Another Approach to Polychotomous Classification

Jerome H. Friedman*
Department of Statistics and
Stanford Linear Accelerator Center
Stanford University

October 2, 1996

Abstract

An alternative solution to the K - class ($K \geq 3$ - polychotomous) classification problem is proposed. It is a simple extension of $K = 2$ (dichotomous) classification in that a separate two-class decision boundary is independently constructed between every pair of the K classes. Each of these boundaries is then used to assign an unknown observation to one of its two respective classes. The individual class that receives the most such assignments over these $\binom{K}{2}$ decisions is taken as the predicted class for the observation. Motivation for this approach is provided along with discussion as to those situations where it might be expected to do better than more traditional methods. Examples are presented illustrating that substantial gains in accuracy can sometimes be achieved.

1. Introduction

Classification is a prediction (learning) problem in which the value of an ("output") variable y to be predicted assumes one of K unorderable categorical values $y \in \{c_1, \cdots, c_K\}$. The prediction is made based on given joint values of a set of ("input") variables $\mathbf{x} = \{x_1, \cdots, x_n\}$. The rule mapping the input values $\mathbf{x}$ to an estimated output value $\hat{y}(\mathbf{x})$ is constructed by a learning algorithm from a ("training") sample T of N previously solved cases

$$T = \{\mathbf{x}_i, y_i\}_1^N \tag{1.1}$$

for which the joint values of both input and output variables have been given.

There are a wide variety of learning algorithms [see for example Bishop (1995) and Ripley (1996)]. Nearly all of them can be viewed as procedures for obtaining estimates $\{\hat{f}_k(\mathbf{x})\}_1^K$ of the set of (conditional) probabilities

$$\{f_k(\mathbf{x}) = \Pr(y = c_k \,|\, \mathbf{x})\}_1^K \tag{1.2}$$

that the output y assumes each of its respective values, at each point $\mathbf{x}$ in the space of input values. These estimates are then inserted into the decision rule

$$\hat{k}(\mathbf{x}) = \arg \min_{1 \leq k \leq K} \sum_{l=1}^{K} L_{lk} \hat{f}_l(\mathbf{x}), \quad \hat{y}(\mathbf{x}) = c_{\hat{k}(\mathbf{x})}, \tag{1.3}$$

where L_{lk} is a (user specified) loss for predicting $\hat{y} = c_k$ when the true value is $y = c_l$ ($L_{ll} = 0$). This rule (1.3) is motivated by the fact that using the (unknown) true conditional probabilities (1.2) in

Work supported in part by the Department of Energy under contract number DE-AC03-76SF00515 and by the National Science Foundation under grant number DMS-9403804.

(1.3) results in an optimal ("Bayes") rule with the smallest possible misclassification "risk" (expected loss)

$$R = E_{\mathbf{x}}\left[\sum_{l=1}^{K} L_{l,\hat{k}(\mathbf{x})} f_l(\mathbf{x})\right]. \tag{1.4}$$

Here (1.4) the expected value (average) is over the distribution of (future) $\mathbf{x}$ values to be predicted.

The $K \times K$ loss matrix L_{lk} is seldom fully general. Usually it is taken to have some special structure such as

$$L_{lk} = L_l \, 1(l \neq k) \tag{1.5}$$

where the function $1(\cdot)$ is an indicator of the truth of its argument

$$1(\eta) = \begin{cases} 1 & \text{if } \eta \text{ is true,} \\ 0 & \text{otherwise.} \end{cases} \tag{1.6}$$

Here (1.5) the loss for misclassifying a case with (true) output value $y = c_l$ is the same irrespective of the alternative value $\hat{y} \neq c_l$ to which it is assigned. For this loss structure (1.5) the decision rule (1.3) reduces to

$$\hat{k}(\mathbf{x}) = \arg \max_{1 \leq k \leq K} L_k \hat{f}_k(\mathbf{x}). \tag{1.7}$$

A more restricted (but even more common) situation is where $L_l = 1$ in (1.5) for which the decision rule (1.3) (1.7) reduces to assigning the output value (class) estimated to be the most probable at $\mathbf{x}$, and the misclassification risk (1.4) is simply the probability of assignment error

$$R = E_{\mathbf{x}} 1(\hat{y}(\mathbf{x}) \neq y). \tag{1.8}$$

2. Conditional probability estimation

There are two general paradigms for obtaining the estimates $\{\hat{f}_k(\mathbf{x})\}_1^K$ of the conditional probabilities (1.2). The density estimation approach makes use of Bayes theorem

$$f_k(\mathbf{x}) = \Pr(y = c_k \,|\, \mathbf{x}) = \frac{\Pr(\mathbf{x} \,|\, y = c_k) \Pr(y = c_k)}{\sum_{l=1}^{K} \Pr(\mathbf{x} \,|\, y = c_l) \Pr(y = c_l)} \tag{2.1}$$

where $\Pr(\mathbf{x} \,|\, y = c_k)$ is the relative probability of observing a set of joint input values $\mathbf{x}$ given the output value $y = c_k$, and $\Pr(y = c_k)$ is the unconditional ("prior") probability of observing the value $y = c_k$. The quantity $\Pr(\mathbf{x} \,|\, y = c_k)$ is just the joint probability density function of the kth class, and can be estimated by density estimation techniques. The training sample T (1.1) is partitioned into K disjoint subsamples, each with the same output value ("class label"), and a density estimation procedure is applied to (separately) estimate $\Pr(\mathbf{x} \,|\, y = c_k)$ from each respective subsample. These estimates are then plugged into (2.1) to obtain a set of conditional (at $\mathbf{x}$) probability estimates, which in turn are used in (1.3) or (1.7) to make an output prediction. Examples of classification techniques employing this density estimation paradigm are "discriminant analysis" [see McLachlan (1992)], Gaussian mixtures [Chow and Chen (1992)], learning vector quantization techniques [Kohonen (1990)], and Bayesian belief networks [Heckerman, Geiger, and Chickering(1994)].

An alternative paradigm that attempts to directly estimate the conditional probabilities (1.2) is based on real valued output prediction ("regression"). The categorically valued output y is encoded into K numerically valued ("dummy") output variables $\{d_k = 1(y = c_k)\}_1^K$, for which one has

$$f_k(\mathbf{x}) = \Pr(d_k = 1 \,|\, \mathbf{x}) = E(d_k \,|\, \mathbf{x}). \tag{2.2}$$

This is in turn the solution to the least-squares problem

$$f_k(\mathbf{x}) = \arg \min_f E[(d_k - f)^2 | \, \mathbf{x}]. \tag{2.3}$$

This motivates applying regression methodology to estimate

$$\hat{f}_k(\mathbf{x}) = \arg\min_{f(\mathbf{x})\in F} \sum_{i=1}^{N} [d_{ik} - f(\mathbf{x}_i)]^2 \tag{2.4}$$

from the K respective training sets $\{d_{ik}, \mathbf{x}_i\}_{i=1}^N$, where F is some function class determined by the regression procedure used. The $\{\hat{f}_k(\mathbf{x})\}_1^K$ thereby obtained are then used in the decision rule (1.3) or (1.7) to make an output prediction. Examples of techniques using this (regression) paradigm are neural networks [Lippmann (1989)], decision tree induction methods [Breiman, et. al. (1984) and Quinlan (1993)], projection pursuit [Friedman (1985)], and nearest neighbor methods [Fix and Hodges (1951)] .

3. Alternative decision rule

As noted, solution to classification problems generally involves a two-step process: conditional probability estimation (Section 2) followed by a decision rule [(1.3) or (1.7)] using those estimates. In this paper a different decision rule is proposed as an alternative to (1.7). That is, it is only applicable for the simpler loss structure (1.5). It is motivated by the (rather obvious) identity

$$\arg\max_{1\le k\le K} a_k = \arg\max_{1\le k\le K} \sum_{l=1}^{K} 1(a_k > a_l) \tag{3.1}$$

where $\{a_k\}_1^K$ are any set of distinct real valued numbers. Applying (3.1) to (1.7) one has the alternative decision rule

$$\hat{k}(\mathbf{x}) = \arg\max_{1\le k\le K} \sum_{l=1}^{K} 1[L_k \hat{f}_k(\mathbf{x}) > L_l \hat{f}_l(\mathbf{x})]. \tag{3.2}$$

Clearly (1.7) and (3.2) give identical results when the same set of estimates $\{\hat{f}_k(\mathbf{x})\}_1^K$ are used in each. Opportunities arise by generalizing (3.2).

Using the (unknown) true conditional probabilities in (3.2) one can recast the optimal Bayes decision rule as

$$k_B(\mathbf{x}) = \arg\max_{1\le k\le K} \sum_{l=1}^{K} 1[L_k f_k(\mathbf{x}) > L_l f_l(\mathbf{x})]. \tag{3.3}$$

This Bayes rule is equivalent to one based on (1.7) but can be viewed differently by expressing it as

$$k_B(\mathbf{x}) = \arg\max_{1\le k\le K} \sum_{l=1}^{K} 1\left[\frac{L_k f_k(\mathbf{x})}{f_k(\mathbf{x}) + f_l(\mathbf{x})} > \frac{L_l f_l(\mathbf{x})}{f_k(\mathbf{x}) + f_l(\mathbf{x})}\right]. \tag{3.4}$$

Each term in this sum (3.4) represents a Bayes optimal two-class decision rule for discriminating only between classes $y = c_k$ and $y = c_l$. The sum counts how many times $y = c_k$ was selected as the predicted value ("winner") at $\mathbf{x}$ in its series of (two-class) decisions with all other $K - 1$ classes. The maximization selects the class with the most winning two-class decisions ("votes") as the overall prediction at $\mathbf{x}$.

From (3.4) it is seen that the optimal K-class Bayes decision rule can be obtained by separately constructing an optimal rule for discriminating between every pair of classes $y \in \{c_k, c_l\}$, where each such two-class rule ignores the existence of the other $K - 2$ classes. The complete decision boundary separating all of the classes from each other is then produced automatically from these individual two-class decisions through the voting rule. This motivates a generalized procedure in which each of individual the two-class decision boundaries is estimated as accurately as possible without imposing

the constraint (3.2) that a common set of K conditional probability estimates $\{\hat{f}_k(\mathbf{x})\}_1^K$ be used for all $\binom{K}{2}$ decisions

$$\hat{k}(\mathbf{x}) = \arg \max_{1 \leq k \leq K} \sum_{l=1}^{K} 1[L_k \hat{f}_k^{(kl)}(\mathbf{x}) > L_l \hat{f}_l^{(kl)}(\mathbf{x})]. \tag{3.5}$$

Here $\hat{f}_k^{(kl)}(\mathbf{x})$ is the conditional probability estimate for $y = c_k$ when separating its class solely from that of $y = c_l$, without considering any of the other classes $y \neq c_l$.

4. Pairwise target functions

With the standard approach to polychotomous classification the (unknown true) "target" functions to be estimated are $\{f_k(\mathbf{x})\}_1^K$ (1.2) (2.1) (2.2). Estimates for them $\{\hat{f}_k(\mathbf{x})\}_1^K$ are used in the classification decision rule (1.7). These estimates can also be used in (3.2) producing identical results. The extension (3.5) of (3.2) allows one to consider instead an expanded set of $\binom{K}{2}$ target functions

$$\begin{aligned} f_k^{(kl)}(\mathbf{x}) &= \frac{f_k(\mathbf{x})}{f_k(\mathbf{x}) + f_l(\mathbf{x})} = E[d_k \,|\, \mathbf{x} \,\&\, (d_k = 1 \text{ or } d_l = 1)] \\ &= \frac{\Pr(\mathbf{x} \,|\, y = c_k) \Pr(y = c_k)}{\Pr(\mathbf{x} \,|\, y = c_k) \Pr(y = c_k) + \Pr(\mathbf{x} \,|\, y = c_l) \Pr(y = c_l)}. \end{aligned} \tag{4.1}$$

Note that since $f_k^{(kl)}(\mathbf{x}) + f_l^{(kl)}(\mathbf{x}) = 1$ only one of the two need be considered. Also note that using (4.1) in (3.5) produces the Bayes optimal rule.

Each of the targets (4.1) can be separately estimated by applying the regression approach (Section 2) to the reduced training set

$$T_{kl} = \{\mathbf{x}_i, y_i \,|\, y_i = c_k \text{ or } y_i = c_l\}_{i=1}^N. \tag{4.2}$$

The density estimation approach could also be applied in this manner. If the probability density functions $\Pr(\mathbf{x} \,|\, y = c_k)$ are each estimated totally separately from the others then the result will be the same as that of standard polychotomous classification. However, it is often the case that the various smoothing (meta) parameters associated with the density estimation method are chosen so as to minimize an estimate of the prediction error of the resulting classification procedure. If instead they are chosen separately to minimize an error estimate of each two-class rule based on (4.2) then different results will be produced.

A potential disadvantage of this alternative procedure (3.5) (4.1) is that more target functions must be estimated [$\binom{K}{2}$ versus K] each with less training data [(4.2) versus (1.1)]. As compensation for this each of the targets (4.1) is likely to be a (much) simpler function of the input variables with respect to the estimation procedure being used. This is especially likely when each class is well separated from most of the others resulting in low Bayes error rate and potentially accurate classification. This is illustrated by the trivial example depicted in Fig. 4.1. Here there are two input variables and three classes. Their respective probability density functions have support in the input space within the respective circle and two ellipses. The (pairwise) decision boundaries between each pair of classes are simple and estimating each target (4.1) with a linear function will produce accurate classification using the alternative approach (3.5). In the context of the standard approach one essentially estimates the decision boundary between each class and the union of the $K-1$ other classes (2.1). For the situation depicted in Fig. 4.1 one of these is also linear. However, the decision boundary is more complicated between each of the left two classes and the union of their complement classes (solid line). Using a simple linear function to approximate their corresponding respective targets (2.2) would not achieve accurate classification using the standard approach (1.7). A more complicated approximator would be required.

Although this example is especially simple it illustrates the essential concepts. All function approximation methods are limited in that for each there are broad classes of ("complex") target functions with which they have difficulty. Even for universal approximators the training sample size places

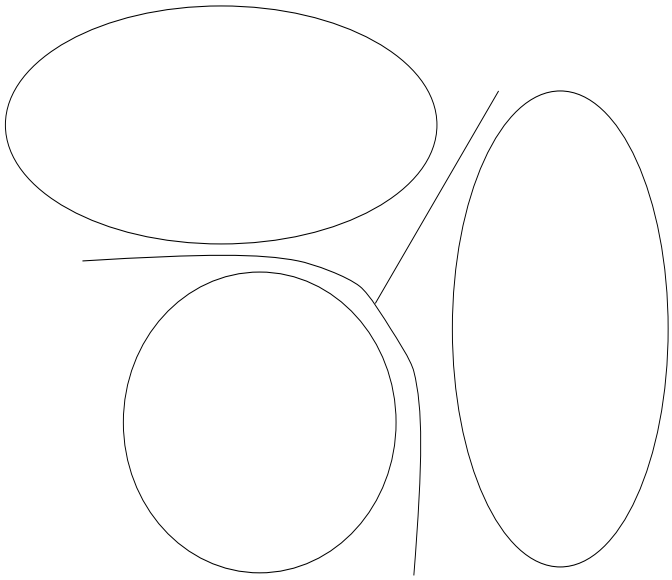

Figure 4.1: A simple three-class problem where the standard polychotomous approach leads to a nonlinear decision boundary.

such limits. In cases where the ("simpler") pairwise targets (4.1) more nearly match those that are amenable to the particular approximator being used, one might expect the alternative polychotomous classification strategy based on (3.5) (4.1) to outperform the standard one based on (1.7) and (2.1) or (2.2).

The above discussion suggests that the alternative polychotomous approach suggested here is basically a bias reduction strategy. The bias of an estimator reflects its inability to represent the target as averaged over repeated (random) training samples of the same size [Geman, Bienenstock, and Doursat (1992) and Friedman (1996)]. The hope is that the less complex pairwise targets (4.1) can be estimated with less bias than the individual ones (1.2) (2.1) (2.2). However, there is a potential increase in variance since smaller samples (4.2) are used to estimate each one. This effect is mitigated by the fact that most function estimation methods have smoothing (meta) parameters that are adjusted to trade increased bias for reduced variance through model selection [e.g. cross-validation - Stone (1974)]. If the pairwise targets are simple enough so that the inherent bias is small then the opportunity (at least potentially) exists for large variance reduction through such model selection, which can be done separately for each pairwise estimate. The situation is further complicated by the fact that the bias and variance of the target function estimates affect the resulting misclassification risk in a complex manner [Friedman (1996)] so the ultimate outcome is seldom clear. The relative merits of the different approaches to polychotomous classification (like everything else) will depend on the specifics of particular problems such as the true unknown target functions, training sample size, and approximation method being used. This must be gauged separately for each particular problem through some model selection technique such as cross-validation. In the following section situations are presented where the alternative approach produces superior results.

5. Illustrations

In this section both the standard approach and the alternative suggested here are applied with three classification methods over a large number of randomly generated targets. The methods are nearest-neighbors, decision tree induction methods (CART) using axis oriented splits (only), and CART

allowing linear combination splits [Breiman, et. al. (1984)]. All three are examples of the regression paradigm discussed in Section 2.

With the standard nearest-neighbor approach the conditional density estimates are given by

$$\hat{f}_k(\mathbf{x}) = \frac{1}{J}\sum_{i=1}^{N} 1[\,||\mathbf{x}-\mathbf{x}_i|| \leq d_J(\mathbf{x})]\,1(y_i = c_k) + t_k \tag{5.1}$$

where $d_J(\mathbf{x})$ is the Jth order statistic of $\{||\mathbf{x}-\mathbf{x}_i||\}_1^N$. The summation in (5.1) evaluates the fraction of class $y = c_k$ training observations among the J closest to $\mathbf{x}$. The second term t_k is a global "bias" [Friedman (1996)] or "threshold" [Rosen, Burke, and Goodman (1995)] adjustment used to compensate for bias. Meta parameters of the procedure are J, $\{t_k\}_1^K$, and the $(n \times n)$ matrix $\mathbf{M}$ used to define the metric distance

$$||\mathbf{x}-\mathbf{x}_i||^2 = (\mathbf{x}-\mathbf{x}_i)^t\mathbf{M}^{-1}(\mathbf{x}-\mathbf{x}_i). \tag{5.2}$$

Each of these are either prespecified or are (jointly) determined through model selection (cross-validation). These estimates (5.1) are inserted into (1.7) to form the classification decision rule.

With the alternative approach each pairwise estimate is given by

$$\hat{f}_k^{(kl)}(\mathbf{x}) = \frac{1}{J_{kl}}\sum_{y_i \in \{c_k, c_l\}} 1[\,||\mathbf{x}-\mathbf{x}_i||_{kl} \leq d_{kl}(\mathbf{x})]\,1(y_i = c_k) + t_{kl} \tag{5.3}$$

where $d_{kl}(\mathbf{x})$ is the J_{kl}th order statistic of $\{||\mathbf{x}-\mathbf{x}_i||_{kl} \,|\, y_i \in \{c_k, c_l\}\}_{i=1}^N$. Note that with this strategy a separate set of meta parameters [J_{kl}, t_{kl}, and metric $\mathbf{M}_{kl}$ (5.2)] are used for each pairwise estimate $\hat{f}_k^{(kl)}(\mathbf{x})$. These estimates are inserted into (3.5) to form the decision rule.

With the standard approach (5.1) there are K bias adjustment parameters $\{t_k\}_1^K$. Good joint values for them are highly problem dependent [Friedman (1996)] and are difficult to judge in advance. Therefore their values must be *jointly* optimized (along with the other meta parameters) through model selection. An exhaustive search through a discretized set of potential joint values requires computation that grows exponentially with the number of classes K. Heuristic search techniques may reduce this somewhat but it is still likely to be prohibitive. For this reason when $K > 2$ one usually sets $\{t_k = 0\}_1^K$. With the alternative strategy (3.5) (5.3) there are $\binom{K}{2}$ bias adjustment parameters, but each one is *separately* optimized for its particular two-class problem. Thus, with the alternative approach a K-parameter joint optimization is replaced by $\binom{K}{2}$ single parameter optimizations. Computation for the latter grows (at most) linearly with the number of classes (see Section 6) so the alternative approach provides a computationally feasible way to extend the bias adjustment strategy to polychotomous classification.

For the examples presented here the bias adjustments $\{t_k\}_1^K$ were set to zero with the standard approach (5.1) and the metric (5.2) was taken to be

$$\mathbf{M} = diag\{q_1^2, \cdots, q_n^2\} \tag{5.4}$$

where q_j is the interquartile range of the jth input variable over the training data (1.1). The only meta parameter selected by cross-validation was the number of nearest-neighbors J. For the alternative approach the metric was taken to be similar to (5.4) but with the interquartile ranges taken over the reduced training sets (4.2). The respective number of nearest-neighbors J_{kl} and bias adjustments t_{kl} (5.3) were jointly optimized (by cross-validation) separately for each respective two-class ($y \in \{c_k, c_l\}$) problem.

With decision tree induction (CART) there is one (basic) meta parameter ("cost complexity") that controls the size (number of terminal nodes) of the resulting decision tree. With the standard approach a single tree is constructed for polychotomous classification with a single associated cost complexity parameter. The conditional probability estimates (1.2) (2.2) are the fraction of class $y = c_k$ training observations among those in the terminal node containing the prediction point $\mathbf{x}$. These are inserted into (1.7) to produce a decision rule. With the alternative approach $\binom{K}{2}$ decision trees are constructed each on the reduced training sets (4.2) using a separately estimated cost complexity parameter for

each one. The conditional probability estimates $\{\hat{f}_k^{(kl)}(\mathbf{x})\}$ are taken to be the fraction of class $y = c_k$, of the two classes $(y \in \{c_k, c_l\})$, in each of the respective $\binom{K}{2}$ terminal nodes containing $\mathbf{x}$. These are inserted into (3.5) to form the decision rule.

5.1. Random Gaussian classes

As noted above the relative merits of different approaches to polychotomous classification will likely depend on the specifics of each particular problem, most notably the true target functions (1.2) and the approximation method being used. To investigate this the three methods (J - nearest neighbors "J-NN", CART with axis oriented splits only "CART-AX", and CART with linear combination splits allowed "CART-LC") were applied to a series of 50 randomly generated problems. Each involved $n = 20$ input variables and $K = 10$ classes. Equal misclassification losses $\{L_l = 1\}_1^{10}$ (1.5) were used so the figure of (lack of) merit is error rate (1.8). The classes were given equal prior probabilities $\{\Pr(y = c_k) = 0.1\}_1^{10}$. The probability density distribution for each class was taken to be a Gaussian

$$\Pr(\mathbf{x} \mid y = c_k) = \frac{1}{(2\pi)^{n/2}|\mathbf{V}_k|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \mathbf{x}_k)^t \mathbf{V}_k^{-1}(\mathbf{x} - \mathbf{x}_k)\right]. \tag{5.5}$$

The locations $\{\mathbf{x}_k\}_1^{10}$ were randomly generated from a uniform distribution $\mathbf{x}_k \sim U^{20}[0, 1]$ in the 20 dimensional input space. The respective covariance matrices $\{\mathbf{V}_k\}_1^{10}$ were random as well. The eigenvectors for each were randomly generated from a uniform distribution on the unit 20-sphere subject to orthogonality constraints. The square-roots of the eigenvalues were each randomly generated from a uniform distribution $U[0.01, 1.01]$. The optimal decision boundaries separating the classes are (random) piecewise quadratic functions in 20 variables most of which are not well suited to the approximation methods being considered here. The optimal Bayes error rates are all less that 1% whereas those for the methods considered here are seen (below) to be substantially larger.

For each of the 50 problems a different set of $K = 10$ random Gaussians were generated to serve as probability density functions (5.5). Thus, different problems of varying degrees of difficulty for each of the three methods were realized. For each problem 100 observations for each class were randomly sampled from each respective Gaussian distribution to produce a training sample of $N = 1000$. An (additional) independent sample of 200 observations per class served as an evaluation data set of size 2000 to estimate average error rates (1.8).

Figure 5.1 summarizes with boxplots the distribution of the 50 error rates for each of the six approaches. They are (from left to right) J-NN using the alternative and standard approaches, CART-LC using each respective approach, and CART-AX with both approaches. The dark area of each boxplot shows the interquartile range of the distribution with the enclosed white bar being the median. The outer hinges represent the points closest to (plus/minus) 1.5 interquartile range units from the (upper/lower) quartiles. The isolated narrow (dark) bars represent individual points outside this range (outliers). One sees that for J-NN and CART-LC the alternative strategy tends to produce error rate distributions with smaller values than with the standard one. For CART-AX the two distributions look quite similar. Also, over these 50 problems, J-NN dominates the other two methods.

A more direct comparison between the two polychotomous strategies is provided by Figure 5.2. Here the distribution of the 50 values of the scaled error rate differences

$$D_m = \frac{[e_m(std) - e_m(alt)]}{e_m(alt)} \tag{5.6}$$

over each of the 50 problems is shown, where $e_m(std)$ is the error rate for method m (J-NN, CART-LC, or CART-AX) using the standard strategy and $e_m(alt)$ is the corresponding error for the alternative strategy.

One sees that the fractional increase in error rate (5.6) by using the standard strategy over that of the alternative one varies greatly over these 50 problems for the first two methods. For J-NN the range of values is from 0.0 to 0.72 with median value 0.31 and for CART-LC 0.13 to 0.71 with median 0.40. For CART-AX the situation is quite different. The range of values of (5.6) goes from -0.09 to

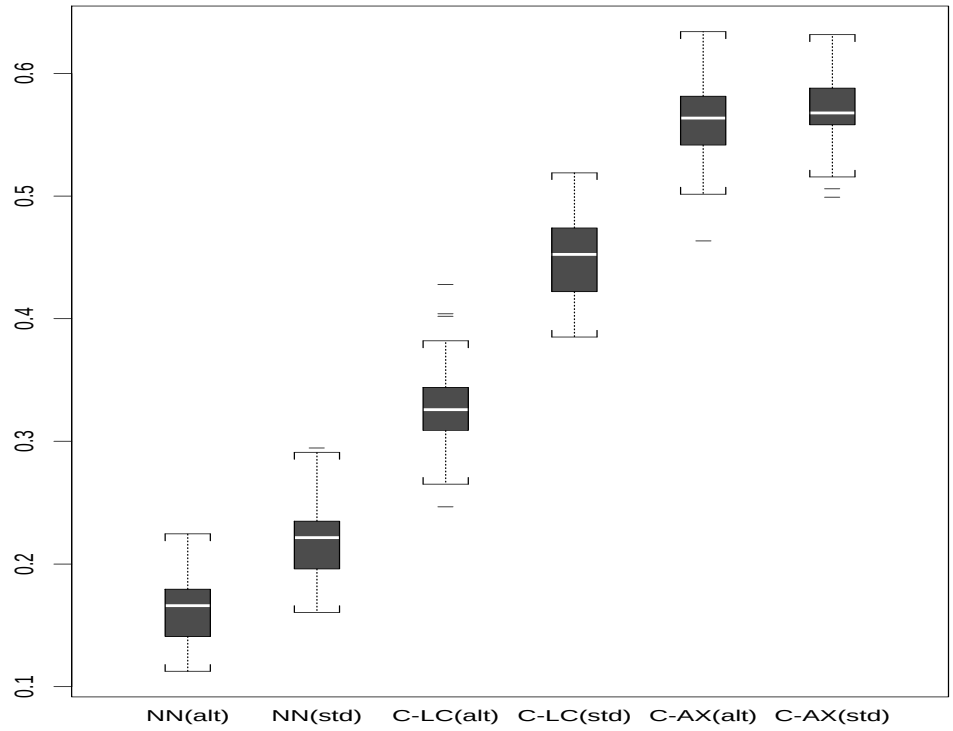

Figure 5.1: Distribution of error rates for the six approaches over the 50 randomly generated ten-class problems.

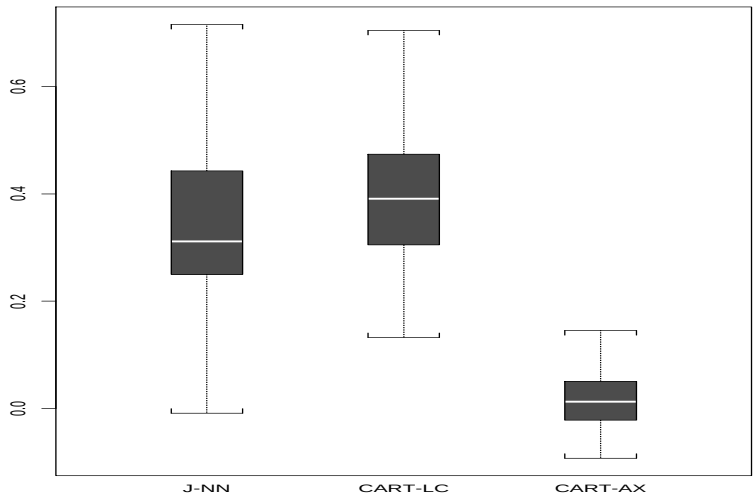

Figure 5.2: Distribution of (fractional) increase in error rate using the standard rather that the alternative strategy for the ten-class problems.

0.15 with median value of 0.01. Therefore, both strategies are giving very similar error rates for each of the 50 problems with CART-AX. Sometimes one is better sometimes the other, but they are always fairly close.

Figure 5.3 shows the corresponding plot for a different set of 50 randomly generated problems. These problems were generated according to the same prescription as the ones above except that there were $K = 5$ rather than $K = 10$ classes. The results are seen to be qualitatively similar but there are differences in detail. On average there is less advantage associated with the alternative strategy for J-NN and CART-LC with the smaller number of classes, and a small increased advantage for CART-AX. However as before this varies considerably between individual problems. For $J-$NN the range of values was -0.10 to 2.0 (not show in Fig. 5.3) with a median value of 0.26. There were two problems out of the 50 with negative values indicating that they had (slightly) higher accuracy with the standard approach. For CART-LC the range was -0.05 to 0.55 with median 0.22 and one negative value. For CART-AX the range was -0.12 to 0.23 with median 0.08 and 12 out of the 50 with negative values where the standard approach was (again slightly) superior.

Figure 5.4 shows a corresponding plot for the $K = 5$ class case but this time with twice as much training data (200 observations per class). The distributions of relative improvements (5.6) are very similar to that of the smaller sample five-class problems for CART-LC and CART-AX with medians of 0.22 and 0.07 respectively. However for J-NN the additional data has resulted in greater relative improvement (on average). The range of values for J-NN here was 0.03 to 1.5 with a median of 0.52.

5.2. Discussion

The most important lesson to be gleaned from the above exercise is that the relative performance of different approaches can strongly depend on the particular problem to which they are applied. Like all other aspects of learning methodology no (reasonable) approach dominates any other over all (reasonable) situations. Even within the restricted class of target functions generated through (5.5) there was a wide range of relative performance increase values (5.6) between the two polychotomous approaches for each classification method. In addition there were large differences among the methods

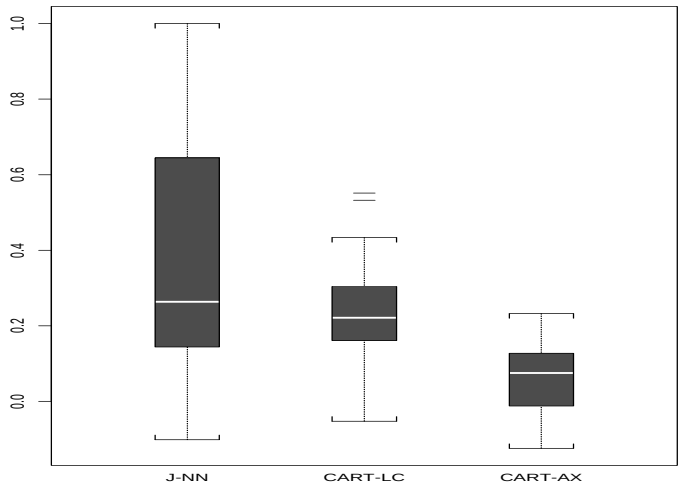

Figure 5.3: Distribution of (fractional) increase in error rate using the standard rather than the alternative strategy for the five-class problems with $N = 500$.

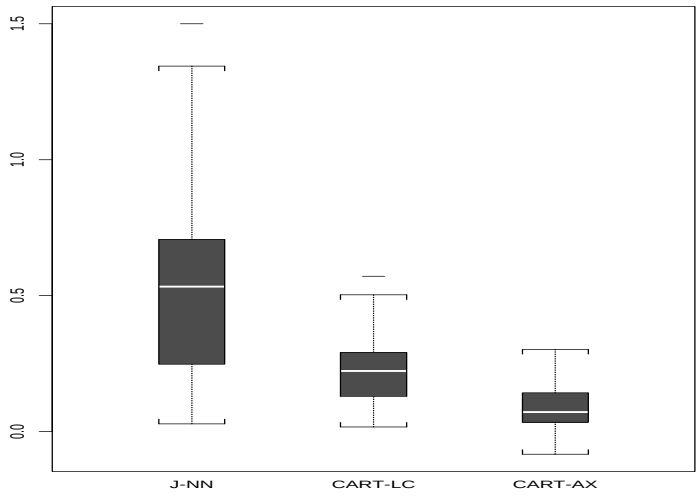

Figure 5.4: Distribution of (fractional) increase in error rate using the standard approach rather than the alternative strategy for the five-class problems with $N = 1000$.

($J-$NN, CART-LC, and CART-AX) themselves. Generally it cannot be known in advance which approach will be the most successful with any given problem at hand. Model selection must be used to estimate this separately for each application.

Bearing this in mind, one can still speculate on the reasons underlying the results obtained here. For J-NN the general superiority of the alternative polychotomous approach is probably due the its ability to separately tune the procedural (meta) parameters for each two-class decision, particularly the bias adjustment parameters t_{kl} (5.3). This is consistent with its increased relative performance for the larger training samples (Fig. 5.4). With more data there is less variance so that reducing bias (through bias adjustment) becomes more profitable. Experiments in Friedman (1996) also indicated increased performance enhancement for ($J = 2$) NN with bias adjustment, as the training sample size is increased. For CART-LC the general increase in performance using the alternative polychotomous strategy may be due to particularities of the algorithm it uses for finding linear combinations. Namely, this algorithm may be more effective for $K = 2$ classes than when used in the context of many classes. Also, linear combination estimates tend to be fairly stable against perturbations of the training data and so (generally) have less variance than individual coordinate axis estimates. In the presence of less variance the bias reduction through the alternative strategy may be more effective.

The reason for the quite small (average) improvement obtained for CART-AX may be due to the high variance inherent in the procedure [Breiman (1995)]. As noted above, the alternative polychotomous strategy attempts to make gains largely through bias reduction, and so will tend to be less effective when applied in the context of high variance methods. Effective techniques for reducing the variance of CART-AX have been proposed ["bagging" - Breiman (1995) and "bumping" - Tibshirani and Knight (1995)]. The alternative polychotomous strategy can be applied to these lower variance axis-oriented tree-based procedures as well. The result may be greater relative performance advantage over the corresponding standard approach due to their reduced variance.

6. Computation

The relative computational requirement of the standard and alternative polychotomous approaches depends on how computation for the former grows with increasing values of the various problem dependent factors. Clearly both have the same dependence on the number of input variables n. The worst case in terms of increased computation for the alternative strategy occurs when the computation for the standard one is independent of the number of classes K and grows linearly with the training sample size N. With the alternative there are $K(K-1)/2$ two-class classification problems each using (on average) a fraction $2/K$ of the data. Thus computation in this case is increased by a factor of $K-1$. If computation for the standard approach grows with increasing K then the trade-off becomes less severe (or perhaps more favorable) to the alternative approach. For example with bias adjustment, computation for the standard approach grows exponentially with K and the alternative is much faster. If computation for the standard strategy grows more rapidly than linearly with N, then there will always be (larger) training sample sizes for which the alternative is faster. Also, the alternative clearly lends itself to parallel implementation even if the standard one upon which it is based does not.

7. Previous work

There have been two previous proposals for polychotomous classification strategies different from the standard one. These are "flexible discriminant analysis" (FDA) [Hastie, Tibshirani, and Buja(1994)] and the "error-correcting output codes" (ECOC) technique of Dietterich and Bakiri (1995). With FDA estimates of the K conditional probabilities (1.2) are obtained through the regression approach (2.2) (2.4). Instead of inserting these directly into a decision rule such as (1.7) they are instead used as input variables for a (kind of) K-class linear discriminant analysis (LDA). LDA is (asymptotically) optimal when the probability density functions for each class are normal with a common covariance matrix, here in the K-dimensional space of the joint values of the conditional probability estimates. The covariance matrix estimates for each class are pooled to form a common estimate used for all

of the classes. Exact normality is usually not essential to good LDA performance but departures from elliptical symmetry can be detrimental. Also very different covariance matrices among the class distributions can degrade performance.

LDA (and thereby FDA) can be generalized by substituting the alternative strategy (3.5) in place of classical LDA. Here a different linear discriminant is estimated for separating each pair of classes. (For FDA this is done in the conditional probability space.) These $\binom{K}{2}$ decision boundaries are then used with the voting rule (3.5) to form the final decision. This strategy eases the assumption of a common covariance matrix for all of the classes since a different one can be used to discriminate between each class pair. However, the real potential of this approach may be realized if a bias adjustment [analog of t_{kl} (5.3) for J-NN] is incorporated and separately optimized for each pairwise decision boundary estimate. Such adjustments can compensate for the bias associated with LDA when the individual class probabilities depart from elliptical symmetry. When employing bias adjustments in this manner one can (but need not) pool the covariance matrix estimates. Whether this alternative to classical LDA improves performance will likely depend (as always) upon the specifics of particular problems.

The spirit of the error-correcting output codes (ECOC) approach [Dietterich and Bakiri(1995)] is more nearly the same as that of the alternative polychotomous strategy presented here. Namely, the overall K-class decision is constructed from the results of a series of two-class problems. The techniques differ in the formulation of the two-class problems and how their results are combined. With ECOC the first ("super") class $C_{A(m)}$ of each pair is the union of a subset $A(m) \subset \{1, \cdots, K\}$ of the original K-classes. That is, $C_{A(m)} = \{c_k \,|\, k \in A(m)\}$, and the other contrasting (super) class is the complement subset $C_{\bar{A}(m)} = \{c_k \,|\, k \notin A(m)\}$. The regression approach (Section 2) is used to estimate the target conditional probabilities

$$f_{A(m)}(\mathbf{x}) = \Pr[y \in C_{A(m)} \,|\, \mathbf{x}] = \sum_{k \in A(m)} f_k(\mathbf{x}) \tag{7.1}$$

where $\{f_k(\mathbf{x})\}_1^K$ are the original class conditional probabilities (1.2) (2.1). A potentially large number $1 \leq m \leq M$ of such two-class problems are defined and the estimates $\{\hat{f}_{A(m)}(\mathbf{x})\}_1^M$ of the targets (7.1) are then used as input variables to a (post) classification procedure. Note that here all of the original classes participate in each (two-class) problem and the entire training set (1.1) is used to obtain each $\hat{f}_{A(m)}(\mathbf{x})$.

A particular procedure based on this paradigm is defined by the strategy used to assign the super-classes $\{A(m)\}_1^M$, the method used to estimate the conditional probabilities (7.1), and the post classification procedure employed. Dietterich and Bakiri (1995) used the theory of error-correcting codes to assign super-classes in a way that evenly covers the M-dimensional (super-class) space. Decision tree induction (C4.5) and neural networks were the estimation methods used for comparison. The post process was a nearest prototype classifier where the prototype vectors (in the M-dimensional space) were derived from the error-correcting codes used to assign the super-classes.

The underlying connections between ECOC and the alternative polychotomous strategy presented here do not seem obvious. The central theme of ECOC is "class aggregation". The super-class target functions (7.1) are likely to be more complex than the individual ones (1.2) defining them. This can potentially result in increased bias in their estimation. However this is mitigated by the use of all of the training data (1.1) to estimate each one, and by the averaging effect of the final (post) decision rule. By contrast, the central theme of the alternative presented here is "class separation". The expectation is that the individual two-class targets (4.1) will be less complex than the original targets (1.2) so that they can be estimated with less bias. However increased variance may result due to the reduced training sample (4.2) used to estimate them. This will depend upon the success of model selection in reducing the variance of each two-class rule.

Evidence so far suggests that ECOC is more successful in improving unstable high variance procedures such as axis-oriented decision trees and neural networks, and does not help the more stable ones like J-NN and radial basis functions [Kong and Dietterich (1995)]. Conversely, over the 150 problems of Section 5, it was seen that the alternative suggested here had small impact on axis-oriented decision trees, but was able to achieve considerable improvement with J-NN. Therefore, the relative merits of

these two approaches (like all others) will likely depend upon the estimation method with which they are being used. The best estimation method in turn depends upon the properties of the particular problem encountered. For example, over the problems considered in Section 5 the performance of J-NN dominated that of CART-LC and especially CART-AX, as seen in Fig. 5.1. [The corresponding plots for the two five-class problems (not shown) were nearly identical.] However, there are surely many problems for which the converse is true. An understanding of these issues may emerge from future research. But for now, model selection techniques (such as cross-validation) seem to be the best (if imperfect) way to make such choices.

8. Concluding remarks

The central idea underlying the alternative polychotomous classification strategy presented here is casting the K-class problem into a series of $\binom{K}{2}$ (smaller) two-class problems. In each a different decision boundary is estimated to separate its two classes. The voting rule (3.5) is applied to form the final (K-class) decision. The goal is to achieve maximum accuracy with each two-class decision. So far, the (tacit) assumption has been that the same classification procedure is used to estimate each two-class rule. This clearly is not a requirement. As with all classification problems the best procedure depends on the specifics of the problem itself (target function, training sample size, etc.). Thus each individual two-class problem may best be served by using a different method. Some of them may involve boundaries that are easily linearly separable so LDA may be most appropriate for them. Others may involve complex boundaries that are more accurately estimated by more flexible techniques such as J-NN, decision trees, or large neural networks. The point is that with the alternative polychotomous strategy each two-class decision rule can be treated as a completely separate classification problem. Model selection techniques can be used to estimate the best method individually for each one.

As repeatedly emphasized, the relative merits of different classification approaches depend on the problem at hand, and are not generally known in advance. In such cases model selection can be used to obtain performance estimates. As shown in Section 5 this is also the case for the standard and alternate polychotomous strategies. To the extent that the results in Section 5 can serve as a guide however, it would appear the alternative strategy is the more conservative choice between the two. Over the 150 problems considered there, the alternative seldom produced a substantial decrease in performance relative to the standard one. The worst case was a *relative* error rate (5.6) increase of 12% whereas the best case was a 200% decrease (1/3 the standard error rate). Typical values (medians) ranged around 30% relative improvement. However, as with all such empirical studies, caution must be exercised in extrapolating such results beyond those situations actually considered.

Important discussions with Trevor Hastie on the subject of this work are gratefully acknowledged.

References

[1] Bishop, C. M. (1995). *Neural Networks for Pattern Recognition.* Clarendon Press · Oxford.

[2] Breiman, L. (1995). Bagging predictors. Dept. of Statistics, University of California, Berkeley, Technical Report.

[3] Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. (1984). *Classification and Regression Trees.* Wadsworth.

[4] Chow, W. S. and Chen, Y. C. (1992). A new fast algorithm for effective training of neural classifiers. *Pattern Recognition* **25**, 423-429.

[5] Dietterich, T. G. and Bakiri, G. (1995). Solving multiclass learning problems via error-correcting output codes. *J. Artificial Intelligence Research* **2**, 263-286.

[6] Fix, E. and Hodges, J. L. (1951). Discriminatory analysis - nonparametric discrimination: consistency properties. Randolph Field Texas: U. S. Airforce School of Aviation Medicine Technical *Report No. 4.*

[7] Friedman, J. H. (1985). Classification and multiple response regression through projection pursuit. Technical Report, Dept. of Statistics, Stanford University LCS012.

[8] Friedman, J. H. (1996). On bias, variance, 0/1 - loss, and the curse-of-dimensionality. Technical Report, Dept. of Statistics, Stanford University.

[9] Geman, S., Bienenstock, E., and Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Comp.* **4**, 1-48.

[10] Hastie, T., Tibshirani, R., and Buja, A. (1994). Flexible discriminant analysis by optimal scoring. *J. Amer. Statist. Assoc.* **89**, 1255-1270.

[11] Heckerman, D., Geiger, D., and Chickering, D. (1994). Learning Bayesian networks: the combination of knowledge and statistical data. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, 293 - 301. AAAI Press and M. I. T. Press.

[12] Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE* **78**, 1464-1480.

[13] Kong, E. B. and Dietterich, T. G. (1995). Error-correcting output coding corrects bias and variance. Technical report, Dept. of Computer Science, Oregon State University.

[14] Lippmann, R. (1989). Pattern classification using neural networks. *IEEE Communications Magazine* **11**, 47-64.

[15] McLachlan, G. J. (1992). *Discriminant Analysis and Statistical Pattern Recognition.* Wiley.

[16] Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning.* Morgan Kaufmann.

[17] Ripley, B. D. (1996). *Pattern Recognition and Neural networks.* Cambridge University Press.

[18] Rosen, D. B., Burke, H. B., and Goodman, P. H. (1995). Local learning methods in high dimension: beating the bias-variance dilemma via recalibration. Workshop *Machines That Learn - Neural Networks for Computing*, Snowbird Utah.

[19] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition,* Vol. 1, Foundations, pp 318. MIT Press.

[20] Stone, M. (1974). Cross-validatory choice and assessment of statistical predictors (with discussion). *J. Roy. Statist. Soc. Ser. B* **36**, 111.

[21] Tibshirani, R. J. and Knight, K. (1995). Model search and inference by bootstrap "bumping". Technical report, Dept. of Statistics, University of Toronto.