Statistics and Computing (1998) **8**, 337–346

Bayesian MARS

D. G. T. DENISON[1], B. K. MALLICK[2] and A. F. M. SMITH[3]

[1]*Department of Mathematics, Imperial College of Science, Technology and Medicine, 180 Queen's Gate, London SW7 2BZ, UK*
[2]*Department of Statistics, Texas A & M University, College Station, TX 77843-3143, USA*
[3]*Principal, Queen Mary and Westfield College, London, E1 4NS UK*

Submitted November 1996 and accepted May 1998

A Bayesian approach to multivariate adaptive regression spline (MARS) fitting (Friedman, 1991) is proposed. This takes the form of a probability distribution over the space of possible MARS models which is explored using reversible jump Markov chain Monte Carlo methods (Green, 1995). The generated sample of MARS models produced is shown to have good predictive power when averaged and allows easy interpretation of the relative importance of predictors to the overall fit.

Keywords: Bayesian methods, reversible jump Markov Chain Monte Carlo, multiple regression, multivariate adaptive regression splines

1. Introduction

A common problem in statistics, and other disciplines, is to approximate adequately a function of several variables. In a statistical setting this is known as multiple regression and the task can be performed either parametrically by global modelling (e.g. linear regression), or nonparametrically (see below for examples of these methods).

The aim is to model the dependence of a response variable Y on one or more predictor variables $\mathbf{X} = (X_1, \ldots, X_m)$ where the data is given by $(y_i, \mathbf{x}_i)(i = 1, \ldots, n)$ and $\mathbf{x}_i = (\mathrm{x}_{1i}, \ldots, \mathrm{x}_{mi})$. The data is assumed to come from a relationship described by

$$y_i = f(\mathbf{x}_i) + \epsilon, \quad \mathbf{x}_i \in D \subset R^m \quad (i = 1, \ldots, n)$$

where f is an unknown regression function that we wish to estimate, ϵ is a (known) zero-mean error distribution, most commonly assumed to be Gaussian, and D is the domain of interest, usually taken to be the convex hull defined by the predictor variables. The regression function f gives the predictive relationship of Y on $\mathbf{X}$, i.e. the conditional expectation of Y given $\mathbf{X}$. Thus we may use f to predict future values of Y at previously unseen points in the domain D. The aim of the regression analysis is to use the data to construct an estimate $\hat{f}(\mathbf{X})$ to the true regression function which can serve as a reasonable approximation over the domain of interest, D.

Many methods exist to model the function of interest f [e.g. Additive models, Hastie and Tibshirani (1990); CART, Breiman *et al.* (1984); Projection pursuit regression, Friedman and Stuetzle (1981); Alternating conditional expectation, Breiman and Friedman (1985)]. We, however, concentrate on the multivariate adaptive regression spline (MARS) methodology proposed by Friedman (1991). This method seems to be both highly flexible and easily interpretable. It was motivated by the recursive partitioning approach to regression (Morgan and Sonquist, 1963; Breiman *et al.*, 1984) but produces a continuous model which can be made to have continuous derivatives and has greater flexibility to model relationships that are nearly additive or involve at most a few variables. The model can be represented in such a way that the additive contributions of each predictor variable and the interactions between variables can be easily identified which helps to identify variables which are important in the model.

To highlight the progression from recursive partition regression to MARS we start by giving the partition regression model,

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{k} a_i B_i(\mathbf{x}) \tag{1}$$

where $\mathbf{x} \in D$ and the $a_i (i = 1, \ldots, k)$ are the suitably chosen coefficients of the basis functions B_i and k is the number of basis functions in the model. These basis functions are such

that $B_i(\mathbf{x}) = I(\mathbf{x} \in R_i)$ where I is the indicator function which is one where the argument is true, zero elsewhere and the $R_i (i = 1, \ldots, k)$ form a partition of D.

The usual MARS model is the same as that given in (1) except that the basis functions are different. Instead the B_i are given by

$$B_i(\mathbf{x}) = \begin{cases} 1, & i = 1 \\ \prod_{j=1}^{J_i} \left[s_{ji} \cdot \left(x_{v(ji)} - t_{ji} \right) \right]_+, & i = 2, 3, \ldots \end{cases} \quad (2)$$

where $(\cdot)_+ = \max(0, \cdot), J_i$ is the degree of the interaction of basis B_i, the s_{ji}, which we shall call the sign indicators, equal ± 1, the $v(j, i)$ give the index of the predictor variable which is being split on the t_{ji} (known as *knot points*) give the position of the splits. The $v(j, \cdot)(j = 1, J.)$ are constrained to be distinct so each predictor only appears once in each interaction term. See Section 3, Friedman (1991) for a comprehensive illustration of the model.

To illustrate the initially confusing notation we present an example. Suppose a MARS model contains the basis function B_i given by

$$B_i = (x_2 - 1.3)_+ [-(x_5 + 3.7)]_+.$$

We can immediately see that there are two factors in the interaction term so $J_i = 2$. The sign indicators are $s_{1,i} = 1, s_{2,i} = -1$ with the knot points given by $t_{1,i} = 1.3, t_{2,i} = -3.7$ and the labels for the predictors split on are $v(1, i) = 2, v(2, i) = 5$.

The MARS model is continuous in D and can be made to have continuous first derivatives by replacing the truncated linear basis functions B_i by truncated cubic basis functions. This has the effect of 'rounding' the basis function at the split points.

The MARS algorithm proceeds as follows. A forward stepwise search for basis functions takes place with the constant basis function the only one present initially. At each step the split which minimises some 'lack-of-fit' criterion from all the possible splits on each basis function is chosen. Splits are only permissable at the marginal predictor values. If the split was on basis B_i with predictor x_* at t_* this corresponds to the two new basis functions, henceforth referred to as a *pair*, $B_i[\pm(x_* - t_*)]_+$. Note that unlike the recursive partitioning algorithm the basis function with which the split was made is not removed. This continues until the model reaches some predetermined maximum number of basis functions, which should be about twice the number expected in the model to aid the subsequent backwards stepwise deletion of basis functions. This just involves removing basis functions one at a time until the lack-of-fit criterion is at a minimum. The basis which improves the fit the most or degrades it the least is removed at each step. Finally the resulting model can be made to have a continuous first derivative by 'rounding' at the split points as mentioned above. The lack-of-fit measure used by Friedman (1991) is the generalized cross-validation criterion which was originally proposed by Craven and Wahba (1979).

The aim of this paper is to provide a Bayesian algorithm which mimics the MARS procedure. This is done by considering the number of basis functions, along with their *type* (see Section 2.1), their coefficients and their form (the positions of the split points and the sign indicators) random. We treat these as additional parameters in the problem and make inference about them using the data.

The problem of routine calculation of the posterior distribution of the models is addressed by designing a suitable Markov chain Monte Carlo (MCMC) reversible jump simulation algorithm as set out by Green (1995). The simulated sample contains many different MARS models with corresponding posterior weights but if a estimate for f with high predictive power is all that is required then pointwise averaging over all the models in the sample is suggested.

This work is an extension to the Bayesian approach to curve fitting in one dimension given by Denison *et al.* (1998b) and is related to the Bayesian CART algorithms proposed by Denison *et al.* (1998a) and Chipman *et al.* (1998).

In Section 2 we outline the Bayesian MARS method and show examples using the method on simulated data in Section 3 and real data in Section 4. Section 5 contains a discussion of the proposed methodology.

2. The Bayesian MARS method

2.1. *Basic ideas*

We must first define what we mean by the *type* of a basis function. Using the notation in (1) and (2), we consider basis functions B_i, B_j to be of the same type if $[v(1, i), \ldots, v(J_i, i)]$ is identical to some permutation of $[v(1, j), \ldots, v(J_j, j)]$. Hence with m predictor variables there are N different types of basis function where N is given by

$$N = \sum_{i=1}^{m} \binom{m}{i} = 2^m - 1. \quad (3)$$

Note that the sum does not include the constant basis function term [for which i would equal 0 in (3)] as this basis B_1 is always the sole constant basis function in the model so it cannot be chosen as a candidate basis. Frequently some maximum order of interaction I is assigned to the model such that $J_i \leq I (i = 1, \ldots, k)$ in which case the sum in (3) only runs from 1 up to I. We let $T_i \in \{1, 2, \ldots, N\}$ denote the type of basis function B_i thus T_i, in effect, just tells us which predictor variables we are splitting on, *i.e.* what the values of $v(1, i), \ldots, v(J_i, i)$ are.

As an example suppose we have a problem with just two predictors ($m = 2$). Then the types of basis functions that could be in the model (not included the constant one) are

$[\pm(x_1 - *)]_+$ (say type 1), $[\pm(x_2 - *)]_+$ (say type 2) and $[\pm(x_1 - *)]_+[\pm(x_2 - *)]_+$ (say type 3). So for all those basis functions which split only on predictor x_2 their types T_i are all equal to 2.

We propose a model which can be used to set up a probability distribution over the space of possible MARS structures. Any MARS model can be uniquely defined by the number of basis functions present, the coefficients and the types of the basis functions, together with the knot points and the sign indicators associated with each interaction term. This means that we make the k, a_i, T_i, t_{ji} and s_{ji} random with the J_i uniquely defined via the T_i.

We change the dimension of the model when we change k and so, for Bayesian computation, we use a reversible jump MCMC approach (Green, 1995; Richardson and Green, 1997) when we are considering changes in the number of basis functions in the model.

Inference is carried out assuming that the 'true' model is unknown but comes from the class of models $M_1, M_2, \ldots$ where M_k denotes the model with exactly k basis functions. The overall parameter space Θ can then be written as a countable union of subspaces $\Theta = \bigcup_1^\infty \Theta_k$ where Θ_k is a subspace of the Euclidean space $\mathbb{R}^{n(k)}$, where $\mathbb{R}^{n(k)}$ denotes the $n(k) = \sum_{i=2}^k 2(1 + J_i)$ dimensional parameter space corresponding to model M_k. Here $\theta^{(k)} = (\mathscr{B}_1, \ldots, \mathscr{B}_k)$ where each $\mathscr{B}_i$ is the $2(1 + J_i)$-dimensional vector $(a_i, T_i, t_{1,i}, s_{1,i}, \ldots, t_{J,i}, s_{J,i})$ which corresponds to basis function B_i.

There is a natural hierarchial structure to this setup, which, denoting a generic element of Θ_k by $\theta^{(k)}$ and the data vector by y, we formalize by modelling the joint distribution of $(k, \theta^{(k)}, y)$ as

$$p(k, \theta^{(k)}, y) = p(k)p(\theta^{(k)}|k)p(y|k, \theta^{(k)})$$

that is, as the product of model probability, parameter prior and likelihood.

Bayesian inference about k and $\theta^{(k)}$ will be based on the joint posterior $p(k, \theta^{(k)}|y)$, which we shall explore and summarize by regarding it as the target distribution for tailored MCMC computations. It will often be useful to consider this in the factorized form

$$p(k, \theta^{(k)}|y) = p(k|y)p(\theta^{(k)}|k, y).$$

We will generate samples from the joint posterior of $(k, \theta^{(k)})$ by using a class of reversible jump Metropolis–Hastings algorithms (Green, 1995). Full details of the method can be found in the reference cited. Here, we focus on the essence of the methodology and the particular forms of the algorithms in our current context.

2.2. *The Bayesian model*

We assume ϵ in (1) follows a $\mathrm{N}(\mathbf{0}, \sigma^2)$ distribution where σ^2 is unknown. As a result we extend the parameter vector θ to include this new unknown. This leads to the log-likelihood of the model, $l_k(\theta|y)$, being given by

$$l_k(\theta|y) = -n \log \sigma - \frac{1}{2\sigma^2} \sum_{i=1}^n \left\{ y_i - \hat{f}(\mathbf{x}_i) \right\}^2.$$

where $\hat{f}$ is of the form given in (1) and (2).

We use a vague, but proper, prior for the variance of the error distribution *i.e.* $\pi(\sigma^{-2}) = \mathrm{Gamma}(0.01, 0.01)$ and the T_i are assumed to be uniformly distributed on $\{1, \ldots, N\}$. The sign indicators s_{ji} and knot points t_{ji} are also assumed uniform on the sets $\{-1, 1\}$ and $\{1, \ldots, n\}$ respectively. We use another vague, but proper prior, for the coefficients of basis functions so that we assume the $a_i \sim \mathrm{N}(0, \tau^2)$ where we take the variance $\tau^2 = 10^4$. These priors may be chosen differently but this formulation is used to let the data dictate the form of the model and leads to a proper posterior distribution as all the priors are themselves proper. In particular, the prior on the type of the basis functions could be chosen more carefully so that, *a priori*, main effects are favoured over interaction terms. This could be useful when interpretation of the model, rather than prediction, is more important. This refinement is used in Mallick *et al.* (1997, 1998) who concentrate on situations where interpretability of the model is of great interest.

A Poisson distribution (with parameter λ) is used to specify the prior probabilities for the number of basis functions, giving

$$p(k) = \frac{\lambda^k}{(e^\lambda - 1)k!}, \quad k = 1, 2, \ldots$$

In practice, a Poisson distribution truncated to $k < k_{\max}$, for a suitable choice of $k_{\max}$, is adopted. However, to help the sampler to mix better and to limit prior influence we put a gamma hyperprior (with both parameters equal to 10) over λ. This reflects knowledge that we expect just a few basis functions will fit the data well which controls overfitting.

2.3. *Computational strategy*

Our aim is to simulate samples from the joint posterior distribution of $p(\theta^{(k)}, k|y)$, since analytic or numerical analyses are totally intractable in this situation. For this purpose we design a reversible jump algorithm of the general type discussed by Green (1995), to which the reader is referred for details.

In the context of our problem, with multiple parameter subspaces of different dimensionality, it will be necessary to devise different types of moves between the subspaces. These will be combined to form what Tierney (1994) calls a hybrid sampler, making random choice between available moves at each transition, in order to traverse freely around the combined parameter space.

We use the following move types: (a) a change in a knot location; (b) the addition of a basis function; (c) the dele-

tion of a basis function. Note that in steps (b) and (c) we are changing the dimension of the model and that we do not add basis functions in pairs as in the standard MARS forward-stepwise procedure: in fact, we depart completely from the any sort of recursive partitioning approach. We have found that adding basis functions singly makes our procedure more flexible and the reversibility condition easier to satisfy.

When we change the MARS structures, as described below, the coefficients of the basis functions $a_i (i = 1, \ldots, k)$ in the new model have little relationship to those in the current model so inference about them is difficult and can lead to labelling problems, as in Richardson and Green (1997). Instead we choose to integrate out the coefficients a_i and the error variance σ^2 from the parameter vector $\theta^{(k)}$, which now only contains the model parameters. This is straightforward because we chose to use conjugate priors for both the coefficients and the error variance. However, to make predictions using the generated sample of models we draw coefficients for each model in the sample from their full conditional distributions given the other model parameters.

Given the current model step (a) is straight-forward. First we pick a basis function, $B_i (2 \leq i \leq k)$, uniformly at random and then we pick one of the factors $j (1 \leq j \leq J_i)$ where we alter the current knot location t_{ji} and with probability $\frac{1}{2}$ reverse the sign indicator. We choose a new knot location from the marginal predictor values of variable $x_{v(j,i)}$ and set this to the new t_{ji}. This move type is then undertaken using a Metropolis step (Metropolis *et al.*, 1953; Hastings, 1970) to accept or reject the proposed new state.

The addition of a basis function (BIRTH), step (b), is carried out by choosing uniformly a type of basis function, say T_i, to add to the model. Then we uniformly choose a knot location and sign indicator for each of the J_i factors in this new basis.

Step (c) (DEATH), the deletion of a basis function is constructed in such a way as to make the jump step reversible. This is easily done by choosing a basis function uniformly from those present (except the constant basis function B_1) and removing it.

At the end of each iteration after the move step has been performed we use Gibbs steps (Gelfand and Smith, 1990) to generate a new λ. This is straightforward as its full conditional is simple to calculate and is given by $(\lambda|\cdot) \sim \text{Gamma}(10 + k, 11)$. So at each full cycle of the algorithm we obtain a sample of $(k, \theta^{(k)})$.

2.4. *Algorithm*

In the reversible jump algorithm we use the three move types described above so that we can write the set of moves as $m = \{C, 1, 2, \ldots\}$. Here C refers to changing a knot location and $m = 1, 2, \ldots$ refers to increasing the number of terminal nodes from m to $m + 1$ or decreasing it from $m + 1$ to m. Independent move types are randomly chosen with probabilities ρ_k for $m = C$, b_k for $m = k$ and d_k for $m = k - 1$ which satisfy $\rho_k + b_k + d_k = 1$ for all k. In this problem we took $b_k = c \min\{1, p(k+1)/p(k)\}$ and $d_{k+1} = c \min\{1, p(k)/p(k+1)\}$ for $k = 2, 3, \ldots$ with the constant c, a parameter of the sampler, taken to be 0.4. For $k = 1$ we put $b_1 = 1$, $d_1 = \rho_1 = 0$.

We find that by marginalizing over the coefficients and the error variance the acceptance probability given in Green (1995) simplifies to

$$\alpha = \min\left\{1, \frac{p(D|\theta')p(\theta')\,S(\theta' \to \theta)}{p(D|\theta)\,p(\theta)\,S(\theta \to \theta')}\right\} \tag{4}$$

where θ denotes the current model parameters, θ' the proposed model parameters, $S(\theta \to \theta')$ the probability of proposing a move to θ' from θ and D the data. Thus the acceptance probability is just a Bayes factor (Kass and Raftery, 1995) multiplied by prior and proposal odds terms. This approach is common in many fixed dimensional parameter inference problems and has recently been used in other contexts where there are an unknown number of parameters (Chipman *et al.*, 1998; Holmes and Mallick, 1997).

The use of conjugate prior distributions allows simple evaluation of the integrals in the Bayes factor term as demonstrated in O'Hagan (1994). This leads to the Bayes factor being given by

$$\frac{p(D|\theta')}{p(D|\theta)} = \frac{|\tau^{-2}I|^{1/2}|V'|^{1/2}}{|\tau^{-2}I'|^{1/2}|V|^{1/2}}\left(\frac{d}{d'}\right)^{\gamma_1+(n/2)} \tag{5}$$

where $'$ refers to the proposed model, I is the identity matrix, $V = (X^tX + \tau^2 I)^{-1}$ and $d = \gamma_2 + Y^tY - \hat{a}^tV^{-1}\hat{a}$. Note that X and Y refer to the usual design and data matrices of the current regression with $\hat{a} = (X^tX + \tau^2 I)^{-1}X^tY$ being the Bayesian least squares regression estimates of the coefficients. The constants γ_1 and γ_2 are the parameters of the gamma prior distribution over σ^{-2} and so, from Section 2.2, were both taken to be 0.01. Note that an ordinary least squares approach to estimating the regression coefficients was undertaken in Denison *et al.* (1998a, b) and this is equivalent to the above method when using reference priors for the coefficients and error variance.

We now show how the prior and proposal odds terms are calculated using the birth step as an example. A birth step (b) adds basis B_{k+1} when there are currently k basis functions in the model. The prior odds are given by

$$\frac{p(\theta')}{p(\theta)} = \frac{\text{prior for the } (k+1) \text{ basis functions and } \theta^{(k+1)}}{\text{prior for the } k \text{ basis functions and } \theta^{(k)}}$$

$$= \frac{p(k+1)\left[k!/N^k\right](1/2n)^{\Sigma_2^{k+1}J_j}}{p(k)[(k-1)!/N^{k-1}](1/2n)^{\Sigma_2^{k}J_j}} \tag{6}$$

$$= \frac{p(k+1)}{p(k)}\frac{k}{N}\left(\frac{1}{2n}\right)^{J_{k+1}} \tag{7}$$

where the terms in the numerator of (6) are given by the prior for the number of basis functions, the prior for the

type of the basis functions and the priors for the knot positions and sign indicators: similarly for the denominator. The prior probability for the set of bases, when there are k in the model, can be thought of as the probability of choosing $(k-1)$ items from a set of N where the ordering does not matter. The constant basis function is fixed so it does not need to be chosen. The prior for the model parameters is the product of the probability of each factor having a certain sign indicator (i.e. $\frac{1}{2}$) and a certain knot point $(1/n)$ to the power of the number of factor terms in the model $(\sum_2^k J_j)$.

The corresponding proposal odds is given by

$$\frac{S(\theta' \to \theta)}{S(\theta \to \theta')} = \frac{p\left(\text{propose death}\theta^{(k+1)} \to \theta^{(k)}\right)}{p\left(\text{propose birth}\theta^{(k)} \to \theta^{(k+1)}\right)} = \frac{d_{k+1}/k}{b_k/[N(2n)^{J_{k+1}}]} \tag{8}$$

where we propose a death by randomly choosing one of the basis functions not including the constant one, and a birth by randomly choosing a type of basis function, with probability $1/N$, together with a sign indicator and knot point for each factor in the new basis $\{N(2n)\}^{-J_{k+1}}$. Hence, using equations (4–5) and (7–8), we can find the acceptance probability for a birth step. The acceptance probability for a death step is worked out similarly.

The algorithm we use is straightforward and works quickly. The BIRTH step is described in detail in the appendix and the other steps are very similar.

Algorithm

1. Start with just the constant basis function present
2. Set k equal to the number of basis function in the current structure.
3. Generate u uniformly on [0, 1].
4. Goto move type determined by u.
   - If $(u \le b_k)$ then goto BIRTH step
   - else if $(b_k < u \le b_k + d_k)$ then goto DEATH step
   - else goto CHANGE step.
5. Draw λ using Gibbs steps.
6. Repeat 2 for a suitable number of iterations once there is evidence of convergence.

3. Simulated examples

3.1. *Bivariate predictors*

We first of all test our methodology on the examples given by Hwang *et al.* (1994) and studied by Roosen and Hastie (1994). Following their approaches we generate 225 pairs of predictors uniformly on the unit square and the response is $f(x_{1i}, x_{2i}) + \epsilon_i$, where $f(x_1, x_2)$ is the true value of the test function and the ϵ_i are drawn from a $N(0, 0.25^2)$ distribution. The test functions are

- Simple interaction function
$$f^{(1)}(x_1, x_2) = 10.391[(x_1 - 0.4)(x_2 - 0.6) + 0.36]$$
- Radial function
$$f^{(2)}(x_1, x_2) = 24.234[r^2(0.75 - r^2)], \quad r^2 = (x_1 - 0.5)^2 + (x_2 - 0.5)^2$$
- Harmonic function
$$f^{(3)}(x_1, x_2) = 42.659\left[0.1 + \hat{x}_1(0.05 + \hat{x}_1^4 - 10\hat{x}_1^2\hat{x}_2^2 + 5\hat{x}_2^4)\right] \quad \text{where } \hat{x}_1 = x_1 - 0.5,\ \hat{x}_2 = x_2 - 0.5$$
- Additive function
$$f^{(4)}(x_1, x_2) = 1.3356\{1.5(1 - x_1) + e^{2x_1 - 1}\sin[3\pi(x_1 - 0.6)^2] + e^{3(x_2 - 0.5)}\sin[4\pi(x_2 - 0.9)^2]\}$$
- Complicated interaction function
$$f^{(5)}(x_1, x_2) = 1.9\{1.35 + e^{x_1}\sin[13(x_1 - 0.6)^2] \times e^{-x_2}\sin(7x_2)\}$$

These functions are scaled and translated to have a standard deviation of one and a non-negative range. We use the fraction of variance unexplained (FVU) to test the fits of the models to the data, given by

$$\text{FVU} = \frac{\text{E}[\hat{f}(\mathbf{x}_i) - f(\mathbf{x}_i)]^2}{\text{E}[f(\mathbf{x}_i) - \bar{f}]^2} \tag{9}$$

where $f(\mathbf{x}_i)$ is the true value of the function, $\hat{f}(\mathbf{x}_i)$ is the fitted value and $\bar{f}$ is the mean of the true values. We use FVU as it is helpful in comparing fits of the model with differently generated datasets. To evaluate the FVU for a fit we replace the expectations by averages over a test set of 10 000 points. For these bivariate examples this is simply a 100 by 100 grid on the unit square i.e. $(1/200, 3/200, \ldots, 199/200) \times (1/200, 3/200, \ldots, 199/200)$. For the higher dimensional examples which follows the test set is composed of 10 000 random uniform values over the domain of interest D. For the training set we calculate the FVU over the training sample treating the observed y values as $f(\mathbf{x})$.

We took the results from the last 30 000 iterations of the sampler after an initial burn-in period which was long enough for convergence to have occurred by the end of it. Convergence was assumed when the mean squared error of the fit had been settled for some time, as in the similar curve fitting algorithm of Denison *et al.* (1998b).

In Table 1 we display the FVU for the training set of data and the test set for the standard MARS algorithm together with the number of basis functions it found. For the BMARS model we give the average FVU for the posterior mean model (obtained by pointwise averaging) from

Table 1. *FVU for Hwang examples*

Function	Method	FVU train	FVU test	No. of basis functions
Simple	BMARS	0.0484	0.0068	5.3
	LMARS	0.0491	0.0140	7
	CMARS	0.0495	0.0102	7
Radial	BMARS	0.0574	0.0178	9.5
	LMARS	0.0538	0.0187	12
	CMARS	0.0558	0.0169	12
Harmonic	BMARS	0.0813	0.0806	15.0
	LMARS	0.1066	0.0763	16
	CMARS	0.0998	0.0723	16
Additive	BMARS	0.0475	0.0172	11.2
	LMARS	0.0509	0.0109	11
	CMARS	0.0482	0.0098	11
Complex	BMARS	0.0575	0.0411	14.5
	LMARS	0.0837	0.0844	17
	CMARS	0.1883	0.1649	17

10 runs of the algorithm and we display the average number of basis functions in the samples produced by the 10 repetitions of the algorithm. The standard MARS models are referred to as LMARS (piecewise-linear MARS) and CMARS (piecewise-cubic MARS) and we give results for both these MARS models. Table 1 shows that the BMARS model gives comparable, and often better, results than both LMARS and CMARS for this wide variety of examples. Also, the average number of basis functions found by BMARS is commonly less than the number found using standard MARS.

The true surfaces for these examples are shown in Figure 1 and the corresponding BMARS estimates are given in Figure 2.

3.2. *High dimensional predictors*

We take this example from Friedman *et al.* (1983). The basic function is

$$f(\mathbf{x}) = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5. \quad (10)$$

Following Friedman *et al.* we generate 200 random uniform predictors from the six-dimensional unit hypercube and take the response to be $f(\mathbf{x}_i) + \epsilon_i$ where the ϵ_i are independent and identically distributed N(0, 1) random variables. The extra predictor is spurious and does not affect the response. Friedman (1991) also uses this function but in this paper a ten-dimensional predictor is used (five being spurious) and the sample size is reduced to 100. As is commonly the case in Friedman (1991) we do not allow more than two-way interactions in the MARS models we use. Higher-order interaction terms do not generally improve the fit and make the model unncessarily complex even though they could be incorporated if required.

In Table 2 we display the FVU for the training and test set for these examples using the standard MARS and BMARS methods. As before we took the results from the last 30 000 iterations of 10 runs of the algorithm after an initial burn-in period. The results shown demonstrate how the BMARS model parsimoniously fits the data when compared to the standard MARS fits.

4. Real data example

We now illustrate our methodology using a real dataset. We use data from a study by Bruntz *et al.* (1974) of the dependence of **ozone** on some meteorological variables on 111 days from May to September 1973 at sites in the New York metropolitan area. As in Cleveland and Devlin (1988) we work with the cube root of **ozone**. This dataset is known as **air** and is available in Splus (Becker *et al.*, 1988). There are three predictor variables, **radiation**, **temperature** and **wind speed** but because of the vastly differing ranges of the response and predictors we standardized the data beforehand, i.e. we linearly transformed each variable so that it had zero mean and unit variance. Again we allow only main-effect and two-way interaction terms in the MARS models.

The MARS fit had six basis functions and the residual sum of squares (RSS) given by

$$\mathrm{RSS} = \sum_{i=1}^{n} \left(y_i - \hat{f}(\mathbf{x}_i) \right)^2. \quad (11)$$

was 18.41 with the linear approximation and 20.19 with the cubic one. The final model was of the form f(**wind**) + f(**temperature**) + f(**wind**, **temperature**) + f(**radiation**, **temperature**) with one basis function for each of these terms except the final one for which two basis functions existed.

Over five runs of the algorithm, using the same priors as in the previous section, the average RSS given by the BMARS model was 18.32 with 4.06 basis functions. This is lower than both the RSS's given by the MARS model with fewer basis functions. As shown by Figure 3 the BMARS estimate in the (**temperature**, **wind**) plane is smooth whereas the piecewise-linear estimate using MARS is not (Fig. 4). The BMARS estimate also has a smaller RSS with less degrees of freedom.

In Table 3 we display the estimated posterior probabilities of the possible terms in the models, the RSS and the average number of basis functions in each of the five runs. Immediately it can be seen that they each produced similar results which suggests that convergence had occurred by the end of the burn-in period. Also, it appears that the most important basis functions in the fit were the main effect terms for **radiation** and **temperature** and the (**wind**, **temperature**) interaction term. This is borne out by the fact that the model which included only the terms just re-

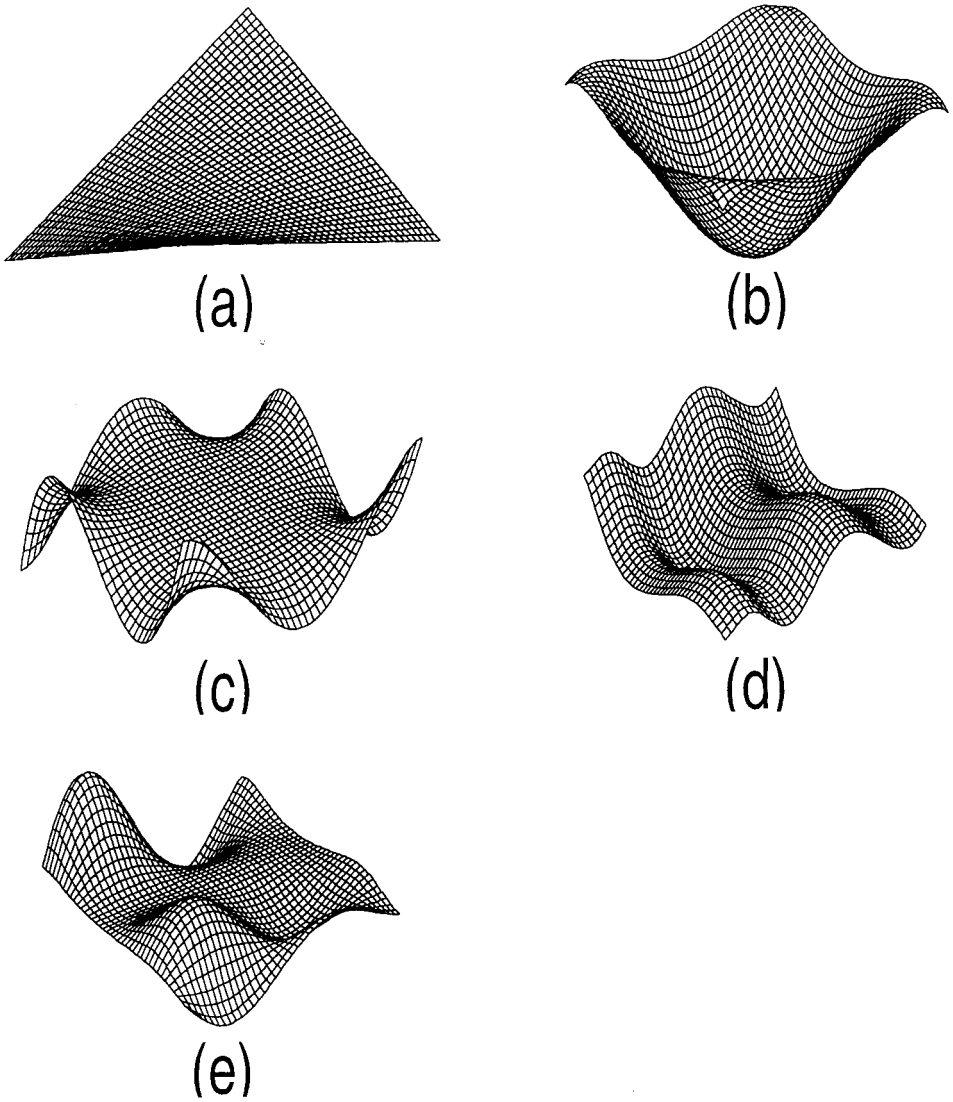

Fig. 1. *Hwang et al. (1994) examples. True surfaces: (a) Simple interaction; (b) Radial; (c) Harmonic; (d) Additive; (e) Complex interaction*

ferred to had easily the largest posterior probability and made up well over 50% of the generated samples.

The use of BMARS as a tool for performing a stochastic search for variable selection comes 'for free' when performing the analysis for prediction. Variable selection, itself, is a difficult problem and one that has received much attention in the literature. The BMARS method could be used in a similar way to the Gibbs sampling-based method outlined in George and McCulloch (1993) which is shown to identify good models using a stochastic search procedure.

5. Discussion

We have presented a Bayesian approach to finding regression surfaces which uses truncated linear basis functions to build up the surface. We use the data to find the knot points, the main effect and/or interaction terms and the number of basis functions that are required to adequately approximate the required surface. We simulate a random sample of models using the reversible jump MCMC approach of Green (1995).

This Bayesian approach to multiple regression produces a model with high predictive power due to the posterior averaging over all the models. This not only leads to a good overall fit for the data but can also help to combat overfitting problems. We can however choose a single model as our estimate by picking out single models from the sample by various criteria. These models have less predictive power but have more interpretability as we can easily produce their ANOVA decompositions. We can only produce expected numbers and posterior probabilities of basis func-

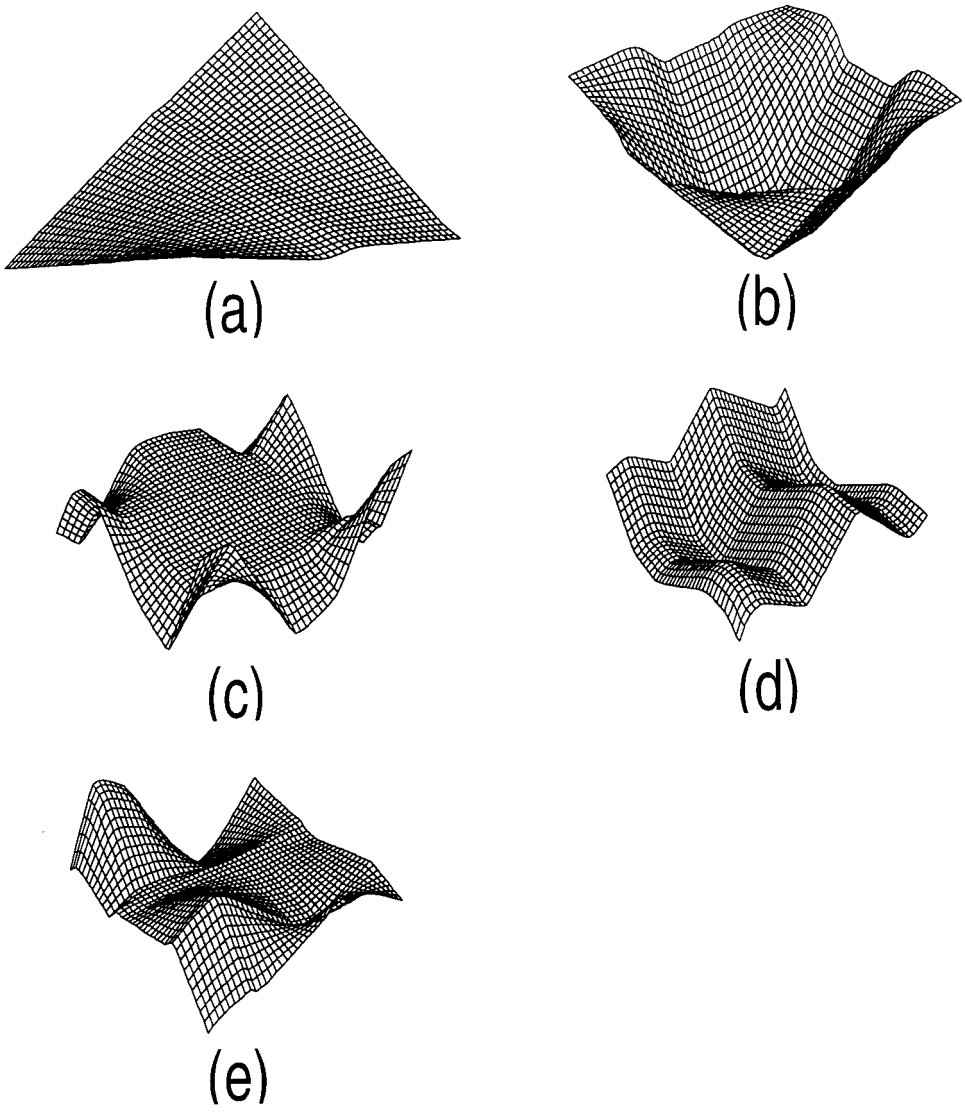

Fig. 2. *Hwang et al. (1994) examples. BMARS estimates: (a) Simple interaction; (b) Radial; (c) Harmonic; (d) Additive; (e) Complex interaction*

tions for the posterior mean estimate but this too can be used to identify variables that are important to the fit.

Table 2. *FVU for Friedman example with different number of spurious predictors*

	Method	FVU train	FVU test	No. of basis functions
Friedman ($m = 6$)	BMARS	0.0416	0.0146	7.9
	LMARS	0.0379	0.0156	16
	CMARS	0.0358	0.0124	16
Friedman ($m = 10$)	BMARS	0.0447	0.0118	8.9
	LMARS	0.0481	0.1582	16
	CMARS	0.0490	0.1092	16

In some applications we may not wish to draw the coefficients but use the mean of their sampling distributions instead. If identifying good models, and not prediction, is the main aim then not drawing the coefficients is slightly quicker and produces models with smaller (squared) error. This may be useful in the related BMARS algorithms for analysing financial time series (Denison, 1997) and failure time data (Mallick *et al.*, 1997). This is the approach Denison *et al.* (1998a) undertake to find 'good' CART models via a stochastic search as in this example prediction using the posterior mean is not helpful.

The structure of the algorithm which allows the sampler to move fluidly about the probability space means that Bayesian MARS does not inherit the problems of Bayesian CART algorithms (Denison *et al.*, 1998a; Chipman *et al.*, 1998) by only searching restricted portions of the entire

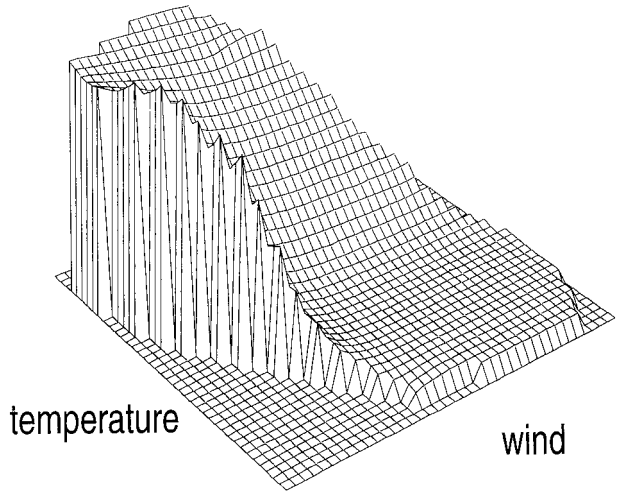

Fig. 3. *Typical (run 4) BMARS estimate in the* (**temperature**, **wind**) *plane*

space. This is greatly helped by not 'splitting' on current basis functions thus inducing no hierarchical structure to the form of the bases. This point is demonstrated in full in Chapter 5 of Denison (1997). If the space of predictor variables is substantial (over 50, say) then convergence of the sampler would seem to be unlikely but still the BMARS approach should yield 'good' models.

The BMARS algorithm we used in this paper was written in ANSI C and is available, together with the datasets, from the World Wide Web address **http : //ma.ic.ac.uk ~dgtd/**. The algorithm takes around 5 min to run on a DEC Alpha workstation when there are 200 datapoints.

Acknowledgements

The work of the first author was supported by an EPSRC research studentship. We acknowledge the helpful comments made by the anonymous referees, the associate editor and Mr C.C. Holmes. The third author's research was partially supported by National Cancer Institute grant CA-57030.

Table 3. *The posterior probabilities of the main effects and interaction terms in the five runs*

Run	1	2	3	4	5
RSS	18.30	18.38	18.24	18.38	18.30
Basis fns	4.08	4.06	4.08	4.06	4.02
p [f (**rad**)]	0.97	0.93	0.89	0.96	0.91
p [f (**temp**)]	1.00	1.00	1.00	1.00	1.00
p [f (**wind**)]	0.00	0.00	0.00	0.00	0.00
p [f (**rad**, **temp**)]	0.03	0.04	0.08	0.04	0.05
p [f (**rad**, **wind**)]	0.02	0.04	0.04	0.03	0.05
p [f (**wind**, **temp**)]	1.00	1.00	1.00	1.00	0.97

References

Becker, R., Chambers, J. M. and Wilks, A. (1988) *The New S Language*, Wadsworth, Belmont, California.

Breiman, L. and Friedman, J. H. (1985) Estimating optimal transformations for multiple regression and correlation (with discussion). *Journal of the American Statistical Association*, **80**, 580–619.

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

Bruntz, S. M., Cleveland, W. S., Kleiner, B. and Warner, J. L. (1974) The dependence of ambient ozone on solar radiation, temperature and mixing height, *in Symposium on atmospheric diffusion and air pollution*. American Meteorological Society, Boston, pp. 125–8.

Chipman, H., George, E. I. and McCulloch, R. E. (1998) Bayesian CART model search (with discussion). *Journal of the American Statistical Association* **93**, 935–960.

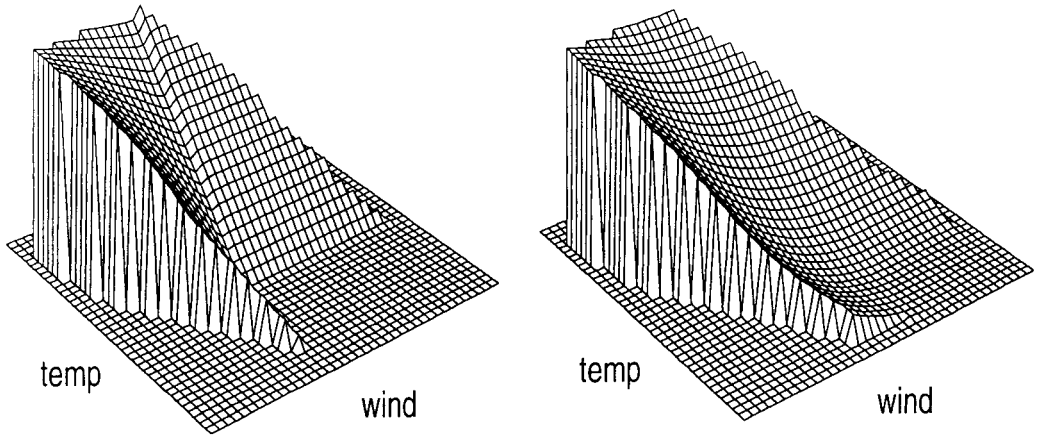

Fig. 4. *Linear-piecewise (left) and cubic-piecewise (right) MARS estimates in the* (**temperature**, **wind**) *plane*

Cleveland, W. S. and Devlin, S. J. (1988) Locally-weighted regression: an approach to regression analysis by local fitting. *Journal of the American Statistical Association*, **83**, 597–610.

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

Denison, D. G. T. (1997) Simulation based Bayesian nonparametric regression methods. Unpublished Ph.D. Thesis. Imperial College, London.

Denison, D. G. T., Mallick, B. K. and Smith, A. F. M. (1998a) A Bayesian CART algorithm. *Biometrika*, **85**, 363–377.

Denison, D. G. T., Mallick, B. K. and Smith, A. F. M. (1998b) Automatic Bayesian curve fitting. *Journal of the Royal Statistical Society*, Series B, **60**, 333–350.

Friedman, J. H. (1991) Multivariate adaptive regression splines (with discussion). *The Annals of Statistics*, **19**, 1–141.

Friedman, J. H., Grosse, E. and Stuetzle, W. (1983) Multidimensional additive spline approximation. *SIAM Journal of Scientific and Statistical Computing*, 291–301.

Friedman, J. H. and Stuetzle, W. (1981) Projection pursuit regression. *Journal of the American Statistical Association*, **76**, 817– 823.

Gelfand, A. E. and Smith, A. F. M. (1990) Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, **85**, 398–409.

George, E. I. and McCulloch, R. E. (1993) Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, **88**, 881–889.

Green, P. J. (1995) Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, **82**, 771–32.

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

Hastings, W. K. (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**, 97–109.

Holmes, C. C. and Mallick, B. K. (1997) Bayesian wavelet networks for non-parametric regression. Technical report. Imperial College, London.

Hwang, J-N, Lay, S-R, Maechler, M., Martin, D. and Schimert, J. (1994) Regression modeling in back-propagation and projection pursuit learning. *IEEE Transactions on Neural Networks*, **5**, 342–53.

Kass, R. E. and Raftery, A. E. (1995) Bayes factors. *Journal of the American Statistical Association,* **90**, 773–795.

Mallick, B. K., Denison, D. G. T. and Smith, A. F. M. (1997) Bayesian survival analysis using a MARS model. Technical report. Imperial College, London.

Mallick, B. K., Denison, D. G. T. and Smith, A. F. M. (1998) Semiparametric generalized linear models: Bayesian approaches, *in Generalized Linear Models*: *A Bayesian Perspective*, Dey, D. K., Ghosh, S. K. and Mallick, B. K. (eds) Marcel-Dekker (to appear).

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E. (1953) Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, **21**, 1087–1091.

Morgan, J. N. and Sonquist, J. A. (1963) Problems in the analysis of survey data and a proposal. *Journal of the American Statistical Association*, **58**, 415–434.

O'Hagan, A. (1994) *Kendall's Advanced Theory of Statistics*, *Volume 2B*, Edward Arnold, London.

Richardson, S. and Green, P. J. (1997) Bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society* (Ser. B), **59**, 731–792.

Roosen, C. B. and Hastie, T. J. (1994) Automatic smoothing spline projection pursuit. *Journal of Computational and Graphical Statistics*, **3**, 235–48.

Tierney, L. (1994) Markov chains for exploring posterior distributions (with discussion). *The Annals of Statistics*, **22**, 1701–1762.

Appendix

The move types CHANGE, BIRTH and DEATH given in the algorithm in Section 2.3 are undertaken similarly so we just describe the BIRTH step in pseudo-code. The notation follows that in Section 2.

BIRTH

1. Uniformly choose a type of basis function T_i, to add from the N possible ones.
2. Uniformly choose the knot positions, predictors to split on and the sign indicators in this new basis remembering that each predictor may only occur once in each basis function.
3. Generate u uniformly on [0, 1].
4. Work out the acceptance probability, α.
5. IF $(u < \alpha)$ accept the proposed model.
   ELSE keep the current model.
6. Return to main algorithm.