Polychotomous Regression

Charles Kooperberg, Smarajit Bose and Charles J. Stone *

Technical Report No. 288
May 10, 1995
University of Washington
Department of Statistics
Seattle, Washington 98195-4322

Abstract

An automatic procedure that uses linear splines and their tensor products is proposed to fit a regression model to data involving a polychotomous response variable and one or more predictors. The fitted model can be used for multiple classification or for solving a finite action problem. The automatic fitting procedure involves maximum likelihood estimation, stepwise addition, stepwise deletion, and model selection by AIC, cross-validation or an independent test set. The algorithm can be modified to make it applicable to large data sets. This has been illustrated using a phoneme recognition data set with 250,000 cases, 45 classes and 63 predictors.

KEY WORDS: Linear splines; Logistic regression; MARS; Maximum likelihood; Model selection; Multiple Classification, Speech recognition, Tensor products.

1 Introduction

The multiple classification problem is well studied in statistics. Typically there is a qualitative random variable Y that takes on a finite number K of values which we refer to as classes. Based on a vector of predictors $\mathbf{X} \in \mathbb{R}^M$, we want to predict Y. Many methods have been proposed for this problem. See Mardia, Kent and Bibby (1979) for a discussion of 'classical' discriminant analysis methods. One of the popular modern multiple classification techniques is CART (Breiman, Friedman, Olshen and Stone, 1984), which approaches the multiple classification problem using recursive

*Charles Kooperberg is Assistant Professor, Department of Statistics, University of Washington, Seattle, WA 98195-4322. Smarajit Bose is Visiting Assistant Professor, Department of Statistics, The Ohio State University, Columbus, OH 43210. Charles J. Stone is professor, Department of Statistics, University of California, Berkeley, CA 94720. Charles Kooperberg was supported in part by National Science Foundation grant DMS-9403371 and National Institutes of Health grant CA61937-01A0. The computations for Section 11 were sponsored in part by the Phillips Laboratory, Air Force Materiel Command, USAF, under cooperative agreement number F29601-93-2-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Phillips Laboratory or the U.S. Government. We wish to thank Andreas Buja for some stimulating conversations. We also wish to thank Su-Lin Wu, Nikki Mirghafo and Nelson Morgan at the International Computer Science Institute (ICSI) in Berkeley for arranging for us to get the phoneme recognition data analyzed in this paper, for their help in understanding this data and its connection with continuous speech recognition, and for their encouragement of our foray into this field.

partitioning techniques that have strong links to nonparametric regression. Hastie, Tibshirani and Buja (1994) introduce flexible discriminant analysis, which combines nonparametric regression techniques with discriminant analysis. Bose (1994) proposes classification using splines which employs least squares regression and additive cubic splines. In computer science and engineering, neural networks seem to be the method of choice (see Ripley, 1994 for an overview).

As is well known, the optimal classification rule predicts Y to be $\arg\max_k P(Y = k|\mathbf{X})$. Most of the popular classification methods try to find $\arg\max_k P(Y = k|\mathbf{X})$ without precise estimation of the conditional class probabilities. However, there are many problems in which direct classification does not suffice. For example, in Section 11 we discuss the approach by Bourlard and Morgan (1994) to the phoneme recognition problem, which requires accurate estimation of the probability of a phoneme being in any particular class. Clearly, pure multiple classification methods are not longer useful in such applications.

On the other hand, multiple logistic regression (polychotomous regression) techniques have been used for a long time (see Hosmer and Lemeshow, 1989). In a (linear) polychotomous regression model we do obtain an estimate of all the conditional class probabilities. (Bose (1992) attempts to estimate conditional class probabilities using a logistic model with additive cubic splines.) In this paper we combine nonparametric regression techniques similar to those used in Friedman (1991) and Kooperberg, Stone and Truong (1995) with polychotomous regression to obtain a classification methodology, POLYCLASS, that provides reliable estimates for conditional class probabilities. Using these estimates in turn, we can also solve the generalization of the multiple classification problem known as the finite action problem; see Section 2.

This paper is organized as follows. In Sections 2, 3, 4 and 5 we set up the polychotomous regression model and describe its relation to multiple classification and the finite action problem, and we discuss the estimation procedure. In Sections 6 and 7 we discuss the model selection procedure, which employs piecewise linear splines and selected tensor products as well as stepwise addition and stepwise deletion of basis functions. In Section 8 we discuss some extensions of the polychotomous regression model, in Section 9 we discuss a least squares approximation to the model selection procedure that can dramatically speed up the computations, and in Section 10 we apply POLYCLASS to some small examples involving simulated data. Our major example, which comes from the area of speech recognition, is discussed in Section 11. Here we analyze a data set of more than 2000 utterances (short sentences) that yielded almost 250,000 cases, which are 12.5 ms time periods. The classes are the 45 possible phonemes that may be spoken at any moment. The main goal in this example is to estimate the conditional probabilities of each phoneme (not to classify the phoneme at that moment) based on 63 predictors, which are obtained from the audible spectrum of the sound. Finally, in Section 12 we give a concluding discussion and in Section 13 we give some numerical details.

2 Polychotomous Regression

Consider a qualitative random variable Y that takes on a finite number K of values. We can think of Y as ranging over $\mathcal{K} = \{1, \ldots, K\}$. Suppose the distribution of Y depends on predictors $x_1, \ldots, x_M$, where $\mathbf{x} = (x_1, \ldots, x_M)$ ranges over the subset $\mathcal{X}$ of $\mathbb{R}^M$. Now let $\mathbf{x}$ be distributed as a random vector; that is, consider a random pair $(\mathbf{X}, Y)$, where $\mathbf{X}$ is an $\mathcal{X}$-valued random vector and Y is a $\mathcal{K}$-valued random variable. Suppose $P(Y=k|\mathbf{X}=\mathbf{x})>0$ for $\mathbf{x}\in\mathcal{X}$ and $k\in\mathcal{K}$, and set

$$\theta(k|\mathbf{x}) = \log \frac{P(Y=k|\mathbf{X}=\mathbf{x})}{P(Y=K|\mathbf{X}=\mathbf{x})}, \qquad \mathbf{x}\in\mathcal{X} \text{ and } k\in\mathcal{K}.$$

Then $\theta(K|\mathbf{x})=0$ for $\mathbf{x}\in\mathcal{X}$ and

$$P(Y=k|\mathbf{X}=\mathbf{x}) = \frac{\exp\theta(k|\mathbf{x})}{\exp\theta(1|\mathbf{x})+\cdots+\exp\theta(K|\mathbf{x})}, \qquad \mathbf{x}\in\mathcal{X} \text{ and } k\in\mathcal{K}. \tag{1}$$

We refer to (1) as the *polychotomous regression model*; when $K=2$ it is referred to as the *logistic regression model.*

The usual parametric approach to the polychotomous regression problem is to use the model

$$\theta(k|\mathbf{x}) = \beta_{k0}+\beta_{k1}x_1+\cdots+\beta_{kM}x_M, \qquad 1\le k<K.$$

In practice it may be more desirable to model the predictor effects by using smooth, nonlinear functions. A *generalized additive model* (Hastie and Tibshirani, 1990) for the polychotomous regression problem is given by

$$\theta(k|\mathbf{x}) = \theta_{1k}(x_1)+\theta_{2k}(x_2)+\cdots+\theta_{Mk}(x_M), \qquad 1\le k<K.$$

In order to allow for interactions between predictors, the generalized additive model can be further refined. To motivate our approach, suppose $\mathbf{x}=(x_1,x_2,x_3)$ and write

$$\theta(k|\mathbf{x}) = \theta_{1k}(x_1)+\theta_{2k}(x_2)+\theta_{3k}(x_3)+\theta_{12k}(x_1,x_2)+\theta_{13k}(x_1,x_3)+\theta_{23k}(x_2,x_3), \qquad 1\le k<K,$$

where all $\theta_{1k}(\cdot),\ldots,\theta_{23k}(\cdot)$ are smooth functions. Here $\theta_{1k}(\cdot)$, $\theta_{2k}(\cdot)$ and $\theta_{3k}(\cdot)$ are referred to as main effects, and $\theta_{12k}(\cdot)$, $\theta_{13k}(\cdot)$, and $\theta_{23k}(\cdot)$ are referred to as interactions.

Given a random sample, consider an estimate

$$\widehat{\theta}(k|\mathbf{x}) = \widehat{\theta}_{1k}(x_1)+\widehat{\theta}_{2k}(x_2)+\widehat{\theta}_{3k}(x_3)+\widehat{\theta}_{12k}(x_1,x_2)+\widehat{\theta}_{13k}(x_1,x_3)+\widehat{\theta}_{23k}(x_2,x_3), \qquad 1\le k<K, \tag{2}$$

where each component is empirically orthogonal to the corresponding lower order components. We can think of $\hat{\theta}(k|\mathbf{x})$ as an estimate of $\theta(k|\mathbf{x})$. Alternatively, we can think of it as an estimate of the corresponding best theoretical approximation

$$\theta^*(k|\mathbf{x}) = \theta^*_{1k}(x_1)+\theta^*_{2k}(x_2)+\theta^*_{3k}(x_3)+\theta^*_{12k}(x_1,x_2)+\theta^*_{13k}(x_1,x_3)+\theta^*_{23k}(x_2,x_3), \qquad 1\le k<K, \tag{3}$$

to $\theta(k|\mathbf{x})$, where best means having the maximum expected likelihood subject to the indicated form and each component is orthogonal to the corresponding lower order components with respect to a suitably defined theoretical inner product. According to Stone (1994), the right sides of (2) and (3) are referred to as the ANOVA decompositions of $\widehat{\theta}$ and θ^*, respectively. If the components of the ANOVA decomposition θ^* are estimated accurately by the corresponding ANOVA components of $\widehat{\theta}$, then examination of the components of the ANOVA decomposition of $\widehat{\theta}$ should shed light on the relationship of the conditional class probabilities through the functions θ^* and, to a lesser extent, through the functions θ.

More generally, consider the approximation θ^* to θ having the form of a specified sum of functions of at most d of the variables $x_1,\ \ldots,\ x_M$ and subject to this form, chosen to maximize the expected conditional log-likelihood. Given a random sample of size n from the distribution of

$(\mathbf{X}, Y)$, if maximum likelihood and suitable sums of (nonadaptive) tensor products of polynomial splines are used to construct estimates of θ^*, its components can achieve the L_2 rate of convergence $n^{-p/(2p+d)}$, where p is a suitably defined smoothness parameter corresponding to θ^* (Hansen, 1994); in particular, $p = 1$ when linear splines and their tensor products are used. Thus by choosing $d = 1$ as in generalized additive modeling or $d = 2$ instead of $d = M$, we can ameliorate the curse of dimensionality.

In this paper we choose $d = 2$, and we use linear splines and their tensor products, but we will be choosing these splines in an adaptive way.

3 The Finite Action Problem

Consider a finite action space $\mathcal{A} = \{1, \ldots, A\}$ of size A and let $C(a; k)$ denote the cost of taking action a when the class is k. We refer to $C(a; k)$, $1 \le k \le K$ and $1 \le a \le A$, as the *cost matrix*. In the context of multiple classification, $A = K$ and $C(a; k)$ is the cost of classifying a class k object as class a. The decision about which action to take will be based on the observed value of $\mathbf{X}$. A decision rule is an $\mathcal{A}$-valued function d on $\mathcal{X}$: given that $\mathbf{X} = \mathbf{x} \in \mathcal{X}$, we take action $a = d(\mathbf{x})$.

The corresponding risk (expected cost) for a decision rule d is given by

$$R(d) = E[C(d(\mathbf{X}); Y)] = E\Big(\sum_k C(d(\mathbf{X}); k) P(Y = k|\mathbf{X})\Big).$$

To obtain an optimal rule (that is, one having minimum risk), for each $\mathbf{x} \in \mathcal{X}$ we choose $d(\mathbf{x})$ such that

$$\sum_k C(d(\mathbf{x}); k) P(Y = k|\mathbf{X} = \mathbf{x}) = \min_{a \in \mathcal{A}} \sum_k C(a; k) P(Y = k|\mathbf{X} = \mathbf{x}). \tag{4}$$

In order to make this choice we need to determine or, at least, estimate $P(Y = k|\mathbf{X} = \mathbf{x})$ for $\mathbf{x} \in \mathcal{X}$ and $k \in \mathcal{K}$. Using (1) we can rewrite (4) as

$$\sum_k C(d(\mathbf{x}); k) \exp\theta(k|\mathbf{x}) = \min_{a \in \mathcal{A}} \sum_k C(a; k) \exp\theta(k|\mathbf{x}) \tag{5}$$

or as

$$d(\mathbf{x}) = \arg\min_{a \in \mathcal{A}} \sum_k C(a; k) \exp\theta(k|\mathbf{x}).$$

(Note that $d(\mathbf{x})$ may not be uniquely defined; ties can be broken arbitrarily.)

4 Linear Models

Let p be a positive integer, and let G be a p-dimensional linear space of functions on $\mathcal{X}$ with basis $B_1, \ldots, B_p$. Consider the model

$$\theta(k|\mathbf{x}) = \theta(k|\mathbf{x}; \boldsymbol{\beta}) = \sum_{j=1}^{p} \beta_{jk} B_j(\mathbf{x}), \qquad \mathbf{x} \in \mathcal{X} \text{ and } k \in \mathcal{K}; \tag{6}$$

here $\boldsymbol{\beta}_k = (\beta_{k1}, \ldots, \beta_{kp})^T$ for $1 \le k \le K-1$, $\boldsymbol{\beta}_K = 0$, and $\boldsymbol{\beta}$ is the $p(K-1)$-dimensional column vector consisting of the entries of $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{K-1}$, which ranges over $\mathcal{B} = \mathbb{R}^{p(K-1)}$. Correspondingly, we set

$$\begin{aligned} P(Y=k|\mathbf{X}=\mathbf{x};\boldsymbol{\beta}) &= \frac{\exp\theta(k|\mathbf{x};\boldsymbol{\beta})}{\exp\theta(1|\mathbf{x};\boldsymbol{\beta}) + \cdots + \exp\theta(K|\mathbf{x};\boldsymbol{\beta})} \\ &= \exp(\theta(k|\mathbf{x};\boldsymbol{\beta}) - c(\mathbf{x};\boldsymbol{\beta})), \qquad \boldsymbol{\beta}\in\mathcal{B},\ \mathbf{x}\in\mathcal{X} \text{ and } k\in\mathcal{K}, \end{aligned}$$

where

$$c(\mathbf{x};\boldsymbol{\beta}) = \log[\exp\theta(1|\mathbf{x};\boldsymbol{\beta}) + \cdots + \exp\theta(K|\mathbf{x};\boldsymbol{\beta})], \qquad \boldsymbol{\beta}\in\mathcal{B} \text{ and } \mathbf{x}\in\mathcal{X}.$$

Now

$$\log P(Y=k|\mathbf{X}=\mathbf{x};\boldsymbol{\beta}) = \theta(k|\mathbf{x};\boldsymbol{\beta}) - c(\mathbf{x};\boldsymbol{\beta}), \qquad \boldsymbol{\beta}\in\mathcal{B},\ \mathbf{x}\in\mathcal{X} \text{ and } k\in\mathcal{K}.$$

Thus

$$\frac{\partial}{\partial\beta_{k_1 j_1}} \log P(Y=k|\mathbf{X}=\mathbf{x};\boldsymbol{\beta}) = B_{j_1}(\mathbf{x})[\delta_{k_1 k} - \exp(\theta(k_1|\mathbf{x};\boldsymbol{\beta}) - c(\mathbf{x};\boldsymbol{\beta}))]$$

for $\boldsymbol{\beta}\in\mathcal{B}$, $\mathbf{x}\in\mathcal{X}$, $k\in\mathcal{K}$, $1\le k_1\le K-1$ and $1\le j_1\le p$, where $\delta_{k_1 k}=1$ for $k_1=k$ and $\delta_{k_1 k}=0$ for $k_1\ne k$. Consequently,

$$\begin{aligned} \frac{\partial^2}{\partial\beta_{k_1 j_1}\partial\beta_{k_2 j_2}} \log P(Y=k|\mathbf{X}=\mathbf{x};\boldsymbol{\beta}) &= B_{j_1}(\mathbf{x})B_{j_2}(\mathbf{x})\Big\{ -\delta_{k_1 k_2}\exp(\theta(k_1|\mathbf{x};\boldsymbol{\beta}) - c(\mathbf{x};\boldsymbol{\beta})) \\ &\quad + \exp(\theta(k_1|\mathbf{x};\boldsymbol{\beta}) - c(\mathbf{x};\boldsymbol{\beta}))\exp(\theta(k_2|\mathbf{x};\boldsymbol{\beta}) - c(\mathbf{x};\boldsymbol{\beta}))\Big\} \end{aligned}$$

for $\boldsymbol{\beta}\in\mathcal{B}$, $\mathbf{x}\in\mathcal{X}$, $k\in\mathcal{K}$, $1\le k_1,k_2\le K-1$ and $1\le j_1,j_2\le p$. Thus the Hessian matrix of $\log P(Y=k|\mathbf{X}=\mathbf{x};\cdot)$ is negative semi-definite on $\mathcal{B}$ for $\mathbf{x}\in\mathcal{X}$ and $k\in\mathcal{K}$.

5 Maximum Likelihood Estimation

In order to choose $d(\mathbf{x})$ approximately to satisfy (5), we need to use sample data to estimate the functions $\theta(k|\cdot)$, $1\le k\le K-1$.

Let $(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)$ be independent random pairs, with each pair having the same joint distribution as $(\mathbf{X}, Y)$. The log-likelihood function corresponding to the finite-parameter model introduced in Section 4 is given by

$$\ell(\boldsymbol{\beta}) = \sum_i [\theta(Y_i|\mathbf{X}_i;\boldsymbol{\beta}) - c(\mathbf{X}_i;\boldsymbol{\beta})], \qquad \boldsymbol{\beta}\in\mathcal{B}, \tag{7}$$

which is a concave function on $\mathcal{B}$.

For numerical reasons, we add a small penalty term to the log-likelihood function. Specifically, set

$$\ell_\epsilon(\boldsymbol{\beta}) = \ell(\boldsymbol{\beta}) - \epsilon\sum_i\sum_{k=1}^K u_{ik}^2, \tag{8}$$

where

$$u_{ik} = \theta(k|\mathbf{X}_i;\boldsymbol{\beta}) - \frac{1}{K}\sum_{k'=1}^K \theta(k'|\mathbf{X}_i;\boldsymbol{\beta}), \qquad k\in\mathcal{K}.$$

The penalized log-likelihood function, in which we have typically used $\epsilon = 10^{-6}$, is guaranteed to have a finite maximum. Without the penalty term, however, it is possible that, when the likelihood function is maximized, some β_{kj} equals $\pm\infty$. This can happen, for example, if $B_j(\mathbf{X}_i) = 0$ for all i such that $Y_i = k$.

It is straightforward to derive that

$$\begin{aligned} \frac{\partial}{\partial \beta_{k_1 j_1}} \ell_\epsilon(\boldsymbol{\beta}) &= \sum_i B_{j_1}(\mathbf{X}_i)[\delta_{k_1 Y_i} - \exp(\theta(k_1|\mathbf{X}_i;\boldsymbol{\beta}) - c(\mathbf{X}_i;\boldsymbol{\beta}))] \\ &\quad - \frac{2\epsilon}{K} \sum_i \sum_{k=1}^{K} u_{ik}(K\delta_{k_1 k} - 1) B_{j_1}(\mathbf{X}_i) \end{aligned} \tag{9}$$

for $\boldsymbol{\beta} \in \mathcal{B}$, $1 \le k_1 \le K-1$ and $1 \le j_1 \le p$, and that

$$\begin{aligned} \frac{\partial^2}{\partial \beta_{k_1 j_1} \partial \beta_{k_2 j_2}} \ell_\epsilon(\boldsymbol{\beta}) &= \sum_i B_{j_1}(\mathbf{X}_i) B_{j_2}(\mathbf{X}_i) \Big\{ - \delta_{k_1 k_2} \exp(\theta(k_1|\mathbf{X}_i;\boldsymbol{\beta}) - c(\mathbf{X}_i;\boldsymbol{\beta})) \\ &\quad + \exp(\theta(k_1|\mathbf{X}_i;\boldsymbol{\beta}) - c(\mathbf{X}_i;\boldsymbol{\beta})) \exp(\theta(k_2|\mathbf{X}_i;\boldsymbol{\beta}) - c(\mathbf{X}_i;\boldsymbol{\beta})) \Big\} \\ &\quad - \frac{2\epsilon}{K} \sum_i (K\delta_{k_1 k_2} - 1) B_{j_1}(\mathbf{X}_i) B_{j_2}(\mathbf{X}_i), \end{aligned} \tag{10}$$

for $\boldsymbol{\beta} \in \mathcal{B}$, $1 \le k_1, k_2 \le K-1$ and $1 \le j_1, j_2 \le p$.

The maximum likelihood estimate $\widehat{\boldsymbol{\beta}}$ is given $\ell_\epsilon(\widehat{\boldsymbol{\beta}}) = \max_{\boldsymbol{\beta}} \ell_\epsilon(\boldsymbol{\beta})$ and the log-likelihood of the model is given by $\hat{\ell}_\epsilon = \ell_\epsilon(\widehat{\boldsymbol{\beta}})$. The corresponding maximum likelihood estimates of $\theta(k|\mathbf{x})$, $\mathbf{x} \in \mathcal{X}$ and $k \in \mathcal{K}$, are given by $\hat{\theta}(k|\mathbf{x}) = \theta(k|\mathbf{x};\widehat{\boldsymbol{\beta}})$, $\mathbf{x} \in \mathcal{X}$ and $k \in \mathcal{K}$. Replacing $\theta(\cdot|\cdot)$ by $\hat{\theta}(\cdot|\cdot)$ in (3), we get the corresponding maximum likelihood estimate $\hat{d}$ of the optimal decision rule d.

Let $\mathbf{S}(\boldsymbol{\beta})$ denote the score at $\boldsymbol{\beta}$ (that is, the $p(K-1)$-dimensional column vector with entries $\partial \ell_\epsilon(\boldsymbol{\beta})/\partial \beta_{kj}$), and let $\mathbf{H}(\boldsymbol{\beta})$ denote the Hessian at $\boldsymbol{\beta}$ (that is, the $(K-1)p \times (K-1)p$ matrix with entries $\partial^2 \ell_\epsilon(\boldsymbol{\beta})/\partial \beta_{k_1 j_1} \partial \beta_{k_2 j_2}$). The Newton–Raphson method for computing $\widehat{\boldsymbol{\beta}}$ is to start with an initial guess $\widehat{\boldsymbol{\beta}}^{(0)}$ and iteratively determine $\widehat{\boldsymbol{\beta}}^{(m+1)}$ from $\widehat{\boldsymbol{\beta}}^{(m)}$ according to the formula $\widehat{\boldsymbol{\beta}}^{(m+1)} = \widehat{\boldsymbol{\beta}}^{(m)} - [\mathbf{H}(\widehat{\boldsymbol{\beta}}^{(m)})]^{-1} \mathbf{S}(\widehat{\boldsymbol{\beta}}^{(m)})$. Here we employ the Newton–Raphson method with step-halving, in which $\widehat{\boldsymbol{\beta}}^{(m+1)}$ is determined from $\widehat{\boldsymbol{\beta}}^{(m)}$ according to the formula

$$\widehat{\boldsymbol{\beta}}^{(m+1)} = \widehat{\boldsymbol{\beta}}^{(m)} - 2^{-\mu} [\mathbf{H}(\widehat{\boldsymbol{\beta}}^{(m)})]^{-1} \mathbf{S}(\widehat{\boldsymbol{\beta}}^{(m)}),$$

where μ is the smallest nonnegative integer such that

$$\ell_\epsilon(\widehat{\boldsymbol{\beta}}^{(m)} - 2^{-\mu} [\mathbf{H}(\widehat{\boldsymbol{\beta}}^{(m)})]^{-1} \mathbf{S}(\widehat{\boldsymbol{\beta}}^{(m)})) \ge \ell_\epsilon(\widehat{\boldsymbol{\beta}}^{(m)} - 2^{-\mu-1} [\mathbf{H}(\widehat{\boldsymbol{\beta}}^{(m)})]^{-1} \mathbf{S}(\widehat{\boldsymbol{\beta}}^{(m)})).$$

We stop the iterations when $\ell_\epsilon(\widehat{\boldsymbol{\beta}}^{(m+1)}) - \ell_\epsilon(\widehat{\boldsymbol{\beta}}^{(m)}) \le \delta$. We typically take $\delta = 10^{-3}$.

Alternatively, we can compute $\widehat{\boldsymbol{\beta}}$ using a quasi-Newton approximation of the Hessian, such as the Broyden–Fletcher–Goldfarb–Shanno inverse updating technique (Fletcher, 1987). This method is usually faster than Newton–Raphson, since it does not require computation of the full Hessian. However, the Rao statistics (see Section 6 below) based on this approximation of the Hessian turn out to be too inaccurate. In practice we therefore alternate between quasi-Newton approximations and Newton–Raphson during the computations, especially when $(K-1)p$ is large.

6 Allowable Spaces

When modeling $\theta(k|\mathbf{x})$ with a linear model the remaining issue to be resolved is the choice of G. In this section we describe an algorithm for determining G in an adaptive fashion, given a family $\mathcal{G}$ of allowable spaces G that is assumed to have the following properties:

- for each $G \in \mathcal{G}$, the space G has dimension $p \geq p_{\min}$;
- there is only one $G \in \mathcal{G}$ with dimension $p_{\min}$;
- if $G \in \mathcal{G}$ has dimension $p > p_{\min}$, there is at least one subspace $G_0 \in \mathcal{G}$ of G with dimension $p' = p - 1$;
- if $G_0 \in \mathcal{G}$ has dimension p, there is at least one space $G \in \mathcal{G}$ containing G_0 as a subspace with dimension $p' = p + 1$.

We refer to $G \in \mathcal{G}$ with minimal dimension $p_{\min}$ as the minimal allowable space.

Initially, we use the minimal allowable space to model $\theta(k|\mathbf{x})$. Then we proceed with stepwise addition. Here we successively replace the $(p-1)$-dimensional allowable space G_0 by a p-dimensional allowable space G containing G_0 as a subspace, choosing among the various candidates for a new basis function by a heuristic search that is designed approximately to maximize the corresponding Rao statistic.

Specifically, let $\widehat{\boldsymbol{\beta}}^{(\mathbf{0})}$ be the maximum likelihood estimate of the coefficient vector corresponding to G, but subject to the constraint that the estimates of $\theta(k|\mathbf{x})$, $1 \leq k \leq K-1$, should be in G_0. Then the Rao statistic for testing the hypothesis that $\theta(k|\mathbf{x})$ is in G_0 for $1 \leq k \leq K-1$ is given by $R = [\mathbf{S}(\widehat{\boldsymbol{\beta}}^{(\mathbf{0})})]^T[\mathbf{I}(\widehat{\boldsymbol{\beta}}^{(\mathbf{0})})]^{-1}\mathbf{S}(\widehat{\boldsymbol{\beta}}^{(\mathbf{0})})$, where $\mathbf{I}(\widehat{\boldsymbol{\beta}}^{(\mathbf{0})}) = -\mathbf{H}(\widehat{\boldsymbol{\beta}}^{(\mathbf{0})})$ with $\mathbf{S}(\cdot)$ and $\mathbf{H}(\cdot)$ corresponding to G. (See (6e.3.6) of Rao (1973).)

Upon stopping the stepwise addition stage according to a rule described in Section 13.2, we proceed to stepwise deletion. Here we successively replace the p-dimensional allowable space G by a $(p-1)$-dimensional allowable subspace G_0 until we arrive at the minimal allowable space, at each step choosing the candidate space G_0 so that the Wald statistic for a basis function that is in G but not in G_0 is smallest in magnitude.

Specifically, let $\widehat{\boldsymbol{\tau}}$ be the $(K-1)$-dimensional vector of those entries of $\widehat{\boldsymbol{\beta}}$ that correspond to the basis function that would be deleted in going from G to G_0. Also, let $\hat{\mathbf{J}}$ denote the $(K-1)\times(K-1)$ submatrix of $[-\mathbf{H}(\widehat{\boldsymbol{\beta}})]^{-1}$ whose rows and columns correspond to these $K-1$ coefficients. Then the Wald statistic for testing the hypothesis that $\theta(k|\mathbf{x})$ is a member of G_0 for $1 \leq k \leq K-1$ equals $\widehat{\boldsymbol{\tau}}^T\hat{\mathbf{J}}\widehat{\boldsymbol{\tau}}$.

Motivation for the use of the Wald and Rao statistics is given in Section 13.1.

The specific models that are considered in this paper involve splines and their tensor products. We confine our attention to linear (rather than quadratic or cubic) splines, since they are easily interpretable in the context of classification, as will be clear from the examples in Section 10. In the present context, it is convenient to define an allowable space by listing its basis functions.

For $1 \leq m \leq M$, let K_m be an integer with $K_m \geq -1$; if $K_m = -1$ there are no basis functions depending on x_m; if $K_m = 0$, consider the basis function $B_{m0}(x_m) = x_m$; if $K_m \geq 1$, consider the basis function $B_{m0}(x_m) = x_m$, let x_{mk} for $1 \leq k \leq K_m$ be distinct real numbers, and consider the additional basis functions $B_{mk}(x_m) = (x_m - x_{mk})_+$ for $1 \leq k \leq K_m$.

Let G be the linear space having basis functions 1, $B_{mk}(x_m)$ for $1 \le m \le M$ and $0 \le k \le K_m$, and perhaps certain tensor products of two such basis functions. It is required that if $B_{lj}(x_l)B_{mk}(x_m)$ be among the basis functions for some $j \ge 1$, then $B_{l0}(x_l)B_{mk}(x_m) = x_l B_{mk}(x_m)$ and hence, if $k > 0$, $x_l x_m$ be among the basis functions. One reason for this requirement is that it leads to models that are simpler and easier to interpret (see Section 10); another is to reduce the variance associated with the overall modeling procedure.

It is easy to check that the collection $\mathcal{G}$ of such spaces satisfies the properties listed in Section 5. In particular, the minimal allowable space $G_{\min}$ for the POLYCLASS model is the space of constant functions. Thus the minimal model for (6) has $p = 1$, $B_1 = 1$ and $\theta(k|\mathbf{x}) = \beta_{k1}$ for $1 \le k \le K - 1$, so $P(k|\mathbf{x})$ does not depend on the vector $\mathbf{x}$ of predictors.

Given the basis of an allowable space G as defined above, it is obvious whether any given basis function can be deleted in one step.

Example. Let $M = 4$, $B_1 = 1$, $B_2 = x_1$, $B_3 = (x_1 - 1)_+$, $B_4 = x_2$, $B_5 = x_3$, and $B_6 = x_1 x_2$. Then $B_1, \ldots, B_6$ span an allowable space G. In this example, B_3, B_5 or B_6 could be removed and the remaining space would still be allowable. If one of the basis functions B_2 or B_4 were removed, however, the remaining space would not be allowable since it would still contain $B_6 = B_2 B_4$ (as well as B_3 in the case of removing B_2). The constant basis function B_1 can never be removed.

Let G_0 be the allowable space having basis functions 1, $B_{mk}(x_m)$ for $1 \le m \le M$ and $1 \le k \le K_m$, and perhaps certain tensor products of two such basis functions. To decide which basis function to add to this model, we compute the Rao statistic as described above:

- **(i)** for all spaces that can be obtained from G_0 by adding a basis function $B_{l0}(x_l) = x_l$ to G_0;
- **(ii)** for all allowable spaces that can be obtained from G_0 by adding a basis function to G_0 that is a tensor product of two basis functions $B_{lj}(x_l)$ and $B_{mk}(x_m)$, $l \ne m$, that are in G_0;
- **(iii)** for an allowable space that can be obtained from G_0 by adding a basis function based upon a potential new knot in predictor m for $1 \le m \le M$, located using the algorithm described in Section 13.3.

As the new space G we choose the one corresponding to the largest absolute value of the Rao statistic among those candidates listed above that are nonvacuous.

Example (continued). Corresponding to (i), we can add the basis function x_4 to the space in the above example. Corresponding to (ii), we can add $B_2 B_5 = x_1 x_3$, $B_3 B_4 = (x_1 - 1)_+ x_2$ or $B_4 B_5 = x_2 x_3$ to the space. The basis function $B_3 B_5 = (x_1 - 1)_+ x_3$ cannot be added, since the resulting space would not contain $B_2 B_5 = x_1 x_3$ so it would not be allowable. Corresponding to (iii), a basis function $(x_1 - x_{1k})_+$ with $x_{1k} \ne 1$, $(x_2 - x_{2k})_+$ or $(x_3 - x_{3k})_+$ could be added. No basis function of the form $(x_4 - x_{4k})_+$ could be added before x_4 is added.

7 Model Selection

During the combination of stepwise addition and stepwise deletion, we get a sequence of models indexed by ν, with the νth model having $(K-1)p_\nu$ parameters. For POLYCLASS the methods of selecting one model from this sequence that we consider are the (generalized) Akaike information criterion (AIC), an independent test set, and cross-validation.

AIC. Let $\hat{l}_\nu$ denote the fitted log-likelihood for the νth model, and let

$$\mathrm{AIC}_{\alpha,\nu} = -2\hat{l}_\nu + \alpha(K-1)p_\nu \tag{11}$$

be the Akaike information criterion with penalty parameter α for this model. We select the model corresponding to the value $\hat{\nu}$ of ν that minimizes $\mathrm{AIC}_{\alpha,\nu}$. In light of Kooperberg and Stone (1992) and our experience in the present investigation, we recommend choosing $\alpha = \log n$ as in the Bayesian information criterion (BIC) due to Schwarz (1978). (Choosing $\alpha = 2$ as in classical AIC tends to yield a model that is unnecessarily complex, has spurious features, and does not predict well on test data.) An interface, similar to the HARE interface (Kooperberg, Stone and Truong, 1995), allows the user to specify the penalty parameter.

Test set. Consider an independent test set $(\mathbf{X}_i^{\mathrm{TS}}, Y_i^{\mathrm{TS}})$, $1 \le i \le n^{\mathrm{TS}}$. Given an estimate $\hat{d}$ of the optimal decision rule, we can estimate its risk by

$$\hat{R}_\nu^{\mathrm{TS}} = \frac{1}{n^{\mathrm{TS}}} \sum_{i=1}^{n^{\mathrm{TS}}} C(\hat{d}_\nu(\mathbf{X}_i^{\mathrm{TS}}); Y_i^{\mathrm{TS}}).$$

Given a finite number of estimates of the optimal decision rule, we can chose the model having the smallest estimated risk. The minimum value of $\hat{R}_\nu^{\mathrm{TS}}$ is an estimate of the risk for classifying a new object using the final POLYCLASS model. This estimate is slightly biased downwards, since the test set is used to minimize the risk.

Cross-validation. Alternatively, cross-validation can be used to estimate the risk of $\hat{d}$. Here we first randomly divide the cases into $c \ge 2$ approximately equally-sized subsets. Then the following procedure is carried out for $j = 1, \ldots, c$:

- Fit a sequence of POLYCLASS models, as described in Section 6, to all cases not in the jth subset.
- For each $\alpha > 0$ select the model $\hat{\nu}_{j\alpha}$ that minimizes $\mathrm{AIC}_{\alpha,\nu}$ (11).
- For each α compute the loss $r_j(\alpha) = \sum C(\hat{d}_{\nu_{j\alpha}}(\mathbf{X}^i); Y_i)$, where the sum is over the cases in the jth subset (which were not used to fit these models).

For every α we now compute the cross-validated loss $R(\alpha) = n^{-1} \sum_{i=1}^{c} r_i(\alpha)$. Let $\tilde{\alpha}$ be the geometric mean of the endpoints of the interval of values of α that minimizes $R(\alpha)$. We proceed by fitting a sequence of POLYCLASS models to all data, using AIC with penalty parameter $\tilde{\alpha}$ to select the model.

Note that min $R(\alpha)$ is a slightly optimistic (downward biased) estimate of the risk for classifying a new object using the final POLYCLASS model.

8 Extended Polychotomous Regression Models

In some situations there may be evidence that the coefficients of certain basis functions are the same in several classes. Corresponding to each such basis function we would then want to divide the collection of classes into disjoint groups. Roughly speaking, the basis function would then be useful in distinguishing between groups but not between classes in a given group. During the stepwise deletion stage, it is possible to extend the class of polychotomous regression models described in

Section 4 to include models with such restrictions. This extension leaves the maximum likelihood estimation procedure essentially unchanged.

Let $B_1, \ldots, B_p$ be the basis functions in the current model. For $1 \leq j \leq p$, let $\mathcal{H}_j = \{\mathcal{H}_{j1}, \ldots, \mathcal{H}_{jV_j}\}$ be a partition of $\mathcal{K}$ such that $K \in \mathcal{H}_{jV_j}$. The interpretation is that the jth coefficients of $\theta(k|\mathbf{X}, \boldsymbol{\beta})$ will coincide for values k in the same set in the partition. As described below, initially we start with $V_j = K$. During the stepwise deletion process V_j will decrease; however, we can assume that $V_j \geq 2$ (otherwise, we can drop the jth basis function). Consider the $J \times K$ matrix Q defined by $Q(j,k) = \nu$ if $k \in \mathcal{H}_{j\nu}$. Thus we can rewrite (6) as

$$\theta(k|\mathbf{x}) = \sum_{j=1}^{p} \boldsymbol{\beta}_{jQ(j,k)} B_j(\mathbf{x}), \qquad \mathbf{x} \in \mathcal{X} \text{ and } k \in \mathcal{K}. \tag{12}$$

Using this equation instead of (6), the likelihood function (8), the elements of the score function (9), and the elements of the Hessian (10) are obtained accordingly.

To specify a model for $\theta(k|\mathbf{x})$ it is now necessary to specify G and $\mathcal{H} = \{\mathcal{H}_1, \ldots \mathcal{H}_p\}$. In particular the notion of an allowable space (Section 6) can be extended to allowable models, depending on both G and $\mathcal{H}$.

To define an allowable model, observe first that for every basis function $B_i(\cdot)$ there is a (possibly empty) set P_i of parent basis functions that are required to be in an allowable space G if $B_i(\cdot)$ is in G. A basis function $B_j(\cdot)$ in P_i is called a child of $B_i(\cdot)$.

Example. For $B_1(\mathbf{x}) = 1$ the set of parent basis functions is empty. If $B_2(\mathbf{x}) = x_1$ then $P_2 = \{B_1(\mathbf{x})\}$. If $B_3(\mathbf{x}) = (x_1 - x_{1k})_+$ then $P_3 = \{B_1(\mathbf{x}), B_2(\mathbf{x})\}$. If $B_4(\mathbf{x}) = x_2$ and $B_5(\mathbf{x}) = B_2(\mathbf{x})B_4(\mathbf{x})$ then $P_5 = \{B_1(\mathbf{x}), B_2(\mathbf{x}), B_4(\mathbf{x})\}$. If $B_6(\mathbf{x}) = B_3(\mathbf{x})B_4(\mathbf{x})$ then $P_6 = \{B_i(\mathbf{x}), 1 \leq i \leq 5\}$. If $B_7(\mathbf{x}) = (x_2 - x_{2k})_+$, $B_8(\mathbf{x}) = B_2(\mathbf{x})B_7(\mathbf{x})$ and $B_9(\mathbf{x}) = B_3(\mathbf{x})B_7(\mathbf{x})$ then $P_9 = \{B_i(\mathbf{x}), 1 \leq i \leq 8\}$.

An allowable space G and a partition $\mathcal{H}$ are said to form an allowable model if the following condition is satisfied: if k_1 and k_2 are in the same partition for the basis function $B_{i_1}(\mathbf{x})$, then k_1 and k_2 have to be in the same partition for every child $B_{i_2}(\mathbf{x})$ of $B_{i_1}(\mathbf{x})$. The interpretation behind this requirement is, that if two classes k_1 and k_2 have the same coefficient for, say, a basis function that is linear in x_1, then these two classes should also have the same coefficient for the basis functions x_1x_2, $(x_1 - x_{1k})_+$ and all other children of x_1; if so, then x_1 would not be useful in distinguishing between class k_1 and k_2.

As we mentioned above, we only use extended polychotomous regression models during the deletion stage. In other words, when using extended polychotomous regression models, the addition stage is identical to the addition for the regular polychotomous regression models as described in Section 6. In particular, during the addition stage every set $\mathcal{H}_{ij}$ has size one.

During the deletion stage we now replace allowable models of dimension r by allowable models of dimension $r-1$. This is done by combining two sets of the partition for a particular basis into one set (thereby reordering the other sets if required). If the partition for a basis function has only two sets left and if no children of this basis function remain, then the basis function can be dropped from the model entirely by combining the two sets in the corresponding partition. Wald statistics are again used to select the 'best' model of dimension $r-1$; in the present context, they have only one degree of freedom.

Example. Consider the example in Section 6, with $K = 4$. Note that in this example P_1 is the empty set, $P_2 = P_4 = P_5 = \{B_1(\mathbf{x})\}$, $P_3 = \{B_1(\mathbf{x}), B_2(\mathbf{x})\}$ and $P_6 = \{B_1(\mathbf{x}), B_2(\mathbf{x}), B_4(\mathbf{x})\}$. An allowable model during the deleting stage could now, for example, have as $V_1 = 4$ partitions for $B_1(\mathbf{x})$: the sets $\mathcal{H}_{j1} = \{j\}$, for $1 \leq j \leq 4$; $V_2 = 3$ partitions for $B_2(\mathbf{x})$: the sets $\mathcal{H}_{12} = \{1\}$, $\mathcal{H}_{22} = \{2,3\}$ and $\mathcal{H}_{32} = \{4\}$; $V_3 = 2$ partitions for $B_3(\mathbf{x})$: the sets $\mathcal{H}_{13} = \{1,2,3\}$ and $\mathcal{H}_{23} = \{4\}$; $V_4 = 3$ partitions for $B_4(\mathbf{x})$: the sets $\mathcal{H}_{14} = \{2\}$, $\mathcal{H}_{24} = \{3\}$ and $\mathcal{H}_{34} = \{1,4\}$; $V_5 = 3$ partitions for $B_5(\mathbf{x})$: the sets $\mathcal{H}_{15} = \{1\}$, $\mathcal{H}_{25} = \{2\}$ and $\mathcal{H}_{35} = \{3,4\}$; $V_6 = 2$ partitions for $B_6(\mathbf{x})$: the sets $\mathcal{H}_{16} = \{2,3\}$ and $\mathcal{H}_{26} = \{1,4\}$. Note that for $B_6(\mathbf{x})$ classes 2 and 3 have to be in the same set since they are in the same set for $B_2(\mathbf{x})$, while classes 1 and 4 have to be in the same set since they are in the same set for $B_4(\mathbf{x})$. At the next deletion, we cannot merge any sets of the partitions for $B_1(\mathbf{x})$ or $B_2(\mathbf{x})$, but we can merge the remaining two sets of the partition for $B_3(\mathbf{x})$ or $B_6(\mathbf{x})$, which would then be removed. We can also merge the sets $\mathcal{H}_{14}$ and $\mathcal{H}_{24}$ of the partition of $B_4(\mathbf{x})$, but we cannot merge either of these sets with $\mathcal{H}_{34}$ since that would cause a conflict with $B_6(\mathbf{x})$. Finally, we can merge any two of the three sets of the partition for $B_5(\mathbf{x})$.

If extended polychotomous regression models are used, not all models are fitted using maximum likelihood. In particular, to save computing time, when the expected drop in the log-likelihood is small, as measured by the Wald statistic, only a quadratic approximation to the log-likelihood is used. See Section 13.1 for details.

Model selection for extended polychotomous regression models employs one of the procedures described in Section 7. In the user interface there is the option to use either regular polychotomous regression models or extended polychotomous regression models.

9 Least Squares Approximation of the Addition Process

The addition process, as described in Section 6 can become computationally too expensive for large data sets. Since quasi-Newton updates for the Hessian matrix do not suffice during addition, we need to compute the full Hessian, which requires $O(K^2p^2n)$ floating point operations (flops), where K is the number of classes, p the number of basis functions and n the number of cases. The computation of a Rao statistic requires $O(K^2pn)$ flops, but typically, for adding a basis function to a model with p basis functions, we compute approximately $O(p)$ Rao statistics, so the computation of all Rao statistics at that stage involves $O(K^2p^2n)$ flops. If the largest model has $P_{\max}$ basis functions the total number of flops required is $O(K^2P_{\max}^3n)$. For the phoneme recognition problem discussed in Section 11, $n = 112115$ and $K = 45$. According to the rule in Section 13.2, $P_{\max}$ is approximately 200. The computations would thus require $O(10^{15})$ flops. We estimated that this would require several years of cpu-time on the SGI workstation that we used for most of our computations.

If we were to use a quasi-Newton (instead of Newton–Raphson) algorithm, we would not have to compute any full Hessians. However, the number of iterations needed is typically larger using a quasi-Newton algorithm. The substantial costs of computing the Rao statistics would not be reduced. Overall, we can expect that a quasi-Newton algorithm would be approximately 60% faster than a Newton–Raphson algorithm, but at the expense of less accurate Rao statistics.

These computations lead us to consider the following least squares approximation to the stepwise addition process when dealing with large data sets. Let $\mathbf{Z}_i$, $1 \leq i \leq n$, be the column vector of length K, of which the kth element is $\text{ind}(Y_i = k)$ (ind is the usual indicator function). Now set

$$\theta(k|\mathbf{x};\boldsymbol{\beta}) = \sum_{j=1}^{p} \beta_{jk} B_j(\mathbf{x}), \qquad \mathbf{x} \in \mathcal{X} \text{ and } k \in \mathcal{K}. \tag{13}$$

We estimate $\boldsymbol{\beta}$ by minimizing

$$V(\boldsymbol{\beta}) = \sum_i \sum_k [Z_{ik} - \theta(k|\mathbf{X}_i;\boldsymbol{\beta})]^2.$$

The selection of the new basis function B_j is also carried out by minimizing $V(\boldsymbol{\beta})$, while otherwise the same allowable spaces as in POLYCLASS are used (see Section 6). The stepwise addition part of the model selection can now be carried out in $O(50P_{\max}^2 n)$ flops, or a few hours for the phoneme recognition problem. See Section 13.4 for more details. Note that this stepwise addition least squares algorithm is similar to the MARS algorithm in Friedman (1991).

The least squares problem described above eventually yields $P_{\max}$ basis functions. We now fit a POLYCLASS model with these basis functions using the method described in Section 4 and a quasi-Newton algorithm. The stepwise deletion procedure remains the same as in Section 6, except that we use the quasi-Hessian for the computation of the Wald statistics. It has been our experience that, though the quasi-Hessian is not adequate for stepwise addition, it does give satisfactory results during stepwise deletion. See Section 13.5 for details. The idea for using a MARS-like algorithm as a preprocessor for POLYCLASS was inspired by Bose (1994) and Hastie, Tibshirani and Buja (1994).

For the example mentioned above, using the approximations described in this section, the cpu time can be reduced to about 60 days. Using a network of workstations this was further reduced to approximately one day. See Section 13.5 for details.

10 Examples

We compared the performance of POLYCLASS to a variety of other classification methods, such as linear discriminant analysis (LDA), flexible discriminant analysis (FDA) (Hastie, Tibshirani and Buja, 1994), classification using splines (CUS) (Bose, 1994), and classification and regression trees (CART) (Breiman, Friedman, Olshen and Stone, 1984), on the waveform and the digit examples from the CART monograph. These examples have been used by many researchers for comparison studies.

Linear discriminant analysis is a classical method which has been used for decades. This method assumes that the predictors have multivariate normal distributions with different means, but the same covariance matrix, for each class. The distributional parameters are estimated and the resulting decision rule is linear in the predictors. See Mardia, Kent and Bibby (1979) for more details.

When the optimal decision rule is nonlinear, linear discriminant analysis may perform poorly. This has motivated researchers to come up with methods capable of approximating highly nonlinear boundaries. CART (Breiman, Friedman, Olshen and Stone, 1984) is one of the most well-known methods. It splits the ranges of individual predictors to separate the measurement vectors from different classes. The resulting boundaries are union of hyper-rectangles. The option of splitting the predictor space by linear combinations of the predictors is also available in CART.

Some boundaries might be hard to approximate by the class of hyper-rectangles used in CART. Several other methods have been developed in the last few years that are capable of approximating highly nonlinear class boundaries. CUS (Bose, 1994), uses an additive cubic spline model to approximate the posterior class-probabilities. However, in contrast to POLYCLASS, this model is estimated using least squares regression, similar to the procedure described in Section 9. The model selection is carried out using a stepwise deletion algorithm and cross-validation.

Breiman and Ihaka (1984) observed that discriminant analysis can also be performed by multiple-response linear regression using optimal scalings to represent the classes. Hastie, Tibishirani and Buja (1994) replaced linear regression by nonparametric regression methods such as MARS or BRUTO and thus developed the FDA classification method.

10.1 Waveform data

In this example (the detailed description of which can be found in Breiman, Friedman, Olshen and Stone, 1984) there are three classes and 21 predictors. Let, h_1, h_2 and h_3 be the triangular 'waveforms' defined by $h_1(i) = \max(6- \mid i-7 \mid, 0)$, $h_2(i) = h_1(i-8)$ and $h_3(i) = h_1(i-4)$, for $i = 1, \ldots, 21$.

The distributions of the 21 predictors conditional on the class of the observation are now defined by

$$x_i = uh_1(i) + (1-u)h_2(i) + \epsilon_i \qquad \text{for class 1,}$$

$$x_i = uh_1(i) + (1-u)h_3(i) + \epsilon_i \qquad \text{for class 2}$$

and

$$x_i = uh_2(i) + (1-u)h_3(i) + \epsilon_i \qquad \text{for class 3,}$$

where u has uniform distribution on (0,1), and the ϵ_i are independent random variables with a standard normal distribution.

The training sample of size 300 was generated using equal priors for each class. For the regular POLYCLASS models, the extended POLYCLASS models, CUS and CART the model selection was performed using ten-fold cross validation. In LDA no model selection is used, and in FDA the model selection is done using a generalized cross-validation criterion. After the models were fitted, the classification was evaluated on an independent test set of size 5000 that was generated the same way as the training set. The whole experiment was repeated 10 times.

Misclassification errors on the training set and the test set based on the ten repetitions, are reported in Table 1. The typical standard errors ranged from .005 to .015. For the methods using cross-validation, the training column in this table contains the resubstitution errors (which can thus be compared with the training set errors for LDA and FDA), while the cross-validation column contains the cross-validation estimate of the error rate.

The results in Table 1 show that POLYCLASS performed quite satisfactorily in this simple example. Except for CART, the other methods performed about the same as or little better than LDA. POLYCLASS had very similar error rates as LDA. FDA, using BRUTO for the nonparametric regression seems to have a slight edge over the other nonlinear methods. Note that LDA, CUS and FDA with BRUTO or MARS (degree 1) use additive models. In this example additive models are probably sufficient, so that the other methods, including POLYCLASS, are somewhat overly complicated. We note that POLYCLASS performs better than the other nonadditive models.

Table 1. Misclassification error rates for the waveform data; averages based on 10 simulation runs.

Method	Training	Test	Cross-validation
POLYCLASS (regular)	.135	.200	.184
POLYCLASS (extended)	.130	.212	.200
LDA	.134	.199	
FDA (BRUTO)	.107	.174	
FDA (MARS)	.114	.197	
FDA (MARS, degree 2)	.074	.216	
CUS	.120	.184	.176
CART	.192	.315	.285
CART (Lin. Comb.)	.129	.241	.234

In Figure 1 we show some plots related to one particular regular POLYCLASS fit. This fit was based on a training set of size 300. The selected model had 14 basis functions: the constant function, nine linear functions, a knot for predictor 13, a knot for predictor 16, an interaction between x_8 and x_9 and an interaction between x_{13} and x_{16}. This is not the best POLYCLASS fit. Most POLYCLASS models selected for different realizations of the waveform data were linear, yielding smaller test set errors. However, we choose this model to illustrate some of the features of POLYCLASS. In particular, in the left top corner of Figure 1, we show the decision boundaries as a function of the value of predictors 13 and 16, when all other predictors have the value 4. In the other panels of Figure 1 we show perspective plots of the probability estimates. We observe from this plot that large values of x_{13} and x_{16} together are associated with class 3 and that small values of x_{13} and x_{16} together are associated with class 2. This seems reasonable in light of the true measurement models for these classes.

10.2 Digit data

This is another example taken from the CART monograph (Breiman, Friedman, Olshen and Stone, 1984). In electronic watches, calculators, etc the ten digits are displayed by four vertical and three horizontal lights in different on-off combinations. Any digit, therefore, is associated with a vector $(x_1, \ldots, x_7)$ (the ordering of the lights being arbitrarily fixed), where each x_i takes the values 0 or 1 according to the corresponding light being on or off, respectively. In this example, the digits are from a faulty calculator where for any digit each x_i takes the right value with probability .9. The training data has 200 cases and the test data has 5000 cases with uniform priors being used for the ten digits.

Misclassification errors on the training set and the test set based on the ten repetitions are reported in Table 2. The typical standard errors ranged from .005 to .010. Since the predictors take only values 0 and 1, FDA with BRUTO as the regression technique, CUS, and CART using linear combinations are not applicable. The results show that LDA nearly achieves the optimal rate of 26%. The other methods, including POLYCLASS, perform better than CART. In this example FDA with MARS can successfully pick up the simple structure of this problem and fits a linear model.

The simple structure of the digits example allows to identify the optimal model. When we use

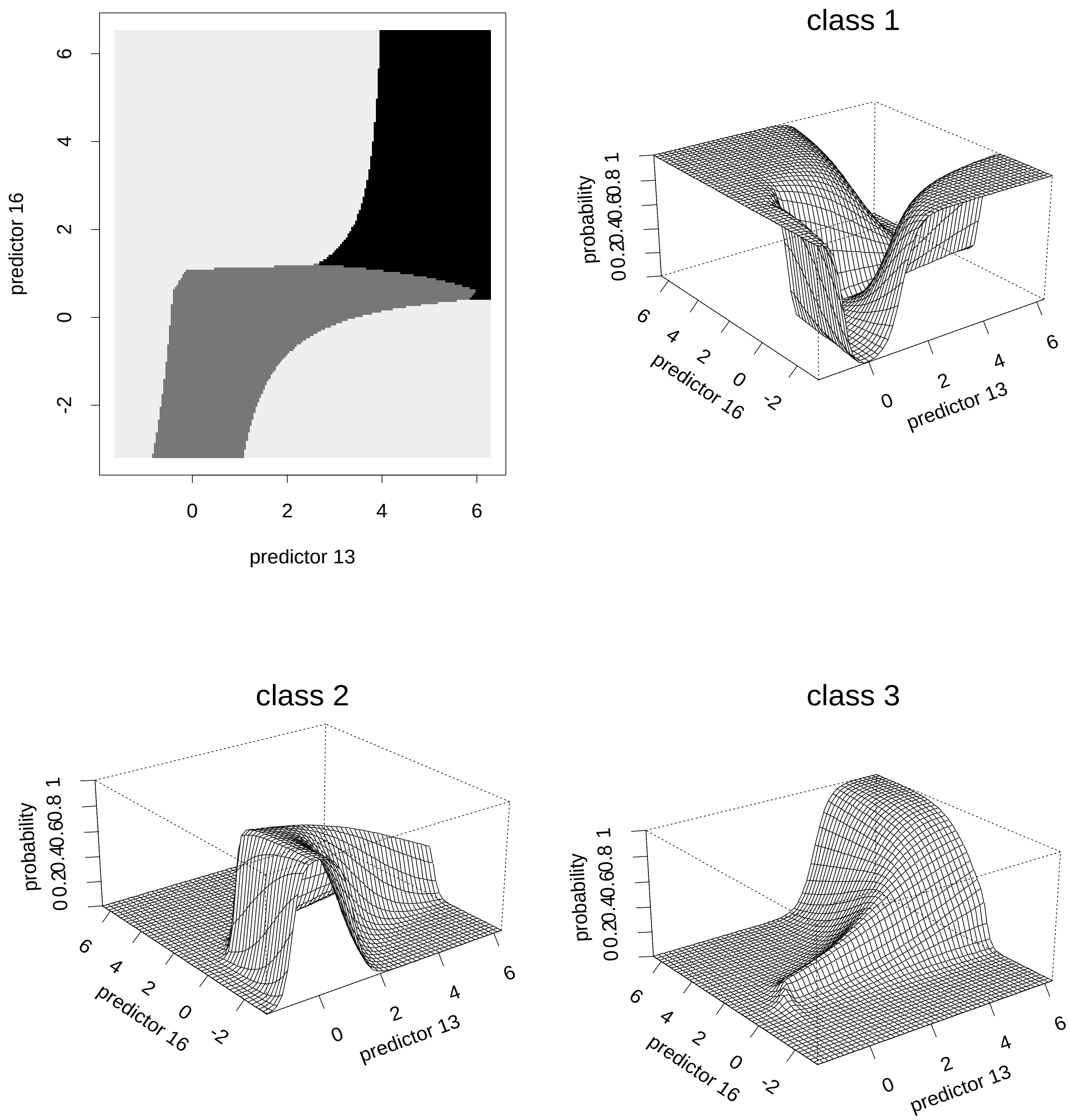

Fig. 1. Classification map and estimated conditional class probabilities as a function of predictors 13 and 16, when all other predictors are equal to 4. White: class 1; grey: class 2; black: class 3.

TABLE 2. Misclassification error rates for the digit data; averages based on 10 simulation runs.

Method	Training	Test	Cross-validation
POLYCLASS (regular)	.218	.315	.286
POLYCLASS (extended)	.229	.290	.289
LDA	.209	.270	
FDA (MARS)	.209	.270	
FDA (MARS,degree 2)	.228	.296	
CART	.258	.330	.295

TABLE 3. How often POLYCLASS ends up with the 'ideal' model in 100 simulations.

n	Model selection	Missed some linear basis functions	The ideal model	Extra parameters but no extra basis functions	Extra basis functions
200	BIC, regular	83	17	NA	0
200	BIC, extended	1	9	17	73
200	CV, regular	46	32	NA	22
200	CV, extended	0	10	9	81
1000	BIC, regular	0	100	NA	0
1000	BIC, extended	0	60	37	3
1000	CV, regular	0	100	NA	0
1000	CV, extended	0	39	16	45

the regular POLYCLASS models, the best model would have eight basis functions, the constant, and each of the seven linear functions. These same basis functions are in the best model when extended POLYCLASS models are used. However, only one nonzero coefficient would be needed for each linear basis function. For example, for predictor 1, which is the horizontal light at the top of the digit, the coefficient would ideally be 0 for classes 0, 2, 3, 5, 6, 7, 8 and 9, since for each of these digits the light is 'on' when the digit is correctly displayed, while the coefficient would be different from 0, but identical to each other, for classes 1 and 4, since for each of these digits the light is 'off' when the digit is correctly displayed. In particular, in one example we observed that in one regular POLYCLASS model for the digit data the coefficient of x_1 for class 1 was estimated as -4.04, for class 2 as -0.75, for class 3 as 0.76, for class 4 as -4.24, for class 5 as 0.30, for class 6 as -1.09, for class 7 as 0.61, for class 8 as -0.53, for class 9 as 0.44 and for class 0 as 0. On the other hand, when we used the extended POLYCLASS models on the same data, the estimate of the coefficient of x_1 for class 1 and 4 was both -4.40, while it was 0 for the other eight classes, thus in the terminology of Section 8 the basis function x_1 had two partitions. (Note that we can also use a model of this form to fit the exact conditional class probabilities.)

We carried out a small simulation study to see how often POLYCLASS selects the best classification model. For various sample sizes, using model selection with BIC and 10-fold cross validation (CV), we counted how often out of 100 simulations regular POLYCLASS models and extended POLYCLASS models end up with the 'ideal' model. The results are summarized in Table 3.

When a linear basis function was missed, it corresponded to the vertical light at the bottom right (x_6) in most cases. Class 2 is the only class for which this light is 'off' when the light is functioning properly, and, when all lights function properly, this class can be distinguished from all other classes on basis of the other lights. When basis functions were missed, usually no more than one was missed. Rarely were more than two interactions included.

As we can see from Table 3, when the sample size gets large POLYCLASS usually selects the correct model. For $n = 1000$ it appears that BIC works as least as well and is much faster, while for $n = 200$ cross-validation seems to work better. (We should point out that missing a linear basis function affects the error rate much more than inclusion of an interaction or an extra parameter.) For $n = 200$ the linear basis functions are all included more often for the extended POLYCLASS models, suggesting that the extended models perform better for smaller sample sizes.

Figure 2 demonstrates the effectiveness of POLYCLASS in estimating the true posterior class probabilities. One training set with 200 cases and one with 1000 cases were generated. Regular and extended POLYCLASS models were fit to both training sets using ten-fold cross-validation. For each of these four models the 10 true and estimated conditional class probabilities were calculated for all 128 possible 7-tuples of 0's and 1's as predictors. They were plotted against each other on logarithmic scales. It is evident that the extended POLYCLASS models are more accurate in estimating the true posterior probabilities than the regular POLYCLASS models. This is expected, since the ideal model is indeed an extended POLYCLASS model. It turned out that the regular POLYCLASS model with $n = 200$ in Figure 2 had six linear terms, but missed predictor 6. The extended model with $n = 200$ included all linear terms, as well as one interaction. The model had 24 parameters, which is 8 more than the optimal number. For $n = 1000$ both models were optimal, having exactly seven linear basis functions, and for the extended model the number of parameters was reduced to 16.

11 Phoneme Recognition

Our last example is taken from from the area of speech recognition. The source of this huge data set is the Center for Spoken Language Understanding in Portland, Oregon (Cole et al. 1992, 1994). It consists of 2165 utterances from telephone calls, which are numbers that typically are parts of addresses, zip-codes and street numbers. Each utterance was processed by one or more listeners, who produced a time-aligned phonetic description of the utterance. For example, for one particular utterance, "3o3" (three-oh-three), it was determined that from 1 millisecond (ms) to 167 ms, the speaker produced phoneme `T`, followed by phoneme `r` from 167 ms to 193 ms, and so on. It should be noted that the person who decided which phoneme was spoken was not aware of the text of the utterance. The phoneme transcription, which we obtained from the International Computer Science Institute (ICSI) in Berkeley, California, is based on the LIMSI phonetic alphabet (Guavain et al. 1994).

The utterances were also processed to produce perceptual linear predictive (PLP) features. Every 12.5 ms the audible spectrum, based on a concentric 25 ms piece of sound, is determined. Since we consider telephone data, which is sampled at the frequency of 8 kHz, there are 200 observations of the sound wave in such a 25 ms interval. A Hamming window is applied to these 200 observations before the spectrum is estimated using the discrete Fourier transform. The estimated spectrum is next transformed to yield a critical-band integrated power spectrum with an equal-loudness pre-emphasis and a cube root nonlinearity to simulate the auditory intensity-loudness relation. Then the eighth order autoregressive all-pole model of the transformed spectrum is obtained. The coeffi-

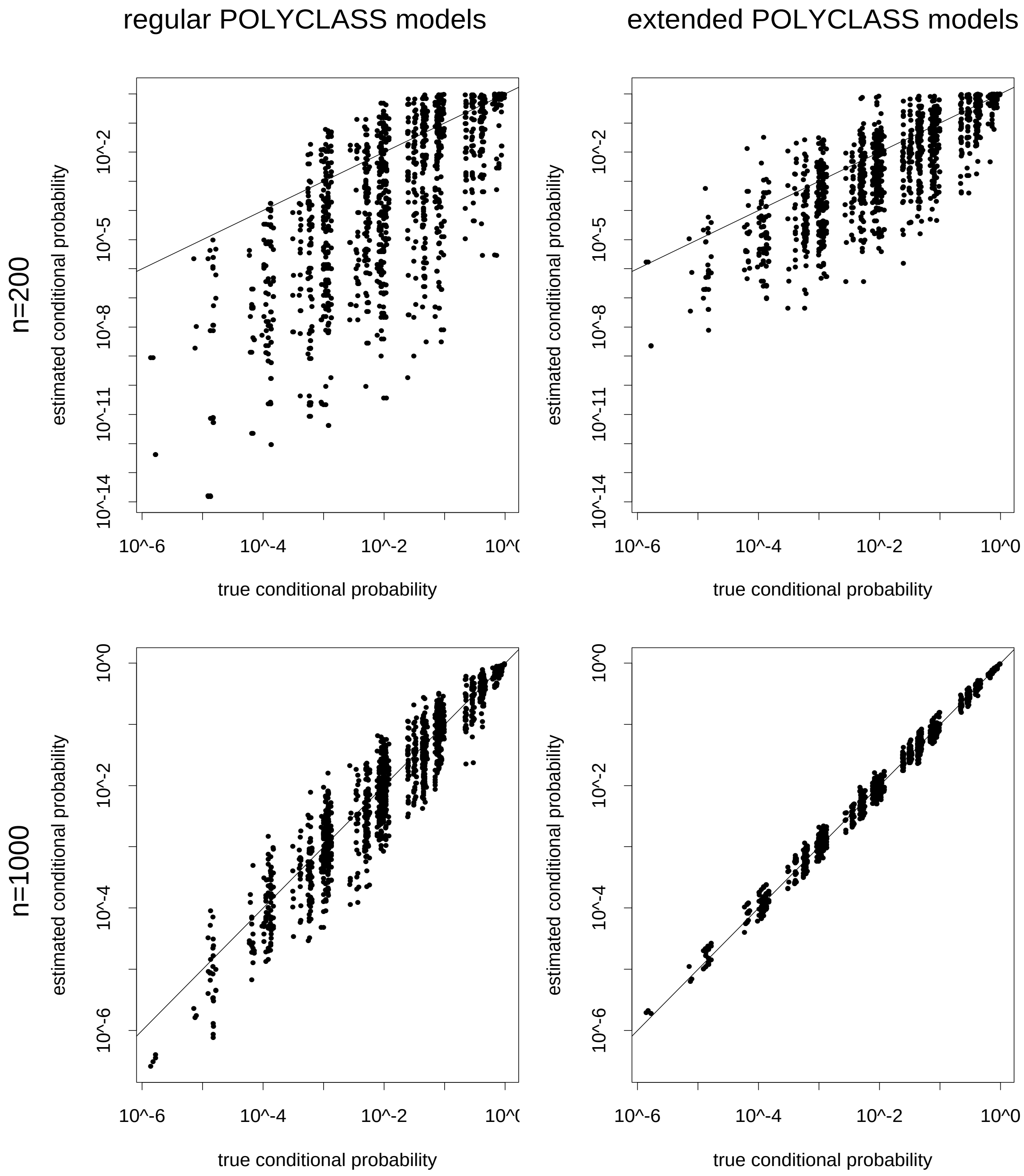

Fig. 2. Estimated conditional class probabilities versus true conditional class probabilities for all 128 possible combinations of the predictors and all 10 classes for the digit data. Left: regular POLYCLASS models; right: extended polyclass models; top: probabilities estimated using 200 cases; bottom: probabilities estimated using 1000 cases.

cients of the Fourier transform representation of the log-magnitude of this model are known as its cepstral coefficients. The PLP features (Hermansky, 1990; Rabiner and Juang, 1993; Bourlard and Morgan, 1994) that we used are the log-gain of the model (similar to the variance) and the next eight cepstral coefficients (similar to the autoregressive coefficients).

The goal in our analysis is to estimate the probability distribution over all phonemes at intervals of 12.5 ms based on the (nine) features available at that time point as well as the features available at the c time points, each 12.5 ms apart, before and after the point at which we want to estimate the phoneme distribution.

Such a probability distribution (or, more precisely, a likelihood that is obtained by weighting the estimated probabilities by the empirically determined frequencies of the phonemes) can be used as input to train (estimate) a hidden Markov model, which in turn can be used for automatic speech recognition (Bourlard and Morgan, 1994). In the hybrid approach described by Bourlard and Morgan, a multilayer perceptron network (a type of artificial neural network) is used to estimate these probabilities.

There were 45 different phonemes, yielding 247039 cases (12.5 ms intervals). We randomly divided the data into a training set of approximately 110000 cases and a test set and final test set of about 65000 cases each.

We used the vector of features at seven different time points, so that $c = 3$ above. The eight cepstral coefficients were used exactly as we received them from ICSI. Since some speakers speak more loudly than others, the log-gain by itself is not an informative predictor of the phoneme that is being spoken. Differences in the log-gain may be more informative. If $e(i)$ is the log-gain at time instance i, we used

$$d(i) = e(i) - \frac{1}{7} \sum_{j=-3}^{3} e(i+j)$$

instead of $e(i)$.

The standard POLYCLASS methodology described in Sections 2–7 would be practically impossible to apply to the phoneme recognition data, for which $K = 45$, $M = 9 \cdot 7 = 63$ and $n = 112115$. Instead, we used the least squares approximation for the stepwise addition procedure (see Sections 9 and 13.4) while we carried out the actual fitting of the model on a network of workstations (see Section 13.5). The largest model that we fit had 350 basis functions. This number is much larger than the default value of 193 (see Section 13.2), but initial analysis suggested that a larger model would yield much better results. (See also the discussion of Figure 3 below.) This maximum number of 350 basis functions was constrained by the computing resources that were available to us. We believe that a larger number of basis functions would give better results. Exhaustion of our resources also prevented us from applying the stepwise deletion algorithm on the largest model. However, intermediate results, not reported here, suggest that the deletion of some basis functions would not significantly improve our results.

Of the 350 basis functions that were selected by the MARS-like algorithm, one is the constant function, 31 are of the form x_i, 45 are of the form $(x_i - x_{ik})_+$, 134 are of the form $x_i x_j$, 87 are of the form $(x_i - x_{ik})_+ x_j$ and eleven are of the form $(x_i - x_{ik})_+(x_j - x_{jl})_+$. Thus, of the 63 features, 32 are not used. Of the remaining 31, ten are involved in all types of basis functions, ten more are involved in all types of basis functions except for $(x_i - x_{ik})_+(x_j - x_{jl})_+$, eight are involved in basis functions of the types x_i, $(x_i - x_{ik})_+$, $x_i x_j$ and $x_i(x_j - x_{jk})_+$. Finally, two features have basis functions of the types x_i, $(x_i - x_{ik})_+$ and $x_i x_j$ only and one feature appears only linear in the

Table 4. The features in the POLYCLASS model.

	time						
cepstral coefficient	-3	-2	-1	0	1	2	3
log-gain	5	4	3	5	3		
lag one	5		4	5			4
lag two	4		5	2			5
lag three	4			4			5
lag four	5			5	1		5
lag five	3			4			4
lag six				4			3
lag seven	3		2				3
lag eight	3		3				4

model.

The 63 features can be organized in a 9 (cepstral coefficients) $\times$ 7 (time points) table. If we label the features from '1', for the feature that occurs only linearly, to '5', for the features that are involved in all types of basis functions, and we ignore the entries for the 32 features that are unused, we obtain Table 4. From this table we clearly see that the most important information is obtained from time points -3 (37.5 ms before the phoneme was spoken), 0 (when the phoneme is spoken) and 3 (37.5 ms after the phoneme was spoken). This table suggests that, in retrospect, it would have been better to use the cepstral coefficients at more than seven time points. (We also see that the log-gain and the shorter lags are more important than the longer lags.)

In Figure 3 we report the misclassification rate and the fitted log-likelihood

$$\frac{\sum_i \log P(Y = Y_i | \mathbf{X} = \mathbf{X}_i)}{n}$$

for the training set and both test sets combined. From these graphs it appears that the fit would continue to improve if we were to increase the number of basis functions.

As mentioned earlier, in this particular application the estimation of conditional class probabilities is more important than classification, since these probabilities can be used as the inputs to the hidden Markov model for the approach to speech recognition described in Bourlard and Morgan (1994) . POLYCLASS is particularly useful in this situation, since, unlike most other classification methods, it provides viable estimates of the conditional class probabilities. In Figure 4 we plot the estimated probability that a case is a particular phoneme grouped in bins of size 0.01 on the horizontal axis and the fraction of cases with that probability that corresponded to the correct phoneme on the vertical axis. Note that every case contributes 45 observations to this graph: one observation per candidate phoneme. These graphs are extremely close to the ideal straight line (fraction true class) = (estimated probability) for both the test sets (left side) and the training set (right side).

Clearly, not all phonemes are correctly estimated with the same probability. In Figure 5 we plot the average probability, over the two test sets combined, assigned to each phoneme. We see from Figure 5 that this probability is much larger for the frequently occurring phonemes than for the infrequently occurring ones.

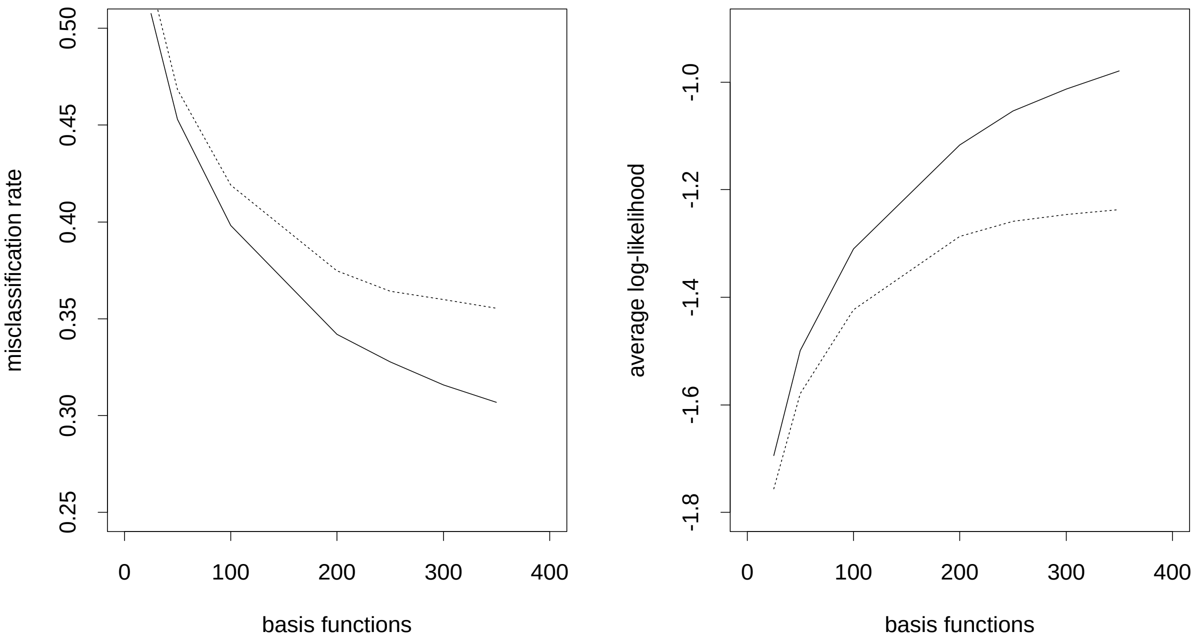

Fig. 3. Misclassification rate (left) and fitted log-likelihood (right) versus the number of basis functions. Solid = training set, dashed = test set combined with final test set.

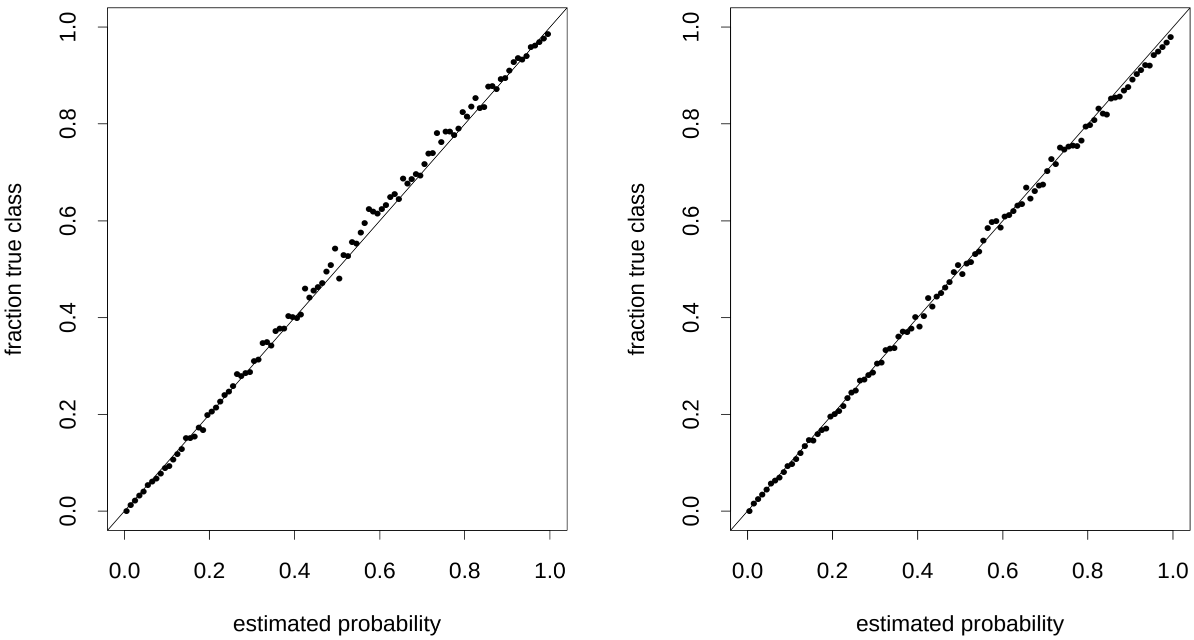

Fig. 4. Fraction of phonemes that correspond to the true class versus the estimated probability. Data has been binned in bins of size 0.02. Left = training set. right = test set combined with final test set.

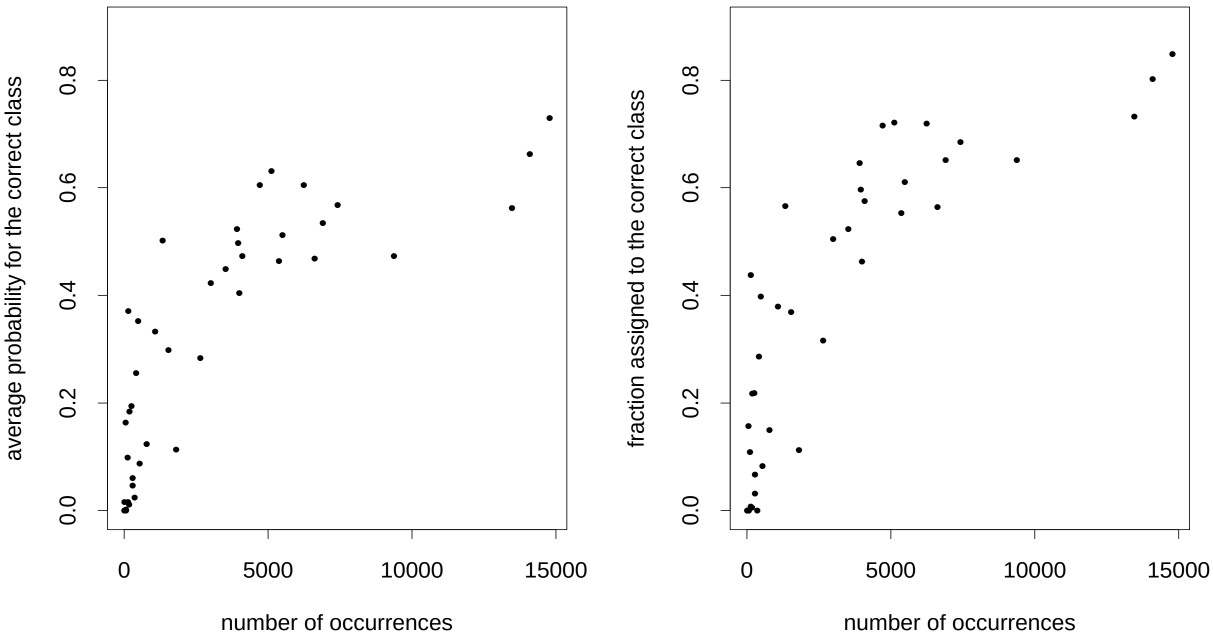

Fig. 5. Average probability assigned to the correct class and fraction correctly classified versus the class frequency. Both test sets combined.

It is also of interest to determine for a given phoneme which incorrect phonemes are assigned high probabilities. In particular, the 45 phonemes that we considered are sometimes divided into approximately nine phonetic classes. It is natural to expect that a misclassified phoneme is frequently classified as another phoneme in the same phonetic class. We define the $K \times K$ matrix $\mathbf{A}$ whose (i, j) entry is

$$\sum_{k=1}^{n} \widehat{P}(Y_k = i|\mathbf{X}_k)\mathrm{ind}(Y_k = j).$$

Clearly, a_{ij} is going to be larger if class i or class j occurs more often in our sample. Therefore, we normalize by setting $\tilde{a}_{ij} = a_{ij}/\sqrt{n_i n_j}$, where n_i is the number of cases from class i. As expected if $\tilde{a}_{ij}$ was large, $\tilde{a}_{ji}$ was large too. In Table 5 we summarize six pairs that yielded the largest ten values of $\tilde{a}_{ij}$ (as well as the two corresponding values of $\tilde{a}_{ji}$ that were not among the top ten).

As can be seen from Table 5, frequently two phonemes that are mixed up are members of the same phonetic class, as we expected. In fact, among the 50 largest values of $\tilde{a}_{ij}$ there were 29 pairs of phonemes for which either or both $\tilde{a}_{ij}$ and $\tilde{a}_{ji}$ were among those 50. Of these 29 pairs, 14 involved two phonemes from the same phonetic class.

In Table 6 we summarize misclassification rates for the phoneme data for various methods. We compare POLYCLASS to linear discriminant analysis using the 63 features, linear discriminant analysis using the 349 nonconstant basis functions selected by MARS (inspired by Hastie, Tibshirani and Buja (1994)), softmax (using only linear regression on the basis functions), and CART, with and without linear combinations. Misclassification rates for neural networks in exactly this data set were not available from either OGI or ICSI. However, they reported that using somewhat different features and/or time periods they get test set error rates of approximately 25%. At the end of this

TABLE 5. The ten largest values of the misclassification index.

phoneme i			phoneme j				
	class	example		class	example	$\tilde{a}_{ij}$	$\tilde{a}_{ji}$
tcl	plossive	fat	·	silence		0.177	0.157
f	fricative	fan	s	fricative	sue	0.170	0.169
i	vowel	beet	e	vowel	bait	0.169	0.156
c	vowel	bought	o	vowel	boat	0.147	0.155
E	vowel	bet	ˆ	vowel	but	0.128	0.094^a
r	semivowel	red	X	reduced vowel	butter	0.114	0.108^b

a: this was the 14th largest $\tilde{a}_{ij}$

b: this was the 11th largest $\tilde{a}_{ij}$

TABLE 6. Classification error rates for the phoneme data.

	training set	test set	final test set
sample size	112115	67731	67193
POLYCLASS	30.68%	36.01%	35.06%
LDA - 63 features	47.85%	49.88%	48.95%
LDA - 349 basis functions	40.04%	42.79%	41.80%
softmax - 349 basis functions	38.82%	40.68%	39.98%
CART	44.80%	53.30%	52.55%
CART - linear combinations	40.77%	48.87%	47.81%

section we briefly discuss how these results compare to the POLYCLASS results.

Table 6 shows that POLYCLASS has the best test set error, 4% better than the next best error rate (softmax) and 13% better than LDA on the features. Since we did not employ stepwise deletion here, the model selection for POLYCLASS is independent of the test set. The difference in performance between the test set and the final test set is therefore due to random variation; the same is true for all other methods but CART, which does use the test set for model selection. The regular CART tree was formed by using 100 as the minimum atom size for splitting and had 828 terminal nodes. When linear combination splits were allowed for nodes of size at least 1000, the resulting tree had 515 terminal nodes.

We now return briefly to the comparison of the POLYCLASS results to the neural network results from OGI and ICSI. The higher misclassification errors of POLYCLASS could be attributed to the following reasons.

1. The set of features that we considered as possible predictors are far from optimal. In particular we already observed in Table 4 that the features that we used at times -2, -1, 1 and 2 were rarely selected, while the features at times -3 and 3 were frequently selected. Further communication with OGI and ICSI suggested that the actual time instances at which they use the features are chosen more optimally, based on considerable experience. For example, when we used the times -7, -4, -2, 0, 2, 4 and 7, instead of $-3, \ldots, 3$, the misclassification errors for the two LDA based methods as well as softmax dropped by approximately 4–5%. To save computing resources we did not apply the other methods to this data set.

2. A model with more than 350 basis functions would likely have led to smaller misclassification errors, as is evident from Figure 4.

3. To make it possible to fit much larger models and to try out many more sets of features the computational tricks that we used (Sections 9, 13.4, 13.5) are insufficient. Conceivably to fit much larger POLYCLASS models, one should use faster methods such as those used in fitting neural networks.

12 Concluding Remarks

Most statistical techniques that are concerned with finding a relation between a qualitative variable and a number of quantitative predictors are primarily concerned with the multiple classification problem. In this paper we introduce a methodology not just for classification, but also for accurate estimation of the conditional class probabilities. Since the optimal classification rule is based upon the conditional class probabilities, accurate estimation of these probabilities should lead to better classification. Thus in many applications our approach to the multiple classification problem will yield more useful results.

Our methodology combines maximum likelihood estimation, polynomial splines and a stepwise addition and deletion algorithm. Model selection can be carried out using cross-validation, an independent test set or BIC. Extensions of the POLYCLASS model were discussed that can reduce the number of parameters by forcing parameters for certain basis functions to be identical for several classes. POLYCLASS with these extensions works well for problems with many classes but otherwise a simple structure such as the digits example. We also provide a least squares approximation to the (maximum likelihood based) model selection, which makes the method computationally feasible for very large problems.

In Section 10 we observe that POLYCLASS is fairly competitive with other methodologies when used as a multiple classification method and that it gives accurate estimates of the conditional class probabilities. At the same time, most of the methods that we used for comparison on the smaller examples are not usable on very large data sets such as the phoneme recognition data. The ones that we could use were outperformed by POLYCLASS. Moreover, our results suggest that, with the modifications discussed at the end of Section 11, POLYCLASS would be competitive with neural networks.

13 Numerical Implementation

13.1 Quadratic Approximations to the Likelihood

Let Q be a quadratic polynomial on $\mathbb{R}^q$ having a negative definite Hessian matrix $\mathbf{H}$, and set $\mathbf{I} = -\mathbf{H}$. Also, let $\widehat{\boldsymbol{\beta}}$ maximize Q on $\mathbb{R}^q$ and let $\widehat{\boldsymbol{\beta}}_0 \in \mathbb{R}^q$. Then

$$0 = \nabla Q(\widehat{\boldsymbol{\beta}}) = \nabla Q(\widehat{\boldsymbol{\beta}}_0) + \mathbf{H}(\widehat{\boldsymbol{\beta}} - \widehat{\boldsymbol{\beta}}_0),$$

so $\widehat{\boldsymbol{\beta}} - \widehat{\boldsymbol{\beta}}_0 = \mathbf{I}^{-1}\nabla Q(\widehat{\boldsymbol{\beta}}_0)$, hence

$$\begin{aligned} Q(\widehat{\boldsymbol{\beta}}) &= Q(\widehat{\boldsymbol{\beta}}_0) + (\widehat{\boldsymbol{\beta}} - \widehat{\boldsymbol{\beta}}_0)^T \nabla Q(\widehat{\boldsymbol{\beta}}_0) + \tfrac{1}{2}(\widehat{\boldsymbol{\beta}} - \widehat{\boldsymbol{\beta}}_0)^T \mathbf{H}(\widehat{\boldsymbol{\beta}} - \widehat{\boldsymbol{\beta}}_0) \\ &= Q(\widehat{\boldsymbol{\beta}}_0) + \tfrac{1}{2}[\nabla Q(\widehat{\boldsymbol{\beta}}_0)]^T \mathbf{I}^{-1} \nabla Q(\widehat{\boldsymbol{\beta}}_0), \end{aligned}$$

and therefore

$$2[Q(\widehat{\boldsymbol{\beta}}) - Q(\widehat{\boldsymbol{\beta}}_0)] = [\nabla Q(\widehat{\boldsymbol{\beta}}_0)]^T \mathbf{I}^{-1} \nabla Q(\widehat{\boldsymbol{\beta}}_0). \tag{14}$$

Suppose now that $\widehat{\boldsymbol{\beta}}_0$ maximizes $Q(\boldsymbol{\beta})$ subject to the constraint that $\mathbf{A}\boldsymbol{\beta} = 0$, where $\mathbf{A}$ is a $p \times q$ matrix having rank p. Then $\mathbf{A}\widehat{\boldsymbol{\beta}}_0 = 0$. By the Lagrange multiplier theorem there is a $\boldsymbol{\lambda} \in \mathbb{R}^p$ such that $\nabla Q(\widehat{\boldsymbol{\beta}}_0) = \mathbf{A}^T\boldsymbol{\lambda}$. It follows from (14) that

$$2[Q(\widehat{\boldsymbol{\beta}}) - Q(\widehat{\boldsymbol{\beta}}_0)] = \boldsymbol{\lambda}^T \mathbf{A}\mathbf{I}^{-1}\mathbf{A}^T\boldsymbol{\lambda}. \tag{15}$$

Moreover, $\widehat{\boldsymbol{\beta}} - \widehat{\boldsymbol{\beta}}_0 = \mathbf{I}^{-1}\mathbf{A}^T\boldsymbol{\lambda}$, so $\mathbf{A}\widehat{\boldsymbol{\beta}} = \mathbf{A}(\widehat{\boldsymbol{\beta}} - \widehat{\boldsymbol{\beta}}_0) = \mathbf{A}\mathbf{I}^{-1}\mathbf{A}^T\boldsymbol{\lambda}$. Thus, by (15),

$$2[Q(\widehat{\boldsymbol{\beta}}) - Q(\widehat{\boldsymbol{\beta}}_0)] = (\mathbf{A}\widehat{\boldsymbol{\beta}})^T(\mathbf{A}\mathbf{I}^{-1}\mathbf{A}^T)^{-1}(\mathbf{A}\widehat{\boldsymbol{\beta}}). \tag{16}$$

Furthermore, $\boldsymbol{\lambda} = (\mathbf{A}\mathbf{I}^{-1}\mathbf{A}^T)^{-1}\mathbf{A}\widehat{\boldsymbol{\beta}}$ and hence

$$\widehat{\boldsymbol{\beta}}_0 = \widehat{\boldsymbol{\beta}} - \mathbf{I}^{-1}\mathbf{A}^T(\mathbf{A}\mathbf{I}^{-1}\mathbf{A}^T)^{-1}\mathbf{A}\widehat{\boldsymbol{\beta}}. \tag{17}$$

If Q is the quadratic approximation to the log-likelihood function at $\widehat{\boldsymbol{\beta}}_0$, then the right side of (14) is the Rao statistic. If Q is the quadratic approximation to the log-likelihood function at $\widehat{\boldsymbol{\beta}}$, then the right side of (16) is the Wald statistic. Also, (17) yields a convenient starting value for the Newton–Raphson method in the context of stepwise deletion.

13.2 Maximum Number of Basis Functions

We stop the addition of basis functions when one of the following three conditions is satisfied:

- the number P of basis functions equals $P_{\max}$, where the default value for $P_{\max}$ equals $\min(4n^{1/3}, n/(2K), 50)$;
- $\hat{l}_P - \hat{l}_p < \frac{1}{2}(P-p) - 0.5$ for some p with $3 \le p \le P-3$, where $\hat{l}_p$ is the log-likelihood for the model with p parameters (since this suggests that addition of more basis functions is not likely to improve the fit);
- the search algorithm yields no possible new basis function.

13.3 Optimizing the Location of a New Knot

In this section we describe the algorithm for finding the location of a potential new knot for the POLYCLASS model when the model selection is not carried out using the least squares approximation discussed in Section 9. Issues that come up in the least squares approximation are discussed in Sections 13.4 and 13.5.

In a polychotomous regression model it is not possible to come up with simple updating formulas to compute score functions and Hessians for many new potential knots as in MARS, which uses least squares estimation (Friedman, 1991). Thus we need to restrict the computation of the Rao statistic to a limited number of knots.

To find a potential new knot in the mth predictor X_m, let $t_1 < t_2 < \cdots < t_{K_m}$ be the corresponding knots presently in the model (to which we want to add one more knot) and let

$X_{(1)}, \ldots, X_{(n)}$ be the values $X_{1m}, \ldots, X_{nm}$ of X_m written in nondecreasing order. Define l_i and u_i by

$$
\begin{aligned}
l_i &= 6 + \arg\max_{1 \le j \le n} X_{(j)} \le t_i, \qquad i = 1, \ldots, K_m, &(18)\\
u_i &= -6 + \arg\min_{1 \le j \le n} X_{(j)} \ge t_{i+1}, \qquad i = 0, \ldots, K_m - 1, &(19)\\
l_0 &= 1 \qquad \text{and}\\
u_k &= n.
\end{aligned}
$$

For $i = 0, \ldots, K_m$ such that $u_i \ge l_i$ we compute the Rao statistic r_i for the model with $(x_m - X_{(j_i)})_+$ as the new basis function, where $j_i = [(l_i + u_i)/2]$. Because of the 6 and -6 in (18) and (19) it is possible that $u_i < l_i$ for some i; if so, then no knot can be added between t_i and t_{i+1}. This forces knots for a given predictor in the model to be at least 6 order statistics apart, which improves the numerical and statistical stability. If there is no i for which $u_i \ge l_i$, then no knots in X_m can be added to the model.

If a knot can be added to the model we place the potential new knot in the interval $[X_{(l_{i^*})}, X_{(u_{i^*})}]$, where $i^* = \arg\max |r_i|$. We proceed by computing the Rao statistic r_l for the model with $(x_m - X_{(l)})_+$ as the new basis function with $l = [(l_{i^*} + j_{i^*})/2]$, and r_u for the model with $(x_m - X_{(u)})_+$ as the new basis function with $u = [(j_{i^*} + u_{i^*})/2]$. If $|r_{i^*}| \ge |r_l|$ and $|r_{i^*}| \ge |r_u|$, we place the new knot at $X_{(j_{i^*})}$; if $|r_{i^*}| < |r_l|$ and $|r_l| \ge |r_u|$, we continue searching for a knot location in the interval $[X_{(l_{i^*})}, X_{(j_{i^*})}]$; and if $|r_{i^*}| < |r_u|$ and $|r_l| < |r_u|$, we continue searching for a knot location on the interval $[X_{(j_{i^*})}, X_{(u_{i^*})}]$.

Note that for each candidate for a new basis function only one column of $\mathbf{H}(\cdot)$ and one element of $\mathbf{S}(\cdot)$ need to be computed, all other elements having already been computed during the most recent set of iterations.

13.4 Issues in the Least Squares Approximation

As part of the least squares approximation to POLYCLASS, we need to solve many equations of the form

$$\widehat{\boldsymbol{\beta}}_k = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}_k, \qquad 1 \le k \le K.$$

Here $\mathbf{X}^T\mathbf{X}$ is a $p \times p$ matrix having an already inverted $(p-1) \times (p-1)$ submatrix. Thus inverting $\mathbf{X}^T\mathbf{X}$ now requires only $O(p^2)$ flops. If we assume that all necessary inner product among predictors and between predictors and responses are known, computing $\widehat{\boldsymbol{\beta}}_k$ for all k requires $O(p^2K)$ flops.

In the context of deciding which basis function to enter next, we need to compute numerous quantities of the form

$$Q_k(\boldsymbol{\beta}_k) = -||\mathbf{Y}_k - \mathbf{X}\boldsymbol{\beta}_k||^2.$$

To compute the corresponding Rao statistics, we need to compute

$$[\nabla Q_k(\widehat{\boldsymbol{\beta}}_{k0})]^T \mathbf{I}^{-1} \nabla Q_k(\widehat{\boldsymbol{\beta}}_{k0}).$$

Here $\mathbf{I} = \mathbf{X}^T\mathbf{X}$ and $\nabla Q_k(\boldsymbol{\beta}_k) = 2(\mathbf{X}^T\mathbf{Y}_k - \mathbf{X}^T\mathbf{X}\boldsymbol{\beta}_k) = 2\mathbf{X}^T(\mathbf{Y}_k - \mathbf{X}\boldsymbol{\beta}_k)$. Only one entry of $\nabla Q_k(\widehat{\boldsymbol{\beta}}_{k0})$ is nonzero, corresponding to the candidate basis function. Since $\mathbf{X}\widehat{\boldsymbol{\beta}}_{k0}$ does not depend on the new basis function under consideration, it can be assumed known. Thus to compute $\nabla Q_k(\widehat{\boldsymbol{\beta}}_{k0})$ we need to compute the component of $\mathbf{X}^T(\mathbf{Y}_k - \mathbf{X}\boldsymbol{\beta}_k)$ corresponding to the candidate basis function.

We also need to compute the "lower-right" entry of $\mathbf{I}^{-1}$, having already computed the inverse of the $(p-1) \times (p-1)$ submatrix corresponding to the existing basis functions. For each k this requires $O(p^2)$ flops once the p entries (inner products) corresponding to the new basis functions are determined. Thus the number of flops required for each candidate basis function is $O(p^2K)$.

The inner products that we need to know for the computations outlined above are:

1. those between basis functions in the model and responses;
2. those among basis functions in the model;
3. those between candidate basis functions and basis functions in the model and responses.

If $P_{\max}$ is the largest number of basis functions that we consider, there are $KP_{\max}$ inner products corresponding to (1) and $\frac{1}{2}P_{\max}^2$ corresponding to (2). If we fix the number of candidate knots in each variable to, say, N_0, the number of candidate basis functions (knots and interactions) remains limited, since typically only a few new interactions are candidates after an addition. In our experience, the total number of candidates is approximately $N_0 P_{\max}$. Thus approximately $N_0 P_{\max} \times (P_{\max} + K)$ inner products need to be computed corresponding to (3). Note that each inner product requires n operations.

In the phoneme recognition problem the computation of the inner products involving candidate basis functions is dominant. When $n = 112115$, $K = 45$, $P_{\max} = 350$ and $N_0 = 50$ this yields $O(10^{12})$ flops, which took about one day cpu time on our SGI workstation.

It should be noted here that our dedicated implementation of MARS (POLYCLASS-MARS) is now much faster than the standard version (Friedman, 1991). In particular, we generated a subset of the phoneme data with 10000 cases, 2 classes and 63 predictors, and applied both POLYCLASS-MARS and Friedman's program. When the maximum number of basis functions was set equal to 40 in both programs, our program took 177 seconds of cpu time, while Friedman's program took 2196 seconds on the same machine. With 80 basis functions the corresponding cpu times were 474 seconds and 12636 seconds. We save considerable cpu time by storing old inner products, which MARS does not and, hence, needs to recompute. Note that the standard version of MARS takes $O(MNP_{\max}^3)$ flops (Friedman, 1991, p. 127), while POLYCLASS-MARS (in the case that $K = 2$) takes $O(N_0 N P_{\max}^2)$ flops. Since N_0 (about 50) and M (63) are comparable in size, the computations are reduced by about a factor of $P_{\max}$. Our illustrative cpu results seem to confirm this order-of-magnitude comparison.

There are some other differences between POLYCLASS-MARS and traditional MARS.

- The addition schemes are different: we add first a linear term and perhaps a knot later, while in MARS two basis functions, essentially corresponding to a linear and a knot, are added at the same time.
- In traditional MARS a piecewise cubic approximation to the piecewise linear function is applied after a basis function is added, which is not the case in POLYCLASS-MARS.

13.5 Speeding up POLYCLASS after MARS Approximations

Fitting the largest POLYCLASS model with basis functions provided by MARS (see Section 9) is now the major problem. This model has $P_{\max}(K-1)$ parameters. In the phoneme recognition this amounts to approximately 15400 such parameters. Although the least squares approximation, discussed in Section 11, does provide us useful basis functions, it does not give us usable starting values.

Our current approach to fit the largest POLYCLASS model is to introduce the basis functions one at a time. The estimates for the previous model with $p-1$ basis functions can then be used as starting values for the current model with p basis functions. However, when we fit this model with $p(K-1)$ parameters, we use only $5p(K-1)$ cases to fit the model. We use quasi-Newton updates for the Hessian matrix, and we stop iterating at the current model when the difference between two consecutive log-likelihoods is less than 10, which yields a very rough convergence criterion. The inverse of the quasi-Hessian that we obtained from fitting the previous model can be used as the initial value for the upper-left $(p-1)(K-1) \times (p-1)(K-1)$ submatrix of the quasi-Hessian for fitting the current model.

Upon completion of the sequential addition process, we fit the largest model with increased precision. Then we fit this model using all data and our usual convergence criterion. This method of gradually increasing the number of cases provides us with good starting values as well as a decent initial guess for the quasi-Hessian, while the computational cost is relatively modest.

In fitting the sequence of models, the most time consuming parts are the computations of the score statistic and the log-likelihood. Both of these require $O(pKn)$ flops. (Thus, for all models from $p = 1$ to $p = P_{\max}$ basis functions the computations require $O(P_{\max}^2 Kn)$ flops.) Typically we may need 200 such computations for a model with $p < P_{\max}$ basis functions for a large problem like the phoneme recognition data, while we need approximately 1000 of them for the model with $p = P_{\max}$. The computations require $O(10^{14})$ flops for the phoneme recognition data, which would take 80 days of cpu time on our SGI workstation, which is a major improvement compared to the several years for POLYCLASS without the least squares approximation. (If one were to apply the modified POLYCLASS algorithm to a ten-class problem with 30,000 cases and $P_{\max} = 200$, which is still a very large problem by the usual statistical standards, the required cpu time would be about one day.)

However, 80 days is still not realistic. Instead, we carried the computations out on 64 workstations from a network of 400 RS6000 workstations with a high-speed communications network at the Maui High Performance Computing Center. We parallelized our computations by sending $\frac{1}{63}$ of the data and $\frac{1}{63}$ of the columns of the quasi-Hessian to each of 63 workstations, while the 64th 'master' workstation coordinated the computations. On this network the computations took 24 hours: 8 hours on the 'master' and 16 simultaneous hours on each of the 63 'slaves'. As pointed out at the end of Section 11, when $p(K-1)$ is huge, there may be considerable potential for further improvement in speeding up POLYCLASS by replacing the quasi-Newton algorithm by a one-case-at-a-time gradient method.

References

Bose, S. (1992), "A Method for Estimating Nonlinear Class Boundaries in the Classification Problem and Comparison with Other Existing Methods," Ph. D. dissertation, Department of Statistics, University of California at Berkeley.

Bose, S. (1994), "Classification Using Splines," Technical Report No. 541, Department Of Statistics, The Ohio State University.

Bourlard, H. A. and Morgan, N. (1994), *Connectionist Speech Recognition*, Boston: Kluwer.

Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. (1984). *Classification and Regression Trees*, Belmont, California: Wadsworth.

Breiman, L., and Ihaka, R. (1984), "Nonlinear Discriminant Analysis via Scaling and Ace," technical report, University of California, Berkeley, Department of Statistics.

Cole, R., Noel, M, Burnett, D. C. Fanty, M., Lander, T. Oshika, B. and Sutton, S. (1994), "Corpus Development Activities at the Center for Spoken Language Understanding," Technical Report, CSLU, Portland, Oregon.

Cole, R. A., Roginski, K and Fanty, M. (1992), "A Telephone Speech Database of Spelled and Spoken Names," *Proceedings of the International Conference on Spoken Language Processing, Banff, Alberta, Canada*, 891–893.

Fletcher, R. (1987), *Practical Methods of Optimization, 2nd ed.*, New York: Wiley.

Friedman, J. H. (1991), "Multivariate Adaptive Regression Splines (with discussion)," *The Annals of Statistics*, 19, 1-141.

Gauvain, J. L., Lamel, L., Adda, G. and Adda-Decker, M. (1994), "Speaker-Independent Continuous Speech Dictation", *Speech Communication*, **15**, to appear.

Hansen, M. (1994). *Extended Linear Models, Multivariate Splines, and ANOVA.* Ph. D. Dissertation, University of California at Berkeley.

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

Hastie, T., Tibshirani, R. and Buja, A. (1994), Flexible Discriminant Analysis by Optimal Scoring. *Journal of the American Statistical Association*, 89, 1255–1270.

Hermansky, H. (1990), "Perceptual Linear Predictive (PLP) Analysis of Speech," *Journal of the Acoustical Society of America*, **87**, 1738–1752.

Hosmer, D. and Lemeshow, S. (1989), *Applied Logistic Regression*, New York: Wiley.

Kooperberg, C. and Stone, C. J. (1992), "Logspline Density Estimation for Censored Data," *Journal of Computational and Graphical Statistics*, 1, 301–328.

Kooperberg, C., Stone, C. J., and Truong, Y. K. (1995), "Hazard Regression," *Journal of the American Statistical Association*, 90, 78–94.

Mardia, K. V., Kent, J. T. and Bibby, J. M. (1979), *Multivariate Analysis*, London: Academic Press.

Rabiner, L. and Juang, B.-H. (1993). *Fundamentals of Speech Recognition*, Englewood Cliffs, NJ: Prentice Hall.

Ripley, B. (1994). "Neural Networks and Related Methods for Classification," *Journal of the Royal Statistical Society Series B*, 56, 409–456.

Rao, C. R. (1973), *Linear Statistical Inference and Its Applications*, second edition, New York: Wiley.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *The Annals of Statistics*, 6, 461–464.

Stone, C. J. (1994), "The Use of Polynomial Splines and Their Tensor Products in Multivariate Dunction Estimation (with discussion). *The Annals of Statistics*, 22, 118-184.