\log \mu_{ij} = \log(p_i) - \log(p_i + p_j), \tag{3}$$

a *log-nonlinear model*

We wish to find $\hat{p}_i$'s so that the $\hat{\mu}_{ij}$'s are close to the r_{ij}'s. There are $K-1$ independent parameters but $K(K-1)/2$ equations, so it is not possible in general to find $\hat{p}_i$'s so that $\hat{\mu}_{ij} = r_{ij}$ for all i, j.

Therefore we must settle for $\hat{\mu}_{ij}$'s that are close to the observed r_{ij}'s. Our closeness criterion is the average (negative, weighted) Kullback-Leibler distance between r_{ij} and μ_{ij}:

$$\ell(\mathbf{p}) = \sum_{i<j} n_{ij} \left[r_{ij} \log \frac{r_{ij}}{\mu_{ij}} + (1 - r_{ij}) \log \frac{1 - r_{ij}}{1 - \mu_{ij}} \right] \tag{4}$$

and we find $\mathbf{p}$ to maximize this function.

This model and criterion is formally equivalent to the Bradley-Terry model for preference data. One observes a proportion r_{ij} of n_{ij} preferences for item i, and the sampling model is binomial:

$$n_{ij} r_{ij} \sim Bin(n_{ij}, \mu_{ij}).$$

If each of the r_{ij} were independent, then $\ell(\mathbf{p})$ would be equivalent to the log-likelihood under this model. However our r_{ij} are not independent as they share a common training set and were obtained from a common set of classifiers. Furthermore the binomial models do not apply in this case; the r_{ij} are evaluations of functions at a point, and the randomness arises in the way these functions are constructed from the training data. We include the n_{ij} as weights in (4); this is a crude way of accounting for the different precisions in the pairwise probability estimates.

The score (gradient) equations are:

$$\sum_{j \neq i} n_{ij} \mu_{ij} = \sum_{j \neq i} n_{ij} r_{ij}; \; i = 1, 2, \ldots K \tag{5}$$

subject to $\sum p_i = 1$. We use the following iterative procedure to compute the $\hat{p}_i$'s:

Algorithm

1. Start with some guess for the $\hat{p}_i$, and corresponding $\hat{\mu}_{ij}$.

2. Repeat $(i = 1, 2, \ldots, K, 1, \ldots)$ until convergence:

$$\hat{p}_i \leftarrow \hat{p}_i \cdot \frac{\sum_{j \neq i} n_{ij} r_{ij}}{\sum_{j \neq i} n_{ij} \hat{\mu}_{ij}}$$

   renormalize the $\hat{p}_i$, and recompute the $\hat{\mu}_{ij}$.

3. $\hat{\mathbf{p}} \leftarrow \hat{\mathbf{p}} / \sum \hat{p}_i$

The algorithm also appears in Bradley & Terry (1952). The updates in step 2 attempt to modify $\mathbf{p}$ so that the sufficient statistics match their expectation, but go only part of the way. We prove in the appendix that $\ell(\mathbf{p})$ increases at each step. Since $\ell(\mathbf{p})$ is bounded above by zero, the procedure converges. At convergence, the score equations are satisfied, and the $\hat{\mu}_{ij}$s and $\hat{\mathbf{p}}$ are consistent. This algorithm is similar in flavour to the Iterative Proportional Scaling (IPS) procedure used in log-linear models. IPS has a long history, dating back to Deming & Stephan (1940). Bishop, Fienberg & Holland (1975) give a modern treatment and many references.

The resulting classification rule is

$$\hat{d}(\mathbf{x}) = \text{argmax}_i [\hat{p}_i(\mathbf{x})] \tag{6}$$

3 Properties of the solution

The weights n_{ij} in (4) can improve the efficiency of the estimates a little, but do not have much effect unless the class sizes are very different. For simplicity, and to facilitate comparison with other techniques, in this section we assume equal weighting ($n_{ij} = 1$ for all i, j). In the examples later in the paper, we experimented with more sophisticated weights such as $n_{ij}/(\mu_{ij}(1-\mu_{ij}))$ but these made very little difference in practice.

A simple non-iterative estimate can be obtained from the row averages

$$\tilde{p}_i = \frac{2}{K}\frac{\sum_{j\neq i} r_{ij}}{(K-1)}. \tag{7}$$

These estimates can be derived as an approximation to the identity

$$p_i = \sum_{j\neq i}\left(\frac{p_i+p_j}{K-1}\right)\left(\frac{p_i}{p_i+p_j}\right) \tag{8}$$

by replacing $p_i + p_j$ in the first ratio by $2/K$, and each of the second ratios by their corresponding r_{ij}. We use these estimates as starting values in the maximum likelihood procedure. In fact, the $\tilde{p}_i$'s are in the same order as the $\hat{p}_i$'s, and hence are sufficient if only the classification rule is required:

Theorem 1: $\tilde{p}_i > \tilde{p}_j$ if and only if $\hat{p}_i > \hat{p}_j$.

Proof: The $\hat{p}_i$ satisfy $\sum_{k\neq i}\hat{\mu}_{ik} = \sum_{k\neq i} r_{ik}$. Now

$$\begin{aligned}\tilde{p}_i > \tilde{p}_j \quad &\Longleftrightarrow \quad \sum_{k\neq i} r_{ik} > \sum_{k\neq j} r_{jk}\\ &\Longleftrightarrow \quad \sum_{k\neq i}\hat{\mu}_{ik} > \sum_{k\neq j}\hat{\mu}_{jk}\\ &\Longleftrightarrow \quad \hat{p}_i > \hat{p}_j\end{aligned}$$

since $p/(p+q)$ is an increasing function of p for $q > 0$. Similarly, one can show that $\tilde{p}_i = \tilde{p}_j$ if and only if $\hat{p}_i = \hat{p}_j$.

Looking at this more closely, we find that the approximate solution $\tilde{\mathbf{p}} = (\tilde{p}_1, \ldots \tilde{p}_K)$ tends to underestimate differences between the $\hat{p}_i$'s. Specifically, the following result shows that $\tilde{\mathbf{p}}$ is closer to the equi-probability vector $(1/K, 1/K, \ldots 1/K)$ in Kullback-Leibler distance than is $\hat{\mathbf{p}}$.

Theorem 2:

$$\sum_i (1/K) \log[1/K\tilde{p}_i] \leq \sum_i (1/K) \log[1/K\hat{p}_i]$$

The proof is given in the Appendix. Given the equivalence of coupling with the Bradley-Terry model, both of these results may already be known in the literature on paired comparisons.

We now take a closer look at Friedman's rule of assigning to the class that wins the most pairwise comparisons with the other classes. Let $I_{ij} = 1$ if $r_{ij} \geq 0.5$ and 0 otherwise. Then we define

$$\begin{aligned} \tilde{p}_i &= \frac{2}{K} \frac{\sum_{j \neq i} I_{ij}}{(K-1)} \\ \tilde{d} &= \text{argmax}_i [\tilde{p}_i] \end{aligned} \tag{9}$$

Theorem 1 tells us that if we start with the I_{ij}'s rather than the r_{ij}'s, then the rules $\hat{d}$ and $\tilde{d}$ assign to the same class.

A second scenario in which they agree is the case where the model $r_{ij} = p_i/(p_i + p_j)$ holds exactly $\forall i, j$ for some p_i. For then $\hat{p}_i = p_i$, and both procedures classify to the largest

p_i, whether or not these are the correct probabilities.

In general, however, some surprising things can occur. Here is a situation where $r_{1j} > 1/2$ for all $j \neq 1$, but $\hat{p}_1$ is not largest:

$$\{r_{ij}\} = \begin{pmatrix} \cdot & 0.56 & 0.51 & 0.60 \\ 0.44 & \cdot & 0.96 & 0.44 \\ 0.49 & 0.04 & \cdot & 0.59 \\ 0.40 & 0.56 & 0.41 & \cdot \end{pmatrix} \tag{10}$$

The solution is $\hat{\mathbf{p}} = c(0.29, 0.34, 0.16, 0.21)$ and

$$\{\hat{\mu}_{ij}\} = \begin{pmatrix} \cdot & 0.46 & 0.64 & 0.57 \\ 0.54 & \cdot & 0.67 & 0.62 \\ 0.36 & 0.33 & \cdot & 0.44 \\ 0.43 & 0.38 & 0.56 & \cdot \\ & & & \end{pmatrix} \tag{11}$$

Here is an example where the classes have an ordering $i > j > k > \ell$ in the sense that $r_{ij} > .5$ for all i, j with $i < j$,

$$\{r_{ij}\} = \begin{pmatrix} \cdot & 0.51 & 0.53 & 0.51 \\ 0.49 & \cdot & 0.54 & 0.55 \\ 0.47 & 0.46 & \cdot & 0.59 \\ 0.49 & 0.45 & 0.41 & \cdot \end{pmatrix} \tag{12}$$

but the solution $\hat{\mathbf{p}} = (0.262, 0.270, 0.254, 0.214)$ does not respect this ordering.

Figure 2 shows another example similar to 1, where we can compare the performance of the rules $\hat{d}$ and $\tilde{d}$. The hatched area in the top left panel is an indeterminate region

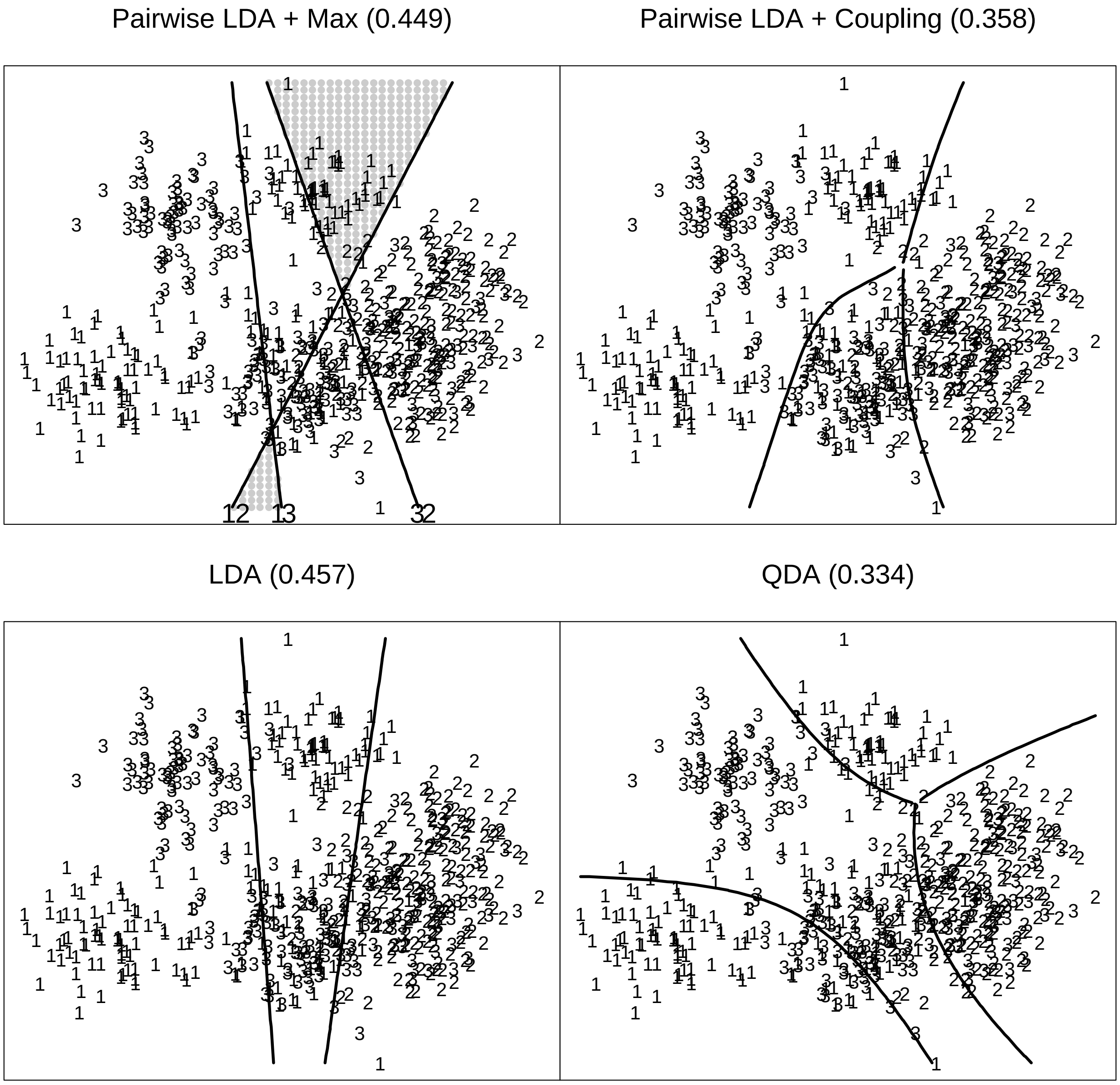

Figure 2: *A three class problem similar to that in figure 1, with the data in each class generated from a mixture of Gaussians. The first panel shows the maximum-wins procedure $\tilde{d}$ in (9). The second panel shows the decision boundary from coupling of the pairwise linear discriminant rules based on $\hat{d}$ in (6). The third panel shows the three-class LDA boundaries, and the fourth the QDA boundaries. The numbers in the captions are the error rates based on a large test set from the same population.*

where there is more than one class achieving $\max(\tilde{p}_i)$. In the top right panel the coupling procedure has resolved this indeterminacy in favor of class 1 by weighing the various probabilities.

There is another interesting phenomenon occurring here — the coupling has *reversed* a decision made by the max-win rule. Notice that in the top-left panel, the region to the left of the upper shaded wedge is a class-3 region, while in the top-right panel this is a class-1 region. Picking a point within this region, we see the matrix of r_{ij}:

$$\{r_{ij}\} = \begin{pmatrix} \cdot & 0.98 & 0.46 \\ 0.02 & \cdot & 0.30 \\ 0.54 & 0.70 & \cdot \end{pmatrix} \tag{13}$$

Class 3 narrowly wins against class 1, while class 1 beats 2 far more resoundly than does class 3. In this example the coupling has improved the misclassification rate (numbers in parentheses in plots) dramatically over both the max-win and LDA procedures. However QDA performs a little better in this example.

The rule max-wins $\tilde{d}$ may also suffer from excess variability, compared to the coupling rule $\hat{d}$. To investigate this, we performed a simple experiment. We defined class probabilities $p_1 = s/K, p_j = (1 - s/K)/(K-1), j = 2, 3, \dots K$. Then we set

$$\begin{aligned} r_{ij} &= \frac{p_i}{p_i + p_j} + 0.1 \cdot z_{ij} \\ r_{ji} &= 1 - r_{ij}; \; ; \; j > i \end{aligned} \tag{14}$$

Here z_{ij} is a standard normal variate, and r_{ij} was truncated at zero below and 1 above. We tried the values $s = 1.5$ and $s = 1.1$ In both scenarios, class 1 has higher probability and hence is the correct class. Figure 3 shows the average number of times that class 1

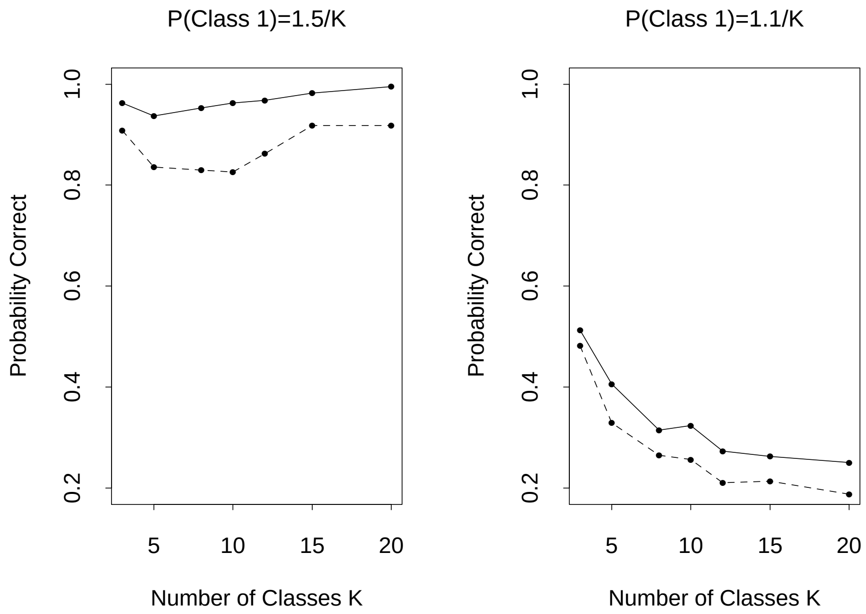

Figure 3: *Probability of predicting the true class for the rules $\hat{d}$ (solid) and $\tilde{d}$ (broken). See the text for details of the problems.*

was selected by the rules $\hat{d}$ (solid) and $\tilde{d}$ (broken). The averages are over 1000 simulations and have a standard error of about 0.01. The number of classes varies along the horizontal axis. We see that the $\hat{d}$ rule outperforms $\tilde{d}$ by about 10% when $s = 1.5$ and 6% when $s = 1.1$.

4 Pairwise threshold optimization

As pointed out by Friedman (1996*a*), approaching the classification problem in a pairwise fashion allows one to optimize the classifier in a way that would be computationally burdensome for a K-class classifier. While the number of computations for a full optimization is often proportional to K^3, the total required for $K(K-1)/2$ optimizations would be proportional to K^2. Here we discuss optimization of the classification threshold.

For each two class problem, let logit $p_{ij}(x) = d_{ij}(x)$. Normally we would classify to class i if $d_{ij}(x) > 0$. Suppose we find that $d_{ij}(x) > t_{ij}$ is better. Then we define $d'_{ij}(x) = d_{ij}(x) - t_{ij}$, and hence $p'_{ij}(x) = \text{logit}^{-1} d'_{ij}(x)$. We do this for all pairs, and then apply the coupling algorithm to the $p'_{ij}(x)$ to obtain probabilities $p'_i(x)$.

In this way we can optimize over $K(K-1)/2$ parameters separately, rather than optimize jointly over K parameters. An example of the benefit of threshold optimization is given in the next section.

5 Examples

A three class problem.

Here we define three classes in the plane as follows: X_1, X_2 were generated uniformly in the square $[-2,2] \times [-2,2]$. We define centers (0,2), $(-\sqrt{2}, -\sqrt{2})$, and $(\sqrt{2}, -\sqrt{2})$. Then if d_j^2 is the distance from a point to the jth center, a point is assigned to the class j satisfying $\text{argmin}[d_j^2 - t_j]$, where $t_1 = 2\log(.05), t_2 = 2\log(.20), t_3 = 2\log(.75)$. Each class has 100 observations. This example is constructed so that the usual linear discriminant threshold $2\log(1/3)$ is not optimal.

The data are shown in Figure 4 along with the decision boundary from pairwise coupling of LDA (solid). Threshold optimization was used in each two-class LDA. The broken line shows the boundary from standard 3-class LDA. The threshold optimization has accurately captured the boundary.

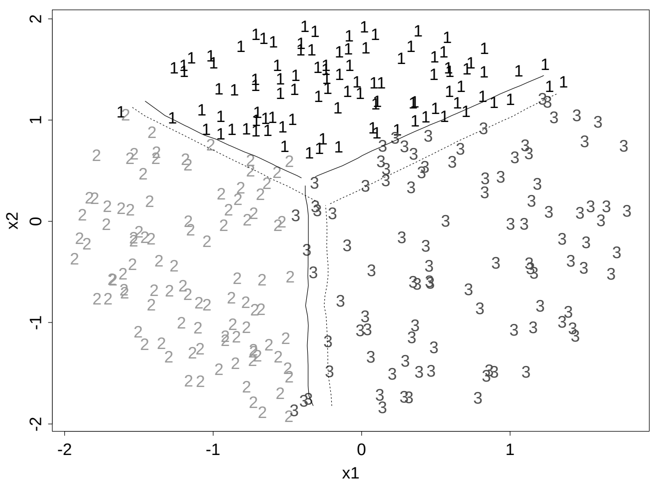

Figure 4: *Simulated three class problem, showing pairwise coupled linear rule with threshold optimization (solid) and standard three-class linear discriminant rule (broken). See text for details of the simulation.*

Various datasets.

Table 2 shows the error rates for the three class problem and a number of other datasets. The classifiers used are:

- *LDA* — linear discriminant analysis
- *QDA* — quadratic discriminant analysis
- *Max* — the rule $\tilde{d}$ from (9)
- *Max/thresh* — the rule $\tilde{d}$ with threshold optimization. The threshold was found to minimize the training error in the two classes, over a grid of possible values.
- *Coupled* — the rule $\hat{d}$ from (6)
- *Coupled/thresh* — the rule $\hat{d}$ with threshold optimization as above.

A summary of the datasets is given in Table 1. The real datasets are available from the machine learning archive at the University of California at Irvine — `ftp://ics.uci.edu`, with the exception of the digits dataset. This consists of 25 constructed features for the classification of handwritten digits 0-9, and is available from the authors.

The pairwise procedures all outperform linear discriminant analysis for most of these problems. Threshold optimization seems to improve performance both for Friedman's max rule and the coupling rule. We note that quadratic discriminant analysis does nearly as well as pairwise coupling for these problems.

Table 1: *Summary of datasets*

Dataset	# Training	# Test	# classes	# features
Vowel	528	462	11	10
Waveform	300	500	3	21
Vehicle	423	423	4	18
Crabs	80	120	4	5
Digits	1000	1000	10	25
Three class	300	300	3	2

6 Example: adaptive nonlinear classification

There have been many proposals for adaptive estimation of nonlinear regression surfaces. Some proposals, such as additive models (Hastie & Tibshirani (1990)(and MARS (Multivariate additive regression splines; Friedman (1991)), cannot be applied in straightforward way to classification problems.

In this section, we propose a new way of generalizing adaptive regression, via the pairwise coupling idea. We start with a simple global procedure, such as LDA or multiple logistic regression. Here we used LDA. Then we find the pair of classes with the highest confusion rate, that is, if $e(j,k)$ is the proportion of times that a class j observation is classified as class k, we find j and k to maximize $e(j,k)+e(k,j)$. Then for only classes j and k (coded 0 and 1), we apply an adaptive regression procedure. All other pairs are modelled via a linear regression. Pairwise coupling is applied to update the joint class

Table 2: *Training errors (top row) and test errors (bottom row) for different examples. Values are mean (standard errors) over 5 simulations.*

Dataset	LDA	QDA	Max	Max/thresh	Coupled	Coupled/thresh
Vowel	.296(.020)	.023(.002)	.132(.013)	.112(.012)	.128(.013)	.118(.015)
	.500(.018)	.490(.014)	.479(.019)	.489(.02)	.480(.017)	.473(.02)
Waveform	.148(.011)	.052(.005)	.121(.010)	.115(.008)	.120(.010)	.115(.008)
	.214(.006)	.220(.009)	.176(.006)	.174(.009)	.173(.005)	.172(.007)
Vehicle	.192(.004)	.073(.004)	.165(.002)	.157(.002)	.165(.003)	.158(.002)
	.233(.006)	.158(.008)	.210(.008)	.213(.008)	.209(.008)	.213(.010)
Crabs	.045(.005)	.038(.003)	.039(.004)	.027(.005)	.039(.004)	.027(.005)
	.051(.006)	.060(.007)	.063(.010)	.063(.012)	.063(.010)	.063(.012)
Digits	.047	.005	.020	.009	.018	.008
	.082	.076	.055	.055	.053	.053
Three class	.065(.004)	.032(.004)	.069(.004)	.030(.002)	.069(.004)	.031(.002)
	.063(.004)	.029(.003)	.068(.005)	.039(.007)	.068(.005)	.035(.007)

probabilities, and the procedure is repeated for some fixed number of iterations (we used 5 iterations below). At each stage, we find the pair of linearly modelled classes wit the highest error rate, and allow a nonlinear model for that pair.

An advantage of this approach is that complex, nonlinear functions are used only for the classes where they are needed. This is in contrast to the methods described above, which use the same set of basis functions for all classes.

For illustration, we applied these methods to some data on vowel recognition. The nonlinear regression procedure used was adaptive backfitting for additive models, using cubic splines. See Hastie (1989) or Hastie & Tibshirani (1990), chapter 9 for details.

The vowel example is a popular benchmark for neural network algorithms, and consists of training and test data with 10 predictors and 11 classes. We obtained the data from the benchmark collection maintained by Scott Fahlman at Carnegie Mellon University. The data were contributed by Anthony Robinson (see Robinson, 1989), and he provided the following (edited) description.

An ascii approximation to the International Phonetic Association symbol and the word in which the eleven vowel sounds were recorded is given in table 3. The word was uttered once by each of the fifteen speakers. Four male and four female speakers were used to train the networks, and the other four male and three female speakers were used for testing the performance. 10 features were derived for each utterance, from a linear filtering of the speech signal.

In the 5 iterations of the adaptive nonlinear classification technique, the procedure built nonlinear rules for classes (9,10), (1,2), (5,6), (5,7) and (2,3). The error rate decreased and

Table 3: *Words used in Recording the Vowels*

vowel	word	vowel	word
i	heed	O	hod
I	hid	C:	hoard
E	head	U	hood
A	had	u:	who'd
a:	hard	3:	heard
Y	hud		

then levelled off. The test set confusion matrices for LDA and the nonlinear pairwise procedure are shown in (15) and (16) below. The columns represent the true class, while the rows represent predicted class.

$$
\left(
\begin{array}{r|rrrrrrrrrrr}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\
\hline
1 & 28 & 23 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & 0 \\
2 & 10 & 16 & 11 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 1 \\
3 & 1 & 2 & 16 & 2 & 0 & 5 & 1 & 0 & 0 & 5 & 2 \\
4 & 0 & 0 & 11 & 33 & 1 & 3 & 0 & 0 & 0 & 0 & 0 \\
5 & 0 & 0 & 0 & 0 & 7 & 8 & 9 & 1 & 0 & 0 & 0 \\
6 & 0 & 1 & 4 & 6 & 22 & 19 & 12 & 0 & 0 & 0 & 11 \\
7 & 0 & 0 & 0 & 0 & 9 & 1 & 11 & 2 & 0 & 0 & 1 \\
8 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 23 & 6 & 0 & 0 \\
9 & 3 & 0 & 0 & 0 & 0 & 0 & 4 & 8 & 15 & 9 & 2 \\
10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & 14 & 13 & 1 \\
11 & 0 & 0 & 0 & 1 & 3 & 6 & 1 & 0 & 5 & 6 & 24
\end{array}
\right) \tag{15}
$$

$$
\left(
\begin{array}{r|rrrrrrrrrrr}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\
\hline
1 & 30 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 \\
2 & 8 & 32 & 3 & 0 & 0 & 0 & 3 & 0 & 1 & 6 & 3 \\
3 & 1 & 0 & 25 & 3 & 0 & 5 & 0 & 0 & 1 & 0 & 0 \\
4 & 0 & 0 & 10 & 31 & 1 & 2 & 1 & 0 & 0 & 0 & 0 \\
5 & 0 & 0 & 0 & 0 & 13 & 9 & 13 & 0 & 0 & 0 & 0 \\
6 & 0 & 0 & 4 & 7 & 16 & 20 & 5 & 0 & 0 & 0 & 10 \\
7 & 0 & 0 & 0 & 0 & 9 & 0 & 13 & 4 & 0 & 0 & 2 \\
8 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 21 & 5 & 0 & 0 \\
9 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 17 & 29 & 7 & 2 \\
10 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 18 & 1 \\
11 & 0 & 1 & 0 & 1 & 3 & 6 & 0 & 0 & 5 & 6 & 24
\end{array}
\right) \tag{16}
$$

The nonlinear construction has successfully reduced the error rates, especially for classes (9, 10) and (1, 2). The overall training and test error rates are shown in Table 4: the pairwise approach produces a substantial improvement over LDA.

This problem was reasonably large (528 training cases, 10 classes, 11 features). The computations for this example took about a minute on a Silicon Graphics Challenge Series R10000 computer. Computations for another examples, all of which used a simple classifier applied to each of the $K(K-1)/2$ problems, took less than a minute. However if we were to apply a nonlinear classifier (like adaptive additive models) to every pairwise problem, the computation would increase considerably. Hence the advantage of the adaptive approach of this section, in which we apply the nonlinear classifier only to the pairs that need it.

Hastie, Tibshirani & Buja (1994) discuss two other approaches for adapting regression procedures to classification problems. The first is to construct a multi-response version of the regression procedure, simultaneously modelling K outcomes in terms of a set of optimally chosen basis functions. Applying such a procedure to an indicator matrix coding the responses, one obtains a K-vector of fitted values for each observation. One can then classify the observation to the class have largest fitted value. This idea has become known as "softmax" in the machine learning literature. In the experiments in Hastie et al. (1994), this procedure did not work particularly well, and there we demonstrate a kind of masking that can occur when the classes are linearly aligned in feature space. The other approach is to apply linear discriminant analysis in the space of fitted values from the multi-response regression above. This is called "flexible discriminant analysis" (FDA) in Hastie et al. (1994): the theory of this technique exploits the connection between linear discriminant analysis and optimal scoring. The error rates for softmax and optimal scoring on the vowel data are 50% and 44% respectively, as given in Hastie et al. (1994).

Table 4: *Vowel recognition results*

Technique	Error rates	
	Training	Test
LDA	0.32	0.56
Adaptive backfitting/pairwise coupling	0.16	0.44

7 The support vector machine

Boser, Guyon & Vapnik (1992) proposed a two-class classifier that finds the hyperplane maximizing the minimum (signed) distance between the plane and the training points. Specifically, the norm vector of the hyperplane $\mathbf{w}\cdot\mathbf{x}+b$ is found by minimizing the functional

$$\begin{aligned} J(\mathbf{w},\xi) &= \frac{1}{2}||\mathbf{w}||^2 + \gamma\sum_1^N \xi_i \\ &\quad \text{subject to } y_i(\mathbf{w}\cdot\mathbf{x}_i) + b \geq 1 - \xi_i \end{aligned} \tag{17}$$

Here the outcome y is coded -1 and 1. The classifier predicts class 2 if

$$\mathbf{w}^T\mathbf{x} + b > 0 \tag{18}$$

and class 1 if < 0. Because of the nature of the criterion $J(\mathbf{w},\xi)$, the solution vector $\hat{\mathbf{w}}$ is a linear combination of a subset of the feature vectors $\mathbf{x}_i$, called the "support vectors". See Vapnik (1996) for a complete discussion. The intercept b is found by minimizing the training error. Normally one would optimize the choice of the regularization parameter γ in (17) for a given problem, but for simplicity and fairness to other procedures considered here, we used the fixed value of $\gamma = 5$.

The support vector machine has shown very promising results in some real-world problems (Vapnik, personal communication). However there seems to be no simple multi-class version, so this is an attractive candidate for the pairwise coupling procedure. To proceed, we need to obtain class probability estimates from the support-vector machine, which we do as follows: Define $z = \mathbf{a}^T\mathbf{x}$, and let m_1, m_2 be the means of z in each class. Let $m = (m_1 + m_2)/2$ and s be the standard deviation of $z - m$. Then we define $m_1' = b - m, m_2' = b + m$ and

$$\begin{aligned} f_1(z) &= \phi(m_1', s) \\ f_2(z) &= \phi(m_2', s) \end{aligned} \tag{19}$$

where $\phi(\mu, \sigma)$ denotes the Gaussian density with mean μ and standard deviation σ. This construction satisfies $f_1(z) > f_2(z)$ if $z < b$ and $f_1(z) < f_2(z)$ if $z > b$, and so is consistent with the classification rule (18).

The results of the multiclass support vector machine are shown in Table 5. The SV/Max used the rule $\tilde{d}$ to combine the pairwise classifications, while SV/coupled use the coupling rule $\hat{d}$. Overall it performs about as well as the coupled linear discriminant method.

8 Experiments with nearest neighbors

A J-nearest neighbor classifier chooses the majority class among the J closest training points to the target point. Typically Euclidean distance $||\mathbf{x} - \mathbf{x}_i|| = (\mathbf{x} - \mathbf{x}_i)^T(\mathbf{x} - \mathbf{x}_i)$ is used to measure distance between the test input $\mathbf{x}$ and the training inputs $\mathbf{x}_i$.

Table 5: *Results for support vector machine. Figures are training and test error rates for a single realization, except for the three-class problem where mean (standard error) over 10 simulations is given.*

Data	LDA	SV/Max	SV/Coupled
Vowel	.316	.097	.097
	.556	.470	.450
Waveform	.153	.067	.067
	.208	.206	.206
Vehicle	.213	.170	.170
	.206	.222	.209
Crabs	.050	.025	.025
	.067	.058	.075
Digits	.047	.022	.023
	.082	.066	.063
3 class	.065(.004)	.020 (.004)	.020 (.003)
	.063(.004)	.026 (.004)	.025 (.003)
Ave % test error			
improvement vs LDA		7.3	15.5

We can view J-nearest neighbors as follows. For each class k we construct class probability estimates

$$\hat{p}_k(\mathbf{x}) = \frac{1}{J}\sum_{i=1}^{N} I(||\mathbf{x}-\mathbf{x}_i|| \leq d_J(\mathbf{x}))I(y_i = k) \tag{20}$$

where $d_J(\mathbf{x})$ is the Jth largest of the $||\mathbf{x}-\mathbf{x}_i||$ values. Then we classify to the class k with highest estimated probability $\hat{p}_k(\mathbf{x})$. One way to potentially improve the performance of nearest neighbors is to multiply each probability estimate by a bias factor — that is, form the estimates $\hat{p}_k(\mathbf{x})b_k$ with each b_k positive and less than 1, and then optimize over the biases $b_1, b_2, \ldots b_K$ (Friedman (1996*b*), Rosen, Burke & Goodman (1995))[1]. The joint optimization of K parameters can be computationally difficult so Friedman (1996*a*) suggested carrying this out in a pairwise fashion and then combining the rules via the max-wins procedure $\tilde{d}$.

Here is a simple example, due to Friedman (1996*b*), that illustrates how bias factors can help. It is illustrated in figure 5. We have 200 data points in each of two classes. The points in the first class are uniformly distributed in the rectangle $[0, 3/4] \times [0, 1]$ while those in the second class are uniformly distributed in the rectangle $[3/4, 1] \times [0, 1]$. Then a large circular neighbourhood, in the class 1 region but near the decision boundary, will tend to have more points in class 2 and hence will misclassify. In order to avoid this misclassification, a standard nearest neighbour rule must shrink the neighborhood. This in turn causes an increase in variance. If we instead include a bias factor of 1/3, the two class densities can be fairly compared in the neighborhood and we don't have to shrink the neighborhood.

[1]In fact, Friedman uses an additive bias $\hat{p}_k(\mathbf{x}) + t_k$; we find a multiplicative bias more natural.

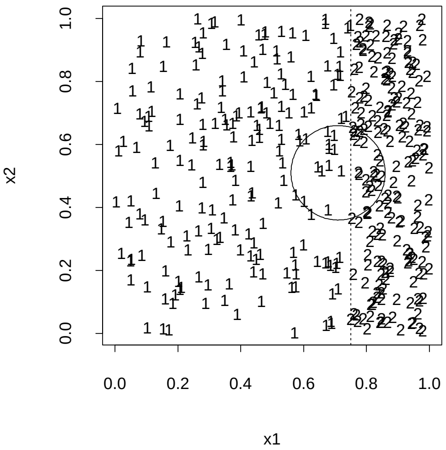

Figure 5: *Example of case where nearest neighbor biasing is needed. There are 200 data points uniformly distributed in each of the two rectangular regions, separated by the broken line. A large circular neighbourhood, centered in the class 1 region but near the decision boundary, will tend to have more points in class 2 and hence will misclassify.*

This biasing of class densities does not work for small J, because the probability estimates are too discrete. We propose instead to view nearest neighbor classification in terms of density estimation. Let $d_J^k(\mathbf{x})$ be the distance of the Jth nearest neighbor to $\mathbf{x}$ computed *separately in each class*. A natural estimate of the class-k density at $\mathbf{x}$ is

$$\hat{f}_k(\mathbf{x}) \propto \frac{J}{n_k[d_J^k(\mathbf{x})]^p}, \tag{21}$$

where p is the dimension of the space and n_k the number of training points in class k. Assuming sample priors n_k/n, the corresponding class probability estimates at $\mathbf{x}$ are

$$\hat{p}_k(\mathbf{x}) \propto \frac{1}{[d_J^k(\mathbf{x})]^p}. \tag{22}$$

Note that when $J = 1$ this is identical to the usual definition of 1-nearest neighbor classification, but not for $J \geq 2$. These estimates do not suffer from the discreteness problem, and can be modified by a bias factor just as before. These (biased) pairwise probabilities are then combined using the coupling procedure (6).

9 Example: ten Gaussian classes with unequal covariance

In this simulated example taken from Friedman (1996*a*), there are 10 Gaussian classes in 20 dimensions. The mean vectors of each class were chosen as 20 independent uniform $[0, 1]$ random variables. The covariance matrices are constructed from eigenvectors whose square roots are uniformly distributed on the 20-dimensional unit sphere (subject to being mutually orthogonal), and eigenvalues uniform on $[0.01, 1.01]$. There are 100 observations per class in the training set, and 200 per class in the test set. The optimal decision boundaries in this problem are quadratic, and neither linear nor nearest-neighbor methods are well-suited. Friedman states that the Bayes error rate is less than 1%.

Figure 6 shows the test error rates for linear discriminant analysis, J-nearest neighbor and their paired versions using threshold optimization. We see that the coupled classifiers nearly halve the error rates in each case. In addition, the coupled rule works a little better than Friedman's max rule in each task. Friedman (1996a) reports a median test error rate of about 16% for his thresholded version of pairwise nearest neighbor.

Why does the pairwise thresholding work in this example? We looked more closely at the pairwise nearest neighbour rules rules that were constructed for this problem. The thresholding biased the pairwise distances by about 7% on average. The average number of nearest neighbours used per class was 4.47 (.122), while the standard J-nearest neighbour approach used 6.70 (.590) neighbours for all ten classes. For all ten classes, the 4.47 translates into 44.7 neighbours. Hence relative to the standard J-NN rule, the pairwise rule, in using the threshold optimization to reduce bias, is able to use about six times as many near neighbours.

10 Discussion

Geoffrey Hinton suggested that pairwise approaches to classification might suffer from the following problem. Suppose for example we are classifying handwritten digits (0-9), and one digit (say 0) tends to be closer on average in feature space to a randomly chosen digit image than are other digits. At prediction time, a test image (say a poorly written 9)

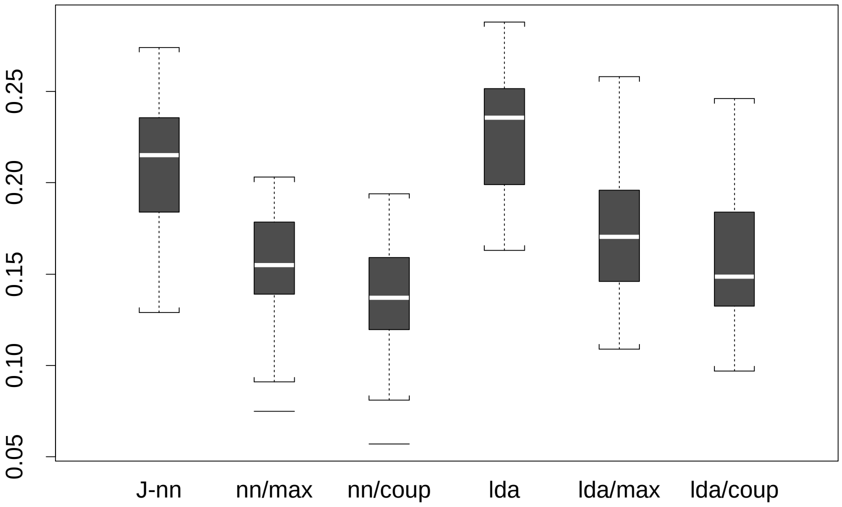

Figure 6: *Test errors for 20 simulations of ten-class Gaussian example.*

is presented to every pairwise classifier (0-1, 0-2 etc). Most of these classifiers were not trained on 9s and hence might give unreliable pairwise conditional probabilities. If the 9 classifier doesn't give high enough conditional probabilities, then the 0 digit might win because it tends to receive higher probability for most random digits. The point is that it may be bad to predict from pairwise classifiers that have not been trained on images of that type of image, so that the prediction requires an extrapolation in feature space.

To investigate the validity of this point, we modified the experiment of Figure 3. If the true class was 1, the class probabilities were $2/K$ for class 1 and $(1-2/K)/(K-1)$ for the rest. If the true class was not 1, the probabilities were $1.5/K$ for the true class, $1.2/K$ for class 1 and probabilities $(1-1.5/K-1.2/K)/(K-2)$ for the remaining classes. Hence the first class always finishes second when it is not the true class. The r_{ij} were as defined in (14), with $s = 1.5$. We generated 500 realizations from this model, choosing the true class at random from $1, 2, \ldots K$ each time.

The left panel of Figure 7 shows the probability of correct classification for the coupling rule (solid) and the max rule (broken). Comparing this to the left panel of Figure 3, we see that the existence of the popular class 1 has increased the error rate from 4% to about 16% when $K = 3$, with less of an increase for larger K. The max rule does consistently worse than the coupling rule.

This simulation suggests that Hinton's suggested problem may be real. However it is not clear whether other (non-pairwise) approaches would fare any better. In addition, when one estimates probabilities, all is not lost. The right panel shows boxplots of the maximum class probability from the coupled classifier with $K = 5$, stratified by whether the classification is correct or not. Not surprisingly, when the classifier errs, it tends to be less sure about its prediction. If one is willing to "punt", i.e., decide not to classify at all, based on the magnitude of the maximum class probability, the results improve. For example, if we punt whenever the maximum class probability is below its 5% point, the error rate decreases from 16% to 12%.

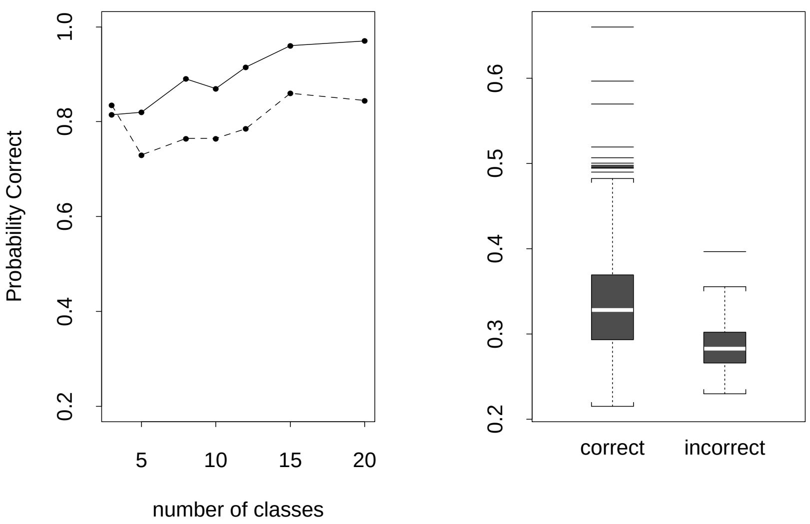

Figure 7: *Left panel: probability of predicting the true class for the rules* $\hat{d}$ *(solid) and* $\tilde{d}$ *(broken). Right panel: maximum class probability from the coupled classifier with* $K = 5$*, stratified by whether the classification is correct or not. See the text for details of the problem.*

Suppose the pairwise classifiers provide not only conditional probability estimates r_{ij} but also estimates of the variance of r_{ij}, say v_{ij}. Then we can use these variances as reciprocal weights in the coupling algorithm. Specifically, we replace the n_{ij} by n_{ij}/v_{ij} in the algorithm. In theory, this should help the extrapolation problem mentioned above: a point in class k that is far away from the training data for classes i and j will have high estimated variance for classifier i, j and hence its r value will be downweighted. However, our experiments with this approach, using pairwise linear classifiers, did not improve upon the results for the unweighted coupling procedure. A more refined approach would also incorporate the covariances of the r_{ij}'s into the model, but we have not pursued this.

The pairwise procedures, both Friedman's max-win and our coupling, are most likely to offer improvements when additional optimization or efficiency gains are possible in the simpler 2-class scenarios. In some situations they perform exactly like the multiple class classifiers. Two examples are:

- each of the pairwise rules are based on QDA: i.e. each class modelled by a Gaussian distribution with separate covariances, and then the r_{ij}s derived from Bayes rule.

- A generalization of the above, where the density in each class is modelled in some fashion, perhaps nonparametrically via density estimates or near-neighbor methods, and then the density estimates are used in Bayes rule.

Pairwise LDA followed by coupling seems to offer a nice compromise between LDA and QDA, although the decision boundaries are no longer linear. For this special case one might derive a different coupling procedure globally on the *logit* scale, which would guarantee linear decision boundaries. Work of this nature is currently in progress with Jerry Friedman.

Acknowledgments

We thank Jerry Friedman for sharing a preprint of his pairwise classification paper with us, and acknowledge helpful discussions with Jerry, Geoff Hinton, Radford Neal and David Tritchler. Comments by an editor and two referees led to valuable improvements in the manuscript. Trevor Hastie was partially supported by grant DMS-9504495 from the National Science Foundation, and grant ROI-CA-72028-01 from the National Institutes of Health. Rob Tibshirani was supported by the Natural Sciences and Engineering Research Council of Canada and the IRIS Centre of Excellence.

Appendix

Convergence of the algorithm

The effect of the update in step 2 of the algorithm (for a single i) is

$$\begin{aligned}
\alpha &= \frac{\sum_{j\neq i} n_{ij} r_{ij}}{\sum_{j\neq i} n_{ij}\hat{\mu}_{ij}} \\
\hat{\mu}_{ij} &\to \frac{\alpha\hat{\mu}_{ij}}{\alpha\hat{\mu}_{ij}+\hat{\mu}_{ji}} \\
\hat{\mu}_{ji} &\to \frac{\hat{\mu}_{ji}}{\alpha\hat{\mu}_{ij}+\hat{\mu}_{ji}} \\
p_i &\to p_i' = \alpha p_i
\end{aligned} \tag{23}$$

The resulting change in $\ell(\mathbf{p})$ is

$$\ell(\mathbf{p}') - \ell(\mathbf{p}) = \left[\sum_{j\neq i} n_{ij} r_{ij}\right] \log \frac{\sum_{j\neq i} n_{ij} r_{ij}}{\sum_{j\neq i} \hat{n}_{ij}\mu_{ij}} - \sum_{j\neq i} n_{ij} \log\left(\frac{\sum_{j\neq i} n_{ij} r_{ij}}{\sum_{j\neq i} \hat{n}_{ij}\mu_{ij}}\hat{\mu}_{ij} + 1 - \hat{\mu}_{ij}\right)$$

For brevity, let $x = \sum_{j\neq i} n_{ij} r_{ij}$, $d = \sum_{j\neq i} n_{ij}\hat{\mu}_{ij}$. Then

$$\begin{aligned}
\ell(\mathbf{p}') - \ell(\mathbf{p}) &= x\log\frac{x}{d} - \sum_{j\neq i} n_{ij}\log[(\frac{x}{d}-1)\hat{\mu}_{ij}+1] \\
&\geq x\log\frac{x}{d} - \sum_{j\neq i} n_{ij}(\frac{x}{d}-1)\hat{\mu}_{ij} \\
&= x\log\frac{x}{d} - (x-d) \\
&\geq 0
\end{aligned} \tag{24}$$

In the second line above, we have used the inequality $\log(1+x) \leq x$ for $x > -1$. In the last line, we have used the inequality $x\log(x/y) \geq x - y$ for $x, y \geq 0$ which can be verified by noting that at the stationary points $x = y$ it has value 0 and the Hessian is positive definite. Note that equality holds when $x = d$, that is, $\sum_{j\neq i} n_{ij} r_{ij} = \sum_{j\neq i} n_{ij}\hat{\mu}_{ij}$.

Therefore the log-likelihood increases at each step. Since it is bounded above by 0, the algorithm converges.

Note that this algorithm differs from standard iterative proportional scaling, in that it doesn't minimize over each p_i at each iteration. Due to non-linearity of the model, this would require a line search at each step, However it does increase the likelihood at each iteration and converges quite quickly in practice.

Theorem 2:

$$\sum_i (1/K)\log[1/K\tilde{p}_i] \leq \sum_i (1/K)\log[1/K\hat{p}_i]$$

Proof: Since $\sum_{j\neq i}\hat{\mu}_{ij} = \sum_{j\neq i} r_{ij}$, we have

$$\begin{aligned}\tilde{p}_i &= \frac{2}{K(K-1)}\sum_{j\neq i}\hat{p}_i/(\hat{p}_i+\hat{p}_j) \\ &= \hat{p}_i\frac{1}{K-1}\sum_{j\neq i}\frac{\frac{2}{k}}{(\hat{p}_i+\hat{p}_j)}\end{aligned}$$

Therefore

$$\begin{aligned}\sum_i(\log\tilde{p}_i - \log\hat{p}_i) &= \sum_i \log\left[\frac{1}{K-1}\sum_{j\neq i}\frac{\frac{2}{k}}{(\hat{p}_i+\hat{p}_j)}\right] \\ &\geq 0\end{aligned}$$

In the second line above, we have used the fact that $\sum_{j\neq i} 1/(\hat{p}_i+\hat{p}_j)$ takes its minimum when the $\hat{p}_i$ are equal. The minimum is $K(K-1)/2$, and the theorem is proved.

References

Bishop, Y., Fienberg, S. & Holland, P. (1975), *Discrete multivariate analysis*, MIT Press, Cambridge.

Boser, B., Guyon, I. & Vapnik, I. (1992), A training algorithm for optimal margin classifiers, *in* 'Proceedings of COLT II', Philadelphia, Pa.

Bradley, R. & Terry, M. (1952), 'The rank analysis of incomplete block designs. i. the method of paired comparisons', *Biometrics* pp. 324–345.

Deming, W. & Stephan, F. (1940), 'On a least squares adjustment of a sampled frequency table when the expected marginal totals are known', *Ann. Math. Statist.* pp. 427–444.

Friedman, J. (1991), 'Multivariate adaptive regression splines (with discussion)', *Annals of Statistics* **19**(1), 1–141.

Friedman, J. (1996*a*), Another approach to polychotomous classification, Technical report, Stanford University.

Friedman, J. (1996*b*), Bias, variance, 0-1 loss and the curse of dimensionality, Technical report, Stanford University.

Hastie, T. (1989), 'Discussion of flexible parsimonious smoothing and additive modelling (with discussion) by friedman and silverman', *Technometrics* **31**, 3–39.

Hastie, T. & Tibshirani, R. (1990), *Generalized Additive Models*, Chapman and Hall.

Hastie, T., Tibshirani, R. & Buja, A. (1994), 'Flexible discriminant analysis by optimal scoring', *J. Amer. Statist. Assoc.* pp. 1255–1270.

Rosen, D., Burke, H. & Goodman, O. (1995), Local learning methods in high dimensions: beating the bias-variance dilemna via recalibration, *in* 'NIPS Workshop: machines that learn- Neural networks for computing'.

Vapnik, V. (1996), *The nature of statistical learning theory*, Springer-Verlag, New York.