SLAC–PUB–4676
July 1988
(M)

FITTING FUNCTIONS TO NOISY DATA IN HIGH DIMENSIONS*

Jerome H. Friedman

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

Abstract

Consider an arbitrary domain of interest in n-dimensional Euclidean space and an unknown function of n arguments defined on that domain. Suppose we are given the value of the function (perhaps perturbed with additive noise) at some set of points. The problem is to find a function that provides a reasonable approximation to the unknown one over the domain of interest. This paper presents a brief review of current methodology aimed at dealing with this problem, and presents a new technique – multivariate adaptive regression splines – that has the potential to overcome some of the limitations of previous approaches.

1.0. Introduction

Suppose a system under study can be described (over some domain $D \in R^n$) by

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

where y is a response or dependent variable of interest, $x_1, \cdots, x_n$ are a set of explanatory or independent variables, and f is a (deterministic) single valued function of its n-dimensional argument. The quantity ϵ is an additive random or stochastic component that (if nonzero) reflects the fact that y depends on quantities other than $x_1 \cdots x_n$ that are also varying. We are given a set of values $\{y_i, x_{1i}, \cdots, x_{ni}\}_1^N$, $(x_{1i}, \cdots, x_{ni}) \in D$, (training sample) and the purpose of the exercise is to obtain a function $\hat{f}(x_1, \cdots, x_n)$ that provides a reasonable approximation to $f(x_1, \cdots, x_n)$. Here reasonable usually means accurate since one often wants to use $\hat{f}$ to approximate f at other

* Work supported in part by the Department of Energy contract DE–AC03–76SF00515

Special Invited paper presented at the 20th Symposium on the Interface: Computing Science and Statistics, Washington, D.C., April 20–29,1988

points not part of the training sample. If in addition one wants to use $\hat{f}$ to try to understand the properties of f (and thereby the system that provided the data) then the interpretability of the representation of $\hat{f}$ is important. It is also sometimes important that $\hat{f}$ be rapidly computable. In addition, for some applications it is important that $\hat{f}$ be a smooth function of its argument; that is, at least its low order derivatives exist everywhere in D.

In low dimensional settings ($n \leq 2$) successful developments have occurred in two general directions: piecewise polynomials and local averaging. The basic idea of piecewise polynomials is to approximate f by several generally low order polynomials each defined over a different subregion of the domain D. The approximation is required to be continuous, and sometimes have continuous low order derivatives. The tradeoff between smoothness and flexibility of the approximation $\hat{f}$ is controlled by the number of subregions (knots) and the order of the lowest derivative allowed to be discontinuous at region boundaries. The most popular piecewise polynomial fitting procedures are based on splines. [See deBoor (1978) for a general review of splines and Schumacker (1976), (1984) for reviews of some two-dimensional extensions.]

Local averaging approximations take the form

$$\hat{f}(x) = \sum_{i=1}^{N} K(x, x_i)\, y_i \tag{2}$$

where $K(x, x')$ (called the kernel function) usually has its maximum value at $x' = x$ with its absolute value decreasing as $|x - x'|$ increases. Thus, $\hat{f}(x)$ is taken to be a weighted average of the y_i where the weights are larger for those observations that are close or local to x. For $n > 1$ the kernel is usually taken to be a function of the Euclidean distance between the points

$$K(\mathbf{x}, \mathbf{x}') = K\left[\left(\sum_{i=1}^{n} |x_i - x_i'|^2\right)^{1/2}\right]. \tag{3}$$

Local averaging procedures have received considerable attention in the statistical literature beginning with their introduction by Parzen (1962). Stone (1977) has shown that this approach has desirable asymptotic properties. They have also seen interest from the mathematical approximation literature [Shepard (1964), Bozzini and Lenarduzzi (1985)]. Roughness penalty methods [smoothing ($n = 1$) and thin plate ($n = 2$) splines] are closely related to kernel methods based on Euclidean distance [see Silverman (1985) and Schumaker (1976)].

The direct extension of piecewise polynomials (splines) or local averaging methods 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 points are needed to densely populate Euclidean spaces of increasing dimension. In the case of spline approximations, extension to higher dimensions is accomplished through tensor products of univariate spline functions. These functions are

associated with a grid of points defined by the outer product of knot positions on each independent variable. For a given number of knots $\bar{K}$ on each variable, the size of the grid, and thus the number of approximating basis functions, grows as K^n. For example, in six dimensions a (tensor product) cubic spline with only *one* interior knot in each variable has 15,625 coefficients to be estimated. That number in ten dimensions is approximately 10^7. Even though only one interior knot per variable might be considered a very coarse grid, it still requires a very large number of data points to estimate the corresponding spline approximation. Finer grids require many more points.

Local averaging methods suffer a similar fate as the dimension of the function argument space increases. For example, let D be the unit hypercube in R^n and consider a uniform kernel with hypercubical support and bandwidth (edge length) covering 10 percent of the range of each coordinate. Then, if the data are roughly uniformly distributed in R^n, the kernel will (on average) contain only $(0.1)^n$ of the sample, thereby nearly always being empty for moderate to large n. If, on the other hand, one adjusts the size of the neighborhood (bandwidth) to contain 10 percent of the sample, it will cover (on average) $(0.1)^{1/n} \times 100$ percent of the range of each variable, resulting in a very crude approximation.

This problem of the inherent sparsity of practical sampling in high dimensions basically limits the straightforward application of both piecewise polynomials and local averaging methods in these settings. It does not, however, limit theoretical investigation. It is straightforward to imagine arbitrarily densely sampling of high dimensional spaces. Asymptotic theoretical calculations can then be done. [See Stone (1977) for pioneering work in this area.] The (practical) difficulty lies only in obtaining the corresponding large samples required for accurate approximations. It should be noted in addition, that local averaging approximations (and to a lesser extent tensor product splines) are slow to compute and difficult to interpret.

The curse-of-dimensionality is fundamental and cannot be directly overcome. If the true underlying function $f(x_1, \cdots, x_n)$ (1) exhibits strong variation of no special structure on all of the variables in every part of the domain D, then accurate approximation with feasible sample sizes is not possible. Fortunately, very few functions of interest exhibit behavior quite this dramatic. Generally there is some (sometimes known, more often unknown) special structure associated with the function that can be exploited by a sufficiently clever algorithm to reduce the complexity and thereby achieve more accurate approximation.

Function approximation in high dimensional settings has been pursued mainly in statistics. The principal approach taken there has been to fit an especially simple parametric form to the training sample. The most common parameterization is the linear function

$$\hat{f}(x_1, \cdots, x_n) = \alpha_0 + \sum_{i=1}^{n} \alpha_i x_i. \tag{4}$$

This is not likely to produce a very accurate approximation to very many functions in R^n, but it has the virtue of requiring relatively few data points, it is easy to interpret, and it is rapidly

computable. Also, if the stochastic component ϵ (1) is large compared to f, then the variability of the estimate dominates, and the systematic error associated with this simple approximation is not the most serious problem.

Recently, the linear model has been generalized nonparametrically to the so-called additive model

$$\hat{f}(x_1, \cdots, x_n) = \sum_{i=1}^{n} f_i(x_i) \tag{5}$$

[Friedman and Stuetzle (1981), Breiman and Friedman (1985), Hastie and Tibshirani (1986), Friedman and Silverman (1987)]. Here the $\{f_i(x_i)\}_1^n$ are each (different) smooth but otherwise arbitrary functions of a single variable. Although additive models are still not able to accurately approximate very general functions in R^n, they do constitute a much richer class than the simple linear approximation (4). They share the high interpretability of the linear model (one can view the univariate functions f_i) and they are not overly difficult to compute.

Linear and additive approximations lack generality in that they have limited ability to adapt to a wide variety of multivariate functions f. Also, as the sample size increases there is a limit to the accuracy of the approximation (unless the true underlying function happens to be exactly linear or additive over D).

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, Friedman, Olshen, and Stone (1984)], and projection pursuit [Friedman and Stuetzle (1981), Friedman, Grosse, and Stuetzle (1983), Friedman, (1985)].

Projection pursuit uses an approximation of the form

$$\hat{f}(x_1, \cdots, x_n) = \sum_{m=1}^{M} f_m \left(\sum_{i=1}^{n} \alpha_{im} x_i \right), \tag{6}$$

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. It can be shown [see Diaconis and Shahshahani (1984)] that any smooth function of n variables can be represented by (6) 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 (1985).] 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 interprative value (for small M) in that one can inspect the functions f_m and the corresponding 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.

Recursive partitioning approximations take the form

$$\hat{f}(x_1, \cdots, x_n) = \sum_{m=1}^{M} f_m(x_1, \cdots, x_n) I[(x_1, \cdots, x_n) \in R_m]. \tag{7}$$

Here $I(\cdot)$ is 0/1 valued function that indicates the truth of its argument and $\{R_m\}_1^M$ are disjoint subregions representing a partition of D. The functions f_m are generally taken to be of quite simple parametric form. The most common is a constant function

$$f_m(x_1, \cdots, x_n) = a_m \tag{8}$$

[Morgan and Sunquist (1963) and Breiman, et al. (1984)]. Linear functions (4) have also been proposed [Breiman and Meisel (1976) and Friedman (1979)], but they have not seen much use. The partitioning is developed in a recursive manner. At each step, M, all existing subregions $\{R_m\}_1^M$ are optimally split into two subregions along one of the variables. The particular split that yields the best improvement in the fit is taken to define two new regions and the parent region (that was split) is deleted. (The starting region is the entire domain D.) The number of subregions in the partition is thereby increased by one at each step. A backwards stepwise strategy for determining the final number of regions is detailed in Breiman, et al. (1984).

The recursive partitioning approach has the potential to provide acceptable approximations in high dimensionalities provided the underlying function has low "local" dimensionality. That is, even though the function f (1) may strongly depend on all of the variables, in any local region of the domain the dependence is strong on only a few of them. These few variables may be different in different regions. Another assumption inherent in the recursive partitioning strategy is that interaction effects have marginal consequences. That is, a local intrinsic dependence on several variables, when best approximated by an additive function, does not lead to a constant model. This is nearly always the case.

Recursive partitioning using piecewise constant approximations (8) 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)]. 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 (8)] and additive functions (5) in more than a few variables (piecewise constant or piecewise linear approximation). 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.

2.0. Multivariate Adaptive Regression Splines.

This section describes a new method of adaptive computation for approximating functions in high dimensionalities. Although it is an extension of the additive modeling (5) procedure developed by Friedman and Silverman (1987), it appears closest in spirit to the adaptive nature of the recursive partitioning approach. Unlike recursive partitioning, however, it produces strictly continuous approximations (with continuous derivatives if desired), it easily approximates linear and additive functions, and it can be represented in a form that permits separate identification of the additive and (multiple) interaction effects associated with the variables that enter into the model.

The approximation takes the form of an expansion in multivariate spline basis functions,

$$\hat{f}(x_1, \cdots, x_n) = \sum_{m=0}^{M} a_m B_m(x_1, \cdots, x_n) \tag{9a}$$

with

$$B_0(x_1, \cdots, x_n) = 1, \tag{9b}$$

$$B_m(x_1, \cdots, x_n) = \prod_{k=1}^{K_m} b(x_{v(k,m)}|t_{km}), \quad m \geq 1. \tag{9c}$$

The $\{a_m\}_0^M$ are the coefficients of the expansion. Each multivariate spline basis function B_m, $m > 0$, is a product of univariate spline basis functions b, each of a single variable $x_{v(k,m)}$, characterized by a knot at t_{km}. The subscripts $v(k,m)$ label the explanatory variables, thereby taking values in the range $1 \leq v(k,m) \leq n$; K_m takes values in the same range $1 \leq K_m \leq n$ and determines the number of factors (univariate spline basis functions) comprising the corresponding B_m. The multivariate spline basis functions B_m are adaptive in that the number of factors K_m, the variable set $V(m) = \{v(k,m)\}_1^{K_m}$ and the knot set $\{t_{km}\}_1^{K_m}$ are all determined by the data.

The approximation is developed in a forward/backwards stepwise recursive manner in analogy with the recursive partitioning approach. Given $\{B_m\}_0^{M-1}$ the Mth term takes the form

$$B_M(x_1, \cdots, x_n) = B_\ell(x_1, \cdots, x_n) b(x_v|t) \tag{10}$$

with $0 \leq \ell \leq M-1$. That is, the next term B_M is taken to be the product of a univariate spline basis function with one of the previously defined multivariate spline basis functions B_ℓ $(0 \leq \ell \leq M-1)$. The values for v, t, and ℓ are chosen so as to jointly maximize the goodness-of-fit of the resulting approximation (see Section 2.2). The defining variable x_v for the new basis function $b(x_v|t)$ is restricted to be one that does not appear in the selected B_ℓ, so that the same variable does not appear more than once in any B_m $(0 \leq m \leq M)$. The resulting optimal values v^*, t^*, and ℓ^* are then used to form the new multivariate spline basis function

$$B_M = \prod_{k=1}^{K_M} b(x_{v(k,M)}|t_{kM})$$

with $K_M = K_{\ell^*} + 1$, $v(K_M, M) = v^*$, $t_{K_M M} = t^*$, and the rest of the factors taken from B_{ℓ^*}.

One of the requirements for this strategy to be computationally feasible is that each univariate basis function be defined by the location of a single knot t_{km}. We therefore use the truncated power basis representation for the (univariate) splines

$$b^{(q)}(x|t) = (x-t)_+^q \tag{11}$$

where q is the order of the spline which controls the degree-of-continuity of the approximation. The subscript denotes the non-negative part. (This basis is known to produce numerical problems, especially for $q > 1$, so a great deal of care must be taken in the implementation.)

This forward stepwise construction of the multivariate spline basis (9) (10) is continued until $M = M_{\max}$ terms have been entered into the approximation. This process yields a sequence of $M_{\max}$ models, each with one more term than the previous one in the sequence. Each model in the sequence has an associated badness-of-fit score (see Section 2.2). That model with the lowest badness-of-fit score is then subjected to a backwards stepwise deletion strategy [see Friedman and Silverman (1987), Section 2.1], to obtain the final model. The upper limit $M_{\max}$ should be taken to be large enough so that the minimizing model is not too close to the end of the sequence. Due to the forward stepwise nature of the procedure it is possible for the badness-of-fit to locally increase a bit as the sequence proceeds, and then start to decrease again.

If one makes the restriction $K_m = 1$ (9c) for all m (that is, always setting $\ell = 0$ rather than including it in the optimization) the approximation becomes a sum of functions, each of a single variable. This is, of course, an additive model (5) and this strategy reduces to the smoothing and additive modeling technique introduced by Friedman and Silverman (1987). The key ingredient

that advances this approach to general settings is the ability to fit (possibly complex) interactions among the variables through the product terms that are permitted to enter the approximation (9), if required by the fit.

Although originally motivated by the work of Friedman and Silverman (1987) this approximation strategy (9)–(11) has more in common with the recursive partitioning approach (see Section 1.0) to function approximation (7). There is a correspondence between the terms in (9) and the regions in (7). Choosing a previous term for multiplication (10) is analogous to choosing a (previous) region to split in (7). The optimization over v and t in (10) is quite similar to finding the optimal splitting variable and split point for partitioning a region.

The correspondence between this basic approach and recursive partitioning is most easily seen by contrasting the piecewise constant approximation (8) of the latter with the use of $q = 0$ splines (11) in the former

$$b^{(0)}(x|t) = I(x - t). \tag{12}$$

Both methods then produce piecewise-constant approximations in this case, and multiplying (sometimes with constraints) is strictly equivalent to splitting. The two methods, even though being most similar in this setting, do not however produce equivalent approximations. This is basically because unlike recursive partitioning, the subregions induced by (9), (10), (12) are not constrained to be disjoint. At any stage during recursive partitioning, only terminal regions are eligible for splitting, i.e. only those regions defined by the intersections of previous splits (terminal nodes on the current binary tree). With the MARS strategy all previously defined regions – not just terminal ones – are eligible for splitting at any stage of the model building process. The previously defined regions are those represented by the internal nodes of the tree and are unions of subsets of current terminal regions.

The strategy associated with the MARS approach has several important advantages. Foremost among them is that it allows close approximations to many of the common functions that present difficulty to recursive partitioning (e.g. nearly linear or additive functions). Another advantage is its interpretability through its ANOVA representation (see below). The most important advantage of this approach, however, is that by choosing $q > 0$ (11) continuous approximations can be achieved. This has been one of the most serious limitations of recursive partitioning. Choosing a value for $q \geq 1$ causes the approximation to be continuous and to possess continuous derivatives to order $q - 1$.

As with recursive partitioning, this method attempts to use to advantage the fact that interaction effects involving several variables will give rise to non-constant dependencies on at least one of those variables individually. This is because in the forward part of the model building strategy, additive terms and lower order interactions must enter before the corresponding higher order interactions. These lower order terms provide information as to where to place knots to capture the

corresponding higher order ones, and they may in fact be removed (through the backwards deletion process) after the higher order interaction terms are entered.

2.1. ANOVA Decomposition.

The representation of the approximation given by (9), (10), (11) resulting from construction of the model

$$\hat{f}(x_1, \cdots, x_n) = a_0 + \sum_{m=1}^{M} a_m \prod_{k=1}^{K_m} b(x_{v(k,m)} - t_{km})_+^q \tag{13}$$

does not provide much insight into the nature of the approximation. By simply rearranging the terms, however, it is able to provide considerable insight into the predictive relationship between y and $x_1, \cdots, x_n$,

$$\begin{aligned} \hat{f}(x_1, \cdots, x_n) = a_0 &+ \sum_{K_m=1} f_i(x_i) + \sum_{K_m=2} f_{ij}(x_i, x_j) \\ &+ \sum_{K_m=3} f_{ijk}(x_i, x_j, x_k) + \cdots. \end{aligned} \tag{14a}$$

Here the first sum is over all terms involving only a single variable and represents the purely additive component of the model. Each additive function $f_i(x_i)$ can be computed by collecting together all single variable terms involving x_i,

$$f_i(x_i) = \sum_{\substack{K_m=1 \\ i \in V(m)}} a_m B_m(x_i). \tag{14b}$$

Here $V(m)$ represents the variable set $\{v(k,m)\}_1^{K_m}$ associated with the mth term. The second sum in (14a) is over all terms involving exactly two variables and represents the pure first order (two variable) interaction part of the model with

$$f_{ij}(x_i, x_j) = \sum_{\substack{K_m=2 \\ (i,j) \in V(m)}} a_m B_m(x_i, x_j). \tag{14c}$$

Similarly, the third sum represents second order (three variable) interactions with

$$f_{ijk}(x_i, x_j, x_k) = \sum_{\substack{K_m=3 \\ (i,j,k) \in V(m)}} a_m B_m(x_i, x_j, x_k), \tag{14d}$$

and so on. The additive terms can be viewed by plotting $f_i(x_i)$ against x_i as one does with additive modeling. The two variable interaction terms $f_{ij}(x_i, x_j)$ can be plotted using either contour or perspective mesh plots. Higher order interactions (if present) are of course more difficult to view. The corresponding (multivariate) knot locations can, however, provide some insight. We refer to (14) as the ANOVA decomposition or representation of the MARS model because of its similarity to decompositions provided by the analysis of variance of contingency tables.

The ANOVA representation identifies the particular variables that enter into the model, whether they enter purely additively or are involved in interactions with other variables, the order of the interactions, and the other variables that participate in them.

2.2. Model Selection.

As in Friedman and Silverman (1987) we use the generalized cross-validation criterion (Craven and Wahba, 1979)

$$GCV(M) = \frac{1}{N}\sum_{i=1}^{N}[y_i - \hat{f}_M(x_{1i}, \cdots, x_{ni})]^2 / \left[1 - \frac{C(M)}{N}\right]^2 \tag{15a}$$

for model selection where M is the number of terms in (9a) and

$$C(M) = (d+1)M + 1. \tag{15b}$$

Minimization of this criterion is used to select the knot variable and its location at each forward step, the terms to delete in the backwards steps, and the size of the final model. The use of (15b) results in a change of $(d+1)$ "degrees-of-freedom" for each term in the model, one for fitting the least-squares coefficient a_m, and d for the optimization associated with the knot placement. Friedman and Silverman (1987) used $d = 2$. This was motivated somewhat on theoretical grounds but mostly on an empirical basis. This value is too small for generalized MARS modeling since we are, in addition, optimizing over the term index $0 \leq \ell \leq M-1$ at each step as well as the knot location. This produces increased variance that must be accounted for in the model selection. A direct approach would be to estimate an optimal d value for the problem at hand through a sample reuse technique such as the 632 bootstrap (Efron, 1983) or cross-validation Stone (1974).

Another approach is to study the variance directly through a modified bootstrapping technique (Hastie and Tibshirani, 1985). Each bootstrap replication consists of replacing each response value by a standard normal deviate. By construction the true underlying function f is the constant zero, and the mean-squared-prediction error is completely dominated by the variance

$$E(f - \hat{f}_M)^2 = E\hat{f}_M^2 = \text{Var}\ \hat{f}_M$$

or equivalently

$$E(y - \hat{f}_M)^2 = E\hat{f}_M^2 + 1. \tag{16}$$

Since the GCV score (15a) is intended to be an estimate for (16) one can obtain an estimate for $C(M)$ through

$$E(ASR_M) / \left[1 - \frac{\hat{C}(M)}{N}\right]^2 = E\hat{f}_M^2 + 1$$

or

$$\hat{C}(M) = N\left[1 - \left(\frac{E(ASR_M)}{Ef_M^2 + 1}\right)^{1/2}\right] \tag{17}.$$

Here the average-squared-residual, ASR_M, is the numerator in (15a). The expected values in (17) are estimated through repeated bootstrap replications.

A wide variety of simulation studies (not detailed here) using this approach indicate the following.

(1) $C(M)$ is a monotonically increasing function with decreasing slope as M increases.

(2) Using the linear approximation (15b), with $d = 2.5$, is fairly effective, if somewhat crude.

(3) The "best" value for d depends (weakly) on M, N, and the distribution of the covariate vectors.

(4) Over a wide variety of situations, the best value of d lies in the range $2.0 \leq d \leq 3.0$.

(5) The *actual* accuracy of the approximation, in terms of integrated squared error

$$ISE = \int [f(\mathbf{x}) - \hat{f}(\mathbf{x})]^2 dF(\mathbf{x}),$$

depends very little on the value chosen for d in the range $2.0 \leq d \leq 3.0$.

(6) The *estimated* accuracy

$$E[ISE - GCV(M^*)]^2,$$

with M^* being the minimizer of (15), does show a moderate dependence on the choice of d.

The consequence of (5) and (6) is that, although how well one is doing with this approach is fairly independent of d, how well one *thinks* he is doing (based on the optimizing GCV score) does depend somewhat on the values chosen for d. Therefore, a sample reuse technique should be used to estimate the predictive capability of the final model, if it needs to be known fairly precisely.

2.3. Degree-of-Continuity.

Another important choice is the degree of continuity to be imposed on the approximating function, i.e. the value for q in (11). This choice affects the accuracy of the approximation, and the speed and numerical stability of the computation. Friedman and Silverman (1987) used $q = 1$ in conjunction with the knot placement and model selection strategy. This produces a continuous piecewise linear approximation with discontinuous derivatives. Advantages of this approach are much more rapid and numerically stable computation compared to higher values of q. Also, it can provide more accurate approximations in some situations. The main disadvantage is discontinuous first derivatives.

Friedman and Silverman (1987) provide for derivative smoothing by replacing the basis functions $b^{(1)}(x|t)$ (11) by closely related ones with continuous first derivatives:

$$C(x|t_-, t, t_+) = \begin{cases} 0 & x \leq t_- \\ p(x - t_-)^2 + r(x - t_-)^3 & t_- < x < t_+ \\ x - t & x \geq t_+ \end{cases} \tag{18a}$$

with $t_- \leq t \leq t_+$. Setting

$$\begin{aligned} p &= (2t_+ + t_- - 3t)/(t_+ - t_-)^2 \\ r &= (2t - t_+ - t_-)/(t_+ - t_-)^3 \end{aligned} \tag{18b}$$

causes these basis functions to be continuous and have continuous first derivatives. This approximation has discontinuous second derivatives at the side knot locations, t_- and t_+. The central knot t, is placed at the corresponding knot location of $b^{(1)}(x|t)$. The two side knots, t_- and t_+, are placed at the midpoints between adjacent central knots on the same variable thereby minimizing the number of second derivative discontinuities. The (central) knots are placed using the $b^{(1)}(x|t)$ (11) basis, taking advantage of the corresponding speed and numerical stability. The approximation with continuous derivatives is accomplished through using the corresponding piecewise cubic basis (18).

The analogue to this approach in the more general setting of MARS modeling is to perform derivative smoothing in the ANOVA representation (14). Each distinct ANOVA function (14b), (14c), (14d), etc. is smoothed separately. The side knots are placed at the midpoints between the central knot locations as projected onto each variable defining the particular function. For the additive ANOVA functions (14b) this of course reduces to the Friedman and Silverman (1987) strategy. Replacing each $b^{(1)}(x|t)$ (11) by its corresponding $C(x|t_-, t, t_+)$ (18) in the MARS model (13) (14) results in a continuous approximation with everywhere continuous derivatives.

2.4. Knot Optimization.

A natural strategy would be to make each distinct observation abscissa value on each predictor variable a potential location for knot placement. Friedman and Silverman (1987) argue that a more effective strategy is to restrict the number of candidate knot locations to very Lth (distinct) observation abscissa value, with L given by

$$L(p, N) = -\log_2 \left[-\frac{1}{pN} \ell n(1-\alpha) \right] / 2.5 \tag{19}$$

and $0.05 \leq \alpha \leq 0.01$. The considerations that lead to this result do not change when one considers the more general MARS setting.

2.5. Computational Considerations.

In order for any method to be practical it must be computationally feasible. If implemented in a straightforward manner the approximation strategy we propose would require prohibitive computation. A full $M + 1$ parameter linear least squares fit for he coefficients $\{a_m\}_0^M$ must be performed to evaluate the model selection criterion (15). This must be done at every potential knot location on every variable for all M (previous) terms at each step M. The only way this can be made to be computationally feasible is through updating formulae. That is, given the solution fit at one potential knot location, the solution at the next one can be obtained through rapidly computable simple updates of the previous solution. Friedman and Silverman (1987, Section 2.3) derived updating formulae for the quantities that enter into the normal equations of the least squares fit for the additive modeling case. Analogous updating formulae can be derived for the more general case of MARS modeling. Use of these updating formulae reduce the computation from

being proportional to M^4pN^2/L to M^3pN/L. As a point of reference, the computation for the three examples (Section 3) each required about two minutes on a SUN Microsystems model 3/260.

3.0. Examples.

This section provides four illustrations of MARS modeling. The data are simulated so that the results can be compared with the known (generated) truth. The first and fourth examples are purely contrived, whereas the middle two are taken form electrical engineering. In all examples the smoothing parameter d (15b) was taken to be $d = 2.5$. (The software automatically reduces it to $d_a = 0.8d = 2.0$ for additive modeling.) The minimum number of observations between knot locations was determined by (19). In all examples the explanatory variables were standardized to aid in numerical stability. (The MARS procedure is, except for numerics, invariant to the predictor variable scales.) The response variable was also standardized so that the GCV score would be an estimate for the fraction of unaccounted for variance ($e^2 = 1 - R^2$).

3.1. Simple Function of Ten Variables.

For this example, $N = 100$ covariate vectors were uniformly generated in a $n = 10$ dimensional unit hypercube. Associated with each such covariate vector is a response value generated as

$$\begin{aligned} y_i = {} & 0.02e^{4x_{1i}+3x_{2i}} + 5\sin(\pi x_{3i}/2) \\ & + 3x_{4i} + 2x_{5i} + 0 \cdot x_{6i} + 0 \cdot x_{7i} + 0 \cdot x_{8i} \\ & + 0 \cdot x_{9i} + 0 \cdot x_{10,i} + \epsilon_i, \quad 1 \le i \le 100, \end{aligned} \tag{20}$$

with the ϵ_i generated from a standard normal distribution. The ratio of standard deviations of the signal to the noise is 3.08 so that the true underlying function accounts for 91% of the variance of y.

The underlying function (20) consists of an interaction in the first two variables, an additive nonlinear dependence in the third, and linear dependencies in the fourth and fifth. The last five, $x_6 - x_{10}$, are pure noise variables independent of the response.

Table 1 displays the results of applying the MARS procedure to these data. Table 1a shows the history of the forward stepwise knot placement. The second column gives the GCV score (15) at each iteration M (first column). The third column shows the effective number of parameters in the fit $C(M)$ (15b). The fourth and fifth columns give the optimizing knot variable v^* and location t^*, while the last column points to the optimizing previous term (multivariate spline basis function) ℓ^* that multiplies the new univariate spline function. This term may in fact point to previous terms for its definition. The value $\ell^* = 0$ indicates that the previous multiplying term is B_0 (9b) so that a new purely additive term is being included in the model. The particular factors comprising the Mth multivariate spline basis function are identified by starting with the Mth row, then preceeding to its parent, then to its parent's parent and so on, until reaching a parent value of $\ell^* = 0$.

Table 1a shows that the first knot was placed on x_1. The second knot was placed on x_2, multiplying the first term. At this point ($M = 2$) the model consists of an additive contribution

on x_1 and an interaction between x_1 and x_2. The next three iterations include purely additive contributions form x_3, x_4, and x_5. The next iteration ($M = 6$) includes an additive term in x_2. This is multiplied by a factor involving x_1 on the subsequent iteration ($M = 7$), resulting in two bivariate splines characterizing the interaction between x_1 and x_2. Up to this point the GCV score has been monotonically decreasing.

The eighth iteration places into the model a term involving an interaction between variables x_9, x_2, and x_1. Note, however, that the GCV score has increased slightly. As more terms are added, the GCV score continues to increase until the present maximum number of terms $M_{\max} = 17$, is reached.

Table 1b shows the result of the backwards stepwise term deletion strategy. The first column gives the term number, m, the second its least squares coefficient, a_m (9a), followed by the knot variable, location, and parent as in Table 1a. A zero coefficient value, $a_m = 0$, means that the term has been deleted. Note that in addition to the deletion of all terms beyond $M = 7$, the purely additive contributions of variables x_1 and x_2 (first and sixth terms) have also been deleted. This leaves only the two terms (second and seventh) involving pure interactions between these two variables.

Table 1c summarizes the ANOVA decomposition of the final model. There are four ANOVA functions. The first three are additive functions on variables x_3, x_4, and x_5 respectively. The fourth ANOVA function is bivariate and represents a (pure) interaction between x_1 and x_2. Table 1c also gives the GCV score for the fit with the corresponding piecewise cubic basis (18). It is seen to be essentially the same as for the piecewise linear basis given in Table 1b.

The second column in Table 1c gives the standard deviation of the corresponding ANOVA function. This gives one indication of its (relative) importance to the model and is interpreted in a manner similar to a (standardized) regression coefficient in a linear model. The third column gives another indication of the importance of the corresponding ANOVA function, by providing the GCV score for the model with all of the terms corresponding to that particular ANOVA function deleted. This can be used to judge whether this ANOVA function is making an important contribution to the model, or whether it just slightly improves the global GCV score. In this example all four ANOVA functions appear to be important with the third one, involving x_5, being the weakest.

Figure 1a provides a pictorial representation of the ANOVA decomposition by plotting the respective (piecewise-cubic) ANOVA functions. The first three frames plot the respective additive functions involving x_3, x_4, and x_5. The fourth frame provides a perspective mesh plot of the bivariate ANOVA function involving x_1 and x_2. Figure 1b is an enlargement of the fourth frame of Figure 1a.

These figures show very nearly linear dependencies on x_3, x_4, and x_5, and a strong nonlinear interaction between x_1 and x_2. It is important to note that Figure 1b does not represent a smooth of the response y on variables x_1 and x_2, but rather it shows the contribution of x_1 and x_2 to a smooth

of y on variables $x_1, \cdots, x_{10}$. The accuracy of the resulting approximation is fairly remarkable considering the high dimensionality, $n = 10$, and the small sample size, $N = 100$. Note also that the procedure (correctly) did not enter $x_6, \cdots, x_{10}$ into the model.

The only shortcoming of the MARS model based on these data is that it did not capture the nonlinearity in the additive contribution of x_3 (20). Figure 1c shows the pictorial representation of the ANOVA decomposition corresponding to Figure 1a when the sample size is increased to $N = 200$. The model looks very similar to that for the smaller ($N = 100$) sample size (Figure 1a) except that it now gives a better approximation to the contribution of x_3.

Tables 1a – 1c and Figures 1a – 1b illustrate the application of the MARS procedure to a single data set (replication) from the particular setting under study (20). They do not give information on the average performance of the procedure when applied to this situation. Table 1d displays the results of a simulation study that addresses this issue. Each row summarizes the results of 100 replications of the following procedure. A sample of N ten-dimensional covariate vectors were randomly sampled from a uniform distribution in $[0, 1]^{10}$. A sample of N random standard normal deviates were then generated and the corresponding response values (20) were assigned to the covariate vectors. The MARS procedure was then applied. A new data set of 5000 observations was then generated and used to estimate the normalized integrated squared error

$$ISE = \int [f(\mathbf{x}) - \hat{f}(\mathbf{x})]^2 d^{10}x / \mathrm{Var}_{\mathbf{x}} f(\mathbf{x}), \tag{21a}$$

and the normalized predictive squared error

$$PSE = (ISE \cdot \mathrm{Var}_{\mathbf{x}} f(\mathbf{x}) + 1)/(\mathrm{Var}_{\mathbf{x}} f(\mathbf{x}) + 1) \tag{21b}$$

(fraction of unaccounted for variance) for the piecewise cubic MARS model.

The second column of Table 1d gives the optimizing GCV score averaged over the 100 replications, whereas the third and fourth columns give the corresponding average PSE and ISE (21) respectively. The quantities in parentheses are the associated standard deviations over the 100 replications. (The standard deviations of the averages are one tenth these values.)

Table 1d shows results for three sample sizes ($N = 50, 100, 200$) and for three sets of constraints applied to the MARS model. These constraints involve the maximum number of factors mi that are permitted to enter a single multivariate spline basis function. This controls the maximum interaction order permitted in the model. Setting $mi = 1$ restricts the model to be additive in the predictor variables, whereas $mi = 2$ limits the model to interactions involving at most two variables, and so on. The value $mi = n$ results in no restriction. Limiting the interaction level of the MARS model can improve accuracy (reduce variance) if the true underlying function f is close to an $\hat{f}$ that involves at most low order interactions. If not, such a limitation will introduce some bias in exchange for the corresponding variance reduction. In terms of interpretability there is a

strong advantage to models with $mi = 2$, owing to their graphical representation by means of the ANOVA decomposition.

In terms of ISE (21a) the accuracy of the MARS model for this problem is seen to increase rapidly as the sample size increases from 50 to 200. The additive model ($mi = 1$) is seen to be distinctly inferior to those involving interactions ($mi = 2, 10$) especially as the sample size increases. The optimizing GCV score is seen very slightly to overestimate the true PSE on average.

The true underlying function (20) in this case happens to involve at most interactions in two variables. Thus, setting $mi = 2$ results here in no increase in bias. Owing to the decrease in variance, the ISE is seen to be somewhat better than for the unrestricted MARS model ($mi = 10$). The size of the effect is seen, however, to be fairly small ($\leq 25\%$ in squared error loss) so that a large penalty is not incurred by fitting the full nonparametric model.

3.2. Alternating Current Series Circuit.

Figure 2a 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

$$V_{ab} = V_o \sin \omega t \tag{21a}$$

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

$$\omega = 2\pi f. \tag{21b}$$

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

$$I_{ab} = (V_o/Z) \sin(\omega t - \phi). \tag{21c}$$

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

$$Z(R, \omega, L, C) = [R^2 + (\omega L - 1/\omega C)^2]^{1/2}, \tag{22a}$$

$$\phi(R, \omega, L, C) = \tan^{-1}\left[\frac{\omega L - 1/\omega C}{R}\right]. \tag{22b}$$

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

$$\begin{aligned} x_1&: \ 0 \leq R \leq 100 \ \text{ohms} \\ x_2&: \ 20 \leq f \leq 280 \ \text{hertz} \\ x_3&: \ 0 \leq L \leq 1 \ \text{henries} \\ x_4&: \ 1 \leq C \leq 11 \ \text{micro farads.} \end{aligned} \tag{23}$$

Two hundred four-dimensional uniform covariate vectors were generated in the ranges (23). For each one, two responses were generated by adding normal noise to (22a) and (22b). The variance of the noise was chosen to give a 3 to 1 signal to noise ratio for both Z (22a) and ϕ (22b), thereby causing the true underlying function to account for 90% of the variance in both cases.

3.2.1. Impedance, Z.

Applying the MARS procedure to the impedance data with $mi = 1$ (additive model) gave an optimizing GCV score of 0.558. The GCV scores for $mi = 2$ and 4 were respectively 0.231 and 0.229. The additive model is seen (not surprisingly) to be inadequate. Perhaps more surprising is the fact that even though the true underlying function (22a) contains interactions to all orders, an approximation involving only two-variable interactions is seen to give nearly as good a fit to these data. Owing to its increased interpretability we show the results of the $mi = 2$ model.

Table 2a shows the ANOVA decomposition in the same format as Table 1c. There is a purely additive contribution from $x_1(R)$, additive contributions from $x_2(\omega)$ and $x_4(C)$, and interactions amongst x_2, $x_3(L)$, and x_4. Of the six ANOVA functions, all but the last one (involving an interaction between the capacitance C and the inductance L) seem important to the model. Figure 2b displays a graphical representation of the ANOVA decomposition. The first frame plots the (additive) contribution from the resistance R. The next three frames display the contributions of the remaining variables that participate in interactions. These perspective mesh plots show the total (additive plus interaction) contributions of each such variable pair. For example, the frame in the upper right corner plots the sum of the second and fourth ANOVA functions, whereas that of the lower left plots the sum of the second, third, and fifth.

The plots have been rotated so as to provide the best perspective view. The indicated zero marks the lowest value and the axis label marks the direction of higher values.

The dependence of the impedance Z on R (first frame) is estimated to be approximately linear. For low frequencies ω, Z is seen to be high and independent of L (upper right frame). For high ω, Z has a mild 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 values. The lower left frame shows that Z is very small and roughly independent of ω and C except when they jointly have very small values, in which case the impedance increases dramatically. The lower right frame of Figure 2b shows that the C, L joint contribution is nearly additive, consistent with the weak contribution of the sixth ANOVA function (Table 2a) to the MARS model.

These interpretations are based on visual examination of the graphic representation of the ANOVA decomposition of the MARS approximation, based on a sample of size $N = 200$. Since the data in this case are generated from known truth one can examine the generating equation (22a) to verify their general correctness.

Table 2b summarizes the results of a simulation study based on 100 replications of data randomly drawn according to the above prescription (22a), (23), in the same format as Table 1d. The

MARS procedure applied to the smallest sample size, $N = 100$, is seen to provide a fairly poor approximation on average in terms of ISE. The approximation accuracy improves substantially with the larger samples, except for additive modeling ($mi = 1$). The approximation accuracy for the constrained ($mi = 2$) models is (on average) nearly identical to the unconstrained ($mi = 4$) ones. It appears that the bias-variance trade-off is exactly off-setting in this case.

The average GCV score is seen to underestimate the corresponding PSE at the smallest sample size. This is due to the sharp joint dependence of Z on ω and C [see (22a) and Figure 2, third frame]. For small sample sizes most replications will fail to sample covariate vectors with very small joint values for ω and C, thereby failing to capture the rapid variation of Z in that region. There is no way that the GCV score (based on the ASR) can detect rapid function variation where there is no data. Note that sample reuse techniques such as cross-validation or bootstrapping have the same problem. As the sample size increases enough data is sampled in this region and the GCV score gives a more accurate estimate of the true PSE (on average).

3.22. Phase Angle, ϕ.

The MARS procedure applied to the phase angle data (22b) (23) with $mi = 1, 2$, and 4 gave optimizing GCV scores of 0.295, 0.219, and 0.203, respectively. Here the additive model, while still being less accurate, is more competitive with those involving interactions. The two variable interaction model again fits the data almost as well as the unconstrained model.

Table 3a summarizes the ANOVA decomposition for the $mi = 2$ MARS model. It involves additive contributions from all but $x_3(L)$ and interactions among all variable pairs except C and L. Two of the ANOVA functions (fifth and seventh) however are seen to make very weak contributions to the final model. Figure 2c is a graphical representation of the ANOVA decomposition in the same format as Figure 2b. 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 2b). The principal interaction effect is to decrease the phase angle for simultaneously high values of the predictor variable pairs.

Table 3b gives the results of 100 replications of phase angle data generated according to (22b), (23). At the smallest sample size ($N = 100$) the additive model produces fits that (on average) are nearly as accurate as those involving interactions. For the larger samples the interaction models are somewhat more accurate in terms of ISR. The average optimizing GCV score is seen to be quite close to the true average PSE.

3.3. Additive Data.

In the preceding examples there were strong interaction effects and it was seen that allowing such effects in the MARS model substantially improved approximation accuracy. This example, taken from Friedman and Silverman (1987), examines what happens when the true underlying function is exactly additive and interactions are allowed to enter the MARS model. One would expect accuracy to deteriorate since allowing for interactions among the variables increases the

variance of $\hat{f}$ while, in this particular case, not decreasing the bias.

Table 4 summarizes (in the same format as Tables 1d, 2b, 3b) the results of 100 replications of the following simulation experiment. $N(= 50, 100, 200)$ 10-dimensional covariate vectors were generated in the unit hypercube. A set of standard normal deviates ϵ_i were then generated and response values were assigned according to

$$\begin{aligned} y_i = {} & 0.1e^{4x_{1i}} + 4/[1 + e^{-20(x_{2i}-1/2)}] \\ & + 3x_{3i} + 2x_{4i} + x_{5i} + 0 \cdot x_{6i} + 0 \cdot x_{7i} \\ & + 0 \cdot x_{8i} + 0 \cdot x_{9i} + 0 \cdot x_{10,i} + \epsilon_i. \end{aligned}$$

Here the signal to noise ratio is 0.28 so that the true underlying function accounts for 92% of the variance of the response.

The ratio of the average ISE values for the additive and $mi = 2$ interaction fits are seen (Table 4) to be about 0.67 at all sample sizes. The corresponding ratio for the $mi = 10$ unconstrained fit is about 0.60. The corresponding square roots of the ratios are 0.81 and 0.77. Thus, the (average) accuracy here is reduced by about 25% when the interactive models are fit to purely additive data. This degradation is surprisingly small given the small sample sizes and the high dimensionality ($n = 10$). Note that the average GCV scores for the interactive models are always slightly worse than that for the corresponding additive fit, so that the interactive models are not (on average) claiming to do better than the additive ones. This suggests a strategy of accepting the additive model if those involving interactions fit no better in terms of the GCV score, especially owing to the increased interpretability of the additive model.

4.0. Remarks.

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

4.1. 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 3, namely limiting the maximum interaction order of the model. 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 (15b) for knots placed under these restrictions should be less than that for the unrestricted knot optimization. (The implementing

software charges $0.8 \cdot$ df and $0.4 \cdot$ df, respectively, for the additive and linear constraints where df 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, the parameters of the corresponding potential new multivariate spline basis function (v, t, ℓ, and B_ℓ) (10) can be examined for consistency with the constraints. If it is inconsistent, it can simply be marked ineligible for inclusion in the model.

4.2. 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 $g(x_1, \cdots, x_n)$ that is thought to capture some aspects of the true underlying function $f(x_1, \cdots, x_n)$. More generally, one may have a set of such functions $\{g_j(x_1, \cdots, x_n)\}_1^J$, each one of which might capture some aspect of the functional relationship. A semiparametric model of the form

$$\hat{f}_{sp}(x_1, \cdots, x_n) = \sum_{j=1}^{J} c_j g_j(x_1, \cdots, x_n) + \hat{f}(x_1, \cdots, x_n), \tag{24}$$

where $\hat{f}(x_1, \cdots, x_n)$ takes the form of the MARS approximation (9), could then be fit to the data. The coefficients c_j in (24) are jointly fit along with the parameters of the MARS model. 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 (24) are easily fit using the MARS strategy. One simply includes $\{g_j(x_1, \cdots, x_n)\}_1^J$ 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 (24).

4.3. Collinearity.

Extreme collinearity of the predictor variables is a fundamental problem in the modeling of observational data. Solely in term of predictive modeling it represents an advantage in that it effectively reduces the dimensionality of the predictor variable space. This is 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 difficult to separate additive and interactive contributions among them. A highly nonlinear dependence on one such 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.

4.4. 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. Unlike linear regression, however, it is not very sensitive to outliers in the predictor variable space, owing to the local nature of the resulting fit; sample covariate vectors far from an evaluation point tend to have less rather than more influence on the model estimate. Response outliers will tend to strongly effect model estimates only close to their corresponding covariate values. They will also (slightly) increase the variance of model estimates elsewhere by increasing the number of multivariate spline basis functions (required to capture the apparent high curvature of the function near each outlier).

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 by one that minimizes another loss criterion (given the current set of multivariate spline basis functions). Using robust/resistant 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 could be developed for other types of linear fitting. For those with access to rich computing environments, this presents no problem. For others, a compromise strategy can mitigate the robustness problem for isolated outliers. The multivariate spline basis functions are selected using the standard MARS approach with least-squares fitting. Given this basis, the expansion coefficients $\{a_m\}_0^M$ (9) are then fit using a robust/resistant linear regression method to form the final model. This reduces the influence of the response outliers on model predictions close to their corresponding covariate vectors. It does not remove the (small) increased variance associated with the additional (now redundant) basis functions.

4.5. Logistic Regression.

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

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

where p is the probability that y assumes its larger value. The coefficients $\{\beta_i\}_1^n$ are estimated by (numerically) maximizing the likelihood of the data. Recently, 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 by

$$\log[p/(1-p)] = \hat{f}(x_1, \cdots, x_p)$$

with $\hat{f}(x_1, \cdots, x_p)$ taking the form of the MARS approximation (9). This is implemented in the MARS algorithm by simply replacing the internal linear least-squares routine 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 analogous to that described in Section 4.4, however, is likely to provide a good approximation; the multivariate spline basis functions are selected using the 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. Note that in this setting the least-squares criterion is *more* robust than the likelihood based criterion.

4.6. Reflection Invariance.

The MARS procedure as described here is not necessarily invariant to reflections of the individual predictor variables. Replacing x_i by $-x_i$ can (slightly) change the MARS model. This is due to the fact that the pure linear term, associated with the piecewise-linear basis on each variable, is not automatically included in the model; but rather it is subjected to the same forward/backward stepwise selection strategy as all other potential basis functions. This gives the procedure the ability to model certain types of dependencies with fewer basis functions than would otherwise be the case. Also, certain kinds of interaction effects require less terms to model than others.

In order to get an idea of the size of this effect a further simulation study was performed on the alternating current series circuit example (Section 3). Fifteen additional simulation studies ($N = 200$, 100 replications each) were done analogous to those that led to Tables 2b and 3b. For each of the (total) 16 studies, the predictor variables were each multiplied by one of the 16 combinations of $(\pm 1, \pm 1, \pm 1, \pm 1)$. The variance of the ISE over these 16 experiments was compared to its average variance over the 100 replications of different training sample sets. For the impedance, this ratio was 0.156 whereas for the phase angle it was 0.036. The higher value for the impedance is due to the very sharp structure for very low joint values of ω and C (Figure 2, lower left frame). In both cases, however, the variability in modeling accuracy due to reflections of the predictor variables is seen to be very small compared to the variability associated with the random nature of the training data.

Several modifications of the MARS procedure that render it invariant under variable reflection are currently under study. It remains to be seen whether they can provide approximations that are as accurate as the method described here.

4.7. Low Dimensional Modeling.

The main advantage of MARS modeling over existing methodology is clearly realized in high dimensional settings. It can, however, be competitive in low dimensions ($n \leq 2$) as well. Friedman and Silverman (1987) studied its properties for the smoothing problem ($n = 1$) and showed that it can produce superior performance, especially in situations involving small samples and low signal to noise. These properties should extend to surface modeling ($n = 2$) as well, although detailed studies have not yet been performed. Friedman and Silverman (1987) also studied this approach in the special case of additive modeling ($mi = 1$). The method was shown to be competitive with existing methodology in this application, again exhibiting superior performance in situations with small samples and low signal to noise.

5.0. Conclusion.

The examples and simulation studies indicate that the MARS approach has the potential to become a useful tool for data modeling. It possesses to some degree the the desirable properties of the recursive partitioning approach; these are its adaptability, automatic variable subset selection, and ability to exploit low "local" dimensionality. Moreover, it is able to overcome some of recursive partitioning's limitations; it produces continuous approximations with continuous derivatives (if desired); it has additional adaptabilty to exploit functions with weak high order interactions, thereby providing better approximations to functions that are nearly linear or additive; and it has increased interpretability through its ANOVA decomposition that breaks up the approximation into its additive and various interaction components.

It is important to note that this is a new methodology for which there is, at present, very little collective experience. Its results should be interpreted with some caution until their reliability is tested over time in a wide variety of settings. No doubt as such experience is gained useful and important modifications to this basic approach will become apparent.

A FORTRAN program implementing the MARS methodology described in ths report is available from the author.

Bibliography

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

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.

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

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.

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

Donoho, D. L. and Johnstone, I. (1985). Projection-based smoothing, and a duality with kernel methods. Department of Statistics, Stanford University, Technical Report No. 238.

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. (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., 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. (1987). Flexible parsimonious smoothing and additive modeling. Stanford Linear Accelerator, Stanford, CA report SLAC-PUB-4390.

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.

Hastie, T. and Tibshirani, R. (1985). Discussion of P. Huber: Projection pursuit, *Ann. Statist.* **13**, 502–508.

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.

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

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

Table 1a

History of the MARS forward stepwise knot placement strategy for Example 3.1.

iter.	gcv	# efprms	variable	knot	parent
1	0.8460	4.5	1.	0.5257	0.
2	0.5781	8.0	2.	-0.6736	1.
3	0.3914	11.5	3.	-1.626	0.
4	0.2885	15.0	4.	-1.170	0.
5	0.2347	18.5	5.	-1.601	0.
6	0.1911	22.0	2.	-1.177	0.
7	0.1599	25.5	1.	-1.164	6.
8	0.1603	29.0	9.	-1.128	2.
9	0.1621	32.5	3.	-0.9315	0.
10	0.1696	36.0	4.	1.015	1.
11	0.1802	39.5	3.	1.013	0.
12	0.1829	43.0	6.	-0.2161	11.
13	0.1936	46.5	4.	-1.675	5.
14	0.2062	50.0	4.	0.2366e-01	11.
15	0.2271	53.5	9.	1.583	3.
16	0.2519	57.0	9.	-0.2349	5.
17	0.2837	60.5	2.	-0.4146	5.

Table 1b

The result of the backwards stepwise term deletion strategy for Example 3.1.

$gcv = 0.1404$ $\quad \#efprms = 18.5$

term	coeff.	variable	knot	parent
1	0.	1.	0.5257	0.
2	0.8746	2.	-0.6736	1.
3	0.4525	3.	-1.626	0.
4	0.3171	4.	-1.170	0.
5	0.2232	5.	-1.601	0.
6	0.	2.	-1.177	0.
7	0.2373	1.	-1.164	6.
8	0.	9.	-1.128	2.
9	0.	3.	-0.9315	0.
10	0.	4.	1.015	1.
11	0.	3.	1.013	0.
12	0.	6.	-0.2161	11.

Table 1c

ANOVA decomposition summary of the MARS model for Example 3.1.

fun.	std. dev.	-gcv	# terms	# efprms	variable(s)	
1	0.4518	0.4109	1	3.5	3	
2	0.2983	0.2520	1	3.5	4	
3	0.2229	0.1974	1	3.5	5	
4	0.7772	0.8867	2	7.0	1	2

piecewise cubic fit on 5 terms, gcv = 0.1457

Table 1d

Summary of 100 replications of Example 3.1, piecewise cubic fit.

mi	$\overline{GCV}$	$\overline{PSE}$	$\overline{ISE}$
$N = 50$:			
1	.46 (.12)	.45 (.097)	.40 (.11)
2	.28 (.13)	.28 (.18)	.22 (.20)
10	.27 (.11)	.30 (.19)	.24 (.21)
$N = 100$:			
1	.36 (.072)	.36 (.064)	.30 (.070)
2	.15 (.043)	.14 (.026)	.059 (.029)
10	.15 (.047)	.16 (.041)	.077 (.044)
$N = 200$:			
1	.32 (.037)	.31 (.022)	.25 (.023)
2	.12 (.029)	.12 (.015)	.033 (.015)
10	.12 (.029)	.12 (.024)	.041 (.025)

Table 2a

ANOVA decomposition summary of the MARS model
on alternating current series circuit impedence, Z.

$gcv = 0.2311$ $\quad \#efprms = 46.5$

fun.	std. dev.	-gcv	# terms	# efprms	variable(s)	
1	0.5096	0.6392	1	3.5	1	
2	1.833	0.6854	3	10.5	2	
3	1.417	0.6431	3	10.5	4	
4	0.4195	0.4401	1	3.5	2	3
5	2.034	0.5704	4	14.0	2	4
6	0.1702	0.2577	1	3.5	3	4

piecewise cubic fit on 13 terms, gcv =0.2447

Table 2b

Summary of 100 replications of the alternating current series circuit impedance, Z, piecewise cubic fit.

mi	$\overline{GCV}$	$\overline{PSE}$	$\overline{ISE}$
$N = 100$:			
1	.65 (.12)	.71 (.092)	.68 (.10)
2	.46 (.15)	.52 (.19)	.46 (.21)
4	.45 (.15)	.52 (.19)	.47 (.21)
$N = 200$:			
1	.60 (.082)	.62 (.050)	.58 (.056)
2	.27 (.064)	.27 (.10)	.20 (.11)
4	.28 (.066)	.28 (.091)	.20 (.11)
$N = 400$:			
1	.57 (.049)	.57 (.026)	.52 (.029)
2	.20 (.057)	.18 (.050)	.095 (.056)
4	.20 (.035)	.18 (.035)	.092 (.038)

Table 3a

ANOVA decomposition of the MARS model
on the alternating current series circuit phase angle, ϕ.

$gcv = 0.2190 \qquad \#efprms = 39.5$

fun.	std. dev.	-gcv	# terms	# efprms	variable(s)	
1	0.6323	0.3257	1	3.5	2	
2	0.7253	0.4180	2	7.0	4	
3	0.9931	0.3041	1	3.5	1	
4	0.6483	0.4015	2	7.0	2	3
5	0.1521	0.2254	1	3.5	2	4
6	0.7754	0.2662	2	7.0	1	4
7	0.2064	0.2248	1	3.5	1	3
8	0.3464	0.2458	1	3.5	1	2

piecewise cubic fit on 11 terms, gcv = 0.2393

Table 3b

Summary of 100 replications of the alternating current series circuit phase angle, ϕ, piecewise cubic fit.

mi	$\overline{GCV}$	$\overline{PSE}$	$\overline{ISE}$
$N = 100$:			
1	.36 (.057)	.35 (.036)	.27 (.040)
2	.33 (.059)	.32 (.047)	.25 (.052)
4	.32 (.059)	.33 (.12)	.26 (.14)
$N = 200$:			
1	.32 (.032)	.31 (.016)	.23 (.017)
2	.25 (.033)	.24 (.022)	.15 (.025)
4	.24 (.032)	.24 (.022)	.15 (.070)
$N = 400$:			
1	.30 (.020)	.29 (.007)	.21 (.008)
2	.22 (.019)	.20 (.011)	.11 (.012)
4	.21 (.019)	.19 (.012)	.10 (.013)

Table 4

Summary of 100 replications of applying MARS to purely additive data, Example 3.3.

mi	$\overline{GCV}$	$\overline{PSE}$	$\overline{ISE}$
$N = 50$:			
1	.30 (.092)	.25 (.053)	.13 (.062)
2	.34 (.077)	.30 (.074)	.19 (.085)
10	.34 (.077)	.29 (.080)	.19 (.092)
$N = 100$:			
1	.22 (.035)	.18 (.020)	.053 (.024)
2	.22 (.040)	.21 (.035)	.081 (.041)
10	.24 (.041)	.21 (.035)	.088 (.042)
$N = 200$:			
1	.17 (.022)	.16 (.008)	.024 (.009)
2	.18 (.024)	.17 (.014)	.036 (.016)
10	.19 (.025)	.17 (.012)	.040 (.015)

Figure Captions

Figure 1a: Graphical representation of the ANOVA decomposition of the piecewise cubic MARS model for Example 3.1.

Figure 1b: Enlargement of the fourth frame of Figure 1a; interaction contribution of (x_1, x_2) to the MARS model for Example 3.1.

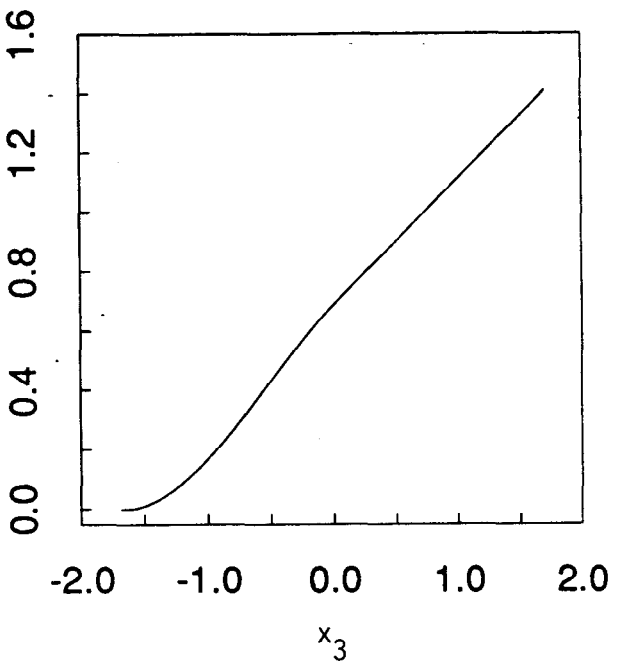
Figure 1c: Graphical ANOVA decompositon of the MARS model for Example 3.1, with 200 observations.

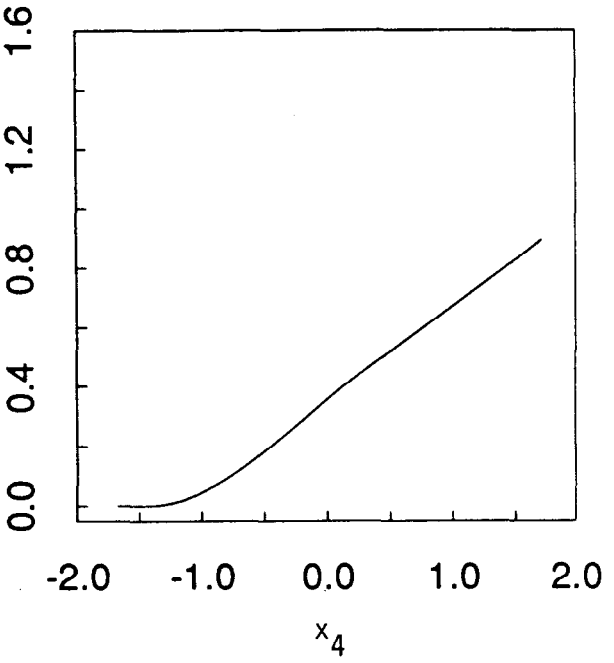
Figure 2a: Schematic diagram of the alternating current series circuit of Example 3.2.

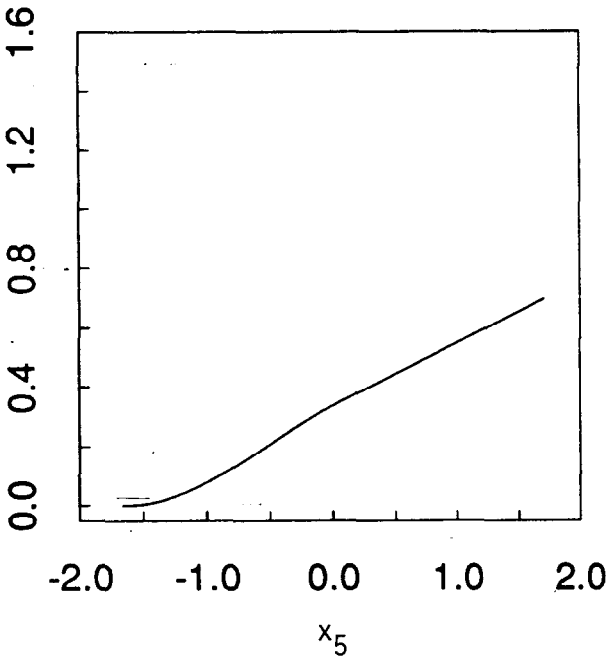
Figure 2b: Graphical ANOVA decomposition for the alternating current series circuit impedance, Z, Example 3.21.

Figure 2c: Graphical ANOVA decomposition for the alternating current series circuit phase angle, ϕ, Example 3.22.

Figure 1a


0.0
0.4
0.8
1.2
1.6
-2.0
-1.0
0.0
1.0
2.0
x_3


0.0
0.4
0.8
1.2
1.6
-2.0
-1.0
0.0
1.0
2.0
x_4


0.0
0.4
0.8
1.2
1.6
-2.0
-1.0
0.0
1.0
2.0
x_5

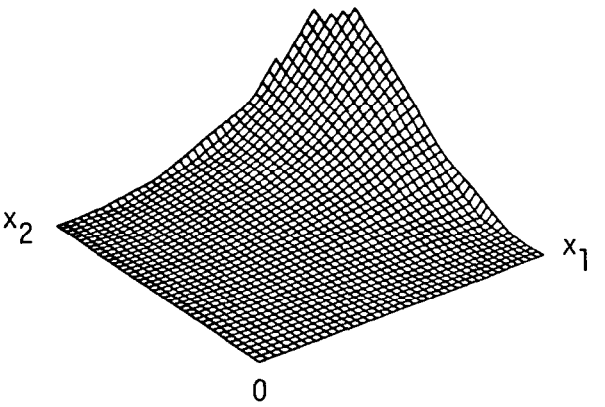

x_2
x_1
0

Figure 1b

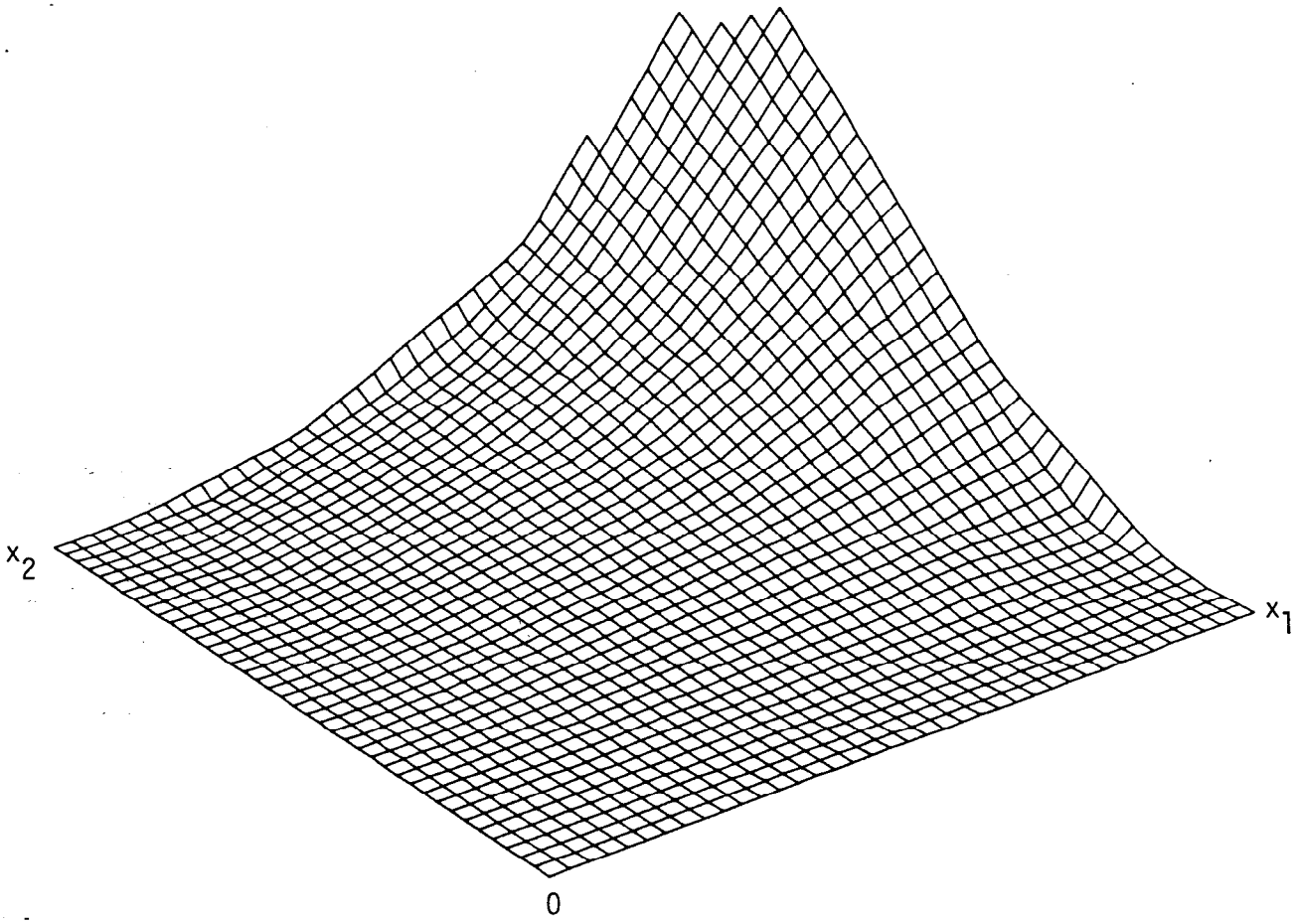
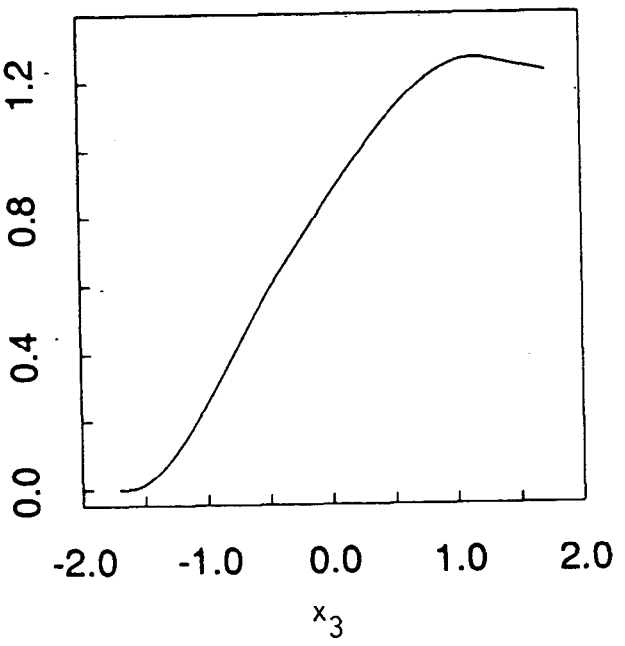
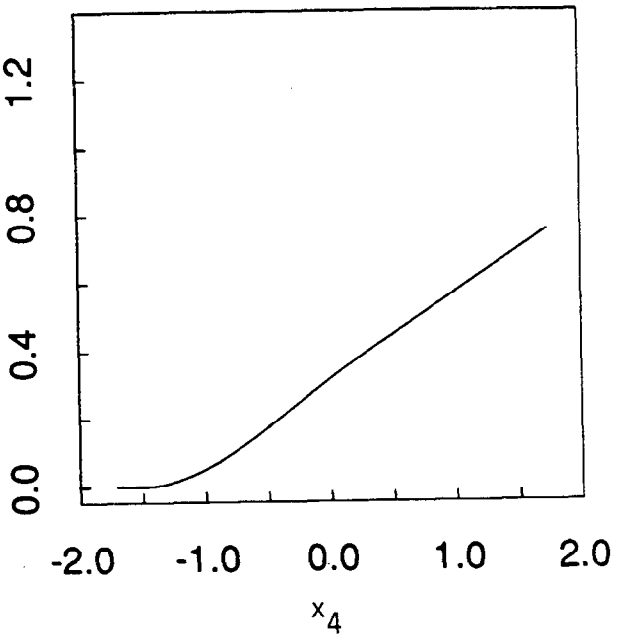
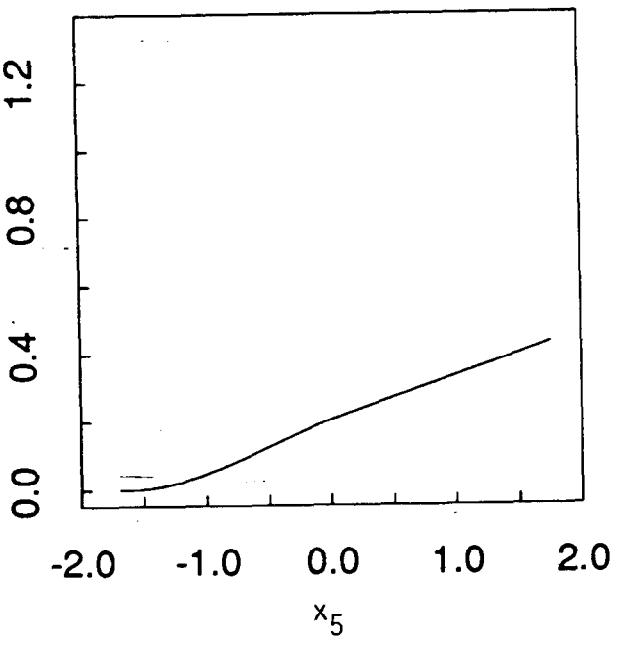
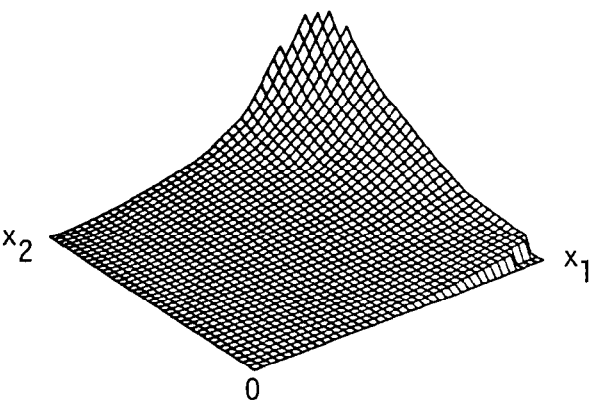


Figure 1c


0.0
0.4
0.8
1.2
-2.0
-1.0
0.0
1.0
2.0
x_3


0.0
0.4
0.8
1.2
-2.0
-1.0
0.0
1.0
2.0
x_4


0.0
0.4
0.8
1.2
-2.0
-1.0
0.0
1.0
2.0
x_5


x_2
x_1
0

- Figure 2a

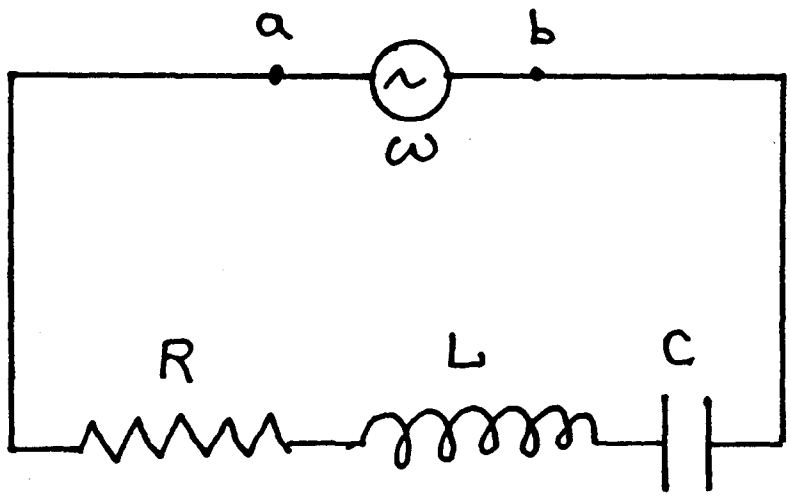
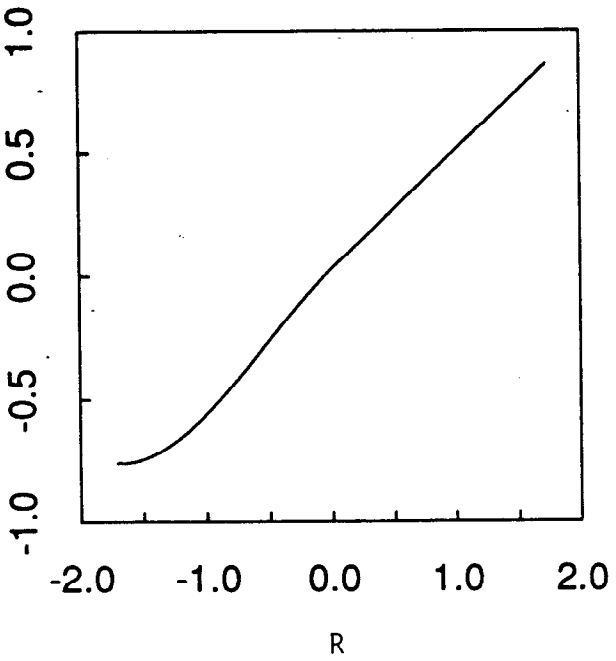
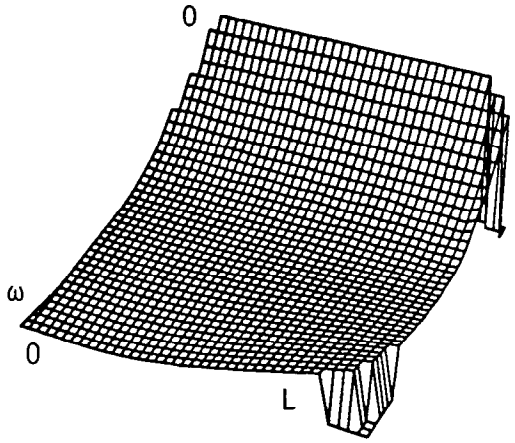
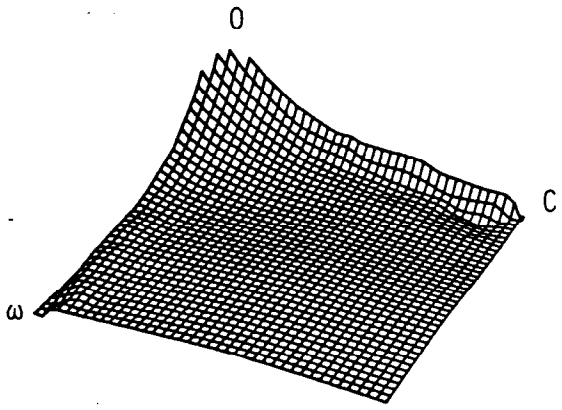


Figure 2b







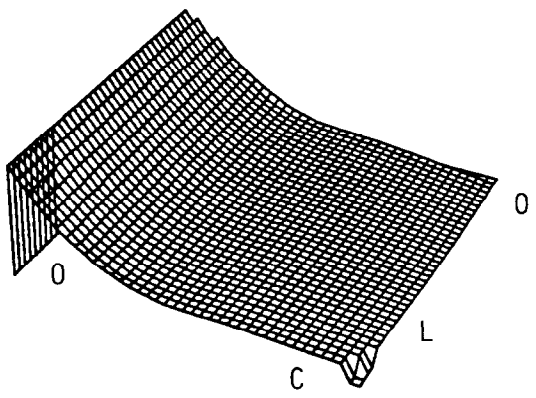


Figure 2c