Polychotomous Regression

Charles Kooperberg, Smarajit Bose and Charles J. Stone [1]

University of Washington
Department of Statistics
Technical Report No. 288
Revised, May 6 1996

An automatic procedure that uses linear splines and their tensor products is proposed for fitting a regression model to data involving a polychotomous response variable and one or more predictors. The fitted model can be used for multiple classification. The automatic fitting procedure involves maximum likelihood estimation, stepwise addition, stepwise deletion, and model selection by AIC, cross-validation or an independent test set. A modified version of the algorithm has been constructed that is applicable to large data sets, and it is illustrated using a phoneme recognition data set with 250,000 cases, 45 classes and 63 predictors.
KEY WORDS: Linear splines; Logistic regression; MARS; Model selection; Multiple classification; Speech recognition; Tensor products.

[1] Charles Kooperberg is Assistant Professor, Department of Statistics, University of Washington, Seattle, WA 98195-4322. Smarajit Bose is Lecturer, Indian Statistical Institute, Calcutta 700035, India. This work was done while Bose was Visiting Assistant Professor of Statistics at the Ohio State University. Charles J. Stone is Professor, Department of Statistics, University of California, Berkeley, CA 94720-3860. Charles Kooperberg was supported in part by National Science Foundation grant DMS-9403371 and National Institutes of Health grant CA61937. Charles J. Stone was supported in part by National Science Foundation grants DMS-9204247 and DMS-9504463. 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. Comments by an associate editor and two referees greatly improved this paper.

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. We want to predict Y based on a random vector $\mathbf{X} \in I\!R^M$. 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 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 (1996) 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 and Cheng and Titterington 1994 for overviews).

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 4.2 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 no 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 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 POLYCLASS classification methodology that provides reliable estimates for conditional class probabilities.

This paper is organized as follows. In Section 2 we set up the polychotomous regression model, describe its relation to multiple classification, and discuss the estimation procedure. In Section 3 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 particular, in Section 3.3 we discuss a least squares approximation, POLYMARS, to the model selection procedure that can dramatically speed up the computations. POLYMARS is a customized multiresponse version of MARS (Friedman 1991) designed to be able to deal with huge datasets. In Section 4 we apply POLYCLASS to a small example involving simulated data and to an example from the area of speech recognition involving a data set of 2000 utterances (short sentences) that yielded almost 250,000 cases. (Each case represents 12.5ms of speech.) 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 possible phoneme (not to classify the current phoneme) based on 63 predictors, which are obtained from the audible spectrum of the sound. In Section 5 we give a few concluding remarks. A number of technical details about the methodology are deferred to the appendices.

Versions of the POLYCLASS and POLYMARS programs, written in C and interfaced to S/S-PLUS, will be made available via `statlib` in the near future.

2. POLYCHOTOMOUS REGRESSION MODELS

2.1 Polychotomous Regression and ANOVA Decompositions

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 the 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 linear, additive model $\theta(k|\mathbf{x}) = \beta_{k0} + \beta_{k1}x_1 + \cdots + \beta_{kM}x_M$, $1 \le k < K$. In practice, however, it may be 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

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

In order to allow for interactions between predictors, the generalized additive model can be further refined.

To illustrate our approach, suppose $\mathbf{x} = (x_1, x_2, x_3)$ and consider the form

$$\theta(k|\mathbf{x}) = \theta_{0k} + \theta_{1k}(x_1) + \theta_{2k}(x_2) + \theta_{3k}(x_3) + \theta_{12k}(x_1, x_2) + \theta_{13k}(x_1, x_3) + \theta_{23k}(x_2, x_3), \qquad 1 \le k < K, \tag{3}$$

where $\theta_{1k}(\cdot), \ldots, \theta_{23k}(\cdot)$ are smooth functions. Here θ_{0k} is the constant term, $\theta_{1k}(\cdot)$, $\theta_{2k}(\cdot)$ 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 two-factor interactions. Given a random sample, consider an estimate

$$\widehat{\theta}(k|\mathbf{x}) = \widehat{\theta}_{0k} + \widehat{\theta}_{1k}(x_1) + \widehat{\theta}_{2k}(x_2) + \widehat{\theta}_{3k}(x_3) + \widehat{\theta}_{12k}(x_1, x_2) + \widehat{\theta}_{13k}(x_1, x_3) + \widehat{\theta}_{23k}(x_2, x_3), \qquad 1 \le k < K. \tag{4}$$

We can think of $\widehat{\theta}(k|\mathbf{x})$ as an estimate of $\theta(k|\mathbf{x})$. Alternatively, if $\theta(k|\mathbf{x})$ does not necessarily have the form specified in (3), we can think of $\widehat{\theta}(k|\mathbf{x})$ as an estimate of the best theoretical approximation

$$\theta^*(k|\mathbf{x}) = \theta^*_{0k} + \theta^*_{1k}(x_1) + \theta^*_{2k}(x_2) + \theta^*_{3k}(x_3) + \theta^*_{12k}(x_1, x_2) + \theta^*_{13k}(x_1, x_3) + \theta^*_{23k}(x_2, x_3), \qquad 1 \le k < K, \tag{5}$$

to $\theta(k|\mathbf{x})$, where best means having the maximum expected log-likelihood subject to the specified form.

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 log-likelihood. Given a random sample of size n from the distribution of $(\mathbf{X}, Y)$, if maximum likelihood and suitable (nonadaptive) sums of polynomial splines and their tensor products are used to construct an estimate $\widehat{\theta}$ of θ^*, where $\widehat{\theta}$ has the same form as θ^*, then this estimate can achieve the L_2 rate of convergence $n^{-p/(2p+d)}$. Here p is a suitably defined smoothness parameter corresponding to θ^*; in particular, $p = 2$ when linear splines and their tensor products are used and the components of θ^* are twice continuously differentiable. Thus, by choosing $d = 1$ as in (2) or $d = 2$ as in (3)–(5) instead of $d = M$, we can ameliorate the curse of dimensionality.

(Taking $d \leq 2$ is similar to the common practice of ignoring interactions involving three or more factors in a factorial design.) For more detailed discussions of theoretical rates of convergence in this and related contexts, see Stone (1994), Hansen (1994), and Stone, Hansen, Kooperberg and Truong (1996).

In this paper we restrict attention to $d \leq 2$ and use linear splines and their tensor products, but we will be choosing these splines in an adaptive way. In practical applications the restriction to $d \leq 2$ rarely worsens the accuracy of the fitted model, but it improves its interpretability and speeds up and simplifies the corresponding computer code. Although our present code is limited to $d \leq 2$, the methodology described in this paper could easily be extended to include interactions involving three or more factors or, equivalently, tensor products of three or more polynomial splines.

2.2 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 $1 \leq k \leq K-1$, $\boldsymbol{\beta}_K = 0$, 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

$$\begin{aligned} P(Y=k|\mathbf{X}=\mathbf{x};\boldsymbol{\beta}) &= \frac{\exp\theta(k|\mathbf{x};\boldsymbol{\beta})}{\exp\theta(1|\mathbf{x};\boldsymbol{\beta}) + \cdots + \exp\theta(K|\mathbf{x};\boldsymbol{\beta})} \\ &= \exp(\theta(k|\mathbf{x};\boldsymbol{\beta}) - c(\mathbf{x};\boldsymbol{\beta})), \qquad \boldsymbol{\beta} \in \mathcal{B},\ \mathbf{x} \in \mathcal{X} \text{ and } k \in \mathcal{K}, \end{aligned} \tag{7}$$

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

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

The first-order and second-order partial derivatives of $\log P(Y=k|\mathbf{X}=\mathbf{x};\cdot)$ are easily obtained; in particular, the Hessian matrix is negative semi-definite on $\mathcal{B}$ for $\mathbf{x} \in \mathcal{X}$ and $k \in \mathcal{K}$.

When using (7) to model the conditional class probabilities, we need to resolve two issues: how to choose the linear space G; and, given G, how to estimate $\boldsymbol{\beta}$. The latter issue is dealt with below, while a discussion of the first issue is postponed to Section 3.2. Here it suffices to note that the basis functions B_j will all be piecewise linear functions in one variable or tensor products of two piecewise linear functions in different variables.

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 (6) is given by

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

which is a concave function on $\mathcal{B}$. (For numerical reasons, we add a small penalty term to the log-likelihood function. See Appendix C for details.)

The maximum likelihood estimate $\widehat{\boldsymbol{\beta}}$ is given by $\ell(\widehat{\boldsymbol{\beta}}) = \max_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta})$, and the log-likelihood of the fitted model is given by $\hat{\ell} = \ell(\widehat{\boldsymbol{\beta}})$. 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 $k \in \mathcal{K}$.

The maximum likelihood estimate $\widehat{\boldsymbol{\beta}}$ can conveniently be computed by a Newton–Raphson algorithm (with step-halving) or by using a quasi-Newton approximation of the Hessian, such as the

Broyden–Fletcher–Goldfarb–Shanno (BFGS) inverse updating technique (Fletcher 1987). Quasi–Newton are usually faster than Newton–Raphson methods, since they do not require computation of the full Hessian. However, the Rao statistics (see Appendix A) based on this approximation of the Hessian turn out to be too inaccurate. In practice we therefore alternate between quasi-Newton and Newton–Raphson steps during the computations, especially when $(K-1)p$ is large.

The Bayes multiple classification rule with unit costs is to assign a case with $\mathbf{X} = \mathbf{x}$ to a class k having the maximum conditional probability $P(Y = k|\mathbf{X} = \mathbf{x})$ or, equivalently, having the maximum value of $\theta(k|\mathbf{x})$. The corresponding POLYCLASS rule is to assign the case to a class having the maximum value of $\hat{\theta}(k|\mathbf{x})$.

3. MODEL SELECTION

3.1 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 $p-1$;
- if $G_0 \in \mathcal{G}$ has dimension p, there is at least one space $G \in \mathcal{G}$ with dimension $p+1$ containing G_0 as a subspace.

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 (score) statistic. See Appendix A for details.

Upon stopping the stepwise addition stage with $p = P_{\max}$ basis functions according to a rule described in Appendix C, 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 (see Appendix A) for a basis function that is in G but not in G_0 is smallest in magnitude.

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 4. 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 $1 \leq m \leq M$ and $0 \leq k \leq K_m$, and perhaps certain tensor products $B_{lj}(x_l)B_{mk}(x_m)$ (with $l \neq m$) of two such basis functions. It is required that if the indicated tensor product be among the basis functions for some $j \geq 1$, then $B_{l0}(x_l)B_{mk}(x_m) = x_l B_{mk}(x_m)$ and hence $x_l x_m$ (if $k > 0$) be among the basis functions.

One reason for adding linear terms before knots and main effects before interactions is to yield models that are simpler and easier to interpret. In particular, if a covariate appears only linearly in the final model, then the model is a traditional parametric model with respect to that covariate (see the examples in Section 4). A second reason is to reduce the variance associated with the overall modeling procedure, and a third is to reduce the likelihood of ending up with spurious terms in the final model. The requirement of adding main effects before interactions is also motivated by theoretical considerations regarding convergence rates (see Section 2).

It is easy to check that the collection $\mathcal{G}$ of such spaces satisfies the properties listed above. 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 easy to check 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:

- **(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 a heuristic algorithm (see Appendix C).

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.

3.2 Selecting the "Best" Model

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 $\mathrm{AIC}_{\alpha,\nu} = -2\hat{l}_\nu + \alpha(K-1)p_\nu$ 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.) Our software 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 estimates $\widehat{\theta}(Y = k|\mathbf{x})$ we can estimate the risk (probability of misclassification) by $\hat{R}_\nu^{\mathrm{TS}} = \sum_i \mathrm{ind}(\widehat{Y}_i^{\mathrm{TS}} \neq Y_i^{\mathrm{TS}})/n^{\mathrm{TS}}$. Given a finite number of estimates of the optimal classifier, we choose 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. Here we first randomly divide the cases into $c \ge 2$ approximately equally-sized subsets. Then the following procedure is carried out for $j = 1, \dots, c$ (see Breiman et al. 1984):

- Fit a sequence of POLYCLASS models, as described in Section 3.1, to all cases not in the jth subset.
- For each $\alpha > 0$ select the model $\hat{\nu}_{j\alpha}$ that minimizes $\mathrm{AIC}_{\alpha,\nu}$.
- For each α compute the loss $r_j(\alpha) = \sum \mathrm{ind}(\widehat{Y}_i \neq Y_i)$, 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 $R(\alpha) = n^{-1} \sum_{j=1}^{c} r_j(\alpha)$. 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.

3.3 POLYMARS: A Least Squares Approximation of the Addition Process

The stepwise addition process, as described in the previous subsections, is computationally too expensive for huge data sets. We determined that for the phoneme recognition problem discussed in Section 4.2, for which $n = 112115$, $K = 45$, $M = 63$ and $P_{\max} = 350$, the computations would require $O(10^{15})$ floating point operations (flops), which would take several years of cpu time on the SGI workstation that we used for most of our computations. (See Appendix B for details.) This computation led us to consider the following least squares approximation to the stepwise addition process when dealing with large data sets. Let $\mathbf{Z}_i$, $1 \le i \le n$, be the column vector of length K, whose kth element is $\mathrm{ind}(Y_i = k)$. The estimate $\widehat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}$ is obtained by minimizing

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

where $\theta(k|\mathbf{X}_i;\boldsymbol{\beta}) = \sum_{j=1}^{p} \beta_{jk} B_j(\mathbf{X}_i)$. The selection of the new basis function is carried out by minimizing $V(\widehat{\boldsymbol{\beta}})$, while the same allowable spaces as in POLYCLASS are used (see Section 3.1). The stepwise addition part of the model selection can now be carried out in a few hours for the phoneme recognition problem. See Appendix B for more details. This least squares version of the stepwise addition algorithm, referred to as POLYMARS, is similar to the MARS algorithm in Friedman (1991), but it is substantially faster.

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 2 and a quasi-Newton algorithm. The stepwise deletion procedure remains the same as in Section 3.1, 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. The idea for using POLYMARS as a preprocessor for POLYCLASS was inspired by Bose (1996) and Hastie et al. (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 Appendix B for details.

4. EXAMPLES

We used two data sets to compare the performance of POLYCLASS to a variety of other classification methods: linear discriminant analysis (LDA); flexible discriminant analysis (FDA) (Hastie et al. 1994); classification using splines (CUS) (Bose 1996); and classification and regression trees (CART) (Breiman et al. 1984). The first example involves the artificial waveform data from the CART monograph, and the second example involves real data from the area of speech recognition.

Linear discriminant analysis, a classical method that has been used for decades, 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 et al. (1979) for more details.

When the assumptions underlying LDA are far from being satisfied, the method may perform poorly. This has motivated researchers to come up with various alternative methods. One such method is CART (Breiman et al. 1984), which predicts the class membership of an individual based on a binary decision tree. Each node of the tree splits the ranges of individual predictors to separate the measurements from different classes. The option of splitting the predictor space by linear combinations of predictors is also available in CART.

CUS (Bose 1996) uses an additive cubic spline model to approximate the conditional class probabilities. However, in contrast to POLYCLASS and like the procedure described in Section 3.3, this model is estimated using least squares regression. 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 scaling to represent the classes. Hastie et al. (1994) replaced linear regression by nonparametric regression methods such as MARS or BRUTO (Hastie 1989) and thus developed the FDA classification method. While MARS is based on linear (or cubic) regression splines and their tensor products, BRUTO uses an additive smoothing splines model. FDA follows a two-step approach: the initial estimates are obtained by least squares regression using MARS or BRUTO as described in Section 3.3, and then an optimal scoring step is performed to obtain final estimates. Hastie et al. showed that the second step (essentially a linear discriminant analysis with the initial estimates treated as predictors) can provide lower error rates than those achieved by the initial estimates.

4.1 Waveform Data

In our first example (the detailed description of which can be found in Breiman et al. 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

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

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

Method	Training	Test	Cross-validation
POLYCLASS	.135	.200	.184
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

where u has uniform distribution on (0,1) and the ϵ_i are independent random variables with a standard normal distribution. Note that for fixed u this problem would exactly satisfy the conditions under which LDA is the optimal classification procedure. Since u is random this is no longer the case; however, we may still expect that LDA would work quite well on this example.

A training set of size 300 was generated using equal priors. For the POLYCLASS models, CUS and CART, the model selection was performed using ten-fold cross validation. In LDA no model selection was used, and in FDA the model selection was 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 error rates on the training and test sets 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. (Cross-validation is never used for standard LDA, while the implementation of FDA that was available to us did not allow for cross-validation.)

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 a 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, particularly since the predictors are highly correlated. We note that POLYCLASS performs better than the other nonadditive models.

In Figure 1 we show some plots related to one particular 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 Figure 1a, 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.

4.2 Phoneme Recognition

Our second example is taken from the area of speech recognition. The source of this data set is the Center for Spoken Language Understanding in Portland, Oregon (Cole, Roginsky and Fanty 1992; Cole et al. 1994). The data set involves 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 determined 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 (Gauvain et al. 1994).

The utterances were also processed to produce perceptual linear predictive (PLP) features. Every 12.5 ms the audible spectrum is determined from a concentric 25 ms piece of sound. 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 preemphasis 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 coefficients 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 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 c time points, 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 about 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 $d(i) = e(i) - [e(i-3) + \cdots + e(i+3)]/7$ instead of $e(i)$.

The POLYCLASS methodology described in Sections 2.1–3.2 would be practically impossible to

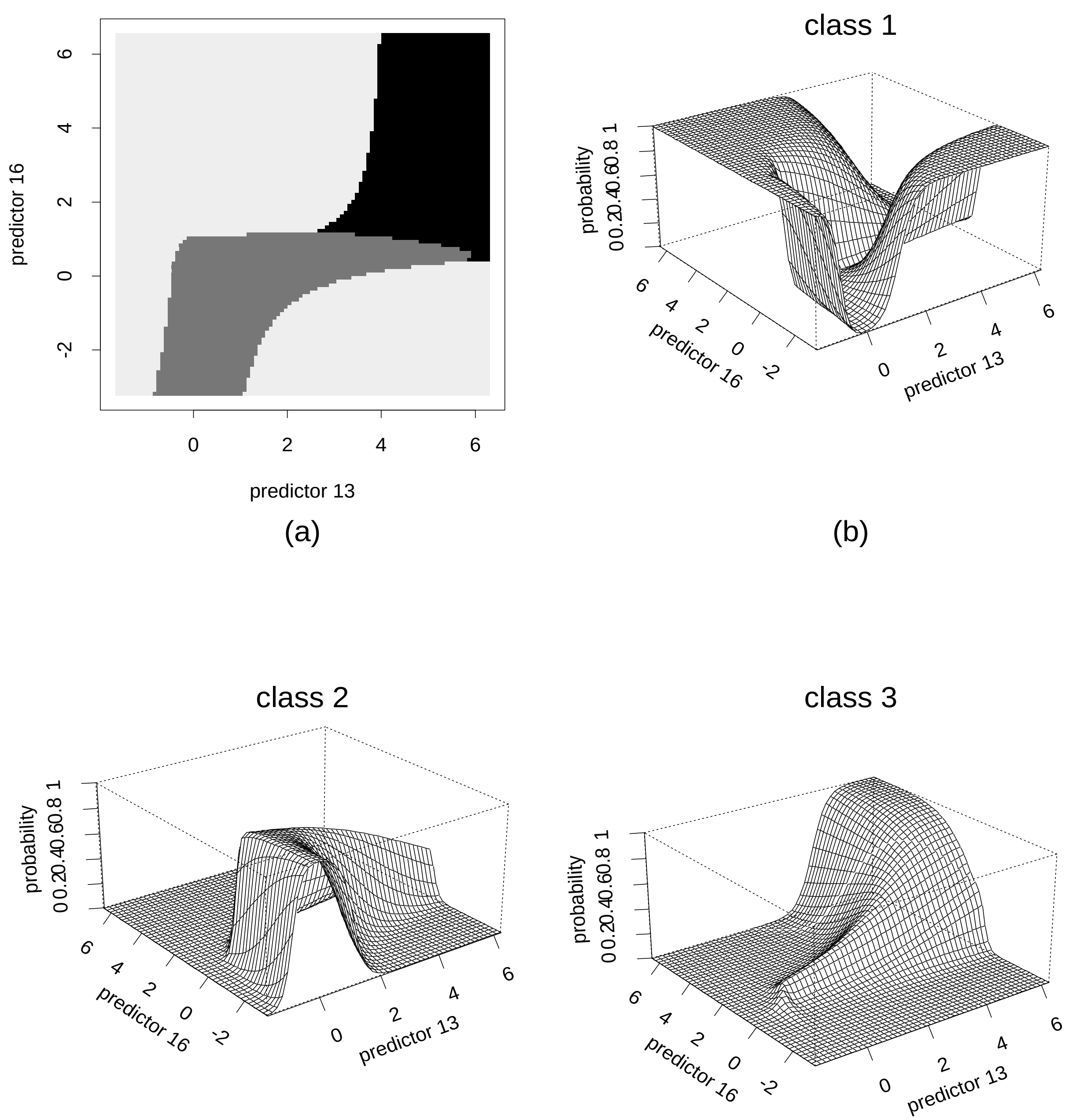

Fig. 1. Classification map and estimated conditional class probabilities as
a function of predictors 13 and 16, when all other predictors are equal to 4.
White: class 1; grey: class 2; black: class 3.

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 and carried out the actual fitting of the model on a network of workstations (see Section 3.3 and Appendix B). The largest model that we fit had 350 basis functions. This number is much larger than the default value of 193 (see Appendix C), but initial analysis suggested that a larger model would yield much better results. (See also the discussion of Figure 2 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.

In Figure 2 we report the misclassification rate and the fitted log-likelihood $\sum_i \log P(Y = Y_i|\mathbf{X} = \mathbf{X}_i)/n$ 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 estimates of the conditional class probabilities that are positive and add up to one. In Figure 3 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 (Figure 3a) and the training set (Figure 3b).

Clearly, not all phonemes are correctly estimated with the same probability. In particular, frequently occurring phonemes are correctly classified more often than infrequently occurring ones. The 22 phonemes that occurred fewer than 1000 times in the test set and the final test set had a total number of 4412 cases of which only 15.4% were correctly classified. The 11 phonemes with between 1000 and 5000 cases in the combined test set had a total number of 35609 cases of which 52.2% were correctly classified. The 12 phonemes with more than 5000 cases in the combined test set had a total number of 94903 cases of which 71.3% were correctly classified.

In Table 2 we summarize misclassification rates for various methods on the phoneme data. We compare POLYCLASS to linear discriminant analysis using the 63 features, POLYMARS (assigning a case to the largest fitted value for the POLYMARS least squares algorithm), and CART, with and without linear combinations. Inspired by Hastie et al (1994), who use a form of discriminant analysis with predictors selected by MARS, we also compare POLYCLASS to linear discriminant analysis using the 349 nonconstant basis functions selected by POLYMARS.

Table 2 shows that POLYCLASS has the best test set error, 4% better than the next best error rate (POLYMARS) and 13% better than LDA on the features. It is interesting to notice that least squares regression on the 349 basis functions (POLYMARS) performs better than LDA on these basis functions. The POLYMARS algorithm that we use to estimate the basis functions gives us the POLYMARS classifier for free, while additional computations have to be carried out for LDA. Conceivably, both for LDA on the basis functions and for POLYMARS the error rate would decrease further if we increased the maximum number of basis functions. (For the other methods the graphs of misclassification rate versus model size look very similar to Figure 2a, except that the misclassification rates are higher.) Since we did not employ stepwise deletion here, the model

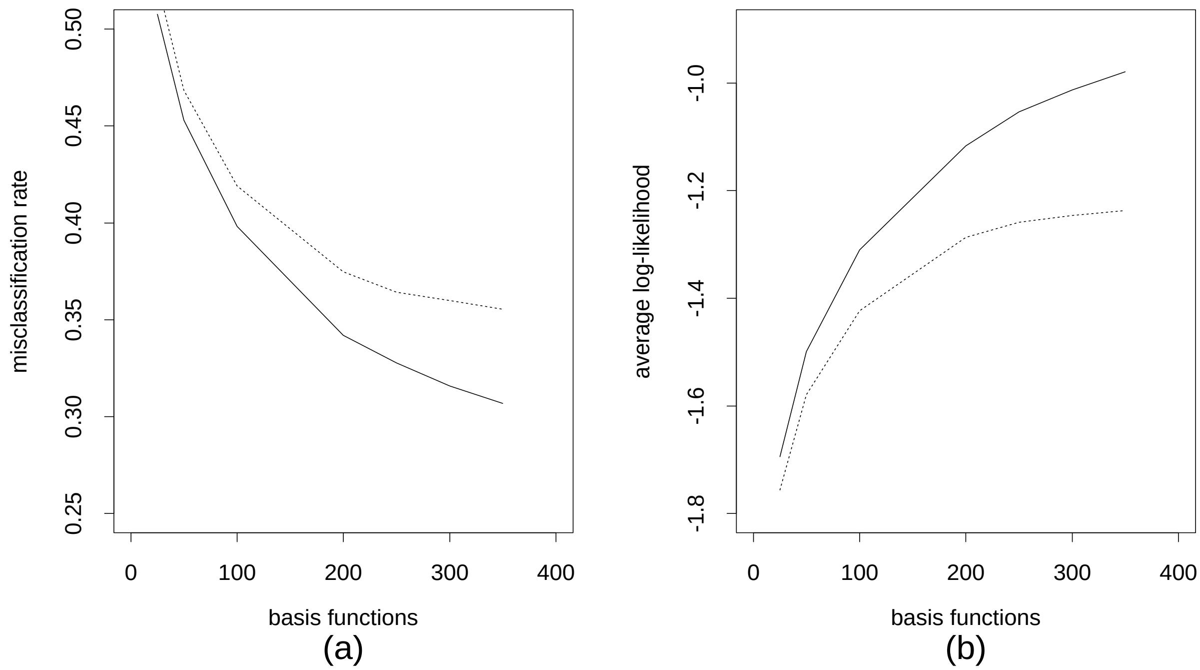

Fig. 2. Misclassification rate (a) and fitted log-likelihood (b) versus the number of basis functions. Solid = training set, dashed = test set combined with final test set.

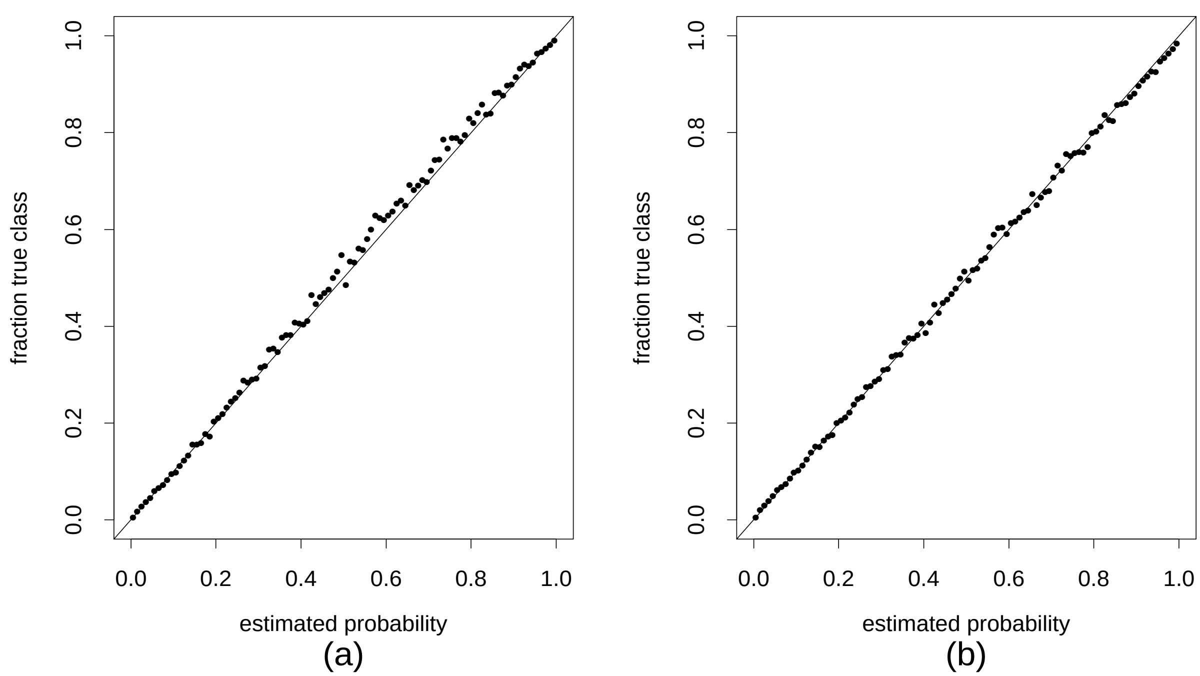

Fig. 3. Fraction of phonemes that correspond to the true class versus the estimated probability. Data has been binned in bins of size 0.01. Fig. 3(a) = training set; Fig. 3(b) = test set combined with final test set.

TABLE 2. Misclassification 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%
POLYMARS - 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%

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.

Misclassification rates for neural networks in exactly this data set were not available from either OGI or ICSI. However, they reported to us informally that, using somewhat different features and/or time periods, they got test set error rates of approximately 25%. 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. Further examination of our fit revealed 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). After our analysis we learned from OGI and ICSI that the actual times at which they use the features are chosen more optimally, based on considerable experience. As confirmation, 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 POLYMARS dropped by approximately 4–5%. To save computing resources we did not apply the other methods to this modified data set.
2. A model with more than 350 basis functions would likely have led to smaller misclassification errors, as is evident from Figure 3.
3. To make it possible to fit much larger POLYCLASS models and try out many more sets of features the computational tricks that we used (Section 3.3, Appendix B) are insufficient.

We currently believe that much faster techniques for fitting huge POLYCLASS models could be developed by using the stochastic gradient methods that are employed in the fitting of neural networks (Boulard and Morgan 1994).

5. CONCLUDING REMARKS

In this paper the polynomial spline methodology that has already been used in density estimation (LOGSPLINE, Kooperberg and Stone 1992), regression (MARS), and hazard regression (HARE) has been extended to handle a categorical response variable with any number of categories (classes) and any number of continuous covariates. The methodology involves maximum likelihood estimation, stepwise addition and stepwise deletion of basis functions, and final model selection using cross-validation, an independent test set, or BIC. The main purpose of the methodology is to provide accurate estimates of conditional class probabilities, which can be used to obtain good estimates of optimal (Bayes) multiple classification rules. As the application to the waveform data in Section 4.1 illustrates, POLYCLASS is competitive with other multiple classification methodologies, including those that do not provide estimates of conditional class probabilities.

In POLYCLASS the number of unknown parameters is the product of the number of basis functions and one less than the number of classes. In the context of the phoneme data discussed in Section 4.2, there are 45 classes and there could easily be 400 or more basis functions, so there could easily be 20,000 unknown coefficients. Also, there are more than 100,000 cases in the training sample. The LOGSPLINE, MARS and HARE algorithms and software were designed to handle up to 50 basis functions and as many unknown coefficients. The standard version of POLYCLASS can easily handle problems substantially larger than the waveform example, but it is unusable on problems having as many cases and, especially, unknown parameters as the phoneme example. Similarly, most of the methods that we used for comparison on the waveform example are not directly usable on problems as large as the phoneme example, and the ones that we could use were outperformed by POLYCLASS.

Perhaps the main contribution of this work has been the development of a modified version of POLYCLASS that is computationally feasible for much larger problems than the standard version. To this end, we developed a linear least squares replacement for the nonlinear maximum likelihood based stepwise addition of basis functions. This least squares stepwise addition procedure in turn was carried out using POLYMARS, a modification of MARS that we developed which is substantially faster when there are many basis functions to be selected. Then, to obtain the nonlinear maximum likelihood fit to the full set of initial basis functions, we employed a quasi-Newton instead of the Newton–Raphson method, we sped up the fitting further by gradually increasing the numbers of basis functions and cases used, and we parallelized the software to enable it to run efficiently on a network of 64 workstations.

In this manner, we obtained a version of POLYCLASS that could handle the phoneme problem. The error rates that we obtained were better than those of the the competing procedures we examined and also better than those reported for neural networks before the start of our project. Since then, however, we have informally learned about still better error rates obtained by experts in the area of speech recognition through the use of neural networks. This should not be surprising in light of the extent of practical experience in improving the computational efficiency in the fitting of neural networks with large numbers of weight parameters and in using such neural networks in the contexts of speech recognition. Moreover, our results suggest that, with the modifications discussed at the end of Section 4.2, POLYCLASS would be competitive with neural networks in this context.

APPENDIX A: QUADRATIC APPROXIMATIONS TO THE LIKELIHOOD

In this appendix we give some motivation for the use of Rao and Wald statistics in the stepwise model selection procedure described in Section 3.

Rao statistics. 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 $\partial^2 \ell_\epsilon(\boldsymbol{\beta})/\partial \beta_{k_1 j_1} \partial \beta_{k_2 j_2}$).

Let $\widehat{\boldsymbol{\beta}}^{(0)}$ be the maximum likelihood estimate of the coefficient vector corresponding to a p-dimensional allowable space G, but subject to the constraint that the estimates of $\theta(k|\mathbf{x})$, $1 \le k \le K-1$, should be in a $(p-1)$-dimensional allowable subspace G_0 of G. Then the

Rao statistic for testing the hypothesis that $\theta(k|\mathbf{x})$ is in G_0 for $1 \le k \le K-1$ 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).)

Wald statistics. Let $\widehat{\boldsymbol\beta}$ be the maximum likelihood estimate of the coefficient vector corresponding to a p-dimensional allowable space G, and let $\hat{\boldsymbol\tau}$ 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 a $(p-1)$-dimensional subspace of 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 $1 \le k \le K-1$ equals $\hat{\boldsymbol\tau}^T\hat{\mathbf{J}}\hat{\boldsymbol\tau}$.

Motivation. Let Q be a quadratic polynomial on $\mathbb{R}^q$ having 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

$$2[Q(\widehat{\boldsymbol\beta}) - Q(\widehat{\boldsymbol\beta}_0)] = [\nabla Q(\widehat{\boldsymbol\beta}_0)]^T\mathbf{I}^{-1}\nabla Q(\widehat{\boldsymbol\beta}_0). \tag{8}$$

Suppose now that $\widehat{\boldsymbol\beta}_0$ maximizes $Q(\boldsymbol\beta)$ subject to the constraint that $\mathbf{A}\boldsymbol\beta = \mathbf{0}$, where $\mathbf{A}$ is an $r\times q$ matrix having rank r. Then $\mathbf{A}\widehat{\boldsymbol\beta}_0 = \mathbf{0}$. By the Lagrange multiplier theorem there is a $\boldsymbol\lambda \in \mathbb{R}^r$ such that $\nabla Q(\widehat{\boldsymbol\beta}_0) = \mathbf{A}^T\boldsymbol\lambda$. It follows from (8) that

$$2[Q(\widehat{\boldsymbol\beta}) - Q(\widehat{\boldsymbol\beta}_0)] = \boldsymbol\lambda^T\mathbf{A}\mathbf{I}^{-1}\mathbf{A}^T\boldsymbol\lambda. \tag{9}$$

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 (9),

$$2[Q(\widehat{\boldsymbol\beta}) - Q(\widehat{\boldsymbol\beta}_0)] = (\mathbf{A}\widehat{\boldsymbol\beta})^T(\mathbf{A}\mathbf{I}^{-1}\mathbf{A}^T)^{-1}(\mathbf{A}\widehat{\boldsymbol\beta}). \tag{10}$$

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

$$\widehat{\boldsymbol\beta}_0 = \widehat{\boldsymbol\beta} - \mathbf{I}^{-1}\mathbf{A}^T(\mathbf{A}\mathbf{I}^{-1}\mathbf{A}^T)^{-1}\mathbf{A}\widehat{\boldsymbol\beta}. \tag{11}$$

If Q is the quadratic approximation to the log-likelihood function at $\widehat{\boldsymbol\beta}_0$, then the right side of (8) is the Rao statistic. If Q is the quadratic approximation to the log-likelihood function at $\widehat{\boldsymbol\beta}$, then the right side of (10) is the Wald statistic. Also, (11) yields a convenient starting value for the Newton–Raphson method in the context of stepwise deletion.

APPENDIX B: LEAST SQUARES APPROXIMATION

B.1 The Stepwise Addition Process

When using the stepwise addition process as described in Section 3.2, quasi-Newton updates for the Hessian matrix do not suffice. Therefore, we need to compute the full Hessian, which requires $O(K^2p^2n)$ 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 for adding a basis function to a model with p basis functions, we typically 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 4.2, $n = 112115$ and $K = 45$, while we used $P_{\max} = 350$. Thus $O(10^{15})$ flops would be required. We estimated that this would take several years on the SGI workstation that we used.

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.

Using the least squares approximation described in Section 3.3, we carry out the stepwise addition part of the model selection in $O(50P_{\max}^2 n)$ flops, or a few hours for the phoneme recognition problem.

As part of the least squares approximation to POLYCLASS, we need to solve many equations of the form $\widehat{\boldsymbol{\beta}}_k = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}_k$ for $1 \leq k \leq K$. Here $\mathbf{X}^T\mathbf{X}$ is a $p \times p$ matrix having a previously inverted $(p-1) \times (p-1)$ submatrix. Inverting $\mathbf{X}^T\mathbf{X}$ now requires only $O(p^2)$ flops. Assuming that all necessary inner products among predictors and between predictors and responses are known, computing all $\widehat{\boldsymbol{\beta}}_k$ 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 evaluate 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 $\nabla Q_k(\boldsymbol{\beta}_k) = 2\mathbf{X}^T(\mathbf{Y}_k - \mathbf{X}\boldsymbol{\beta}_k)$. 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 $O(p^2K)$.

If $P_{\max}$ is the largest number of basis functions that we consider, there are $KP_{\max}$ inner products between basis functions in the model and the responses and $\frac{1}{2}P_{\max}^2$ between basis functions in the model. If we fix the number of candidate knots in each variable at 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 $N_0P_{\max}$. Thus approximately $N_0P_{\max} \times (P_{\max} + K)$ inner products need to be computed between candidate basis functions and basis functions in the model and responses. 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 of cpu time on our SGI workstation.

It should be noted here that our dedicated implementation POLYMARS of 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 POLYMARS 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 must recompute. Note that the standard version of MARS takes $O(MNP_{\max}^3)$ flops (Friedman 1991, p. 127), while POLYMARS (in the case that $K = 2$) takes $O(N_0NP_{\max}^2)$ 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 agree with this order-of-magnitude comparison.

There are other differences between POLYMARS and standard MARS: the stepwise addition schemes are different: we add first a linear term and perhaps later a knot, while in MARS two basis functions, essentially corresponding to a linear function and a knot, are added at the same time; in MARS, but not in POLYMARS, a piecewise cubic approximation to the piecewise linear function is applied after a basis function is added.

B.2 Speeding up POLYCLASS after POLYMARS

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

Our current approach to fitting 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. 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. Upon completion of the sequential addition process, we fit the largest model using all data with increased precision. 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 tolerable.

In fitting the sequence of models, the most time consuming parts are the computations of the score statistic and the log-likelihood, each of which requires $O(pKn)$ 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 60 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.

However, 60 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 1/63 of the data and 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".

APPENDIX C: NUMERICAL ISSUES

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

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

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 $\hat{\beta}_{kj}$ equals $\pm\infty$. This can happen, for example, if $B_j(\mathbf{X}_i) = 0$ for all i such that $Y_i = k$.

The effect of this penalty term is negligible when $|\hat{\beta}_{kj}| < \infty$ for all j and k; that is, in our experience the estimates of the parameters with and without the penalty parameter are extremely close, while the estimates of the conditional class probabilities are indistinguishable. Actually, we choose ϵ as small as possible subject to providing numerically stable estimates.

Maximum number of basis functions. Unless we use the least squares approximation to the stepwise addition procedure, we stop the addition of basis functions when one of the following three conditions is satisfied:

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

Optimizing the location of a new knot. 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 3.3 is identical to the algorithm for finding a new knot in a covariate that was employed in HARE (Kooperberg et al. 1995, sec 11.3).

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.

——— (1996), "Classification Using Splines," *Computational Statistics and Data Analysis*, to appear.

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, Department of Statistics, University of California, Berkeley.

Cheng, B., and Titterington, D. M. (1994), "Neural Networks: A Review from a Statistical Perspective (with discussion)," *Statistical Science*, 9, 2–54.

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, 21–37.

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

Hastie, T. (1989), Discussion of "Flexible Parsimonious Smoothing and Additive Modeling" by Friedman and Silverman, *Technometrics*, 31, 3–39.

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* (2nd ed.), 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 Function Estimation (with discussion)," *The Annals of Statistics*, 22, 118–184.

Stone, C. J., Hansen, M., Kooperberg, C., and Truong, Y. K. (1996), "Polynomial Splines and Their Tensor Products in Extended Linear Modeling," *The Annals of Statistics*, accepted for publication.