An introduction to multivariate adaptive regression splines

Jerome H Friedman and Charles B Roosen

Stat Methods Med Res 1995 4: 197
DOI: 10.1177/096228029500400303

Statistical Methods in Medical Research 1995; **4**: 197–217

An introduction to multivariate adaptive regression splines

Jerome H Friedman and **Charles B Roosen** Department of Statistics and Stanford Linear Accelerator Center, Stanford University, Stanford, California, USA

Multivariate Adaptive Regression Splines (MARS) is a method for flexible modelling 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 recursive partitioning (e.g. CART) and shares its ability to capture high order interactions. However, it has more power and flexibility to model relationships that are nearly additive or involve interactions in at most a few variables, and produces continuous models with continuous derivatives. In addition, the model can be represented in a form that separately identifies the additive contributions and those associated with different multivariable interactions.

This paper summarizes the basic MARS algorithm, as well as extensions for binary response, categorical predictors, nested variables and missing values. It presents tips on interpreting the output of the standard FORTRAN implementation of MARS, and provides an example of MARS applied to a set of clinical data.

1 Introduction

A problem common to many disciplines is that of adequately estimating a response which is a function of many predictors knowing only the value of the response (often perturbed by noise) at various sets of predictor values. For example, researchers may be interested in predicting blood cholesterol level given covariates such as age, gender, weight, blood pressure, other blood chemistry values and perhaps prior cholesterol levels (Garber *et al.*).[1]

The goal is to model the dependence of a response variable y on one or more predictor variables $x_1,. . .x^n$ given observed realizations $\{y_i, x_{1i},. . .,x_{ni}\}_{i=1}^{N}$. The system that generated the data is presumed to be described by

$$y = f(x_1, \ldots, x_n) + \epsilon \tag{1.1}$$

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

Address for correspondence: Jerome H Friedman, Department of Statistics, Stanford University, Sequoia Hall, Stanford CA 94305-4065, USA.

$$E = \frac{1}{N}\sum_{i=1}^{N}[f(\mathbf{x}_i) - \hat{f}(\mathbf{x}_i)]^2. \tag{1.2}$$

In least-squares fitting we take the observed y_i as a proxy for $f(\mathbf{x}_i)$ and select $\hat{f}(\mathbf{x})$ to minimize the above criterion subject to some restrictions on the functional form of $\hat{f}(\mathbf{x})$. Simple examples are linear regression, in which we take $\hat{f}(\mathbf{x}) = \alpha_0 + \alpha^T\mathbf{x}$ to be a linear combination of the predictor variables, and additive modelling, in which we take $\hat{f}(\mathbf{x}) = \alpha_0 + \hat{f}_1(x_1) + \ldots + \hat{f}_n(x_n)$ to be a sum of univariate functions from a certain class of functions.

A difficult problem in modelling is selecting the appropriate functional class for $\hat{f}(\mathbf{x})$. Particularly when the model is to be used for interpretation, it is desirable to have an $\hat{f}(\mathbf{x})$ which represents the true $f(\mathbf{x})$ as accurately as possible. While the model class should be general enough to include the true function (or at least closely approximate it), care must be taken to avoid overfitting in the presence of noise. Multivariate adaptive regression splines (MARS) uses an expansion in product spline basis functions where the basis functions are carefully chosen so as accurately to approximate $f(\mathbf{x})$ while avoiding overfitting.

The MARS technique may be viewed as either a generalization of the recursive partitioning regression strategy[2,3] such as in CART, or as a generalization of additive modelling.[4,5] Unlike recursive partitioning, this method produces continuous models with continuous derivatives. Additive models are a subset of the models implementable by MARS, but MARS also allows interactions up to an order specified by the investigator, and trades off the interaction order and complexity of the additive functions and interactions, using a single global criterion. MARS also provides a natural approach to modelling categorical variables, nested variables and missing values.

The ordinal and logistic versions of MARS are described in great detail in Friedman,[6] while the extension to categorical variables, nested variables and missing values is presented in Friedman.[7] This paper provides a simplified description of the basic ideas used in MARS along with practical information on using the current implementation of MARS and interpreting the resulting fitted model.

2 Multivariate Adaptive Regression Splines

This section gives a brief overview of the MARS procedure described much more completely in Friedman,[6] as well as the categorical variable, nested variable, and missing value techniques described in Friedman.[7] The MARS procedure is in turn based on a generalization of spline methods for function fitting. Splines have been extensively studied and have many desirable properties. [See refs 8–12.]

2.1 Basic ideas

2.1.1 Regression splines

First consider the case of only one predictor variable, x ($n = 1$). We wish to estimate the function $f(x)$ relating the response y to x. An approximating (qth order regression) spline function $\hat{f}_q(x)$ is obtained by dividing the range of x values into $K + 1$ disjoint regions separated by K points (called 'knots'). The approximation takes the form of a separate qth degree polyromial in each region, constrained so that the function and its

$q - 1$ lowest order derivatives are everywhere continuous. These continuity constraints and the use of polynomials within each region yields a smooth fitted function. Generally the order of the spline is taken to be low ($q \leq 3$). Each qth degree polynomial is defined by $q + 1$ parameters so that there are a total of $(K + 1)(q + 1)$ parameters to be adjusted to best fit the data, usually by least squares. The continuity requirement however places q constraints at each knot location making a total of K_q constraints. The total number of free parameters is thus $K + q + 1$.

Regression spline fitting can be implemented by directly solving the constrained minimization problem described above. Usually, however, the problem is converted to an unconstrained optimization problem by choosing a set of basis functions $\{B_k^{(q)}(x)\}_0^{K+q}$ that span the space of all qth order spline functions (given the chosen knot locations) and performing a (linear) least-squares fit of the response on this basis function set. In this case the approximation takes the form

$$\hat{f}_q(x) = \sum_{k=0}^{K+q} a_k B_k^{(q)}(x) \tag{2.1}$$

where the values of the expansion coefficients $\{a_k\}_0^{K+q}$ are unconstrained, and the continuity constraints are intrinsically embodied in the basis functions $\{B_k^{(q)}(x)\}_0^{K+q}$. One such basis ('truncated power basis') is comprised of the functions

$$\{x^j\}_{j=0}^q, \ \{(x - t_k)_+^q\}_1^K. \tag{2.2}$$

Here $\{t_k\}_1^K$ are the knot locations defining the $K + 1$ regions and the truncated power functions are defined by

$$(x - t_k)_+^q \begin{cases} 0 & x \leqslant t_k \\ (x - t_k)^q & x > t_k. \end{cases} \tag{2.3}$$

The truncated power basis (2.2) is not the only basis appropriate for this application. Any set of $K + q + 1$ linearly independent linear combinations of these basis functions (2.2) will also span the same space. The most popular basis is the (minimum support) 'B-spline' basis owing to its superior numerical properties when used in conjunction with least-squares fitting. B-spline basis functions have support over, and are defined by, $K + 2$ adjacent knot locations, whereas the truncated power functions have maximal support but are each defined by a single knot location. This latter property has important algorithmic consequences for adaptive regression spline strategies (see below).

Regression splines (of order q) are characterized by the number of knots K and their locations $\{t_k\}_1^K$. This provides the user with a great deal of flexibility in specifying the nature of the approximating function. This is in contrast to other techniques such as kernel methods[13] and smoothing splines[14] which are characterized by a single (smoothing) parameter. If the user has a good deal of knowledge about the nature of the true underlying function $f(x)$ (1.1) and sufficient intuition concerning the effect on the approximation of changes in the knot specification, this increased flexibility can be used to great advantage. On the other hand, lack of such knowledge can make choosing a good set of knots difficult.

The variance of the function estimate $\hat{f}(x)$ in any local region is proportional to the ratio of the local knot density to the local data (predictor variable) density. The bias is proportional to the local second derivative of the true underlying function $f''(x)$

divided by the local knot density. For any given $f(x)$ (1.1) and distribution of (abscissa) data points there is an optimal specification for the knots. This is however usually unknown. Standard defaults often involve placing the knots equispaced along the abscissa or at the $1/K(\times 100)$ percentiles of the data abscissa values. The regression spline approximation using these (restricted) defaults is then also characterized by a single parameter (number of knots K) as are kernel and smoothing-spline methods.

2.1.2 Adaptive knot selection

The flexibility of the regression spline approach can be enhanced by incorporating an automatic strategy for knot selection as part of the data fitting process. Many such strategies have been proposed, most of them involving a numerical minimization of the least-squares criterion

$$\sum_{i=1}^{N}\left[y_i - \sum_{k=0}^{K+q} a_k B_k^{(q)}(x)\right]^2 \tag{2.4}$$

jointly with respect to the expansion coefficients $\{a_k\}_0^{K+q}$ and the knot locations $\{t_k\}_1^K$. Although sometimes effective, these approaches have many difficulties and can be computationally expensive. [See ref. 11 and references therein.]

An especially simple and effective strategy for automatically selecting both the number and locations for the knots was described by Smith.[15] She suggested using the truncated power basis (2.2) so that (2.4) becomes

$$\sum_{i=1}^{N}\left[y_i - \sum_{j=0}^{q} b_j x^j - \sum_{k=1}^{K} a_k (x - t_k)_+^q\right]^2. \tag{2.5}$$

Here the coefficients $\{b_j\}_0^q$, $\{a_k\}_1^K$ can be regarded as the parameters associated with a multiple linear (least-squares) regression of the response y on the 'variables' $\{x^j\}_0^q$ and $\{(x - t_k)_+^q\}_1^K$. Adding or deleting a knot t_k is viewed as adding or deleting the (corresponding) variable $(x - t_k)_+^q$. Smith's strategy consists of starting with a very large number of eligible knot locations $\{t_1, \ldots, t_{K^{\max}}\}$ (say one at every interior data point, $K_{\max} = N - 2$) and considering the corresponding 'variables' $\{(x - t_k)_+^q\}_1^{K^{\max}}$ as candidates to be selected through a statistical variable subset selection procedure (Smith suggested a standard forward/backward stepwise approach).

Although quite simple, this approach to knot selection is both elegant and powerful. It automatically selects both the number of knots K and their locations $t_1, \ldots, t_K$. It thereby not only estimates the overall (global) amount of smoothing to be applied (controlled by K), but in addition it uses the data to estimate the separate relative amount of smoothing to be applied at different (abscissa) locations. In a large simulation study comparing many different smoothers over a wide variety of situations,[16] this method proved to be the best or among the best over the situations (true underlying function, abscissa design) considered. This approach has the additional virtue of being very simple to implement and fast to compute.

An attractive property of regression splines with adaptive knot selection is that observations which are outliers in the response affect the fit locally rather than globally. If the observed data had an outlier or cluster of outlying points, the adaptive knot selection procedure is likely to put a knot on either side of the point or cluster. This would lead to a local spike at the location of the outlier, but would not affect the fit in other regions. The user can then decide whether the fitted spike is a true feature of the signal, or reflects an outlier or outliers which should be removed. If the latter is the case, the model may be refit excluding the point or points in question.

The adaptive regression spline strategy introduced by Smith[15] was developed for the univariate ($n = 1$) smoothing problem. The real potential of this idea however is realized in the multivariate setting ($n \gg 1$) where the function to be estimated can depend on many (measured) variables. The multivariate adaptive regression spline method[6] can be viewed as a multivariate generalization of Smith's strategy.[15]

2.1.3 *Tensor product splines*

An approximating (qth order regression) spline function $\hat{f}_q(\mathbf{x})$ of n variables ($\mathbf{x} = \{x_1, \ldots, x_n\}$) is defined analogously to that for one variable. The n-dimensional space R^n is divided into a set of disjoint regions and within each one $\hat{f}_q(\mathbf{x})$ is taken to be a polynomial in n variables with the maximum degree of any single variable being q. The approximation $\hat{f}_q(\mathbf{x})$ is constrained so that it and all its derivatives to order $q - 1$ are everywhere continuous. This places constraints on the approximating polynomials in the separate regions along the ($n - 1$-dimensional) region boundaries. As in the univariate case, the approximation is most easily constructed by choosing a basis function set (of n-variables) that spans the space of all qth order n-dimensional spline functions given the particular set of chosen regions. The approximation is then obtained by fitting the coefficients of this expansion to the data.

For $n > 2$ (and usually for $n = 2$) the disjoint regions defining the spline approximation are taken to be tensor products of disjoint intervals on each of the variables, delineated by knot locations. Thus, placing K_j knots on each of the variables ($1 \leq j \leq n$) produces $\prod_{j=1}^{n}(K_j + 1)$ regions. A basis function set that spans the space of spline functions over this set of regions is the tensor product of the corresponding univariate spline bases associated with the knot locations on each of the variables

$$\hat{f}_q(\mathbf{x}) = \sum_{k_1=0}^{K_1+q} \cdots \sum_{k_n=0}^{K_n+q} a_{k_1, \ldots, k_n} \prod_{j=1}^{n} B_{k_j}^{(q)}(x_j). \tag{2.6}$$

Here $\{B_{k_j}^{(q)}(x_j)\}_{k_j=0}^{K_j}$ is the basis function set for a qth order spline approximation given the locations of the K_j knots on x_j ($1 \leq j \leq n$). The size of this tensor product basis (2.6) and thus the number of coefficients to be estimated in a (linear least-squares) fit to the data is

$$\prod_{j=1}^{n} (K_j + q + 1). \tag{2.7}$$

For cubic splines ($q = 3$) with $K_j = 5$ knots (only) on each variable there are 59 049 coefficients to be estimated in five dimensions. In six dimensions ($n = 6$) that number is 531 441, while for $n = 10$ it is 3.5×10^9. This exponential increase in both estimation and computational complexity with increasing dimension (for the same level of refinement) is a reflection of the 'curse-of-dimensionality'.[17] Gargantuan training samples are required for straightforward tensor product spline approximations in high dimensions.

The multivariate adaptive regression spline (MARS) strategy employs the tensor product representation (2.6) with the truncated power basis (2.2), and considers a very large number [$K_j \lesssim O(N)$] of eligible knot locations on each variable. In analogy with the Smith strategy,[15] each of the $\prod_{j=1}^{n}(K_j + q + 1)$ basis functions so derived is taken to be a candidate 'variable' to be potentially selected through a statistical variable subset selection procedure. A small subset of these basis functions are then selected for inclusion in the fitted model.

2.1.4 Basis function selection

As in the univariate ($n = 1$) case, this multivariate adaptive spline strategy can be motivated from geometrical considerations. The goal is to choose a good set of regions to define the spline approximation for the problem at hand [target function $f(\mathbf{x})$ (1.1)]. Both statistical and computational considerations restrict their number to be very small relative to that generated by a complete tensor product of univariate intervals. Selecting a small subset of basis functions from those representing the complete tensor product has the effect of producing a spline approximation on a corresponding (small) set of (larger) regions, each of which is a selected union of regions from the original tensor product.

The attractive aspects of such a procedure are far more dramatic in the multivariate case than in univariate ($n = 1$) settings.[15] First (and foremost) its adaptability, which can be useful in univariate fitting, is absolutely crucial in approximating all but the simplest functions of high dimensional arguments. The procedure automatically chooses the approximating regions in the n-dimensional predictor variable space. As a consequence it chooses the number of (distinct) variables that enter into each corresponding basis function (interaction order). It also chooses which particular variables comprise the basis functions that enter the model, thereby providing automatic variable subset selection. Candidate basis functions involving predictor variables unrelated to the response are less likely to be selected. Moreover, this variable subset selection aspect is a local property; namely, in any local region of the predictor variable space, basis functions defining its subregions are most likely to involve only the variables most strongly associated with the response in that particular region. This local variable subset selection property, along with the ability automatically to adjust the relative amount of smoothing in each local region of the n-dimensional predictor space, provides considerable flexibility parsimoniously to approximate a wide range of functions.

A consequence of the basis function subset selection implementation is the ease with which constraints can be applied to the solution. Basis functions in the candidate tensor product pool that violate any (user supplied) constraints are simply made ineligible for selection. For example, if an additive model

$$\hat{f}(\mathbf{x}) = \sum_{j=1}^{n} f_j(x_j) \tag{2.8}$$

(no interactions among the variables) was potentially thought to be adequate, all candidate basis functions involving more than one variable would be made ineligible for inclusion in the model.[4] Just as easily interactions can be limited to particular variables, and even to the particular other variables with which they are permitted to interact. A particular variable may also be restricted to enter only linearly. MARS may thus be used to fit semiparametric models in which some variables may enter linearly and others in interactions produced by tensor product splines. Hence linear models, additive models and semiparametric models are all subclasses of models which may be fit using the MARS procedure.

A feature of approximations based on tensor product functions is the straightforward ability to handle predictor variables of different types. Predictor variables that assume values in different kinds of sets are easily incorporated into the same regression model. So far it has been assumed that they all have values on real intervals. For these types of variables ordinary spline functions (2.2) are appropriate. For other types of predictors (e.g. periodic, categorical) the ordinary spline functions may be replaced by an alternate set of basis functions tailored to the specific predictor type. This is discussed for the case of categorical predictors in Section 2.3, Categorical variables.

There are two basic problems that limit the straightforward application of the MARS strategy outlined above; they are computational feasibility and model selection. The total number of candidate basis functions in the full tensor product is $O(N^n)$ which, except for very small values for both quantities (n, N), would require prohibitive resources to compute and store. Implementing the procedure as it is described above would require $O(N^n)$ (partial) linear least-squares fits to enter each new basis function. In order for the procedure to be practical, a computationally feasible algorithm is necessary. This is described in Section 2.2.

Model selection also presents a difficult problem. Like all variable selection procedures that use the data response values to choose a subset, MARS is a highly nonlinear fitting procedure. This provides it with its power and flexibility but causes all of the usual model selection criteria for linear procedures to be inappropriate (see ref. 18). Of these only ordinary cross-validation implemented by explicitly refitting with observations removed[19] or (explicit) bootstrapping[20] survive as statistically viable alternatives. Model selection based on cross-validation, and an approximate criterion that is more rapidly computable, are described in Section 2.2, Model selection.

In addition to these two basic problems, there are a large number of 'engineering details' concerning the implementation that while having no direct bearing on the fundamental ideas, nonetheless have a substantial impact on performance. These are discussed in Friedman.[6]

2.2 The MARS algorithm

This section presents a brief overview of the MARS algorithm that is described in full detail in Friedman.[6] The goal is to provide a computationally feasible approach that approximates the basis function subset selection procedure outlined in the previous section. It chooses a (relatively small) sub-basis, based on the data at hand, from the (very large) n-variable complete tensor product spline basis (2.6) with knots at every distinct marginal data value. One representation for these basis functions is

$$B_m(\mathbf{x}) = \prod_{k=1}^{K_m} [s_{km}(x_{v(k,m)} - t_{km})]_+^q. \tag{2.9}$$

Here K_m is the number of factors (interaction order) in the mth basis function, s_{km} assumes only two values, $s_{km} = \pm 1$, and indicates the (left/right) sense of the truncation, $v(k, m)$ labels the predictor variables, $1 \leq v(k, m) \leq n$ and t_{km} is a knot location on each of the corresponding variables. The exponent q is the order of the spline approximation. This 'two-sided' truncated power basis (2.9) is equivalent to the tensor product truncated power basis (2.2), (2.6) when the monomials $\{x_j^k\}_{k=1}^{q-1}{}_{j=1}^{n}$ on each variable, and an overall constant $B_0(\mathbf{x}) = 1$, are included.

2.2.1 Forward/backward stepwise basis function selection

The MARS algorithm uses a forward/backward stepwise strategy to produce a set of basis functions (2.9). The forward part is an iterative (recursive) procedure. Each iteration simultaneously constructs an expanded list of basis functions to be considered and then decides which ones to enter at that step. Each iteration adds two new basis functions to the current model. This forward stepwise procedure is continued until a relatively large number of basis functions are included, in a deliberate attempt to overfit the data.[3] A final appropriately sized basis function set is then selected through a backward stepwise variable subset selection procedure using the basis functions produced by the forward algorithm as candidate 'variables'. The model selection criterion used with the backward stepwise procedure is described in the next section, Model selection.

The forward stepwise procedure begins with one basis function in the model

$$B_0(\mathbf{x}) = 1. \tag{2.10}$$

After the Mth iteration there are $2M + 1$ functions

$$\{B_m(\mathbf{x})\}_0^{2M} \tag{2.11}$$

in the model, each of the form (2.9). The $(M + 1)$st iteration adds two new basis functions

$$\begin{aligned} B_{2M+1}(\mathbf{x}) &= B_{l(M+1)}(\mathbf{x})[+(x_{v(M+1)} - t_{M+1})]_+^q \\ B_{2M+2}(\mathbf{x}) &= B_{l(M+1)}(\mathbf{x})[-(x_{v(M+1)} - t_{M+1})]_+^q. \end{aligned} \tag{2.12}$$

Here $B_{\ell(M+1)}(\mathbf{x})$ is one of the $2M + 1$ basis functions already chosen (2.9), (2.11), $0 \leq \ell(M + 1) \leq 2M$, $v(M + 1)$ is one of the predictor variables (not represented in $B_{\ell(M+1)}(\mathbf{x})$), and t_{M+1} is a knot location on that variable. The three parameters $\ell(M + 1)$, $v(M + 1)$, and t_{M+1} defining the two new basis functions are chosen to be those that provide the most improvement in the fit of the (new) model to the data

$$\begin{aligned} (l(M+1), v(M+1), t_{M+1}) = \underset{\substack{l,v,t \\ \{a_m\}_0^{2M+2}}}{\operatorname{argmin}} \sum_{i=1}^{N} \Bigg\{ y_i - \sum_{m=0}^{2M} a_m B_m(\mathbf{x}) \\ - a_{2M+1} B_l(\mathbf{x})[+(x_v - t)]_+^q \\ - a_{2M+2} B_l(\mathbf{x})\ [-(x_v - t)]_+^q \Bigg\}^2. \end{aligned} \tag{2.13}$$

Since $B_{\ell(M+1)}(\mathbf{x})$ has the form given by (2.9) the two basis functions $B_{2M+1}(\mathbf{x})$ and $B_{2M+2}(\mathbf{x})$ will also have that form. Their interaction levels K_{2M+1} and K_{2M+2} will be one higher than $K_{\ell(M+1)}$, the interaction level of $B_{\ell(M+1)}(\mathbf{x})$. For example, if $\ell(M + 1) = 0$ (2.10) then two additive (main effect) terms are entered into the model. If $\ell(M + 1) = 0$ (2.10) happens to be chosen at every iteration, then the result will be an additive model (2.8) (sum of functions each of a single variable). Interaction effects are produced by choosing $\ell(M + 1) > 0$.

Although the forward/backward stepwise MARS algorithm produces a basis function subset of the form given by (2.9), and was motivated by the basis function (variable) subset selection strategy described in the previous section, it is not equivalent to that strategy. The MARS algorithm must enter basis functions of low interaction order before it can (construct and) enter basis functions of higher interaction level. It can, of course, later delete the low order interaction terms through the backward stepwise part of the procedure. A faithful implementation of the multivariate adaptive regression spline strategy would however allow any basis function in the complete tensor product basis to enter at any stage. Especially with small to moderate training samples and a large number of variables, the MARS algorithm is likely to favour the entering of lower order interaction terms compared with a faithful rendering of the adaptive spline strategy. This bias toward producing models with relatively low order interactions can represent a strong statistical advantage in those cases where the true underlying function $f(\mathbf{x})$ (1.1) is not dominated by interactions of the very highest order. The strength of this bias is inversely proportional to the training sample size. For small samples the MARS algorithm will try to produce models involving lower order interactions, whereas for larger sample sizes, it will more favourably entertain higher order interactions as potential candidates.

2.2.2 *Model selection*

The forward stepwise MARS algorithm is iterated until $M_{\max}$ (tensor product spline) basis functions are synthesized. An important aspect of the MARS strategy is to choose this number to be substantially larger than would be optimal, and then to delete excess basis functions. The deletion strategy is a standard linear regression backward subset selection procedure with the $M_{\max}$ basis functions representing the stock of 'variables' to be potentially selected/deleted. The motivation for this strategy lies in the (suboptimal) greedy nature of the forward stepwise algorithm. At each iteration it produces two new basis functions using only those that have already been produced in earlier iterations. Thus, the simpler basis functions synthesized early may tend to be highly suboptimal and not very useful when used in conjunction with more complex ones produced in later iterations. Their main contribution in this case is to serve as ingredients (factors) for developing the later basis functions. In order to provide adequate opportunity for the possible synthesis of these more complex (higher interaction order) basis functions, the forward stepwise procedure is allowed to produce an excess number of basis functions, which then compete (on an equal basis) with the earlier ones for inclusion in the final model.

In order to implement this type of model selection, a criterion is required that estimates (future) lack-of-fit on representative data not part of the training sample. The model that minimizes this criterion, when used with the deletion strategy described above, is taken to be the final function estimate. Since the MARS procedure is highly nonlinear, only criteria based on sample reuse such as cross-validation[19] or bootstrapping[20] can be (strictly) justified. The cross-validation criterion is

$$CV(M) = \frac{1}{N}\sum_{i=1}^{N}[y_i - \hat{f}_{M\backslash i}(\mathbf{x}_i)]^2 \tag{2.14}$$

where the dependence of the criterion (and model) on the number of basis functions M is explicitly indicated. Here (2.14) $\hat{f}_{M\backslash i}$ is the M basis function model considered in the backward stepwise deletion process, estimated with the ith (training) observation removed. Due to the hierarchical structure of the set of models considered with the stepwise strategy, this criterion (2.14) can be evaluated for all ($0 \leq M \leq M_{\max}$) models with the same computation required for the evaluation of just one of them (the largest).

The cross-validation criterion (2.14) requires the entire modelling procedure to be reapplied N times, each with one of the observations removed. It is often approximated by an analogous procedure (F-fold cross-validation), that reapplies the modelling $F < N$ times with (approximately N/F different observations being removed each time. [$F = 10$ is often used—see ref. 3.] Friedman[6] proposed an approximation to (2.14) that requires only one evaluation of the model. It is a modification of the generalized cross-validation (GCV) criterion proposed by Craven and Wahba[14] for use in conjunction with linear fitting methods.

$$GCV(M) = \frac{1}{N}\sum_{i=1}^{N}[y_i - \hat{f}_M(\mathbf{x}_i)]^2 / \left[1 - \frac{C(M)}{N}\right]^2 . \qquad (2.15)$$

The numerator of (2.15) is the lack-of-fit on the training data and the denominator represents an (inverse) penalty for increasing model complexity $C(M)$. This criterion can be (strictly) motivated for linear fitting where the basis function expansion is prespecified and only the (linear) expansion coefficients are adjusted to best fit the data. In this case $C(M) = M$, the number of parameters being fitted. The proposed modification[6] for the more general case, where both the basis function set and the expansion coefficients are data determined, is to increase the 'cost-complexity' $C(M)$ to reflect the additional degree to which the model is being fit to the data,

$$C(M) = M \cdot (d/2 + 1) + 1 \qquad (2.16)$$

where here (2.16) M is the number of nonconstant basis functions in the model $\hat{f}_M(\mathbf{x})$, (2.15) being considered. The quantity d in (2.16) represents an additional contribution by each basis function to the overall model complexity resulting from the (nonlinear) fitting of the basis function parameters ℓ, v, and t (2.14) to the data at each iterative step. Its contribution for each basis function is $d/2$ since each such nonlinear fit gives rise to two basis functions.

The quantity d in (2.16) can be regarded as a smoothing parameter of the procedure. Larger values result in fewer basis functions being retained tiereby producing smoother estimates. An optimal value can be estimated through cross-validation. This is equivalent to cross-validating the number of basis functions M (2.14) since there is a one-to-one correspondence between a value for d and the size of the corresponding model produced in any particular situation. A possible advantage to using d is that its value should be more stable across situations involving differing sample sizes since N is explicitly accounted for in the penalty (2.15).

The modified GCV criterion (2.15), (2.16) is motivated by ad hoc heuristics and can only be justified to the extent that it performs well in model selection. Simulation results[6] indicate that this is the case over a wide variety of situations using $d = 3$. The advantage over cross-validation is computational; the MARS algorithm need only be applied once. In many situations (depending on the problem size and computing platform) regular cross-validation (2.14) is routinely feasible. In those cases for which it is not, the modified GCV criterion (2.15), (2.16) represents a computationally feasible alternative, especially for initial exploratory work.

2.3 Extensions to MARS

The basic MARS algorithm described above has been extended to handle non-ordinal predictors and different types of responses.

2.3.1 Categorical variables

The MARS procedure described above assumes that all predictor variables are ordinal, that is, there is an order relation among and a notion of distance between their possible values. After ordinal variables, the most commonly occurring type of variable is nominal or categorical. Such variables assume a discrete set of values

$$x \in \{c_1, \ldots, c_K\} \tag{2.17}$$

that are neither orderable nor possess a distance relation. To model such predictors we must use a different set of basis functions than the splines mentioned previously.

An underlying assumption motivating the use of splines in modelling is that the target function $f(x)$ is relatively smooth, and hence is best modelled by a smooth function. A smooth function $f(x)$ on a categorical variable x is one whose values tend to cluster about a relatively small number of different values, as x ranges over its complete set of values (2.17). A categorical variable 'smoothing' procedure would attempt to discover the particular subsets of x values corresponding to each of the clusters and then produce as its function estimate the mean response value within each cluster.

Such a procedure can be implemented in direct analogy to an adaptive spline strategy by taking the basis functions for categorical predictors to be indicator variables

$$I(x \in A), \qquad I(x \notin A) \tag{2.18}$$

where A is a subset of the possible values of x. The search for knot locations is replaced with a search over subsets, and as before basis functions are entered in complementary pairs. Computational details are presented in Friedman.[7]

Using the above strategy for each categorical predictor, we may then model a function of multiple categorical variables, or of a mixture of categorical and ordinal variables, using the MARS algorithm described previously. By using the spline basis functions for ordinal variables and indicator basis functions for categorical variables, both types of variables along with their interactions may be included in a MARS model.

2.3.2 Nested variables

In some problems there are predictor variables that are meaningful only when some other categorical predictor variable takes on values within a particular subset. For example, a treatment variable x_j may have three possible values: medication, therapy, or surgery. Associated with each of these values is a distinct set of other variables that characterize each corresponding treatment, and only have meaning if that particular treatment is applied. These latter variables are said to be nested within the treatment variable.

In order for nested variables to be treated properly one must ensure that each one only contributes to the model when its value has meaning, as defined by the corresponding value of its nestor. In the context of MARS modelling this constraint can be met by requiring that any basis function involving a nested variable x_v in one of its factors also involves a factor of the form $I(x_j \in A)$, where x_j is the variable to which x_v

is nested, and the set A contains a case of x_j for which x_v has meaning. This ensures that any basis function involving x_v will influence the model only when values of x_v have meaning.

Through a minor modification to the forward stepwise part of the MARS algorithm, a nested variable may be considered as a candidate to enter the model only if the corresponding nestor variable is in the model. Note that as the nestor variable must enter the model before the nestee variable may be considered, this modification places the nested variables at a competitive disadvantage relative to the non-nested variables. It is not unusual for a nested variable to have considerably more predictive power than its nestor variable, especially when the sole purpose of the nesting is to define the existence of values for the nestee (as with missing values in the next section, Missing values), and this predictive power will not show up using the greedy algorithm previously described. To resolve this inequity, a partial 'look ahead' feature is used for nested variables, in which the nestor and nestee variables may be considered together as a possible factor in the model. Details are given in Friedman.[7]

2.3.3 *Missing values*

One of the most useful applications of variable nesting in MARS is in dealing with missing values among the predictor variables. In many problems one is forced to do prediction and/or training in the presence of incomplete date (values for some of the predictor variables are missing). This often has serious consequences for many procedures, either severely degrading their performance or rendering their application impossible.

Missing values among the predictor variables can be handled by introducing an additional indicator variable $x_{v'}$ for each original variable x_v which has missing values. These new variables indicate the presence of a nonmissing value for each corresponding original variable, i.e. $x_{v'} = 0$ if x_v is missing and $x_{v'} = 1$ otherwise. The strategy for variable nesting then ensures that the approximation $\hat{f}(\mathbf{x})$ will exhibit a dependence on each variable x_v only when a value for that variable is present ($x_{v'} = 1$). The partial 'look ahead' for nested variables ensures that variables with missing values compete for entry into the model on the same basis as those with no missing values to the extent that their values are present.

This strategy also allows variables that are highly associated with one another to act as 'surrogates' for one another[3] when their values are missing. Other variables may be involved in a product with $I(x_{v'} = 0)$ and hence used in place of x_v when it is missing. Ramifications of this strategy are explored in greater detail in Friedman.[7]

2.3.4 *Binary response*

When the response y assumes only two values, linear logistic regression is often used.[21] The model takes the form

$$\log[p/(1-p)] = \beta_0 + \sum_{j=1}^{n} \beta_i x_j \tag{2.19}$$

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. This model may be generalized to

$$\log[p/(1-p)] = \hat{f}(\mathbf{x}) \tag{2.20}$$

with $\hat{f}(\mathbf{x})$ taking the form of the MARS approximation. This could be 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, 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 found by fitting 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, while producing a fitted model which takes the binary nature of the response into account. This compromise strategy is available in the FORTRAN implementation of MARS described in Section 4.

Although the logistic regression strategy often improves classification accuracy,[6] sometimes the accuracy of the linear least-squares (fitting the 0/1 response using regular least-squares) and logistic models do not differ markedly. At such times, the user may prefer to use the linear least-squares approach for ease of interpretability. The difference between the two approaches is that the linear approach fits on the probability scale while the logistic approach fits on the log-odds scale. If we are interested in exploring the effect of the predictors on the probability p, it is often preferable to have a model which may be decomposed (as discussed in Section 3) into terms which are additive on the probability scale as opposed to on the log-odds scale. A drawback of using a logistic response when the fitting procedure is flexible is that in pure regions (i.e. regions where the fitted probability is near 0 or 1) the fitted model will approach $\pm\infty$ on the log-odds scale. The large dynamic range of the resulting plots will make it hard to determine the shape of the fitted effects near the decision boundary, which is often the region of interest. One hybrid approach for avoiding this interpretational difficulty is to use the logistic fit for classification, but the linear fit when plotting the effects of the predictors on the response (as in Section 3).

2.3.5 *Other response types*

A number of statisticians have developed methods which are either extensions of MARS or similar in concept to MARS. Hastie, Tibshirani and Buja[22] have developed a multiple response version of MARS and used it for multigroup classification as part of their Flexible Discriminant Analysis method. Kooperberg, Stone and Truong[23] use an approach similar to that of MARS for hazard regression.

3 Interpreting MARS

Applying the MARS procedure produces a model in the form of an expansion in (two-sided) tensor product basis functions (2.9)

$$\hat{f}(\mathbf{x}) = a_0 + \sum_{m=1}^{M} a_M \prod_{k=1}^{K_m} [s_{km}(x_{v(k,m)} - t_{km})]_+^q. \tag{3.1}$$

It can be directly used to estimate missing response values y given a set of predictor variables $\mathbf{x} = (x_1, \ldots, x_n)$. In this form however it is of little interpretive value. One can increase its value for interpreting the nature of the target function $f(\mathbf{x})$ (1) by a simple rearrangement of terms:

$$\hat{f}(\mathbf{x}) = a_0 + \sum_{K_m=1} f_i(x_i) + \sum_{K_m=2} f_{ij}(x_i, x_j) + \sum_{K_m=3} f_{ijk}(x_i, x_j, x_k), \cdots. \tag{3.2}$$

The first sum in (3.2) collects together all basis functions that involve only one variable ($K_m = 1$). Each function $f_i(x_i)$ in that sum is itself a weighted sum of spline basis functions, namely those that involve x_i (and only x_i). Thus each $f_i(x_i)$ is a spline representation of a univariate function (2.1), (2.2). If its argument, x_i, does not appear in any higher order products ($K_m > 1$), then the contribution of x_i to the model is additive (main effect) and can be viewed by simply plotting $f_i(x_i)$ versus x_i.

The second sum analogously collects together all basis functions involving two (and only two) variables ($K_m = 2$). Each $f_{ij}(x_i,x_j)$ is the weighted sum of those basis functions involving both x_i and x_j, but no other variables. These functions (if present) represent two-variable interactions between x_i and x_j, and when added to the corresponding main effect functions (if any)

$$f^*_{ij}(x_i,x_j) = f_i(x_i) + f_j(x_j) + f_{ij}(x_i,x_j) \tag{3.3}$$

yield a tensor product spline representation of a bivariate function. If neither x_i nor x_j appear in higher order interactions, then (3.3) represents their joint contribution to the model that can be visually interpreted by viewing a contour or perspective mesh plot of $f^*_{ij}(x_i,x_j)$ against its arguments. Joint contributions from variables involved in higher (than two) variable interactions (if any) are constructed in an analogous manner by combining their highest order interaction terms with the corresponding lower order ones that are present in the model (3.1), (3.2). These contributions however are not readily viewable through simple graphical techniques.

The representation of the MARS model given by (3.2) is called the ANOVA decomposition since it breaks up the model into main (additive) effects and interaction effects of various orders. Each individual function in (3.2) is called an 'ANOVA function' and is an expansion in tensor produce spline functions involving identical predictor variable sets. (Since the locations of each of the ANOVA functions can be arbitrarily defined, they are each individually translated to have zero minimum value, and the additive constant a_0 (3.2) is adjusted appropriately.) The ANOVA decomposition identifies the variables that enter the model, whether they contribute additively or are involved in interactions, the order of the interaction effects and the particular variables that participate in them.

In many situations the best fitting MARS model is additive (3.8) or involves at most two variable interactions ($K_m \leq 2$). In these cases the model can be fully viewed graphically as described above. When interactions involving more than two variables are required, more advanced techniques must be used. The main approach is to use slicing, in which one chooses a subset of the variables so that when their values are simultaneously fixed, the functional dependence of the MARS model on the complement variables involves at most two-variable interactions which can then be viewed graphically. By examining the changing nature of these graphs as the values of the selected conditioning variables are changed, one can often gain some insight into the multivariate functional relationship. Due to the simple tensor product representation of the MARS model (2.21) such a strategy is especially straightforward to implement. An alternative approach to slicing is to use bivariate or univariate projections of the fitted MARS model onto selected predictor axes. The development of techniques for forming and examining such projection-based functional ANOVAs is a current area of research of the authors.[24]

Note that interpretations drawn from examining ANOVA terms reflect the dependence of the response on the predictors in the MARS model estimated from the data. As such, they strictly describe only the model, and can be inferred to describe the actual dependencies in the data to the extent that the derived model accurately captures the structure underlying the data. As with all statistical inference, it is important to assess the variability of these effects with respect to sampling fluctuations. Owing to the highly adaptive (nonlinear) nature of the MARS procedure, standard methods of statistical inference are not valid. However, bootstrapping techniques[25] can—and should—be applied to assess the sampling variability of any effect deemed to be important. The authors are currently developing methods for exploring the variability and importance of ANOVA terms in a MARS model.[24]

3.1 Degree-of-continuity

One of the properties that characterizes a spline approximation is its order q (3.1). The approximation and its derivatives to order $q - 1$ are constrained to be continuous. There are important statistical and computational considerations involved with this choice in the context of an adaptive spline strategy. These are discussed in detail in Friedman.[12] The strategy outlined there is to use $q = 1$ (piecewise-linear) splines to construct an initial model (2.13), (2.14), (3.1). The discontinuous (first) derivatives thereby produced are then smoothed by using the initial model to derive an analogous piecewise-cubic basis with continuous first derivatives. An important aspect of this strategy is that derivatives are smoothed separately within each ANOVA function (3.2) [see ref. 6, Section 3.7].

3.2 Categorical variables

The ANOVA approach discussed above generalizes directly to categorical variables. Although the response for a categorical variable could be plotted as a step function with the different categories arranged along the predictor axis, it is perhaps more useful to think of displaying categorical ANOVA functions as tables. That is, for each level of a categorical predictor (or in many cases subsets of levels) there is a number representing the contribution to the overall response. In the case of categorical–categorical interactions, the interaction becomes a two-way table. In a categorical–ordinal interaction, we have different curves representing the ordinal variable's contribution to the response for different categorical values.

Slicing is particularly effective with categorical variables. Three way interactions between a categorical variable and two ordinal variables may be represented by a series of perspective plots, as is displayed in Section 5. With missing values, we may slice upon the missingness indicators to create different functions given different sets of variables with missing values.

4 Fitting MARS using FORTRAN

The basic implementation of MARS is a set of FORTRAN subroutines. These routines are available from the StatLib archive at Carnegie Mellon University as `mars3.5`. This may be obtained using anonymous FTP to `lib.stat.cmu.edu` in directory `general` (i.e. `ftp://lib.stat.cmu.edu/general/mars3.5`). These subroutines represent a set of tools that can be invoked from a user coded program to perform various analyses. The user routine is responsible for reading the data into memory and passing it to the MARS subroutines, along with the various parameter settings, as arguments. This set of subroutines can also form the basis for incorporating this methodology into a statistical language or package. This section mentions some of the parameter options available, and the resulting output. A complete description entitled 'A Micro User's Guide to MARS 3.5' written by JH Friedman is distributed as comments to the FORTRAN code.

4.1 Parameter options

The basic inputs to the MARS algorithm are a matrix of predictors, a vector of responses, and optionally a vector of weights to be associated with each (respective) observation. In addition, predictors must be flagged as ordinal or categorical. Preprocessing routines must be called if nesting or missing values are present. For each predictor, the user may restrict it to enter only additively, further restrict it to enter only linearly, or allow its contribution to be fully general (allowing nonlinearity and interactions with other variables). If the response is binary, the user may specify whether linear least-squares fitting or logistic regression (as discussed in Section 2.3, Binary response) should be used.

The two primary user-specified parameters are the maximum number of basis functions included during the forward step (*nk*) and the maximum interaction level (*mi*). Friedman[12] suggests that an *nk* value twice the (estimated) optimal number of basis functions is sufficient. Using $mi = 1$ specifies an additive model, $mi = 2$ a model with at most second order interactions, and $mi = n$ places no restrictions on interaction order.

Additional parameters are available which affect the degree of optimization, the form of GCV criterion used, and the type of cross-validation performed. Although these are generally not of interest to the casual MARS user, the user may want to vary the degree of optimization and the type of cross-validation based on computational resources and time.

4.2 MARS output

The primary MARS subroutine constructs a pair of matrices representing the MARS fit which may be used by auxiliary routines. Subroutines are included with the MARS distribution which take these matrices as inputs and construct new matrices and arrays representing gridded values of the additive and second order ANOVA functions. These may in turn be input to a language such as S to produce plots of the ANOVA functions. Slicing is also supported, and of course predicted values are available given a set of covariates.

When MARS is run, a summary of diagnostic information is sent to the screen. It includes summary information describing the response and predictors. Also, the order in which predictor/knot combinations entered the model is displayed, as is the final set of basis function coefficients. Of particular interest is a summary of the number of effective parameters used on each variable or interaction (a measure of function complexity), and the change in GCV excluding each variable or interaction (a measure of variable importance).

5 Example: heart attack survival data

In this section we illustrate the application of MARS to heart attack survival data and the interpretation of the results. The data used here was provided by the Specialized Center of Research on Ischemic Heart Disease at the University of California, San Diego, and consists of patients who had heart attack symptoms. There are $p = 18$

predictor variables $\mathbf{x} = (x_1, \ldots, x_{18})$ representing various clinical measurements, health history and demographic information. The response variable y is an indicator of one-year survival ($y = 0 \Rightarrow$ survived, $y = 1 \Rightarrow$ died). It is intended as a surrogate for seriousness of the disease. The regression function $f(\mathbf{x}) = E[y \mid \mathbf{x}] = \Pr[y = 1 \mid \mathbf{x}]$ is the nonsurvival probability given a specific set of predictor variable values $\mathbf{x}$; larger values of $f(\mathbf{x})$ indicate higher severity. Applying MARS to these data will provide an estimate $\hat{f}(\mathbf{x})$ for this probability at all simultaneous sets of predictor variable values $\mathbf{x}$, thereby providing an estimate of the severity at the time these measurements were taken. As the logistic response approach did not markedly improve the fit, and the fit is a bit easier to interpret on the probability scale than on the log-odds scale (see Section 2.3, Binary response), we report the results for a linear least-squares fit.

There were a total of $N = 779$ patients in the sample, 702 of whom survived one year and 77 who did not. Since it is generally considered more serious to judge a severely ill patient as not being so, than conversely, each observation for which $y = 1$ (died) is given higher weight in the analysis. We used a weight ratio of 9 to 1 for this analysis.

Table 1 shows the ANOVA decomposition (Section 3) of the resulting MARS estimate. The first column labels each ANOVA function (3.2) that appears in the model. The second column shows the number of basis functions that formed each such ANOVA function, and the last column lists the particular predictor variables that comprise it. Examination of this ANOVA decomposition shows that MARS selected only four of the 18 variables to enter its predictive model. Table 2 provides a description of each of these four variables.

The ANOVA decomposition shows that the MARS model involves several pairs of variables in two-variable interactions and there is one triple involved in a three-way interaction. Most of the interactions involve the variable x_4 (`disdig`). Therefore, the MARS model conditioned ('sliced') on specific values of `disdig` has an especially simple representation involving at most two variable interactions. Furthermore, since

Table 1 MARS ANOVA decomposition

ANOVA function	Number of basis functions	Variable
1	1	4
2	1	14
3	1	4 15
4	3	4 12
5	1	4 14
6	1	12 15
7	1	4 14 15

Table 2 Description of predictor variables selected by MARS

Variable number	Name	Description
4	**disdig**	patient taking digitalis (no = 0, yes = 1)
12	**maxbun**	maximum blood urea nitrogen
14	**age**	patient's age
15	**maxqrs**	enlargement of heart chambers (from electrocardiogram)

disdig takes on only two distinct values (Table 2), the entire MARS model can be visualized in two parts represented by its separate dependence on x_{12},x_{14},x_{15} for disdig = 0, and for disdig = 1.

Figure 1 provides a graphical representation of these two MARS models. The upper two frames are for disdig = 0 and the lower two for disdig = 1. For disdig = 0 the dependence on x_{14} (age) is additive (main effect only) so that in this case (disdig = 0) the dependence on age is independent of the values of x_{12} (maxbun) and x_{15} (maxqrs) and vice versa. This dependence is plotted as a function of age in the upper left frame. One sees that severity (probability of nonsurvival) increases with age reaching a peak at around 70 years old and then steeply decreases after that. For (disdig = 0) the dependence on maxbun and maxqrs involves interaction effects between these two variables so they cannot be represented as separate bivariate plots.

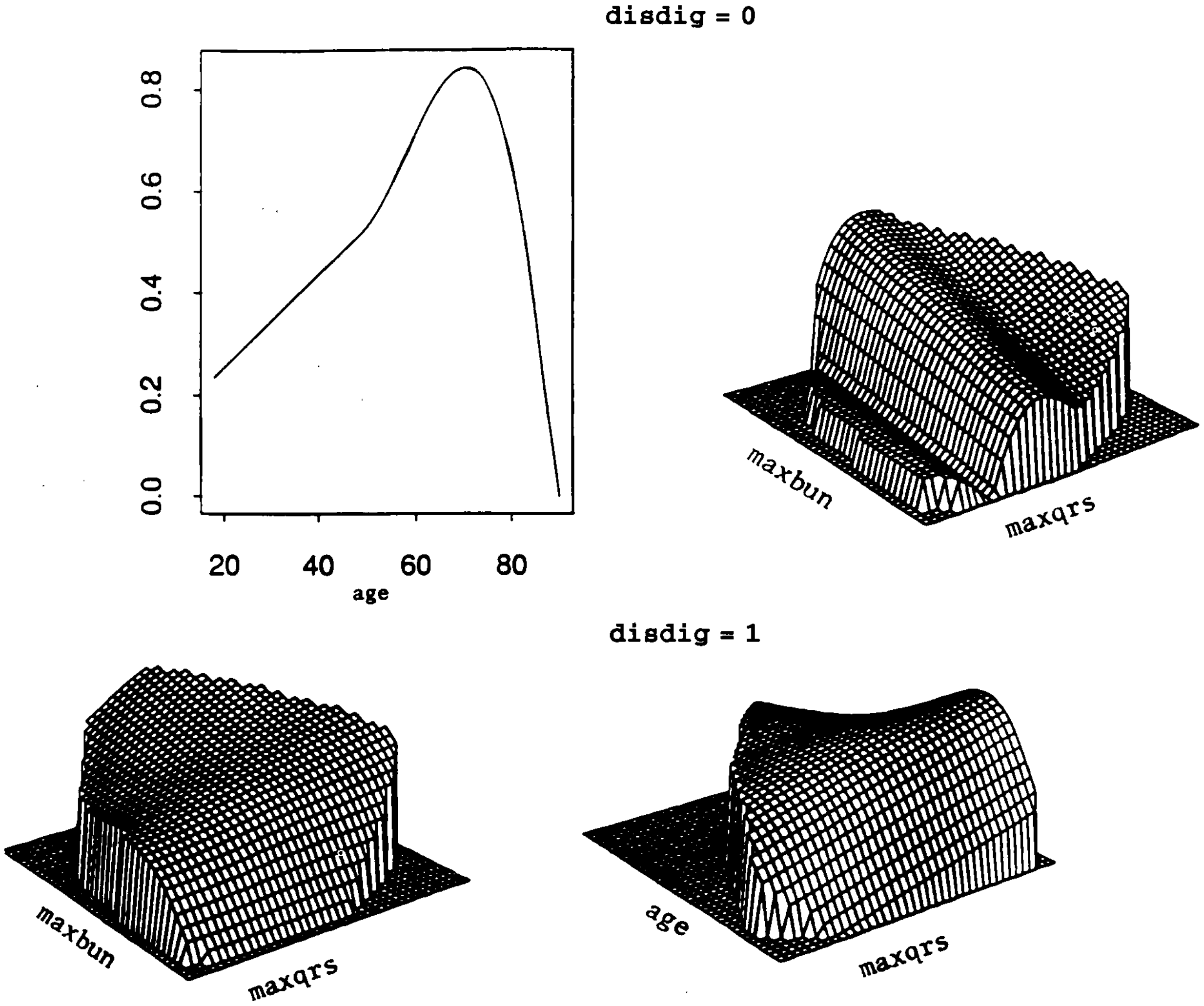

Figure 1 Graphical representation of the ANOVA decomposition for the MARS model fit to the heart attack survival data

Table 3 Relative importance to MARS model of the predictor variables that it selected

Variable importance	disdig	maxbun	age	maxqrs
	100	95	72	83

The upper right frame of Figure 1 represents their joint contribution (for disdig = 0) as a perspective mesh plot. One sees that the shape of the dependence on maxqrs is roughly the same for all values of maxbun, but the amplitude (strength) of that dependence increases with increasing values of maxbun.

For the disdig = 1 model, maxqrs is involved in two-variable interactions with both maxbun and age. These contributions are shown, respectively, as the bottom two frames in Figure 1. For low values of maxqrs, severity tends to increase sharply with increasing (small) values of maxbun levelling off for higher values of maxbun. For higher values of maxqrs, this levelling off effect diminishes somewhat so that severity increases with increasing maxbun over most of its values. The interaction between age and maxqrs (bottom right frame) is somewhat surprising. Severity is least when these two variables jointly assume either their lowest values or their highest values, and is greatest for intermediate age and high values of maxqrs.

In addition to examining the detailed (joint) dependence of the response on the various predictor variables, the MARS model can be used to assess the relative overall importance of each predictor variable separately. Importance of a predictor is defined as the increase in lack-of-fit of the model when that variable is removed from it. Table 3 shows the relative importance of each variable that entered the model, scaled so that the most important one received a value of 100. One sees that disdig was the predictor most important to the model, but the other three (maxbun, age, maxqrs) were also all highly relevant. The importance of the remaining 16 predictor variables (not entering the model) is zero by this measure.

It is important to stress that these interpretations concern the dependence of survival probability on these variables as reflected by the MARS model estimated from the data. As mentioned in Section 3, they strictly describe only the model, and can be inferred to describe the actual dependencies in the data to the extent that the derived model accurately captures the structure underlying the data. The degree to which this is true may be explored using bootstrapping techniques[25] to assess the sampling variability of the effected deemed to be important.

One way to gauge the accuracy of a predictive model is its predictive capability. For a binary valued response, this can be quantified by the misclassification risk when it is used to predict the binary outcome ($y = 0/1$) with the rule

$$\hat{y}(\mathbf{x}) = \begin{cases} 1 & \text{if } \hat{f}(\mathbf{x}) > 0.5 \\ 0 & \text{otherwise.} \end{cases}$$

Here $\hat{y}(\mathbf{x})$ is the prediction given a set of predictor variable values $\mathbf{x}$, and $\hat{f}(\mathbf{x})$ is the corresponding value of the MARS model. The misclassification risk is defined by

$$R = E[L(y,\hat{y}(\mathbf{x}))]$$

where $L(y,\hat{y})$ is the loss (cost) for predicting $\hat{y}$ when the truth is y and the expected value (average) is taken over future patients not involved in deriving the model. In our case $L(0,0) = L(1,1) = 0$, $L(0,1) = 1$ and $L(1,0) = 9$, reflecting our assumption that

Table 4 MARS cross-validated cross-classification matrix

		Predicted value	
		0	1
True	0	0.75	0.25
Value	1	0.39	0.61

it is nine times more costly to misclassify a severely ill patient as not being so, than vice versa. This can be compared to the (null) risk

$$R_0 = \min\{E[L(y,0)], E[L(y,1)]\}$$

which is the minimum risk associated with assigning all patients to the same outcome (in this case $\hat{y}(\mathbf{x}) = 1$). Since we do not have a set of future patients with which to evaluate the above expected values, 10-fold cross-validation is used. The data is randomly partitioned into 10 subsets and for each the misclassification risk is evaluated using the MARS model derived from its complement set of observations. The average risk over these 10 subsets is then used as an estimate of the overall misclassification risk.

The 10-fold cross-validated risk for the MARS model is $R = 0.57$, whereas the null risk is $R_0 = 0.89$. Thus, using the MARS model for classification reduced the misclassification risk by about 32%. This is not a large reduction but it should be kept in mind that absolute risk reduction does not directly reflect the accuracy of the model in predicting the true underlying probability function $f(\mathbf{x}) = \Pr[y = 1 \mid \mathbf{x}]$, which is the objective of the exercise. Accuracy of the model is more directly related to its reduction of risk as compared with that of the (minimum 'Bayes') risk rule associated with using the true underlying probability function $f(\mathbf{x})$ for classification. The Bayes risk is, of course, not known for this problem, but it will have the value zero only if the true probability function takes on one of only two values ($f(\mathbf{x}) = 0$ or $f(\mathbf{x}) = 1$) for all values of the predictor variables.

Table 4 displays the (cross-validated) cross-classification matrix for predicting with the MARS model. One sees that 75% of the survivors ($y = 0$) and 61% of the non-survivors ($y = 1$) were correctly classified.

6 Conclusion

This paper has described the basic Multivariate Adaptive Regression Spline (MARS) algorithm along with many of its extensions. The primary strength of MARS is that it provides a way to fit a highly general regression model while avoiding overfitting. MARS incorporates a wide variety of predictor types in a natural manner, including both continuous and categorical variables. It has the ability to handle nested variables, and can adjust for missing values without discarding data. It is also a good building block for use with more complicated responses, such as the binary response case.

Acknowledgements

Work supported in part by the Department of Energy under contract number DE-AC03-76SF00515, and by the National Science Foundation under grant number DMS-9403804. The authors thank Richard Olshen for information regarding the heart attack survival data used in Section 5.

References

1 Garber AM, Olshen RA, Zhang H, Venkatraman ES. Predicting high-risk cholesterol levels. *International Statistical Review* 1994; **62**: 203–28.

2 Morgan JN, Sonquist JA. Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association* 1963; **58**: 415–34.

3 Breiman L, Friedman JH, Olshen RA, Stone CJ. *Classification and regression trees.* Belmont, CA: Wadsworth, 1984.

4 Friedman JH, Silverman BW. Flexible parsimonious smoothing and additive modeling (with discussion). *Technometrics* 1989; **31**: 3–39.

5 Hastie T, Tibshirani R. *Generalized additive models.* London: Chapman and Hall, 1990.

6 Friedman JH. Multivariate adaptive regression splines (with discussion). *Annals of Statistics.* 1991; **19**: 1–141.

7 Friedman JH. Estimating functions of mixed ordinal and categorical variables using adaptive splines. Department of Statistics, Stanford University, Technical Report No. LCS108, 1991.

8 de Boor C. *A practical guide to splines.* New York, NY: Springer-Verlag, 1978.

9 Shumaker LL. Fitting surfaces to scattered data. In: *Approximation theory III.* In: Lorentz GG, Chui CK, Shumaker LL, eds. New York: Academic Press, 1976: 203–68.

10 Shumaker LL. On spaces of piecewise polynomials in two variables. In: Singh SP, Burry JH, Watson B, eds. *Approximation theory and spline functions.* Boston: D Reidel Publishing Co, 1984: 151–97.

11 Eubank RL. *Spline smoothing and nonparametric regression.* New York: Marcel Dekker, 1988.

12 Wahba G. *Spline models for observational data.* Monograph: SIAM, CBMS-NSF Regional Conference Series in Applied Mathematics, Vol. 59, 1990.

13 Parzen E. On estimation of a probability density function and mode. *Annals of Mathematical Statistics.* 1962; **33**: 1065–76.

14 Craven P, Wahba G. Smoothing noisy data with spline functions. Estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerical Mathematics.* 1979; **31**: 317–403.

15 Smith PL. Curve fitting and modeling with splines using statistical variable selection techniques. Hampton, VA: NASA, Langley Research Center, 1982.

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

17 Bellman RE. *Adaptive control processes.* Princeton, NJ: Princeton University Press, 1961.

18 Breiman L. Submodel selection and evaluation in regression I. The *x*-fixed case and little bootstrap. Department of Statistics, University of California, Berkeley, Technical Report No. 169, 1989.

19 Stone M. Cross-validatory choice and assessment of statistical predictors (with discussion). *Journal of the Royal Statistical Society* 1974; **B36**: 111–47.

20 Efron B. Estimating the error rate of a prediction rule: improvement on cross-validation. *Journal of the American Statistical Association* 1983; **78**: 316–31.

21 Cox DR. *Analysis of binary data.* London: Chapman and Hall, 1970.

22 Hastie T, Tibshirani R, Buja A. Flexible discriminant analysis by optimal scoring. AT&T Bell Laboratories, Technical Report, 1993.

23 Kooperberg C, Stone CJ, Troung YK. Hazard regression. *Journal of the American Statistical Association* 1995; **90**: 78–94.

24 Roosen CB. Visualization and exploration of high-dimensional functions using the functional ANOVA decomposition [Doctoral Dissertation]. Department of Statistics, Stanford University, 1995.

25 Efron B, Tibshirani R. *An introduction to the bootstrap.* London: Chapman and Hall, 1993.