SLAC PUB-4960 Rev
Tech Report 102 Rev†
August 1990
M

MULTIVARIATE ADAPTIVE REGRESSION SPLINES*

Jerome H. Friedman,

Stanford Linear Accelerator Center
and
Department of Statistics
Stanford University
Stanford, California 94309

ABSTRACT

A new method is presented for flexible regression modeling of high dimensional data. The model takes the form of an expansion in product spline basis functions, where the number of basis functions as well as the parameters associated with each one (product degree and knot locations) are automatically determined by the data. This procedure is motivated by the recursive partitioning approach to regression and shares its attractive properties. Unlike recursive partitioning, however, this method produces continuous models with continuous derivatives. It has more power and flexibility to model relationships that are nearly additive or involve interactions in at most a few variables. In addition, the model can be represented in a form that separately identifies the additive contributions and those associated with the different multivariable interactions.

(Submitted to *The Annals of Statistics*)

*This work was supported in part by the Department of Energy under contract DE-AC03-76SF00515, and the National Security Agency under contract MDA904-88-H-2029.
† Also published as Dept. of Statistics Tech. Report 102 Rev, August 1990.

1.0. Introduction

A problem common to many disciplines is that of adequately approximating a function of several to many variables, given only the value of the function (often perturbed by noise) at various points in the dependent variable space. Research on this problem occurs in applied mathematics (multivariate function approximation), statistics (nonparametric multiple regression), and in computer science and engineering (statistical learning "neural" networks). The goal is to model the dependence of a response variable y on one or more predictor variables $x_1, \cdots, x_n$ given realizations (data) $\{y_i, x_{1i}, \cdots, x_{ni}\}_1^N$. The system that generated the data is presumed to be described by

$$y = f(x_1, \cdots, x_n) + \epsilon \tag{1}$$

over some domain $(x_1, \cdots, x_n) \in D \subset R^n$ containing the data. The single valued deterministic function f, of its n-dimensional argument, captures the joint predictive relationship of y on $x_1, \cdots, x_n$. The additive stochastic component ϵ, whose expected value is defined to be zero, usually reflects the dependence of y on quantities other than $x_1, \cdots, x_n$ that are neither controlled nor observed. The aim of regression analysis is to use the data to construct a function $\hat{f}(x_1, \cdots, x_n)$ that can serve as a reasonable approximation to $f(x_1, \cdots, x_n)$ over the domain D of interest.

The notion of reasonableness depends on the purpose for which the approximation is to be used. In nearly all applications however accuracy is important. Lack of accuracy is often defined by the integral error

$$I = \int_D w(\mathbf{x}) \Delta[\hat{f}(\mathbf{x}), f(\mathbf{x})] d\mathbf{x} \tag{2}$$

or the expected error

$$E = \frac{1}{N} \sum_{i=1}^{N} w(\mathbf{x}_i) \Delta[\hat{f}(\mathbf{x}_i), f(\mathbf{x}_i)]. \tag{3}$$

Here $\mathbf{x} = (x_1, \cdots, x_n)$, Δ is some measure of distance, and $w(\mathbf{x})$ is a possible weight function. The integral error (2) characterizes the average accuracy of the approximation over the entire domain of interest whereas the expected error (3) reflects average accuracy only on the design points $\mathbf{x}_1, \cdots, \mathbf{x}_N$. In high dimensional settings especially, low integral error is generally much more difficult to achieve than low expected error.

If the sole purpose of the regression analysis is to obtain a rule for predicting future values of the response y, given values for the covariates $(x_1, \cdots, x_n)$, then accuracy is the only important virtue of the model. If future joint covariate values $\mathbf{x}$ can only be realized at the design points $\mathbf{x}_1, \cdots, \mathbf{x}_N$ (with probabilities $w(\mathbf{x}_i)$) then the expected error (3) is the appropriate measure; otherwise the integral error (2) is more relevant. Often, however, one wants to use $\hat{f}$ to try to understand the properties of the true underlying function f (1) and thereby the system that generated the data. In this case the interpretability of the representation of the model is also very important. Depending on the application, other desirable properties of the approximation might include rapid computability and smoothness; that is $\hat{f}$ be a smooth function of its n-dimensional argument and at least its low order derivatives exist everywhere in D.

This paper presents a new method of flexible nonparametric regression modeling that attempts to meet the objectives outlined above. It appears to have the potential to be a substantial improvement over existing methodology in settings involving moderate sample sizes, $50 \leq N \leq 1000$, and moderate to high dimension, $3 \leq n \leq 20$. It can be viewed as either a generalization of the recursive partitioning regression strategy (Morgan and Sonquist, 1963, and Breiman, Friedman, Olshen, and Stone, 1984), or as a generalization of the additive modeling approach of Friedman and Silverman (1989). Its immediate ancestor is discussed in Friedman (1988). Although the procedure described here is somewhat different from that in Friedman (1988), the two procedures have a lot in common and much of the associated discussion of that earlier procedure is directly relevant to the one described here. Some of this common discussion material is therefore repeated in this paper for completeness.

2.0. Existing Methodology

This section provides a brief overview of some existing methodology for multivariate regression modeling. The intent here is to highlight some of the difficulties associated with each of the methods when applied in high dimensional settings in order to motivate the new procedure described later. It should be borne in mind however that many of these methods have met with considerable success in a variety of applications.

2.1. Global Parametric Modeling.

Function approximation in high dimensional settings has (in the past) been pursued mainly in statistics. The principal approach has been to fit (usually a simple) parametric function $g(\mathbf{x} \mid \{a_j\}_1^p)$ to the training data most often by least-squares. That is

$$\hat{f}(\mathbf{x}) = g(\mathbf{x} \mid \{\hat{a}_j\}_1^p) \tag{4}$$

where the parameter estimates are given by

$$\{\hat{a}_j\}_1^p = \underset{\{a_j\}_1^p}{\operatorname{argmin}} \sum_{i=1}^{N} [y_i - g(\mathbf{x} \mid \{a_j\}_1^p)]^2.$$

The most commonly used parametrization is the linear function

$$g(\mathbf{x} \mid \{a_j\}_0^p) = a_0 + \sum_{i=1}^{p} a_i x_i, \quad p \leq n. \tag{5}$$

Sometimes additional terms, that are preselected functions of the original variables (such as polynomials) are also included in the model. This parametric approach has limited flexibility and is likely to produce accurate approximations only when the form of the true underlying function $f(\mathbf{x})$ (1) is close to the prespecified parametric one (4). On the other hand, simple parametric models have the virtue of requiring relatively few data points, they are easy to interpret, and rapidly computable. If the stochastic component ϵ (1) is large compared to $f(\mathbf{x})$, then the systematic error associated with model misspecification may not be the most serious problem.

2.2. Nonparametric Modeling.

In low dimensional settings ($n \lesssim 2$) global parametric modeling has been successfully generalized using three (related) paradigms – piecewise and local parametric fitting and roughness penalty methods. The basic idea of piecewise parametric fitting is to approximate f by several simple parametric functions (usually low order polynomials) each defined over a different subregion of the domain D. The approximation is constrained to be everywhere continuous, and sometimes have continuous low order derivatives as well. The tradeoff between smoothness and flexibility of the approximation $\hat{f}$ is controlled by the number of subregions (knots) and the lowest order derivative allowed to be discontinuous at subregion boundaries. The most popular piecewise polynomial fitting procedures are based on splines, where the parametric functions are taken to be polynomials of degree q and derivatives to order $q-1$ are required to be continuous ($q = 3$ is the most popular choice). The procedure is implemented by constructing a set of (globally defined) basis functions that span the space of qth order spline approximations, and fitting the coefficients of the basis function expansion to the data by ordinary least-squares. For example, in the univariate case ($n = 1$) with $K+1$ regions delineated by K points on the real line ("knots"), one such basis is represented by the functions

$$1,\ \{x^j\}_1^q,\quad \{(x-t_k)_+^q\}_1^K \tag{6}$$

where $\{t_k\}_1^K$ are the knot locations. (Here the subscript "+" indicates a value of zero for negative values of the argument.) This is known as the "truncated power" basis and is one of many that span the space of q-degree spline functions of dimension $K+q+1$. [See deBoor (1978) for a general review of splines and Shumacker (1976), (1984) for reviews of some two-dimensional ($n = 2$) extensions.]

The direct extension of piecewise parametric modeling to higher dimensions ($n > 2$) is straightforward in principle but difficult in practice. These difficulties are related to the so called "curse-of-dimensionality," a phrase coined by Bellman (1961) to express the fact that exponentially increasing numbers of (data) points are needed to densely populate Euclidean spaces of increasing dimension. In the case of spline approximations the subregions are usually constructed as tensor products of $K+1$ intervals (defined by K knots) over the n variables. The corresponding global basis is the tensor product over the $K+q+1$ basis functions associated with each variable (6). This gives rise to $(K+q+1)^n$ coefficients to be estimated from the data. Even with a very coarse grid (small K), a very large data sample is required.

Local parametric approximations ("smoothers") take the form

$$\hat{f}(\mathbf{x}) = g(\mathbf{x} \mid \{\hat{a}_j(\mathbf{x})\}_1^p)$$

where g is a simple parameric function (4). Unlike global parametric approximations, here the parameter values are generally different at each evaluation point $\mathbf{x}$ and are obtained by locally weighted least-squares fitting

$$\{\hat{a}_j(\mathbf{x})\}_1^p = \underset{\{a_j\}_1^p}{\operatorname{argmin}} \sum_{i=1}^{N} w(\mathbf{x}, \mathbf{x}_i)[y_i - g(\mathbf{x}_i \mid \{a_j\}_1^p)]^2. \tag{7}$$

The weight function $w(\mathbf{x}, \mathbf{x}')$ (of $2n$ variables) is chosen to place the dominant mass on points $\mathbf{x}'$ "close" to $\mathbf{x}$. The properties of the approximation are mostly determined by the choice of w and to a lesser extent by the particular parametric function g used. The most commonly studied g is the simple constant $g(\mathbf{x} \mid a) = a$ [Parzen (1962), Shepard (1964), Bonzzini and Lenarduzzi (1985)]. Cleveland (1979) suggested that local linear fitting (5) produces superior results, especially near the edges, and Cleveland and Devlin (1988) suggest local fitting of quadratic functions. Stone (1977) shows that, asymptotically, higher order polynomials can have superior convergence rates when used with simple weight functions (see below), depending on the continuity of properties of f (1).

The difficulty with applying local parametric methods in higher dimensions lies with the choice of an appropriate weight function w (7) for the problem at hand. This strongly depends on f (1) and thus is generally unknown. Asymptotically any weight function that places dominant mass in a (shrinking) convex region centered at $\mathbf{x}$ will work. This motivates the most common choice

$$w(\mathbf{x}, \mathbf{x}') = K(|\mathbf{x} - \mathbf{x}'|/s(\mathbf{x})) \tag{8}$$

with $|\mathbf{x} - \mathbf{x}'|$ being a (possibly) weighted distance between $\mathbf{x}$ and $\mathbf{x}'$, $s(\mathbf{x})$ is a scale factor ("bandwidth"), and K is a ("kernel") function of a single argument.The kernel is usually chosen so that its absolute value decreases with increasing value of its argument. Commonly used scale functions are a constant $s(\mathbf{x}) = s_0$ ("kernel" smoothing) or $s(\mathbf{x}) = s_0/\hat{p}(\mathbf{x})$ ("near neighbor" smoothing), where $\hat{p}(\mathbf{x})$ is some estimate of the local density of the design points. In low dimensional ($n \lesssim 2$) settings, this approximation of the weight function w of $2n$ variables by a function K of a single variable (8), controlled by a single parameter (s_0), is generally not too serious since asymptotic conditions can be realized without requiring gargantuan sample sizes. This is not the case in higher dimensions. The problem with a kernel based on interpoint distance (8) is that the volume of the corresponding sphere in n-space grows as its radius to the nth power. Therefore to ensure that w (8) places adequate mass on enough data points to control the variance of $\hat{f}(\mathbf{x})$, the bandwidth $s(\mathbf{x})$ will necessarily have to be very large, incurring high bias.

Roughness penalty approximations are defined by

$$\hat{f}(\mathbf{x}) = \underset{g}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} [y_i - g(\mathbf{x}_i)]^2 + \lambda R(g) \right\}.$$

Here $R(g)$ is a functional that increases with increasing "roughness" of the function $g(\mathbf{x})$. The minimization is performed over all g for which $R(g)$ is defined. The parameter λ regulates the tradeoff between the roughness of g and its fidelity to the data. The most studied roughness penalty is the integrated squared Laplacian

$$R(g) = \sum_{k=1}^{n} \sum_{\ell=1}^{n} \int \left| \frac{\partial^2 g}{\partial x_k \partial x_\ell} \right|^2 d\mathbf{x} \tag{9}$$

leading to Laplacian smoothing ("thin-plate") spline approximations for $n \leq 3$. For $n > 3$ the general thin-plate spline penalty has a more complex form involving derivatives of higher order than

two. [See Wahba (1990), Section 2.4).] The properties of roughness penalty methods are similar to those of kernel methods (7) (8), using an appropriate kernel function K (8) with λ regulating the bandwidth $s(\mathbf{x})$. They therefore encounter the same basic limitations in high dimensional settings.

2.3. Low Dimensional Expansions.

The ability of the nonparametric methods to often adequately approximate functions of a low dimensional argument, coupled with their corresponding inability in higher dimensions, has motivated approximations that take the form of expansions in low dimensional functions

$$\hat{f}(\mathbf{x}) = \sum_{j=1}^{J} \hat{g}_j(\mathbf{z}_j). \tag{10}$$

Here each $\mathbf{z}_j$ is comprised of a small (different) preselected subset of $\{x_1, \cdots, x_n\}$. Thus, a function of an n-dimensional argument is approximated by J functions, each of a low ($\lesssim 2$) dimensional argument. [Note that any (original) variable may appear in more than one subset $\mathbf{z}_j$. Extra conditions (such as orthogonality) can be imposed to resolve any identifiability problems.] After selecting the variable subsets $\{\mathbf{z}_j\}_1^J$, the corresponding functions estimates $\{\hat{g}_j(\mathbf{z}_j)\}_1^J$ are obtained by nonparametric methods, for example, using least-squares

$$\{\hat{g}_j(\mathbf{z}_j)\}_1^J = \underset{\{g_j\}}{\operatorname{argmin}} \sum_{i=1}^{N} \left[y_i - \sum_{j=1}^{J} g_j(\mathbf{z}_{ij}) \right]^2 \tag{11}$$

with smoothness constraints imposed on the $\hat{g}_j$ through the particular nonparametric method used to estimate them.

In the case of piecewise polynomials (splines) a corresponding basis is constructed for each individual $\mathbf{z}_j$ and the solution is obtained as a global least-squares fit of the response y on the union of all such basis functions [Stone and Koo (1985)]. With roughness penalty methods the formulation becomes

$$\hat{f}(\mathbf{x}) = \underset{\{g_j\}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left[y_i - \sum_{j=1}^{J} g_j(\mathbf{z}_{ij})) \right]^2 + \sum_{j=1}^{J} \lambda_j R(g_j) \right\}. \tag{12}$$

These are referred to as "interaction splines" [Barry (1986), Wahba (1986), Gu, Bates, Chen and Wahba (1990), Gu and Wahba (1988), and Chen, Gu and Wahba (1989)].

Any low dimensional nonparametric function estimator can be used in conjunction with the "backfitting" algorithm to solve (11) [Friedman and Stuetzle (1981), Breiman and Friedman (1985), and Buja, Hastie and Tibshirani (1989)]. The procedure iteratively reestimates $\hat{g}_j(\mathbf{z}_j)$ by

$$\hat{g}_j(\mathbf{z}_j) \leftarrow \underset{g_j}{\operatorname{argmin}} \sum_{i=1}^{N} \left[\left(y_i - \sum_{k \neq j} g_k(\mathbf{z}_{ik}) \right) - g_j(\mathbf{z}_{ij}) \right]^2$$

until convergence. Smoothness is imposed on the $\hat{g}_j$ by the particular estimator employed. For example, if each iterated function estimate is obtained using Laplacian smoothing splines (9) with parameter λ_j (12), then the backfitting algorithm produces the solutions to (12). [See Buja, Hastie and Tibshirani (1989)]. Hastie and Tibshirani (1986) generalize the backfitting algorithm to obtain solutions for criteria other than squared-error loss.

The most extensively studied low dimensional expansion has been the additive model

$$\hat{f}(\mathbf{x}) = \sum_{j=1}^{n} \hat{g}_j(x_j) \tag{13}$$

since nonadaptive smoothers work best for one dimensional functions and there are only n of them (at most) that can enter. Also, in many applications the true underlying function f (1) can be approximated fairly well by an additive function.

Nonparametric function estimation based on low dimensional expansions is an important step forward and (especially for additive modeling) has met with considerable practical success (see references above). As a general method for estimating functions of many variables, this approach has some limitations. The abilities of nonadaptive nonparametric smoothers generally limit the expansion functions to low dimensionality. Performance (and computational) considerations limit the number of low dimensional functions to a small subset of all those that could potentially be entered. For example, there are $n(n+1)/2$ possible univariate and bivariate functions. A good subset will depend on the true underlying function f (1) and is often unknown. Also, each expansion function has a corresponding smoothing parameter causing the entire procedure to be defined by many such parameters. A good set of values for all these parameters is seldom known for any particular application since they depend on f (1). Automatic selection based on minimizing a model selection criterion generally requires a multiparameter numerical optimization which is inherently difficult and computationally consuming. Also the properties of estimates based on the simultaneous estimation of a large number of smoothing parameters are largely unknown, although progress is being made [see Gu and Wahba (1988)].

2.4. Adaptive Computation.

Strategies that attempt to approximate general functions in high dimensionality are based on adaptive computation. An adaptive computation is one that dynamically adjusts its strategy to take into account the behavior of the particular problem to be solved, e.g. the behavior of the function to be approximated. Adaptive algorithms have been in long use in numerical quadrature [see Lyness (1970); Friedman and Wright (1981).] In statistics, adaptive algorithms for function approximation have been developed based on two paradigms, recursive partitioning [Morgan and Sonquist (1963), Breiman, et al. (1984)], and projection pursuit [Friedman and Stuetzle (1981), Friedman, Grosse, and Stuetzle (1983), and Friedman, (1985)].

2.4.1. Projection Pursuit Regression.

Projection pursuit uses an approximation of the form

$$\hat{f}(\mathbf{x}) = \sum_{m=1}^{M} f_m \left(\sum_{i=1}^{n} \alpha_{im} x_i \right), \tag{14}$$

that is, additive functions of linear combinations of the variables. The univariate functions, f_m, are required to be smooth but are otherwise arbitrary. These functions, and the corresponding coefficients of the linear combinations appearing in their arguments, are jointly optimized to produce a good fit to the data based on some distance (between functions) criterion – usually squared-error loss. Projection pursuit regression can be viewed as a low dimensional expansion method where the (one dimensional) arguments are not prespecified, but instead are adjusted to best fit the data. It can be shown [see Diaconis and Shahshahani (1984)] that any smooth function of n variables can be represented by (14) for large enough M. The effectiveness of the approach lies in the fact that even for small to moderate M, many classes of functions can be closely fit by approximations of this form [see Donoho and Johnstone (1989).] Another advantage of projection pursuit approximations is affine equivariance. That is, the solution is invariant under any nonsingular affine transformation (rotation and scaling) of the original explanatory variables. It is the only general method suggested for practical use that seems to possess this property. Projection pursuit solutions have some interpretive value (for small M) in that one can inspect the solution functions f_m and the corresponding loadings in the linear combination vectors. Evaluation of the resulting approximation is computationally fast. Disadvantages of the projection pursuit approach are that there exist some simple functions that require large M for good approximation [see Huber (1985)], it is difficult to separate the additive from the interaction effects associated with the variable dependencies, interpretation is difficult for large M, and the approximation is computationally time consuming to construct.

2.4.2 Recursive Partitioning Regression

The recursive partitioning regression model takes the form

$$\text{if } \mathbf{x} \in R_m, \text{ then } \hat{f}(\mathbf{x}) = g_m(\mathbf{x}|\{a_j\}_1^p). \tag{15}$$

Here $\{R_m\}_1^M$ are disjoint subregions representing a partition of D. The functions g_m are generally taken to be of quite simple parametric form. The most common is a constant function

$$g_m(\mathbf{x}|a_m) = a_m \tag{16}$$

[Morgan and Sunquist (1963) and Breiman, et al. (1984)]. Linear functions (5) have also been proposed [Breiman and Meisel (1976) and Friedman (1979)], but they have not seen much use (see below). The goal is to use the data to simultaneously estimate a good set of subregions and the parameters associated with the separate functions in each subregion. Continuity at subregion boundaries is not enforced.

The partitioning is accomplished through the recursive splitting of previous subregions. The starting region is the entire domain D. At each stage of the partitioning all existing subregions are each optimally split into two (daughter) subregions. The eligible splits of a region R into two daughter regions R_ℓ and R_r take the form

$$\begin{aligned}&\text{if } \mathbf{x} \in R \text{ then}\\ &\quad \text{if } x_v \leq t \text{ then } \mathbf{x} \in R_\ell\\ &\quad\quad \text{else } \mathbf{x} \in R_r\\ &\text{end if.}\end{aligned}$$

Here v labels one of the covariates and t is a value on that variable. The split is jointly optimized over $1 \leq v \leq n$ and $-\infty \leq t \leq \infty$ using a goodness-of-fit criterion on the resulting approximation (15). This procedure generates hyperrectangular axis oriented subregions. The recursive subdivision is continued until a large number of subregions are generated. The subregions are then recombined in a reverse manner until an optimal set is reached, based on a criterion that penalizes both for lack-of-fit and increasing number of regions (see Breiman et al., 1984).

Recursive partitioning is a powerful paradigm, especially if the simple piecewise constant approximation (16) is used. It has the ability to exploit low "local" dimensionality of functions. That is, even though the function f (1) may strongly depend on a large number of variables globally, in any local region the dependence is strong on only a few of them. These few variables may be different in different regions. This ability comes from the recursive nature of the partitioning which causes it to become more and more local as the splitting proceeds. Variables that locally have less influence on the response are less likely to be used for splitting. This gives rise to a local variable subset selection. Global variable subset selection emerges as a natural consequence. Recursive partitioning (15) based on linear functions (5) basically lacks this (local) variable subset selection feature. This tends to limit its power (and interpretability) and is probably the main reason contributing to its lack of popularity.

Another property that recursive partitioning regression exploits is the marginal consequences of interaction effects. That is, a local intrinsic dependence on several variables, when best approximated by an additive function (13), does not lead to a constant model. This is nearly always the case.

Recursive partitioning models using piecewise constant approximations (15) (16) are fairly interpretable owing to the fact that they are very simple and can be represented by a binary tree. [See Breiman et al. (1984) and Section 3.1 below.] They are also fairly rapid to construct and especially rapid to evaluate.

Although recursive partitioning is the most adaptive of the methods for multivariate function approximation it suffers from some fairly severe restrictions that limit its effectiveness. Foremost among these is that the approximating function is discontinuous at the subregion boundaries. This is more than a cosmetic problem. It severely limits the accuracy of the approximation, especially when the true underlying function is continuous. Even imposing continuity only of the function

(as opposed to derivatives of low order) is usually enough to dramatically increase approximation accuracy.

Another problem with recursive partitioning is that certain types of simple functions are difficult to approximate. These include linear functions with more than a few nonzero coefficients [with the piecewise constant approximation (16)] and additive functions (13) in more than a few variables (piecewise constant or piecewise linear approximation). More generally, it has difficulty when the dominant interactions involve a small fraction of the total number of variables. In addition, one cannot discern from the representation of the model whether the approximating function is close to a simple one, such as linear or additive, or whether it involves complex interactions among the variables.

3.0. Adaptive Regression Splines

This section describes the multivariate adaptive regression spline (MARS) approach to multivariate nonparametric regression. The goal of this procedure is to overcome some of the limitations associated with existing methodology outlined above. It is most easily understood through its connections with recursive partitioning regression. It will therefore be developed here as a series of generalizations to that procedure.

3.1. Recursive Partitioning Regression Revisited.

Recursive partitioning regression is generally viewed as a geometrical procedure. This framework provides the best intuitive insight into its properties, and was the point of view adopted in Section 2.4.2. It can however also be viewed in a more conventional light as a stepwise regression procedure. The idea is to produce an equivalent model to (15) (16) by replacing the geometrical concepts of regions and splitting with the arithmetic notions of adding and multiplying.

The starting point is to cast the approximation (15) (16) in the form of an expansion in a set of basis functions

$$\hat{f}(\mathbf{x}) = \sum_{m=1}^{M} a_m B_m(\mathbf{x}). \tag{17}$$

The basis functions B_m take the form

$$B_m(\mathbf{x}) = I[\mathbf{x} \in R_m] \tag{18}$$

where I is an indicator function having the value one if its argument is true and zero otherwise. The $\{a_m\}_1^M$ are the coefficients of the expansion whose values are jointly adjusted to give the best fit to the data. The $\{R_m\}_1^M$ are the same subregions of the covariate space as in (15) (16). Since these regions are disjoint only one basis function is nonzero for any point $\mathbf{x}$ so that (17) (18) is equivalent to (15) (16).

The aim of recursive partitioning is not only to adjust the coefficient values to best fit the data, but also to derive a good set of basis functions (subregions) based on the data at hand. Let $H[\eta]$ be a step function indicating a positive argument

$$H[\eta] = \begin{cases} 1 & \text{if } \eta \geq 0 \\ 0 & \text{otherwise,} \end{cases} \tag{19}$$

and let $LOF(g)$ be a procedure that computes the lack-of-fit of a function $g(\mathbf{x})$ to the data. Then the forward stepwise regression procedure presented in Algorithm 1 is equivalent to the recursive partitioning strategy outlined in Section 2.4.2.

Algorithm 1: (recursive partitioning)

$B_1(\mathbf{x}) \leftarrow 1$
For $M = 2$ to $M_{\max}$ do: $lof^* \leftarrow \infty$
  For $m = 1$ to $M - 1$ do:
    For $v = 1$ to n do:
      For $t \in \{x_{vj} | B_m(\mathbf{x}_j) > 0\}$
        $g \leftarrow \sum_{i \neq m} a_i B_i(\mathbf{x}) + a_m B_m(\mathbf{x}) H[+(x_v - t)] + a_M B_m(\mathbf{x}) H[-(x_v - t)]$
        $lof \leftarrow \min_{a_1, \ldots, a_M} LOF(g)$
        if $lof < lof^*$ then $lof^* \leftarrow lof$; $m^* \leftarrow m$; $v^* \leftarrow v$; $t^* \leftarrow t$ end if
      end for
    end for
  end for
  $B_M(\mathbf{x}) \leftarrow B_{m^*}(\mathbf{x}) H[-(x_{v^*} - t^*)]$
  $B_{m^*}(\mathbf{x}) \leftarrow B_{m^*}(\mathbf{x}) H[+(x_{v^*} - t^*)]$
end for
end algorithm

The first line in Algorithm 1 is equivalent to setting the initial region to the entire domain. The first For-loop iterates the "splitting" procedure with $M_{\max}$ being the final number of regions (basis functions). The next three (nested) loops perform an optimization to select a basis function B_{m^*} (already in the model), a predictor variable x_{v^*} and a "split point" t^*. The quantity being minimized is the lack-of-fit of a model with B_{m^*} being replaced by its product with the step function $H[+(x_{v^*} - t^*)]$, and with the addition of a new basis function which is the product of B_{m^*} and the reflected step function $H[-(x_{v^*} - t^*)]$. This is equivalent to splitting the corresponding region R_{m^*} on variable v^* at split point t^*. Note that the minimization of $LOF(g)$ with respect to the expansion coefficients (line 7) is a linear regression of the response on the current basis function set.

The basis functions produced by Algorithm 1 have the form

$$B_m(\mathbf{x}) = \prod_{k=1}^{K_m} H[s_{km} \cdot (x_{v(k,m)} - t_{km})]. \tag{20}$$

The quantity K_m is the number of "splits" that gave rise to B_m, whereas the arguments of the step functions contain the parameters associated with each of these splits. The quantities s_{km} in (20) take on values ± 1, and indicate the (right/left) sense of the associated step function. The $v(k, m)$ label the predictor variables and the t_{km} represent values on the corresponding variables. Owing to the forward stepwise (recursive) nature of the procedure the parameters for all the basis functions

can be represented on a binary tree that reflects the partitioning history (see Breiman et al., 1984). Figure 1 shows a possible result of running Algorithm 1 in this binary tree representation, along with the corresponding basis functions. The internal nodes of the binary tree represent the step functions and the terminal nodes represent the final basis functions. Below each internal node are listed the variable v and location t associated with the step function represented by that node. The sense of the step function s is indicated by descending either left or right from the node. Each basis function (20) is the product of the step functions encountered in a traversal of the tree starting at the root and ending at its corresponding terminal node.

With most forward stepwise regression procedures it makes sense to follow them by a backwards stepwise procedure to remove basis functions that no longer contribute sufficiently to the accuracy of the fit. This is especially true in the case of recursive partitioning. In fact the strategy here is to deliberately overfit the data with an excessively large model, and then to trim it back to proper size with a backwards stepwise strategy (see Breiman, et al., 1984).

In the case of recursive partitioning the usual straightforward "one at a time" stepwise term (basis function) deletion strategy does not work. Each basis function represents a disjoint subregion and removing it leaves a hole in the predictor variable space within which the model will predict a zero response value. Therefore it is unlikely that any term (basis function) can be removed without seriously degrading the quality of the fit. To overcome this a backward stepwise strategy for recursive partitioning models must delete (sibling) regions in adjacent pairs by merging them into a single (parent) region in roughly the inverse splitting order. One must delete splits rather than regions (basis functions) in the backwards stepwise strategy. One method for doing this is the optimal complexity tree pruning algorithm described in Breiman et al. (1984).

3.2. Continuity.

As noted in Section 2.4.2, a fundamental limitation of recursive partitioning models is lack of continuity. The models produced by (15) (16) are piecewise constant and sharply discontinuous at subregion boundaries. This lack of continuity severely limits the accuracy of the approximation. It is possible, however, to make a minor modification to Algorithm 1 which will cause it to produce continuous models with continuous derivatives.

The only aspect of Algorithm 1 that introduces discontinuity into the model is the use of the step function (19) as its central ingredient. If the step function were replaced by a continuous function of the same argument everywhere it appears (lines 6, 12, and 13), Algorithm 1 would produce continuous models. The choice for a continuous function to replace the step function (19) is guided by the fact that the step function as used in Algorithm 1 is a special case of a spline basis function (6).

The one-sided truncated power basis functions for representing qth order splines are

$$b_q(x-t) = (x-t)_+^q$$

where t is the knot location, q is the order of the spline, and the subscript indicates the positive part of the argument. For $q > 0$ the spline approximation is continuous and has $q-1$ continuous

derivatives. A two-sided truncated power basis is a mixture of functions of the form

$$b_q^{\pm}(x-t) = [\pm(x-t)]_+^q. \tag{21}$$

The step functions appearing in Algorithm 1 are seen to be two-sided truncated power basis functions for $q = 0$ splines.

The usual method for generalizing spline fitting to higher dimensions is to employ basis functions that are tensor products of univariate spline functions (see Section 2.2). Using the two-sided truncated power basis for the univariate functions, these multivariate spline basis functions take the form

$$B_m^{(q)}(\mathbf{x}) = \prod_{k=1}^{K_m} [s_{km} \cdot (x_{v(k,m)} - t_{km})]_+^q, \tag{22}$$

along with products involving the truncated power functions with polynomials of lower order than q. (Note that $s_{km} = \pm 1$.) Comparing (20) with (22) we see that the basis functions (20) produced by recursive partitioning are a subset of a complete tensor product ($q = 0$) spline basis with knots at every (distinct) marginal data point value. Thus, recursive partitioning can be viewed as a forward/backward stepwise regression procedure for selecting a (relatively very small) subset of regressor functions from this (very large) complete basis.

Although replacing the step function (19) by a $q > 0$ truncated power spline basis function (21) in Algorithm 1 will produce continuous models (with $q - 1$ continuous derivatives), the resulting basis will not reduce to a set of tensor product spline basis functions (as was the case for $q = 0$). Algorithm 1 permits multiple splitting on the same variable along a single path of the binary tree (see Fig. 1). Therefore the final basis functions can each have several factors involving the same variable in their product. For $q > 0$ this gives rise to dependencies of higher power than q on individual variables. Products of univariate spline functions on the same variable do not give rise to a (univariate) spline function of the same order, except for the special case of $q = 0$ (21). Each factor in a tensor product spline basis function must involve a different variable and thereby cannot produce dependencies on individual variables of power greater than q. Owing to the many desirable properties of splines for function approximation (de Boor, 1978) it would be nice for a continuous analog of recursive partitioning to also produce them. Since permitting repeated (nested) splits on the same variable is an essential aspect contributing to the power of recursive partitioning, we cannot simply prohibit it in a continuous generalization. A natural resolution to this dilemma emerges from the considerations in Section 3.3.

3.3. A Further Generalization.

Besides lack of continuity, another problem that plagues recursive partitioning regression models is their inability to provide good approximations to certain classes of simple often occurring functions. These are functions that either have no strong interaction effects, or strong interactions each involving at most a few of the predictor variables. Linear (5) and additive (13) functions are among those in this class. From the geometric point of view, this can be regarded as a limitation

of the axis-oriented hyperrectangular shape of the generated regions. These difficult functions (for recursive partitioning) have isopleths that tend to be oriented at oblique angles to the coordinate axes, thereby requiring a great many axis oriented hyperrectangular regions to capture the functional dependence.

One can also understand this phenomenon by viewing recursive partitioning as a stepwise regression procedure (Algorithm 1). It is in this framework that a natural solution to this problem emerges. The goal, in this context, is to find a good set of basis functions for the approximation. The final model is then obtained by projecting the data onto this basis. The bias associated with this procedure is just the average distrance of the true underlying function f (1) from its projection onto the space spanned by the derived basis functions. The variance of the model estimate is directly proportional to the dimensionality of this space, namely the number of basis functions used. In order to achieve good accuracy (small bias and variance) one must derive a small set of basis functions that are close to the true underlying function in the above sense (small bias).

The problem with the basis derived through recursive partitioning (20), or the continuous analog (22), is that it tends to mostly involve functions of more than a few variables (higher order interactions). Each execution of the outer loop in Algorithm 1 (split) removes a basis function of lower interaction order, and replaces it by two functions, each with interaction order one level higher, unless it happens to split on a variable already in the product. Thus, as the partitioning proceeds the average interaction level of the basis function set steadily increases. One simple consequence is that recursive partitioning cannot produce an additive model in more than one variable. The overriding effect is that such a basis involving high order interactions among the variables cannot provide a good approximation to functions with at most low order interactions, unless a large number of basis functions are used. This is the regression analog of trying to approximate with rectangular regions, functions that have isopleths oblique to the axes.

As noted in Section 3.2, recursive partitioning ($q = 0$) can be regarded as a stepwise procedure for selecting a small subset of basis functions from a very large complete tensor product spline basis. The problem is that all members of this complete basis are not eligible for selection, namely many of those that involve only a few of the variables. The problem can be remedied by enlarging the eligible set to include all members of the complete tensor product basis. This in turn can be accomplished by a simple modification to Algorithm 1, or its continuous analog (Section 3.2).

The central operation in Algorithm 1 (lines 6, 12, 13) is to delete an existing (parent) basis function and replace it by both its product with a univariate truncated power spline basis function and the corresponding reflected truncated power function. The modification proposed here involves simply *not* removing the parent basis function. That is, the number of basis functions increases by two as a result of each iteration of the outer loop (split). All basis functions (parent and daughters) are eligible for further splitting. Note that this includes $B_1(\mathbf{x}) = 1$ (line 1). Basis functions involving only one variable (additive terms) can be produced by choosing $B_1(\mathbf{x})$ as the parent. Two-variable basis functions are produced by choosing a single variable basis function as the parent, and so on. Since no restrictions are placed on the choice of a parent term, the modified

procedure is able to produce models involving either high or low order interactions or both. It can produce purely additive models (13) by always choosing $B_1(\mathbf{x})$ as the parent.

This strategy of not removing a parent basis function, after it has been selected for splitting, also resolves the dilemma presented in the last paragraph of Section 3.2. A prohibition against more than one split on the same variable along the path leading to a single basis function can now be enforced without limiting the power of the procedure. Repeated splitting on the same variable is used by ($q = 0$) recursive partitioning to attempt to approximate local additive dependencies. This can now be directly accomplished by repeated selection of the same parent for splitting (on the same variable) thereby introducing additional terms but not increasing the depth of the splitting. There is no longer a need for repeated factors associated with the same variable in a single basis function.

Combining the considerations of this and the preceding section leads to a generalization of recursive partitioning regression involving the following modifications to Algorithm 1:

(a) replacing the step function $H[\pm(x - t)]$ by a truncated power spline function $[\pm(x - t)]_+^q$.

(b) not removing the parent basis function $B_{m^*}(\mathbf{x})$ after it is split, thereby making it and both its daughters eligible for further splitting.

(c) restricting the product associated with each basis function to factors involving distinct predictor variables.

An important consideration in this generalization of recursive partitioning is the degree-of-continuity to impose on the solution; that is, the choice of q (21) (22). There are both statistical and computational trade-offs. These are discussed in Sections 3.7 and 3.9, where it is argued that only continuity of the approximating function and its first derivative should be imposed. Furthermore, the proposed implementing strategy is to employ $q = 1$ splines in the analog of Algorithm 1, and then to use the resulting solution (with discontinuous derivatives) to derive a continuous derivative solution. The detailed discussion of this is deferred to Section 3.7.

3.4. MARS Algorithm.

Algorithm 2 implements the forward stepwise part of the MARS strategy by incorporating the modificatons to recursive partitioning (Algorithm 1) outlined above. Truncated power basis functions ($q = 1$) are substituted for step functions in lines 6, 12, and 13. The parent basis function is included in the modified model in line 6, and remains in the updated model through the logic of lines 12–14. Basis function products are constrained to contain factors involving distinct variables by the control loop over the variables in line 4 [see (20), (22)]. This algorithm produces $M_{\mathbf{max}}$ $q = 1$ tensor product (truncated power) spline basis functions that are a subset of the complete tensor product basis with knots located at all distinct marginal data values. As with recursive partitioning, this basis set is then subjected to a backwards stepwise deletion strategy to produce a final set of basis functions. The knot locations associated with this approximation are then used to derive a piecewise cubic basis, with continuous first derivatives (Section 3.7), thereby producing the final (continuous derivative) model.

Algorithm 2 (MARS – forward stepwise)

$B_1(\mathbf{x}) \leftarrow 1$; $M \leftarrow 2$
Loop until $M > M_{\max}$: $lof^* \leftarrow \infty$
  For $m = 1$ to $M - 1$ do:
    For $v \notin \{v(k,m)|\ 1 \leq k \leq K_m\}$
      For $t \in \{x_{vj}|B_m(\mathbf{x}_j) > 0\}$
        $g \leftarrow \sum_{i=1}^{M-1} a_i B_i(\mathbf{x}) + a_M B_m(\mathbf{x})[+(x_v - t)]_+ + a_{M+1} B_m(\mathbf{x})[-(x_v - t)]_+$
        $lof \leftarrow \min_{a_1,\cdots,a_{M+1}} LOF(g)$
        if $lof < lof^*$ then $lof^* \leftarrow lof$; $m^* \leftarrow m$; $v^* \leftarrow v$; $t^* \leftarrow t$ end if
      end for
    end for
  end for
  $B_M(\mathbf{x}) \leftarrow B_{m^*}(\mathbf{x})[+(x_{v^*} - t^*)]_+$
  $B_{M+1}(\mathbf{x}) \leftarrow B_{m^*}(\mathbf{x})[-(x_{v^*} - t^*)]_+$
  $M \leftarrow M + 2$
end loop
end algorithm

Unlike recursive partitioning, the basis functions produced by Algorithm 2 do not have zero pairwise product expectations; that is, the corresponding "regions" are not disjoint but overlap. Removing a basis function does not produce a "hole" in the predictor space (so long as the constant basis function B_1 is never removed). As a consequence, it is not necessary to employ a special "two at a time" backward stepwise deletion strategy based on sibling pairs. A usual "one at a time" backward stepwise procedure of the kind ordinarily employed with regression subset selection can be used. Algorithm 3 presents such a procedure for use in the MARS context.

Algorithm 3 (MARS – backwards stepwise)

$J^* = \{1, 2, \cdots, M_{\max}\}$; $K^* \leftarrow J^*$
$lof^* \leftarrow \min_{\{a_j|j\in J^*\}} LOF(\sum_{j\in J^*} a_j B_j(\mathbf{x}))$
For $M = M_{\max}$ to 2 do: $b \leftarrow \infty$; $L \leftarrow K^*$
  For $m = 2$ to M do: $K \leftarrow L - \{m\}$
    $lof \leftarrow \min_{\{a_k|k\in K\}} LOF(\sum_{k\in K} a_k B_k(\mathbf{x}))$
    if $lof < b$ then $b \leftarrow lof$; $K^* \leftarrow K$ end if
    if $lof < lof^*$ then $lof^* \leftarrow lof$; $J^* \leftarrow K$ end if
  end for
end for
end Algorithm

Initially (line 1) the model is comprised of the entire basis function set J^* derived from Algorithm 2. Each iteration of the outer For-loop of Algorithm 3 causes one basis function to be deleted. The inner For-loop chooses which one. It is the one whose removal either improves the fit the most or degrades it the least. Note that the constant basis function $B_1(\mathbf{x}) = 1$ is never eligible

for removal. Algorithm 3 constructs a sequence of $M_{\max} - 1$ models, each one having one less basis function than the previous one in the sequence. The best model in this sequence is returned (in J^*) upon termination.

3.5. ANOVA Decomposition.

The result of applying Algorithms 2 and 3 is a model of the form

$$\hat{f}(\mathbf{x}) = a_0 + \sum_{m=1}^{M} a_m \prod_{k=1}^{K_m} [s_{km} \cdot (x_{v(k,m)} - t_{km})]_+. \tag{23}$$

Here a_0 is the coefficient of the constant basis function B_1, and the sum is over the basis functions B_m (22) produced by Algorithm 2 that survive the backwards deletion strategy of Algorithm 3 and $s_{km} = \pm 1$. This (constructive) representation of the model does not provide very much insight into the nature of the approximation. By simply rearranging the terms, however, one can cast the model into a form that reveals considerable information about the predictive relationship between the response y and the covariates $\mathbf{x}$. The idea is to collect together all basis functions that involve identical predictor variable sets.

The MARS model (23) can be recast into the form

$$\begin{aligned} \hat{f}(\mathbf{x}) = a_0 &+ \sum_{K_m=1} f_i(x_i) + \sum_{K_m=2} f_{ij}(x_i, x_j) \\ &+ \sum_{K_m=3} f_{ijk}(x_i, x_j, x_k) + \cdots. \end{aligned} \tag{24}$$

The first sum is over all basis functions that involve only a single variable. The second sum is over all basis functions that involve exactly two variables, representing (if present) two-variable interactions. Similarly, the third sum represents (if present) the contributions from three-variable interactions, and so on.

Let $V(m) = \{v(k,m)\}_1^{K_m}$ be the variable set associated with the mth basis function B_m (23). Then each function in the first sum of (24) can be expressed as

$$f_i(x_i) = \sum_{\substack{K_m=1 \\ i \in V(m)}} a_m B_m(x_i). \tag{25}$$

This is a sum over all single variable basis functions involving only x_i, and is a $q = 1$ spline representation of a univariate function. Each bivariate function in the second sum of (24) can be expressed as

$$f_{ij}(x_i, x_j) = \sum_{\substack{K_m=2 \\ (i,j) \in V(m)}} a_m B_m(x_i, x_j), \tag{26}$$

which is a sum over all two-variable basis functions involving the particular pair of variables x_i and x_j. Adding this to the corresponding univariate contributions (25) (if present)

$$f_{ij}^*(x_i, x_j) = f_i(x_i) + f_j(x_j) + f_{ij}(x_i, x_j) \tag{27}$$

gives a $q = 1$ bivariate tensor product spline approximation representing the joint bivariate contribution of x_i and x_j to the model. Similarly, each trivariate function in the third sum can be obtained by collecting together all basis functions involving the particular variable triples

$$f_{ijk}(x_i, x_j, x_k) = \sum_{\substack{K_m=3 \\ (i,j,k)\in V(m)}} a_m B_m(x_i, x_j, x_k). \tag{28}$$

Adding this to the corresponding univariate and bivariate functions (25) (26) involving x_i, x_j and x_k, provides the joint contribution of these three variables to the model. Terms involving more variables (if present) can be collected together and represented similarly. Owing to its similarity to decompositions provided by the analysis of variance for contingency tables, we refer to (24) as the ANOVA decomposition of the MARS model.

Interpretation of the MARS model is greatly facilitated through its ANOVA decomposition (24). This representation identifies the particular variables that enter into the model, whether they enter purely additively or are involved in interactions with other variables, the level of the interactions, and the other variables that participate in them. Interpretation is further enhanced by representing the ANOVA decomposition graphically. The additive terms (25) can be viewed by plotting $f_i(x_i)$ against x_i as one does in additive modeling. The two-variable contributions can be visualized by plotting $f^*_{ij}(x_i, x_j)$ (27) against x_i and x_j using either contour or perspective mesh plots. Models involving higher level interactions can be (roughly) visualized by viewing plots on variable pairs for several (fixed) values of the other (complementary) variables (see Section 4.7).

3.6. Model Selection

Several aspects of the MARS procedure (Algorithms 2 and 3) have yet to be addressed. Among these are the lack-of-fit criterion LOF (Algorithm 2, line 7, and Algorithm 3, lines 2 and 5), and the maximum number of basis functions $M_{\max}$ (Algorithm 2, line 2, and Algorithm 3, lines 1 and 3). The lack-of-fit criterion used with the algorithm depends on the distance (loss) function Δ specified with the integral (2) or expected (3) error. The most often specified distance is squared-error loss

$$\Delta[\hat{f}(\mathbf{x}), f(\mathbf{x})] = [\hat{f}(\mathbf{x}) - f(\mathbf{x})]^2 \tag{29}$$

because its minimization leads to algorithms with attractive computational properties. As will be seen in Section 3.9, this aspect is very impotant in the context of Algorithm 2, and so squared-error loss is adopted here as well. The goal of a lack-of-fit criterion is to provide a data based estimate of future prediction error (2) (3) which is then minimized with respect to the parameters of the procedure.

As in Friedman and Silverman (1989) and Friedman (1988) we use a modified form of the generalized cross-validation criterion originally proposed by Craven and Wahba (1979)

$$LOF(\hat{f}_M) = GCV(M) = \frac{1}{N}\sum_{i=1}^{N}[y_i - \hat{f}_M(\mathbf{x}_i)]^2 \Big/ \left[1 - \frac{C(M)}{N}\right]^2. \tag{30}$$

Here the dependencies of $\hat{f}$ (23), and the criterion, on the number of (nonconstant) basis functions M is explicitly indicated. The GCV criterion is the average-squared residual of the fit to the data (numerator) times a penalty (inverse denominator) to account for the increased variance associated with increasing model complexity (number of basis functions M).

If the values of the basis function parameters (number of factors K_m, variables $v(k,m)$, knot locations t_{km} and signs s_{km}) associated with the MARS model were determined independently of the data response values $(y_1, \cdots, y_N)$, then only the coefficients $(a_0, \cdots, a_M)$ are being fit to the data. Consequently the complexity cost function is

$$C(M) = \text{trace}(\mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T) + 1 \tag{31}$$

where $\mathbf{B}$ is the $M \times N$ "data"matrix of the M (nonconstant) basis functions ($B_{ij} = B_i(\mathbf{x}_j)$). This is equal to the number of linearly independent basis functions in (23) and therefore $C(M)$ here (31) is just the number of parameters being fit. Using (31) in (30) leads to the GCV criterion proposed by Craven and Wahba (1979).

The MARS procedure (like recursive partitioning) makes heavy use of the response values to construct a basis function set. This is how it achieves its power and flexibility. This (usually dramatically) reduces the bias of model estimates, but at the same time increases the variance since additional parameters (of the basis functions) are being adjusted to help better fit the data at hand. The reduction in bias is directly reflected in reduced (expected) average squared residual (numerator (30)). The (inverse) denominator (30) (31) is, however, no longer reflective of the (increased) variance owing to the additional number of (basis function) parameters as well as their nonlinear nature.

Friedman and Silverman (1989) suggested using (30) as a lack-of-fit criterion in these circumstances, but with an increased cost complexity function $\tilde{C}(M)$ to reflect the additional (basis function) parameters that, along with the expansion coefficients $(a_0, \cdots, a_M)$, are being fit to the data. Such a cost complexity function can be expressed as

$$\tilde{C}(M) = C(M) + d \cdot M. \tag{32}$$

Here $C(M)$ is given by (31) and M is the number of nonconstant basis functions in the MARS model, being proportional to the number of (nonlinear) basis function parameters. The quantity d in (32) represents a cost for each basis function optimization and is a (smoothing) parameter of the procedure. Larger values for d will lead to fewer knots being placed and thereby smoother function estimates.

In the case of additive modeling, Friedman and Silverman (1989) gave an argument for choosing the value $d = 2$, based on the expected decrease in the average-squared residual by adding a single knot to make a piecewise-linear model. The MARS procedure can be forced to produce an additive model by simply modifying the upper limit of the outer For-loop in Algorithm 2 (line 3) to always have the value one. With this modification only the constant basis function B_1 is eligible for

"splitting" and the resulting model is a sum of functions each of a single variable ($K_m = 1$) which, after the corresponding ANOVA decomposition (24), assumes the form of an additive model (13). Restricting MARS in this manner leads to a palindromically invariant version of the Friedman and Silverman (1989) procedure. This type of additive modeling is a restricted version of general MARS modeling. A higher degree of optimization (over m) is being performed by the latter causing the data at hand to be fit more closely, thereby increasing variance. In order for the GCV criterion (30) (31) (32) to reflect this, an even larger value for d is appropriate.

One method for choosing a value for d in any given situation would be to simply regard it as a parameter of the procedure that can be used to control the degree of smoothness imposed on the solution. Alternatively it could be estimated through a standard sample reuse technique such as bootstrapping (Efron, 1983) or cross-validation (Stone, 1974). In a fairly wide variety of simulation studies (a subset of which are presented in Section 4) the resulting model and its accuracy (2) (3) are seen to be fairly independent of the value chosen for the parameter d (32). These simulation studies indicate:

(1) The optimal cost complexity function $\tilde{C}(M)$ to be used in the GCV criterion (30) (in the context of MARS modeling) is a monotonically increasing function with decreasing slope as M increases.

(2) The approximation (32), with $d = 3$, is fairly effective, if somewhat crude.

(3) The best value for d in any given situation depends (weakly) on M, N, n and the distribution of the covariate values in the predictor space.

(4) Over all situations studied, the best value for d is in the range $2 \leq d \leq 4$.

(5) The actual accuracy in terms of either integral (2) (29) or expected (3) (29) squared error is fairly insensitive to the value of d in this range.

(6) The value of the GCV criterion for the final MARS model does exhibit a moderate dependence on the value chosen for d.

A consequence of (5) and (6) is that while how well one is doing with the MARS approach is fairly independent of d, how well one thinks one is doing (based on the optimized GCV score) does depend somewhat on its value. Therefore, a sample reuse technique might be used to obtain an additional estimate of the goodness-of-fit of the final model if it needs to be known fairly precisely.

The strategy in recursive partitioning regression (Breiman et al., 1984) is to let the forward stepwise procedure produce a fairly large number of regions (basis functions) and then have the backwards stepwise procedure trim the model back to an appropriate size. The arguments in favor of this apply equally well to the MARS approach. Therefore, the value chosen for $M_{\max}$ in Algorithms 2 and 3 should be considerably larger than the optimal (minimal GCV) model size M^*. Typically choosing $M_{\max} = 2M^*$ is sufficient.

3.7. Degree-of-Continuity.

One of the central ideas leading to the MARS generalization of recursive partitioning is to replace the step function implicit in the latter with a truncated power spline basis function (21). This leads to an approximation in the form of an expansion in tensor product spline basis functions.

The continuity properties of this approximation are governed by the order q (21) chosen for the univariate spline functions comprising the tensor products; derivatives exist to order q.

If the intent is accurate estimation of the function (as opposed to its derivatives of various orders) then there is little to be gained by imposing continuity beyond that of the function itself. If the true underlying function nowhere has a very large local second derivative then a small additional increase in accuracy can be achieved by imposing continuous first derivatives. Also, continuous (first) derivative approximations have considerably more cosmetic appeal. There is, however, little to be gained and in fact much to lose, by imposing continuity beyond that of the first derivative, especially in high dimensional settings.

The difficulty with higher order regression splines centers on so called "end effects." The largest contribution to the average approximation error (2) (3) emanates from locations $\mathbf{x}$ near the boundaries of the domain. This phenomenon is well known even in univariate smoothing ($n = 1$), and is especially severe in higher dimensions. As the dimension of the covariate space increases, the fraction of the data points near a boundary increases rapidly. Fitting high degree polynomials (associated with high degree regression splines) in these regions leads to very high variance of the function estimate there. This is mainly due to the lack of constraints on the fit at the boundaries.

One approach that has been suggested (Stone and Koo, 1985) is to modify the spline basis functions so that near the ends of the data interval (on each variable) they smoothly join a linear function. This can substantially help moderate the bad end effects of (unmodified) regression splines in the case of smoothing ($n = 1$) and additive modeling (13), although the approximating basis functions can still have very large slope near the boundaries. A computationally simpler way to ensure a linear approximation near the boundaries is to make a piecewise-linear approximation everywhere by using $q = 1$ tensor product splines. This is accomplished in the MARS approach by using $q = 1$ truncated power (univariate) spline basis functions (21) in Algorithm 2 (lines 6, 12, and 13).

A piecewise-linear approximation, of course, does not possess continuous derivatives. The lowest order spline approximation with continuous derivatives involves $q = 2$ univariate spline basis functions. Their use, however, leads to the problems cited above. Motivated by the approach of Stone and Koo (1985) we fit with a modified basis set. These functions resemble $q = 1$ splines, but have continuous derivatives.

The model (23) produced by Algorithms 2 and 3 involves a sum of products of functions of the form

$$b(x|s,t) = [s(x-t)]_+. \tag{33}$$

Our strategy for producing a model with continuous derivatives is to replace each such function by

a corresponding truncated cubic function of the form

$$C(x|s=+1,t_-,t,t_+) = \begin{cases} 0 & x \le t_- \\ p_+(x-t_-)^2 + r_+(x-t_-)^3 & t_- < x < t_+ \\ x - t & x \ge t_+ \end{cases}$$
$$C(x|s=-1,t_-,t,t_+) = \begin{cases} -(x-t) & x \le t_- \\ p_-(x-t_+)^2 + r_-(x-t_+)^3 & t_- < x < t_+ \\ 0 & x \ge t_+ \end{cases} \tag{34}$$

with $t_- < t < t_+$. Setting

$$\begin{aligned} p_+ &= (2t_+ + t_- - 3t)/(t_+ - t_-)^2 \\ r_+ &= (2t - t_+ - t_-)/(t_+ - t_-)^3 \\ p_- &= (3t - 2t_- - t_+)/(t_- - t_+)^2 \\ r_- &= (t_- + t_+ - 2t)/(t_- - t_+)^3 \end{aligned} \tag{35}$$

causes $C(x|s,t_-,t,t_+)$ to be continuous and have continuous first derivatives. There are second derivative discontinuities at $x = t_\pm$. Each truncated linear function (33) is characterized by a single knot location t, whereas each corresponding truncated cubic function (34) (35) is characterized by three knots; these are a central knot t and upper/lower side knots $t_\pm$. Figure 2a compares the two functions.

Each factor (33) in every basis function in the approximation produced by Algorithms 2 and 3 (23) is replaced by a corresponding truncated cubic factor (34) (35). The central knot t (34) is placed at the same location as the (single) knot for its associated truncated linear function (33). The side knots $t_\pm$, $t_- < t < t_+$, are located so as to reduce the number of second derivative discontinuities. This is accomplished through the ANOVA decomposition of the MARS model (24) (25) (26) (28).

Each basis function m in (23) has a knot set $\{t_{km}\}_1^{K_m}$. The ANOVA decomposition collects together all basis functions corresponding to exactly the same variable set $\{v(k,m)\}_1^{K_m}$. Thus, the knot sets associated with each ANOVA function (25) (26) (28) can be viewed as a set of points (multivariate knots) in the same K_m-dimensional space. The projections of these points onto each of the respective K_m axes, $v(k,m)$, gives the knot locations of the factors that correspond to that variable. These are the central knot locations for the piecewise cubic factors. The side knots $t_\pm$ for each cubic factor are placed at the midpoints between its central knot and the two adjacent central knots in the same projection. The lower/upper side knots for the corresponding smallest/largest projected central knot value are placed midway between the central knot and the smallest/largest data value on that variable. Figure 2b illustrates this process for one and two dimensional ANOVA functions.

The final model is obtained by fitting the resulting piecewise cubic basis function set to the data. It will have continuous first (but not second) derivatives. The contribution to the fitted model of each basis function far from its central knot location will be the same as its corresponding piecewise linear basis function. Therefore this continuous derivative model will tend to have the same highly desirable boundary properties as the piecewise linear model produced by Algorithms

2 and 3. The important ingredient is that the slope of each univariate basis factor never exceeds a value of one.

3.8. Knot Optimization.

The MARS procedure (Algorithm 2), as well as recursive partitioning (Algorithm 1), can be viewed as a technique for developing a multivariate model, based on sums of products of univariate functions (17) (20) (23), through the use of a univariate smoother. This can be seen by casting the (MARS) model in the form

$$\hat{f}(\mathbf{x}) = \hat{f}_{\setminus mv}(\mathbf{x}) + B_m(\mathbf{x})\varphi_{mv}(x_v), \tag{36}$$

with

$$\hat{f}_{\setminus mv}(\mathbf{x}) = \sum_{\substack{V(i)\neq V(m) \\ V(i)\neq V(m)+\{v\}}} a_i B_i(\mathbf{x}). \tag{37}$$

The function $\varphi_{mv}(x_v)$ has the form

$$\varphi_{mv}(x_v) = c_0 + \sum_{j=1}^{J} c_j[s_j(x_v - t_j)]_+ \tag{38}$$

which is just a ($q = 1$) piecewise linear spline representation of a univariate function. (In the case of recursive partitioning it would be a $q = 0$ piecewise constant representation.) The second term in (36) isolates the contributions to the model of the variable set $V(m)$ of the mth basis function, and the set including the variable v with $V(m)$. The first term (37) represents the contributions of the variables from the other basis functions. Minimization of the lack-of-fit criterion (30) in Algorithm 2 (line 7) performs a joint optimization of the current model with respect to the coefficients $(c_0, \cdots, c_J)$ of the univariate function φ_{mv} (38) and those of the other basis functions $\{a_i\}$ (37).

Expressing the current model in the form given by (36) and letting

$$R_{\setminus mv} = y - \hat{f}_{\setminus mv} \tag{39}$$

this optimization can be written in the form

$$E[R_{\setminus mv} - B_m\varphi_{mv}(x_v)]^2 = \min. \tag{40}$$

Here the dependence of the respective quantities on the multivariate argument $\mathbf{x}$ has been surpressed. If one fixes the values of the coefficients $\{a_i\}$ (37) then (40) has the general solution

$$\varphi_{mv}(x_v) = E\left[B_m^2\left(\frac{R_{\setminus mv}}{B_m}\right)\middle| x_v\right] / E[B_m^2|x_v], \tag{41}$$

which can be estimated by a weighted smooth of $R_{\setminus mv}/B_m$ on x_v, with weights given by B_m^2. Letting this smooth take the form of a piecewise linear spline approximation (38) gives rise (in an indirect way) to the MARS approach.

This framework provides the connection between MARS and the smoothing (and additive modeling) method (TURBO) suggested by Friedman an Silverman (1989). They presented a forward stepwise strategy for knot placement in a simple piecewise linear smoother. The inner For-loop of Algorithm 2 can be viewed as an application of this strategy for choosing the best location for the next knot t_J in $\varphi_{mv}(x_v)$ (38) (41) in the more general context of MARS modeling. In fact, in the univariate case ($n = 1$) the MARS algorithm simply represents a (palindromically invariant) version of TURBO. As noted above, restricting the upper limit of the outer For-loop to always have the value one (Algorithm 2) gives rise (for $n > 1$) to the TURBO method of additive modeling.

Both recursive partitioning (Algorithm 1, line 5) and MARS (Algorithm 2, line 5) make every distinct (nonzero weighted) marginal data value eligible for knot placement. As pointed out by Friedman and Silverman (1989), this has the effect of permitting the corresponding piecewise linear smoother (38) (41) to achieve a local minimum span of one observation. In noisy settings this can lead to locally high variance of the function estimate. There is no way that a smoother, along with its lack-of-fit criterion, can distinguish between sharp structure in the true underlying function f (1), and a run of either positive or negative error values ϵ (1). If one assumes (as one must) that the underlying function is smooth compared to the noise, then it is reasonable to impose a minimum span on the smoother that makes it resistant to runs in the noise of length likely to be encountered in the errors. In the context of piecewise linear smoothing this translates into a minimal number L of (nonzero weighted) observations between each knot. Assuming a symmetric error distribution, Friedman and Silverman (1989) use a coin tossing argument to propose choosing $L = L^*/2.5$ with L^* being the solution to

$$\Pr(L^*) = \alpha. \tag{42}$$

Here $\Pr(L^*)$ is the probability of observing a (positive or negative) run of length L^* or longer in nN_m tosses of a fair coin, and α is a small number (say $\alpha = 0.05$ or 0.01). The quantity N_m is the number of observations for which $B_m > 0$ (Algorithm 2, line 5). The relevant number of tosses is nN_m since there are that many potential locations for each new knot (inner two For-loops in Algorithm 2) for each basis function B_m (outer For-loop).

For $nN_m \geq 10$ and $\alpha < 0.1$ a good approximation to L^* (42) is

$$L^* = -\log_2\left[-\frac{1}{nN_m}\ell n(1-\alpha)\right],$$

so that a reasonable number of counts between knots is given by

$$L(\alpha) = -\log_2\left[-\frac{1}{nN_m}\ell n(1-\alpha)\right]/2.5. \tag{43}$$

The denominator in (43) arises from the fact that a piecewise linear smoother must place between two and three knots in the interval of the run to respond to it, and not degrade the fit anywhere else. Using (43) gives the procedure resistance, with probability $1 - \alpha$, to a run of positive or negative error values in the interior of the interval.

The arguments that lead to (43) do not apply to the ends of the interval; that is, $L(\alpha)$ refers to the number of counts between knots but not to the number of counts between the extreme knot locations and the corresponding ends of the interval defined by the data. As discussed in Section 3.7, it is essential that end effects be handled well for the procedure to be successful. An argument analogous to the one that leads to $L(\alpha)$ (43) for the interior, can be advanced for the ends. The probability of a run of length L^* or longer of positive or negative error values at the beginning or end of the data interval is 2^{-L^*+3}. There are n such intervals, corresponding to the n predictor variables, so that the total probability of encountering an end run is

$$\Pr(L^*) = n2^{-L^*+3}. \tag{44}$$

Therefore requiring at least

$$Le(\alpha) = 3 - \log_2(\alpha/n) \tag{45}$$

observations between the extreme knots and the corresponding ends of the interval provides resistance (with probability $1 - \alpha$) to runs at the ends of the data intervals.

The quantity α in (43) (45) can be regarded as another smoothing parameter of the procedure. Both $L(\alpha)$ (43) and $Le(\alpha)$ (45) are, however, fairly insensitive to the value of α. The differences $L(.01)$–$L(.05)$ and $Le(.01)$–$Le(.05)$ are both approximately equal to 2.3 observations. In any case both expressions can only be regarded as approximate since they only consider the signs and ignore the magnitudes of the errors in the run.

It can be noted that (36), (37), (39), and (41) could be used to develop a generalized backfitting algorithm based on a (univariate) local averaging smoother, in direct analogy to the backfitting algorithm for additive modeling (Friedman and Stuetzle, 1981, Breiman and Friedman, 1985, and Buja, Hastie and Tibshirani, 1989). Like MARS, this generalized backfitting algorithm could be used to fit models involving sums of products of univariate curve estimates. It would, however, lack the flexibility of the MARS procedure (especially in high dimensions) owing mainly to the latter's close relation to the recursive partitioning approach (local variable subset selection – see Sections 2.4.2 and 6). It would also tend to be computationally far more expensive.

3.9. Computational Considerations

In Sections 3.2 and 3.3 the MARS procedure was motivated as a series of conceptually simple extensions to recursive partitioning regression. In terms of implementation, however, these extensions produce a dramatic change in the algorithm. The usual implementations of recursive partitioning regression [AID (Morgan and Sonquist, 1963) and CART (Breiman, et al., 1984)] take strong advantage of the special nature of step functions, along with the fact that the resulting basis functions have disjoint support, to dramatically reduce the computation associated with the middle and inner For-loops of Algorithm 1 (lines 4 and 5). In the case of least-squares fitting, very simple updating formulae can be employed to reduce the computation for the associated linear (least-squares) fit (line 7) from $O(NM^2 + M^3)$ to $O(1)$. The total computation can therefore

be made proportional to $nN\ M_{\max}$, after sorting. Unfortunately, these same tricks cannot be applied to the implementation of the MARS procedure. In order to make it computationally feasible, different updating formulae must be derived for the MARS algorithm.

The minimization of the lack-of-fit criterion (30) in Algorithm 2 (line 7) is a linear least-squares fit of the response y on the current basis function set (line 6). There are a variety of techniques for numerically performing this fit. The most popular, owing to its superior numerical properties, is based on the QR decomposition (see Golub and Van Loan, 1983) of the basis "data" matrix $\mathbf{B}$,

$$B_{mi} = B_m(\mathbf{x}_i). \tag{46}$$

As noted above, however, computational speed is of paramount importance since this fit must be repeated many times in the course of running the algorithm. A particular concern is keeping the computation linear in the number of observations N, since this is the largest parameter of the problem. This rules out the QR decomposition technique in favor of an approach based on using the Cholesky decomposition to solve the normal equations

$$\mathbf{B}^T\mathbf{B}\mathbf{a} = \mathbf{B}^T\mathbf{y} \tag{47}$$

for the vector of basis coefficients $\mathbf{a}$ (line 6). Here $\mathbf{y}$ is the (length N) vector of response values. This approach is known to be less numerically stable than the QR decomposition technique. Also, the truncated power basis (21) is the least numerically stable representation of a spline approximation. Therefore, a great deal of care is required in the numerical aspects of an implementation of this approach. This is discussed below.

If the basis functions are centered to have zero mean then the matrix product $\mathbf{B}^T\mathbf{B}$ is proportional to the covariance matrix of the current basis function set. The normal equations (47) can be written

$$\mathbf{V}\mathbf{a} = \mathbf{c} \tag{48}$$

with

$$\begin{aligned} V_{ij} &= \sum_{k=1}^{N} B_j(\mathbf{x}_k)[B_i(\mathbf{x}_k) - \bar{B}_i], \\ c_i &= \sum_{k=1}^{N} (y_k - \bar{y})B_i(\mathbf{x}_k), \end{aligned} \tag{49}$$

and $\bar{B}_i$ and $\bar{y}$ the corresponding averages over the data. These equations (48) (49) must be resolved for every eligible knot location t, for every variable v, for all current basis functions m, and for all iterations M of Algorithm 2 (lines 2, 3, 4, and 5). If carried out in a straightforward manner this would require computation proportional to

$$C \sim nNM_{\max}^4(\alpha N + \beta M_{\max})/L \tag{50}$$

with α and β constants of proportionality and L given by (43). This computational burden would be prohibitive except for very small problems or very large computers. Although it is not possible

to achieve the dramatic reduction in computation for MARS as can be done for recursive partitioning regression, one can reduce the computation enough, so that moderate sized problems can be conveniently run on small computers. Following Friedman and Silverman (1989), the idea is to make use of the special properties of the $q = 1$ truncated power spline basis functions to develop rapid updating formulae for the quantities that enter into the normal equations (48) (49), as well as to take advantage of the rapid updating properties of the Cholesky decomposition (see Golub and Van Loan, 1983).

The most important special property of the truncated power basis used here is that each (univariate) basis function is characterized by a single knot. Changing a knot location changes only one basis function, leaving the rest of the basis unchanged. Other bases for representing spline approximations, such as the minimal support B-splines, have superior numerical properties but lack this important computational aspect. Updating formulae for B-splines are therefore more complex giving rise to slower computation.

The current model (Algorithm 2, line 6) can be reexpressed as

$$g' \leftarrow \sum_{i=1}^{M-1} a_i B_i(\mathbf{x}) + a_M B_m(\mathbf{x}) x_v + a_{M+1} B_m(\mathbf{x})(x_v - t)_+. \tag{51}$$

The inner For-loop (line 5) minimizes the GCV criterion (30) jointly with respect to the knot location t and the coefficients $a_1, \cdots, a_{M+1}$. Using g' (51) in place of g (line 6) yields an equivalent solution with the same optimizing GCV criterion lof^* (line 8) and knot location t^* (line 8). (The solution coefficient values will be different.) The advantage of using g' (51) is that only one basis function is changing as the knot location t changes.

Friedman and Silverman (1989) developed updating formulae for least-squares fitting of $q = 1$ splines by visiting the eligible knot locations in decreasing order and taking advantage of that fact for $t \leq u$

$$(x - t)_+ - (x - u)_+ = \begin{cases} 0 & x \leq t \\ x - t & t < x < u \\ u - t & x \geq u. \end{cases}$$

The Friedman and Silverman (1989) updating formulae can be extended in a straightforward man-

ner to the more general MARS setting, giving ($t \leq u$)

$$
\begin{aligned}
c_{M+1}(t) &= c_{M+1}(u) + \sum_{t \leq x_{vk} < u} (y_k - \bar{y}) B_{mk}(x_{vk} - t) \\
&\quad + (u - t) \sum_{x_{vk} \geq u} (y_k - \bar{y}) B_{mk}, \\
V_{i,M+1}(t) &= V_{i,M+1}(u) + \sum_{t \leq x_{vk} < u} (B_{ik} - \bar{B}_i) B_{mk}(x_{vk} - t) \\
&\quad + (u - t) \sum_{x_{vk} \geq u} (B_{ik} - \bar{B}_i) B_{mk}, \qquad (1 \leq i \leq M), \\
V_{M+1,M+1}(t) &= V_{M+1,M+1}(u) + \sum_{t \leq x_{vk} < u} B_{mk}^2 (x_{vk} - t)^2 \\
&\quad + (u - t) \sum_{x_{vk} \geq u} B_{mk}^2 (2x_{vk} - t - u) + (s^2(u) - s^2(t))/N,
\end{aligned}
\tag{52}
$$

with $s(t) = \sum_{x_{vk} \geq t} B_{mk}(x_{vk} - t)$. In (52) B_{ik} and B_{mk} are elements of the basis function "data" matrix (46), the x_{vk} are elements of the original data matrix, and y_k are the data response values.

These updating formulae (52) can be used to obtain the last $(M + 1)$st row (and column) of the basis covariance matrix $\mathbf{V}$ and last element of the vector $\mathbf{c}$ at all eligible knot locations t with computation proportional to $(M + 2)N_m$. Here N_m is the number of observations for which $B_m(\mathbf{x}) > 0$ (line 5). Note that all the other elements of $\mathbf{V}$ and $\mathbf{c}$ do not change as the knot location t changes. This permits the use of updating formulae for the Cholesky decomposition to reduce its computation from $O(M^3)$ to $O(M^2)$ (in solving the normal equations (48)) at each eligible knot location. Therefore the computation required for the inner For-loop (lines 5–9) is proportional to $\alpha M N_m + \beta M^2 N_m / L$. This gives an upper bound on total computation for Algorithm 2 as being proportional to

$$
C^* \sim n N M_{\max}^3 (\alpha + \beta M_{\max}/L). \tag{53}
$$

Thus, comparing with (50) the use of updating formulae is seen to reduce the computation roughly by a factor of $N M_{\max}/L$. For typical values of $N = 200$, $M_{\max} = 30$, and $L = 5$ this reduces the required computation by roughly a factor of 1000.

Table 1 shows the total computation time (sec.) of the MARS procedure as a function of $M_{\max}$ for one of the examples (AC circuit impedance, Section 4.4.1) discussed below. These times were obtained on a SUN Microsystems Model 3/260 (with floating point accelerator). For this example $n = 4$ and $N = 200$. The computation scales linearly in both n and N with the MARS algorithm.

Three timing sequences are shown in Table 1, corresponding to different constraints being placed on the final MARS model. The first row ($mi = 1$) corresponds to an additive model where interactions among the variables are prohibited. As mentioned above, this is accomplished by suppressing the outer For-loop in Algorithm 2 (line 3) and only allowing the constant basis function $B_1(\mathbf{x}) = 1$ to appear in the products with the univariate spline basis functions. This, of course, reduces the total computation roughly by a factor proportional to $M_{\max}$. The second row

($mi = 2$) only allows two-variable interactions to appear in the model. This reduces computation a little since only previous basis functions involving one variable are permitted to appear in the products. The last row in Table 1 ($mi = n$) shows the times for the fully unconstrained MARS model.

It should be noted that in all the examples discussed in this paper (some of which, like this one, involve fairly complex functions) the optimal number of basis functions was between 10 and 15, so that $M_{\max}$ values around 20 to 30 were appropriate. Setting $M_{\max} = 50$ permits the procedure to use from 125 to 200 degrees of freedom to fit the final model, if required, thereby allowing it to approximate very complex functions. Clearly though, for very large problems (say $n > 20$ and $N > 1000$) either long execution times or fast computers (compared to the one used here) would be required. It can also be noted that the MARS algorithm admits a high degree of parallelization so that it could run very fast on computers with parallel architectures.

Updating formulae for higher order ($q > 1$) truncated power spline basis functions (21) can be developed in analogy to (52). They would, however, be far more complex than those for $q = 1$ leading to much slower execution of the algorithm. Also, their corresponding numerical properties would be very much worse.

At any point during the execution of Algorithm 2 the current basis function set need not be linearly independent. (The basis functions set comprising the final model is, however, always linearly independent.) Therefore, the covariance matrix $\mathbf{V}$ appearing in the normal equations (48) may be singular. This presents no fundamental problem since they can be solved by applying pivoting in the Cholesky decomposition (see Dongarra, Moler, Bunch and Stewart, 1979). A better strategy from the point of view of MARS modeling is, however, to slightly modify the normal equations via

$$(\mathbf{V} + \epsilon \mathbf{D})\mathbf{a} = \mathbf{c} \tag{54}$$

where $\mathbf{D}$ is a diagonal $(M + 1) \times (M + 1)$ matrix comprised of the diagonal elements of $\mathbf{V}$. The coefficients for the basis function set $\mathbf{a}$ are then taken to be the solution derived from (54). The average-squared-residual

$$ASR(\mathbf{a}) = \frac{1}{N}\left[\sum_{k=1}^{N}(y_k - \bar{y})^2 - \sum_{i=1}^{M+1} a_i(c_i + \delta D_{ii}a_i)\right] \tag{55}$$

is still used as the numerator of the GCV criterion (30). The value for δ is taken to be a small number just large enough to maintain numerical stability.

The principal advantage of this "ridge regression" approach (54) is that it eliminates the need for pivoting in the Cholesky decomposition update, thereby increasing execution speed. Additional advantages are that it increases numerical stability to help compensate for the bad numerical properties of the truncated power spline basis representation, and it applies a small overall shrinkage to the solution coefficients to help compensate for the selection bias inherent in stepwise regression procedures (see Copas, 1983).

The updating formulae (52) are not necessarily numerically stable. Widely different locations and scales for the predictor variables can cause instabilities that adversely effect the quality of the final model. The MARS procedure is (except for numerics) invariant to the locations and scales of the predictor variables. It is therefore reasonable to perform a transformation that causes the resulting locations and scales to be most favorable from the point of view of numerical stability. Standardizing them to each have zero location and unit scale provides good numerical properties.

4.0. Simulation Studies and Examples

In the following sections we present the results of applying the MARS procedure to a series of simulated and real data sets. The goal is to try to gain some understanding of its properties and to learn in what situations one might expect it to provide better performance than existing methodology. In all the examples the smoothing parameter d (32) was taken to be $d = 3$. The software automatically reduces it to $2d/3$ $(= 2)$ for additive modeling. The minimum number of observations between knots was determined by (43) and the number between the extreme knots and the edges was determined by (45), both with $\alpha = 0.05$. In all examples the explanatory variables were standardized to aid in numerical stability (see Section 3.9). In all simulation studies the covariate vectors were independently drawn (from the same sampling distribution) for each replication of the experiment. Therefore, nonidentical (random) designs were realized for each of the 100 replications. All results reported are for the continuous derivative (piecewise cubic) model (see Section 3.7) unless otherwise noted.

4.1. Modeling Pure Noise

With a modeling procedure as flexible as MARS, a reasonable concern is that it might find considerable spurious structure in data for which the signal to noise ratio is small. This false structure would reflect the sampling fluctuations in the noise ϵ (1) and would provide a misleading indication of the association between the response and predictor variables. One would expect this effect to be especially severe for small samples in high dimensions. Our first simulation study indicates that this tends not to be the case for the MARS procedure.

Tables 2a and 2b summarize the results of applying MARS to pure noise $f(\mathbf{x}) = 0$ (1). Results are presented for two dimensionalities $(n = 5, 10)$ and three sample sizes $(N = 50, 100, 200)$. The summary consists of the percent points of the lower half of the distribution of the optimizing GCV score (30) for the MARS model, scaled by that for the corresponding constant model $\hat{f}(\mathbf{x}) = \bar{y}$. These distributions were obtained by applying MARS to 100 data sets for which the response values were randomly generated from a normal distribution and the covariate vectors were randomly generated from a uniform distribution in R^n. As in Table 1, Tables 2a and 2b show the results for three types of constraints being placed on the model. These constraints are controlled by the parameter mi, which is the maximum number of variables allowed to appear in any basis function, $K_m \leq mi$ (23), thereby controlling the number of variables that can participate in interaction effects. For $mi = 1$ the model is restricted to be additive in the variables, whereas for $mi = 2$, interactions are limited to those involving (at most) two variables. Setting $mi = n$ places no

constraint on the number of variables that can enter into interactions.

This simulation study represents one test of the lack-of-fit criterion based on the GCV score (30) using the cost complexity criterion $\tilde{C}(M)$ (31) (32). Tables 2a and 2b show that the MARS procedure in this situation seldom claims to produce a model that fits the data markedly better than the response mean. Over half of the time (as reflected by the median) it claims to provide no better fit than the constant model at all dimensionalities and sample sizes shown. Even the best MARS fit over the 100 trials (1% point) does not produce a distinctly superior GCV value to the constant (no structure) fit on the same data. This is especially noteworthy given the small sample sizes for these dimensionalities.

4.2. MARS Modeling on Additive Data.

A related concern to that of the previous section concerns what happens when MARS is applied in situations where the true underlying function f (1) is additive (13) in the predictor variables. It might be expected that given the ability of MARS to introduce a large number of complex interactions into its models, that it might be somewhat at a disadvantage in these situations when compared to procedures that restrict the model to be additive. This section presents a simulation study that indicates that this is not the case.

We use for this study an example presented in Friedman and Silverman (1989)

$$f(\mathbf{x}) = 0.1e^{4x_1} + 4/[1 + e^{-20(x_2-1/2)}] + 3x_3 + 2x_4 + x_5 + 0 \cdot \sum_{i=6}^{10} x_i. \tag{56}$$

This function has a nonlinear additive dependence on the first two variables, a linear dependence on the next three, and is independent of the last five (pure noise) variables. A simulation study was performed consisting of 100 replications of the following experiment. First $N(= 50, 100, 200)$ ten dimensional ($n = 10$) covariate vectors were generated in the unit hypercube. Then corresponding response values were assigned according to

$$y_i = f(\mathbf{x}_i) + \epsilon_i, \qquad 1 \leq i \leq N \tag{57}$$

with the ϵ_i randomly generated from a standard normal and $f(\mathbf{x})$ given by (56). Here the signal to noise ratio is 3.28 so that the true underlying function (56) accounts for 91% of the variance of the response (57).

A reasonable strategy to be used with MARS modeling is to fit both an additive model ($mi = 1$) and one that permits interactions ($mi = 2$ or $mi = n$). The respective GCV scores of both models can then be compared and the one corresponding to the lowest score chosen. With this strategy a model involving interactions is only used if it claims (through its GCV score) to do better than an additive model on the same data.

For each of the 100 replications (at each sample size N) the MARS procedure was applied with $mi = 1, 2$, and n. The first value ($mi = 1$) corresponds to additive modeling, the second ($mi = 2$) permits interactions in at most two variables, whereas the last ($mi = n = 10$) fits an

unconstrained MARS model. For the last two mi values the corresponding interaction fit was only used if it produced a smaller GCV score than the additive model on the same data.

Table 3 compares the average accuracy of using this strategy to that of additive modeling on the purely additive data (56) (57). The principal measure of accuracy is the (scaled) integral-squared-error ISE (2) (29)

$$ISE = \int_D [\hat{f}(\mathbf{x}) - f(\mathbf{x})]^2 d^n x / \mathrm{Var}_{\mathbf{x}\in D} f(\mathbf{x}), \tag{58}$$

where here $n = 10$ and D is the ten-dimensional unit hypercube. For each replication of the simulation study, the ISE (58) was estimated by Monte Carlo integration using 5000 points randomly generated from a uniform distribution over D.

A closely related quantity of interest is the (scaled) predictive-squared-error

$$PSE = E[y - \hat{f}(\mathbf{x})]^2 / \mathrm{Var}\ \ y \tag{59}$$

which is related to the ISE (58) by

$$PSE = (ISE \cdot\ \mathrm{Var}_{\mathbf{x}\in D} f(\mathbf{x}) +\ \mathrm{Var}\ \epsilon)/\mathrm{Var}\ \ y \tag{60}$$

with ϵ being the error component (1). If the response values $y_1, \cdots, y_N$ are standardized to have unit variance then the GCV criterion (30) (31) (32) is intended as an estimate of the PSE (59) (60), and the ratio GCV/PSE provides an estimate of how well the optimized GCV criterion for the model is estimating the true PSE.

Shown in Table 3 are the average ISE (58), PSE (59) and GCV/PSE, along with the corresponding standard deviations (in parentheses) over the 100 replications at each sample size N $(= 50, 100, 200)$. (Note that the standard deviations of the averages shown in the table are one tenth the corresponding standard deviations shown.) Comparing the first row $(mi = 1)$ to the next two $(mi = 2, n)$ shows that there is little sacrifice in accuracy using interaction models in the context of the above strategy, even though the true underlying function (56) involves no interactions. Table 3 also shows that the optimized GCV score produced by the MARS fit slightly overestimates (on average) the actual predictive-squared-error for this problem at all sample sizes studied. This effect is most pronounced for the smallest sample size $(N = 50)$. (Note that the variability of this ratio as reflected by its standard deviation over the 100 replications if fairly high.)

The above strategy chooses the additive model over that involving interactions only if the former produces a GCV score no worse than the latter. The first column of Table 4 shows the number of times the additive model was chosen in the 100 replications of this simulation experiment. As can be seen, the additive model was being chosen most of the time. This is why the loss in accuracy was so slight. A more conservative strategy would be to accept the additive model if its GCV score is only slightly (say 5 or 10 percent) worse. The second and third columns of Table 4 show the number of times the additive model is chosen under these two slightly more conservative scenarios. In these cases the fit involving interactions is seen to be almost never chosen.

The results of this simulation study indicate that on data for which the true underlying function f (1) is additive, the MARS procedure does not produce fits involving interactions that appear distinctly superior to an additive model. This basically is another test of the lack-of-fit criterion (30) (31) (32), and especially the choice of $d = 3$ (32) for general MARS modeling and $d = 2$ for additive modeling with MARS (Friedman and Silverman, 1989).

4.3. A Simple Function of Ten Variables.

The previous examples (Sections 4.1 and 4.2) tested the ability of MARS to avoid finding structure when it is not present. It is at least equally important that it find structure when it does exist. The next several examples examine the ability of MARS to uncover interaction effects that are present in data. The first test example is taken from Friedman, Grosse and Stuetzle (1983). They considered trying to model the function

$$f(\mathbf{x}) = 10\sin(\pi x_1 x_2) + 20\left(x_3 - \frac{1}{2}\right)^2 + 10x_4 + 5x_5 \tag{61}$$

in the $n = 6$ dimensional unit hypercube using $N = 200$ points. The covariates were randomly generated from a uniform distribution and the responses were assigned using (57) with $f(\mathbf{x})$ given by (61) and with ϵ being a standard normal deviate.

We consider here this same function (61) but in a more difficult setting. First we reduce the sample size to $N = 100$. In addition we increase the dimensionality of the covariate space to $n = 10$, so that instead of one noise variable, there are now five such variables that are independent of $f(\mathbf{x})$. For this study however the MARS procedure has no prior knowledge of the nature of the dependence of $f(\mathbf{x})$ on any of the variables. The signal to noise ratio for this example is high (4.8/1); the true underlying function accounts for 96% of the variance of the response. On the other hand, the dimension of the covariate space is high ($n = 10$), the sample is small ($N = 100$) and the function (61) is fairly highly structured.

Table 5a summarizes the MARS model derived from a data set generated from the above prescription. The data response values were standardized so that the resulting GCV scores are an estimate of the PSE (59). The first two lines in Table 5a give the optimizing GCV scores for the corresponding piecewise linear (23) and piecewise cubic (Section 3.7) fits. The third line gives the total number of (nonconstant) basis functions M in the final model, whereas the fourth line gives $\tilde{C}(M)$ (31) (32), the estimated number of linear degrees-of-freedom used in the fit. The ANOVA decomposition is summarized by one row for each ANOVA function. The columns represent summary quantities for each one. The first column lists the function number. The second gives the standard deviation of the function. This gives one indication of its (relative) importance to the overall model and can be interpreted in a manner similar to a standardized regression coefficient in a linear model. The third column provides another indication of the importance of the corresponding ANOVA function, by listing the GCV score for a model with all of the basis functions corresponding to that particular ANOVA function removed. This can be used to judge whether this ANOVA function is making an important contribution to the model, or whether it just slightly helps to

improve the global GCV score. The fourth column gives the number of basis functions comprising the ANOVA function while the fifth column provides an estimate of the additional number of linear degrees-of-freedom used by including it. The last column gives the particular predictor variables associated with the ANOVA function.

The MARS fit is seen in Table 5a to have produced seven ANOVA functions, the first five involving only one variable ($K_m = 1$), and two comprised of two variables ($K_m = 2$). Judging from the second and (especially) the third columns, the last ANOVA function (involving an interaction between variables 2 and 6) is not very important to the fit. Its standard deviation is much smaller than that of the other ANOVA functions and removing it from the fit degrades the GCV score for the entire fit imperceptibly (see line 1). All other ANOVA functions seem important to the model in that the removal of any of them substantially degrades the quality of the fit.

After removing the unneeded (seventh) ANOVA function, the resulting MARS model is seen to be additive in variables 3, 4, and 5, and involves a two- variable interaction effect between variables 1 and 2. Note that this model shows no indication of a dependence of the response on the last five (pure noise) variables. Figure 3 shows a graphical representation of these ANOVA functions. The three additive contributions $f_4(x_4)$, $f_5(x_5)$ and $f_3(x_3)$ (25) are plotted in the first three frames. The joint contribution of the first two variables $f^*_{1,2}(x_1, x_2)$ (27) is presented in three different views of a perspective mesh plot of this function (last three frames). Low values of the corresponding variables are indicated by the position of the "0", whereas higher values are in the direction of the axis label. Note that the plotting routine does not show the bivariate function outside the convex hull of the projected data points. As discussed in Section 3.7, the MARS procedure basically extrapolates linearly beyond the boundaries of the data. It is important to note that this surface does not represent a smooth of y on x_1 and x_2, but rather it shows the contribution of x_1 and x_2 to a smooth of y on the *ten* variables $x_1, \cdots, x_{10}$.

Comparing the results of the MARS fit to these data (Table 5a and Figure 3) with the true underlying function $f(\mathbf{x})$ (61), shows that the resulting model provides a fairly accurate and interpretable description. This is especially noteworthy given the high dimensionality ($n = 10$) and the small sample size ($N = 100$).

Table 5a (and Figure 3) show results for one realization of a data set with $N = 100$. In order to assess the general ability of MARS to model data of this kind, it must be applied to a large number of realizations of this situation. Table 5b summarizes the results of running MARS on 100 replications of this example at three sample sizes ($N = 50$, 100, and 200), in the same format as Table 3. (Here the strategy of choosing the additive fit if it produced a better GCV score was not used for the $mi = 2, 10$ models.) For the very smallest sample size ($N = 50$) the additive model ($mi = 1$) is seen to actually produce more accurate fits than those involving interactions ($mi = 2, 10$), even though there are strong interaction effects (involving x_1 and x_2) in the generated data. This is due to the bias-variance tradeoff. Even though the additive model is highly biased, its lower variance leads to lower estimation errors. When the sample size is increased, however, this is no longer the case and the models involving interactions produce vastly superior accuracy.

As in the additive case (Table 3) the optimized GCV criterion is seen to overestimate the actual PSE on average (again with fairly high sample to sample variability). It is interesting to note that even though the true underlying function (61) exhibits interactions involving only two variables, the totally unconstrained MARS fit ($mi = 10$) produces models nearly as accurate as when the fit is constrained to have at most two-variable interactions ($mi = 2$).

4.4. Alternating Current Series Circuit.

Figure 4a shows a schematic diagram of a simple alternating current series circuit involving a resistor R, inductor L, and capacitor C. Also in the circuit is a generator that places a voltage

$$V_{ab} = V_o \sin \omega t \tag{62a}$$

across the terminals a and b. Here ω is the angular frequency which is related to the cyclic frequency f by

$$\omega = 2\pi f. \tag{62b}$$

The electric current I_{ab} that flows through the circuit is also sinusoidal with the same frequency,

$$I_{ab} = (V_o/Z) \sin(\omega t - \phi). \tag{62c}$$

Its amplitude is governed by the impedance Z of the circuit and there is a phase shift ϕ, both depending on the components in the circuit:

$$Z = Z(R, \omega, L, C),$$
$$\phi = \phi(R, \omega, L, C).$$

From elementary physics one knows that

$$Z(R, \omega, L, C) = [R^2 + (\omega L - 1/\omega C)^2]^{1/2}, \tag{63a}$$

$$\phi(R, \omega, L, C) = \tan^{-1}\left[\frac{\omega L - 1/\omega C}{R}\right]. \tag{63b}$$

The purpose of this exercise is to see to what extent the MARS procedure can approximate these functions and perhaps yield some insight into the variable relationships, in the range

$$\begin{aligned} &0 \leq R \leq 100 \text{ ohms} \\ &20 \leq f \leq 280 \text{ hertz} \\ &0 \leq L \leq 1 \text{ henries} \\ &1 \leq C \leq 11 \text{ micro farads.} \end{aligned} \tag{64}$$

Two hundred four-dimensional uniform covariate vectors were generated in the ranges (64). For each one, two responses were generated by adding normal noise to (63a) and (63b). The variance of the noise was chosen to give a 3 to 1 signal to noise ratio for both Z (63a) and ϕ (63b), thereby

causing the true underlying function to account for 90% of the variance in both cases. As with the previous example, the data response values were standardized.

4.4.1. Impedance Z

Applying MARS to the impedance data (63a) (64) (with 3/1 signal to noise) gave an optimizing GCV score of 0.19. The corresponding GCV Scores for an additive model ($mi = 1$) was 0.56, whereas that for $mi = 2$ was 0.19. The additive model is seen (not surprisingly from the known truth) to be inadequate. Perhaps more surprising is the fact that even though the true underlying function (63a) has interactions to all orders, an approximation involving at most two-variable interactions gives just as good a fit to these data. Owing to its increased interpretability we select the $mi = 2$ model.

Table 6a summarizes the ($mi = 2$) MARS fit. There are five ANOVA functions all of which, except for the last, seem important to the model. There is an additive contribution from R, and interactions between ωC and ωL. Figure 4b displays a graphical representation of the ANOVA decomposition. The upper left frame shows the additive contribution $f_R(R)$ (25), the upper right shows the joint contribution of ω and C, $f^*_{\omega C}(\omega, C)$ (27), while the bottom two frames show $f^*_{\omega L}(\omega, L)$ (27) from two different perspectives.

The dependence of the impedance Z (62) on the resistance R of the circuit is seen to be roughly linear. Its joint dependence on ω and C is seen to be fairly mild except when they both achieve simultaneously very low values, in which case the impedance increases very sharply. For low frequencies ω, the impedance Z is seen to be high and independent of the inductance L. For high ω, Z has a monotonically increasing dependence on L. For low L, Z monotonically decreases with increasing ω, whereas for high L values, the impedance is seen to achieve a minimum for moderate ω. These interpretations are based on visual examination of the graphical representation of the ANOVA decomposition of the MARS model, based on a sample of size $N = 200$. Since the data are in this case generated from known truth, one can examine the generating equation (63a) to verify their general correctness.

Table 6b summarizes the results of a simulation study based on 100 replications of AC circuit impedance data (63a) (64) (3/1 signal to noise) at three sample sizes ($N = 100$, 200, and 400). Additive modeling ($mi = 1$) is seen to perform badly at all sample sizes. The accuracy of the models involving interactions improves sharply with increasing sample size. The $mi = 2$ models offer slightly higher accuracy in most situations. Unlike the previous examples the GCV score is seen to underestimate the true predictive squared error PSE (59) a little on average.

4.4.2. Phase Angle φ

The MARS procedure applied to the phase angle data (63b) (64) (3/1 signal to noise) with $mi = 1, 2, 4$ gave optimizing GCV scores of 0.30, 0.22, and 0.22, respectively. Here the additive model, while still being less accurate, is more competitive with those involving interactions. The model limited to two-variable interactions ($mi = 2$) is again seen to fit the data as well as the general ($mi = 4$) MARS model. Table 7a summarizes the $mi = 2$ model. It involves nine ANOVA

functions, two of which are clearly unimportant (6 and 7), and three more that are of marginal importance (5, 8, and 9). Figure 4c shows perspective mesh plots of all six bivariate functions f^* (27) associated with the four variables. The dependence of the phase angle φ on all of the variables is seen to be more gentle and more nearly additive than the impedance Z (Figure 4b).

Table 7b gives the results of applying MARS to 100 replications of the phase angle data at $N = 100$, 200, and 400. At the smallest sample size additive fitting is seen to be almost as accurate as with interactions. This is, however, no longer true at the larger sample sizes. The optimizing GCV score is seen to slightly underestimate the true PSE (on average) for most of the situations, but as with the previous examples, the variance of the ratio (GCV/PSE) dominates this small bias.

4.5. Portuguese Olive Oil.

For this example MARS is applied to data from analytical chemistry. The observations consist of 417 samples of olive oil from Portugal (Forina, et al., 1983). On each sample, measurements were made on the concentrations of 10 acids and three sterols (see Table 8). Also recorded was the location where the sample originated. The purpose was to see if there is a relation between the chemical composition and geographical origin. Of particular interest was the extent to which olive oil from northeastern Portugal (Douro Valley – 90 samples) differed from that of the rest of Portugal (327 samples). One way to address this question is to examine the results of trying to model the probability that a sample originates from the Douro Valley given its measured chemical composition (Table 8). The response variable y in this case takes on only two values: 1 = Duoro Valley, 0 = rest of Portugal. Since $\Pr(y = 1 \mid \mathbf{x}) = E(y \mid \mathbf{x})$ one can estimate this probability through regression techniques.

Linear logistic regression (Cox, 1970) is often used when the response variable assumes only two values. The model takes the form

$$\log[p/(1-p)] = \beta_0 + \sum_{i=1}^{n} \beta_i x_i$$

where p is the probability that y assumes its larger value. The coefficients $\{\beta_i\}_0^n$ are estimated by (numerically) maximizing the likelihood of the data. Hastie and Tibshirani (1986) extended this approach to additive logistic regression

$$\log[p/(1-p)] = \sum_{i=1}^{n} f_i(x_i).$$

The smooth covariate functions are estimated through their "local scoring" algorithm. The model can be further generalized to involve potential interaction effects by

$$\log[p/(1-p)] = \hat{f}(\mathbf{x}) \tag{65}$$

with $\hat{f}(\mathbf{x})$ taking the form of the MARS approximation. This can be implemented in the MARS algorithm by simply replacing the internal linear least-squares routine (LOF – Algorithm 2, line 7,

and Algorithm 3, lines 2 and 5) by one that does linear logistic regression (given the current set of multivariate spline basis functions). Unless rapid updating formulae can be derived this is likely to be quite computationally intensive. A compromise strategy, however, is likely to provide a good approximation; the multivariate spline basis functions are selected using the MARS squared-error based loss criterion, and the coefficients $\{a_m\}_0^M$ for the final model are fit using a linear logistic regression on this basis set. In this mode one takes advantage of the local variable subset selection aspect of MARS as well as its ability to produce continuous models. The detailed knot placement on the selected variables will, however, be optimally placed for the untransformed response rather than for the logistic fit.

Table 9 gives the results of six different analyses (rows) on the Portuguese olive oil data set. The first column describes the method. The first two rows are for MARS runs on the untransformed 0/1 response using its least-squares criterion. The next two rows give results when a post (linear) logistic regression is applied to the final basis function set as described above. The parameter mi is (as before) the maximum number of variables permitted in any basis function ($mi = 1$ gives an additive model; $mi = 2$ allows two variable interactions). The last two rows are (respectively) ordinary stepwise linear logistic regression and recursive partitioning [CART: Breiman, et al. (1984)]. The second column gives the number of variables that entered each final model; the third column gives the GCV estimate of the PSE (59); the fourth column gives another estimate of this quantity (CV) based on tenfold cross-validation; and the last column gives the cross-validated error rate of a prediction rule that assigns $\hat{y} = 1$ if the estimated conditional probability $\hat{E}(y \mid \mathbf{x})$ is greater than 0.5 and assigns $\hat{y} = 0$ otherwise. In the case of logistic regression the GCV and CV scores were computed using as a (squared-error) loss function

$$[y - 1/(1 + e^{-\hat{f}(\mathbf{x})})]^2,$$

with $\hat{f}(\mathbf{x})$ the corresponding MARS estimate of the log-odds (65).

Table 9 indicates that the (post) logistic transformation improves the fit substantially. The GCV and CV estimates are seen to be fairly close, especially for the (untransformed) least-squares fits. The introduction of interaction effects ($mi = 2$) seems to improve the quality of the fit, especially for the logistic models. (Increasing the value for mi beyond two did not result in further improvement.) CART and stepwise linear logistic regression do not perform as well as the logistic MARS model involving two variable interactions on these data, although (like MARS) they do indicate a strong association between geographical origin (Douro Valley versus rest of Portugal) and the chemical composition (Table 8) of the olive oil samples. This (strong) association can be expressed with a small fraction of the 13 original predictor variables. (When using GCV or CV scores for comparisons it should be remembered that they not only estimate the accuracy of the approximation (2) (3) but also include the irreducible (binomial) error. Therefore their ratios understate the actual accuracy ratios of the methods being compared.)

Table 10 provides a summary of the logistic ($mi = 2$) MARS model. There are four ANOVA functions involving three (of the 13) predictor variables. All four ANOVA functions appear important to the fit, including the last two that involve interactions. Figure 5 shows the joint contribution

(to the log-odds) of the second and twelfth variables $f^*_{2,12}(x_2, x_{12})$ (27) in the upper left frame, and $f^*_{6,12}(x_6, x_{12})$ in the upper right frame. The two lower frames show the same two plots respectively from a different perspective. The MARS model for the log-odds is (in this case) the sum of these two bivariate functions.

4.6. Low Dimensional Modeling

The main advantage of MARS modeling over existing methodology is clearly realized in high dimensional settings. It is, however, (unlike recursive partitioning) competitive in low dimensions ($n \leq 2$) as well. Friedman and Silverman (1989) studied its properties for the nonparametric smoothing problem ($n = 1$) and showed that it can produce superior performance especially in situations involving small samples and low signal to noise. (They also showed that for additive modeling (13) ($n > 1$, $mi = 1$) it is quite competitive with procedures based on the "backfitting" algorithm using local averaging smoothers.) In the univariate case ($n = 1$) the MARS method can be viewed as a modification of an approach first suggested by Smith (1982). In a massive simulation study Breiman and Peters (1988) showed that this was one of the best all around smoothers of those they tested.

In this section we illustrate the use of MARS for two-dimensional nonparametric smoothing ($n = 2$). The first example is taken from Andrews and Herzberg (1985). The data comes from an experiment on the recoiling of guns (Brouncker, 1734). A total of 109 shots were fired at different distances D between the muzzle and the target, and with varying grains of powder in the charge C. The upper left frame of Figure 6 shows the experimental design. Note that there are 40 distinct points in the design so that some points represent more than one firing. The response variable is the (standardized) distance by which the resulting shot missed the target.

The MARS procedure applied to these data resulted in a model with three basis functions and an optimized GCV score of 0.39. The upper right and lower left frames of Figure 6 show two views of the MARS surface smooth. The average shooting error is seen to increase linearly with shooting distance for all powder charges. As might be expected, at the shortest distance the error is very small and independent of the size of the powder charge. As the distance increases, the dependence of the shooting error on charge becomes nonmonotonic with the degree of nonmonotonicity increasing with shooting distance. The optimal (lowest shooting error) charge is seen to increase somewhat as the distance increases. This error is seen to be asymmetric about the minimum with the degree of asymmetry increasing with distance. For moderate to large distances it appears to be much more costly (in terms of average accuracy) to shoot with too small a powder charge than with one that is too large.

Our second example is an artificial one used by Gu, Bates, Chen, and Wahba (1990) to illustrate interaction spline smoothing (see Section 2.3) They generated 300 points (more or less) randomly from a uniform distribution in the unit square and set the response to

$$\begin{aligned} y_i = {} & 40\exp\{8[(x_{1i}-.5)^2+(x_{2i}-.5)^2]\}/\exp\{8[(x_{1i}-.2)^2+(x_{2i}-.7)^2]\} \\ & +\exp\{8[(x_{1i}-.7)^2+(x_{2i}-.2)^2]\}+\epsilon_i, \qquad 1 \leq i \leq 300. \end{aligned} \tag{66}$$

The errors ϵ_i were drawn from a standard normal distribution. Here the signal to noise is 3.15/1 so that the true underlying function accounts for about 91% of the variance of the response. Figure 7a shows a mesh plot of the true underlying function $y - \epsilon$ (66) over the unit square.

The optimized MARS model on these data consisted of 18 basis functions using an estimated 45 linear degrees-of-freedom for the fit. The corresponding surface estimate is shown in Figure 7b. Although this estimate is a bit smoother than the one produced by interaction splines (see Gu et al., 1990, Fig. 4) they both have nearly the same accuracy in terms of expected squared error (3) (29). Since this is a relatively well behaved function and the sample size is quite large ($N = 300$) it is likely that methods based on kernel smoothing (Cleveland and Devlin, 1988) and (nonadaptive) tensor product splines (de Boor, 1978) would also do well in this case.

Although MARS is competitive with other methodologies in low dimensions, what sets it apart is its ability to handle problems involving many variables. Suppose the variables (x_1, x_2) in (66) were two out of ten variables, all of which (jointly) effect the response y in an unknown way, and the goal is to estimate the dependence of the response on x_1, x_2, accounting for the effect of the other eight (nuisance) variables $x_3, \cdots, x_{10}$. To get an idea of how well MARS might do in such a problem, $N = 300$ points were generated in the $n = 10$ dimensional unit hypercube. The dependence of this response $\tilde{y}$ was taken to be

$$\tilde{y}_i = y_i + f(\mathbf{x}_i) \tag{67}$$

with y_i given by (66) and $f(\mathbf{x})$ given by (61) (shifted by two arguments). Thus, the dependence of $\tilde{y}$ on the first two variables is the same as in the previous two-dimensional example, whereas its dependence on the last eight is the same as the first eight variables of the problem studied in Section 4.3. The sample size ($N = 300$) and the pure noise level are the same as in the two-dimensional problem. The apparent noise in the $x_1 x_2$-plane is now however many times greater owing to the variability induced in the response by the many nuisance variables. It is hoped that one can account for the variance associated with the nuisance variables by fitting a (nonparametric) model jointly with respect to them and the variables of interest, thereby obtaining a better estimate of the dependence of the response on x_1, x_2.

Applying MARS to this ten dimensional data resulted in a model with 27 basis functions using an estimated 68.5 linear degrees-of-freedom. Eleven of the basis functions were associated with the dependence on variables x_1 and x_2, accounting for 27.5 linear degrees-of-freedom. Figure 8a shows the resulting surface estimate. Although it is not quite as accurate as the smooth (Figure 7b) produced in the absence of the eight nuisance variables, it still gives a good indication of the nature of the joint dependence of the response on x_1 and x_2. The estimate (Figure 8a) is smoother than that of Figure 7b owing to the fact that it is based on 11 rather than 18 basis functions. Figure 8b shows plots of all of the ANOVA functions produced by the MARS fit to the 10-dimensional data. The estimates corresponding to the (nuisance) variables associated with $f(\mathbf{x})$ (61) (67) are actually better than those obtained in the example of Section 4.3. This is the result of having 300 observations here, whereas only 100 were used in Section 4.3.

4.7. Slicing a Mars Model

In the examples so far presented the dominant interactions involved at most two variables. The resulting MARS models could then be visualized through plots of the contributing ANOVA functions (25) (27). When substantial interaction effects involving more than two variables are present the model becomes more difficult to interpret. This section describes an interpretational tool called "slicing" that can aid in the visualization of such models.

The MARS model (23) (34) is a sum of products of univariate functions

$$\hat{f}(\mathbf{x}) = a_0 + \sum_{m=1}^{M} a_m \prod_{k=1}^{K_m} b_{km}(x_{v(k,m)}). \tag{68}$$

Here b_{km} is either a $q = 1$ spline basis function (33) or its cubic analog (34). If the number of factors K_m in any product is greater than two then $\hat{f}(\mathbf{x})$ is more difficult to interpret since the ANOVA decomposition (24) contains functions of more than two variables which are difficult to plot or visualize. It is possible, however, to get a rough idea of the behavior of $\hat{f}(\mathbf{x})$ by reducing the dimensionality of the predictor variable space by repeatedly conditioning on subsets of the variables.

Let $\mathbf{Z}$ represent a (selected) subset of the predictor variables $\{x_1 \cdots x_n\}$, with dimension d ($< n$), and $\tilde{\mathbf{Z}}$ the complement subset of dimension $n - d$. Define a d-dimensional *slice* of the predictor variable space by simultaneously assigning specific values to the selected variables

$$\text{slice} : \{Z_1 = z_1, \cdots, Z_d = z_d\} \equiv \mathbf{Z} = \mathbf{z}. \tag{69}$$

The MARS model along the slice will be a function of the variables $\tilde{\mathbf{Z}}$ complement to those defining the slice

$$\hat{f}(\mathbf{x} \mid \mathbf{Z} = \mathbf{z}) = \hat{f}(\tilde{\mathbf{Z}} \mid \mathbf{z}) = a_0 + \sum_{m=1}^{M} a_m \prod_{k=1}^{K_m} b_{km}(x_{v(k,m)} \mid \mathbf{Z} = \mathbf{z}). \tag{70}$$

The particular form of the MARS model (68) makes the sliced model (70) especially straightforward to compute by decomposing its products into those factors involving variables defining the slice, and those involving the complement variables,

$$\hat{f}(\mathbf{x}) = a_0 + \sum_{m=1}^{M} a_m \prod_{x_{v(\ell,m)} \in \mathbf{Z}} b_{\ell m}(x_{v(\ell,m)}) \prod_{x_{v(k,m)} \in \tilde{\mathbf{Z}}} b_{km}(x_{v(k,m)}). \tag{71}$$

For a given slice ($\mathbf{Z} = \mathbf{z}$) the first product in (71) evaluates to a constant (multiplying the coefficient a_m) and the second product gives the dependence on the complement variables. The MARS model along the slice can therefore be represented as

$$\hat{f}(\tilde{\mathbf{Z}} \mid \mathbf{z}) = c_0(\mathbf{z}) + \sum_{m=1}^{\tilde{M}} c_m(\mathbf{z}) \prod_{k=1}^{\tilde{K}_m} b_{km}(\tilde{\mathbf{Z}}_{v(k,m)}) \tag{72a}$$

with

$$c_0(\mathbf{z}) = a_0 + \sum_{\{x_{v(\ell,m)}\} \subseteq \mathbf{Z}} a_m \prod b_{\ell m}(z_{v(\ell,m)}), \tag{72b}$$

$$c_m(\mathbf{z}) = \sum_{\{x_{v(k,m)}\} \subseteq \tilde{Z}} a_m \prod_{x_{v(\ell,m)} \in \mathbf{Z}} b_{\ell m}(z_{v(\ell,m)}). \tag{72c}$$

The sum in (72b) is over all basis functions in (68) (71) that involve only variables defining the slice. The sum in (72c) is over all basis functions that involve exactly the same complement variables $\{\tilde{\mathbf{Z}}_{v(k,m)}\}$.

For a given slice (69) the MARS model along the slice (72a) has the same (constructive) form as any ordinary MARS model (68), and thus has a corresponding ANOVA decomposition that can be interpreted graphically as illustrated in the previous sections. This suggests the following strategy for trying to visualize models that contain interactions involving more than two variables:

(1) Use the ANOVA decomposition of the full MARS model $\hat{f}(\mathbf{x})$ (24) to identify those variables that participate in interactions with more than one other variable.

(2) Choose a variable subset $\mathbf{Z}$ for slicing, such that the MARS model along the slice $\hat{f}(\tilde{\mathbf{Z}} \mid \mathbf{z})$ involves (at most) two-variable interactions.

(3) Examine graphically the ANOVA decomposition (25) (27) of $\hat{f}(\tilde{\mathbf{Z}} \mid \mathbf{z})$ for various values of the slice $(\mathbf{Z} = \mathbf{z})$.

This slicing strategy is illustrated on testing data taken from a semiconductor component. The predictor variables $V_1 \cdots V_4$ are the simultaneous voltages applied to the terminals of a four terminal semiconductor resistor in the ranges $-6 \leq V_1 \leq 6$, $-.75 \leq V_2 \leq 10.75$, and $-.5 \leq (V_3, V_4) \leq 5.5$. The response is the current I into one of the terminals. There were 599 observations taken. Table 11 provides a summary of the results of four MARS runs on these data. The first column gives the maximum number of variables mi allowed to participate in interactions. The second column gives the GCV estimate (30) (32) of the $\sqrt{PSE}$ (59) and the third gives another estimate of this quantity based on 10-fold cross-validation. The last three columns give respectively the median, 75th percentile, and maximum values of the distribuiton of the absolute cross-validated residuals divided by the absolute deviation of the response from its mean value. Table 11 shows results only for piecewise-linear fits; in all cases the corresponding piecewise-cubic models gave rise to much larger values and thus worse fits.

Increasing the permissible interaction level is seen to improve the general quality of the fit. The GCV score appears to rather dramatically underestimate the PSE as estimated from cross-validation. Inspection of the (cross-validated) residual distributions reveals that they are highly skewed toward large values. The GCV score is seen to reflect the size of the typical residuals (median - 75% point) but not the few extremely large ones. Increasing the interaction level beyond two seems to preferentially reduce the larger residuals.

Figure 9a shows a graphical representation of the ANOVA decomposition for the MARS model involving ony two-variable interactions ($mi = 2$). The resistor is seen to be a very nonlinear device. The current appears to be roughly independent of the terminal voltages except when one or more of them take on extreme values where the current changes rapidly.

As seen in Table 11 the MARS model involving four-variable interactions provides a subtantially better fit to these data. Figures 9b and 9c explore this function of four variables using the

slicing strategy described above. Figure 9b shows the current as a function of V_1 and V_2 along four slices defined by V_3 and V_4. In this figure the functions are not plotted on the same vertical scale. In order to relate the scales, the maximum value of each function relative to that of the first (upper left frame) is shown above each plot. Since the relative locations of the plots are arbitrary they are each plotted to have a minimum value of zero, so that the maximum value is equal to the range. The (V_3, V_4) slices are taken at the four extreme corners of the $V_3 - V_4$ design. Both the magnitude and shape of the dependence of the current on V_1 and V_2 are seen to depend rather strongly on the values of V_3 and V_4. For simultaneously low values of V_3 and V_4 (upper left) the dependence is seen to be roughly linear, whereas when V_4 takes on its highest value, with V_3 at its lowest value (lower left) the current is seen to vary much less as a function of V_1 and V_2. For high values of V_3 the dependence is similar to that of the lower left frame except for the existence of the dramatic peak for low values of V_2.

Figure 9c shows the dependence of the current (as reflected by the MARS fit) on V_3 and V_4 for four slices on (V_1, V_2). Here the slices do not include the two extreme corners on V_1 for low values of V_2 since they are outside the support of the design. (This can be seen on the $V_1 - V_2$ plots; the functions are plotted only within the convex hull of the bivariate distributions.) In Figure 9c the functions are all plotted on the same scale. As would be expected from the previous results, the dependence of the current on V_3 and V_4 changes substantially with differing values of V_1 and V_2.

Exploring the resulting MARS model in this manner provides some insight into the nature of the cross-validated residual distributions observed in Table 11. The function has very high (and increasing) first and second derivatives very near some of the (joint) boundaries. When extreme observations on these boundaries are deleted during the cross-validation, the resulting slopes are underestimated and the extrapolation to the left out observation gives rise to a large error. This phenomenon also explains why the piecewise-linear models give rise to much better fits. There are clearly small local regions where the second derivatives are very large. By approximating these by infinite second derivatives the piecewise-linear model is able to come closer than the piecewise-cubic fit which tries to moderate these locally very high second derivatives.

Figures 9b and 9c represent a small subset of all possible revealing slices of this four-variable function. In general, slicing is likely to be most effective when performed in an interactive manner. Functional dependencies revealed by inspecting the results of a particular slice will likely suggest further slices to try. The straightforward and rapid calculation of sliced models (72) from the complete MARS model (68) (71) might make feasible the computation of sliced functions in real time on modern workstations. In this case the values defining the slice could be defined (and changed) in a continuous manner through a graphical input device (such as by moving a mouse) and the continuously changing functions along the slice can be viewed in real time.

5.0. Remarks.

This section covers various aspects (extensions, limitations, etc.) of the MARS procedure not discussed in the previous sections.

5.1. Constraints.

The MARS procedure is nonparametric in that it attempts to model arbitrary functions. It is often appropriate, however, to place constraints on the final model, dictated by knowledge of the system under study, outside the specific data at hand. Such constraints will reduce the variance of the model estimates, and if the outside knowledge is fairly accurate, not substantially increase the bias. One type of constraint has already been discussed in Section 4, namely limiting the maximum interaction order (mi) of the model. Setting this maximum to a small value ($mi \lesssim 2$) causes the resulting (restricted) model to be more interpretable since its ANOVA components (Section 3.5) can be directly visualized graphically. With this restriction the MARS model has the same form as a low dimensional expansion (additive model, interaction splines – see Section 2.3). Unlike those methods however, which require the variable subsets to be preselected (in advance), MARS automatically selects them separately for each problem at hand based on the data. It also automatically (and adaptively) selects the (different) degree of smoothing to be applied in estimating the separate functions of each of the variable subsets it produces. In situations where this adaptability is not important one might apply MARS to obtain the low dimensional variable subsets (ANOVA decomposition) and then apply a less adaptive smoothing method (kernel with the backfitting algorithm, or interaction splines (12)) on these subsets to obtain the final function estimates.

One might in addition (or instead) limit the specific variables that can participate in interactions. If it is known a priori that certain variables are not likely to interact with others, then restricting their contributions to be at most additive can improve accuracy. If one further suspects that specific variables can only enter linearly, then placing such a restriction can improve accuracy. The incremental charge d (32) for knots placed under these constraints should be less than that for the unrestricted knot optimization. (The implementing software charges $2d/3$ and $d/3$ respectively, for the additive and linear constraints where d is the charge for unrestricted knot optimization.)

These constraints, as well as far more sophisticated ones, are easily incorporated in the MARS strategy. Before each prospective knot is considered (Algorithm 2, lines 6–8), the parameters of the corresponding two new potential multivariate spline basis functions (v, t, and B_m) can be examined for consistency with the constraints. If they are inconsistent, they can be made ineligible for inclusion in the model by simply skipping lines 6–8 in Algorithm 2.

5.2. Semiparametric Modeling

Another kind of a priori knowledge that is sometimes available has to do with the nature of the dependence of the response on some (or all) the predictor variables. The user may be able to provide a function $g(\mathbf{x})$ that is thought to capture some aspects of the true underlying function $f(\mathbf{x})$. More generally, one may have a set of such functions $\{g_j(\mathbf{x})\}_1^J$, each one of which might capture some special aspect of the functional relationship. A semiparametric model of the form

$$\hat{f}_{sp}(\mathbf{x}) = \sum_{j=1}^{J} c_j g_j(\mathbf{x}) + \hat{f}(\mathbf{x}), \tag{73}$$

where $\hat{f}(\mathbf{x})$ takes the form of the MARS approximation, could then be fit to the data. The coefficients c_j in (73) are jointly fit along with the parameters of the MARS model in Algorithms 2 and 3. To the extent that one or more of the g_j successfully describe attributes of the true underlying function, they will be included with relatively large (absolute) coefficients, and the accuracy of the resulting (combined) model will be improved.

Semiparametric models of this type (73) are easily fit using the MARS strategy. One simply includes $\{g_j(\mathbf{x})\}_1^J$ as J additional predictor variables $(x_{n+1}, \cdots, x_{n+J})$ and constrains their contributions to be linear. One could also, of course, not place this constraint, thereby fitting more complex semiparametric models than (73).

Another strategy that is often employed in this context is to first fit only the parametric component to the data and then apply a nonparametric method (such as MARS) to the residuals of the parametric fit. In general, this strategy is likely to be less successful because the residual function may be more highly structured than the original one (and thus more difficult to approximate) especially if the parametric approximation is not close to the true underlying function. The more general approach (73) allows the fitting procedure to automatically adjust the strength of the parametric components as part of the fitting process.

5.3. Collinearity

Extreme collinearity of the predictor variables is a fundamental problem in the modeling of observational data. Solely in terms of predictive modeling it represents an advantage in that it effectively reduces the dimensionality of the predictor variable space. This is only true provided that the observed collinearity is a property of the population distribution and not an artifact of the sample at hand. Collinearity presents, on the other hand, severe problems for interpreting the resulting model.

This problem is even more serious for (interactive) MARS modeling than for additive or linear modeling. Not only is it difficult to isolate the separate contributions of highly collinear predictor variables to the functional dependence, it is also difficult to separate the additive and interactive contributions among them. A highly nonlinear dependence on one such (highly correlated) variable can be well approximated by a combination of functions of several of them, and/or by interactions among them.

In the context of MARS modeling one strategy to cope with this (added) problem is to fit a sequence of models with increasing maximum interaction order (mi). One first fits an additive model ($mi = 1$), then one that permits at most two variable interactions ($mi = 2$), and so on. The models in this sequence can then be compared by means of their respective optimizing GCV scores. The one with the lowest mi value that gives a (relatively) acceptable fit can then be chosen.

Another (complementary) strategy is to directly resolve the ambiguity by enforcing parsimony on the number of variables that enter the model (Friedman and Silverman, 1989). This will discourage spurious interaction effects caused by collinearity (or concurvity) and, in addition, partially stabilize the function estimates. It will also aid in interpretation in that the resulting approximation will involve fewer variables.

Variable parsimony can be accomplished by introducing a small incremental penalty to the lack-of-fit criterion for choosing factors (knots) that involve introducing a new variable (not already in the model) as part of the forward stepwise procedure (Algorithm 2, line 7):

$$LOF(g) \leftarrow LOF(g)\left[1 + \gamma I\left(v \notin \bigcup_{m=1}^{M-1} \{v(k,m)\}_1^{K_m}\right)\right]. \tag{74}$$

At the Mth iteration there are $M-1$ basis functions currently in the model and the indicator function in (74) will be zero if the vth variable appears in a least one of them; otherwise it will be equal to one. The parameter γ (> 0) regulates the strength of the penalty for entering new variables and can be used to control the lack-of-fit/variable parsimony tradeoff. Note that this penalty (74) is only introduced as part of the (forward stepwise) knot selection process and it is not used to reflect the overall lack-of-fit of the model.

In highly collinear settings the (unmodified) lack-of-fit criterion ($LOF(g)$) has very little preference on which particular variable to enter from a highly correlated set, and a small value for γ will cause the modified criterion (74) to repeatedly enter the same one from that set, without seriously degrading the quality of the approximation. A good value for γ depends on the particular situation (degree of collinearity) and how much goodness-of-fit the user is willing to sacrifice for variable parsimony. This can be judged by examining the resulting fit quality for several (increasing) values of γ as reflected by either the final GCV score (30) or a sample reuse method.

We illustrate these two approaches on data taken from the Places Rated Almanac (Boyer and Savageau, 1986). They rated 329 American cities on the nine criteria listed in Table 12. For this exercise we attempt to model housing cost ($y = x_2$) on the other eight criteria. Table 13 shows the resulting number of variables and GCV estimate (30) of the PSE (59) for running MARS with different values of γ (74). The first three rows are for additive modeling ($mi = 1$) and the second three for models with two variable interactions permitted ($mi = 2$). The models involving interactions are seen to not be distinctly superior to the additive ones, so that using the first strategy (above) one would be inclined to choose the latter. As the value of γ (74) is steadily increased, MARS produces models with progressively fewer variables, as one would expect. For these particular data, however, one is able to reduce the number of variables from (nearly) the full set (at $\gamma = 0$) to only three ($.05 \leq \gamma < .15$) without seriously degrading the quality of the fit as estimated by the solution GCV score. Note that this GCV score (30) does not reflect the additional penalty imposed by setting $\gamma > 0$, so that differences between scores involving larger and smaller values of γ underestimate (on average) their actual differences. Ordinary cross-validation (CV) does account for this increased penalty. For example, the CV estimate (10 replications) for the $\gamma = 0$ ($mi = 1$) model is 0.56 whereas the corresonding score for $\gamma = 0.1$ is 0.52.

Figure 10 shows the graphical ANOVA decomposition for the three variable additive model produced for $0.05 \leq \gamma < .15$. From this analysis it appears that average (increasing) housing costs are most strongly affected by increasingly good climate (especially for the highest values) and are associated to a somewhat lesser degree with economic conditions and access to the arts.

The dependence on climate might be somewhat surprising since in these data housing costs reflect utility bills, which are likely to decrease with good climate, as well as taxes and mortgage payments. Any interpretations, however, must be tempered by the existence of the collinearities present in the design and the fact that the model is estimated to account for only 50% of the variance of the response.

5.4. Robustness

Since the MARS method as described here uses a model selection criterion based on squared-error loss it is not robust against outlying response values. There is nothing fundamental about squared-error loss in the MARS approach. Any criterion can be used to select the multivariate spline basis functions, and construct the final fit, by simply replacing the internal linear least squares fitting routine (LOF – Algorithm 2, line 7, and Algorithm 3, lines 2 and 5) by one that minimizes another loss criterion (given the current set of multivariate spline basis functions). Using robust/resistant linear regression methods would provide resistance to outliers. The only advantage to squared-error loss in the MARS context is computational. It is difficult to see how rapid updating formulae (Section 3.9) could be developed for other types of linear regression.

Gross outliers (in both the response and covariates) that can be detected through a preliminary (exploratory) analysis of the data, should be considered for removal before applying MARS. The MARS procedure is less sensitive than linear regression to covariate outliers owing to the local nature of the fit; sample covariate vectors far from an evaluation point tend to have less rather than more influence on the model estimate. Covariate outliers can have a strong influence on the fit near the corresponding data boundaries. This can be quite helpful if the corresponding response values for the outliers are correctly measured. If not, these outliers will contribute to end effect errors.

Recursive partitioning responds to outlying response values by trying to isolate them. It produces a series of splits so as to place each such outlier in its own region. This localizes the effect of the outlier(s) so that they only distort the fit for covariate values close to that of the outlier(s). The MARS procedure operates similarly. It will also try to isolate outliers through a series of corresponding "splits" producing basis functions that attempt to capture the (apparent) high curvature of the function near each outlier. The outliers will tend to heavily influence the values of the coefficients of their corresponding basis functions, but have much less influence on the rest of the fit. The particular basis functions introduced in this manner by outlying response values, may tend to involve interactions of high order depending on their location in the covariate space. Thus, interpreting the presence of interaction effects can be highly distorted by the existence of outlying response values.

Computationally feasible methods of robustifying the MARS procedure are currently under investigation.

6.0. Conclusion

The aim of the MARS procedure is to combine recursive partitioning and spline fitting in a

way that best retains the positive aspects of both, while being less vulnerable to their unfavorable properties. This has been accomplished, at least to some extent. The greatest strength of recursive partitioning is its adaptability, through its *local* variable subset selection strategy. This makes it a highly dynamic computation (Section 2.4) capable of tracking the dependencies associated with a wide variety of complex functional forms. The two weaknesses of recursive partioning are the lack of continuity of its models, and its inability to capture simple relationships such as linear, additive, or interactions of low order compared to n. Nonadaptive (tensor product) spline fitting produces continuous models with continuous derivatives. It strongly suffers, however, from the "curse-of-dimensionality" in that very large basis function sets are usually required in high dimensions to capture relatively simple functional relationships.

The MARS procedure completely retains the adaptability of recursive partitioning by its close adherence to the recursive splitting paradigm (compare Algorithms 1 and 2). It is in fact much more adaptive because it permits the recursive "splitting" of all basis functions (nodes) in the model and not just those that are currently terminal. This causes it to overcome the second problem (mentioned in the previous paragraph) associated with recursive partitioning. It produces continuous models by replacing the step functions (19) (20) by $q = 1$ truncated power spline basis functions (21) (22). Continuous derivatives are obtained through the strategy outlined in Section 3.7. From the point of view of tensor product spline methods, MARS can be viewed as a hierarchical forward/backward stepwise subset selection procedure for choosing a subbasis appropriate for the problem at hand, from the complete ($q = 1$) truncated power tensor product basis with knots at every (distinct) marginal data value. MARS models have a fair degree of interpretability through the ANOVA decomposition (Section 3.5) that breaks up the approximation into an additive component and into interaction contributions of various orders. Slicing (Section 4.7) can be used to explore the higher dimensional aspects of the models

The implementation of the adaptive regression spline strategy presented here represents a "first attempt" in that particular choices have been made concerning many of the "engineering details" in the absence of a great deal of experience with the procedure. Although incidental to the fundamental ideas, these details can have a strong bearing on performance. As experience is gained it is likely that many of the choices taken here will be seen to be less than optimal and suitable modifications will emerge that improve the performance of the procedure. The attempt here has been to demonstrate that the adaptive regression spline strategy, first introduced by Smith (1982) (in the univariate setting), holds substantial promise as a tool for multivariate function estimation.

A FORTRAN program implementing the MARS procedure is available from the author.

Acknowledgements

Helpful discussions with Terry Therneau are gratefully acknowledged. The author is grateful to Kishore Singhal for providing the semiconductor test data used in Section 4.7.

Bibliography

Andrews, D. F. and Herzberg, A. M. (1985). *Data. A Collection of Problems from Many Fields for the Student and Research Worker.* Springer-Verlag, New York.

Barry, D. (1986). Nonparametric Bayesian regression. *Ann. Statist.*, **14**, 934–953.

Bellman, R. E. (1961). *Adaptive Control Processes.* Princeton University Press, Princeton, New Jersey.

Boyer, R. and Savageau, D. (1986). *Places Rated Almanac.* Rand McNally (order number 0-528-88008-x).

Bozzini, M. and Lenarduzzi, L. (1985). Local smoothing for scattered noisy data. *International Series of Numerical Mathematics* **75**, Birkhauser Verlag, Basel, 51–60.

Breiman, L. and Friedman, J. H. (1985). Estimating optimal transformations for multiple regression and correlation (with discussion). *J. Amer. Statist. Assoc.* **80**, 580–619.

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

Breiman, L. and Meisel, W. S. (1976). General estimates of the intrinsic variability of data in nonlinear regression models. *J. Amer. Statist. Assoc.* **71**, 301–307.

Breiman, L. and Peters, S. (1988). Comparing automatic bivariate smoothers (A public service enterprise). Department of Statistics, University of California, Berkeley, Technical Report No. 161, June.

Brouncker, Lord (1734). Experiments of the recoiling of guns. *In the History of the Royal Society of London* (fourth edition). MDCCXXXIV by Thomas Sprat, 233–239.

Buja, A., Hastie, T. and Tibshirani, R. (1989). Linear smoothers and additive models (with discussion). *Ann. Statist.*, **17**, 453–555.

Chen, Z., Gu, C., and Wahba, G. (1989). Comment: "Linear smoothers and additive models," by Buja, Hastie and Tibshirani, *Ann. Statist.*, **17**, 515–521.

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatter plots. *J. Amer. Statist. Assoc.*, **74**, 828–836.

Cleveland, W. S. and Devlin, S. J. (1988). Locally weighted regression: an approach to regression analysis by local fitting. *J. Amer. Statist. Assoc.* **83**, 596–610.

Copas, J. B. (1983). Regression prediction and shrinkage (with discussion), *J. Roy. Statist. Soc. B*, **45**, 311–354.

Cox, D. R. (1970). *Analysis of Binary Data*, London: Chapman and Hall.

Craven, P. and Wahba, G. (1979). Smoothing noisy data with spline functions. Estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerische Mathematik* **31**, 317–403.

deBoor, C. (1978). *A Practical Guide to Splines.* Springer-Verlag, New York, NY.

Diaconis, P. and Shahshahani, M. (1984). On non-linear functions of linear combinations. *SIAM J. Sci. Stat. Comput.* **5**, 175–191.

Dongarra, J. J., Moler, C. B., Bunch, J. R., and Stewart, G. W. (1979). *Linpack Users' Guide.* SIAM, Philadelphia, PA.

Donoho, D. L. and Johnstone, I. (1989). Projection-based approximation, and a duality with kernel methods. *Ann. Statist.* **17**, 58–106.

Efron, B. (1983). Estimating the error rate of a prediction rule: Improvement on cross-validation. *J. Amer. Statist. Assoc.* **78**, 316–331.

Forina, M., Armanino, C., Lanteri, S., Calcagno, C., and Tiscornia, E. (1983). Evaluation of the chemical characteristics of olive oils as a function of production year by multivariate methods. *La Revista Italiana delle Sostanze Grasse*, **60**, Oct.

Friedman, J. H. (1979). A tree-structured approach to nonparametric multiple regression. In *Smoothing Techniques for Curve Estimation*, T. H. Gasser and M. Rosenblatt (eds.), Springer-Verlag, New York, 5–22.

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

Friedman, J. H. (1988). Fitting functions to noisy data in high dimensions. In *Computer Science and Statistics*: Proceedings of the 20th Symposium (E. Wegman, D. Gantz, and J. Miller, eds.). Amer. Statist. Assoc., Washington, D.C., 13–43.

Friedman, J. H., Grosse, E., and Stuetzle, W. (1983). Multidimensional additive spline approximation. *SIAM J. Sci. Stat. Comput.*, **4**, 291–301.

Friedman, J. H. and Silverman, B. W. (1989). Flexible parsimonious smoothing and additive modeling. *Technometrics,* **31**, 3–39.

Friedman, J. H. and Stuetzle, W. (1981). Projection pursuit regression, *J. Amer. Statist. Assoc.* **76**, 817–823.

Friedman, J. H. and Wright, M. J. (1981). A nested partitioning algorithm for numerical multiple integration. *ACM Trans. Math. Software*, March.

Golub, G. H. and Van Loan, C. F. (1983). *Matrix Computations.* The Johns Hopkins University Press, Baltimore, MD.

Gu, C., Bates, D. M., Chen, Z., and Wahba, G. (1990). The computation of GCV function through Householder tridiagonalization with application to the fitting of interaction spline models. *SIAM J. Matrix Analysis* **10**, 457–480.

Gu, C., and Wahba, G. (1988). Minimizing GCV/GML scores with multiple smoothing parameters via the Newton method. Univ. of Wisconsin, Dept. of Statistics, Tech. Report 847. *SIAM J. Sci. Statist. Comput.* 1990 (to appear).

Hastie, T. and Tibshirani, R. (1986). Generalized additive models (with discussion), *Statist. Science* **1**, 297–318.

Huber, P. J. (1985). Projection Pursuit (with discussion), *Ann. Statist.* **13**, 435–475.

Lyness, J. N. (1970). Algorithm 379-SQUANK (Simson Quadrature Used Adaptively – Noise Killed), *Comm. Assoc. Comp. Mach.* **13**, 260–263.

Morgan, J. N., and Sonquist, J. A. (1963). Problems in the analysis of survey data, and a proposal. *J. Amer. Statist. Assoc.* **58**, 415–434.

Parzen, E. (1962). On estimation of a probability density function and mode. *Ann. Math. Statist.* **33**, 1065–1076.

Shepard, D. (1964). A two-dimensional interpolation function for irregularly spaced data. *Proc. 1964 ACM Nat. Conf.*, 517–524.

Shumaker, L. L. (1976). Fitting surfaces to scattered data. In *Approximation Theory III*, G. G. Lorentz, C. K. Chui, and L. L. Shumaker, eds. Academic Press, New York, 203–268.

Shumaker, L. L. (1984). On spaces of piecewise polynomials in two variables. In *Approximation Theory and Spline Functions*, S. P. Singh et al. (eds.). D. Reidel Publishing Co., 151–197.

Silverman, B. W. (1985). Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *J. Roy. Statist. Soc. B* **47**, 1–52.

Smith, P. L. (1982). Curve fitting and modeling with splines using statistical variable selection techniques. NASA, Langley Research Center, Hampton, VA, Report NASA 166034.

Stone, C. J. (1977). Nonparametric regression and its applications (with discussion). *Ann. Statist.* **5**, 595–645.

Stone, C. J. and Koo, Cha-Yong (1985). Additive splines in statistics. *Proceedings, Annual Meeting of Amer. Statist. Assoc., Statist. Comp. Section*, August, 45–48.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictors (with discussion). *J. R. Statist. Soc.*, **B36**, 111–147.

Wahba, G. (1986). Partial and interaction splines for the semiparametric estimation of functions of several variables. In *Computer Science and Statistics*: Proceedings of the 18th Symposium (T. Boardman, ed.) *Amer. Statist. Assoc.*, Washington, DC, 75–80.

Wahba, G. (1990). *Spline Models for Observational Data.* Monograph: SIAM, CBMS-NSF Regional Conference Series in Applied Mathematics, Vol. 59.

Table 1

Computation time in seconds for executing the MARS procedure on the alternating current impedance example of Section 4.4.1, as a function of the maximum size, $M_{\max}$, of the basis function set. Computations were performed on a SUN 3/260 with FPA. For this example $n = 4$ and $N = 200$. Execution times scale linearly with both these parameters. The quantity mi is the maximum number of variables that are allowed to interact.

$M_{\max}$ / mi	5	10	15	20	30	40	50
1	1.5	3.0	4.4	8.4	17.2	32.1	54.2
2	2.3	7.2	14.1	24.6	88.1	227.8	536.9
n	2.4	8.1	15.4	33.7	119.1	271.3	546.2

Table 2a

Results of applying MARS to pure noise in five dimensions ($n = 5$). Shown are the lower quantiles for the ratio of the MARS GCV score to that of a constant model $\hat{f}(\mathbf{x}) = \bar{y}$.

mi	1%	5%	10%	25%	50%
$N = 50$					
1	.79	.88	.93	.98	1.02
2	.79	.88	.91	.99	1.04
n	.81	.86	.91	.99	1.04
$N = 100$					
1	.88	.94	.96	.99	1.00
2	.89	.94	.96	.98	1.01
n	.89	.94	.95	.98	1.00
$N = 200$					
1	.97	.98	.99	1.00	1.01
2	.94	.97	.98	1.00	1.01
n	.94	.96	.98	1.00	1.01

Table 2b

Results of applying MARS to pure noise in ten dimensions ($n = 10$). Shown are the lower quantiles of the ratio of the MARS GCV score to that of a constant model $\hat{f}(\mathbf{x}) = \bar{y}$.

mi	1%	5%	10%	25%	50%
$N = 50$					
1	.77	.82	.87	.97	1.00
2	.71	.75	.86	.98	1.01
n	.64	.75	.86	.96	1.02
$N = 100$					
1	.93	.94	.96	.98	1.00
2	.84	.91	.92	.97	1.00
n	.84	.89	.92	.96	1.00
$N = 200$					
1	.94	.96	.98	1.00	1.00
2	.94	.95	.97	.99	1.01
n	.92	.94	.95	.98	1.00

Table 3

Comparison of the accuracy of MARS modeling, with interactions permitted ($mi = 2, 10$), to that of purely additive modeling ($mi = 1$) on additive data (Section 4.2).

mi	$\overline{ISE}$	$\overline{PSE}$	$\overline{GCV/PSE}$
$N = 50$			
1	.14 (.07)	.26 (.06)	1.16 (.32)
2	.15 (.08)	.27 (.07)	1.12 (.32)
10	.15 (.08)	.27 (.07)	1.10 (.32)
$N = 100$			
1	.053 (.027)	.18 (.02)	1.06 (.19)
2	.060 (.034)	.19 (.03)	1.02 (.20)
10	.063 (.043)	.19 (.04)	1.02 (.21)
$N = 200$			
1	.026 (.011)	.16 (.01)	1.08 (.16)
2	.033 (.021)	.17 (.02)	1.04 (.16)
10	.037 (.017)	.17 (.03)	1.02 (.16)

Table 4

Number of replications (out of 100) for which the optimizing GCV score for the additive model ($mi = 1$) is less than (or equal to) THR times that for models with interactions allowed ($mi = 2, 10$), on additive data (Section 4.2).

mi	$THR = 1.0$	$THR = 1.05$	$THR = 1.1$
$N = 50$			
2	86	93	97
10	84	91	95
$N = 100$			
2	83	95	98
10	81	94	99
$N = 200$			
2	70	94	100
10	72	94	100

Table 5a

Summary of the MARS model for the data of Section 4.3.

GCV (piecewise-linear) = .059

GCV (piecewise-cubic) = .055

total number of basis functions = 11

total effective number of parameters = 32.3

ANOVA decomposition:

Fun.	σ	$\backslash GCV$	# basis	# parms	variable(s)	
1	.56	.67	1	2.8	4	
2	.29	.087	1	2.8	2	
3	.50	.26	1	2.8	1	
4	.28	.21	1	2.8	5	
5	.30	.22	2	5.7	3	
6	.46	.37	4	11.3	1	2
7	.064	.059	1	2.8	2	6

Table 5b

Results of applying MARS to 100 data sets, at three sample sizes N, for the situation described in Section 4.3.

mi	$\overline{ISE}$	$\overline{PSE}$	$\overline{GCV/PSE}$
$N = 50$			
1	.16 (.08)	.20 (.08)	1.17 (.40)
2	.20 (.12)	.24 (.11)	1.23 (.63)
10	.20 (.12)	.23 (.11)	1.23 (.63)
$N = 100$			
1	.11 (.02)	.15 (.02)	1.07 (.27)
2	.035 (.025)	.074 (.024)	1.19 (.33)
10	.038 (.025)	.077 (.024)	1.16 (.30)
$N = 200$			
1	.097 (.009)	.13 (.009)	1.01 (.16)
2	.017 (.007)	.056 (.007)	1.09 (.21)
10	.018 (.008)	.058 (.008)	1.05 (.20)

Table 6a

Summary of the MARS model for the alternating current series circuit impedance Z.

GCV (piecewise - linear) = 0.21

GCV (piecewise - cubic) = 0.19

total number of basis functions = 9

total effective number of parameters = 31.0

ANOVA decomposition:

Fun.	σ	$\backslash GCV$	# basis	# parms	variable(s)	
1	.35	.25	2	6.7	ω	
2	.62	.44	1	3.3	R	
3	.53	.52	3	10.0	ω	C
4	.53	.40	2	6.7	ω	L
5	.17	.22	1	3.3	R	L

Table 6b

Results of applying MARS to 100 data sets, at each of three sample sizes N, for the alternating current impedance Z example.

mi	$\overline{ISE}$	$\overline{PSE}$	$\overline{GCV/PSE}$
$N = 100$			
1	.68 (.11)	.71 (.10)	.86 (.19)
2	.28 (.17)	.35 (.15)	.88 (.28)
4	.31 (.18)	.38 (.16)	.81 (.28)
$N = 200$			
1	.59 (.05)	.63 (.05)	.96 (.14)
2	.12 (.06)	.21 (.06)	1.01 (.21)
4	.12 (.07)	.21 (.06)	.96 (.20)
$N = 400$			
1	.56 (.04)	.60 (.04)	.98 (.11)
2	.067 (.015)	.16 (.01)	1.01 (.15)
4	.069 (.028)	.16 (.03)	.97 (.15)

Table 7a

Summary of the MARS model for the alternating current series circuit phase angle φ.

GCV (piecewise - linear) = .19
GCV (piecewise - cubic) = .22
total number of basis functions = 14
total effective number of parameters = 40.2

ANOVA decomposition:

Fun.	σ	$\backslash GCV$	# basis	# parms	variable(s)	
1	.45	.32	2	5.6	ω	
2	.86	.34	2	5.6	C	
3	.62	.38	2	5.6	L	
4	.42	.26	3	8.4	R	C
5	.23	.21	1	2.8	ω	L
6	.12	.19	1	2.8	L	C
7	.14	.19	1	2.8	ω	C
8	.28	.22	1	2.8	R	L
9	.24	.23	1	2.8	R	ω

Table 7b

Results of applying MARS to 100 data sets, at each of three sample sizes N, for the alternating current phase angle φ example.

mi	$\overline{ISE}$	$\overline{PSE}$	$\overline{GCV/PSE}$
$N = 100$			
1	.27 (.04)	.34 (.04)	.98 (.18)
2	.24 (.10)	.32 (.09)	.92 (.24)
4	.24 (.07)	.32 (.06)	.92 (.23)
$N = 200$			
1	.24 (.02)	.31 (.02)	1.01 (.11)
2	.16 (.03)	.25 (.02)	.95 (.15)
4	.16 (.03)	.25 (.03)	.90 (.16)
$N = 400$			
1	.22 (.01)	.30 (.01)	1.04 (.09)
2	.12 (.01)	.21 (.01)	1.00 (.13)
4	.12 (.02)	.21 (.02)	.94 (.13)

Table 8

Measured variables for the Portuguese olive oil data

1	C16:0 palmitic acid
2	C16:1 palmitoleic acid
3	C17:0 heptadecanoic acid
4	C17:1 heptadecenoic acid
5	C18:0 stearic acid
6	C18:1 oleic acid
7	C18:2 linoleic acid
8	C18:3 linolenic acid
9	C20:0 eicosanoic acid
10	C24:0 lignoceric acid
11	Beta-sitosterol
12	Campesterol
13	Stigmasterol

Table 9

Portuguese Olive Oil

method	# vars	GCV	CV	error (CV)
MARS ($mi = 1$) (least-squares)	3	.23	.21	.050
MARS ($mi = 2$) (least-squares)	3	.19	.20	.038
MARS ($mi = 1$) (logistic)	3	.16	.19	.036
MARS ($mi = 2$) (logistic)	3	.13	.16	.026
linear logistic (stepwise)	4	.25	.26	.070
CART	2	–	.22	.058

Table 10

Summary of the ($mi = 2$) logistic MARS model for the Portuguese olive oil example.

piecewise-linear: $GCV = .10$ $CV = .15$
piecewise-cubic: $GCV = .13$ $CV = .15$

total number of basis functions = 9
total effective number of parameters = 29.5

ANOVA decomposition:

Fun	\ GCV	# basis	# parms	variable(s)	
1	.60	3	9.5	2	
2	.26	1	3.2	6	
3	.27	2	6.3	2	12
4	.20	3	9.5	6	12

Table 11

MARS on semiconductor data, piecewise-linear fits – see Section 4.7.

	GCV	cross-validation (10-reps)			
			residual distribution		
mi	$\sqrt{e^2}$	$\sqrt{e^2}$	med.	.75	max.
1	.458	.508	.159	.316	18.9
2	.041	.211	.012	.032	7.6
3	.016	.118	.010	.025	4.6
4	.0096	.090	.0078	.017	2.9

Table 12

The nine criteria used to rate U.S. cities by the *Places Rated Almanac*.

1 climate
2 housing costs
3 health care and environment
4 crime rate
5 transportation
6 education
7 access to the arts
8 recreational opportunities
9 economics

Table 13

The number of variables and corresponding GCV score for a sequence of MARS models on the *Places Rated* data, produced by increasing the penalty γ (74) for adding variables.

$mi = 1$							
γ	0	.01	.02	.03	.05	.10	.15
# vars	7	7	7	4	3	3	2
GCV	.49	.49	.49	.50	.51	.51	.58
$mi = 2$							
γ	0	.01	.02	.05	.07	.10	.15
# vars	8	6	5	5	4	3	2
GCV	.48	.50	.48	.47	.51	.51	.58

Figure Captions

1. A binary tree representing a recursive partitioning regression model with the associated basis functions.

2a. Comparison of $q = 1$ truncated power spline functions and the corresponding continuous derivative truncated cubics, with central knot $t = 0.5$ and side knots at $t_- = 0.2$ and $t_+ = 0.7$.

2b. Illustration of side knot placement for a one-dimensional ANOVA function comprised of three basis functions (upper frame), and a two-dimensional ANOVA function with two basis functions (lower frame).

3. ANOVA functions for the MARS model of Example 4.3.

4a. Schematic diagram of the alternating current series circuit for Examples 4.4.

4b. ANOVA functions of the MARS model for the alternating current series circuit impedance Z, Section 4.4.1.

4c. ANOVA functions of the MARS model for the alternating current series circuit phase angle φ, Section 4.4.2.

5. ANOVA functions for the log-odds MARS model on the Portuguese olive oil data, Section 4.5.

6. Experimental design (upper left) and MARS surface smooth for the gun recoil data, first example, Section 4.6.

7a. True underlying function (eqn. 66) for the second example, Section 4.6.

7b. MARS surface smooth for second example, Section 4.6.

8a. MARS surface smooth for the second example, Section 4.6, in the presence of eight highly structured nuisance variables.

8b. ANOVA functions for the ten variable version of the second example in Section 4.6.

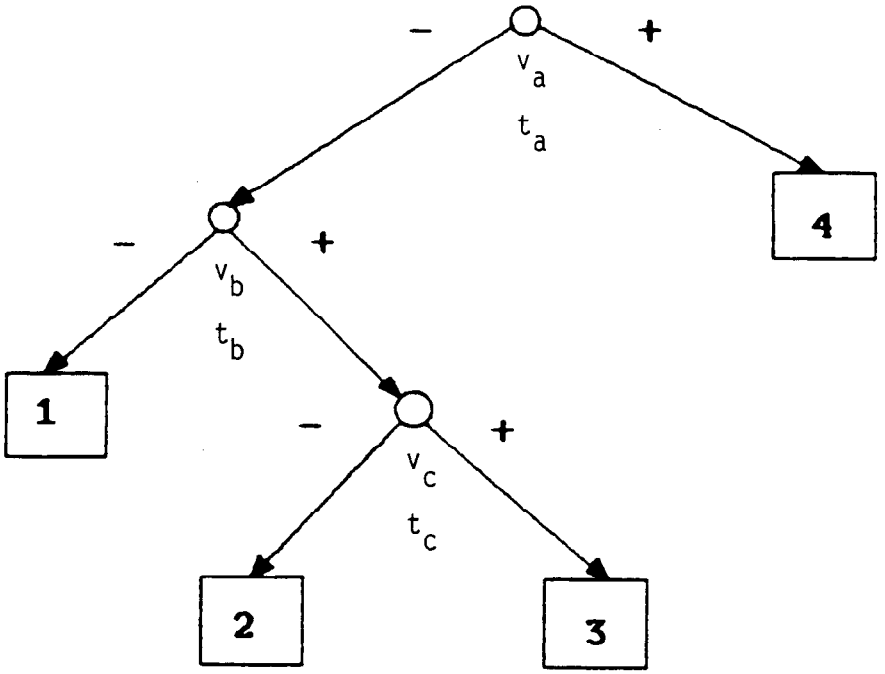
9a. MARS model for semiconductor component data restricted to two-variable interactions ($mi = 2$).

9b. Full ($mi = 4$) MARS model for semiconductor component data. Functions of V_1 and V_2 along various slices defined by V_3 and V_4.

9c. Full ($mi = 4$) MARS model for semiconductor component data. Functions of V_3 and V_4 along several slices defined by V_1 and V_2.

10. Graphical ANOVA decmposition of the three variable additive MARS model on the *Places Rated* data, Section 5.3.

Figure 1



$$B_1 = H[-(x_{v_a} - t_a)]H[-(x_{v_b} - t_b)]$$

$$B_2 = H[-(x_{v_a} - t_a)]H[+(x_{v_b} - t_b)]H[-(x_{v_c} - t_c)]$$

$$B_3 = H[-(x_{v_a} - t_a)]H[+(x_{v_b} - t_b)]H[+(x_{v_c} - t_c)]$$

$$B_4 = H[+(x_{v_a} - t_a)]$$

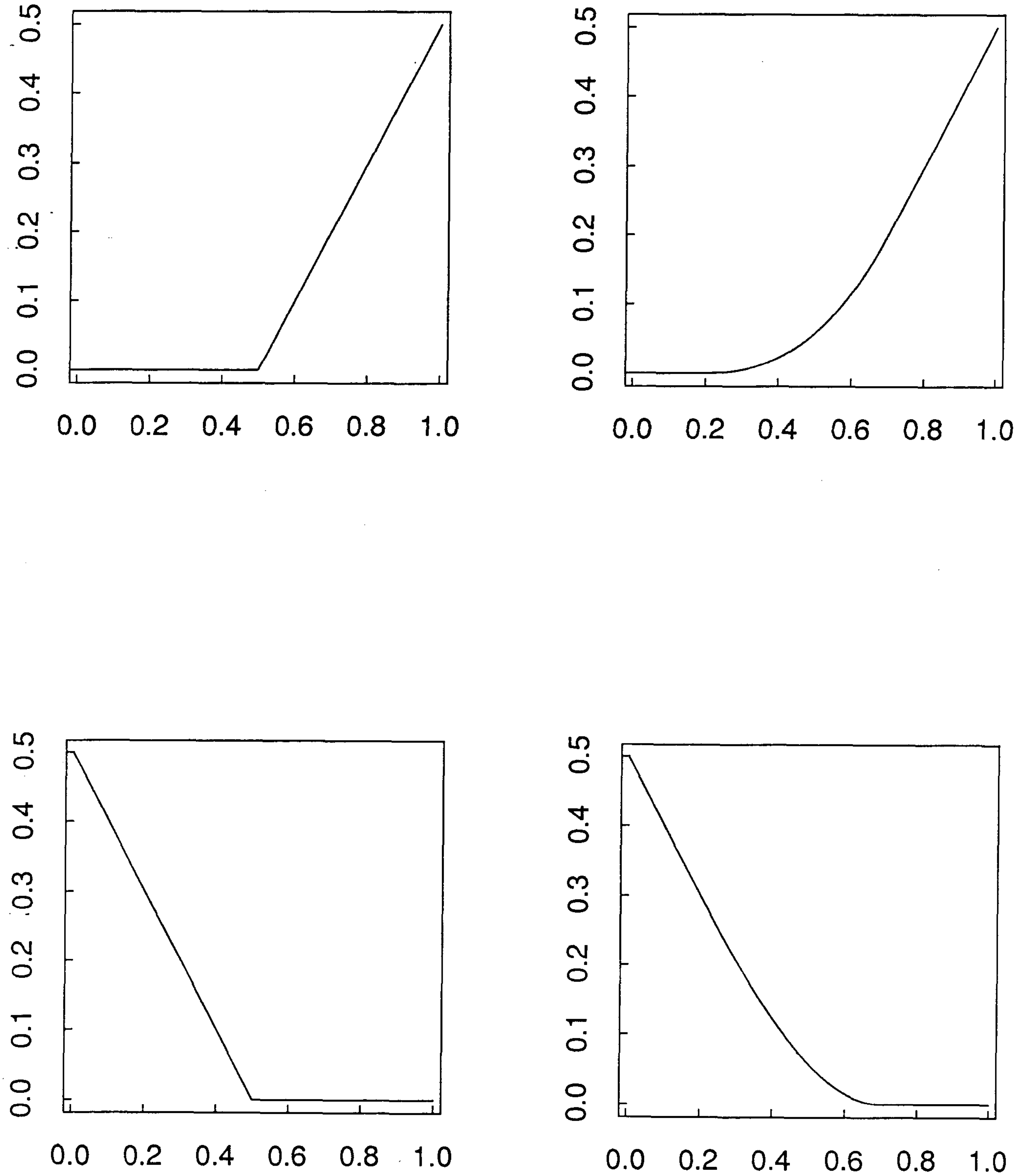
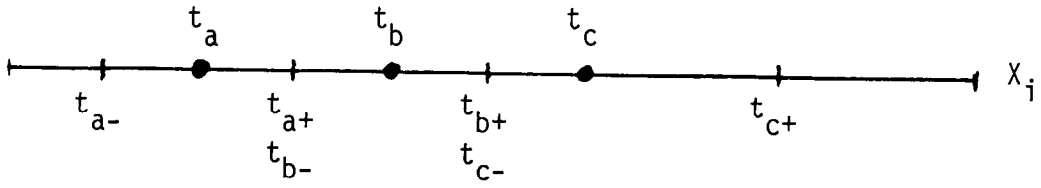


Figure 2a



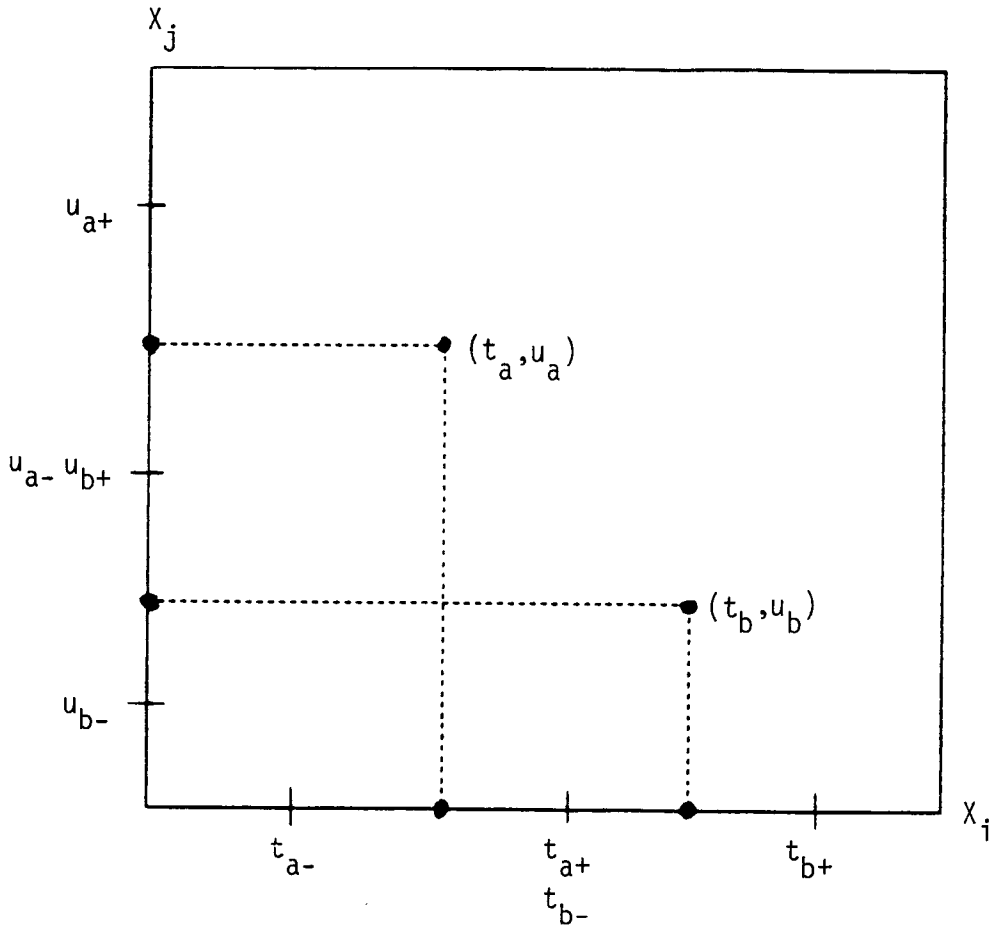
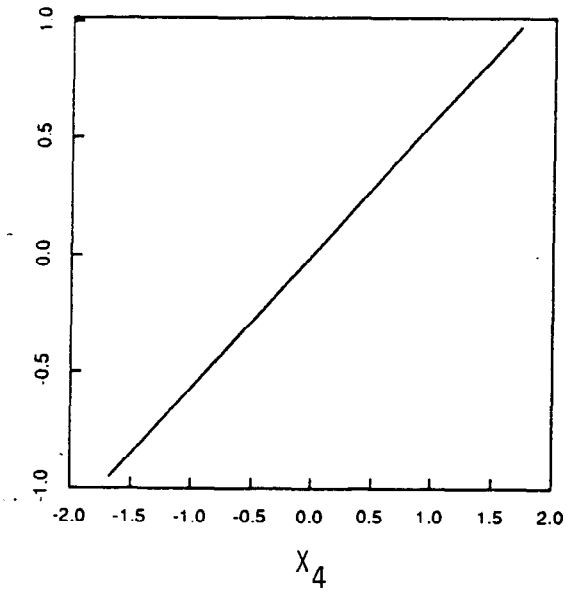
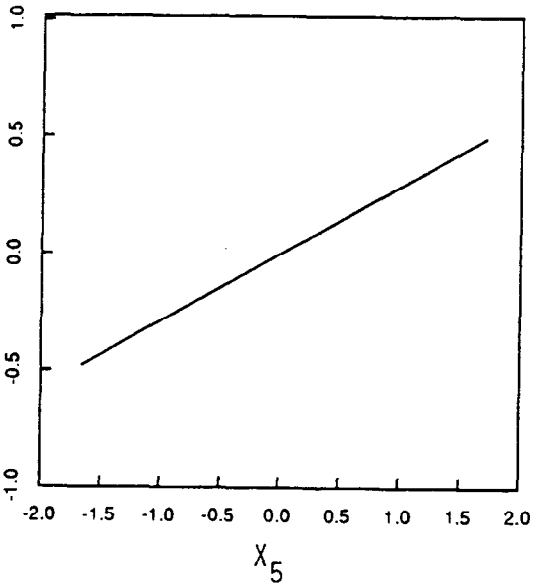
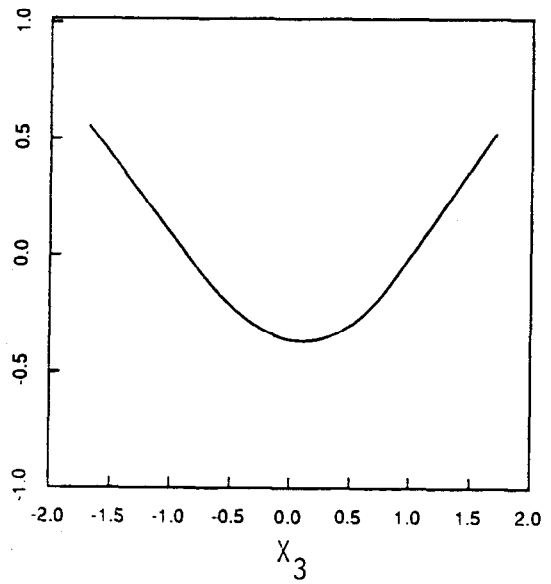
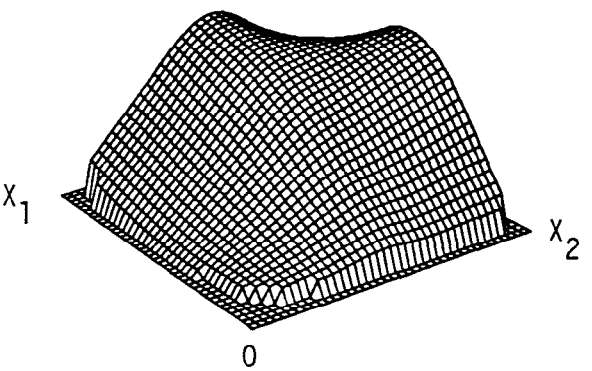


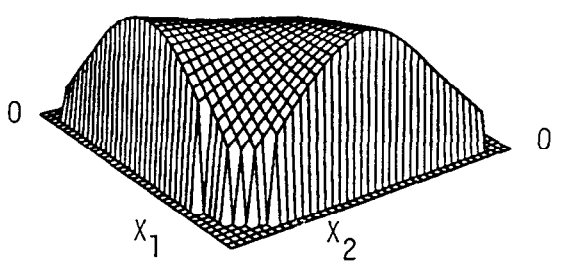
Figure 2b


1.0
0.5
0.0
-0.5
-1.0
-2.0 -1.5 -1.0 -0.5 0.0 0.5 1.0 1.5 2.0
X4


1.0
0.5
0.0
-0.5
-1.0
-2.0 -1.5 -1.0 -0.5 0.0 0.5 1.0 1.5 2.0
X5


1.0
0.5
0.0
-0.5
-1.0
-2.0 -1.5 -1.0 -0.5 0.0 0.5 1.0 1.5 2.0
X3


X1
X2
0


0
0
X1
X2

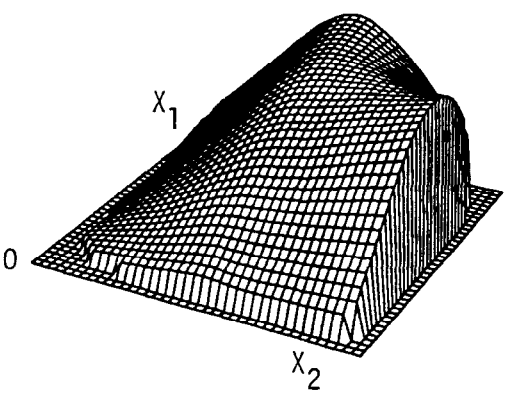

X1
0
X2

Figure 3

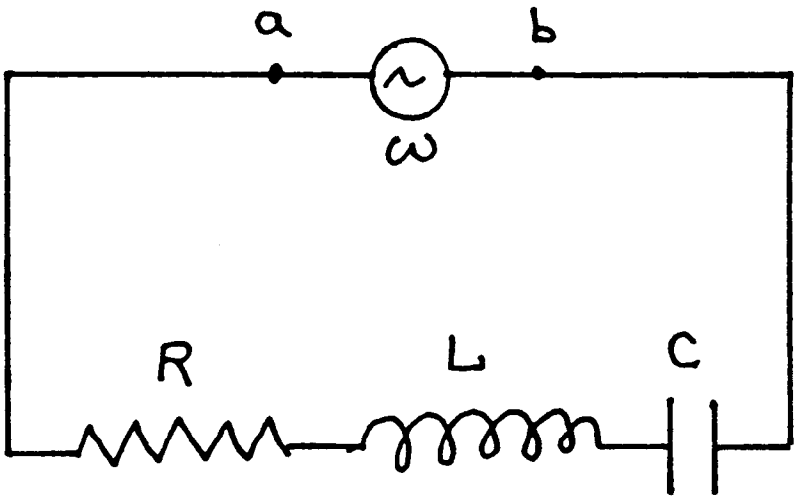
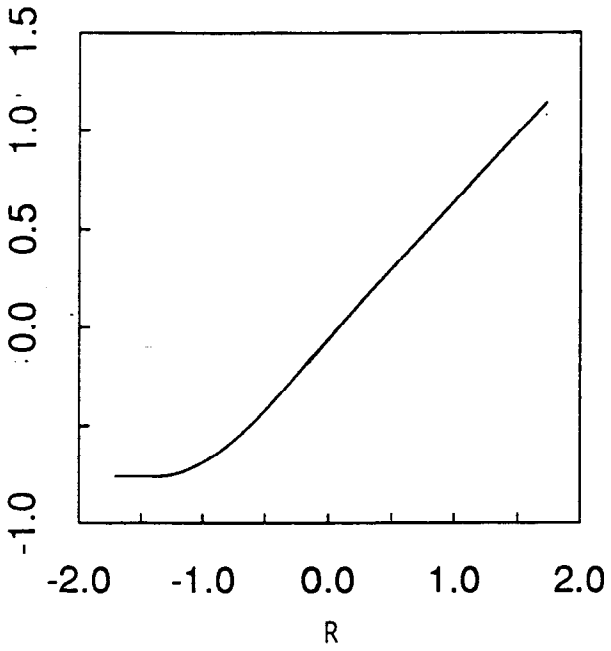
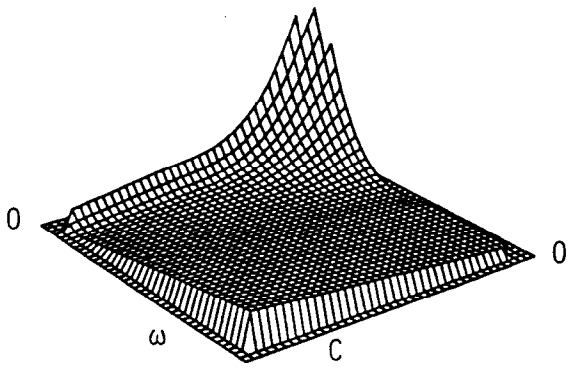
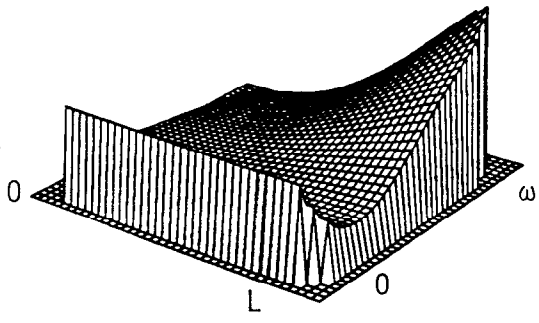


Figure 4a







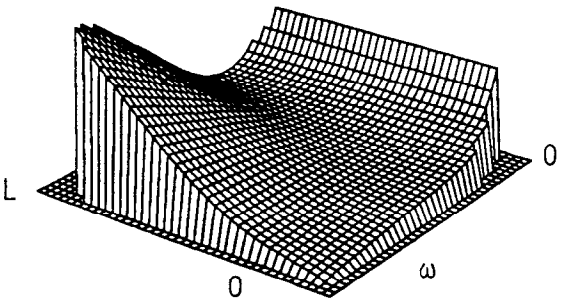


Figure 4b

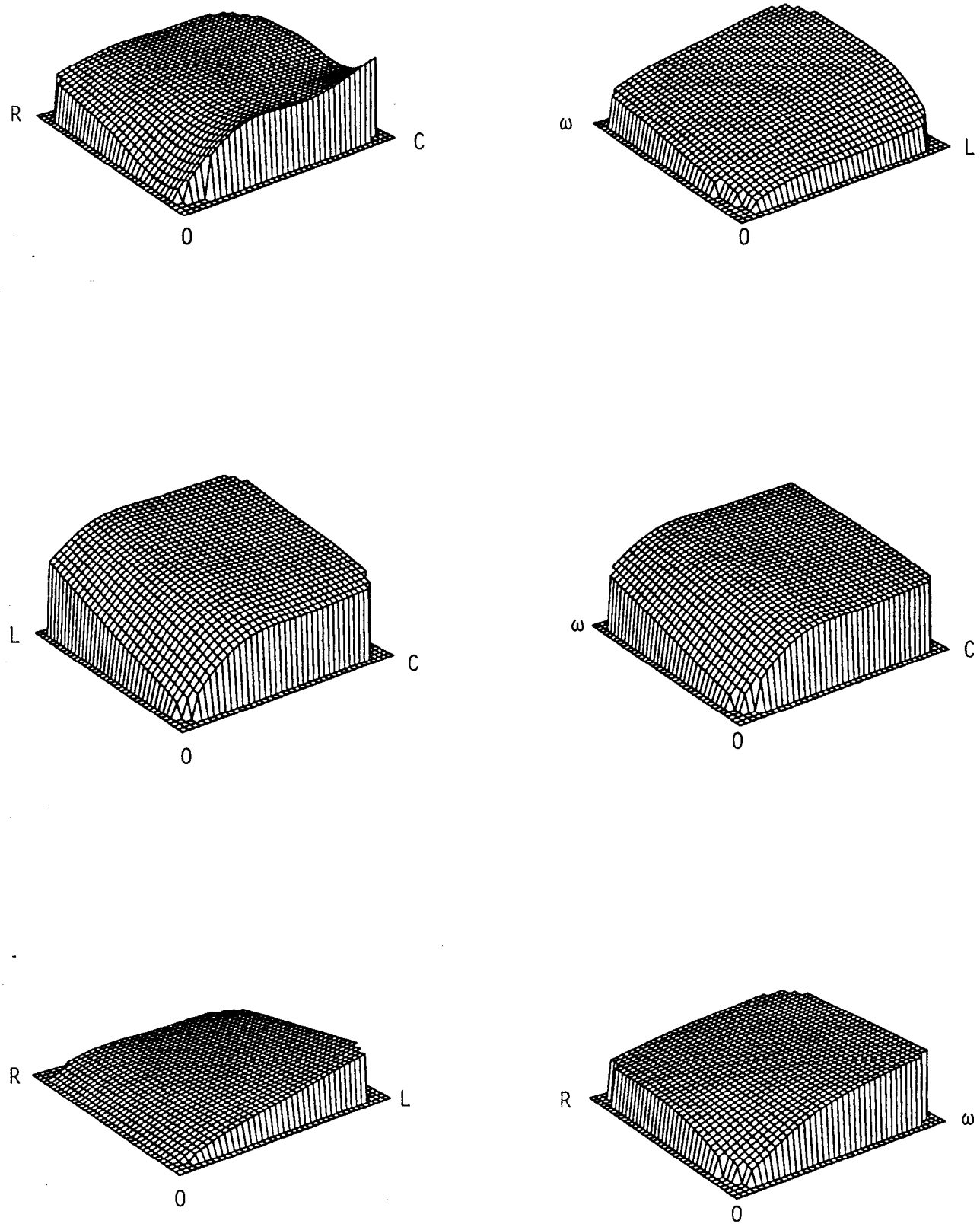
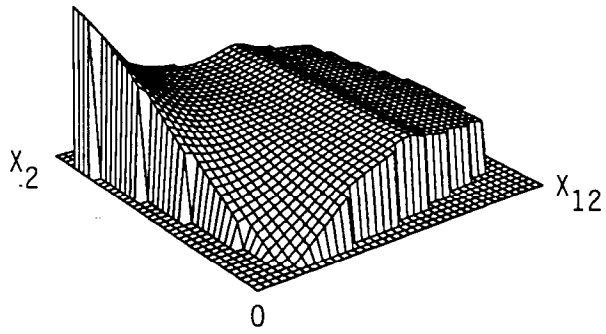
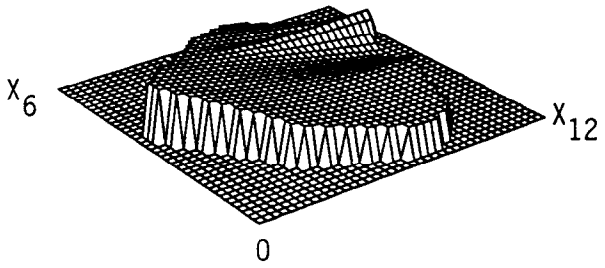
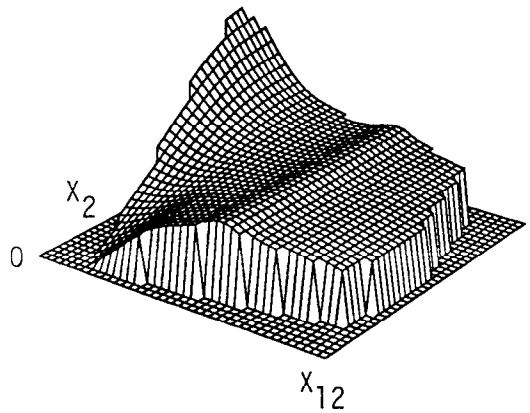


Figure 4c







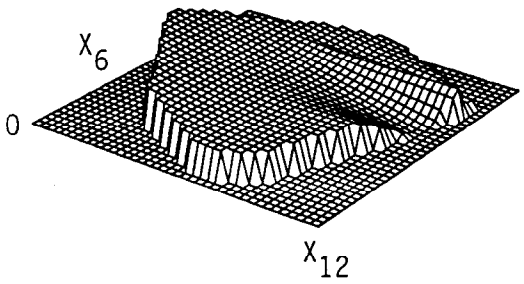
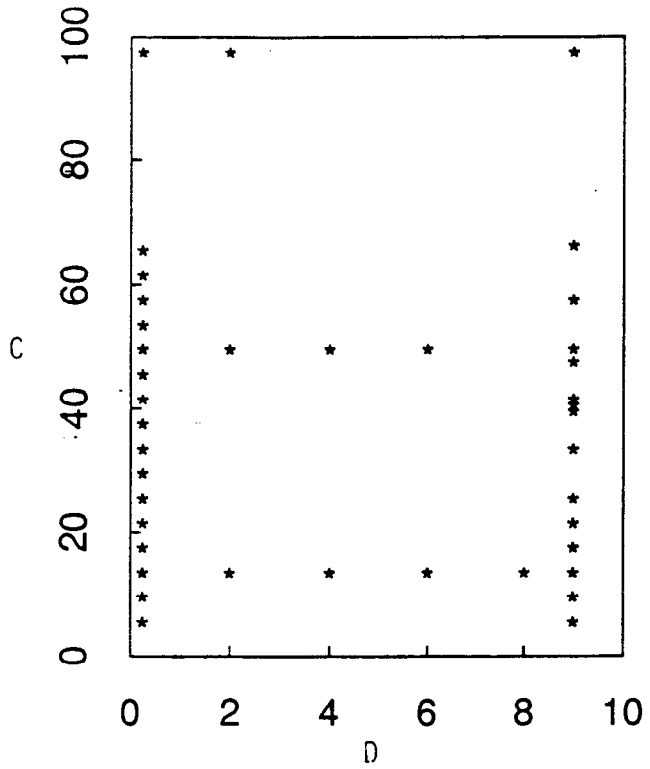
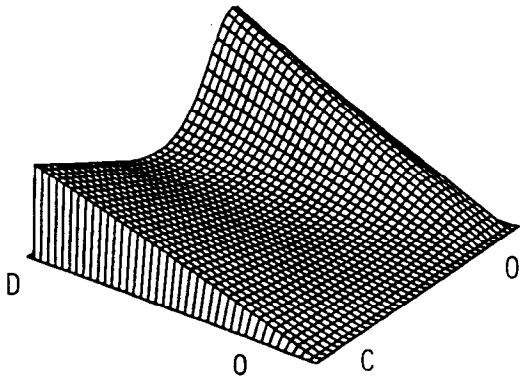


Figure 5





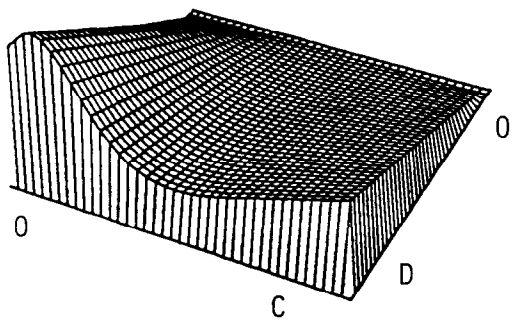


Figure 6

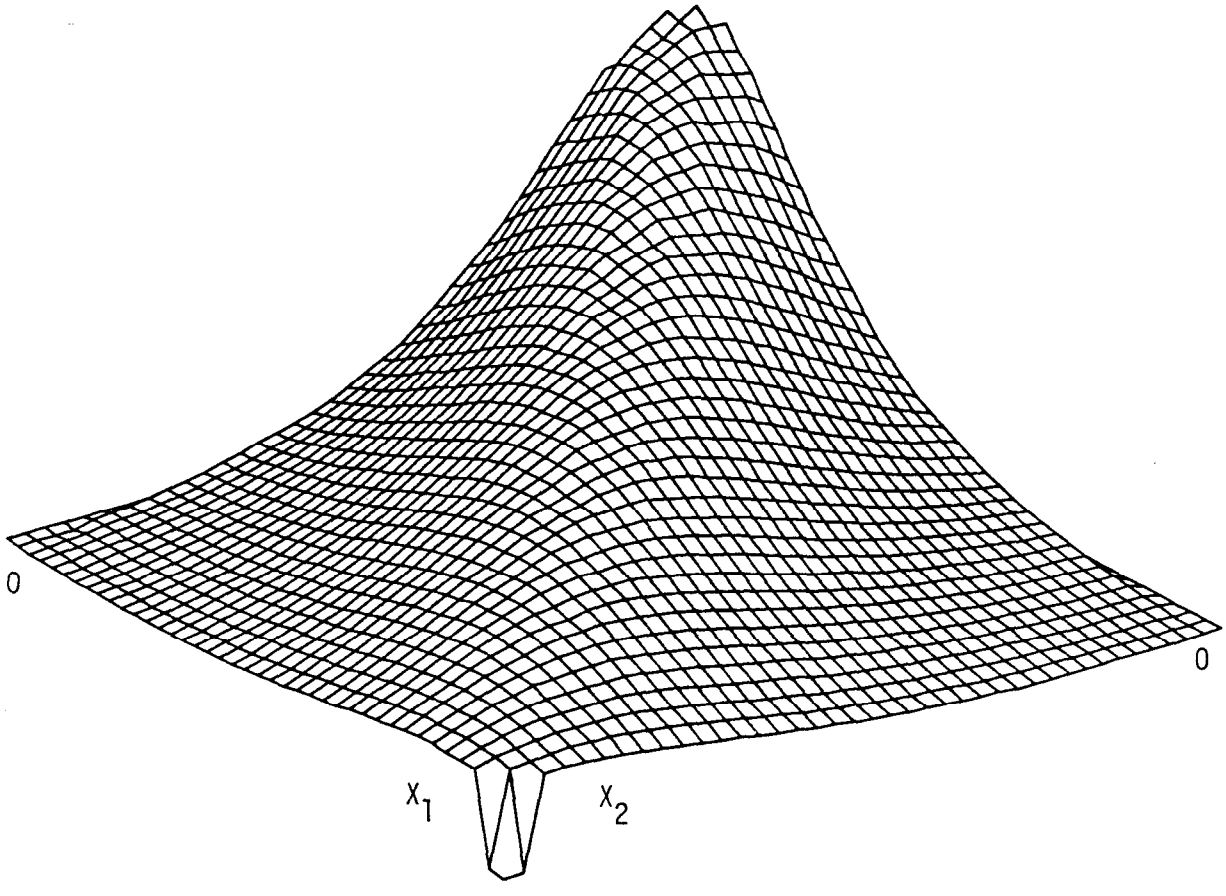


Figure 7a

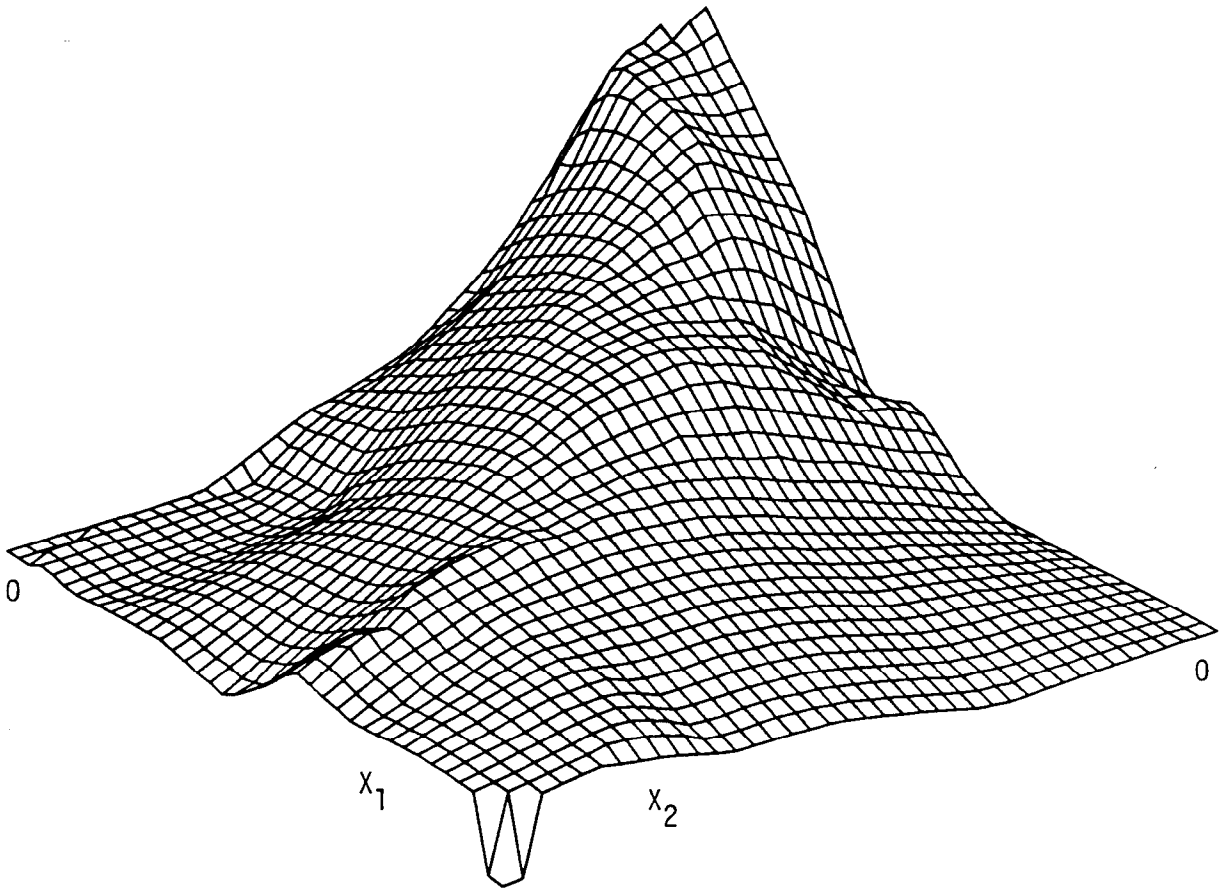


Figure 7b

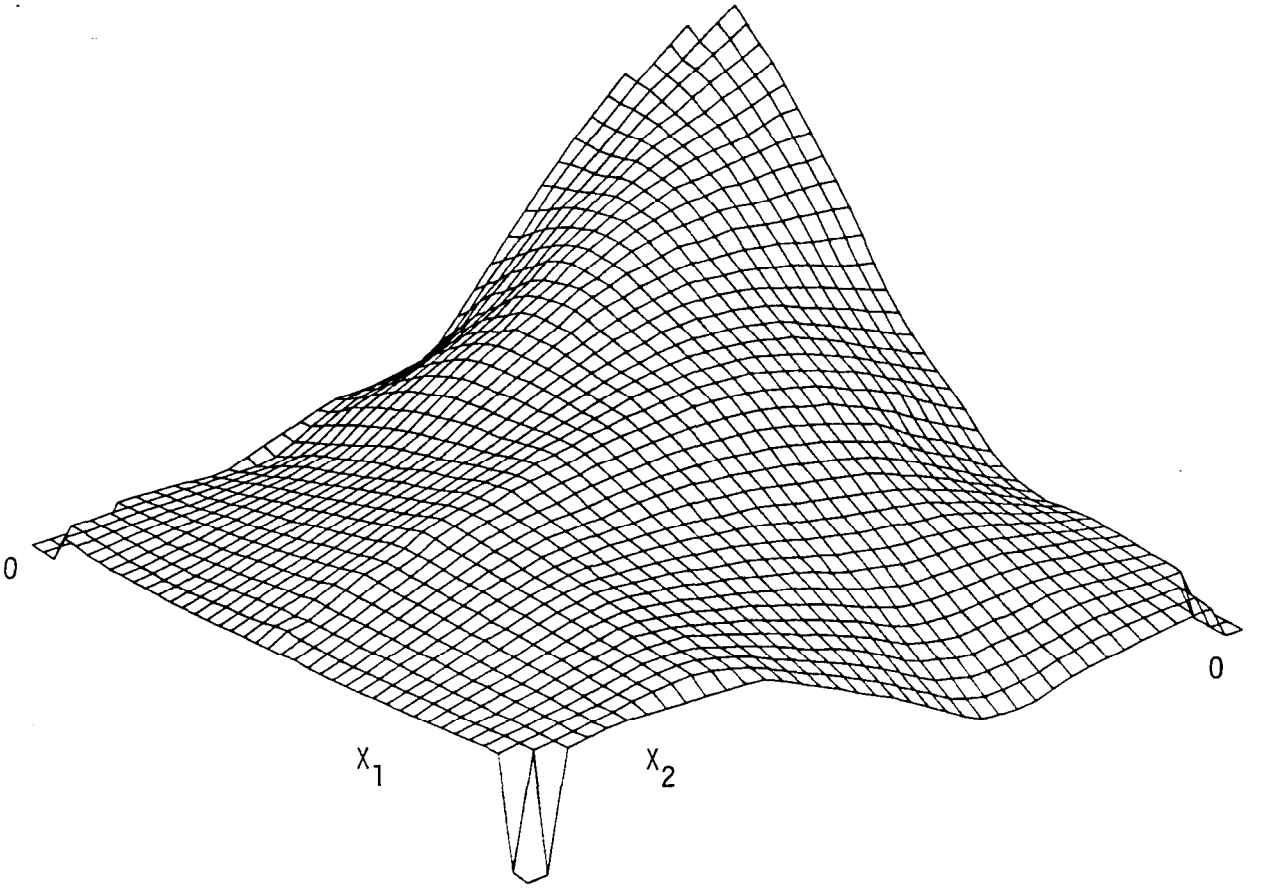
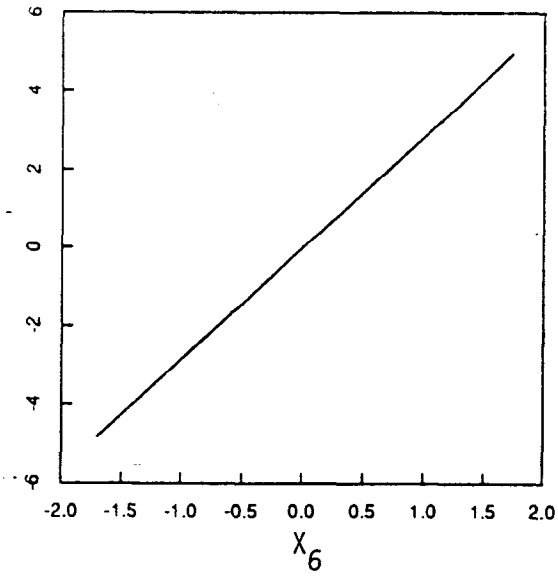
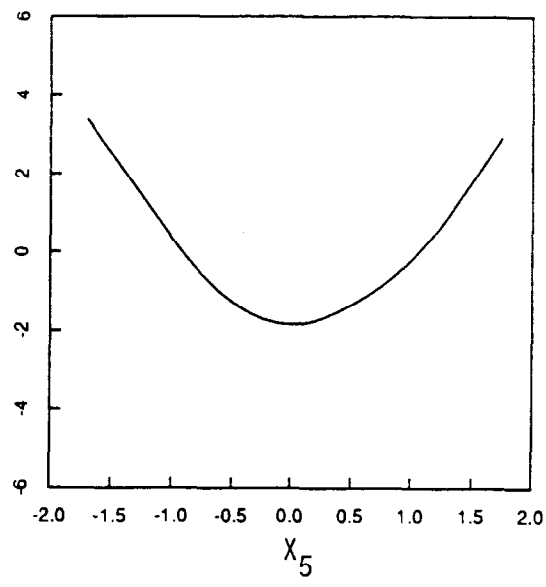
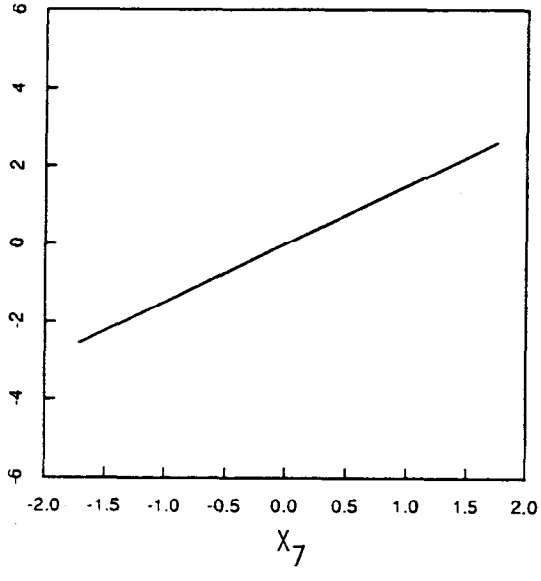
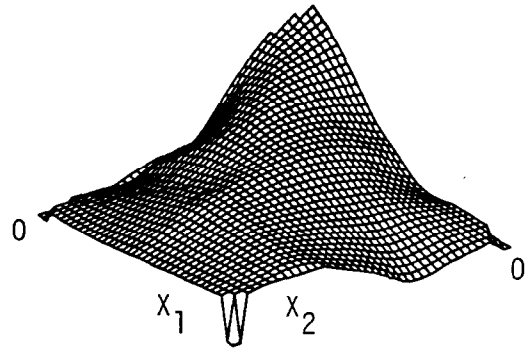


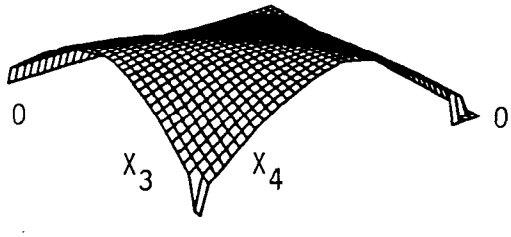
Figure 8a


6
4
2
0
-2
-4
-6
-2.0 -1.5 -1.0 -0.5 0.0 0.5 1.0 1.5 2.0
X6


6
4
2
0
-2
-4
-6
-2.0 -1.5 -1.0 -0.5 0.0 0.5 1.0 1.5 2.0
X5


6
4
2
0
-2
-4
-6
-2.0 -1.5 -1.0 -0.5 0.0 0.5 1.0 1.5 2.0
X7


0
0
X1
X2


0
0
X3
X4

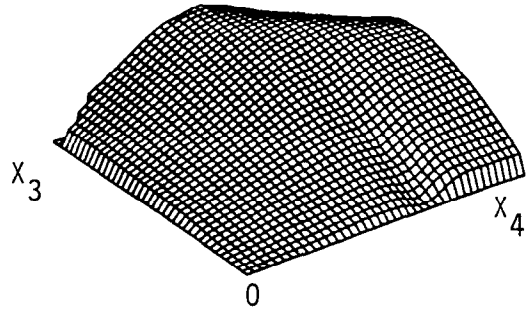
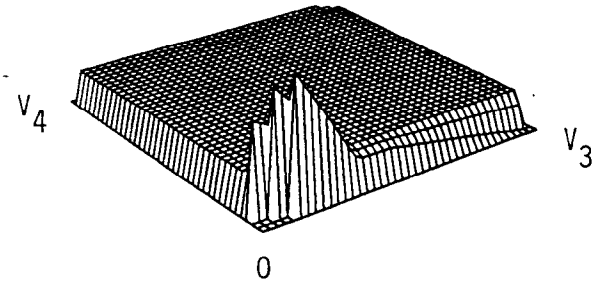
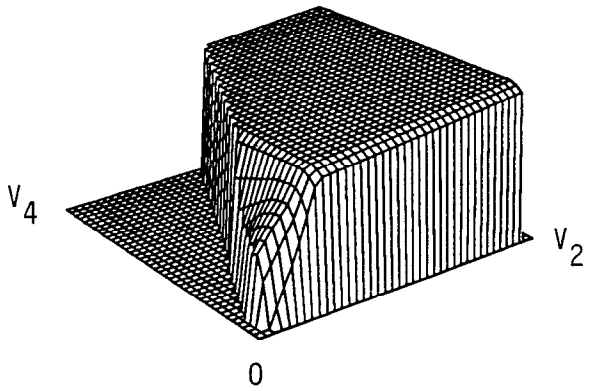
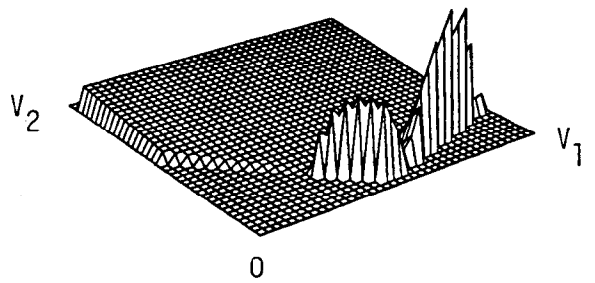
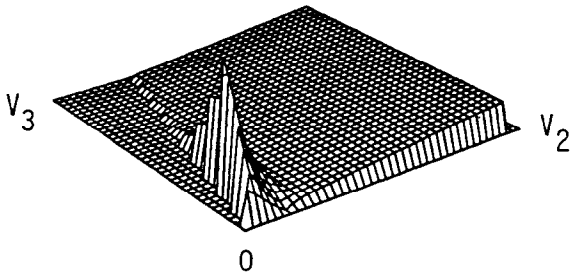

X3
X4
0

Figure 8b









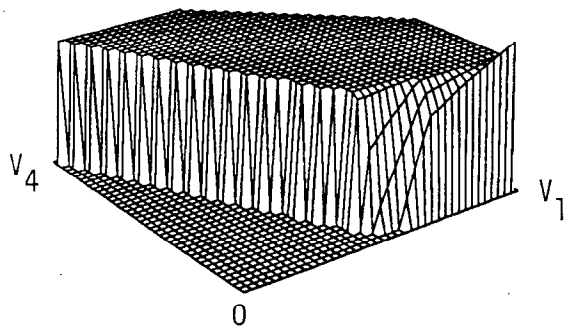
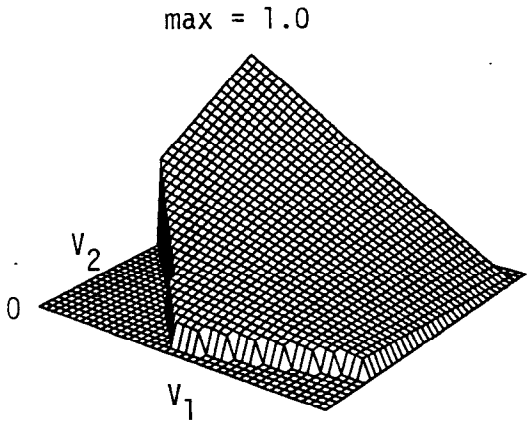
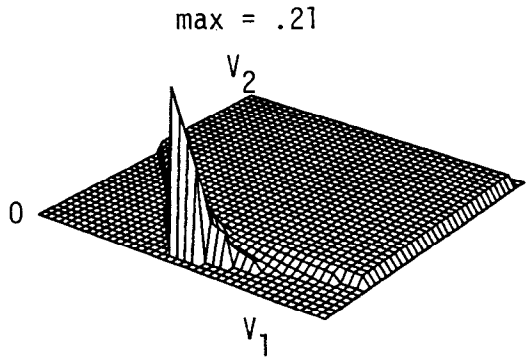


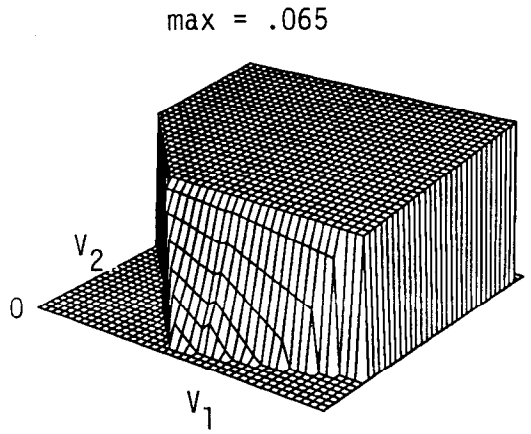
Figure 9a



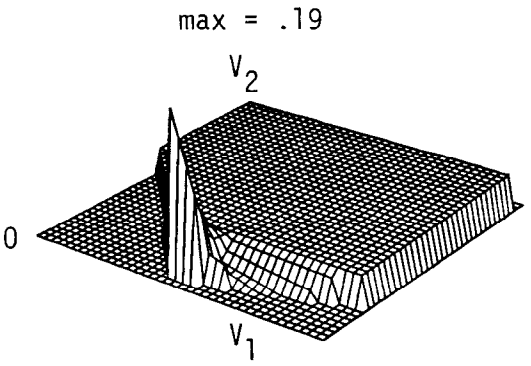
$V_3 = -.5$, $V_4 = -.5$



$V_3 = 5.5$, $V_4 = -.5$

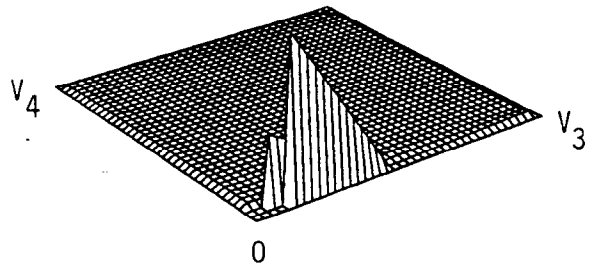


$V_3 = -.5$, $V_4 = 5.5$

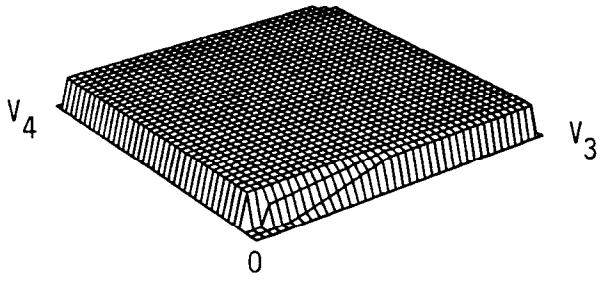


$V_3 = 5.5$, $V_4 = 5.5$

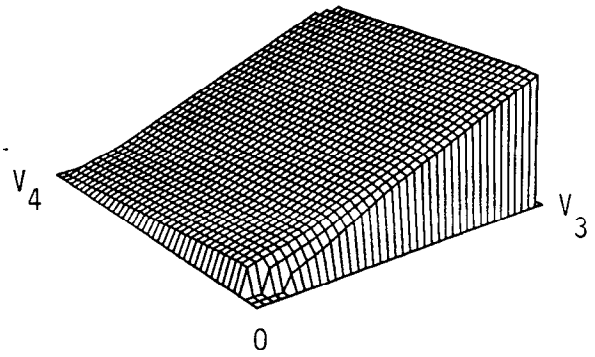
Figure 9b



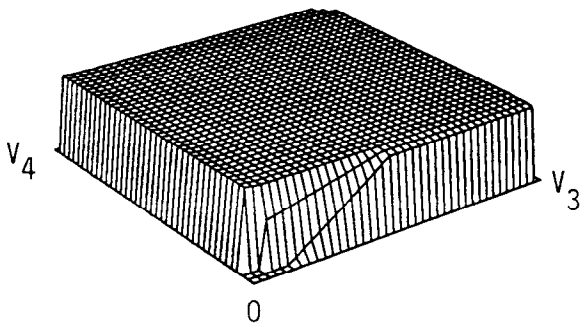
$V_1 = -6.0,\ V_2 = 5.0$



$V_1 = 6.0,\ V_2 = 2.0$

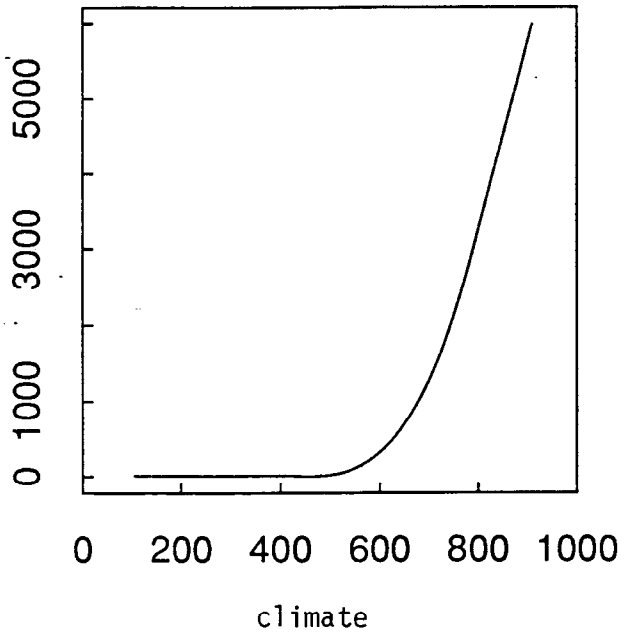


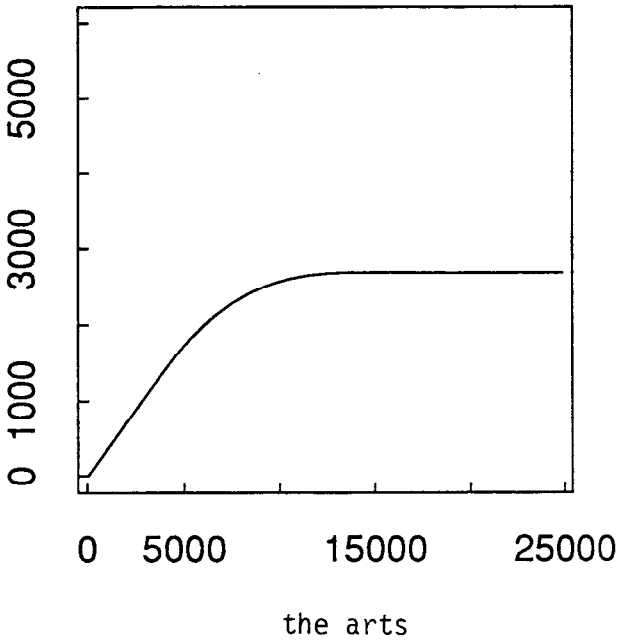
$V_1 = 0.0,\ V_2 = -.75$



$V_1 = 6.0,\ V_2 = 10.75$

Figure 9c


0 1000 3000 5000
0 200 400 600 800 1000
climate


0 1000 3000 5000
0 5000 15000 25000
the arts

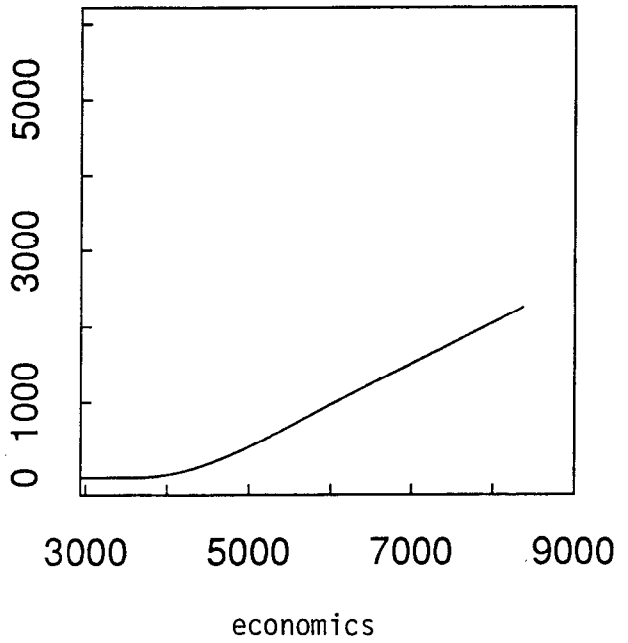

0 1000 3000 5000
3000 5000 7000 9000
economics

Figure 10

DISCUSSION

Discussion by Larry L. Schumaker of
Multivariate Adaptive Regression Splines by J. H. Friedman

Clearly the fitting of functions of more than two variables is an important problem, and it is nice to see that statisticians are willing to tackle it. Mathematicians have tended to concentrate on the bivariate case (perhaps because even there, much remains to be done). The recently published bibliography [1] provides a fairly uptodate list of what approximation theorists have been doing. Some of this work does deal with the many variable case. In particular, the papers [2–6] deal with adaptive fitting of piecewise polynomials, much in the spirit of the paper under discussion. These papers also deal with the problem of giving error bounds.

Approximation theorists have also recently been interested in the problem of approximating multivariate functions by sums of univariate functions. In this connection I would like to cite [7,8], (see also the bibliography [1] mentioned above). Other references can be found the the book [8].

Next, a few comments to the paper under discussion. I am a bit puzzled by the assertion in Section 2.4.2 and later in Section 3.2 that lack of smoothness of the approximating functions limits the accuracy of the approximation. Generally it is true that lack of smoothness of the function to be approximated limits accuracy, while for the approximating functions it is the degree of the polynomials used which is critical. Similarly, I do not understand the discussion of end effects in Section 3.7. The classical natural splines perform badly near the boundaries precisely because they smoothly match linear functions there; i.e., they are constrained at the endpoints in the wrong way. The author uses a basis of piecewise linear functions which are smoothed out to be C^1. If one does not need C^1 functions, it seems it would be better to simply use linear splines to begin with. As far as I know, the approximation properties of the modified basis function are not understood. Do they approximate with the same power as linear splines? Surely they not do as well as quadratic ones. My next remark relates to the basis functions being used. As noted in Section 3.9, the one-sided truncated power basis is well-known to be very badly conditioned whereas the classical B-splines are very well-conditioned. Why not use the latter? Updating might even be easier.

The idea of simplifying the model by removing knots (recombining pieces) strikes me as very important. This idea has recently been discovered by approximation theorists in connection with general spline fitting. The papers [9–11] are representative.

1. R. Franke and L. L. Schumaker, A bibliography of multivariate approximation, in *Topics in Multivariate Approximation*, C. K. Chui, L. L. Schumaker, and F. Utreras (eds.), Academic Press, New York, 1987, 275–335.
2. M. S. Birman and M. E. Solomiak, Approximation of the functions of the classes W_p^α by piecewise polynomial functions, Soviet Math. Dokl. **7** (1966), 1573–1577.
3. Ju. A. Brudnyi, Piecewise polynomial approximation and local approximations, Soviet Math. Dokl. **12** (1971), 1591–1594.
4. C. de Boor and J. R. Rice, An adaptive algorithm for multivariate approximation giving optimal convergence rates, J. Approximation Theory **25** (1979), 337–359.

5. R. A. DeVore and V. A. Popov, Free multivariate splines, Constructive Approximation **3** (1987), 239–248.
6. V. Popov, Nonlinear Multivariate Approximation, in *Approximation Theory VI*, C. Chui, L. Schumaker and J. Ward (eds.), Academic Press, New York, 1989, 519–560.
7. W. A. Light and E. W. Cheney, On the approximation of a bivariate function by the sum of univariate functions, J. Approximation Th. **29** (1980), 305–322.
8. W. A. Light and E. W. Cheney, *Approximation Theory in Tensor Product Spaces*, Springer-Verlag Lecture Notes 1169, New York, 1985.
9. T. Lyche and K. Mørken, Knot removal for parametric B-spline curves and surfaces, Computer-Aided Geometric Design **4** (1987), 217–230.
10. T. Lyche, T. and K. Mørken, A data reduction strategy for splines, IMA J. Numer. Anal. **8** (1988), 185–208.
11. T. Lyche, T. and K. Mørken, A discrete approach to knot removal and degree reduction algorithms for splines, in *Algorithms for the Approximation of Functions and Data*, M. G. Cox and J. C. Mason (eds.), Oxford Univ. Press, 1987.

Discussion of: Multivariate Adaptive Regression Splines

ART OWEN

Stanford University

I like MARS. It looks like a good tool for pulling out the most useful parts of large interaction spaces. Most of my comments are directed at accounting issues: How many degrees of freedom are used in knot selection? How can the cost be lowered? At the end, there are some comments on how one might apply MARS to models for which fast updating is not available.

My main interest in MARS stems from work in computer experiments. In these applications smooth functions, of fairly high complexity, are evaluated over high dimensional domains with no sampling error. I plan to use MARS on such functions evaluated over Latin hypercube designs (McKay, Conover and Beckman, 1979). Some theory for linear modeling of nonrandom responses over such designs is given in Owen (1990).

When there is no noise, one expects that a larger number of knots might be warranted. It then becomes worthwhile to lower the price of a knot somehow.

Degrees of Freedom in Broken Line Regression. Consider the broken line regression model

$$Y_i = b_0 + b_1 t_i + \beta(t_i - \theta)_+ + \epsilon_i \quad i = 1, \cdots, n \tag{1}$$

where $t_1 \le t_2 \le \ldots \le t_n$ are nonrandom with $\sum t_i = 0$ and $\sum t_i^2 = n\sigma_t^2$, ϵ_i are independent $N(0,1)$ and b_0, b_1, β and θ are parameters. Taking $\beta = 0$ in (1) yields a one segment model. Taking $\beta \neq 0$ and $t_1 < \theta < t_n$ yields a two segment model. This model has been studied by Feder (1967), Hinkley (1969), Davies (1977,1987) and Knowles and Siegmund (1989). All these authors point out that nonstandard asymptotics apply when $\beta = 0$. In particular estimation of β and θ "uses up" more than 2 degrees of freedom in that case. Hinkley (1969) reports that approximately 3 degrees of freedom are used, based on a simulation in which the potential knots were uniformly spaced. When $\beta \neq 0$ the standard asymptotics, that is 2 degrees of freedom, are relevant (Feder 1967, Part 2).

This suggests that when the evidence that $\beta \neq 0$ is extremely strong, that fewer degrees of freedom should be "charged" than when the evidence is borderline. Of course for small $\beta \neq 0$ and finite n the nonstandard asymptote might be the more accurate one.

Following Davies (1987) we consider testing whether the line breaks at θ via

$$Z(\theta) = V^{-1/2}(\theta) \sum_{i=1}^{n} \hat{\epsilon}_i (t_i - \theta)_+ \tag{2}$$

where

$$\hat{\epsilon}_i = Y_i - \overline{Y} - t_i \overline{tY}/\sigma_t^2$$

$$V(\theta) = R_{02} - \frac{1}{n}R_{01}^2 - \frac{1}{n\sigma_t^2}R_{11}^2$$

and

$$R_{jk} = R_{jk}(\theta) = \sum_{i=1}^{n} t_i^j (t_i - \theta)_+^k. \tag{3}$$

The null distribution of $Z(\theta)$ is $N(0,1)$ for $t_2 < \theta < t_{n-1}$. To test whether the regression line breaks at all we consider the supremum of $Z(\theta)$ or of $|Z(\theta)|$ over an interval. The null covariance of $Z(\theta)$ and $Z(\phi)$ is

$$\rho(\theta,\phi) = V^{-1/2}(\theta)V^{-1/2}(\phi)\left[R_{01,01}(\theta,\phi) - \frac{1}{n}R_{01}(\theta)R_{01}(\phi) - \frac{1}{n\sigma_t^2}R_{11}(\theta)R_{11}(\phi)\right] \tag{4}$$

where

$$R_{01,01}(\theta,\phi) = \sum_{i=1}^{n}(t_i - \theta)_+(t_i - \phi)_+. \tag{5}$$

Bounds for

$$P\left(\sup_{\theta_0 \le \theta \le \theta_1} Z(\theta) > c\right)$$

may be derived from the expected number of upcrossings of c by Z. For example

$$P\left(\sup_{\theta_0 \le \theta \le \theta_1} Z(\theta) > c\right) \le \Phi(-c) + \frac{1}{2\pi}\exp\left(-\frac{1}{2}c^2\right)\int_{\theta_0}^{\theta_1} \rho_{11}^{1/2}(\theta)d\theta \tag{6}$$

where Φ is the standard normal distribution function and

$$\rho_{11}(\theta) = \frac{\partial^2}{\partial\theta\partial\phi}\rho(\theta,\phi)\,|_{\phi=\theta}\,. \tag{7}$$

Davies (1977) gives (6) as his (3.7), except that he uses

$$\rho_{11}(\theta) = -\frac{\partial^2}{\partial\phi^2}\rho(\phi,\theta)\,|_{\phi=\theta}\,. \tag{7'}$$

Definitions (7) and (7') are equivalent when both derivatives exist (since the process has constant variance). For broken line regression neither exists but the problem is easily handled: replace $(t_i - \theta)_+$ by $q_\epsilon(t_i - \theta)$ where

$$q_\epsilon(x) = \begin{cases} 0, & x \le -\epsilon \\ \frac{\epsilon}{4} + \frac{x}{2} + \frac{x^2}{4\epsilon}, & |x| \le \epsilon \\ x, & x \ge \epsilon \end{cases}$$

for small positive ϵ in (1), (2), (3), (5). The function q_ϵ is Friedman's piecewise cubic approximation to x_+ which in this case is piecewise quadratic. Now $\partial^2\rho(\theta,\phi)/\partial\theta\partial\phi$ exists and is continuous so (6) follows from (3.3) of Leadbetter (1972). The right side of (6) is continuous in $\epsilon > 0$ when this substitution is made. If we take the limit as $\epsilon \downarrow 0$ we get

$$\rho_{11}(\theta) = -\frac{1}{4}\left(\frac{V'(\theta)}{V(\theta)}\right)^2 + V^{-1}(\theta)\left[R_{00} - \frac{1}{n}R_{00}^2 - \frac{1}{n\sigma_t^2}R_{10}^2\right]$$

where

$$V' = \frac{d}{d\theta}V(\theta) = 2R_{01} + \frac{2}{n}R_{00}R_{01} + \frac{2}{n\sigma_t^2}R_{10}R_{11}$$

with

$$R_{j0} = R_{j0}(\theta) = \sum_{t_i > \theta} t_i^j. \tag{8}$$

One could also take the sum in (8) over $t_i \geq \theta$; it makes no difference in (6) because the integrand is only affected at finitely many places.

Since both positive and negative β are of interest we write

$$\begin{aligned} P\left(\sup_{\theta_0 \leq \theta \leq \theta_1} |Z| > c\right) &= P\left(\sup_{\theta_0 \leq \theta \leq \theta_1} Z > c\right)\left[2 - P\left(\inf_{\theta_0 \leq \theta \leq \theta_1} Z < -c \mid \sup_{\theta_0 \leq \theta \leq \theta_1} Z > c\right)\right] \\ &\leq 2P\left(\sup_{\theta_0 \leq \theta \leq \theta_1} Z > c\right) \\ &= P(\chi^2_{(1)} > c^2) + \frac{1}{\pi}\exp\left(-\frac{1}{2}c^2\right)\int_{\theta_0}^{\theta_1} \rho_{11}^{1/2}(\theta)d\theta \\ &= P(\chi^2_{(1)} > c^2) + \frac{1}{\pi}\int_{\theta_0}^{\theta_1} \rho_{11}^{1/2}(\theta)d\theta P(\chi^2_{(2)} > c^2). \end{aligned} \tag{9}$$

For a given set of t_i, ρ_{11} can be computed and the integral in (9) can then be numerically evaluated. Updating formulae are easy to derive for the R_{jk}. Assume that $\sum t_i = 0$ and $\sum t_i^2 = n$, that is σ_t^2=1. The integral of $\rho_{11}^{1/2}$ is invariant when a nonsingular location scale transformation is applied to the t_i and to the limits of integration. Suppose we want $\rho_{11}(\theta_i)$ for $t_2 < \theta_1 \leq \theta_2 \leq \ldots \leq \theta_m < t_{n-1}$. Let $s_1 \leq s_2 \leq \ldots \leq s_{n+m}$ be obtained by pooling and sorting the data points t_i and the evaluation points θ_i. Let D_i be 1 if s_i arose as a data point and 0 if s_i arose as an evaluation point. With this set up, $R_{00}(s_1) = n - 1$, $R_{01}(s_1) = -nt_1$, $R_{10}(s_1) = -t_1$ and $R_{11}(s_1) = n$. For $2 \leq i \leq n + m$, $R_{00}(s_i) = R_{00}(s_{i-1}) - D_i$, $R_{01}(s_i) = R_{01}(s_{i-1}) - (s_i - s_{i-1})R_{00}(s_{i-1})$, $R_{10}(s_i) = R_{10}(s_{i-1}) - D_i s_i$, and $R_{11}(s_i) = R_{11}(s_{i-1}) - (s_i - s_{i-1})R_{10}(s_{i-1})$. Now ρ_{11} can be computed on noting that $R_{02}(\theta) = R_{11}(\theta) - \theta R_{01}(\theta)$ and the values corresponding to evaluation points can be extracted. At an evaluation point θ_i that coincides with a data point t_j, $\rho_{11}(\theta_i)$ will depend on the order in which the two points appear in the list of s_i.

For uniformly distributed t_i we can find a simple approximation to (9). Let $t_i = (i - .5)/n - .5$ so $\sigma_t^2 = (n^2 - 1)/(12n^2) \doteq 1/12$, and approximate R_{jk} by

$$\tilde{R}_{jk}(\theta) = n\int_{\theta}^{1/2} u^j(u - \theta)^k du$$

and similarly for $\tilde{R}_{01,01}$. Use $\tilde{V}$ and $\tilde{\rho}$ to denote the corresponding changes to V and ρ. Some calculus gives

$$\tilde{V}(\theta) = \frac{n}{3}\left(\frac{1}{4} - \theta^2\right)^3$$

and

$$\tilde{\rho}_{11} = \frac{3}{4}\left(\frac{1}{4} - \theta^2\right)^{-2}$$

and so

$$P\left(\sup_{\theta_0 \le \theta \le \theta_1} |Z| > c\right) \le P(\chi^2_{(1)} > c^2) + \frac{\sqrt{3}}{2\pi}\int_{\theta_0}^{\theta_1} \frac{d\theta}{(\frac{1}{4} - \theta^2)^{1/2}} P(\chi^2_{(2)} > c^2)$$
$$= P(\chi^2_{(1)} > c^2) + \frac{\sqrt{3}}{2\pi}\left(\log\frac{\frac{1}{2} + \theta_1}{\frac{1}{2} - \theta_1} - \log\frac{\frac{1}{2} + \theta_0}{\frac{1}{2} - \theta_0}\right) P(\chi^2_{(2)} > c^2).$$

Therefore for $0 < \epsilon < \frac{1}{2}$

$$P\left(\sup_{|\theta| \le \frac{1}{2} - \epsilon} |Z| > c\right) \le P(\chi^2_{(1)} > c^2) + \frac{\sqrt{3}}{\pi}\log\left(\frac{1-\epsilon}{\epsilon}\right) P(\chi^2_{(2)} > c^2). \tag{10}$$

Knowles and Siegmund (1989, Section 4) obtain (10) for uniformly spaced t_i and large n using the Hotelling-Naiman "volume of tubes" approach. For large n the c above corresponds to $n^{1/2}w$ in their notation. One substitutes their equation (10) in their (6) and letting $n \to \infty$ one gets the one tailed (i.e. sup Z) version of (10) above.

Figure 1 compares the bound in (10) to tail probabilities from chi-squared distributions on degrees of freedom ranging from 1 through 4 by steps of 0.5. Chi-squared tail probabilities are plotted by lines, probabilities based on (10) are plotted as asterisks. For Figure 1, $\epsilon = .2, .05, .01$ and .0001 are used.

The points for $\epsilon = .01$ fall close to the curve for $\chi^2_{(3)}$. So searching the central 98% of the range for θ uses roughly 3 degrees of freedom under the null hypothesis. Hinkley (1969) notes from simulations that roughly 3 degrees of freedom are used. The relative error in these probabilities is less than 0.1 when the value of the bound (10) is between .05 and .005. While the tail probability in (10) is not in the chi-squared family it would appear that if one were to approximate it in the chi-squared family for purposes of model selection, that 3 degrees of freedom would be a reasonable choice. Using 3 degrees of freedom might even be a little conservative since it would be common to search over less than the central 98% of the range.

The choice $\epsilon = .2$ corresponds to searching the central 60% of the range and uses approximately 2 degrees of freedom under the null hypothesis. It also uses (asymptotically) 2 degrees of freedom under the alternative $\beta \neq 0, |\theta| < 0.3$. So it might be reasonable to make all splits in the central 60% of whatever range is being searched, and charge 2 degrees of freedom.

Figure 2 shows 10 realizations of the Gaussian process $Z(\theta)$ on $\left(-\frac{1}{2}, \frac{1}{2}\right)$ with covariance given by $\tilde{\rho}$. The mean is 0 and the variance is 1 over the whole range. The process "turns more rapidly" (correlations are smaller) for large $|\theta|$, so more upcrossings occur near the edges.

Figure 3 compares $\tilde{\rho}_{11}^{1/2}$ and $\rho_{11}^{1/2}$ where the latter is computed for $n = 25$. The steep parts of the jagged curve are meant to be vertical. Since (10) only uses integrals of $\rho_{11}^{1/2}$ the continuous approximation using $\tilde{\rho}$ should be reasonably accurate, especially if each limit of integration is near the midpoint of two consecutive t_i.

Now suppose that t_i is the $(i - .5)/n$ quantile of the distribution with density

$$f(x) = e^{-x-1}, \qquad x \geq -1. \tag{11}$$

This is the unit exponential distribution shifted left one unit so as to have mean zero. For this distribution $\tilde{R}_{00} = \tilde{R}_{01} = e^{-1-\theta}$, $\tilde{R}_{10} = (\theta + 1)e^{-1-\theta}$, $\tilde{R}_{02} = 2e^{-1-\theta}$ and $\tilde{R}_{11} = (2 + \theta)e^{-1-\theta}$, for $\theta \geq -1$. The function $\tilde{\rho}$ tends to $1/4$ as $\theta \to \infty$ and $\tilde{\rho}$ is asymptotic to $(3/4)(1 + \theta)^{-2}$ as $\theta \to -1$. So searching for a breakpoint in an interval of given length should "cost" more if that interval is near the left edge of the predictor space than if it is near the right edge. Perhaps this is to be expected because there tend to be more points t_i per unit length at the left end. Figure 4 plots $\tilde{\rho}_{11}^{1/2}(F^{-1}(u))/f(F^{-1}(u))$ versus u where f is the density in (11) and F is the corresponding distribution function. Asterisks are used for $\tilde{\rho}_{11}^{1/2}$ taken from the exponentially distributed t_i and the smooth curve is for $\tilde{\rho}_{11}^{1/2}$ taken from the uniform distribution. If one decides to search over the range between two sample quantiles then the null probability of an upcrossing is very close under an exponential design to what it is under a uniform design. Bounds like those in (10) based on the uniform design would be conservative for an exponential design, since the asterisks lie below the curve in Figure 4.

Equation (1) is a model for the problem of deciding if and where to bend a line. If a variable has not yet entered one might consider

$$Y_i = b_0 + \beta(t_i - \theta)_+ + \epsilon_i \tag{12a}$$

or

$$Y_i = b_0 + \beta(\theta - t_i)_+ + \epsilon_i. \tag{12b}$$

Note that (12a) at $\theta = t_1$ and (12b) at $\theta = t_n$ are the same (affine) model and that (12a) at $\theta = t_n$ and (12b) at $\theta = t_1$ are both the constant model.

For the model in (12a) we take

$$Z(\theta) = V^{-1/2}(\theta) \sum_{i=1}^{n} \hat{\epsilon}_i (t_i - \theta)_+$$

where $\hat{\epsilon}_i = Y_i - \bar{Y} = \epsilon_i - \bar{\epsilon}$ and $V(\theta) = R_{02} - R_{01}^2/n$. The correlation is

$$\rho(\theta, \phi) = V^{-1/2}(\theta) V^{-1/2}(\phi) \left[R_{01.01}(\theta, \phi) - \frac{1}{n} R_{01}(\theta) R_{01}(\phi) \right]$$

with

$$\rho_{11}(\theta) = V^{-1}(\theta)[R_{00} - \frac{1}{n}R_{00}^2] - \frac{1}{4}(V'(\theta)/V(\theta))^2.$$

For a uniform spacing of t_i one finds

$$\tilde{\rho}_{11}^{1/2}(\theta) = \frac{12^{1/2}(1-\lambda)^{1/2}}{\lambda(4-3\lambda)}$$

where $\lambda = \lambda(\theta) = 1/2 - \theta$.

If $\beta = 0$ then

$$P\left(\sup_{-\frac{1}{2}\le\theta\le\theta_1} |Z(\theta)| > c\right) \le P(\chi^2_{(1)} > c^2) + \frac{1}{\pi}\int_{-1/2}^{\theta_1} \tilde{\rho}_{11}^{1/2}(\theta)d\theta P(\chi^2_{(2)} > c^2).$$

Since the process Z can be "glued onto" another one for testing $\beta \neq 0$ in (12b) the null probability of "splitting the constant model" is bounded by

$$P(\chi^2_{(1)} > c^2) + \frac{2}{\pi}\int_{-1/2}^{\theta_1} \tilde{\rho}_{11}^{1/2}(\theta)d\theta P(\chi^2_{(n)} > c^2). \tag{13}$$

For searching the central 98% of the range the coefficient of $P(\chi^2_{(2)} > c^2)$ in (10) is approximately 2.53. If one considers model (12a) over all but the rightmost 1% of the range and (12b) over all but the leftmost 1% of the range the corresponding coefficient is approximately 2.64. So the degrees of freedom used up in deciding whether to split a constant regression are much the same as those used in splitting a linear regression.

In the backward stepwise part (Algorithm 3), how many degrees of freedom should be charged when a knot caused two regressors to be added to the model, and one of them gets dropped? I would guess from the analysis of (12ab) that the full charge for the knot should be assessed, but from Friedman's talk at Interface '90 it seems that half the charge for the knot is assessed.

An alternative to restricting the search to a central subinterval, such as the central 60%, is to search the whole interval, but apply a penalty that increases as the potential knot location nears the end of the range. Davies (1977, equation 3.3) quotes an upcrossing bound for $P(\sup_\theta Z(\theta) - c(\theta) > 0)$ for continously differentiable $c(\theta)$. For the process described by the continuous approximation to the uniform design case, the upcrossing bound is especially simple when $c(\theta) = A + B\log((0.5 + |\theta|)/(0.5 - |\theta|))$ where $A, B > 0$. (This c has a cusp, but the bound should still be applicable.) One finds that

$$P(\sup_\theta Z(\theta) - c(\theta) > 0) \le 2\,(\Phi(B') - 1/2 + \varphi(B')/B')\,P(\chi^2_{(1)} > A^2) \tag{14}$$

where $B' = 2B/3^{1/2}$ and φ is the standard normal density.

Conclusions. It appears that Hinkley's (1969) heuristic of charging 3 degrees of freedom for adding a line segment is reasonably accurate in a variety of settings.

By restricting the search to a subinterval it may be possible to reduce the cost of breaking a line to 2 degrees of freedom. A smoothly varying preference for central splits based on (14) could also be used to lower the cost of knot selection.

The calculations in the preceding section are most relevant to splitting the constant function B_0. When splitting another basis function B_m along variable T, perhaps

$$Y_i = B_m(X_i)(b_0 + b_1 t_i + \beta(t_i - \theta)_+) + \epsilon_i \quad i = 1, \cdots, n$$

should replace (1), or a similar change should be made to (12ab) depending on context.

Finally I would like to pick up on Friedman's comments on updating formulae in Section 5.4. When searching a variable for a split point, it may not be necessary to consider every value. The test statistic $Z(\theta)$ should tend to have very smooth sample paths. In the smoothly approximated uniform design case

$$\min_{|\theta| \le .4} \max_{\phi \in \{0, \pm .2, \pm .35\}} \tilde{\rho}(\theta, \phi) \ge 0.94.$$

That is the central 80% of the range can be effectively scanned by considering only 5 points. Note that the realizations in Figure 2 tend to be quite smooth in the middle. I would expect the true underlying function would also make a smooth contribution to $Z(\theta)$, perhaps smoother than that of the noise, though I also expect pathological cases are possible. So after 5 evaluations one should have a good idea where the maximum is and whether it is worth including in the model. Then one could spend a few more evaluations on a more local search, or wait until the backward stepwise algorithm has finished to refine the search. When a knot is put at 0.2, the next step would involve looking at five places between -.5 and .2 and at five places between .2 and .5. If this works it should be possible to extend MARS to robust regressions, generalized linear models and the proportonal hazards model. Davies (1977, Section 5) has some suggestions on how to perform various tasks on representative points θ, and on picking those representative points.

Acknowledgements. I would like to thank Iain Johnstone and David Siegmund for helpful discussions. In particular Professor Johnstone showed me some unpublished work of his applying the Hotelling-Naiman volume of tubes methodology to broken line regression and including a plot similar to Figure 1.

REFERENCES

Davies, R.B. (1977). Hypothesis testing when a nuisance parameter is present only under the alternative. *Biometrika* **64**, 247–54.

Davies, R.B. (1987). Hypothesis testing when a nuisance parameter is present only under the alternative. *Biometrika* **74**, 33–43.

Feder, P.I. (1967). *On the Likelihood Ratio Statistic with Applications to Broken Line Regression.* Dissertation, Dept. of Statistics, Stanford Universtiy.

Hinkley, D.V. (1969). Inference about the intersection in two-phase regression. *Biometrika* **56**, 495–504.

Knowles, M.D. & Siegmund D.O. (1989). On Hotelling's approach to testing for a nonlinear parameter in regression. *International Stat. Rev.* **57**, 205–220.

Leadbetter, M.R. (1972). Point processes generated by level crossings. *Stochastic Point Processes* John Wiley, New York.

McKay, M.D., Conover, W.J. & Beckman, R.J. (1979). A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a Computer Code. *Technometrics* **21**, 239–245.

Owen, A.B (1990). "A Central Limit Theorem for Latin Hypercube Sampling". Technical Report. Department of Statistics, Stanford University.

Captions for Figures

Figure 1: Tail Probabilities.

The solid lines give (right) tail probabilities from chi-squared distributions on degrees of freedom 1 through 4 in steps of 1/2. The asterisks plot tail probabilities from formula (10) using $\epsilon = .2, .05, .01, .0001$.

Figure 2: Realizations of $Z(\theta)$.

Shown are 10 realizations of the Gaussian process with mean 0, constant unit variance and correlation $\tilde{\rho}$, a continuous approximation to ρ of formula (4).

Figure 3: Comparison of $\tilde{\rho}_{11}^{1/2}$ and $\rho_{11}^{1/2}$.

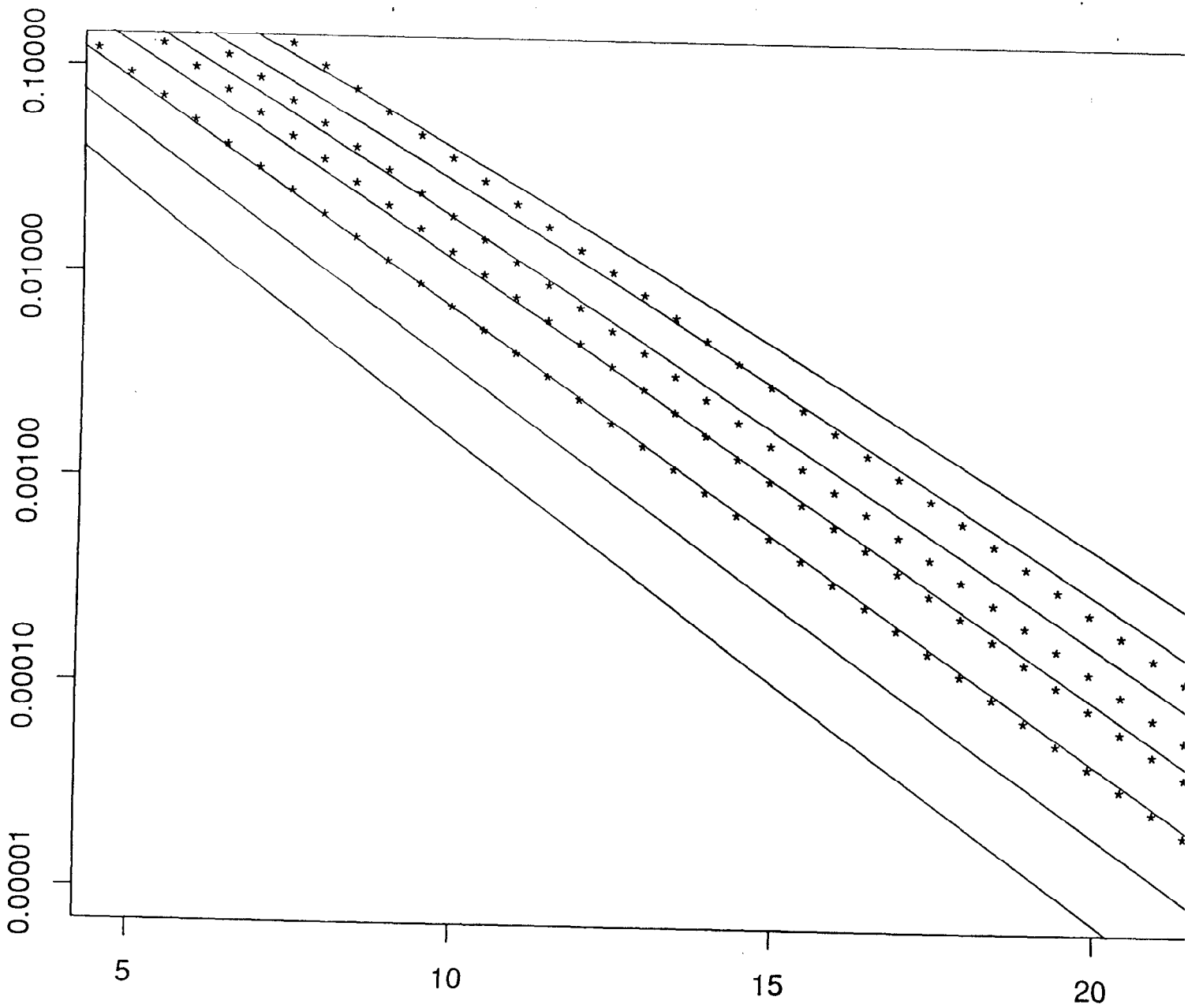
The jagged curve is $\rho_{11}^{1/2}$ from formula (4) with $n = 25$. The smooth curve is the continuous approximation $\tilde{\rho}_{11}^{1/2}$.

Figure 4: Comparison of uniform and exponential designs.

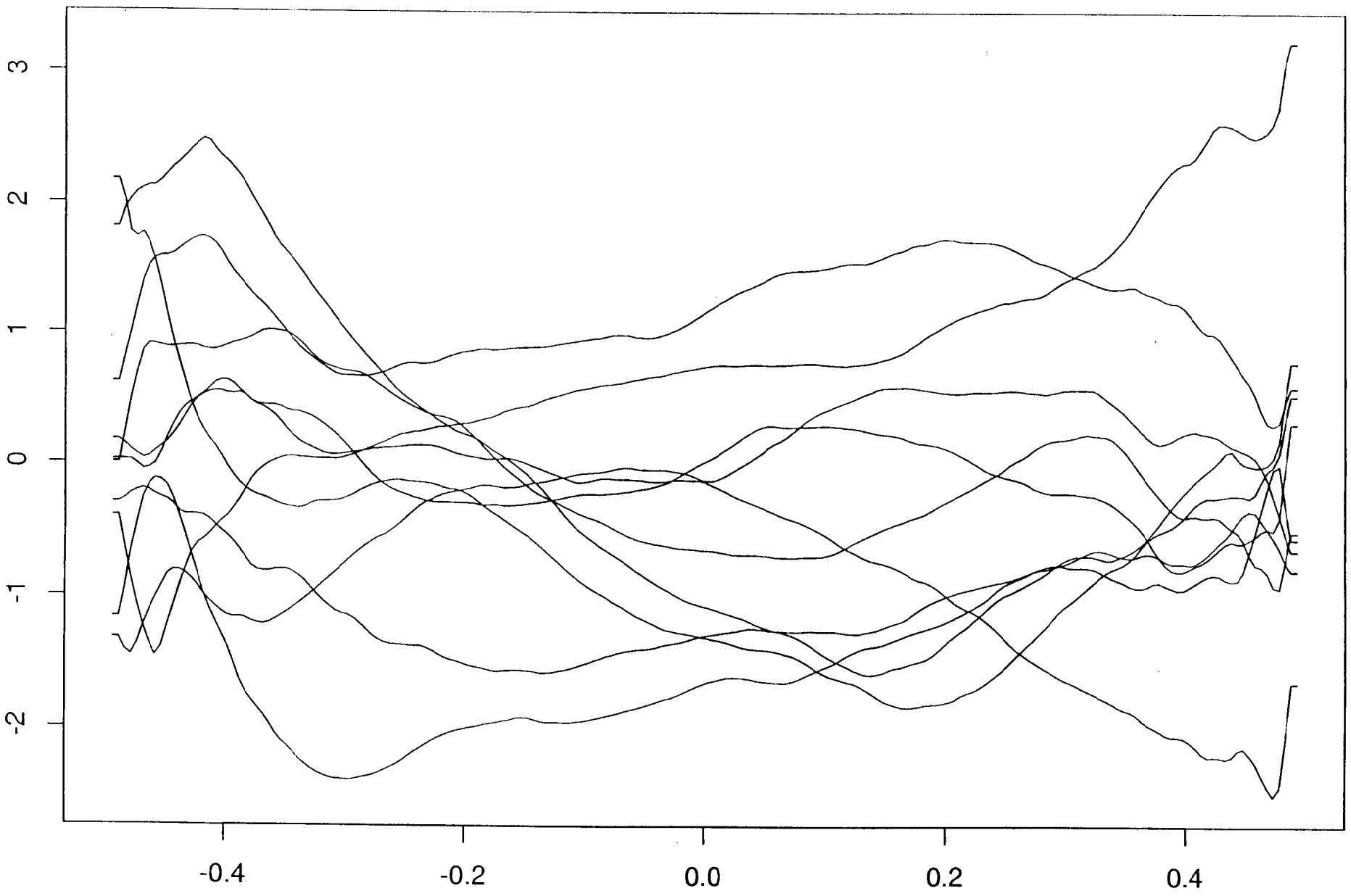
The smooth curve is the continuous approximation $\tilde{\rho}_{11}^{1/2}(\theta)$ assuming a uniform distribution of design points. The asterisks are $\tilde{\rho}_{11}^{1/2}(\theta)/f(\theta)$ where $\tilde{\rho}$ is the continuous approximation to ρ for design points from the exponential distribution f given by equation (11). The x axis is $F^{-1}(\theta)$ where F is the appropriate distribution (uniform or exponential).

DEPARTMENT OF STATISTICS
STANFORD UNIVERSITY
STANFORD CA, 94305, USA

l

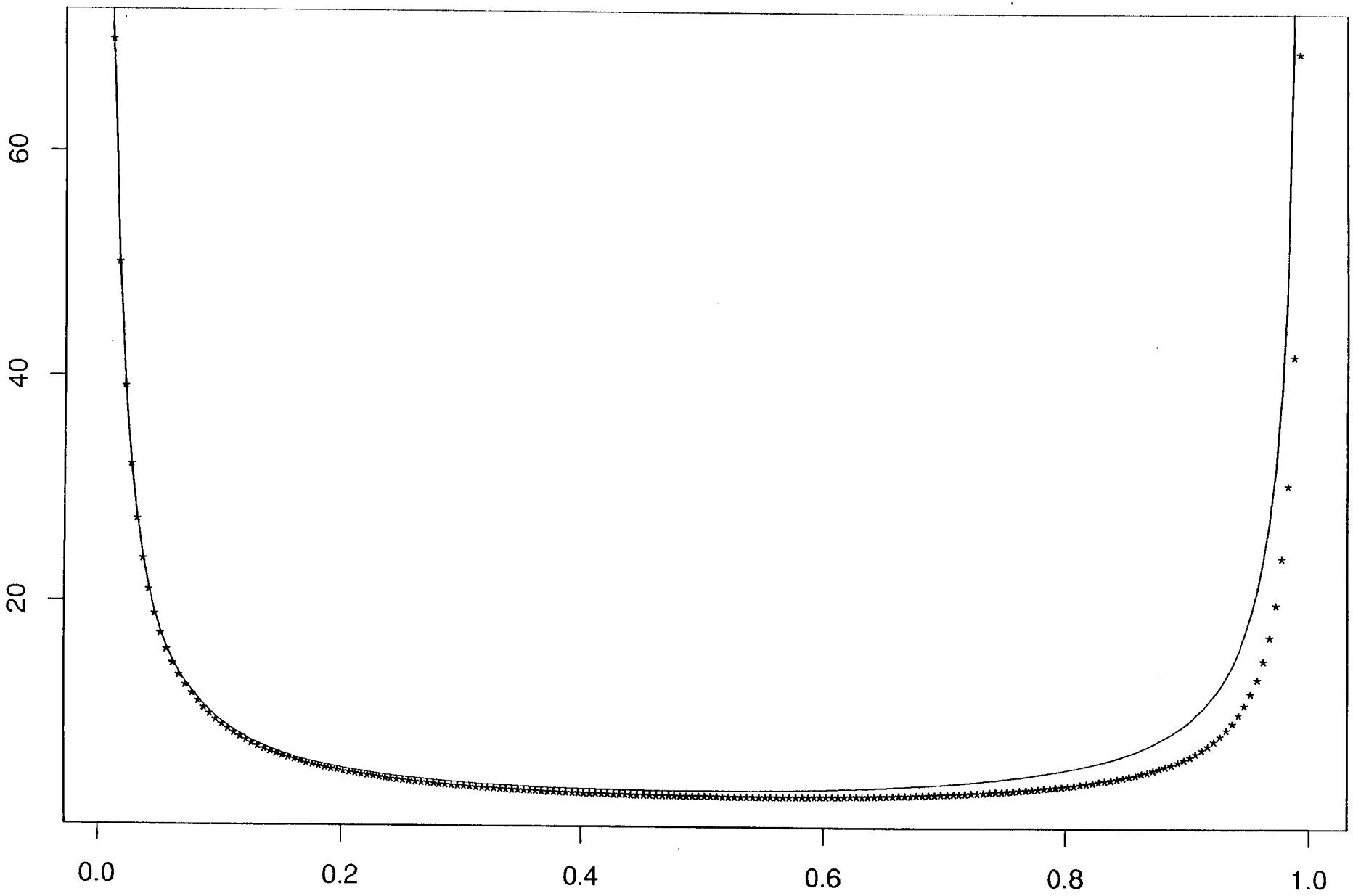


2



3

4



DISCUSSION OF "MULTIVARIATE ADAPTIVE REGRESSION SPLINES"

CHARLES J. STONE

University of California at Berkeley

This pioneering paper successfully combines creative breakthroughs (especially, *not* removing the parent basis function) with numerous techniques developed over the years by the author and his collaborators and others (especially, Patricia Smith). The tactics of MARS are specific to least-squares estimation of a regression function, but the general strategy is much more widely applicable.

In Algorithm 2, the residual sum of squares could be used for *LOF*. This would allow the entering of new basis functions to be broken up into two steps: decide which basis function next to enter into the model; enter the selected basis function. Using the "t to enter" algorithm from stepwise regression would then eliminate the need to fit the various candidate models in the course of determining the next basis function to enter.

Algorithm 3 could be divided into three tasks: decide which basis function next to remove from the model; remove the selected basis function; determine the final model. In the first two of these steps, the residual sum of squares could be used for *LOF*. Using the "t to remove" algorithm would then eliminate the need to fit the various candidate models in the course of determining the next basis function to remove.

With these changes, MARS could be extended to handle logistic regression in a more natural manner. The "t to enter" step in Algorithm 2 could be replaced by an algorithm based on Rao's score test and the "t to remove" step in Algorithm 3 could be replaced by an algorithm based on Wald's test. The actual maximum-likelihood fitting (based, say, on the Newton-Raphson algorithm and taking advantage of the concavity of the log-likelihood function) would be applied $M_{\max}$ times in Algorithm 2, once after each application of the score test, and $M_{\max}$ times in Algorithm 3. The final model selection could be based on a variant of AIC, modified along the lines of (32).

Consider now the estimation of an unknown density or profability function f on a set $\mathcal{Y}$. In order to guarantee positivity, we can model $\log(f)$ as a member of some adaptively selected space $\mathcal{S}$ that does not contain the constant functions. Letting $B_1, \cdots, B_K$ be a

basis of $\mathscr{S}$, we can write the estimate of f as $\hat{f} = \exp(\Sigma_k \hat{\theta}_k H_k - c(\hat{\theta}))$, where $\hat{\theta} = (\hat{\theta}_1, \cdots, \hat{\theta}_K)^t$ and $c(\hat{\theta})$ is the normalizing constant. This has the form of an exponential family. If $\mathscr{Y} = \mathbb{R}$ and the functions in $\mathscr{S}$ have linear tails, then $\hat{f}$ has exponential tails. The selection of $\mathscr{S}$ could be done by applying the general strategy of MARS. Ignoring the model selection, we can choose $\hat{\theta}$ by maximum-likelihood. The asymptotic theory of such estimates, with $\mathscr{Y}$ a compact interval in $\mathbb{R}$ and $B_1, \cdots, B_K$ consisting of B-splines and without model selection, has been treated in Stone (1990). The numerical behavior of such estimates, modified to incorporate stepwise knot deletion based on Wald's test and a variant of AIC, is studied in Kooperberg and Stone (1990).

Consider next a random pair $(\mathbf{X}, Y)$, where $\mathbf{X}$ is an n-dimensional random vector and Y is a $\mathscr{Y}$-valued random variable. Let $f(y \mid \mathbf{x})$ denote the conditional density or probability function of Y given that $\mathbf{X} = \mathbf{x}$. Consider an estimate $\hat{f}(y \mid \mathbf{x})$ of the form $\hat{f}(y \mid \mathbf{x}) = \exp(\Sigma_k \hat{\theta}_k(\mathbf{x}) B_k(y) - c(\hat{\theta}(\mathbf{x})))$, where $B_1, \cdots, B_K$ are suitable basis functions of a possibly adaptively selected space $\mathscr{S}$ of functions on $\mathscr{Y}$. (The functions in $\mathscr{S}$ should be piecewise linear for $c(\cdot)$ to be computed rapidly.) We could model $\theta_1(\cdot), \cdots, \theta_K(\cdot)$ in turn as members of possibly adaptively selected spaces $\mathscr{H}_1, \cdots, \mathscr{H}_K$ respectively. Letting H_{jk}, $1 \le j \le J_k$, be a basis of $\mathscr{H}_k$, we can write $\hat{\theta}_k(\mathbf{x}) = h_k(\mathbf{x}; \hat{\beta}_k) = \Sigma \hat{\beta}_{jk} H_{jk}(\mathbf{x})$. This leads to an estimate of $f(y \mid \mathbf{x})$ having the form

$$\hat{f}(y \mid \mathbf{x}) = \exp(\Sigma_k \Sigma_j \hat{\beta}_{jk} H_{jk}(\mathbf{x}) B_k(y) - c(\mathbf{h}(\mathbf{x}; \hat{\beta}))), \quad y \in \mathscr{Y}, \tag{1}$$

where $\hat{\beta}$ is the JK-tuple consisting of $\hat{\beta}_{jk}$, $1 \le k \le K$ and $1 \le j \le J_k$, in some order. This estimate has the form of a multiparameter exponential family, so the corresponding log-likelihood function is again concave. The asymptotic theory of such estimates, with $\mathscr{Y}$ a compact interval in $\mathbb{R}$, $\mathscr{H}_1 = \cdots = \mathscr{H}_K$ and bases consisting consisting of B-splines and without model selection, has been treated in Stone (1989). It remains to investigate the numerical behavior of such estimates, especially as modified to incorporate the strategy of MARS. Perhaps the resulting technology should be referred to as "multivariate adaptive response splines (MARES)."

Suppose, in particular, that $\mathcal{Y} = \{0, 1\}$. Then we can let $\mathcal{S}$ be the one-dimensional space having basis $B_1(y) = y$. In this context, (1) reduces to logistic regression. Similarly, by letting $\mathcal{Y}$ be a finite set of size 3 or more, we can apply the strategy of MARS to the polytomous extension of logistic regression.

The more general setup given by (1) allows for the estimation of the conditional variance and conditional quantiles of an arbitrary random variable Y given $\mathbf{X}$ as well as estimation of the conditional mean of Y given $\mathbf{X}$, which is treated in the present paper.

The general strategy of MARS is also applicable to time series.

REFERENCES

KOOPERBERG, C. and STONE, C. J. (1990). A study of logspline density estimation. Manuscript.

STONE, C. J. (1989). Asymptotics for doubly-flexible logspline response models. Manuscript.

STONE, C. J. (1990). Large-sample inference for logspline models. *Ann. Statist.* 17 ?–?.

DEPARTMENT OF STATISTICS
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720

Some Comments on Multivariate Adaptive Regression Splines

by Jerome H. Friedman

Finbarr O'Sullivan

Dept. of Biostatistics and Statistics

University of Washington

Seattle, WA 98195

June 30, 1990

[1]This research was supported in part by the Dept. of Energy under FG0685-ER2500 by the National Cancer Institute under 2P01-CA-42045

This article reviews a set of key developments in non-parametric function estimation, many of them due in part or in large to Professor Friedman, which have radically changed the scope of modern Statistics. MARS is an impressive addition to this set. There is a growing practical interest in innovative adaptive function estimation techniques. For example, I am aware of the need for sophisticated covariate adjustment in connection with survival analysis of a large clinical trail where $N = 27,000$ and $n \geq 200$; The thought of sending these data to MARS for analysis will have undoubted appeal!

1 General Comments

With any adaptive regression technique it is of interest to know the kinds of functions which cause greatest difficulty. MARS is coordinate sensitive. A rotation of the coordinate axes in the examples in sections 4.2 and 4.3 will destroy the simple additive and low order interactive structure. Will this substantially degrade the performance (ISE) of MARS? Perhaps the effect could be ameliorated by allowing linear combination splits in the algorithm. A natural set of split coordinates would be those obtained by successive orthogonally restricted regression of residuals, r, at the M'th order model on the covariates: The linear combination c_1 determining the first split coordinate solves the least squares regression of r on covariates, the linear combination c_2 determining the second split coordinate solves the least squares regression of r on covariates but subject to the orthogonality constraint $c_2'c_1 = 0$ and so on. The relevant formulas are available in Seber [5, pages 84-85]. Algorithm 2 only requires a minor change to incorporate consideration of linear combination splits.

Obviously it would no longer make sense to have a constraint on the order of interaction $k \leq K_m$ but it would perhaps be natural to put a constraint on the number of split coordinates to be examined. The rapid updating formulae in equation (52) don't apply but for split coordinate and knot optimization it should be adequate to compute the lack of fit in the innermost loop of Algorithm 2 by leaving $a_1, a_2 \ldots a_{M-1}$ provisionally fixed and minimizing only over a_M and a_{M+1}. Optimal coefficients can be evaluated after completing the inner loop.

With a modification of this type and with more elaborate function estimation algorithms, the problem arises of how to interpret/visualize the non-parametric regression surface, $\hat{f}$. The output will not be a simple sum of first, second and higher order interaction terms, so the attractive decomposition in equation (24) will not be available. However, numerical integration can of course be used to obtain a decomposition in terms of variables of interest. For example, if the x-variables are split as $x = (x_1, x_2)$ then

$$\hat{f}(x) = \hat{f}_1(x_1) + \hat{f}_2(x_2) + \hat{f}_{12}(x_1, x_2)$$

where $\hat{f}_1(x_1) = \int_{x_2} \hat{f}(x)dx$, $\hat{f}_2(x_2) = \int_{x_1} \hat{f}(x) - \hat{f}_1(x_1)dx$ and $\hat{f}_{12}(x_1, x_2) - \hat{f}_1(x_1) - \hat{f}_2(x_2)$. The percent variance explained by these orthogonal components would be of interest.

A further visualization tool, focusing on isolating local collinearity type effects, could be obtained by apply multivariate statistical density exploration procedures, such as clustering and principal component projections, to the x-distribution associated with specified levels of $\hat{f}$. For example, the analysis of the distribution of x-values for which $a \leq \hat{f}(x) \leq b$ would be of interest. Function visualization is an area where there is a growing need for better

statistical tools.

The MARS algorithm offers considerable power particularly in situations where there are additive low order nonlinear interactions. One of the motivations for MARS given in the paper is dissatisfaction with the lack of continuity in CART. I'll finish by briefly describing an alternative continuous modification to CART which retains some of its algorithmic and interpretative simplicity.

2 Smoothed CART by Finite Elements

The CART model in (17) is the represented as

$$\hat{f}(x) = \sum_{m=1}^{M} a_m B_m(x)$$

where $B_m = I_m$, the indicator function for an n-dimensional rectangular region, R_m. Replace the indicator function by a smooth element $I_m(x,s) \geq 0$ whose support is allowed to extend beyond R_m and define a smoothed CART model by

$$\hat{f}(x;s) = \sum_{m=1}^{M} a_m B_m(x;s) \tag{1}$$

Here $B_m(x;s)$ is forced to satisfy a local partition of unity by setting

$$B_m(x;s) = \frac{I_m(x;s)}{\sum_{m'} I_{m'}(x;s)}$$

so $\sum_{m=1}^{M} B_m(x;s) = 1$. I require that $I_m(x,s) > 0$ for $x \in R_m$. I've introduced a parameter s which gives control over smoothness. Models of the above form connect with mixture models used in the interpretation of multichannel image data, see Adams[1], Choi et al [2] and O'Sullivan [3] for example.

Laplacian finite elements based on triangular grids have been extensively analyzed in the approximation theory literature, see the references in Schumaker[4]. With the rectangular grids of CART, a reasonable choice for $I_m(x;s)$ is defined by tensor products of coordinate functions.

$$I_m(x;s) = \prod_{j=1}^{n} w_{mj}(x_j;s)$$

where $w_{mj}(x_j;s)$ is a smooth non-negative function whose support for $s > 0$ will extend beyond the projection of R_m onto the j'th coordinate. Specifically, suppose the set of split points on the j'th variable are $t_j^{(k)}$ for $k = 1,2,\ldots K_j$ and the projection of R_m is $[t_j^{(k_m)}, t_j^{(k_m+1)})$ then $w_{mj}(\cdot;s)$ can be a B-spline basis element, of any specified order, supported on $[t_j^{(k_m-[s])}, t_j^{(k_m+1+[s])}] \bigcap [t_j^{(1)}, t_j^{(K_j)}]$. Here $[s]$ is the closest integer to s. If we use cubic order elements then $\hat{f}$ will have continuous second order mixed partial derivatives.

The smoothed version of CART, call it SCART ('scairt' is the Irish word for a bush or bushy place!), is easily computed. Let $0 \leq p \leq 1$ be given. The algorithm applies partitioning and pruning as in CART with a couple of minor modifications: (i) For M fixed, the tree predictions are $\hat{f}(\cdot;s)$ with $s = pM$ and the coefficients a_m in (1) optimized by least squares (likelihood can also be used). (ii) At stage M, the selection of the potential split point for R_m is done to improve the local fit. Thus if r are the residuals from the M'th order model then the algorithm just applies the CART splitting rule to components of these residuals lying in R_m. The local support of $I_m(x;s)$ must be exploited for rapid computation of $\hat{f}$. Cross-validation is used to compare trees for different values of p. A preliminary least squares version of SCART with piecewise linear elements was developed

and applied to some of the examples in the paper - those used to compute Tables 3, 5b,6b and 7b. The ISE was evaluated and compared to that achieved by MARS. MARS is a clear winner for the additive model in equation (56) and the additive model with the single low order interaction in equation (61). For example with $N = 200$ the ISE obtained by SCART was on the order of .17 so MARS is 90% better here. SCART wins on the alternating current impedance example in equation (63a) with a 50% or better improvement in the ISE at all sample sizes. A smaller improvement between $10 - 30\%$ is achieved by SCART on phase angle data in equation (63b).

I suppose the message here is that no single adaptive regression technique can perform uniformly best on all examples, which echoes the point made by Professor Friedman in section 2.0.

References

[1] Smith M.O. Adams J.B. and Johnson P. Spectral mixture modelling, a new analysis of rock and soil types at the viking lander 1 site. *J. Geophys. Res.*. 91:8098-8112, 1986.

[2] Choi H.S., Haynor D.R., and Kim Y. Multivariate tissue classification of mri images for 3-d volume reconstruction - a statistical approach. *SPIE Medical Imaging III: Image Processing*, 1092:183–193, 1989.

[3] F. O'Sullivan. Mixture models for multi-channel image data. Technical report, Department of Statistics, University of Washington, 1990.

[4] L.L. Schumaker. Fitting surfaces to scattered data. In Lorentz G.G., Chui C.K., and Schumaker L.L., editors, *Approximation Theory II*, 1976.

[5] G.A.F. Seber. *Linear Regression Analysis*. J. Wiley & Sons, 1976.

Leo Breiman
Department of Statistics
University of California, Berkeley

DISCUSSION

This is an exciting piece of methodology. The highest compliment I can pay is to express my feeling that "I wish I had thought of it". The basic idea is simple and powerful. The examples are interesting and illuminating. My sense of it is that this is a methodology that will become widely used in applications. Naturally, I have a few reservations and questions. But first, I want express my sense of wonderment that this article is published in the Annals of Statistics.

There is not a single theorem, lemma, or proposition in the whole paper. Have my senses taken leave of me? What, no asymptotics or results concerning the rate at which MARS approaches the "True Model" as the sample size goes to infinity? For one of the few times in its history, the Annals of Statistics has published an article based only on the fact that this may be a useful methodology. All is ad hoc; there is no maximum likelihood, no minimax, no rates of convergence, no distributional or function theory. Is nothing sacred? What kind of statistical science is this? My thanks go to the editor and the others involved in this sacrilegious departure.

Now on to issues concerning Friedman's article:

If one fits a linear regression to data, then one is projecting the data onto a fairly small space--the set of all linear combinations of the x-variables. But now, suppose that one has 100 x-variables and perhaps 200 data cases, and we are trying to find the best linear predictor of y based on a subset of the x-variables.

There are billions of different subsets of the x-variables. There are a few standard methods for choosing between these subsets. Forward variable addition can be used, and so can backwards variable deletion, and with enough computing power, the minimum RSS least squares regression equation using a specified number of variables can be found.

The procedure Friedman proposes is analogous to forward variable selection. There are a very large number of variables, consisting of all tensor products of spline functions. Forward stepwise addition is used up to a point, followed by stepwise deletion. The dimensionality of the final model is governed by what Friedman calls "generalized cross-validation", but what is actually an adjusted residual-sum-of-squares, with only a distant connection to true cross-validation.

MARS defines a very large class of candidate models by the specification of a large set of basis elements. Model selection is equivalent to selection of a subset of basis elements, since the coefficients are then defined by least squares regression. For data with 10 variables and a sample size of 100, there are 1000 univariate splines and 450,000 bivariate spline products, where I am counting only splines zero to the left assuming that to each data value of each variable, there is a spline with knot at that point. The number of different candidate models using, say, 10 of these basis elements is staggering.

In principle, the way Friedman would get the "best" of all candidate models is to compute the PSE for all all such models and select the one having the lowest PSE. Since this is not possible, the GCV is used as an estimate for the PSE, and basis elements are found by a stepwise forward addition method. This procedure raises problems which are important not only to MARS , but to the entire venture of fitting more general multivariate models.

I. The set of candidate models in MARS may be too large.

A) The Packing Problem

The predecessor to MARS is TURBO. This is the program for fitting additive models reported on in the Friedman-Silverman 1989 paper. In the discussion of this paper, Trevor Hastie criticized TURBO for having high variability. He generated 50 data sets of sample size 100 from the model:

$$y = .667\sin(1.3x_1) - .465x_2^2 + \varepsilon$$

where ε, x_1, x_2 are N(0,1) and x_1, x_2 have correlation .4. He ran TURBO on this data, and plotted the resulting transformations. These graphs are given below in Figure 1. Later. as I began working with additive models using different construction methods, I understood better what Hastie was driving at.

Figure 1.

Consider the following simple method for constructing additive models--put K knots down on each predictor variable, and using the power basis for splines, do stepwise backward spline deletion. Decide how many splines to leave in the model by finding the minimum value of the cross-validation estimate of PSE.

Initially, I had thought that the value of K would not be critical as long as it was large. For instance, I might typically begin with 15-20 knots per variable and then do the deletion. The reasoning for taking K large was to have plenty of knots around to fit the functions. I thought that having too many would not be a problem since all but a few would be deleted.

Much experimentation later, I realized that I was wrong. If the process was started with K too large, then the resulting models were noisy and could contain odd artifacts due to local quirks in the data. One way to think of this is that the deletion process forms a path through the space of all candidate models. The larger the space of candidate models, the more tightly they are packed together and the path will be forced to select between nearby models on the basis of small local properties. The result was that not only were noisy transformations produced, but also that prediction error increased as K got too large.

The procedure finally selected was this: for each value of K from one on up, set K initial knots on each variable, go through the deletion process and let PE(K) be the minimum cross-validated PSE estimate encountered in the deletion. Now select K to minimize PE(K). This process was carried out using Hasties data, and resulted in the graphs given below. For more details see Breiman [1989a].

Figure 2

The lesson is that relative to some measure of the efficacy of the data the class of candidate models should not be packed too tightly together. Otherwise the results will be noisy, possibly containing local artifacts, and with a loss in prediction accuracy. My concern is that the candidate models in MARS are very tightly packed together. There are many more candidate models than in TURBO. The examples do not seem to show any signs of the packing problem, but we comment further on this below.

The above is really not a criticism particular to MARS . It could apply also to CART and to ACE. It appears to me as a fundamental issue in model fitting. The larger the class one selects from, the more sensitive the procedure is to noise. This issue cries for some theoretical investigation, and deserves at least as much energy and attention as one-dimensional density estimation.

B. The Rashumon Effect

Suppose that we assume that we actually know or can compute the PSE for each model and can go along with the idea that the best model is the one with minimum PSE. From a predictive point of view, this is a perfectly defensible procedure. However, very often predictive procedures are carried out for the purposes of interpretation. Then the question posed is "how well does the estimated function mimic the TRUE function?", or implicitly, "how well can we recover the mechanism used for generating the data?".

Usually, this question is dealt with by setting up simulated data where the true function is known and seeing how well the estimation reproduces the known function. This is the strategy followed in Friedman's examples of sections 4.2 and 4.3. That this works is almost always due to the simple structure of the simulated data. In most cases of complex real data we are up against the "Rashumon Effect".

For instance, consider the "best subsets" procedure in regression for choosing the best regression equation depending, say, on 5 variables, out of 30. If one prints out the residual-sum-of-squares for the, say 10 lowest RSS equations depending on 5 variables, then most often the first few of these will have RSS values within a small smidge of each other. Yet the variables used may be quite different. The analogous effect would take place if we could compute the 10 lowest PSE equations.

The major causes of this lack of uniqueness lies in the sheer size of the class of candidate models and in the dependence between the basis elements. Now, if we assume that a model with low PSE gives a "good" picture of the data generating mechanism, then what we are getting is a multiplicity of equally "good", but different, pictures of what goes on within the black box.

Thus, for complex data, there can be many different and equally valid (or equally invalid) pictures of the mechanism generating the data. Unfortunately, most procedures will produce only one picture: i.e. running MARS on a data set will give only one picture. Yet there may other models based on much different sets of basis elements that give either as low as or lower PSE.

Unfortunately, much of classical statistics is predicated on there being one unique and best answer. The data emanates from a black box, so the idea is to assume a stochastic model for the mechanism in the inside of the black box, estimate a few parameters and bingo, we know what truth is. But for creative data analysis, the desideratum is to get as many different views as possible of what may be going on. Given this, if I were running MARS, then my predilection would be to run it on a number of bootstrap or leave-some-out samples, and see what different results emerged.

II RSS or GCV is not PSE

Another implicit assumption made in many model fitting procedures is that all other things being equal (for instance, in comparing two models both of which use the same number of parameters) that the model with lower RSS will have lower PSE. This is assumed in MARS, since for the same M, the lower RSS model will have lower GCV.

Unfortunately, this assumption is not valid. For the same dimensionality, the minimum RSS model may be quite different than the minimum PSE model, and the PSE corresponding to the minimum RSS model may be considerably higher than the PSE of the minimum PSE model. Is this an inherent and unsurmountable difficulty, or is there some way around it?

MARS uses the GCV values to select dimensionality of the final model. No matter what you call it, the GCV criterion is not cross-validation. The reason for GCV is computational efficiency. Ten-fold cross-validation would take about ten time as long, and MARS is not all that fast to begin with. Friedman has a number of examples showing that his version of GCV does a pretty good job. But I still have some reservations.

For instance, in the example modeling pure noise, about half of the time MARS produces a model that has a better GCV score than estimating the noise by its average. Would using cross-validation improve on this? Again, the problem is interpretation. Fitting noise with some structure can lead to embarrassing conclusions.

Near the end of section 3.6, Friedman puts up a fight for GCV based on simulation results, and claims that "the resulting model and its accuracy are seen to be fairly inde-

pendent of the value chosen for the parameter d". He concludes that the best value for d is between 2 and 4, that 3 is fairly effective, and that the accuracy of the result is not sensitive to the value of d in the 2 to 4 range. This is contrary to my experience in other contexts.

Selecting the dimensionality of the model used is critical. Selecting too large a model leads to inflated variance and too small to lack-of-fit bias. But simulations have shown that GCV in linear regression usually selects too large a model, whereas cross-validation or bootstrap do a good job in selecting the right-sized model (Breiman and Spector[1989]) Therefore, even at increased computational cost, I would suggest that the author include a CV or bootstrap facility in the MARS program.

III Data is not always high signal/noise.

With the exception of the pure noise example, all of the examples given by Friedman had high signal to noise rations. For instance, the example of (56) had s/n=3.28 (91% of variance explained). The example of (61) had s/n=4.8 (96% of variance explained). The circuit examples of Section 4.4 had s/n=3 (90% of variance explained). The olive oil example of section (4.5) had a 3-5% misclassification rate. Finally, the example of equation (66) has s/n=3.15 with 91% of the variance explained. For the other simulated example (67), the s/n ratio was not specified.

Any propensity of MARS to produce artifacts due to the noisy behavior referred to above will be most apparent in moderate-to-low signal/noise ratios. To the extent that Friedman has stayed away from such data, the impression given by the examples in the paper may be misleading.

Even so, there are some disturbing results in the examples. For instance, for the additive data of section (4.2) the number of times that a non-additive model is preferred by GCV increases as the sample size increases. For the data of Section (4.3) with one bivariate interaction, allowing an unlimited number of interactions is about as good as allowing only bivariate interactions. Can the author give explanations for these results?

IV Is stepwise forward the only way to go?

Stepwise forward procedures make me a bit apprehensive. There is always the risk that with a poor step in the initial phases, it will produce a decidedly suboptimal fit. There is a similar problem in CART. While with tree-structured procedures we have been unable to come up with computationally effective alternatives to stepwise forward splitting, in fitting continuous functions to multivariate data there are other methods that have appeared in the literature.

For fitting additive equations, there is the backfitting method used in ACE, with continued research in the Buja,Hastie and Tibshirani[1989] article. Another interesting method using backfitting was proposed by Hastie in his discussion of the Friedman-Silverman [1989] work. There is also the work mentioned above doing backward knot de-

letion (Breiman [1989a]).

There has been less work on fitting interaction surfaces. This is where MARS breaks new ground in being the first published method that has an effective approach to the problem. However, as Friedman points out, the group at Wisconsin is making progress in the computation of interaction splines. There is also another method which depends on the decomposition of the of the function to be estimated into a sum of products of univariate functions (Breiman[1989b]).

This Π-method has given promising results. To illustrate we ran it on the example given in section 4.6, equation(66), which originally appeared in the Chong, et.al.[1988]paper. Figure 3 shows the original function, the interaction spline fit, the MARS fit and the fit of the Π–method.

Figure 3

None of the alternative methods are as fully developed as MARS. The MARS algorithm, with the setting of a few parameters, produces a fit up to whatever degree of interaction is wanted. Whether other methods can provide improved accuracy and comparable automation remains to be seen.

V Quo Vadis? The development and use of effective multivariate methods for fitting complex data is an endeavor largely carried on outside of statistics by diverse and active groups interested in results, rather than theorems. For instance, most of the CART applications that we know about have been done by non-statisticians. The rapidly growing field of neural networks is built around a new class of algorithms for multivariate regression and classification with the principle protagonists being engineers and computer scientists. It was gratifying to find that at a recent neural network conference there was widespread knowlege of CART. I think that MARS will similarly become widely known and used in application areas.

Brieman, L. [1989a] Fitting additive models to data. Technical Report no. 210, Statiostics Department, University of California at Berkeley

Breiman, L. and Spector, P.[1989] Submodel selection and evaluation in regression X-Random case. Technical Report no. 191, Statistics Department, University of California at Berkeley

Breiman, L. [1989b] The Π-method for estimating multivariate functions from noisy data, Technical Report No. 231, Statistics Department, University of California at Berkeley

Figure 1

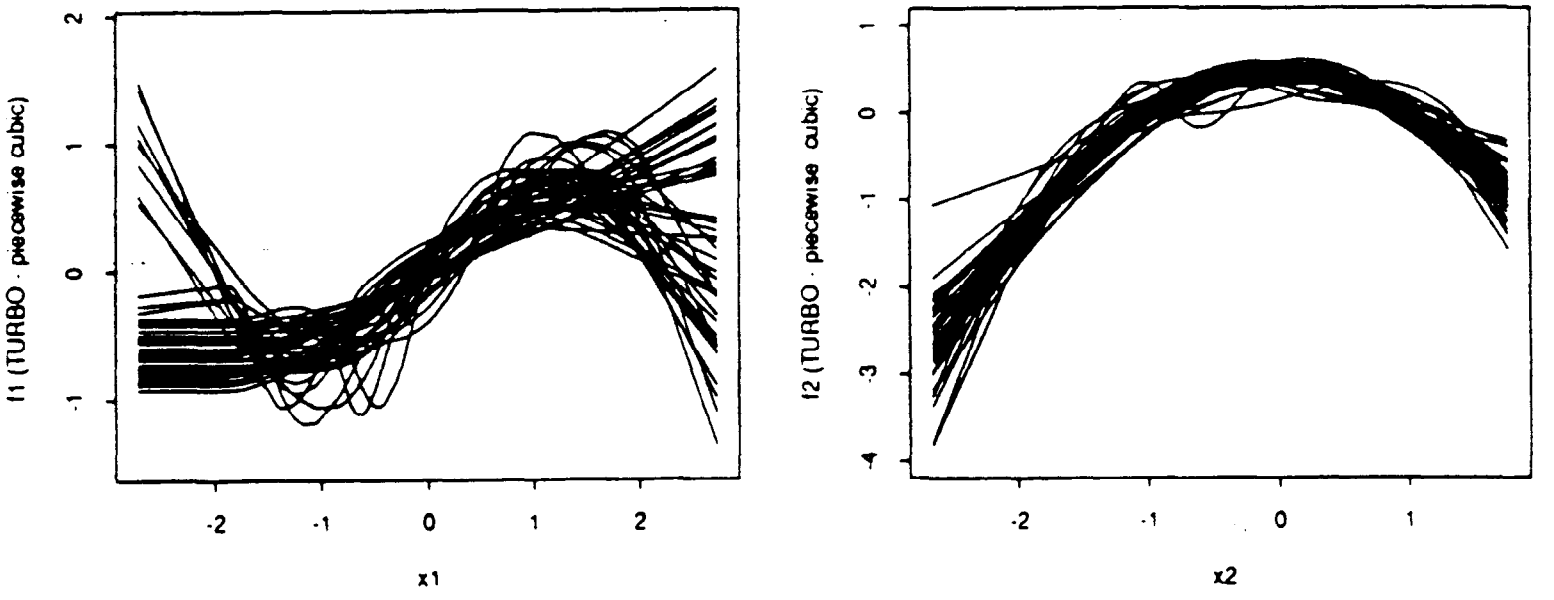


Figure 2

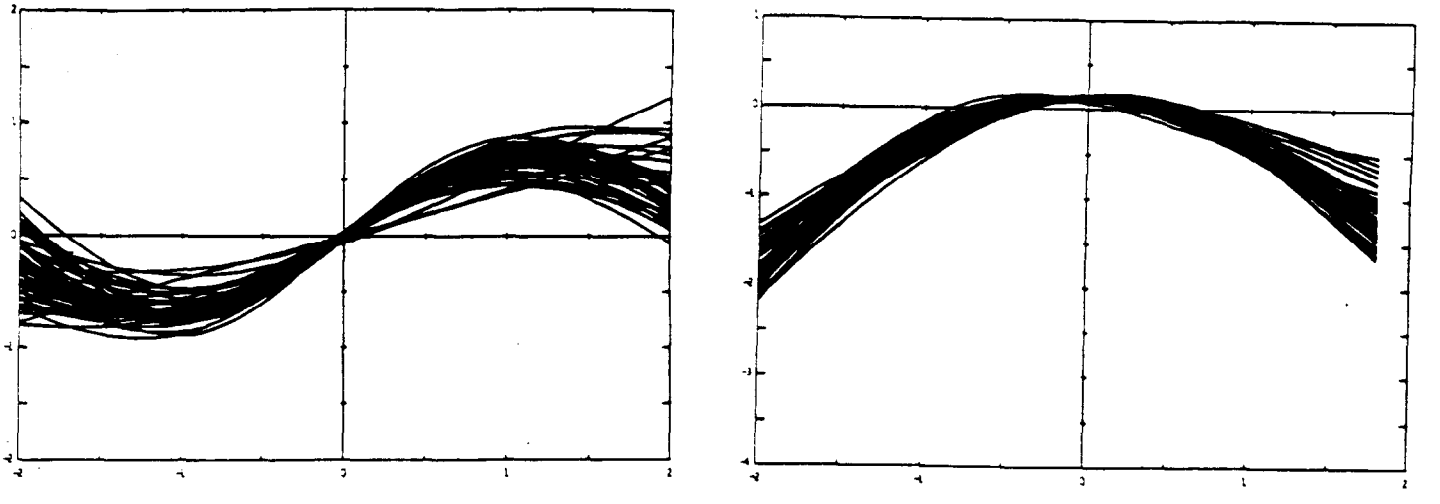
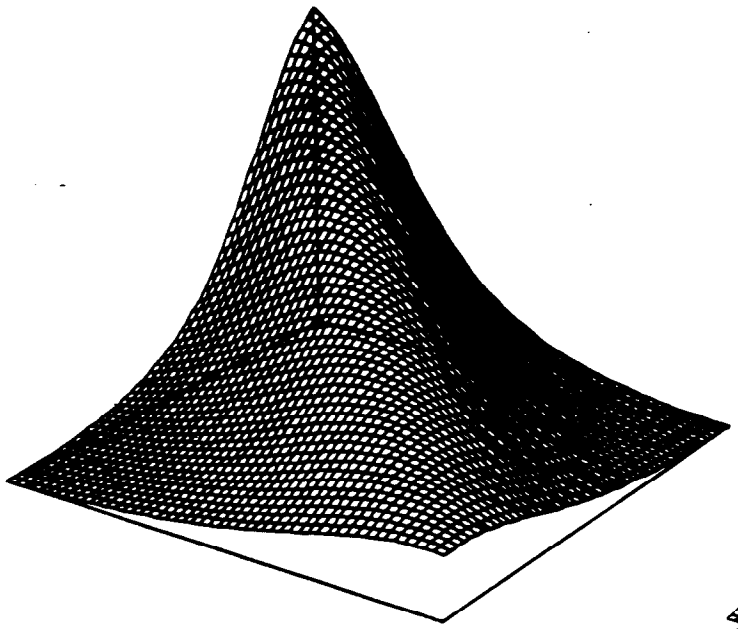
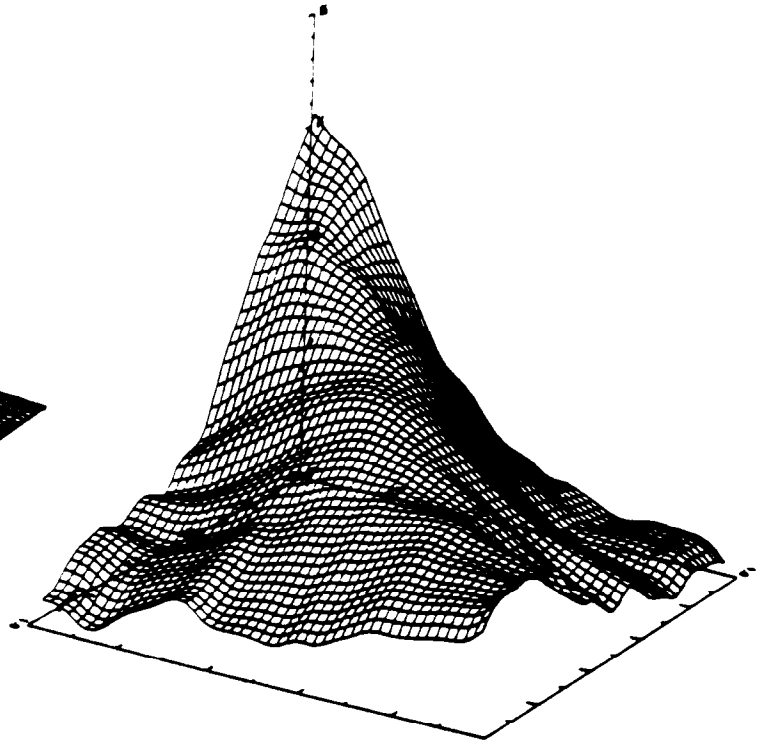


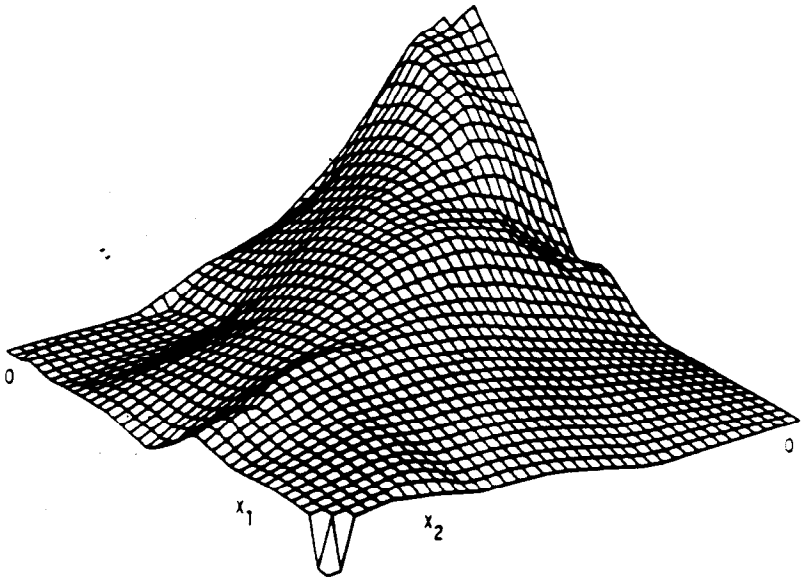
Figure 3



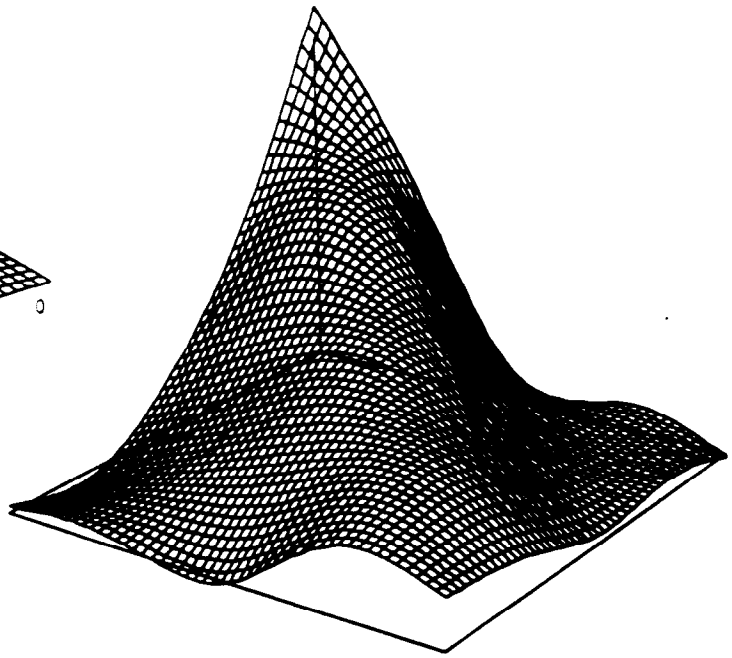
Original Function



Interaction Spline Fit



Friedman's "MARS" Fit



II - Method Fit

George K.Golubev, Rafael Z.Hasminskii

J.H. Friedman presents the new recursive method of regression estimation for high dimensional data. This method is very interesting and has very good perspective. The main idea is an adaptive and recursive construction of the system of basis functions. The proposed estimation method has the good flexibility and it is convenient for computer realization. We think that this approach is applicable for other nonparametrical estimation problems, for instance in the spectral density estimation for stationary Gaussian data.

The interesting problem connected with the proposed method is the theoretical study of quality this method for different classes of smooth regression functions. (The reasons for consideration the classes of smooth functions lie not only in practical importance of such constraints. From our point of view the most important theoretical results are established for these functional classes.) Let us remind some known results in this direction.

1. The best in minimax sense order of the rate of convergence of the L_p , $1 \leq p < \infty$, risks to zero for the regression function of the smoothness of β in R^k is equal to $n^{-\beta/(2\beta+k)}$ (Ibragimov, Hasminskii (1980), Stone (1982)).

2. Speckman (1985), Nussbaum (1985) found regression estimators, which can't be improved not only in the sense of order the rate of convergence but also in the sense of constant. Impossibility of improvement (in minimax sense) of this constant for special case "ellipsoids" in the Sobolev spaces and integrated mean square error was proved by Nussbaum (1985), who used the results of M.Pinsker (1980).

3. Recently Golubev, Nussbaum (1990) have constructed the estimator, having the best constant and the best order for a'priori unknown ellipsoid in the Sobolev space. This estimator is adaptive in this sense, it uses adaptive choice of basis too. As family of bases are used Reinsch, Demmler (1975) bases $\{\varphi_i^{\beta}\}$ for different orders of smoothness β. The estimator has the form

$$\hat{f}_n(x) = \sum_{i=1}^{n} \left[1 - \left(\frac{i}{W} \right)^{\beta} \right]_+ < X_n, \varphi_i^{\beta} > \varphi_i^{\beta}(x)$$

Here

$$\langle X_n, \varphi_i^{\beta} \rangle = \frac{1}{n} \sum_{l=1}^{n} X(t_l^n) \varphi_i^{\beta}(t_l^n)$$

t_l^n , $l = 1,\ldots,n$, is equidistant observation design. The values of q and width "window" W are chosen adaptively on base of data.

We think that the interesting in the theoretical sense question is: have Friedman's estimators or some their modifications analogous asymptotical properties or not?

In the conclusion we would like to repeat that Friedman's estimator is very attractive for applications independently from the answer to this question.

REFERENCES

Demmler A. and Reinsch C. (1975). Oscillation matrices with spline smoothing. Numer. Math., 24, 375-382.

Golubev G. and Nussbaum M. (1990). Adaptive spline estimators in regression model, Probab. Appl. ,35 (to be published).

Ibragimov I. and Hasminskii R. (1980) Asymptotical bounds of quality for nonparametrical regression estimation in L_p . - In : Investigation in the mathematical statistics, III, 97, 88-101.

Nussbaum M. (1985) Spline smoothing in regression models and asymptotic efficiency in L_2. Ann. Statist. 13, 984-997.

Pinsker M. (1980). Optimal filtering of square integrable signals in white Gaussian noise (in Russian). Problems Inform. Transmission, 16, No 2, 52-68.

Speckman P. (1985). Spline smoothing and optimal rates of convergence in nonparametric regression models. Ann. Statist. 13, 970-983.

Stone C. (1982) Optimal global rates of convergence for nonparametric regression. Ann. Statist., 10, 1040-1053.

Institute of Problems of Information
Transmission Ac. Sci. USSR, Ermolovoy,
19 Moscow 101447 USSR

Discussion of "Multivariate adaptive regression splines"

Andreas Buja, Diane Duffy, Trevor Hastie, and Robert Tibshirani

We would like to congratulate Prof. Friedman on this characteristically ingenious advancement in non-parametric multivariate regression modelling. MARS is a triumph of statistical computing and heuristics: the clever algorithmic and heuristic ideas make extensive searching computationally feasible. The resulting modeling technology offers the data analyst a remarkably flexible tool which we found very useful on a difficult real-world problem. We will address a few issues that arose in our reading of this excellent paper and our experience using the MARS program.

1. Some experience with MARS

Two of us (Buja and Duffy) acquired some experience with MARS in an extensive analysis of data concerning memory usage in electronic switches. The data comprised 241 observations on 27 variables. It was known from the onset that the available variables gave an incomplete description of the response. Careful and creative regression modeling yielded fits with good global properties ($R^2 \approx 0.995$), but there were still unacceptably large residuals and poor performance on cross-validation tests. The fits which we obtained from MARS, on the other hand, excelled in prediction and cross-validation. In addition, the robustness to influential points which MARS inherits from the local adaptivity of the selected basis functions was very advantageous in our context. Our set of observations was (purposely) chosen to include a subset consisting of notoriously difficult cases. These cases, as expected, wreaked havoc on regression models but MARS was able to adapt to them without degrading the fits to the rest of the cases. In addition, highly accurate MARS models could be built with fewer variables (13 as opposed to 18) which happens to be a true benefit in this situation. The MARS models involved several second and third order interactions which, while impossible to anticipate by subject matter experts, seemed reasonable in the sense that they involved variables which are expected to have large effects on the way memory is used.

An interesting aspect of this analysis is that the data exhibit genuine noise despite the fact that switches are basically deterministic systems. This is because the 27 predictors were selected from a complete set of about 300 predictors based on what information is available to engineers who operate these systems. The success of the MARS fits can only be explained as a result of strong interdependence within the large set of predictors, rendering most of them redundant. Thus, we are profiting from what we call 'concurvity' which in most contexts is a cause for concern. Further, models based on the theory of the switching systems would necessarily involve many of the 300 predictors and would therefore be useless to the engineers. One danger in this, or any, data-driven as opposed to theory-driven approach is that the model may be misleading if future predictions occur in areas of the predictor space where data are sparse. It would be useful if MARS were accompanied by diagnostics tools which indicated when a future set of covariate values is stepping dangerously outside the range of the training data. A first naive attempt at deriving such a tool would be to compute the Mahalanobis distance of test covariate vectors in the linear predictor space spanned by the basis functions of a given MARS model. However, such an approach may have problems since the constant zero stretches of the spline basis functions lead to clumps of data in the extended predictor space.

We found it quite useful that the first order truncated basis splines are of an exceedingly simple form. A fitted model is easily communicated to practitioners and it is trivial to implement on arbitrarily small machines. By comparison, we do not see a use for the enhanced cubic models in this (prediction) context. For graphical display and qualitative data analysis, they may have their advantages.

In using MARS to analyze our data the following questions and comments arose.

1. Friedman recommends running MARS with $M_{\max}$ (the maximum number of basis functions added in the forwards stepwise procedure) approximately equal to $2M^*$ where M^* is the GCV-minimizing choice for the number of basis functions in the model. In our context M^* is in the neighborhood of 35–40. Based on our experience with honest cross-validation, this is too large for the sample size and it may indicate that, at least for these data, the default cost being charged for basis function optimization is probably too low.
2. It appears that no complete description of the heuristic choice of the cost parameter d is given. If there is no restriction on the degree of interaction ($mi = n$), we understand that the default value is $d = 3$. The question for which we were unable to find an answer was: how does d depend on mi, the maximal degree of interaction, if it is specified to be less than n? When the degree of interaction is limited ($mi < n$) d is, quite logically, decreased. We chose in one instance $mi = 5$, which seemed to result in a value of d closer to 2 than 3.
3. As mentioned in 2., the cost d is set to 3.0 when unlimited interactions are permitted. Based on our calculations, it appears that the value of d is also being adjusted based on M_{max}. How exactly does it affect d? On one occasion, we observed an apparent oddity in the behavior of d which seems counterintuitive: d can decrease as M_{max} increases with mi (the maximum permitted degree of interaction) held fixed.
4. One of the startling features of our MARS runs is the fact that the piecewise cubic GCV values are often an order of magnitude larger than the corresponding piecewise linear GCV values. In addition, there is little correspondence between the piecewise linear and the piecewise cubic GCVs for these data. For example, the model which minimizes the piecewise linear GCV has an associated piecewise cubic GCV which is over three times larger than the piecewise cubic GCV of another model; this other model, however, has a piecewise linear GCV which is almost twice the minimal value. Further, the two models are very different; the first has 35 basis functions and interactions up to the third degree whereas the second has only 21 basis functions and all are restricted to be main effects only. Hence, if a smooth (piecewise cubic) model had been our ultimate objective, we would have been led very far astray by basing the model choice on minimizing the piecewise linear GCV. We can see at least three reasons for the unpredictable behavior of the piecewise cubic modifications: First, the residual sum of squares is very nonrobust and responds dramatically to a few bad residuals. Second, in high-dimensional predictor spaces and in the presence of higher order interactions, the seemingly inocuous piecewise cubic modification is far from minor because of compounding effects in the products. And third: our data may very well be better described by piecewise linear functions due to threshold effects which we observed while performing graphical exploration of the data.
5. In fitting a series of models with increasing values of M_{max}, the number of basis functions in the final model grew quite unevenly. While this might be expected for choices of M_{max} which produce poorer fitting models, it was surprising to us for a choice of M_{max} yielding near optimal models (i.e., models with piece-wise linear GCV values near the minimum). We are unsure whether to interpret the widely varying numbers of basis functions as an artifact of our data or as a property of the MARS methodology.

2. Generalized Mars models

Friedman proposes using MARS for logistic models. This can, of course, be easily extended to include all generalized linear models. The standard method for fitting such models is to maximize the likelihood or, equivalently, to minimize the deviance. While it would be natural to use the penalized deviance as the criterion for knot inclusion or deletion in direct analogy to the penalized RSS or LOF used in the present paper, this is computationally impractical because iteration is required to estimate the parameters and the crucially important ability to rely on the updating formulae is lost. Consequently Friedman offers an approximation and here we offer another.

Suppose the basis set has k members, and we wish to find the $(k+1)$st. The exact inclusion of a candidate b_{k+1} can be achieved by using an iteratively reweighted least squares algorithm, with the initial values and working response provided by the fit to the set of size k. Instead of iterating to convergence, we propose using one iteration, and instead of using the deviance to evaluate the fit, we propose using the weighted residual sum of squares or Chi-squared approximation to the deviance. Since the fits for all candidates for b_{k+1} use the same use the same working response and the same weights, Friedman's entire updating approach carries over. Once the approximately optimal b_{k+1} has been selected, the corresponding iterations can be completed to estimate the associated coefficients.

This approximation for evaluating a candidate b_{k+1} is exactly that used in Rao's score test (Pregibon, 1980) with the additional advantage that we exploit the updating facility to simultaneously perform multiple score tests.

3. ANOVA decomposition

The ANOVA decomposition is achieved in MARS by grouping together all terms involving the same variables. Thus all the functions involving only X_1 comprise the main effect for X_1, all terms involving only X_1 or X_2 the interaction for X_1 and X_2, and so on. The usual ANOVA decomposition for categorical designs ensures that interactions are free of lower order effects by imposing suitable summation constraints. Note that the tendency for these surfaces to include lower order effects in MARS is exacerbated because MARS can destroy the hierarchical structure of its basis during knot deletion. It would be useful if MARS could produce an interaction surface which was free of lower order effects. One could then use this surface to asess the way in which the variables interact, without being distracted by the lower order effects.

Hastie and Tibshirani (1990, page 266) propose a strategy for this which can be adapted to MARS. As in standard ANOVA, one needs only to uncouple the components in the fitted model, not during the fitting (unless one requires an a priori hierarchy in the terms). Let us focus first on a bivariate interaction term, say $f(X_1, X_2) = \sum_k \alpha_k b_{1k}(X_1) b_{2k}(X_2)$. We first identify all the univariate basis functions in each of the tensor products pairs. In this case they are the b_{1k} and b_{2k}. We then project the interaction surface onto the joint space defined by them and the other main-effect basis functions involving those two variables. This additive main effect component of the interaction is then removed. and lumped together with the original main effects, leaving a residual component which can be interpreted as a pure interaction and which is orthogonal to these (new) main effects. It is important to stress that the fit of the model has not been changed during this operation, simply its ANOVA decomposition.

Of course, if higher order interactions are present, this procedure would have to be used in a top down fashion. It is not entirely obvious how this would proceed. For example, if the term in question is a 3rd order interaction, then we should isolate all bivariate interaction basis pairs. These would be grouped with similar and lower order terms involving the same variables, and the entire set used to remove the second order effects from the 3rd order interaction.

Incidently, in the simple linear regression model, $Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \gamma X_1 X_2$, we would not need to do all of this to understand the interaction effect. Decomposing such a fit amounts to

looking at the coefficients (their sign and magnitude). Although these are fitted jointly, we know from the Gram-Schmidt process that $\hat{\gamma}$ is also the coefficient of X_1X_2 adjusted for X_1 and X_2. This would not be the case if the model were $Y = \alpha + \beta_1 X_1 + \gamma X_1 X_2$.

4. Shrinking versus knot deletion

In a discussion of the additive predecessor to MARS, called TURBO, Hastie (1989) outlined in some detail a method for "shrinking" a TURBO model. The idea is that the forward stepwise algorithm results in a rich set of knots/basis functions for each additive term. In particular, the knots will be denser on some variables, and locally, within a variable, there may be clusters of knots in regions of high signal to noise ratio. At this point the model is (purposefully) overparametrized, and some regularization is needed. As an alternative to backward knot deletion, Hastie suggested regularizing by shrinking according to an appropriate smoothness penalty. He suggested the penalty $\sum_j \lambda_j \int (f_j''(x_j))^2 dx_j$, and pointed out that the resulting procedure is a generalized ridge regression. Furthermore, since the second derivatives of the piecewise-cubic approximations to the piecewise-linear basis functions have local support, the ridge penalty matrix is diagonal.

With an appropriate set of penalty functionals, a similar approach can be taken with MARS. Wahba (1990, chapter 10) outlines in some detail an approach using tensor-product splines, which are exactly what MARS uses to build up its bases. In Wahba's setting, models are fit in subspaces of the tensor product space of all the univariate reproducing-kernel Hilbert spaces, and the penalty functionals (norms) of these subspaces are inherited from the univariate spaces. For example, functions involving only X_1 and X_2 would be penalized using $\int\int (\partial^4 f(x_1, x_2)/\partial x_1^2 \partial x_2^2)^2 dx_1 dx_2$. In practice then, the terms are grouped according to their components (much like the ANOVA grouping in MARS), each group gets assigned an appropriate penalty (and potentially its own smoothing parameter), and then the fit is computed by penalized least squares. Thus suppose the MARS model after the stepwise inclusion stage can be written

$$f(x_1, x_2, \ldots, x_p) = \alpha + \sum_{k \in I_1} f_k(x_k) + \sum_{(l,m) \in I_2} f_{(lm)}(x_l, x_m) + \ldots$$

where I_j denotes the sets of j-tuples corresponding to interactions of order j. Each of the f_* has a linear representation in terms of an appropriate set of tensor-product bases. Then the shrunken model is the minimizer of the penalized criterion

$$\sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2 + \sum_{k \in I_1} \lambda_k P_k f_k) + \sum_{(l,m) \in I_2} \lambda_{lm} P_{lm}(f_{lm}) + \ldots$$

where the P_* are the penalty functionals.

Without going into all the details, it is worth pointing out that each of the $P_*(f_*)$ evaluates to a quadratic form $\boldsymbol{\alpha}_*^T M_* \boldsymbol{\alpha}_*$ in the coefficients for the basis functions in f_*; if the cubic approximations to the piecewise-linear functions are used, then each of the M_* is diagonal, and once again the problem is a generalized ridge regression. We have used a different regularization parameter λ_* for each of the components above. In practice one could simply use a single global λ, and trust that the forward knot selection will give some terms more importance than others. Alternatively, one could lump all terms of the same interaction order together with the same penalty, and shrink them all at the same rate.

This is a current research project (Friedman and Hastie), and we are experimenting with other strategies and penalty functionals.

5. MARS for classification

One of us (Tibshirani) has made some progress in the development of a methodology for classification that tries to combine some of the features of MARS and CART. Consider a two class problem with $Y = 0$ or 1.. The working model is

$$\log \frac{Pr(Y=1\,|\mathbf{x})}{1-Pr(Y=1\,|\mathbf{x})} = \alpha + \sum_{k\in I_1} f_k(x_k) + \sum_{(l,m)\in I_2} f_{lm}(x_l, x_m) + \ldots$$

where the I_js are as defined in the previous section, but where the fs are tensor products of order 0, that is, products of indicator functions of the form $(x_j - t)^+$ and $(x_j - t)^-$. The motivation for 0 order splines is, as in CART, the ease with which models can be substantively interpreted when terms are of this form.

The model estimation is carried out as follows:

1. The model is constructed in a forward stepwise manner, exactly as in MARS. A score test (analogous to that described in Section 2 above) is used to select the "split point" for each variable. In contrast to CART, a basis function is not removed from the model after it has been "split", thereby encouraging main and lower order effects to appear.
2. The model is pruned in a backwards hierarchical fashion, similar to the pruning in CART. In detail, a pruning operation consists of deleting a pair of functions of the form $b(\mathbf{x})(x_j - t)^+$ and $b(\mathbf{x})(x_j - t)^-$ and any higher order terms in which they appear. k-fold cross-validation (of the entire estimation procedure) is used to determine the optimal amount of pruning. In contrast to CART, we favor the use of deviance rather than misclassification cost to guide the pruning; this enhances the interpretability of the final model.

The result of this process is an estimated binary logistic model with a linear predictor that is a sum of products of indicator functions. In addition, a global (cross-validation) estimate of its classification performance is available, which is hopefully a more accurate estimate of future performance than GCV estimates, such as those used by MARS, which do not involve cross-validating the term selection process. The model can be interpreted either in terms of its basis functions or of the binary partition of the predictor space that they define. Initial studies suggest that this procedure is more effective (as a descriptive tool) than CART in cases where main effects dominate. On the other hand, in general it does not seem to classify as well as CART does. Extensions to ordered and unordered multiple class problems are possible. Further details will appear in a forthcoming technical report.

revised June 5, 1990

Comments on MARS by Jerome Friedman

Chong Gu* Grace Wahba†

We would like to begin by thanking Professor Friedman for a very interesting and thought provoking paper. The idea of combining splines with recursive partitioning ideas is clearly an idea whose time has come, and this paper is sure to generate much interest!

We have a number of comments that fall into several areas.

1 Nonparametric function estimation is a rich field

Of course, the general area of nonparametric function estimation is a rich and growing one. The major methods in one variable are, kernel methods, smoothing spline methods, regression spline methods, orthogonal series methods, and nearest neighbor methods. When the data points are uniformly spaced, if these methods have their various tuning parameters matched up they are quite similar for medium sized data sets, and can essentially be tuned so that they have the same convergence rates for functions with the same number of (square integrable, say) derivatives. Even when the data points are not uniform, kernel methods and spline methods can be shown to be similar under certain circumstances. As soon as we get into more than one dimension, however, choices proliferate, and results are not necessarily so similar. Narrowing our consideration to kernel estimates, smoothing splines and regression splines, we may, at the outset, consider what might be called a "tensor product" structure versus a "thin plate" structure – we will define these by example. Considering two variables, $\boldsymbol{x} = (x_1, x_2)$, and given a knot $\boldsymbol{t} = (t_1, t_2)$, a basis function of tensor product type is of the form $B_{t_1,t_2}(x_1, x_2) = H(x_1 - t_1)H(x_2 - t_2)$, where H stands for a generic function, usually depending on some order parameter q, whereas a basis function of thin plate type is of the form $B_{t_1,t_2}(x_1, x_2) = H(\|\boldsymbol{x}-\boldsymbol{t}\|)$ where $\|\boldsymbol{x}-\boldsymbol{t}\|$ is the Euclidean distance between

*Department of Statistics, University of British Columbia, Vancouver, British Columbia V6T 1W5, Canada.
†Department of Statistics, University of Wisconsin, Madison, WI 53706.

$x = (x_1, x_2)$ and $t = (t_1, t_2)$. Thin plate splines do not know the difference between north and east, whereas tensor product splines do. Of course kernels as well as regression splines come in both types. Friedman's splines are regression splines of tensor product type with a sophisticated procedure for choosing the number, and order(s) of the spline basis functions and the knots. It of course equally possible to do regression splines on thin plate basis functions. See Poggio and Girosi (1990) who discuss regression thin plate splines with moveable knots in the context of neural nets for multidimensional function estimation. Which type one might prefer would certainly be related to the nature of the variables one is dealing with. In principle it is quite possible to mix the various types of basis functions. However Friedman's recursive partitioning approach to knot selection fits naturally into the tensor product setup and probably not in the thin plate setup (since there are not natural cutting planes in that case), while Poggio and Girosi's approach appears to fit in to the thin plate case, and not the tensor product setup. In both these cases knot selection is a non-trivial operation, and clearly as time goes on further insight will be gained.

2 Smoothing splines with multiple smoothing parameters

We agree with Friedman that automatic selection of multiple smoothing parameters is inherently difficult and computationally consuming. Nevertheless, our experience is that it is feasible for relatively small number of covariates and medium sample sizes on modern workstations. Recall that given responses y_i and covariates x_i where $y_i \sim p(y; \eta(x_i))$, a smoothing spline regression fit with multiple smoothing parameters is the solution to the problem: Find $\eta \in \mathcal{H}$ to minimize

$$-\sum_{1}^{n} l_i(\eta(x_i)) + (n/2)\lambda \sum_{\beta=1}^{p} \theta_\beta^{-1} J_\beta(f_\beta) \tag{2.1}$$

where $l_i(\eta) = \log p(y_i; \eta)$, $\eta = \sum_{\beta=0}^{p} f_\beta$, and $\mathcal{H} = \mathcal{H}_0 \oplus \mathcal{H}_1 \oplus \cdots \oplus \mathcal{H}_p$ is a reproducing kernel Hilbert space with reproducing kernel $R = R_0 + R_1 + \cdots + R_p$; see Wahba (1990). Here x_i may be quite general consisting of several components, in arbitrary index sets. $\mathcal{H}_0$, on which there is no penalty, is of necessity of finite dimension M, say, which is less than n, and as λ tends to infinity the estimate tends to the maximum likelihood estimate in $\mathcal{H}_0$. f_β is the component (projection) of η in $\mathcal{H}_\beta$ and J_β is a suitable quadratic penalty, which we will take as the squared norm in $\mathcal{H}_\beta$. The elements in each $\mathcal{H}_\beta$ may actually depend on all or only a few of the components of x. It can

be shown that the solution of (2.1) has an expression

$$\eta(\boldsymbol{x}) = \sum_{\nu=1}^{M} \phi_\nu(\boldsymbol{x}) d_\nu + \sum_{i=1}^{n} (\sum_{\beta=1}^{p} \theta_\beta R_\beta(\boldsymbol{x}_i, \boldsymbol{x})) c_i = \boldsymbol{\phi}^T(\boldsymbol{x})\boldsymbol{d} + \boldsymbol{\xi}^T(\boldsymbol{x})\boldsymbol{c}, \tag{2.2}$$

where $\{\phi_1, \cdots, \phi_M\}$ span $\mathcal{H}_0$, $\boldsymbol{\xi}^T(\boldsymbol{x}) = (\xi_1(\boldsymbol{x}), \cdots, \xi_n(\boldsymbol{x}))$, $\xi_i(\boldsymbol{x}) = \sum_{\beta=1}^{p} \theta_\beta R_\beta(\boldsymbol{x}_i, \boldsymbol{x})$, and $\boldsymbol{c}$ and $\boldsymbol{d}$ are the minimizers of

$$-\sum_{i=1}^{n} l_i(\boldsymbol{\phi}^T(\boldsymbol{x}_i)\boldsymbol{d} + \boldsymbol{\xi}^T(\boldsymbol{x}_i)\boldsymbol{c}) + (n/2)\lambda \sum_{\beta=1}^{p} \theta_\beta \boldsymbol{c}^T Q_\beta \boldsymbol{c}, \tag{2.3}$$

where Q_β is a $n \times n$ matrix with (i,j)th entry $R_\beta(\boldsymbol{x}_i, \boldsymbol{x}_j)$. ξ_i's in the smoothing spline examples have knots at the data points $\boldsymbol{x}_i$, and under certain circumstances they span the same space as commonly used local bases such as the B-splines on the real line. Let S be the matrix with (j,ν)th entry $\phi_\nu(\boldsymbol{x}_i)$, $Q = \sum_{\beta=1}^{p} \theta_\beta Q_\beta$, $W = \mathrm{diag}(w_1, \cdots, w_n)$ where $w_i = -dl^2 l_i / d\eta_i^2$, and $\boldsymbol{u} = (u_1, \cdots, u_n)$ where $u_i = -dl_i/d\eta_i$. The Newton iteration for minimizing (2.3) proceeds by solving

$$\begin{aligned} W^{1/2}(Q + n\lambda W^{-1})\boldsymbol{c} + W^{1/2} S\boldsymbol{d} &= W^{1/2}\tilde{\boldsymbol{y}} \\ S^T\boldsymbol{d} &= 0, \end{aligned} \tag{2.4}$$

where $\tilde{\boldsymbol{y}} = \boldsymbol{\eta}_0 - W^{-1}\boldsymbol{u}$, $\boldsymbol{\eta}_0$ is the fit at the previous step, and W and $\boldsymbol{u}$ are evaluated at $\boldsymbol{\eta}_0$. See, e.g., Gu (1990a). This setup provides a unified numerical treatment for broad varieties of nonparametric regression problems with either Gaussian or non-Gaussian responses and with all kinds of covariate structures; see the next section. Generic algorithms for solving (2.4) with automatic smoothing parameters λ and θ's appear in Gu *et al.* (1989) and Gu and Wahba (1988) mentioned by Friedman. Transportable code is available from the netlib under the name RKPACK; see Gu (1989). We feel comfortable with these algorithms for n up to 500 and p up to 6 on the contemporary workstations, and with smaller p we can afford larger n.

3 ANOVA decomposition and varieties of subspaces

A nice feature of the MARS product is the ANOVA decomposition which greatly enhances the interpretability of the computed fit. It is known that the same structure can also be obtained via interaction smoothing splines, which are important specializations of (2.1). Recall that for a bivariate covariate $\boldsymbol{x} = (x_1, x_2)$ on a domain $\mathcal{T}_1 \times \mathcal{T}_2$, given reproducing kernel Hilbert spaces

$\mathcal{H}^i = \mathcal{H}_0^i \oplus \mathcal{H}_1^i$ of functions on $\mathcal{T}_i$, $i = 1, 2$, the tensor product Hilbert space of functions on $\mathcal{T}_1 \times \mathcal{T}_2$ has a tensor sum decomposition $\mathcal{H} = \mathcal{H}^1 \otimes \mathcal{H}^2 = (\mathcal{H}_0^1 \otimes \mathcal{H}_0^2) \oplus (\mathcal{H}_1^1 \otimes \mathcal{H}_0^2) \oplus (\mathcal{H}_0^1 \otimes \mathcal{H}_1^2) \oplus (\mathcal{H}_1^1 \otimes \mathcal{H}_1^2)$. Assuming finite dimensional $\mathcal{H}_0^i$, $i = 1, 2$, an interaction spline is obtained by specializing (2.1) with $\mathcal{H}_0 = \mathcal{H}_0^1 \otimes \mathcal{H}_0^2$, $\mathcal{H}_1 = \mathcal{H}_1^1 \otimes \mathcal{H}_0^2$, $\mathcal{H}_2 = \mathcal{H}_0^1 \otimes \mathcal{H}_1^2$, and $\mathcal{H}_3 = \mathcal{H}_1^1 \otimes \mathcal{H}_1^2$. When $\mathcal{H}_0^i = \{1\}$, $i = 1, 2$, the setup provides an ANOVA decomposition of the estimate by construction. When we take the tensor sum or tensor product of any two reproducing kernel spaces we just add or multiply their reproducing kernels to get the reproducing kernel of the resulting space. Obviously the tensor sum on each coordinate can take more than two operands and so can the tensor product.

Examples of interaction splines in subspaces of tensor products of $W_2^m[0, 1]$'s appear to be the most popular in the existing literature. However, the foregoing general framework covers a much broader spectrum of model specifications. Consider a covariate x on a categorical domain $\mathcal{T} = \{1, \cdots, C\}$. A real function on $\{1, \cdots, C\}$ is just a real vector in the Euclidean space R^C. Adopting a roughness penalty proportional to the Euclidean norm of the projection of a vector onto $\{\mathbf{1}\}^\perp$, a "smooth" vector would then be a one with a small variance. The corresponding Hilbert space decomposition is $R^C = \{\mathbf{1}\} \oplus \{\mathbf{1}\}^\perp$ with a reproducing kernel $R(x, x')$, $x, x' \in \{1, \cdots, C\}$, representable as a $C \times C$ real matrix, $\mathbf{1}\mathbf{1}^T/C + [I - \mathbf{1}\mathbf{1}^T/C]$, where the term in brackets is the reproducing kernel for $\{\mathbf{1}\}^\perp$. Using R^C for categorical (nominal) covariates in the construction of tensor product Hilbert space, one can incorporate both continuous and categorical covariates simultaneously to build a model with a natural ANOVA decomposition. We understand from Friedman's talk at Interface 90 that categorical covariates can also be incorporated into the MARS framework. It would be interesting to compare the two approaches.

We have discussed the "tensor product" structure versus the "thin-plate" structure in Section 1. In general a "tensor product" structure is appropriate for combining individually interpretable covariates and a "thin-plate" structure is appropriate for dealing with rotation invariant problems. The example below indicates that certain problems require mixed structures. From the Eastern Lake Survey of 1984 implemented by the Environment Protection Agency of the United States, a data set has been derived by Douglas and Delampady (1990) which contains geographic information, water acidity measurements, and main ion concentrations of 1798 lakes in four regions in the Eastern United States. An attempt is made to explore the dependence of the water acidity on the geographic locations and other information concerning the lakes. Preliminary analysis and

consultation with a water chemist suggest that a model for the surface pH in terms of the geographic location and the calcium ion concentration is appropriate. Obviously, a "thin-plate" structure is appropriate for the geographic location. To account for the joint effect of geographic location and the calcium concentration, however, a "tensor product" structure appears to be appropriate. A tensor product reproducing kernel Hilbert space with a "thin-plate space" component for the geographic locations and a W_2^2 component for the calcium concentrations does the job simply. This example actually illustrates the fact that the general framework of interaction splines can paste up arbitrarily complicated components to provide an interpretable ANOVA decomposition.

To use a thin plate spline as a component of an ANOVA model as we have just described, one needs an explicit reproducing kernel. (Only a so-called "semi-kernel" is needed for the construction of a thin plate spline by itself, see the references in Wahba (1990).) An explicit reproducing kernel appears in Wahba and Wendelberger (1980) but it is not a natural one to use in an ANOVA model. We will provide a more natural one here. Let $\mathcal{X}$ be the thin plate function space (in two variables) consisting of linear functions plus all functions (modulo the linear functions) for which the thin plate penalty functional

$$J(f) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(f_{uu}^2 + 2f_{uv}^2 + f_{vv}^2)\,du\,dv$$

is well defined and finite. Now let $\boldsymbol{t}_j$, $j = 1, \cdots, K$ be any set of $K \geq 3$ points in R^2 not falling on a straight line. Let $\phi_1(\boldsymbol{x}) = 1/\sqrt{K}$ and let ϕ_2 and ϕ_3 be linear functions satisfying $\sum_{j=1}^K \phi_\mu(\boldsymbol{t}_j)\phi_\nu(\boldsymbol{t}_j) = 1$, $\mu = \nu$, 0, otherwise, and let $w_j(\boldsymbol{x}) = \sum_{\nu=1}^3 \phi_\nu(\boldsymbol{t}_j)\phi_\nu(\boldsymbol{x})$. If we endow $\mathcal{X}$ with the squared norm

$$\|f\|_{\mathcal{X}}^2 = \sum_{\nu=1}^{3}(\sum_{j=1}^{K}\phi_\nu(\boldsymbol{t}_j)f(\boldsymbol{t}_j))^2 + J(f),$$

it can then be shown that the reproducing kernel for $\mathcal{X}$ is

$$\begin{aligned} R(\boldsymbol{x},\boldsymbol{x}') &= \sum_{\nu=1}^{3}\phi_\nu(\boldsymbol{x})\phi_\nu(\boldsymbol{x}') + \\ &\quad [E(\boldsymbol{x},\boldsymbol{x}') - \sum_{j=1}^{K} w_j(\boldsymbol{x})E(\boldsymbol{t}_j,\boldsymbol{x}') - \sum_{k=1}^{K} w_k(\boldsymbol{x}')E(\boldsymbol{t}_k,\boldsymbol{x}) + \sum_{j=1}^{K}\sum_{k=1}^{K} w_j(\boldsymbol{x})w_k(\boldsymbol{x}')E(\boldsymbol{t}_j,\boldsymbol{t}_k)], \end{aligned}$$

where $E(\boldsymbol{x},\boldsymbol{x}') = (\|\boldsymbol{x}-\boldsymbol{x}'\|^2 \log\|\boldsymbol{x}-\boldsymbol{x}'\|)/(8\pi)$. The term in brackets is the reproducing kernel for $\mathcal{H}_1$ if $\mathcal{H}_0$ is taken as the span of the linear functions. Furthermore each element f of $\mathcal{H}_1$ satisfies the discrete orthogonality conditions $\sum_{j=1}^K \phi_\nu(\boldsymbol{t}_j)f(\boldsymbol{t}_j) = 0$, $\nu = 1,2,3$. It is natural to choose $K = n$ and the $\boldsymbol{t}_j$'s at the data points. With this construction thin plate splines can be included

in a nonparametric ANOVA model, possibly by moving span $\{\phi_2, \phi_3\}$ into $\mathcal{H}_1$. This construction extends to higher order and higher dimensional thin plate splines in a straightforward way.

4 Model selection and concurvity

Model selection in MARS is built into the step-wise estimation procedure by choosing bases and knots to minimize an intuitive GCV score. In a smoothing spline setup (2.1), model selection is by selective inclusion of subspaces and selection of smoothing parameters. Step-wise nonparametric estimation procedures are likely to be confused by concurvity (collinearity), for which Friedman proposes several cures in his algorithms. On the other hand, the direct approach of Section 2 estimates all terms simultaneously and is not bothered by negligible terms and concurvity. For the sake of parsimony and interpretability, however, one wishes to remove aliasing effects and noise terms in a fit. Recently Gu (1990b) proposed simple geometric diagnostics to tackle the problem. Recall that the solution $\hat{\eta}$ of (2.1) also minimizes

$$\sum_1^n w_i(\tilde{y}_i - \eta(\boldsymbol{x}_i))^2 + n\lambda \sum_{\beta=1}^p \theta_\beta^{-1} J_\beta(f_\beta), \tag{4.1}$$

where the $\tilde{y}_i$'s and the w_i's are those of (2.4) evaluated at $\hat{\eta}$; see, e.g., Gu (1990a). Evaluating the computed fit on the data points, one obtains a retrospective linear model

$$W^{1/2}\tilde{\boldsymbol{y}} = W^{1/2}(\tilde{\boldsymbol{f}}_0 + \tilde{\boldsymbol{f}}_1 + \cdots + \tilde{\boldsymbol{f}}_p + \tilde{\boldsymbol{e}}) = W^{1/2}S\boldsymbol{d} + W^{1/2}\tilde{F}\boldsymbol{1} + W^{1/2}\tilde{\boldsymbol{e}}, \tag{4.2}$$

where $\tilde{\boldsymbol{f}}_\beta = (f_\beta(\boldsymbol{x}_1), \cdots, f_\beta(\boldsymbol{x}_n))^T$ and $\tilde{F} = (\tilde{\boldsymbol{f}}_1 \cdots \tilde{\boldsymbol{f}}_p)$. Removing the null model effect by projecting (4.2) onto the orthogonal space of $W^{1/2}S$, one gets

$$\boldsymbol{z} = F\boldsymbol{1} + \boldsymbol{e}. \tag{4.3}$$

The collinearity indices of F (Stewart 1987), which is equivalent to the cosines between the columns of F, measure the concurvity in the fit. The columns of F are supposed to predict the "response" $\boldsymbol{z}$ so a near orthogonal angle between a column of F and $\boldsymbol{z}$ indicates a noise term. Signal terms should be reasonably orthogonal to the residuals hence a large cosine between a column of F and $\boldsymbol{e}$ makes a term suspect. $\cos(\boldsymbol{z}, \boldsymbol{e})$ and $R^2 = \|\boldsymbol{z} - \boldsymbol{e}\|^2/\|\boldsymbol{z}\|^2$ are informative *ad hoc* measures for the signal to noise ratio in the data. Finally, a *very* small norm of a column of F relative to that of $\boldsymbol{z}$ also indicates a negligible term. It could be argued that the cosine diagnostics can be treated as absolute

measures provided an automatic smoothing parameter selection is adopted. These diagnostics can be used to sequentially delete redundant subspaces to build a parsimonious model in a backward fashion. See Gu (1990b) for details and examples.

5 Accuracy estimates

For any method, it would be nice to be able to say something about the accuracy of the estimate. Monte Carlo bootstrap is one method that has found use in the application of smoothing splines. In the Monte Carlo bootstrap, one generates a new set of data via a random number generator centered about the estimated model. In this setup the distribution of the y_i has to be assumed up to parameters which can be estimated, i.e., p is Gaussian, Bernoulli, etc. From this new data set one estimates a curve or surface, and by repeating the operation obtains a "cloud" of curves or surfaces which hopefully gives some idea of the accuracy of the estimate. Although the "clouds" one gets seem reasonable, it still appears that not much is known about their properties, although they have been around a while. In particular, since the estimate is generally a bit "smoother" than the truth it is possible that these clouds give a possibly too rosy picture. This method might be used with Friedman's approach here, if (assuming the ϵ_i are i.i.d. zero mean Gaussian), an estimate of σ^2 were available.

In the single smoothing parameter smoothing spline case, there are coverage bands (also called Bayesian "confidence intervals") which have the property that the expected number of "true" data points they cover is about $.95n$. See Wahba (1983) and Nychka (1988). It would be interesting to see what happens in the multiple smoothing parameter case, and also, if there is anything like a counterpart for MARS. Some authors, in the case of (nonadaptive) regression splines, have suggested the usual parametric confidence intervals for the estimated basis coefficients, as though the estimate were really in the span of the basis functions. This has yet to be justified if the true function is not in this span and hence there is bias. There is, however, no free lunch in the case of nonparametric regression, since the good methods are all biased and we have to accept a somewhat weaker definition of "confidence interval" in the nonparametric regression case if we wish to remain honest.

6 Hybrid methods

Hybrid methods which combine regression spline and smoothing spline ideas have also been considered; see, e.g., Nychka *et al.* (1984), Hutchinson and Bischof (1983), and O'Sullivan (1990). Let $\{B_l(\boldsymbol{x})\}_{l=1}^N$ be a set of N basis functions where N is generally (much) smaller than n. The estimate of η is then that η *in the span of the* $\{B_l\}$ which minimizes (2.1). If N is closer to n, then the estimate will be a good approximation to the minimizer in $\mathcal{H}$ of (2.1), and the λ and θ_i's will be controlling the smoothing. In this case, the basis functions are mainly used to ease the computation, and the location of the knots may not be very crucial. For example, if n is large, one might choose a set of basis functions with the knots a regular subset of the data points, as did Hutchinson and Bischof (1983). In their case the basis functions were formed from *representers of evaluation* at the knot points. That would amount to using $\phi_1, \cdots, \phi_M$ and a subset of the ξ's of (2.2) in the present context. See also Wahba (1990), Chapter 7. O'Sullivan (1990) used a large basis of tensor products of B-splines, with a thin plate penalty functional.

If N is smaller, then the basis functions may be helping along the smoothing (and they might be explicitly used to eliminate the possibility of too much fine structure that is known not to be there), in this case their number and locations of their knots may be more influential.

7 An omnibus GCV?

We think that all of these methods and combinations will come into use, and no one of the possibilities is going to turn out to be uniformly superior – which method is "best" is going to depend on the context. Of course it would be lovely if there was one grand criterion for comparing among the different methods, given a particular data set. It would be nice to have something like an omnibus GCV criterion, which would compare different model building procedures (for example a pure smoothing vs a pure regression procedure), but this is something remaining to be done. As Friedman takes pains to note, all of the degrees of freedom for signal must be accounted for. In order to compare across different methods this accounting should be done in comparable ways. In the case of smoothing splines, when a subspace is added, if this subspace carries an independent smoothing parameter then the minimum GCV value is non increasing. To see this, note that if the smoothing parameter for the new subspace is estimated as infinity, then we have reverted to

the model without the new subspace. It is not yet known how to "charge" for adding another free smoothing parameter in this context. The proceeding suggestions on methods for choosing subspaces as a form of model selection were in part motivated by this lack.

Given a sufficiently large data set, in practice it is of course possible to do the classical double cross validation, say, divide the set in half, fit and tune two or more models for comparison on the first half of the data, and compare them as to their predictive ability on the second half. Naturally, this brings up the question of what is a "significant" difference – which, again, could be answered if one could partition the data into several subsets. Data sets are getting bigger and computers more powerful, so statisticians are not likely to be out of business soon!

Acknowledgements

Chong Gu's research was supported by the NSERC of Canada. Grace Wahba's research was supported in part by AFOSR under Grant AFOSR 90-0103 and in part by NSF under Grant DMS-8801996.

References

Douglas, A. and Delampady, M. (1990), "Eastern Lake Survey – Phase I: Documentation for the Data Base and the Derived Data Sets," SIMS Technical Report, Dept. of Statistics, University of British Columbia, Vancouver.

Gu, C. (1989), "RKPACK and Its Applications: Fitting Smoothing Spline Models," Technical report 857, Dept. of Statistics, University of Wisconsin, Madison.

—— (1990a), "Adaptive Spline Smoothing in Non-Gaussian Regression Models," *Journal of the American Statistical Association*, to appear.

—— (1990b), "Diagnostics for Nonparametric Additive Models," Technical report 92, Dept. of Statistics, University of British Columbia, Vancouver.

Gu, C., Bates, D. M., Chen, Z., and Wahba, G. (1989), "The Computation of GCV Functions through Householder Tridiagonalization with Application to the Fitting of Interaction Spline Models," *SIAM Journal on Matrix Analysis and Applications*, 10, 457 – 480.

Gu, C. and Wahba, G. (1988), "Minimizing GCV/GML Scores with Multiple Smoothing Parameters via the Newton Method," *SIAM Journal on Scientific and Statistical Computing*, to appear.

Hutchinson, M. and Bischof, R. (1983), "A New Method for Estimating the Spatial Distribution of Mean Seasonal and Annual Rainfall Applied to the Hunter Valley, New South Wales," *Australia Meteorology Magazine*, 31, 179 – 184.

Nychka, D. (1988), "Confidence Intervals for Smoothing Splines," *Journal of the American Statistical Association*, 83, 1134 – 1143.

Nychka, D., Wahba, G., Goldfarb, S., and Pugh, T. (1984), "Cross-Validated Spline Methods for the Estimation of Three Dimensional Tumor Size Distributions from Observations on Two Dimensional Cross Sections," *Journal of the American Statistical Association*, 79, 832 – 846.

O'Sullivan, F. (1990), "An Iterative Approach to Two-Dimensional Laplacian Smoothing with Application to Image Restoration," *Journal of the American Statistical Association*, 85, 213 – 219.

Poggio, T. and Girosi, F. (1990), "Regularization Algorithms for Learning that are Equivalent to Multilayer Networks," *Science*, 247, 978–982.

Stewart, G. W. (1987), "Collinearity and Least Square Regression" (with discussions), *Statistical Science*, 2, 68 – 100.

Wahba, G. (1983), "Bayesian "Confidence Intervals" for the Cross-Validated Smoothing Spline," *Journal of the Royal Statistical Society, Ser. B*, 45, 133 – 150.

—— (1990), *Spline Models for Observational Data*, CBMS–NSF Regional Conference Series in Applied Mathematics, Vol. 59, SIAM.

Wahba, G. and Wendelberger, J. (1980), "Some New Mathematical Methods for Variational Objective Analysis Using Splines and Cross-Validation," *Monthly Weather Review*, 108, 1122-1145.

Discussion on Multivariate Adaptive Regression

Andrew R. Barron Xiangyu Xiao

University of Illinois
July 1990

1. Introduction. We describe the multivariate adaptive polynomial synthesis (MAPS) method of multivariate nonparametric regression and compare it to the multivariate adaptive regression spline (MARS) method of Friedman (1990). Both MAPS and MARS are specializations of a general multivariate regression algorithm that builds hierarchical models using a set of basis functions and stepwise selection. We compare polynomial and spline bases in this context. Our experience is that there is no substantial difference in the statistical accuracy for the data sets that we have investigated, provided that some care is taken in the choice of the model selection criterion. It is argued that the polynomial methods, with a smaller set of basis functions to select from at each step, should yield a computationally faster algorithm.

A potential difficulty of either polynomial methods with high-order terms or spline methods with closely spaced knots is the high sensitivity of the estimated response to slight changes in the inputs. We advocate the use of a roughness penalty in the performance criterion that forces smoother models to be accepted. A consequence of this roughness penalty in the polynomial case is that large coefficients in high-order terms are avoided. We believe that the MARS algorithm could also benefit from the use of the roughness penalty.

Polynomial networks are synthesized by a simple variant of the algorithm. In particular, an option of the MAPS algorithm is to allow the outputs of tentatively selected models to be considered along with the original explanatory variables as inputs for subsequent steps of the algorithm. This algorithm exhibits many of the properties of more complicated polynomial networks, and other "neural" networks for multivariate regression. For an overview of statistical learning networks see Barron and Barron (1988).

The advantage of adaptively synthesized model structure compared fixed model structure is the opportunity to seek out accurate lower-dimensional nonlinear models in the high-dimensional space of functions of several variables. MARS, MAPS, and some adaptive network techniques have the potential for selecting accurate yet parsimonious models in these high-dimensional settings.

2. Adaptive Regression. Multivariate adaptive regression is a stepwise procedure for the automatic selection of basis functions from observed data. The selected basis functions $B_m(\mathbf{x})$ yield models of the form

$$f_M(\mathbf{x},\theta) = \sum_{m=1}^{M} \theta_m B_m(\mathbf{x}),$$

for $\mathbf{x}$ in $\mathbf{R}^n$. These models are fit to observed data $(\mathbf{x}_i, y_i)_{i=1}^{N}$.

Briefly, the forward algorithm takes the following form. The initial function estimate is taken to be a constant by setting $B_1(\mathbf{x}) = 1$. Then the following steps are

repeated until a model selection criterion stops the growth of the model. From a list of candidate basis functions Γ_M, we choose one or two new basis functions $B_{M+1}(\mathbf{x})$, $B_{M+2}(\mathbf{x})$ to append to the current list of basis functions $\{B_1(\mathbf{x}), B_2(\mathbf{x}), \ldots, B_M(\mathbf{x})\}$ and then regress the data onto the span of the new list. At each step we adopt the choice of basis functions that provide the best improvement in a model selection criterion.

The key to the algorithm is the specification of a parsimonious yet flexible set Γ_M of candidate basis functions (or pairs of candidate basis functions) from which the new terms $B_{M+1}(\mathbf{x})$ and in some cases $B_{M+2}(\mathbf{x})$ are selected. Naive choices of Γ, such as the set of all polynomial terms in n variables up to a given degree, are exponentially large sets in the dimension n, and, consequently, would be computationally prohibitive in moderately high-dimensions.

Friedman's strategy is to adapt the set Γ_M of candidate basis functions to the current list of terms, by taking the set of products of terms in the current list with one-dimensional basis functions. In particular, MARS (with $q=1$) takes Γ_M to be the set of pairs of candidate terms $B_m(\mathbf{x})[\pm (x_j - t)]_+$ for $i=1,2,\ldots,M$ and $j=1,2,\ldots,n$ with the restrictions that x_j is not already in a factor of $B_i(\mathbf{x})$ and t is in a list of candidate knot locations determined by the sample quantiles of x_j.

The MAPS algorithm, in its simplest form, takes $\Gamma_M = \{ B_i(\mathbf{x})x_j : i=1,\ldots,M,\ j=1,\ldots,n \}$. Thus each step of the MAPS procedure yields a polynomial, where in one coordinate the degree is incremented by one. Experimentation has revealed that in some cases it is better to introduce pairs of basis functions $B_i(\mathbf{x})x_j$ and $B_i(\mathbf{x})x_j^2$ on each step. This has two benefits: firstly, it tends to orient the search in early steps toward models with low interaction order that are more reliably estimated, and, secondly, it avoids some of the traps of stepwise selection. [For instance, if $y = x_1^2$ and x_1 is symmetrically distributed about the origin, then the linear regression of y onto the span of $\{1, x_1\}$ is constant, and, consequently, without forced consideration of higher-order terms, the stepwise algorithm would stop with a constant as its best estimate.] With this modification, each step of the MAPS algorithm selects $B_{M+1}(\mathbf{x})$ and $B_{M+2}(\mathbf{x})$ from the set

$$\Gamma_M = \{ B_i(\mathbf{x})x_j,\ B_i(\mathbf{x})x_j^2 : i=1,\ldots,M,\ j=1,\ldots,n \}.$$

Here the second new term may be rejected from inclusion at the current step if it does not yield an improvement in the performance criterion, whereas the first new term may be rejected only after consideration of the performance with both new terms.

As in Friedman's MARS algorithm, MAPS provides an option to restrict the order of interaction of candidate terms to be not greater than a specified limit mi. Here $mi = 1$ yields an additive polynomial, $mi = 2$ allows cross terms $B(\underline{x}) = x_i^{r_i} x_j^{r_j}$ and $mi = n$ allows general polynomial terms $B(\mathbf{x}) = x_1^{r_1} x_2^{r_2} \cdots x_n^{r_n}$ to eventually be synthesized by the algorithm. Even with $mi = n$, the algorithm often stops before such high-order interactions are considered.

Following the completion of the forward stepwise algorithm, it is advisable to perform a backward stepwise pass that at each step removes the term that permits the best improvement in the performance criterion for the remaining terms. This backward pass is implemented by Friedman. With the backward pass implemented, one is

free to allow the forward pass to proceed past the point at which the performance criterion is optimized with respect to forward selection. Extraneous terms are left to be removed by the backward pass provided their removal is determined to be beneficial. At the present time, we have only implemented the forward stepwise synthesis in the MAPS program.

We hasten to point out that stepwise selection procedures can not in general be guaranteed to provide the best set of terms of a given size, see, for instance, Cover (1974). To get the best set of terms essentially exhaustive subset selection procedures would be needed. Such exhaustive procedures are feasible for certain linear regression problems, but they are not feasible for multivariate nonlinear regression, because of the exponential explosion of number of terms from which the subsets are selected. Therefore, stepwise selection is a necessary compromise in the multivariate nonlinear case.

In the related context of iterative L^2 approximation of functions, a recent result of Jones (1990) states that if the target function is in the closure of the convex hull of a given set of functions with bounded norm, then the squared norm of the error is of order $O(1/m)$ for an $m-$step forward stepwise selection. That result can be adapted to the context of adaptive polynomial regression with Γ equal to all polynomial terms up to a given degree, to yield bounds on the statistical risk. Thus forward stepwise selection can yield a reasonably accurate set of terms even if it is not, strictly speaking, the best set of terms. It is not clear whether theory analogous to that provided by the result of Jones can be developed that applies to the smaller sets of candidate terms used by MARS or MAPS.

The approximation capabilities of polynomials and splines are known in the case of complete bases (in which all polynomial terms up to a prescribed order are included). These results show, for instance, that for any given $k \geq 1$, if $f(x)$ is an k times differentiable function on $[0,1]^n$, then there exists a polynomial $f_k(\mathbf{x}) = \sum \theta_r x_1^{r_1} \cdots x_n^{r_n}$, where the sum is for all $r = (r_1,\ldots,r_n)$ with $0 \leq r_j \leq k$, for which the L^2 approximation error is bounded by

$$\int_{[0,1]^n} (f(\mathbf{x}) - f_k(\mathbf{x}))^2 d\mathbf{x} \leq \frac{1}{(2k+1)!4^k} \sum_{d=1}^{n} \int \left(\frac{\partial^k}{\partial x_d^k} f(\mathbf{x})\right)^2 d\mathbf{x}.$$

This particular bound is a specialization of the multivariate extension in Sheu (1989) of a bound due to Cox (1988). It shows the exponential convergence rate of polynomial approximation for analytic functions, assuming that the norm of the k partial derivatives is bounded by a multiple of $k!$. Splines approximations of a fixed order q, which are chosen to have roughly the same number of basis functions, k^n, are not capable of the same accuracy. The integrated squared error saturates at the slower polynomial rate $(1/k)^{2q}$, see for instance, Schumaker (1981).

Unfortunately, there is not yet an analogous theory for the approximation with subsets of the complete set of basis functions. An exception is the case of additive approximation. For instance, to approximate the additive part of a function, the best additive polynomial approximation (which uses only $nk+1$ terms instead of $(k+1)^n$ terms), achieves the same accuracy as derived above by Sheu, whereas spline approximation again saturates at the slower rate. It also may be possible to the use the theory

to characterize the error of approximation of the second-order interaction component of a function. Such a theory would hopefully show that parsimonious approximation is possible for all function with negligible higher- order interactions.

3. Complexity Penalties for Adaptive Regression Selection. At each step of the adaptive regression algorithms, terms are chosen to optimize a statistical performance criterion. The criterion depends on the average squared residuals (ASR) but incorporates a modification to penalize the number of parameters.

There is a proliferation of criteria that have been proposed for model selection. They can be roughly categorized into two groups. The first group seeks to estimate the mean squared error of prediction $MSEP_{M,N} = E(Y - f_M(X,\hat{\theta}))^2$ or related quantities of cross-validation, where X,Y denotes a sample drawn independently of the training data. The idea is that the best model is the one with the minimum $MSEP_{M,N}$. Criteria that estimate the $MSEP$ can be interpreted as adding a penalty to ASR which is roughly equal to $2(M/N)$ times an estimate of the variance of the error incurred by the best function of $\mathbf{x}$, where M is the number of parameters and N is the sample size. A representative criterion in this group is the generalized cross validation (GCV), a modification of which is used by Friedman in his MARS program. For models of the form $f_M(\mathbf{x},\hat{\theta}) = \sum_{m=1}^{M} \hat{\theta}_m B_m(x)$, the generalized cross validation (not accounting for the selection bias) takes the form

$$GCV = \frac{ASR}{(1-M/N)^2},$$

where $ASR = (1/N)\sum_{i=1}^{N}(y_i - f_M(\mathbf{x}_i,\hat{\theta}))^2$ is the average squared residual. See for instance, Eubanks (1988) for properties of the generalized cross validation and its relationship to other criteria, including Mallows' C_p, Akaike's final prediction error FPE, and Akaike's information criterion AIC. The predicted squared error PSE criterion studied in Barron (1984) is defined as $PSE = ASR + 2(M/N)\sigma^2$, where σ^2 is either the known error variance $E(Y - E(Y \mid X))^2$ or a rough estimate of it provided prior to the stepwise selection process. Under certain formulations, it is equivalent as a selection criterion to C_p and AIC. The final predication error $FPE = ASR(1+M/N)/(1-M/N)$, is the minimum variance unbiased estimator of the mean squared error of prediction, in the case of a correctly specified model and Gaussian errors. A surprising fact, shown in Barron (1984), is that for reasonable choices of σ^2, PSE is a more accurate estimate of the mean squared error of prediction. All of the criteria mentioned above suffer from the problem of considerable selection bias if the criteria is minimized over too large a set of candidate models: that is, the minimized criterion value may be significantly smaller than the value for the best model.

At first we ran our $MAPS$ program with the GCV criterion, to facilitate comparison with the $MARS$ procedure, believing that more conservative criteria would probably not be necessary in our case. However, as the results show in section 5, we encountered persistent problems of overfit. This overfit invariably occurred whenever the selection was taken over a very large sets of candidate terms, a large fraction of which are spurious. Friedman avoids the overfit problem by modifying the GCV criterion. He replaces the number of parameters M with an associated cost $C(M)$ in the expression $GCV = ASR/(1-C(M)/N)^2$, where $C(M)$ is between $3M$ and $5M$.

After encountering the overfit problems with the ordinary *GCV*, we found greater success using criteria which are specifically designed for selection and not just for the estimation of risk.

This second group of criteria, to which we turned our attention, includes criteria designed to approximate the test statistic that minimizes the overall probability of error in a Bayesian formulation of the model selection problem (such as the *BIC* of Schwarz 1978) or seeks to approximate the length of an asymptotically optimal information-theoretic code that describes the observed response values given the explanatory variables (such as the *MDL* criterion of Rissanen 1983). For either case, there is a formulation in which the dominate terms of the statistic define a criterion equivalent to the following

$$BIC = MDL = ASR + \frac{M}{N}\sigma^2 \ln(N).$$

This criterion is similar to the first group of criteria, but it incorporates a penalty which is a factor $(1/2)\ln N$ greater. For N between 50 and 400, the factor $(1/2)\ln N$ is between 2 and 3. So for conservative values of σ^2 (values believed to be not smaller than the true error variance), the *BIC*/*MDL* criteria and Friedman's modification of the *GCV* criterion should give similar results. Indeed, it appears that in practice, Friedman's modified *GCV* is closer to the *BIC*/*MDL* criterion than the original *GCV* criterion upon which the modification is based.

The *MAPS* algorithm is set up to compute any of the above criteria, *GCV*, *FPE*, *PSE*, and *MDL*, as well as the *AIC* and *BIC* criteria that obtain when the error variance σ^2 is regarded as an unknown parameter. An option selects which criterion is used for the minimization.

For theoretical properties, the work of Shibata (1981) and Li (1987) demonstrates an asymptotic optimality property satisfied by any of the criteria in the first group (with penalty equal to $2(M/N)$ times a consistent estimate of σ^2). In particular, Li (1987) gives conditions such that if $MSE_{M,N} = E(\hat{f}_M(X) - f(X))^2$ denotes the mean squared error in the estimation of $f(X) = E(Y \mid X)$ by a linear model M fit by ordinary least squares, then the mean squared error $MSE_{\hat{M},N}$ incurred by the selected model $\hat{M}$ satisfies $MSE_{\hat{M},N}/MSE_{M_N,N} \to 1$ in probability as $N \to \infty$, where $MSE_{M_N,N} = \min_M MSE_{M,N}$. The minimizations are assumed to be taken for a fixed sequence of lists of models H_N, rather than for an adaptively determined list. The theory assumes a condition that effectively limits the asymptotic number of candidate linear models that may be considered. Namely, the quantity $\sum_{M \in H_N}(NMSE_{M,N})^{-r}$ must be negligible (as $N \to \infty$), for some r for which the 2r'th moment of the distribution of the error $(Y - E(Y \mid X))$ is finite. For very large sets of candidate models this quantity is not negligible, and the theory is not applicable. Indeed, in this case, significant selection biases can occur that are characterized by a tendency to overfit. Another implication of Li's condition is the requirement that the mean squared error for the best sequence of models $MSE_{M_N,N}$ tends to zero slower than the rate $(1/N)$ that is achieved if the true function were finite dimensional. In the case that the true function $f(X)$ is in one of the finite-dimensional families, it is known that models selected by criteria in the first group have non-zero probability of asymptotically selecting an overfit model.

Asymptotic theory for model selection by the more conservative BIC or MDL criteria is given in Barron and Cover (1990) and Barron (1989,1990). This theory gives conditions such that the mean squared error of the selected model $MSE_{\hat{M},N}$ converges to zero at rate bounded by $MSE_{M_N,N} + (M_N/N) \ln N$. Convergence at this rate holds in both parametric and nonparametric cases and holds without restriction on the number of candidate models. As for the Shibata and Li theory, it is assumed that the criterion is optimized for a fixed sequence of lists of models, indexed by the sample size, rather than optimized stepwise for an adaptively determined set of models. Nevertheless, it suggests useful guidelines that might also be appropriate in the adaptive context. Chief among these is the need for care in the choice of criteria when a very large number of candidate models are considered. Somewhat larger penalties are required for accurate model selection in this case.

4. Roughness Penalty for Polynomial Smoothing. Essential to polynomial methods of regression in the presence of noise and/or model uncertainty is the use of a criterion which incorporates a roughness penalty. In particular, the MAPS algorithm chooses the parameters of each model so as to minimize

$$ASR + RP,$$

where ASR is the average squared residual

$$ASR = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(\mathbf{x}_i,\theta))^2$$

and RP is the roughness penalty

$$RP = \delta^2 \frac{1}{N}\sum_{i=1}^{N} \| \nabla_x f(x_i,\theta) \|^2.$$

Then $ASR + RP$ is used in place of the average squared residuals ASR in the model selection criteria discussed in the preceding section.

Here δ is a parameter that controls the smoothness of the model. One interpretation of the roughness penalty is that it captures the sensitivity of the average squared error to slight changes in the input variables. If the inputs are permitted to be changed from x_{ji} to $x_{ji} \pm \delta$, then $ASR + RP$ is an estimate of the average squared error that would be incurred by the function with perturbed inputs. The δ may often be set by the scientist or engineer who supplies the data as quantifying the size of changes in the input that should not be accompanied by significant changes in the response. Or it may be set by the statistician by inspection of a few runs to determine the one with which he is most satisfied. The selection of δ may also be automated by generalized cross-validation, but at considerable additional computational expense.

There is a relationship between polynomial smoothing and smoothing splines. The 2-nd order smoothing spline arises as the solution to the problem of minimization of

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - f(\mathbf{x}_i))^2 + \delta^2 \int \| \nabla f(\mathbf{x}) \|^2.$$

For smoothing splines the minimization is taken over all continuously differentiable

functions $f(\mathbf{x})$. In contrast, for polynomial smoothing, the integral is replaced, for convenience, with the sample average, and the minimization is taken over the restricted class of polynomial functions with specified bases. It is our experience that polynomial smoothing approximates the capabilities of spline smoothing, while providing advantages of speed of computation, due to the reduced size of the dimension of the linear system that is solved to obtain the polynomial approximation.

It is seen that polynomial smoothing with a roughness penalty is a generalized form of ridge regression. For a linear model

$$f(\mathbf{x},\theta) = \sum_{j=1}^{m} \theta_j B_j(\mathbf{x}),$$

the roughness penalty is a positive definite quadratic function of the parameters $\theta = (\theta_1,\ldots,\theta_m)^T$,

$$RP = \theta^T R\, \theta,$$

where R is the m by m matrix with entries

$$R_{jk} = \delta^2 \frac{1}{N} \sum_{i=1}^{N} \sum_{d=1}^{n} \frac{\partial B_j(\mathbf{x}_i)}{\partial x_d} \frac{\partial B_k(\mathbf{x}_i)}{\partial x_d}.$$

The roughness penalty prevents the estimation of models with large coefficients for terms that contribute large derivatives. With polynomial basis functions $B_j(\mathbf{x})$ each derivative $\partial B_j(\mathbf{x}_i)/\partial x_d$ is also a polynomial but with the degree reduced by one in the coordinate x_d. The larger derivatives are typically associated with the higher-order terms.

The set of equations to be solved for θ is obtained as follows. Let $V = (1/N)\mathbf{B}^T\mathbf{B}$ and $c = (1/N)\mathbf{B}^T\mathbf{y}$ be defined as in Friedman, equation (49), where $\mathbf{y} = (y_1,\ldots,y_N)^T$. Assume that the sample average of the response variable has been subtracted off so that $\bar{y} = (1/N)\sum_{i=1}^{n} y_i = 0$ and let $\hat{\sigma}_y^2 = (1/N)\,||\,\mathbf{y}\,||^2$ denote the sample variance of y. The penalized average squared residual may then be expressed as the following quadratic function of θ,

$$\begin{aligned} ASR + RP &= (1/N)\,||\,y - B\,\theta\,||^2 + \theta^T R\,\theta \\ &= \hat{\sigma}_y^2 + \theta^T V\,\theta - 2c^T\theta + \theta^T R\,\theta \\ &= \hat{\sigma}_y^2 + \theta^T \tilde{V}\,\theta - 2c^T\theta, \end{aligned}$$

where $\tilde{V} = V + R$. The parameter vector which minimizes this expression is found by solving the modified normal equations

$$\tilde{V}\,\theta = c.$$

Each new basis function introduces a new row and column of R and V, so the solution may be updated by Cholesky decomposition in the same manner as explained by Friedman.

In its simplest form, the roughness penalty treats each of the variables equally. More generally, it may be desirable to associate a parameter $\delta_{d,i}$ for each value $x_{d,i}$ of the explanatory variables. The roughness penalty then takes the form

$$RP = \frac{1}{N} \sum_{i=1}^{N} \sum_{d=1}^{n} \delta_{d,i}^2 \left(\frac{\partial f(\mathbf{x}_i,\theta)}{\partial x_d}\right)^2.$$

In like manner, the expression for the matrix R is modified to bring $\delta_{d,i}^2$ inside the double summation. The choice $\delta_{d,i} = \hat{\sigma}_{x_d}^2 \delta$ allows the penalty to be scaled to the observed spread of the explanatory variables as measured by the standard deviation $\hat{\sigma}_{x_d}^2$. Alternatively, it may be desirable to let $\delta_{d,i}$ be proportional to the magnitude of the observed values of the variables, that is

$$\delta_{d,i} = |x_{d,i}| \delta.$$

The latter choice helps to mitigate the effect of extreme observations. Also, in the case of polynomial basis functions, it allows the entries in the matrix R to be determined as a weighted sum of entries of the matrix V, thereby avoiding much of the additional computational expense otherwise associated with the use of the roughness penalty. Specifically, R may be expressed as

$$R_{jk} = \delta \sum_{d=1}^{n} r_{jd} r_{kd} V_{jk},$$

where r_{jd} denotes the exponent of variable x_d in the basis function $B_j(\mathbf{x}) = x_1^{r_{j1}} \cdots x_n^{r_{jn}}$. Note that the off-diagonal entries R_{jk} of the matrix R are zero for those pairs of basis functions B_j and B_k that share no common factors. The largest entries are typically on the diagonal and correspond to the terms with large exponents.

Some of the characteristics of the roughness penalty are incorporated in Friedman's MARS algorithm. He adds a small multiple of the diagonal entries of V to numerically stabilize the resulting modified normal equations.

5.Experimental Results. First, we took 10 replications of the simulated data from Friedman's example 4.2, each with a sample of size $N=200$. In this example, the dependence of y on $x_1,\ldots,x_{10}$ is additive, as given in Friedman's equations (56) and (57), and the inputs are drawn uniformly over the unit cube $[0,1]^{10}$. For each replication the observed response was scaled to have sample mean zero and sample variance one, but the input variables are left unscaled. The parameter for the roughness penalty was set to be $\delta = 0.01$, a moderately small value that allows for the high gradients of the response near $x_1 = 1.0$ and near $x_2 = 1/2$ in equation (56). The generalized cross validation *GCV* (without modification) was used as the selection criterion. The results of the 10 *MAPS* runs for each interaction limit $mi=1,2,10$ are summarized in Table A. Depicted are the averages and standard deviations based on the ten runs of the standardized integrated squared error (*ISE*), the standardized mean squared error of prediction (*MSEP*), the generalized cross validation (*GCV*), and the number of selected terms (*TERMS*). In accordance with the definitions in Friedman, the integrated squared error and the mean squared error of prediction are computing using knowledge of the true function and 5,000 new sample points for Monti Carlo integration.

The results in Table A may be compared with those reported in Friedman's Table 3 in the $N = 200$ case. It shows that when the unmodified *GCV* criterion is used, if the model is forced to be additive, the polynomial method is nearly as good as the spline method; however, in the case that interaction terms are considered, $mi=2,10$, we have noticeably worse integrated squared error. The large average numbers of

terms 17.3 and 20.7 reveal that the polynomial models are overfit using the unmodified *GCV* criterion. Indeed, the first 11 or 12 terms were almost exclusively additive terms in the meaningful variables (x_1 through x_5), but the additional terms chosen in the $mi=2$ and $mi=10$ case were almost exclusively spurious cross product terms and terms involving the nuisance variables x_6 through x_{10}. This large number of spurious models contribute to the large selection bias that results in overfit with the unmodified *GCV* criterion.

We then repeated the experiment using the more conservative *BIC/MDL* criterion. In the definition of the *BIC/MDL* we used the known variance of the noise $\sigma^2 = 1$. The results are summarized in Table B. The results in Table B for adaptive polynomial modeling compare quite favorably with those for adaptive splines in Friedman's Table 3. Indeed, the averages and standard deviations of the integrated squared error and the mean squared error of prediction are either equally good or slightly better in every case. The most noticeable improvement is in the case that arbitrary interactions are allowed ($mi=10$). With the *BIC/MDL* criterion almost all of the spurious interaction terms are rejected.

Next, we drew one sample of size 100 in accordance with Friedman's example 4.3. Again there are ten variables. The first two variables contribute to the response through a term which is a sinusoidal function of the product x_1x_2. The remaining contributions are additive, specifically, a quadratic term in x_3 and linear terms in x_4 and x_5. (Admittedly, the polynomial terms in the true response give polynomial modeling an unfair advantage for this example.) The other variables x_6 through x_{10} are not used by the true response. We set $\delta = 0.01$, $\sigma = 1.0$, $mi = 2$, and standardized the observed response. With the *BIC/MDL* criterion the terms and coefficients for the normalized model are given in Table C in the order in which they are selected. (The model is expressed in standardized form; it is unitized by multiplying by $\hat{\sigma}_y = 5.8189$ and then adding $\bar{y} = 15.0549$.) The criteria values suggest that the MAPS model and the MARS model are roughly equally accurate for this sample from example 4.3.

Table C shows that apparently meaningful terms were selected by the MAPS algorithm with the *BIC/MDL* criterion, with the exception of the third term, which is a quadratic in x_4 while the true response is linear in x_4. It is anticipated that this term would be removed by a backward stepwise selection. When the unmodified *GCV* is used as the criterion, six more terms are selected, all of which involve interactions with the extraneous variables, x_6, x_7, x_9, and x_{10}. Evidently, the extraneous variables introduce two many diverse candidate terms for the *GCV* to provide a uniformly accurate criterion.

Finally, we considered the Portuguese olive oil data, a copy of which was obtained from Friedman. We standardized the response variable to have sample mean zero and sample variance one. Inspection of the data set shows that the variables are rounded to the nearest one-tenth, so we set $\delta = 0.05$, accordingly. Using a ten-fold cross validation as explained in Friedman, we ran *MAPS* ten times each with 41 or 42 observations removed, using the unmodified *GCV* criterion, and a maximum term limit of 30. The resulting standardized average squared residual (*CV*) and relative frequence of misclassifications (error *CV*) on the cross-validated data is given in

Table D. Also show is the unmodified *GCV* and number of variables obtained from the *MAPS* procedure with all observations included. With the *GCV* criterion the *MAPS* procedure hit the maximum term limit of 30. (Subsequent runs with an increased term limit stopped at 42 terms, fueling suspicion of overfit). Despite the excessive number of terms selected with the unmodified *GCV* criterion, the cross-validation results in Table D for MAPS are as good as obtained by Friedman in Table 9 with MARS and the least squares criterion. Nevertheless, improved cross validation results may be possible with MAPS using a more conservative performance criterion.

In Table E, we show the selected model, when MAPS is run with the *BIC/MDL* criterion and $\sigma^2 = 0.03$ (which corresponds to a standardized variance of 0.18). Here we standardized both the observed explanatory variables and the response variables to have sample mean zero and sample variance one. In this case a more parsimonious model (16 terms) is selected. The values of the *BIC/MDL* and *GCV* criteria for this selected model appear to be reasonable, but we have not yet completed the ten-fold cross validation of models selected by *BIC/MDL* to provide additional confirmation of the apparent accuracy.

6. An adaptive network example. In this last section, we illustrate a simple feed-forward polynomial network. This is created using a option of the *MAPS* algorithm. With the feedforward network option, each step of the algorithm augments the X matrix with the current model output for consideration at subsequent steps. The set of candidate new terms, when submodel outputs are fed forward, is determined as before. An exception is that terms which are linear in a previous model output are not permitted, since such a term would introduce a linear dependence with other terms still on the term list. Consequently, we require previous model outputs to be initialized in the list as a nonlinear product with itself or with other variables.

Table F depicts the results using the Portuguese olive oil data with the network option and the *BIC/MDL* criterion. Here the outputs from model 3 and model 6 were selected by the criterion to be input to subsequent models. The bases functions for models 3 and 6, respectively, are the same as the first 3 and first 6 terms in Table F. However, the coefficients for these submodels are somewhat different than for the final linear combination in model 11.

The effective number of parameters of the network for use in the performance criteria is defined as the number of coefficients directly computed in the present model plus α times the number of parameters in submodels that feed into the present model. For the results in Table F we are using $\alpha = 0.3$. When a step of the algorithm selects a pair of terms, only the output with both included is passed as an input to subsequent models; this explains why there is no $M2$, m_5, m_7, or m_9 in Table F.

The results with the network option for the olive oil data show a slight improvement in the values of the criteria. It is interesting that in Table F the algorithm proved eager to include successive powers of an intermediate output, rather than successive powers of a single variable as in Table E. Cross validation values have not yet been computed for the models in Tables E and F. We suspect that the CV will be slightly better but not substantially better than reported in Table D.

The network option also was tried on the data from examples 4.2 and 4.3. In both of these cases, no factors were selected from the list of previous models. So in these cases, the result of the network synthesis algorithm reverts to the results obtained with the conventional adaptive selection algorithm. This is not surprising since the synthetic examples 4.2 and 4.3 are defined directly as a sum of terms rather than indirectly through a composition. Experience has shown, as reported in Barron et.al. (1984, 1988), that network methods frequently provide useful answers for large-dimensional data from real engineering and scientific problems for which conventional linear techniques have not been as successful.

7. Conclusions. Adaptive synthesis of nonlinear models is essential in those empirical modeling contexts where scientific or engineering considerations do not provide a complete parametric solution, and where the high-dimensionality of the space of candidate inputs prohibits the use of other nonparametric smoothing techniques.

The techniques that gain widest acceptance among empirical modelers are those that are statistically accurate, computationally reasonable, flexible to use in diverse contexts, and (sometimes most importantly) well understood by the scientist or engineer and his clients. The paper by Jerry Friedman goes a long way toward making a powerful technique clearly understood. Also, the thoroughness of his methodological and experimental studies of a statistical modeling technique provide an excellent prototype for what, hopefully, will be many more such papers in the field.

The preliminary comparison provided here of adaptive regression splines with adaptive polynomial smoothing on several data sets suggests that the spline method does not provide any substantial gain in accuracy over the polynomial method. This should provide some pause for the empirical modeler who is debating whether to switch from the more customary polynomial models to the sometimes less familiar splines.

In addition to the known approximation capabilities, polynomial models have in their favor the relative ease of interpretation in many scientific and engineering contexts. Against polynomial models has been the fact that least squares polynomials are prone to wild extrapolative behavior in the high-order case. Here we have pointed out that the simple device of a roughness penalty, familiar to spline smoothers, can be used for polynomial smoothing to mitigate this wild behavior. We recommend properly smoothed and adaptively synthesized polynomial modeling as a serious competitor to adaptively synthesized splines.

References

Barron, A. R. (1984). The predicted squared error: a criterion for automatic model selection. *Self-Organizing Methods in Modeling*, S. J. Farlow, editor, Marcel Dekker, New York, 87-103.

Barron, A. R. (1989). Statistical properties of artificial neural networks. *Proceedings of the 28th Conference on Decision and Control.* IEEE, New York.

Barron, A. R. (1990). Complexity regularization. *Proceedings of the NATO Advanced Study Institute on Nonparametric Functional Estimation and Related Topics.*. Kluwer

Academic Publ., Hingham, MA.

Barron, A. R. and Barron, R. L. (1988). Statistical learning networks: a unifying view. *Computing Science and Statistics: Proceedings of the 20th Symposium on the Interface.* E. Wegman, et. al., editors, American Statistical Association, Alexandria, VA.

Barron, A. R. and Cover, T. M. (1990). Minimum complexity density estimation. *IEEE Transactions on Information Theory.* To appear.

Barron, R. L., Mucciardi, A. N., Cook, F. J., Craig, J. N., and Barron, A. R. (1984). Adaptive learning networks: development and application in the United States of Algorithms related to GMDH. *Self-Organizing Methods in Modeling*, S. J. Farlow, editor, Marcel Dekker, New York, 25-65.

Cover, T. M. (1974). The best two independent measurements are not the two best. *IEEE Transactions on Systems, Man and Cybernetics.* SMC-**4** 116-117.

Cox, D. D. (1988). Approximation of least squares regression on nested subspaces. *Annals of Statistics.* **18** 713-732.

Eubank, R. L. (1988). *Spline Smoothing and Nonparametric Regression.* Marcel Dekker, New York.

Friedman, J. H. (1990). Multivariate adaptive regression splines. *Annals of Statistics.* Paper presently under discussion.

Jones, L. (1990). A simple lemma on iterative sequences in Hilbert space and convergence rates for projection pursuit regression. Technical Report, No. 16, Department of Mathematics, University of Lowell, Lowell, MA.

Li, K-C. (1987). Asymptotic optimality for C_p, C_L, cross-validation, and generalized cross-validation: discrete index set. *Annals of Statistics* **15** 958-975.

Rissanen, Y. (1983). A universal prior for integers and estimation by minimum description length. *Annals of Statistics* **11** 416-431.

Schumaker, L. L. (1981). *Spline Functions: Basic Theory.* Wiley, New York.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics* **6** 461-464.

Sheu, C-H. (1989). Ph.D. Thesis. Department of Statistics, University of Illinois, Champaign, IL.

Shibata, R. (1981). An optimal selection of regression variables. *Biometrika* **68** 45-54.

Table A

Summary of the results of MAPS modeling with the unmodified GCV criterion on Friedman's additive data (Section 4.2).

mi	$\overline{ISE}$	$\overline{MSEP}$	$\overline{GCV}$	$\overline{TERMS}$
1	.030(.010)	.16(.01)	.15(.01)	12.0(1.4)
2	.053(.018)	.18(.02)	.14(.01)	17.3(2.9)
10	.086(.037)	.21(.31)	.13(.01)	20.7(5.2)

Table B

Summary of the results of MAPS modeling with the BIC/MDL criterion on Friedman's additive data (Section 4.2).

mi	$\overline{ISE}$	$\overline{MSEP}$	$\overline{BIC,MDL}$	$\overline{TERMS}$
1	.025(.006)	.155(.006)	.16(.01)	11.1(1.3)
2	.030(.007)	.159(.007)	.16(.01)	11.5(1.6)
10	.031(.007)	.160(.007)	.16(.01)	11.3(1.6)

Table C

Coefficients and exponents of the selected polynomial terms for the model in Example 4.3.

Term	Coefficient	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
1	-1.6809	0	0	0	0	0	0	0	0	0	0
2	2.0970	0	0	0	1	0	0	0	0	0	0
3	-0.3138	0	0	0	2	0	0	0	0	0	0
4	0.2363	0	1	0	0	0	0	0	0	0	0
5	-0.2508	0	2	0	0	0	0	0	0	0	0
6	-0.3203	1	0	0	0	0	0	0	0	0	0
7	0.3347	2	0	0	0	0	0	0	0	0	0
8	-4.0523	0	0	1	0	0	0	0	0	0	0
9	3.9369	0	0	2	0	0	0	0	0	0	0
10	0.7853	0	0	0	0	1	0	0	0	0	0
11	-0.0794	2	1	0	0	0	0	0	0	0	0
12	-6.9382	2	2	0	0	0	0	0	0	0	0
13	6.9397	1	1	0	0	0	0	0	0	0	0

GCV	BIC,MDL
0.041	0.049

Table D

Portuguese Olive Oil

method	#variables	GCV	CV	error CV
MAPS(mi=2)	7	.15	.22	.036

Table E

Coefficients and exponents of the selected polynomial terms for the Portuguese olive oil data.

Term	Coefficient	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13
1	-0.5731	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0.2923	0	1	0	0	0	0	0	0	0	0	0	0	0
3	0.2829	0	2	0	0	0	0	0	0	0	0	0	0	0
4	0.3247	0	2	0	0	0	0	0	0	0	0	0	1	0
5	-0.0088	0	0	0	0	0	1	0	0	0	0	0	0	0
6	0.0588	0	0	0	0	0	2	0	0	0	0	0	0	0
7	-0.2483	0	1	0	0	0	0	0	0	0	0	0	1	0
8	-0.0794	0	2	0	0	0	0	0	0	0	0	0	2	0
9	-0.0417	0	2	0	0	0	0	0	0	0	0	0	3	0
10	-0.0417	0	2	0	0	0	0	0	0	0	0	0	0	1
11	-0.7796	0	3	0	0	0	0	0	0	0	0	0	0	0
12	0.0527	0	4	0	0	0	0	0	0	0	0	0	0	0
13	0.1896	0	5	0	0	0	0	0	0	0	0	0	0	0
14	-0.0520	0	2	0	0	0	0	0	0	1	0	0	0	0
15	-0.0264	0	0	0	0	0	2	0	0	0	0	0	1	0
16	0.0440	0	4	0	0	0	0	0	0	0	0	0	0	1

GCV	BIC,MDL
0.159	0.188

Table F

Coefficients and exponents of the selected polynomial terms with the feedforward network option, for the Portuguese Olive Oil data.

Term	Coefficient	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	m3	m4	m6	m8	m10
1	-0.71278	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	-0.25350	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0.07698	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0.18919	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
5	-0.00770	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
6	0.02824	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
7	1.89370	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0
8	0.49567	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0
9	-1.26261	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0
10	0.32184	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0
11	-0.05317	0	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Effective #Parameters	GCV	BIC,MDL
14	0.14	0.174

Rejoinder

J. H. Friedman

Department of Statistics
and
Stanford Linear Accelerator Center
Stanford University

I thank the Editors for inviting such a distinguished group of researchers to discuss this paper, and the discussants for their valuable contributions. The discussants are among the leaders in the field of function approximation and estimation; it is therefore no surprise that their comments are so perceptive and stimulating. Many important suggestions are made for improving the MARS procedure. These discussions provide a clearer and deeper understanding of both the strengths and limitations of the MARS approach. Each of them raises many very important issues, some of which I respond to below. Space limitations preclude a more thorough discussion of all of the cogent points and innovative ideas presented.

Shumaker

I thank Professor Shumaker for providing the additional references, especially the recent ones that were not available in 1987 when I performed the main body of this work. Of the five that relate to multivariate adaptive approximation, only one (deBoor and Rice, 1979) presents a procedure that could possibly lead to a practical method in high dimensions. Their procedure is essentially recursive partitioning (15) using polynomials for the parametric functions $g_m(\mathbf{x} \mid \{a_j\}_1^p)$ in each subregion. The limitations of recursive partitioning methods are discussed in Section 2.4.2. It is perhaps interesting to note than in none of these papers was the method proposed therein ever actually applied to any (real or synthesized) multivariate problems.

In the asymptotic limit with no noise it may be true that only the continuity properties of the function to be estimated and the degree of the polynomials in each subregion limit the accuracy of a recursive partitioning scheme. As discussed in Section 2.2, all practical sample sizes are far from being asymptotic in high dimensional problems. In such cases there is a trade-off between the number of subregions and the maximum degree of the approximating polynomial in each one. A q-degree polynomial in n dimension is characterized by more than $\binom{q+n-1}{n-1}$ parameters. Thus, even globally, there is a severe restriction on the degree of a polynomial fit. One can use the available degrees-of-freedom to fit moderate degree polynomials in a few regions or very low degree ones in many regions. Which is the best strategy in a particular application will depend upon the true underlying function $f(\mathbf{x})$ (1), but a great body of experience indicates that the latter approach gives the best results in general. In fact, the most popular choice with recursive partitioning methods is zero degree (AID, Morgan and Sonquist, 1963, and CART, Breiman, et al., 1984). With this rather strong limitation on the allowable (and desirable) polynomial degree in each subregion, the lack of continuity of approximations derived from recursive partitioning methods does have an adverse effect on their accuracy, especially when compared to similar methods that produce continuous approximations. This is provided, of course, that the function to be approximated is relatively smooth.

The discussion concerning end effects in Section 3.7 centers on the bias-variance trade-off of estimates near the edges. In the mathematical approximation literature the value of the function is generally assumed to be measured without error [$\epsilon = 0$ (1)]; that is, all relevant variables that contribute to the variation of the response are available to help construct the approximation. In this case there is no variance and only bias considerations are relevant. When there is error, the variance of the function estimate near the edges is a dominant concern. Having the estimate smoothly match a linear function near the edges removes only the first order bias there, but has a strong moderating influence on the variance. In fact, even this precaution

is sometimes inadequate and additional measures are required (see Section 3.8). As discovered by both Breiman (1989a) and Stone and Koo (1985) restricting the dependence of the approximation near the edges to be (at most) linear is an essential ingredient for controlling variance.

I agree that if one does not need C^1 functions is is often better to use the approximation based on linear splines (see second paragraph of Section 3.7). The continuous derivative approximation often does better, but the effect is seldom dramatic, whereas the converse is not true. (See the example in Section 4.7 and the comments of Buja, Duffy, Hastie, and Tibshirani.) When the piecewise-linear approximation is doing dramatically better this is evident from the estimated lack-of-fit (relative GCV scores). Again the converse is not necessarily true. Since the model optimization is based on the linear spline basis, the piecewise-cubic fit to the data is generally slightly (to moderately) worse than the corresponding piecewise-linear fit for comparable (future prediction) accuracy. I generally use the piecewise-cubic model if its optimized GCV score is no more than 10 to 20 percent worse than that for the linear spline fit. Otherwise, I prefer the piecewise-linear model. Of course one could use cross-validation to more directly judge the relative quality of the two models.

I also agree that the approximation properties of the modified (piecewise-cubic) basis functions are not yet well understood. Whether they approximate as well as linear, quadratic or other splines will depend on the particular application. It seems reasonably clear that they will perform better than quadratic or higher order splines in noisy situations (with a smooth underlying function) owing to the end effects discussed above (Breiman, 1989a, and Stone and Koo, 1985).

The reason for using the truncated power spline basis is given in Section 3.9; there is a one-to-one correspondence between knots and basis functions. This was the central ingredient in the (univariate) adaptive regression spline strategy originally proposed by Smith (1982). Lyche and Mørken (1988) propose essentially the same idea but use a B-spline implementation. As described in their paper, severe computational complexities are involved with the B-spline approach, as compared to the simple and elegant method proposed by Smith (1982). In the multivariate case the analog of these increased computational complexities are likely to lead to insurmountable problems.

Owen

I am grateful to Professor Owen for contributing his valuable piece of research to the discussion of this paper. He has set down a theoretical framework that explains many of the results that I obtained empirically through Monte Carlo studies. More importantly his work provides the understanding and insight necessary to suggest improvements to the simple model selection approach that I implemented.

The results contained in his last paragraph concerning the relative insensitivity of the fit to the precise locations of the placed knots are surprising and encouraging. I observed this phenomenon during empirical studies on varying the minimum span (43). The quality of the fit appeared to be remarkably insensitive to the value used provided it was not too small (in high noise situations). Also, I implemented an algorithm for post knot optimization ("model polish") given the topology of the model (23) derived form the forward/backward stepwise strategy. This algorithm minimized the lack-of-fit criterion simultaneously with respect to all the knot parameters $\{t_{km}\}$ in (23). This can be done surprisingly rapidly using a minor variation of an algorithm originally proposed by Breiman (private communication) in the context of recursive partitioning. Given the suboptimal way in which the knots are placed with the stepwise strategy, the expectation was that this model polish would provide distinctly superior fits. This was not the case. In all situations in which I tried model polish, the resulting fits were changed very little from that provided originally by the stepwise algorithm. It should be noted however that the examples that I used in both studies (minimum span variation and model polish) involved very smooth functions with no really sharp local structure. For functions with such (sharp) structure, model polish may provide substantial improvement. Routinely applying model polish would then

provide insurance against this possibility.

Owen's suggestion is to deliberately use a very coarse grid during the initial stepwise knot placement and then to rely on model polish to produce the final refined model. This will not improve speed for least-squares fitting since the updating formulae (52), in conjunction with suggestions in Stone's discussion, allow all possible (data point) knot locations to be evaluated with nearly the same computation as any subset of them. As Owen points out however, for lack-of-fit criteria that to not admit fast updating, this strategy will produce dramatic computational savings, perhaps enough to make their routine application feasible.

I agree with Owen that MARS may have potential as a method for modeling noiseless data such as that produced by computer simulation experiments. Although it may produce adequate results in its present formulation it is quite likely that its performance can be improved for this purpose by changing aspects of the implementation to take advantage of the knowledge that there is no noise (as noted by Owen). For example, the strong concern about end effect variance, that motivated several important design decisions, could be relaxed and different strategies may be more appropriate in this case. Also, different model selection strategies are likely to help improve performance in these settings.

Stone

Stone's suggestions in the second and third paragraphs of his discussion turn out to be quite useful. The principal idea in the "t-to-enter" algorithm is to orthogonalize both the response and the yet to be entered (predictor) variables with respect to each predictor as it is entered into the regression equation. In the context of MARS the corresponding predictors are the basis functions derived during the forward stepwise procedure (algorithm 2). Of course, during the execution of the forward stepwise algorithm all of the yet to be entered basis functions are not known. However, one can orthogonalize each basis function as it is entered with respect to all of those already selected, thereby keeping all currently entered basis functions orthonormal to each other. Let $B_M(\mathbf{x})$ be one of two (centered) next basis functions selected as a result of executing the outer loop of algorithm 2. Then $\tilde{B}_M(\mathbf{x})$ and c_M are saved where

$$\tilde{B}_M(\mathbf{x}) = \left[B_M(\mathbf{x}) - \sum_{m=1}^{M-1} b_{Mm}\tilde{B}_m(\mathbf{x})\right] / b_{MM}^{1/2},$$

with

$$b_{Mm} = \sum_{i=1}^{N} B_M(\mathbf{x}_i)\tilde{B}_m(\mathbf{x}_i), \quad 1 \leq m \leq M-1 \tag{75}$$

$$b_{MM} = \sum_{i=1}^{N}[B_M(\mathbf{x}_i) - \sum_{m=1}^{M-1} b_{Mm}\tilde{B}_m(\mathbf{x}_i)]^2$$

and

$$c_M = \sum_{i=1}^{N} y_i \tilde{B}_M(\mathbf{x}_i).$$

The denominator b_{MM} in (75) is zero if and only if $B_M(\mathbf{x})$ has an exact linear dependence on the previously entered basis functions $\{B_m(\mathbf{x})\}_1^{M-1}$. In this case $\tilde{B}_M(\mathbf{x})$ and c_M need not be saved. $B_M(\mathbf{x})$ must still be retained however to serve as a candidate parent for future basis functions.

Consider now the search for the next basis function $B_{M+1}(\mathbf{x})$. The (least-squares) lack-of-fit criterion to be evaluated in the innermost loop of algorithm 2 (line 7) at each potential knot location t is proportional to $-I(t)$ where

$$I(t) = \frac{\left[c_{M+1}(t) - \sum_{i=1}^{M} c_i V_{i,M+1}(t)\right]^2}{V_{M+1,M+1}(t) - \sum_{i=1}^{M} V_{i,M+1}^2(t)} \tag{76}$$

with $\{c_i\}_1^M$ given in (75) and the other quantities given by the updating formulae (52) with the orthonormal basis functions $\{\tilde{B}_i\}_1^M$ replacing the $\{B_i - \bar{B}_i\}_1^M$ appearing there. The quantity $I(t)$ (76) is the improvement in the residual sum of squares resulting from adding the corresponding basis function with knot location t. This must be computed at every eligible knot location.

A computational advantage results from the fact that $I(t)$ (76) can be computed rapidly in $O(M)$ time. The strategy described in Section 3.9 required a partial Cholesky decomposition $[O(M^2)]$ and a full back-substitution [also $O(M^2)$] at each eligible knot location. Since the computation associated with the updating (52) is $O(M)$ (as it was before) the total computation for the new approach is

$$C \sim nNM_{\max}^3$$

in general, and $C \sim nNM_{\max}^2$ for additive modeling ($mi = 1$). Here $M_{\max}$ is the maximum number of basis functions. Basically the second term in the sum of (53) has been eliminated by this approach reducing the computation by roughly a factor of $M_{\max}$ for (very) large values. This will enable the MARS procedure to be applied to much larger and more complex problems than with the initial implementation described in Section 3.9.

I have implemented this new approach into the latest version of the MARS software. In order to get an idea of the computational savings involved, I reran the example of Table 1 (this time on a DECstation 3100) contrasting the running times of the new versus the old implementation. Table 14 shows the ratio of running times (new/old) in the same format as Table 1.

Table 14

Ratio of running times (sec.) for the new versus old (new/old) implementations of the least-squares updating algorithm on the example of Table 1. Computations were performed on a DECstation 3100.

$M_{\max}$	5	10	20	40	50
mi					
1	$\frac{.3}{.4}$	$\frac{.5}{.8}$	$\frac{1.0}{2.3}$	$\frac{2.4}{8.4}$	$\frac{3.1}{14.1}$
2	$\frac{.4}{.6}$	$\frac{1.4}{2.1}$	$\frac{3.7}{7.8}$	$\frac{16.0}{55.2}$	$\frac{22.9}{107.3}$
4	$\frac{.5}{.7}$	$\frac{1.4}{2.3}$	$\frac{5.3}{11.4}$	$\frac{16.4}{114.7}$	$\frac{24.8}{228.2}$

The last two columns especially of Table 1, representing the higher $M_{\max}$ values, indicate that the computational savings associated with the new approach are nontrivial. For example, with $mi = 4$ and $M_{\max} = 40$ or 50, one can now do a ten-fold cross-validation to access the quality-of-fit in the same time that the old program could do just a single fit. Buja and Duffy (see discussion of Buja, Duffy, Hastie and Tibshirani) consider very large $M_{\max}$ values, for which the computational savings ought to be even more dramatic.

Stone's suggestions for extending MARS to logistic regression parallel those made by Buja, Duffy, Hastie, and Tibshirani. I think this idea has substantial potential and that it should lead to a better method than that described in Section 4.5. His ideas for density and conditional density (MARES) estimation are quite clever and intriguing, and hold practical promise. The computational burden, especially in the multivariate case, is likely to be heavy unless special tricks can be found. Perhaps the ideas mentioned by Owen in the last paragraph of his discussion might be helpful here.

Lewis and Stevens (1990) have been studying the application of MARS to the autoregressive modeling of time series. They report considerable success.

O'Sullivan

Professor O'Sullivan presents some valuable ideas for enhancing the power and interpretability of MARS-like approaches. MARS is indeed coordinate sensitive as are most of the commonly used regression techniques

(linear regression with variable subset selection, CART, additive modeling). The very notions of main effects and interactions are coordinate sensitive. Coordinate sensitive procedures will outperform their affine equivariant counterparts in situations for which the dependence of the true underlying function is simplest in the coordinate system chosen (usually the original measured predictor variables). Here simplicity is defined in terms of the ease with which a given procedure can approximate the function. Introducing adjustable linear combinations in place of the original coordinates removes the coordinate sensitivity and allows the approximation procedure to search for linear combinations that provide the simplest representation. This potential reduction in bias comes at the expense of increased variance (optimizing with respect to the linear combination coefficients) and usually substantially increased computational complexity. Interpretation of the fitted models is most easily achieved in terms of the original measured variables so that special interpretation/visualization tools are necessary for affine equivariant procedures. O'Sullivan presents some nice ideas along these lines in his discussion.

Experience with the linear combination split option in CART (Breiman et al., 1984) has yielded somewhat surprising results. Employing linear combination splitting seems to only rarely give substantially improved performance over axis oriented splitting and surprisingly often it does worse. This may be a reflection of the types of problems to which CART has been applied. It also might be a reflection of the local variable subset selection property of axis oriented recursive partitioning (see Section 2.4.2) which, of course, is a coordinate sensitive concept.

As discussed in Section 3.3, recursive partitioning procedures produce basis function sets that involve high order interactions. This explains their poor performance in situations where main effects and low order interactions dominate. On the other hand, in converse situations where the true underlying function happens to dominately involve high order interactions this aspect of recursive partitioning becomes an asset. The alternating current examples (Sections 4.4–4.4.2) were included because they involve strong interaction effects to all orders. (Such examples involving real situations are not easy to find.) The MARS results indicate that it does a credible, if less than perfect, job in this case. CART with its bias toward high order interactions ought to do better here except that lack of smoothness limits its accuracy. A smoothed version of CART, such as O'Sullivan's SCART, mitigates this limitation and, as reported in his discussion, it does somewhat better than MARS. The two examples where he reports MARS does substantially better are ones that involve dominately main effects and/or only low order interactions.

O'Sullivan suggests a clever approach based on finite element techniques (SCART) for smoothing CART models. Another way to implement a smoothed version of CART would be to directly follow the paradigm outlined in Section 3.2. That is, replace the CART step functions by corresponding higher order (univariate) spline functions, without employing the strategy described in Section 3.3, i.e. the parent basis function would be removed as in CART. Repeated factors involving the same variable would be allowed in the same basis function product with this strategy. Also, the backward stepwise procedure would follow that used in CART. A possible advantage of this approach over one based on finite elements is that no additional global smoothing (really smearing) parameter s (or p) need be introduced and adjusted to optimize the fit. All (local) smoothing is controlled directly by the forward/backward knot placement algorithm as in CART.

Breiman

I share Professor Breiman's wonderment at this article appearing in the *Annals of Statistics*. I was surprised when the journal solicited this paper and astonished when I discovered that they were serious. I hope that as a result traditional readers of the *Annals* will not cancel their subscriptions, but instead swallow hard and wait for the next issue where things ought to be back to normal.

Breiman's remarks center on the issue of model selection and specifically on the use of a penalized least-squares criterion for this purpose. I agree that the (historical) name "generalized cross-validation" for the

GCV criterion (30) is misleading and that, especially with nonlinear fitting, it bears little resemblance to ordinary leave-one-out cross-validation. This fact, in and of itself, does not indicate its superiority or lack thereof. Ordinary cross-validation has its detractors and the issue of model selection is still being hotly debated in what has become a vast literature on the subject. I wholeheartedly agree with Breiman that the use of model selection criteria derived for linear fitting in nonlinear contexts (such as the use of C_p or F-testing with variable subset selection procedures) represents a long standing abuse in our field. In fact, Breiman has been one of the (few) leaders in pointing this out and suggesting remedies [Breiman (1989c) and Breiman and Spector (1989)]. The modification to the (linear) GCV criterion proposed in Section 3.6 is an (admittedly crude) attempt to account for the nonlinear aspects of MARS fitting. As noted by Breiman the motivation for this approach was largely computational. The extent of its statistical success seems remarkable given the crudeness of the approximation. There is nothing intrinsic in the MARS approach to the use of any particular model selection criterion. If a better criterion can be found, this will improve the performance of MARS, and so the results obtained with it so far represent lower bounds on what may be possible with this approach in the future. With the increased speed of the MARS algorithm obtained as a result of Stone's suggestion (Table 14) model selection through ordinary cross-validation is certainly computationally feasible except for large problems. (For these an independent test set can be used.) I intend to implement this approach and compare its performance to that of the current implementation. To the extent that Breiman's speculations are correct this should lead to improved statistical performance for MARS.

I also found the "packing problem" to be an important concern. This was the motivation for the introduction of a "minimum span" described here in Section 3.8 and in Friedman and Silverman (1989), Section 2.3. Requiring a minimum number of observations between successive knot locations limits the number of candidate models in precisely the same way as Breiman's strategy of placing a limited number K of initial knots in his backwards stepwise method. They both limit the number of eligible knot locations and prevent nearby models from becoming too closely packed in the space of candidate models. The analogy in MARS to Breiman's strategy of choosing K through model selection, would be to adjust the parameter L that controls the minimum number of observations between knots, to optimize a model selection criterion such as cross-validation. The generic default value for L given by (43) is conservative in the sense that it is set as small as possible consistent with some resistance to runs in the noise. The motivation is to keep the procedure as sensitive as possible to potential sharp structure in the function. When MARS is used in an initial exploratory mode, this seems like a reasonable choice. This of course has the collateral effect of increasing the variance of the estimates (possibly reacting to noise masquerading as sharp structure). When the true underlying function is very smooth with no sharp structure anywhere (such as $f(x) = .667\sin(1.3x_1) - .465x_2^2$) then there is no bias increase in increasing L, but there is a corresponding variance decrease. Figures 1 and 2 of Breiman's discussion compare MARS using its generic default value (no adjustment) to a procedure for which the corresponding parameter has been adjusted to do best on this particular example (through cross-validation). Such adjustment is a good idea, and I recommend it with MARS modeling, especially in the latter stages of obtaining a final model estimate.

Restricting the minimum number of observations between knots (or the number of initial knots K in a backward strategy like Breiman's) is an indirect way of trying to limit the global (absolute) second derivative of the final estimate. In his discussion of TURBO (Friedman and Silverman, 1989) Hastie (1989) suggested that the sometimes "wild" behavior of adaptive spline estimates (seen in low signal to noise, small sample situations) could be mitigated by directly placing a mild bound (or penalty) on the average squared second derivative. (As can be seen in Figures 1 and 2 of Breiman's discussion, the "wild" estimates tend to have much higher absolute second derivatives.) Such a bound or penalty is global in nature, so that if one wanted to retain the flexibility of adaptive regression splines to adapt the degree of smoothing locally, the penalty should be made just large enough to inhibit possible wild behavior. In their discussion of this paper Buja,

Duffy, Hastie, and Tibshirani show how to extend this idea to the more general context of MARS, and, in addition, suggest the possibility using this approach for general model selection in place of backward basis function deletion. Both ideas are straightforward to implement in MARS and are currently under investigation (jointly with Hastie). We expect this approach to produce the biggest improvements in small sample, low signal to noise, situations where the underlying function is very smooth.

Breiman points to some of the simulation study results as indicating a general failure of the (modified) GCV criterion (30) when used in the context of MARS. As noted above the MARS procedure in no way requires the use of this criterion and one that can be shown to work better would be enthusiastically welcomed. However, I feel an obligation to those who support GCV to respond to some of Breiman's concerns. Cross-validation produces an estimate of average predicted squared error. It can be shown to be an unbiased estimate but it surely has variance [sometimes rather high – see Efron (1983)]. For this reason it will not select the best model every time. I view the pure noise results in Tables 2a and 2b to be highly encouraging. While the GCV score estimated the MARS model as doing (slightly) better than the response mean about half the time, the actual distribution (percent points) show that it hardly ever claims that the MARS model is very much better, making it unlikely that one would be led to embarrassing conclusions. Whether ordinary cross-validation would do better, given the high variance of its estimates, is by no means certain. In any case, the results in Tables 2a and 2b are really more a test of the smoothing parameter choice $d = 3$ (32), rather than the criterion (30), since one can make the procedure arbitrarily conservative by increasing the value of d. The fact that on the data of Section 4.3 (Table 5b) full MARS modeling does as well as when interactions are restricted to be bivariate ($mi = 2$), represents a (fairly dramatic) success for the model selection procedure.

When the number of basis functions that can potentially enter is very large, beginning with forward stepwise selection is the only viable option. Sample size limitations, if nothing else, do not permit fitting with the full (astronomically large) basis set. In fact Breiman's (1989b) implementation of the Π-method employs a forward then backward stepwise approach (using GCV for model selection). It's my guess that in low dimensional applications ($n \lesssim 3$), the Π-method will emerge as a strong competitor to other procedures intended for those settings, especially for certain very highly structured functions. [More detailed comments on the Π-method appear in the discussion of Breiman (1991)].

Golubev and Hasminskii

Golubev and Hasminskii raise the important issue of the theoretical properties of MARS when applied to various classes of smooth functions. I appreciate their inclusion of the known general results in this area. Gaining any theoretical understanding of the performance properties of MARS would likely be an enormous help in improving it.

Buja, Duffy, Hastie and Tibshirani

The discussion by Buja, Duffy, Hastie and Tibshirani describes important enhancements to MARS (and related procedures) that will likely improve both performance and interpretability. I am especially grateful for the discussion of their experiences with MARS on an important real problem. One tends to learn much more about a procedure when it is applied in a setting where the actual answer is the important thing, rather than serving simply as a test bed for the procedure.

A diagnostic tool indicating when a future covariate vector is outside the range of the training data is important. With flexible fitting procedures like MARS the notion of "outside" needs to be broadened. Simply being inside (say) the convex hull of the design may not be enough to be safe. If the density of the design contains extended sparse or empty regions ("holes"), predictions within those regions may be suspect, especially if they are dramatic (far from the response mean). A diagnostic procedure that could indicate

such situations would be an invaluable companion to any flexible fitting procedure in practical applications.

Buja and Duffy's experiences with using MARS raised a series of questions put forward in their discussion. I have observed some of the phenomena they relate in my experiences with MARS. I will try to address the issues they raise in their points 1–5 of Section 1 of their discussion.

1. The recommendation at the end of Section 3.6 for choice of $M_{\max}$ (maximum number of basis functions) was intended to serve as a starting guide. It will not likely be the final best choice in all situations. As was done by Buja and Duffy, I recommend some experimentation with several values, using cross-validated performance as a guide, before a final model is chosen.
2. The "cost" parameter d (32) is the primary smoothing parameter of the MARS procedure. [Secondary less influential smoothing parameters are $M_{\max}$ and $L(\alpha)$ (43)]. Increasing its value will cause fewer basis functions (knots) to be entered, thereby increasing smoothness. The penalty increase (in the GCV criterion) that it regulates is intended to compensate for the increased degree of data fitting associated with the stepwise basis function selection. As such its value should depend on the degree of optimization performed. This in turn has a mild dependence on other parameters that limit the optimization procedure such as $M_{\max}$, L (43), and mi (maximum interaction order). Its greatest dependence can be, however, on the underlying function $f(\mathbf{x})$ (1) when it is highly structured. The default value $d = 3$ [$d = 2$ for additive modeling ($mi = 1$)] was chosen (mainly through simulation studies) to be appropriate for the null situation, $f(\mathbf{x}) =$ constant. This is a conservative choice and is appropriate as a starting value in order to avoid the embarrassment (discussed by Breiman) of fitting pure noise with a structured approximation. The results shown in Tables 2a and 2b indicate that it works reasonably well for this purpose. Theoretical results quoted in Owen's discussion, however, suggest that this choice might be too conservative for highly structured functions (far from null situations). The intuitive reason is that such highly structured functions give rise to highly preferred knot locations and the sampling functions induced by the noise are not strong enough to cause (initial) knots to be placed far from these preferred locations. The optimization procedure is thereby (indirectly) restricted as to where it can place knots, reducing the variance of the estimates. In the null case the true underlying (constant) function has no preference at all for where knots are located and knot placement is totally driven by the noise, inducing the most variance. The results quoted in Section 3.6 were based on simulation studies involving very smooth weakly structured functions, where the default values motivated by the null case seemed to work reasonably well. The situation faced by Buja and Duffy seems to be one involving a highly structured function with sharp threshold effects. The default values may not be appropriate in this case.
3. In the present implementation of MARS, d (32) is not adjusted for $M_{\max}$ although it would be reasonable to do so. The oddities observed by Buja and Duffy may stem from implementation details associated with the backward stepwise strategy like those pointed out in Owen's discussion.
4. The anomalous behavior associated with the piecewise-cubic fits observed by Buja and Duffy is (as they concluded) due to the (apparently) highly structured nature of their function. This same effect was observed in the semiconductor design example in Section 4.7. It also has sharp threshold effects that cause problems for the derivative smoothing. (Note that Section 4.7 was added to a later version of the manuscript than was supplied to the discussants.) This anomalous behavior is in some sense a blessing in disguise. It stems from the ability of the piecewise-linear basis to approximate sharp thresholding with a small number of basis functions; a single threshold can be captured with at most two knots.

   Imposing a higher level of smoothness (say by using higher order splines) would require placing more knots in the vicinity of each threshold to allow the derivative to change very rapidly there. The piecewise-cubic approximation used in MARS has an even greater disadvantage since it attempts derivative smoothing using only the knot locations derived from the piecewise-linear fitting. It therefore does not have additional knots near each threshold to help it rapidly adjust the derivative. Thus it even

more dramatically over smooths the derivative in such regions. Approximations that impose a higher level of smoothness of course perform best when the underlying function is very smooth; that is, functions that nowhere have locally high second derivatives. For such functions, piecewise-linear splines will require several knots over the data interval to adequately approximate the smoothly changing (first) derivative. The derivative smoothing strategy (Section 3.7) can then use these to fashion a piecewise cubic basis to remove the resulting mild derivative discontinuities.

As observed by Buja and Duffy, anomalous behavior of the piecewise cubic fit (when it occurs) is readily diagnosed by simply examining the respective GCV scores of the two models. Simulation studies on relatively smooth functions indicate that the continuous derivative approximation tends to fit the data at hand slightly worse (since the knots are optimized for the linear basis) but has slightly better predictive performance. Experience with sharply structured functions (such as those involving threshold effects) indicate that the piecewise-cubic basis fits both the data at hand and future data badly. This suggests a strategy of accepting the cubic fit if its GCV score is at most slightly (10%–20%) worse than the piecewise-linear fit; otherwise use the linear basis fit. Considerably worse GCV scores for piecewise cubic fits can serve as a useful diagnostic indicating sharp structure in the response. One (but not the only) source for such sharp structure can be single or multiple response outliers.

5. The evenness of the growth of the selected model size on $M_{\max}$ (maximum number of basis functions) likely depends on a variety of different characteristics associated with each particular problem. These include sample size, properties of the design, and the true underlying function.

Gu and Wahba

Gu and Wahba present a lovely concise descriptive summary of the "smoothing spline" approach to fitting functions of several variables. Researchers at the "Madison spline school" led by Wahba have been pioneers in this important area of statistics. Thin-plate and interaction splines represent important contributions that are theoretically attractive and are, in addition, highly competitive in practical settings involving low to moderate dimensionalities and relatively small sample sizes.

In their discussion, Gu and Wahba point out some of the aspects that MARS and interaction splines hold in common. There are also important differences. These differences basically stem from differing motivations associated with intended applications. Applications motivating MARS are similar to those that motivated CART (Breiman, et al., 1984), namely (relatively) large complex data sets where little is known about the true underlying function $f(\mathbf{x})$ (1). In such settings one needs a general procedure that requires as its only input specification the training data, from which it produces a reasonably accurate and interpretable approximation $\hat{f}(\mathbf{x})$. If such a procedure is successful in this, the user may wish to use the derived information to further refine the model. This can be done by reapplying the general procedure in various restricted modes, where the restrictions are guided by the results of the earlier general application. For example, if a MARS run indicates that certain variables enter additively or participate only in limited interactions, further tuning of the model would be done in the presence of these constraints. This refinement can also be done by applying a different less general procedure that requires a more detailed input specification, with these details provided by the output of the previously applied general procedure.

Application of interaction splines requires as part of its user specified input an ANOVA decomposition (10) (24). That is, the user must tell the procedure what variables enter the model, which specific variables (if any) they interact with, and the levels of those interactions. Since these reflect properties of the true underlying function $f(\mathbf{x})$ (1) the quality of the approximation can depend strongly on this input in any given situation. Statistical and computational considerations strongly limit the number of ANOVA functions that can be entered into an interaction spline model. In low dimensional settings this problem is mitigated by the (relatively) small number of ANOVA functions that can potentially enter. The user can simply enter them

all or experiment with different possible subsets. With increasing dimension of the predictor variable space the number of possible ANOVA functions grows very rapidly and this is no longer a viable strategy.

MARS does not require an ANOVA decomposition as part of its input specification. Using only the data, it provides an ANOVA decomposition as part of its output. As noted in the discussions, this can be a valuable tool for interpreting the approximating function, and (to the extent it reflects the true underlying function) also the system under study that generated the data. If it turns out that MARS produces an ANOVA decomposition with a small number of ANOVA functions each involving only a few of the variables (as is often the case), this information can be used as input for an interaction spline model. Whether this will result in improved accuracy will largely depend on the smoothness properties of the underlying function.

Other differences between the two approaches also largely center on issues of generality. In its most general application, interaction splines associate a single smoothing parameter with each specified ANOVA function (10), (12), (24), that constrains its overall global smoothness. As a consequence of its adaptive knot placement algorithm, MARS attempts to adapt its smoothness constraint locally within each ANOVA function that it produces. Note that this will only provide a potential benefit if there is sharp structure present, as was the case in the semiconductor design example (Section 4.7) and in the problem encountered by Buja and Duffy (discussion of Buja, Duffy, Hastie, and Tibshirani). In cases where the (true underlying) ANOVA functions are all globally very smooth, this local adaptability can be counterproductive in that the associated increased variance is not offset by decreased bias. For these cases model refinement using an interaction spline approach might provide a real benefit. (Applying MARS with a strong global second derivative penalty may also help in these situations.)

The method for incorporating categorical variables into interaction spline models described by Gu and Wahba basically encodes them into real valued (0/1) dummy variables. Regularization is provided by grouping together the coefficients associated with each original categorical variable and shrinking them towards a common mean by penalizing the variance of their solution estimates. This is a clever idea. Categorical variables are incorporated into MARS using a different strategy (Friedman, 1990). This strategy is motivated by that of CART, which does not use dummy variables, but instead attempts to find subgroups of categorical values within each variable over which the response (conditioned on the rest of the model) is roughly constant (small variance). The differences between these two approaches for categorical variables reflect the respective differences between the two methods for ordinal variables. Interaction splines apply a global smoothness constraint on each variable or ANOVA function, whereas adaptive spline strategies like MARS, attempt to adjust the smoothing constraint locally within each variable or ANOVA function, to adapt to possible sharp local structure.

Barron and Xiao

Barron and Xiao suggest several clever modifications to the MARS procedure, some of which will likely improve its performance. Like Hastie (1989) and Buja, Duffy, Hastie and Tibshirani (discussion herein), they propose the imposition of a global roughness penalty on the solution. Their criterion penalizes increasing (squared) first, rather than second, derivatives but it should produce similar results. In fact, the Barron-Xiao penalty has an especially nice intuitive appeal. As discussed in the rejoinder to Breiman, a global roughness penalty will likely provide substantial benefit when the true underlying function varies quite smoothly over the predictor space with no (relatively) sharp local structure anywhere. There is however no free lunch. While helping with such very gentle functions, the price paid for using a strong (global) roughness penalty will be to inhibit the ability of MARS to capture local structure when it is present (without degrading the fit elsewhere). At least for (initial) exploratory applications, the penalty should be set just large enough to inhibit only possible wild behavior (see Breiman rejoinder) and not limit the flexibility of the procedure. Considerations are different for procedures based on global polynomials (such as MAPS). Since polynomials

(unlike splines) already have inherent difficulty dealing with local sharp variation of the underlying function (see below), nothing (additional) is lost by imposing a moderate to strong global roughness penalty on them.

Barron and Xiao also raise the issue of model selection and suggest alternatives to the GCV criterion used in the present implementation of MARS. As mentioned in the rejoinder to Breiman, MARS is in no way wedded to the GCV criterion. More effective model selection can only help improve performance. The BIC/MDL criterion suggested by Barron and Xiao has strong intuitive appeal and, along with ordinary cross-validation (proposed by Breiman), will be tested in the context of MARS. The evidence provided for its superiority in Table B of their discussion however is only partially convincing. This is due to the fact that the criterion was used there in conjunction with the true underlying error variance ($\sigma^2 = 1$, known only because this is a simulated example) rather than trying to estimate it from the data at hand. Knowing that the true error is homoscedastic, and the value of its variance, provides a strong advantage to any model selection procedure. If an estimated value of σ^2 happens to be too large the model selection criterion will tend to include too few terms whereas conversely, too many will be entered. The variance of estimates of σ^2 can be quite high, especially in conjunction with (nonlinear) flexible fitting procedures. There may also be considerable bias unless one knows how to correct for the (basis function) selection aspect. There is no obvious analog here for obtaining an unbiased estimate based on the largest possible model.

The MAPS procedure introduced by Barron and Xiao closely follows the MARS strategy and as a consequence inherits many of its characteristics. The only basic difference is the substitution of global polynomials in place of (adaptive) splines. If one is constrained to produce only polynomial models this represents an attractive approach. If not, the difficulties associated with polynomial approximations can impose (sometimes rather strong) limitations.

Global polynomials are held in somewhat low esteem as general tools for function approximation (or estimation) in both the mathematical approximation and statistical curve and surface estimation literatures. Nearly all the recommended procedures that have emerged so far possess in common a locality property; the estimate at a point is most strongly influenced by (training) observations close to that point and observations further away have little or no influence. This gives rise in large part to their flexibility. They can respond to (sharp) local properties of the function without affecting the fit everywhere else. Global polynomial fits do not share this property; the function estimate at a point can be strongly influenced by data points very far away from it in the predictor space. As a consequence, locally sharp structure anywhere can influence the fit everywhere. This is the likely motivation for the quote from J. W. Tukey, "polynomials cut peoples' throats," and the observation by deBoor (1978), "If the function to be approximated is badly behaved anywhere in the interval of approximation, then the approximation is poor everywhere. This global dependence on local properties can be avoided using piecewise polynomials (splines)." [See deBoor (1978), Chapter II for a nice discussion of the limitations of polynomial approximations.]

If the true underlying function $f(\mathbf{x})$ (1) is everywhere gently varying with no sharp structure anywhere then approximations based on global polynomials perform very well. In fact if $f(\mathbf{x})$ happens to actually be a polynomial (or very close to one) then they will give the best performance. However, local methods such as splines also do quite well in these settings. Thus, if one expects to only encounter situations such as this, there is (as noted by Barron and Xiao) little to choose between them. If, however, one wants to maintain the additional ability to adequately deal with more structured functions then local methods might be preferred.

The MARS procedure is based on (adaptive) spline functions because they emerge naturally as a generalization of recursive partitioning. It thereby inherits the attractive properties of the recursive partitioning approach discussed in Sections 2.4.2 and 6.0. These include local variable subset selection and automatic local adjustment of the degree of smoothing within each ANOVA function produced. Substituting polynomials for the adaptive spline functions sacrifices the local aspects of both these properties; only global variable subset selection, and automatic adjustment of global smoothing on each ANOVA function are retained.

If in a particular situation the nature of the underlying function happens to be such that this additional flexibility of MARS gives rise to no advantage, then there is little to choose between MARS and MAPS. Therefore, issues of generality will (as above) likely guide the choice.

Using adaptive splines also causes MARS to inherit the ability to isolate local sharp structure and deal with it separately without affecting the fit in other regions of the predictor space. Buja and Duffy (discussion by Buja, Duffy, Hastie and Tibshirani) report this as a crucial advantage in their application. Also, the semiconductor component example of Section 4.7 likely benefits from this property.

The examples of Sections 4.2 and 4.3 were chosen largely because they do not play to the particular strengths of MARS. They are globally very smooth quite gentle functions and (as noted by Barron and Xiao) global low order polynomials make up a substantial part of their definitions. One would expect approximations based on low order polynomials to do very well here. This is partially verified by the results presented by Barron and Xiao (Tables B and C). Perhaps somewhat surprising is the degree of competitiveness displayed by adaptive splines in these situations. For these examples, local variable subset selection and adaptive local smoothing provide little or no advantage. The MAPS procedure on the other hand has the additional benefit of a global roughness penalty constraint (not yet incorporated into MARS) which as noted above is a real help with very smooth functions. It also has the advantage (Tables B and C) of being supplied with the true underlying error variance. As discussed above, this provides the model selection criterion with an important advantage. This can be seen by comparing Tables A and B of Barron and Xiao's discussion.

The results presented for MARS in Tables 3 and 5a were obtained using its default smoothing parameter value $d = 3$ (32) without any attempt to even estimate a best value from the data. The best value is controlled by the underlying error variance which for the purposes of illustration was assumed not to be known. As discussed in the rejoinder to Breiman, using this default value may be reasonable for initial exploratory work, but one may wish to refine the fit in the later stages of the analysis by estimating a better value through cross-validation. To get an idea of whether this can give rise to substantial improvement the simulation study on the example of Section 4.2 was rerun ($N = 200$ only) but this time using cross-validation to estimate the smoothing parameter rather than using the default value. Table 15 shows the results based on 100 replications. Also shown are the average estimated smoothing parameter values. (The quantities in parentheses are the standard deviations over the 100 trials.)

Table 15

Accuracy of MARS applied to the example of Section 4.2 ($N = 200$) with the smoothing parameter d (32) selected through cross-validation.

mi	$\overline{ISE}$	$\overline{PSE}$	$\bar{d}$
1	.015 (.013)	.15 (.01)	4.9 (1.2)
2	.030 (.015)	.16 (.01)	5.7 (1.3)
10	.031 (.016)	.16 (.01)	5.8 (1.3)

Table 15 (column 4) shows that the cross-validation procedure was choosing (on average) considerably larger smoothing parameter values that the default ($d = 3.0$) in this case. This reflects the very smooth nature of the underlying function. Comparing Tables 3 and 15 shows substantial improvement [24% in $\sqrt{\overline{ISE}}$ (58)] only for $mi = 1$ (additive model). However, as Table 4 indicates, this is the one most likely to be chosen. Comparing Table 15 with Table B of the discussion of Barron and Xiao shows that even without a global roughness penalty and knowledge of the true underlying error, adaptive splines compare favorably with global polynomials in this setting.

The motivations put forward by Barron and Xiao for substituting global polynomials for adaptive splines in MARS are mainly computational and (to a lesser extent) interpretability. An implementation based on

polynomials did gain a computational advantage over one based on adaptive splines when the old implementation strategy described in Section 3.9 was used for the latter. This is, however, no longer the case with the new implementation discussed in the rejoinder to Stone. The computation for both global polynomials and adaptive splines (new implementation) is proportional to $nNM^3_{\max}$. The actual relative computing speeds will depend on implementation details but are not likely to be very much different. Adaptive splines may have a slight advantage since fewer basis functions $M_{\max}$ are often needed for comparable accuracy. This is a consequence of the fact that each basis function included in a MARS model is adapted to the data through the adjustment of its knot locations.

Interpretability of the resulting approximation was an important design goal motivating the MARS approach. The ANOVA decomposition (Section 3.5) and slicing (Section 4.7) are intended as important interpretational aids. The MAPS procedure by closely following the MARS strategy inherits these aspects. The most powerful interpretational aids in understanding each ANOVA function are likely to be graphical representations (curves or surface displays). For these, whether the approximation is internally represented by a polynomial or spline function is of little consequence. If the polynomial representation involves more that a very few terms and/or high degrees some may argue that spline representations are more interpretable, especially since they are likely to be more parsimonious (see discussion of Buja, Duffy, Hastie, and Tibshirani). On the other hand, if a function can be approximated by a few low-degree polynomials, those familiar with polynomials may feel more comfortable interpreting them. Such interpretations can be misleading however since (except for artificially constructed examples) the corresponding true underlying function is seldom really a polynomial.

One very important aspect in interpreting these approximations is a small number of resulting ANOVA functions. In this respect adaptive splines may have a distinct advantage (as noted above). In the Portuguese olive oil example (Section 4.5) the MARS model resulted in four ANOVA functions, involving only three variables (Table 10), that can be represented by only two (surface) plots (Figure 5). While achieving comparable accuracy in this case, the polynomial approximation produced by MAPS was far more complex as reflected in Table E (and F) of the Barron-Xiao discussion. The underlying function for this example is also globally quite smooth as can be seen in Figure 5. [Note that Figure 5 displays the log-odds. The underlying function estimate $1/(1 + e^{-\hat{f}(x)})$ is correspondingly much more gentle.]

For those who happen to have a strong preference for approximations based on polynomials (or other global basis functions) but would still like to retain the flexibility of the adaptive spline approach there are several possibilities. One simple possibility would be to apply both to the data. If they provide comparable estimated accuracy then the user could interpret the one he/she finds to be most understandable. Perhaps a more elegant approach would be to include both global functions (such as polynomials), and adaptive splines, in a hybrid MARS modeling strategy. To enhance interpretability one could (optionally) forbid interactions between the global and spline basis functions. An incremental penalty might be attached for including adaptive splines to reflect the user's preference for the global basis functions. The global basis function part of the resulting approximation could then be interpreted as reflecting the very smooth aspects of the underlying function whereas the spline part (in this case) would reflect possible local sharp structure. (Such a hybrid strategy is easily added to the MARS program as a user option.)

The "feed-forward" idea proposed by Barron and Xiao is quite intriguing. It would allow the procedure (MARS or MAPS) to more rapidly build up (synthesize) higher order interaction terms but in a constrained manner. The resulting models (if feed-forward inputs are selected) will be far more complex and difficult to interpret. Whether this approach will give rise to substantially improved prediction accuracy in some situations awaits the results of further investigation.

Additional References

Breiman, L. (1989c). Submodel selection and evaluation in regression I. The X-fixed case and little bootstrap. Technical Report 169, Statistics Department, University of California at Berkeley.

Breiman, L. (1991). The II-method for estimating multivariate functions from noisy data. *Technometrics*. To appear.

Efron, B. (1983). Estimating the error rate of a prediction rule: improvement on cross-validation. *J. Amer. Statist. Assoc.* **78**, 316–331.

Friedman, J. H. (1990). Estimating functions of mixed ordinal and categorical variables using multivariate adaptive regression splines. Technical Report LCS 107, Statistics Department, Stanford University.

Hastie, T. (1989). Discussion of Friedman and Silverman (1989), *Technometrics* **31**, 23–29.

Lewis, P. A. W. and Stevens, J. G. (1990). Nonlinear modeling of time series using multivariate adaptive regression splines (MARS). Technical report, Naval Postgraduate School, Monterey, CA.