Spline Adaptation in Extended Linear Models

Mark H. Hansen and Charles Kooperberg [1]

Abstract. In many statistical applications, nonparametric modeling can provide insight into the features of a dataset that are not obtainable by other means. One successful approach involves the use of (univariate or multivariate) spline spaces. As a class, these methods have inherited much from classical tools for parametric modeling. For example, stepwise variable selection with spline basis terms is a simple scheme for locating knots (breakpoints) in regions where the data exhibit strong, local features. Similarly, candidate knot configurations (generated by this or some other search technique), are routinely evaluated with traditional selection criteria like AIC or BIC. In short, strategies typically applied in parametric model selection have proved useful in constructing flexible, low-dimensional models for nonparametric problems.

Until recently, greedy, stepwise procedures were most frequently suggested in the literature. Research into Bayesian variable selection, however, has given rise to a number of new spline-based methods that primarily rely on some form of Markov chain Monte Carlo to identify promising knot locations. In this paper, we consider various alternatives to greedy, deterministic schemes, and present a Bayesian framework for studying adaptation in the context of an extended linear model (ELM). Our major test cases are Logspline density estimation and (bivariate) Triogram regression models. We selected these because they illustrate a number of computational and methodological issues concerning model adaptation that arise in ELMs.

Key Words and phrases: Adaptive triangulations; AIC; Density estimation; Extended linear models; Finite elements; Free knot splines; GCV; Linear splines; Multivariate splines; Regression.

1. Introduction

Polynomial splines are at the heart of many popular techniques for nonparametric function estimation. For regression problems, TURBO (Friedman and Silverman, 1989), multivariate adaptive regression splines or MARS (Friedman, 1991) and II (Breiman, 1991) have all met with considerable success. In the context of density estimation, the Logspline procedure of Kooperberg and Stone (1991, 1992) exhibits excellent spatial adaptation, capturing the full height of spikes without overfitting smoother regions. And finally, among classification procedures, classification and regression trees or CART (Breiman *et al.* 1984) is a de facto standard, while the more recent PolyMARS models (Kooperberg *et al.* 1997) have been able to tackle even large problems in speech recognition. Stone *et al.* (1997) and a forthcoming monograph by Hansen, Huang, Kooperberg, Stone and Truong are the prime references for the application of polynomial splines to function estimation. In this paper, we review a general methodological framework common to procedures like MARS and Logspline, and contrast it with several Bayesian approaches to spline modeling. We begin with some background material on splines.

[1] Mark H. Hansen is a Member of the Technical Staff, Bell Laboratories, Murray Hill, NJ 07974. Charles Kooperberg is Associate Member, Division of Public Health Sciences, Fred Hutchinson Cancer Research Center, Seattle, WA 98109-1024.

1.1 Splines

Univariate, polynomial splines are piecewise polynomials of some degree d. The breakpoints marking a transition from one polynomial to the next are referred to as *knots*. In this paper, we will let the vector $\boldsymbol{t} = (t_1, \ldots, t_K) \in \mathbb{R}^K$ denote a collection of K knots. Typically, a spline will also satisfy *smoothness constraints* describing how the different pieces are to be joined. These restrictions are specified in terms of the number of continuous derivatives, s, exhibited by the piecewise polynomials. Consider, for example, piecewise linear curves. Without any constraints, these functions can have discontinuities at the knots. By adding the condition that the functions be globally continuous, we force the separate linear pieces to meet at each knot. If we demand even greater smoothness (say, continuous first derivatives), we loose flexibility at the knots and the curves become simple linear functions. In the literature on approximation theory, the term "linear spline" is applied to a continuous, piecewise linear function. Similarly, the term "cubic spline" is reserved for piecewise cubic functions having two continuous derivatives, allowing jumps in the third derivatives at the knots. In general, it is common to work with splines having *maximal smoothness* in the sense that any more continuity conditions would result in a global polynomial.

Given a degree d and a knot vector $\boldsymbol{t}$, the collection of polynomial splines having s continuous derivatives forms a linear space. For example, the collection of linear splines with knot sequence $\boldsymbol{t}$ is spanned by the functions

$$1, x, (x - t_1)_+, \ldots, (x - t_K)_+ \tag{1}$$

where $(\cdot)_+ = \max(\cdot, 0)$. We refer to this set as the *truncated power basis* of the space. In general, the basis for a spline space of degree d and smoothness s is made up of monomials up to degree d together with terms of the form $(x - t_k)_+^{s+j}$, where $1 \leq j \leq d - s$. Using this formula the classical cubic splines have $d = 3$ and $s = 2$ so that the basis has elements

$$1, x, x^2, x^3, (x - t_1)_+^3, \ldots, (x - t_k)_+^3 \,. \tag{2}$$

From a modeling standpoint, the truncated power basis is convenient because the individual functions are tied to knot locations. In the expressions (1) and (2), there is exactly one function associated with each knot, and eliminating that function effectively removes the knot. This observation is at the heart of many statistical methods that involve splines and will be revisited shortly.

The truncated power functions (1) and (2) are known to have rather poor numerical properties. In linear regression problems, for example, the condition of the design matrix deteriorates rapidly as the number of knots increases. An important alternative representation is the so-called *B-spline basis* (de Boor, 1978). These functions are constructed to have support only on a few neighboring intervals defined by the knots. (For splines having maximal smoothness, this means $d + 1$ neighboring intervals.) A detailed description of this basis is beyond the scope of this paper, but the interested reader is referred to Schumaker (1993). For the moment, assume we can find a basis $B_1(x; \boldsymbol{t}), \ldots, B_J(x; \boldsymbol{t})$ for the space of splines of degree d with smoothness s and knot sequence $\boldsymbol{t}$ so that any function in the space can be written as

$$g(x; \boldsymbol{\beta}, \boldsymbol{t}) = \beta_1 B_1(x; \boldsymbol{t}) + \cdots \beta_J B_J(x; \boldsymbol{t}) \,, \tag{3}$$

for some coefficient vector $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_J)^t$. If we are dealing with spline spaces of maximal smoothness, then $J = K + d + 1$, as we have seen in (1) and (2). Given this structure, we now briefly describe a broad collection of estimation problems that admit relatively natural techniques for identifying good fitting functions g.

1.2 Extended linear models

Extended linear models (ELMs) were originally defined as a theoretical tool for understanding the properties of spline-based procedures in a large class of estimation problems (Hansen, 1994; Stone *et al.* 1997; Huang, 1998, 2001). This class is extremely rich, containing all of the standard generalized linear models as well as density and conditional density estimation, hazard regression, censored regression, spectral density estimation and polychotomous regression. To describe an ELM, we begin with a probability model $p(W|h)$ for a (possibly vector-valued) random variable W that depends on an unknown (also possibly vector-valued) function h. Typically, h represents some component of the probability model about which we hope to make inferences. For example, in a normal linear model, h is the regression function; while for density estimation, we take h to be the log-density.

Let $l(W|h) = \log p(W|h)$ denote the log-likelihood for an ELM, and assume that there exists a unique function ϕ that maximizes the expected log-likelihood $El(W|h)$ over some class of functions H. The maximizer ϕ defines "truth," and is the target of our estimation procedures. The class H is chosen to capture our beliefs about ϕ, and is commonly defined through smoothness conditions (e.g., we might assume that the true regression function in a linear model has two continuous, bounded derivatives). These weak assumptions about ϕ tend to result in classes H that are infinite dimensional. Therefore, for estimation purposes we choose to work with flexible, finite-dimensional spaces G that have good approximation properties. That is, the elements $g \in G$ can capture the major features of functions $\phi \in H$, or $\min_{g \in G} \|g - \phi\|$ is small in some norm for all $\phi \in H$. Splines are one such approximation space.

Given a series of observations $W_1, \ldots, W_n$ from the distribution of W, we estimate ϕ by maximizing the log-likelihood

$$l(g) = \sum_i l(g, W_i) \qquad \text{where} \qquad g \in G\,. \tag{4}$$

We refer to this set-up as an *extended linear model* for ϕ. In this case, the term "linear" refers to our use of a linear space G. Our appeal to maximum likelihood in this context does not imply that we believe $p(W|\phi)$ to be the true, data-generating distribution for W. Rather, p may be chosen for simple computational ease in the same way that ordinary least squares can be applied when the assumption of strict normality is violated. In theoretical studies, it is common to let the dimension of G depend on the sample size n. For example, if G is a spline space with K knots, we let $K = K(n) \to \infty$ as $n \to \infty$. As we collect more data, we are able to entertain more flexible descriptions of ϕ. Asymptotic results describing the number of knots $K(n)$ and their placement needed to achieve optimal mean squared error behavior are given in Stone (1985), Stone (1994), Hansen (1994), and Huang (1998, 2001) and Stone and Huang (2001).

An ELM is said to be concave if the log-likelihood $l(w|h)$ is concave in h for each value of $w \in \mathcal{W}$ and if $El(W|h)$ is strictly concave in h (when restricted to those h for which $El(W|h) > -\infty$). Strict concavity holds for all of the estimation problems listed at the beginning of this section. Now, let G be a spline space of degree d with knot sequence $\boldsymbol{t}$ so that any $g \in G$ can be written in the form (3). Then since $g = g(\cdot; \boldsymbol{\beta}, \boldsymbol{t})$, the log-likelihood (4) can be written as $l(\boldsymbol{\beta}, \boldsymbol{t})$. Because of concavity, the MLEs $\hat{\boldsymbol{\beta}}$ for the coefficients $\boldsymbol{\beta}$ and a fixed $\boldsymbol{t}$ can be found efficiently in reasonably-sized problems through simple Newton-Raphson iterations. Therefore, it is possible to compute

$$l(\boldsymbol{t}) = \max_{\boldsymbol{\beta}} l(\boldsymbol{\beta}, \boldsymbol{t})\,. \tag{5}$$

After making the dependence on $\boldsymbol{t}$ explicit in this way, we can consider adjusting the knot locations $t_1 < \cdots < t_K$ to maximize the log-likelihood. It is intuitively clear that the knot sequence $\boldsymbol{t} = (t_1, \ldots, t_K)$ controls the flexibility of elements in g to track local features: tightly-spaced knots can capture peaks, while widely-separated knots produce smooth fits.

However, even in the simplest case, linear regression with a single univariate predictor, maximizing (5) over knot sequences is a difficult optimization problem. To see this, we first translate univariate regression into an ELM: Let $W = (X, Y)$ and define $p(W|h)$ via the relationship

$$Y = h(X) + \epsilon\,,$$

for an unknown regression function h. The error ϵ is assumed independent of X with a normal distribution having mean zero and variance σ^2. For a spline space G, the negative log-likelihood for $\boldsymbol{\beta}$ is proportional to the regression sum of squares

$$RSS(\boldsymbol{\beta}, \boldsymbol{t}) = \sum_i (Y_i - g(x_i; \boldsymbol{\beta}, \boldsymbol{t}))^2 = \sum_i (Y_i - \beta_1 B_1(X_i; \boldsymbol{t}) - \cdots - \beta_J B_J(X_i; \boldsymbol{t}))^2\,.$$

If we hold $\boldsymbol{t}$ fixed, then

$$l(\boldsymbol{t}) \propto \max_{\boldsymbol{\beta}} \{-RSS(\boldsymbol{\beta}, \boldsymbol{t})\} = -RSS(\hat{\boldsymbol{\beta}}, \boldsymbol{t})\,, \tag{6}$$

where $\hat{\boldsymbol{\beta}}$ is the ordinary least squares estimate of $\boldsymbol{\beta}$. Jupp (1978) demonstrated that $-RSS(\boldsymbol{t})$ has local maxima along lines of the form $t_k = t_{k+1}$, making the solution to (6) difficult for standard optimization software. Not surprisingly, this problem persists in even more exotic ELMs.

Several authors have considered special transformations, penalties or ad hoc optimization schemes to maximize the log-likelihood with respect to $\boldsymbol{t}$ (Jupp, 1978; Lindstrom, 1999; Kooperberg and Stone, 2001). In this paper, we will instead consider an approximate solution that begins by connecting knot placement with model selection.

1.3 Model selection

The concavity of an ELM together with the association of knot locations to terms in the truncated power basis suggests simple approximations to maximizing (5) based on fast, stepwise approaches to model selection. Consider splines of degree d having maximal smoothness and knot sequence $\boldsymbol{t}$. According to Section 1.1, this means that $s = d - 1$ and each knot point t_k in $\boldsymbol{t}$ is associated with only one of the truncated monomials $(x - t_k)_+^d$; the linear (1) and cubic (2) splines are two examples. Therefore, moving t_k effects only one basis element in G, and in fact removing t_k entirely is equivalent to deleting $(x - t_k)_+^d$ from the model. Many existing spline methods use this idea in some form. It was originally proposed by Smith (1982ab) and it has been the workhorse of many procedures suggested since (TURBO, DKCV, MARS, PolyClass).

Returning to the problem of maximizing (5), suppose we have a finite set of candidate knots $\mathcal{T} = \{t'_1, \ldots, t'_{K'}\}$, from which we want to select a subset of size K, $\boldsymbol{t} = \{t_1, \ldots, t_K\}$, $K \leq K'$. The connection between knots and basis functions suggests that finding a good sequence $\boldsymbol{t}$ is really a problem in model selection where we are choosing from among candidate basis functions of the form $(x - t)_+^d$, $t \in \mathcal{T}$. For linear regression and moderate numbers of candidate knots K', we can find the sequence of length K that minimizes (6) using traditional branch-and-bound techniques. However, when K' gets large, or when we

have a more exotic ELM requiring Newton-Raphson iterations to evaluate (5), this approach quickly becomes infeasible.

For computational efficiency, the algorithms discussed by Stone *et al.* (1997) take a stepwise approach, introducing knots in regions where the unknown function ϕ exhibits significant features, as evaluated through the log-likelihood, and deleting knots in regions where ϕ appears relatively smooth. More formally, starting from a simple spline model, knots are added successively, at each step choosing the location that produces the greatest increase in the log-likelihood. This is followed by a pruning phase in which unnecessary knots are removed, at each stage eliminating the basis element that results in the smallest change in the log-likelihood. Because we are always taking the best single alteration to the current model, these schemes are often referred to as *greedy*. To prevent this process from tracking spurious patterns in the data, it is common to impose constraints on the initial model, the size M of the largest model fit during addition, and the minimal number of data points between each knot. These restrictions are defined in terms of *allowable spaces*, a topic we will discuss in more detail in the next section.

Several facts about ELMs make this approach attractive computationally. Consider placing a single knot in a linear regression model. Then, among all basis sets of the form $1, x, (x-t)_+$, we want to find the one that minimizes the criterion (6), which in this case is a function of t. It is not hard to show that $RSS(t)$ is a piecewise smooth function of t, with breaks in the first derivative at each of the data points. It is generally true that the log-likelihood (5) for an ELM is a piecewise smooth function of any single knot in $\boldsymbol{t}$. This means we can derive fast heuristics to guide the search for new knots during the addition phase without having to evaluate all the candidates. Next, the concavity of the ELMs listed in Section 1.2 means that we can quickly approximate the change in log-likelihood from either adding or deleting a knot without actually fitting each candidate model. We now describe each alteration or "move" in more detail.

Knot addition Let G be a J-dimensional spline space with a given knot sequence, and let $\widehat{\boldsymbol{\beta}}$ denote the MLE of $\boldsymbol{\beta}$. Inserting a new knot is equivalent to adding a single basis function to G, taking us to a new $J+1$-dimensional space G_1 with coefficient vector $\boldsymbol{\beta}_1$. To evaluate the improvement, we employ a Taylor expansion of the log-likelihood $l(\boldsymbol{\beta}_1)$ around $\boldsymbol{\beta}_1 = (\widehat{\boldsymbol{\beta}}, 0)$, which specifies a function in G_1. This approximation yields the well-known Rao (score) statistic and is convenient because it allows us to entertain a large number of candidate knot locations without having to compute the MLE $\widehat{\boldsymbol{\beta}}_1$ in each candidate space.

Knot deletion Again, let G be a given spline space and $\widehat{\boldsymbol{\beta}}$ the associated MLE. Removing a knot from G reduces the dimension of G by one and takes us to a space G_0. To evaluate the impact of this alteration, we again employ a Taylor expansion, this time around $\widehat{\boldsymbol{\beta}}$. If $a \in \mathbb{R}^J$ represents the linear constraint that effectively removes a given knot, this expansion yields the Wald statistic for testing the hypothesis that $a^t\boldsymbol{\beta} = 0$. For the truncated power basis, a is a binary vector with a single non-zero entry. With this approach, we can compare the impact of removing each knot in G without having to compute the MLE in these reduced spaces.

This produces a sequence of models, from which we select the single best according to some selection criterion like generalized cross validation (GCV)

$$\mathrm{GCV}_a(\boldsymbol{t}) = \frac{RSS(\boldsymbol{t})}{n} \Bigg/ \left[1 - \frac{a(J(\boldsymbol{t})-1)}{n}\right]^2, \tag{7}$$

or a variant of "A Information Criterion" (AIC)

$$\mathrm{AIC}_a(\boldsymbol{t}) = -2\widehat{l}(\boldsymbol{t}) + aJ(\boldsymbol{t}), \tag{8}$$

(Akaike, 1974) where $J(\boldsymbol{t})$ is the dimension of the spline space. The parameter a in each of these expressions controls the penalty assigned to models with more knots and is introduced to offset the effects of selection bias (Friedman and Silverman, 1989; Friedman, 1991). In Stone *et al.* (1997) the default value of a in (8) is $\log n$, resulting in a criterion that is commonly referred to as BIC (Schwarz, 1978).

Notice that our search for good knot locations based on the log-likelihood (5) has led to a heuristic minimization of a selection criterion like (7) or (8). Several comments about this reduction are in order. First, greedy schemes are often criticized for not exploring a large enough set of candidate models. In the stepwise algorithms of Stone *et al.* (1997), for example, the simple two-pass scheme (knot addition to a model of size M followed by deletion) evaluates at most $2M$ different knot sequences. These $2M$ candidates are also highly constrained, representing a potentially narrow path through the search space. As a result, when we identify the "best model" according to some selection criterion, we have visited at most a handful of its "good-fitting" neighbors, those spline spaces with about the same number of knots found during either addition or deletion. However, as is typical with variable selection problems, many spline models offer essentially equivalent fits (in terms of AIC or GCV).

Despite these caveats, examples in Stone *et al.* (1997) and other papers show that greedy algorithms for knot selection can work quite well. They lead to a surprising amount of spatial adaptivity, easily locating extra knots near sharp features, while removing knots in smooth areas. It is natural, however, to question whether or not alternative methods might prove more effective. In the discussion following Stone *et al.* (1997), for example, the Bayesian framework of Smith and Kohn (1996) is shown to approximately minimize the same objective function (8), but with a stochastic search algorithm. In general, the recent work on Bayesian model selection offers interesting solutions to the shortcomings of greedy methods.

1.4 A Bayesian approach

The desire to compare alternative search schemes is half the motivation for this paper. As mentioned earlier, a major source of inspiration comes from the recent work on Bayesian model selection and the accompanying Markov chain Monte Carlo (MCMC) methods for identifying promising models. To date, several Bayesian spline methods have appeared that make the connections with model selection listed above. The first was Halpern (1973) who constructed a hierarchical model for regression with linear splines. This application necessarily focused on small problems with a limited number of potential knots, succumbing to the computational resources of the day. More modern research in this area has followed a similar approach in terms of prior assignment, but makes use of Markov chain Monte Carlo (MCMC) to sample from a (possibly very) large set of candidate knots. Perhaps the first such procedure was exhibited by Smith (1996) and Smith and Kohn (1996) for univariate and additive regression models. Similar in spirit are the Bayesian versions of TURBO and CART proposed by Denison *et al.* (1998ab), which employ reversible jump MCMC (Green, 1995).

In a Bayesian setup, *model uncertainty* comes from both the structural aspects of the space G – knot placement – as well as from our selection of members $g \in G$ – determining coefficients in expression (3). We now spell out a simple hierarchical formulation that we will revisit in the next section. At the first level of the hierarchy, we assign a prior distribution $p(G)$ to some set of candidate models G. In the set-up for

univariate regression using linear splines, for example, we would typically do that by first choosing a prior distribution on the number of knots $p(K)$, and then by choosing an additional prior on the collection of knots $\boldsymbol{t}$ given K, $p(\boldsymbol{t}|K)$. Through $p(\boldsymbol{t}|K)$ we can prevent knots from getting too close, reducing the chance that the fitted model will track spurious features in the data. Next, given a space G, we generate elements g according to the distribution $p(g|G)$. Consistent with our motivation for modeling with splines in the first place, our prior on g should somehow reflect our beliefs about the smoothness of the underlying function of interest in an ELM, ϕ. In the literature on smoothing splines we find a class of such priors that given a basis for G and an expansion (3) involves the coefficients $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_J)$. This amounts to a partially improper, normal distribution for $\boldsymbol{\beta}$ (Silverman, 1985; Wahba, 1990; and Green and Silverman, 1994), which we will return to in Section 2.

Given a prior for spline functions g, we can generate a sample from the posterior distribution of g using Markov chain Monte Carlo (MCMC). In particular, in Sections 2 and 3 we will use the reversible jump algorithm of Green (1995) for Logspline density estimation and Triogram regression, respectively. Details about how to choose priors and how to tune algorithms are discussed in these sections. When properly tuned, these stochastic methods can identify many more good-fitting knot configurations than their greedy, deterministic competitors; By focusing their search in regions of model space that have high posterior probabilities, the MCMC schemes listed above visit many more "promising" configurations.

The second major motivation for this paper is the form of the final function estimate itself. Since deterministic searches identify a very small number of usable models, the unknown function is typically estimated by straight maximum likelihood applied to some basis for the identified spline space. Suppose for the moment, that the function being estimated is smooth in some region, perhaps requiring not more than a single knot to adequately describe the curve. From the point of view of mean squared error, there are many roughly equivalent ways to place this knot in the region. Therefore, if given a number of good knot configurations, it might be more reasonable to combine these estimates in some way. This is roughly a spline or knot-selection version of the classical motivation for Bayesian model averaging. In later versions of the Gibbs sampling approach of Smith and Kohn (1998) and the Bayesian versions of TURBO and MARS by Denison *et al.* (1998ab), the final function estimate is a posterior mean.

In this paper, we compare greedy (stepwise) algorithms with non-greedy (stochastic, Bayesian) algorithms for model selection. We evaluate different approaches to adaptation by examining strategies for both knot placement and coefficient estimation. We focus on four classes of methods: greedy, stepwise procedures with maximum likelihood estimates in the final spline space; MCMC for selecting the posterior mode in a Bayesian framework; Bayesian model averaging using maximum likelihood estimates of the coefficients; and finally a fully Bayesian approach with model and coefficient averaging. Our two main estimation problems will be Logspline density estimation and (bivariate) Triogram regression. We selected these because they illustrate a number of computational and methodological issues concerning model adaptation that arise in ELMs.

In Section 2 we discuss greedy and Bayesian model selection approaches in the context of Logspline density estimation. In Section 3 we turn to Triogram regression, contrasting it with Logspline. Finally, in Section 4 we identify areas of future research. Our goal in preparing this paper was not to advocate one scheme over another, but rather to investigate the performance of various approaches to model selection in the context of univariate and multivariate nonparametric estimation with splines.

2. Logspline density estimation

Recall that density estimation is an example of an ELM. In the notation of the previous section, the target of our analysis, ϕ, is a log-density, and $W = Y$, a random variable taking values in some interval (L, U). If the density of Y has infinite support, then L, U will be $\pm\infty$. In Stone and Koo (1986), Kooperberg and Stone (1991, 1992) and Section 4 of Stone *et al.* (1997) a technique known as Logspline is developed in which ϕ is modeled with a *natural cubic spline.* Like the ordinary cubic splines in (2), these functions are also twice continuously differentiable, piecewise polynomials defined relative to a knot sequence $\boldsymbol{t} = (t_1, \ldots, t_K)$. Within each interval $[t_1, t_2], \ldots, [t_{K-1}, t_K]$, natural cubic splines are cubic polynomials, but on $(L, t_1]$ and $[t_K, U)$ they are forced to be linear functions. It is not difficult to see that this *tail constraint* again yields a linear space, but with dimension K. Also, the space will contain spline terms providing we have at least $K \geq 3$ knots (otherwise we have only linear or constant functions). In this application, we choose a basis of the form $1, B_1(y; \boldsymbol{t}), \ldots, B_J(y; \boldsymbol{t})$, where $J = K - 1$. We chose to make the constant term explicit in this way because it disappears from our model; recall that each density estimate is normalized to integrate to one. Therefore, let G denote the J-dimensional span of the functions $B_1, \ldots, B_J$.

A column vector $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_J)^T \in \mathbb{R}^J$ is said to be *feasible* if

$$C(\boldsymbol{\beta}, \boldsymbol{t}) = \log\left(\int_L^U \exp(\beta_1 B_1(y; \boldsymbol{t}) + \cdots + \beta_J B_J(y; \boldsymbol{t}))dy\right) < \infty.$$

Let $\mathcal{B}$ denote the collection of such feasible column vectors. Given $\boldsymbol{\beta} \in \mathcal{B}$, we define a family of positive density functions on (L, U) of the form

$$g(y; \boldsymbol{\beta}, \boldsymbol{t}) = \exp(\beta_1 B_1(y; \boldsymbol{t}) + \cdots + \beta_J B_J(y; \boldsymbol{t}) - C(\boldsymbol{\beta}, \boldsymbol{t})), \qquad L < y < U. \tag{9}$$

Now, given a random sample $Y_1, \ldots, Y_n$ of size n from a distribution on (L, U) having an unknown density function ϕ. The log-likelihood function corresponding to the Logspline model (9) is given by

$$l(\boldsymbol{\beta}, \boldsymbol{t}) = \sum_i \log g(Y_i; \boldsymbol{\beta}, \boldsymbol{t}) = \sum_i \sum_j \beta_j B_j(Y_i; \boldsymbol{t}) - nC(\boldsymbol{\beta}; \boldsymbol{t}), \qquad \boldsymbol{\beta} \in \mathcal{B}.$$

The maximum likelihood estimate $\widehat{\boldsymbol{\beta}}$ is given by $\widehat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta} \in \mathcal{B}} l(\boldsymbol{\beta}, \boldsymbol{t})$, corresponding to $\widehat{g}(y) = g(y; \widehat{\boldsymbol{\beta}}, \boldsymbol{t})$ for $L < y < U$.

Stepwise knot addition begins from an initial model with K_{init} knots, positioned according to the rule described in Kooperberg and Stone (1992). Given a knot sequence $t_1, \ldots, t_K$, the addition scheme finds a location for a candidate knot corresponding to the largest Rao statistic. For numerical stability, we do not allow the breakpoints $t_1, \ldots, t_K$ to be separated by fewer than a n_{sep} data points. We say that in this context, a space G is *allowable*, providing the knot sequence satisfies this condition. Stepwise addition continues until a maximum number of knots $K_{\max}$ is reached. Knot deletion is then performed according to the outline in the previous section, and a final model is selected according to the generalized AIC criterion (8) with parameter $a = \log n$.

2.1 A Bayesian framework

We set up the framework for a Bayesian approach to Logspline density estimation by selecting several priors: first a prior $p(G)$ on the structure of the model space G, and then a prior $p(g|G)$ on the splines g in a given space. In addition, we will need to specify how we sample from the posterior distributions.

Priors on model space For Logspline we choose to specify $p(G)$ by creating a distribution on knot sequences $\boldsymbol{t}$ formed from some large collection of candidates $\mathcal{T} = \{t_1, \ldots, t_{K'}\}$. We construct $p(G)$ hierarchically, first choosing the number of knots $K < K'$ (in this case recall that the dimension J of G is $K-1$) according to $p(K)$, and then given K, we generate $\boldsymbol{t}$ from the distribution $p(\boldsymbol{t}|K)$. Regularity conditions on the structural aspects of the associated spline space G can be imposed by restricting the placement of $t_1, \ldots, t_K$ through $p(\boldsymbol{t}|K)$. While other authors have also considered a discrete set of candidate knot sequences (Denison *et al.* 1998a; Smith and Kohn, 1996), we could also specify a distribution that treats the elements of $\boldsymbol{t}$ as continuous variables (e.g. Green 1995). In our experiments we have found that for Logspline density estimation the discrete approach is sufficient, and we consider those spaces G for which all K knots are located at data points. This restriction is purely for convenience, but represents little loss of flexibility especially in the context of density estimation (where peaks in the underlying density naturally produce more candidate knots). For numerical stability, we require that there are at least n_{sep} data points in between any two knots.

This leaves us with the task of specifying $p(K)$. To the extent that the number of knots also acts as a smoothing parameter, this distribution can have a considerable effect on the look of the final curves produced. We explore several of the proposals that have appeared in the literature. The first is a simple Poisson distribution with mean γ suggested by Green (1995). Denison *et al.* (1998a) take the same distribution for more general spline spaces and argue that their results are somewhat insensitive to the value of γ. The next prior we will consider was suggested by Smith and Kohn (1996). Either by greatly reducing the number of candidate knots or by scaling the prior on the coefficients, these authors suggest that K be distributed uniformly on the set $K_{\min} \ldots, K_{\max}$.

The final proposal for $p(K)$ is somewhat more aggressive in enforcing small models. To properly motivate this distribution, we think of the model selection procedure as two stages: in the first we find the posterior average of all models with k knots by integrating out $\boldsymbol{t}$ and g, to obtain, say $\bar{g}_k$ and its posterior probability $p(\bar{g}_k|Y_1, \ldots, Y_n, K = k)$. Suppose that we consider $\bar{g}_k$ to have k degrees of freedom (an admittedly questionable assumption). If we now were to use an AIC-like criterion to choose among the $\bar{g}_k$, we would select the model that minimized

$$-2 \log p(\bar{g}_k|Y_1, \ldots, Y_n, K = k) + ak\,,$$

compare (8). On the other hand, using the posterior to evaluate the best model suggests maximizing

$$p(\bar{g}_k|Y_1, \ldots, Y_n, K = k)p(K = k).$$

If we take $p(K = k) \propto \exp(-ak/2)$ these two approaches agree. Thus, taking a geometric distribution for $p(K)$ implies an AIC-like penalty on model dimension. In particular $a = \log n$ and $q = 1/\sqrt{n}$ imposes the same cost per knot as AIC with penalty $\log n$. For reasonable settings of $K_{\min}$ and $K_{\max}$, however, the expected prior number of knots under this prior will tend to zero with n. While it is certainly intuitive that the prior probability of K decreases monotonically with k, this drop may be at a faster rate than we would expect! If $a \geq 2$ then $p(K = k+1)/p(K = k) \leq 1/e$.

Priors on splines in a given space We parameterize $p(g|G)$ through the coefficients $\boldsymbol{\beta}$ in the expansion (3), and consider priors on $\boldsymbol{\beta}$ that relate to our assumptions about the smoothness of g. Recall that as the solution to a penalized maximum likelihood fit, smoothing splines (Wahba, 1990) have a straightforward

Bayesian interpretation (Silverman, 1985). In univariate smoothing, for example, G is a space of natural splines (given some knot sequence $\boldsymbol{t}$), and the "roughness" of any $g \in G$ is measured by the quantity $\int (g'')^2$. Expanding g in a basis, it is not hard to see that

$$\int (g'')^2 = \boldsymbol{\beta}' A \boldsymbol{\beta}, \quad \text{where} \quad A_{ij} = \int B_i''(x) B_j''(x) dx \quad \text{for } 1 \leq i, j \leq J. \tag{10}$$

The traditional smoothing spline fit maximizes the penalized likelihood

$$\text{argmax}_{\boldsymbol{\beta}} \{ l(\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}' A \boldsymbol{\beta} \},$$

for some parameter λ. Silverman (1985) observes that the solution to this problem can be viewed as a posterior mode, where $\boldsymbol{\beta}$ is assigned a partially improper, normal prior having mean $\mathbf{0}$ and variance-covariance matrix $(\lambda A)^{-1}$. This setup has the favorable property that it is invariant to our choice of basis. This is desirable, as the choice of the basis will often be made for computational reasons.

In our simulations we will compare this smoothing prior to the scheme of Denison *et al.* (1998a) in which no stochastic structure is assigned to the coefficients $\boldsymbol{\beta}$ once G is selected. Instead, these authors employ maximum likelihood to make a deterministic choice of $\boldsymbol{\beta}$.

Markov chain Monte Carlo In order to treat a variety of estimation problems simultaneously, we have chosen the reversible jump MCMC scheme developed by Green (1995). Denison *et al.* (1998a) implement this technique in the context of general univariate and additive regression. We refer to these papers for the details of the scheme, and we instead focus on the type of moves that we need to implement the sampler. In general, we alternate (possibly at random) between the following moves.

- **Increase model dimension.** In this step, we introduce a new knot into an existing collection of breakpoints. Given the concavity properties of ELMs the change in the log-likelihood could either be computed exactly, or approximated using the appropriate Rao statistic. In our experiments we have computed the change in the log-likelihood exactly. The new knot is selected uniformly from among the set that yields an allowable space.

- **Decrease model dimension.** As with the greedy scheme, knots are deleted by imposing a constraint on one or more coefficients in the spline expansion. We can either evaluate the drop in the log-likelihood exactly, or through the Wald statistics. Any knot can be removed at any time (assuming we have more than $K_{\min}$ breakpoints to chose from).

- **Make structural changes to G that do not change dimension.** Unlike our standard greedy scheme, non-nested steps like moving a knot are now possible. Moving a knot from t_k to t_k^* technically involves deleting t_k and then inserting a new breakpoint at t_k^*. With smart initial conditions on the Newton-Raphson steps, we can calculate the change in the log-likelihood exactly and still maintain an efficient algorithm.

- **Update (possibly) g.** In a nonlinear model like Logspline, we can either apply a suitable approximation to the posterior and integrate with respect to the coefficients $\boldsymbol{\beta}$, or we can fold sampling them into our Markov chain.

Following Green (1995) and Denison *et al.* (1998a), we cycle between proposals for adding, deleting and moving knots, assigning these moves probabilities b_J, d_J and $1 - b_J - d_J$ (see Denison *et al.*, 1998a). New

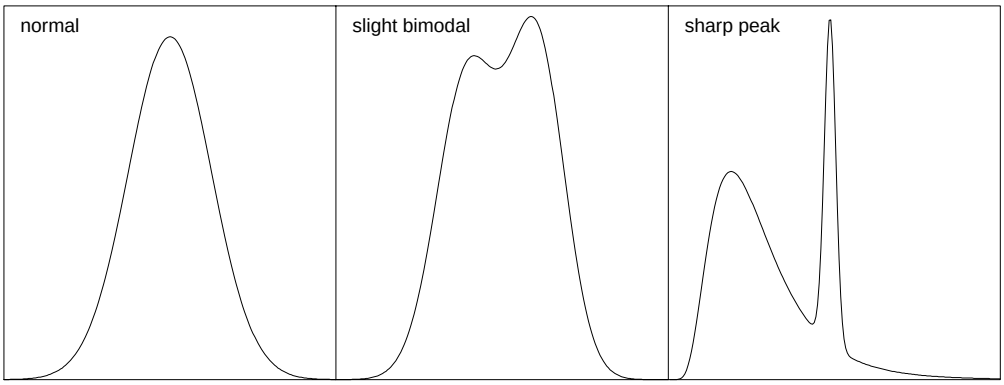


FIG. 1. *Densities used in the simulation study.*

knots can be positioned at any data point that is at least n_{sep} data points removed from one of the current knots. Subject to this constraint, knot addition follows a simple two step procedure. First, we select one of the intervals $(L, t_1), (t_1, t_2), \ldots, (t_K, U)$ uniformly at random (where the t_k are the current breakpoints). Within this interval, the candidate knot is then selected uniformly at random from one of the allowable data points. When moving a knot, we either propose a large move (in which a knot is first deleted, and then added using the addition scheme just described) or a small move (in which the knot is only moved within the interval between its two neighbors). Each of these two proposals have probability $(1 - d_J - b_J)/2$.

After each reversible jump step, we update the coefficients $\boldsymbol{\beta}$. To do this, we use the fact that for a given set of knots, we have a parametric model, and that the posterior distribution of $\boldsymbol{\beta}$ given G, K and the data is thus approximately multivariate normal with covariance matrix $\Sigma = (\lambda A + H)^{-1}$, and mean $\Sigma H \widehat{\boldsymbol{\beta}}$, where $\widehat{\boldsymbol{\beta}}$ is the maximum likelihood estimate of $\boldsymbol{\beta}$ in G, and H is the Hessian of the log-likelihood function at $\widehat{\boldsymbol{\beta}}$. An observation from this distribution is used as a proposal in a Metropolis step. Because we are using (partially improper) smoothing priors, the acceptance ratio for this proposal is formally undetermined (recall that the prior covariance matrices are degenerate). We solve this problem by "canceling" the zero eigenvalue in the numerator and the denominator (see also Besag and Higdon, 1999).

2.2 A simulation study

To compare the performance of the various possible implementations of Logspline density model selection procedures, we carried out a simulation study. We generated data from three densities:

normal the standard normal density;

slight bimodal $f(y) = 0.5 * f_Z(y; 1.25, 1) + 0.5 * f_Z(y; -1.25, 1.1)$, where $f_Z(y; \mu, \sigma)$ is the normal density with mean μ and standard deviation σ;

sharp peak $f(y) = 0.8 * g(y) + 0.2 * f)Z(y; 2, .07)$, where $g(Y)$ is the density of the lognormal random variable $Y = \exp(Z/2)$ and Z has a standard normal distribution.

These three densities are displayed in Figure 1. From each we generated 100 independent samples of size $n = 50$, 200, 1000, and 10000. We applied a variety of Logspline methods, see Table 1. For all the Bayesian

TABLE 1

Versions of Logspline Density Estimation used in the simulation study.

	model size	parameters
(i)	greedy optimization of AIC proposed by Stone *et al.* (1997)	
(ii)	simulated annealing optimization of AIC, termed SALSA for Simulated Annealing LogSpline Approximation	
(iii)	geometric	ML
(iv)	Poisson (5)	ML
(v)	uniform	$\lambda = 1/n$
(vi)	uniform	$\lambda = 1/\sqrt{n}$
(vii)	uniform	$\lambda = 1$
(viii)	geometric	$\lambda = 1/n$

methods we estimated the posterior mean by a simple pointwise average of the MCMC samples. Otherwise, the Bayesian approaches differ in two aspects:

- The prior on the model size: we used the geometric prior with parameter $p = 1 - 1/\sqrt{n}$, the Poisson prior with parameter 5, and a uniform prior; and
- Parameter estimates $\widehat{\boldsymbol{\beta}}$: we took either the maximum likelihood (ML) estimate, or we assigned a multivariate normal prior to $\boldsymbol{\beta}$ (for one of several choices for λ).

Table 1 summarizes the versions of Logspline which are reported here.

For simulated annealing (ii) (termed SALSA for "Simulated Annealing LogSpline Approximation") we ran the same MCMC iterations as for version (iii), but rather than selecting the mean of the sampled densities, we chose the density which minimizes AIC. As described above this is very similar to taking the density with the largest a posteriori probability (the mode), except that we ignore the prior on knot locations given the number of knots, K. This would have changed the penalty in the AIC criterion from $K \log n$ to $K \log n + \frac{1}{2} \log \binom{n}{K}$. Since version (ii) begins with the fit obtained by the greedy search (i), it is guaranteed to improve as far as AIC is concerned. Version (iii) uses the same penalty structure as version (ii), but averages over MCMC samples. Version (iv) is included since a Poisson (5) prior was proposed by Denison *et al.* (1998a). It applies a considerably smaller penalty on model size. Versions (v)–(viii) experiment with penalties on the coefficients. (After trying several values for λ, we decided that $1/n$ seemed reasonable.) Generating the parameters using a multivariate normal prior distribution implies smoothing with a AIC-like penalty. As such, we would expect that using $\lambda = 1/n$ with a uniform prior (version v) may give reasonable results, but that using a geometric prior (version ix) would smooth too much. Choosing λ too large, as in versions (vi)–(vii), leads to oversmoothing, while choosing λ too small tends to produce overly wiggly fits.

For versions (iii) and (iv) we ran 600 MCMC iterations, of which we discarded the first 100 as burn-in. Some simple diagnostics (not reported) suggest that after 100 iterations the chain is properly mixed. For versions (v)–(viii) each structural change was followed by an update of the coefficients $\boldsymbol{\beta}$.

In Table 2, we report ratios of integrated squared errors between the greedy scheme and the other methods outlined above. In addition, we feel that it is at least as important for a density estimate to provide the correct general "shape" of a density as to have a low integrated squared error. To capture the shape of our

TABLE 2

Mean Integrated Squared Error (MISE) for the simulation study.

	version	(i)	(ii)	(iii)	(iv)	(v)	(vi)	(vii)	(viii)
distribution	n	MISE	ratio of MISE over MISE of the greedy version (i)						
normal	50	0.02790	0.73	1.52	1.84	0.66	0.40	0.26	0.67
normal	200	0.01069	0.49	0.60	1.23	0.79	0.50	0.24	0.66
normal	1000	0.00209	0.59	0.58	1.33	0.87	0.90	0.42	0.73
normal	10000	0.00020	0.33	0.49	1.45	1.35	1.10	0.80	0.87
slight bimodal	50	0.02502	0.88	1.09	1.34	0.48	0.36	0.36	0.50
slight bimodal	200	0.00770	0.80	0.61	1.14	0.70	0.38	0.46	0.61
slight bimodal	1000	0.00164	0.57	0.60	1.13	0.89	0.66	0.40	0.77
slight bimodal	10000	0.00020	0.77	0.61	0.88	0.71	0.82	0.51	0.84
sharp peak	50	0.15226	0.97	0.78	0.81	0.68	0.90	1.12	0.72
sharp peak	200	0.03704	0.89	0.75	0.94	0.93	2.02	3.62	1.13
sharp peak	1000	0.00973	0.81	0.67	0.81	0.67	2.01	8.90	0.74
sharp peak	10000	0.00150	0.72	0.57	0.57	0.64	0.58	21.43	0.76
	average	1.00	0.71	0.74	1.12	0.78	0.89	3.21	0.75

estimates, we counted the number of times that a scheme produced densities having too few, too many and the correct number of modes. These results are summarized in Tables 3 and 4. Table 5 calculates the "total" lines of Tables 3 and 4. Note that for simulations of a normal distribution it is not possible for an estimate to have too few modes.

From Table 2 we note that most methods show a moderate overall improvement over the greedy version of Logspline, except for (vii). This scheme oversmoothes the data, so that the details (like the mode in the sharp peaked distribution) are frequently missed. We note that version (ii), choosing the mode of a Bayesian approach, is the only version that outperforms the greedy version for all 12 simulation set-ups. Otherwise, the difference between versions (ii), (iii), and (viii) seems to be minimal. In particular, if we had chosen another set of results than those for (i) to normalize by, the order of the average MISE for these four methods was often changed.

From Table 3 we note that version (vii), and to a lesser extent (ii) and (vi), have trouble with the slight bimodal density, preferring a model with just one peak. Versions (vi) and (vii) find too few modes, leading us to conclude that λ should be chosen smaller than $1/\sqrt{n}$ when using a uniform prior on model size. On the other hand, the Poisson prior leads to models exhibiting too many peaks, as do versions (iii) and (v).

Overall, it appears that the greedy, stepwise search is not too bad. It is several orders of magnitude faster than any of the other methods. The greedy approach, as well as SALSA have the advantage that the final model is again a Logspline density, which can be stored for later use. For the other methods, we must record the posterior mean at a number points. This has the potential of complicating later uses of our estimate. Among the Bayesian versions that employ ML estimates, version (iii) seems to perform best overall, while among those that put a prior on the coefficient vector, versions (v) and (viii) (both of which set $\lambda = 1/n$) are best. It is somewhat surprising that version (viii) performs so well, since it effectively imposes twice the AIC penalty on model size: one coming from the geometric prior, and one from the normal prior on the

TABLE 3

Number of times out of 100 simulations that a Logspline density estimate had too few modes.

	Version	(i)	(ii)	(iii)	(iv)	(v)	(vi)	(vii)	(viii)
distribution	n								
slight bimodal	50	45	52	4	0	21	74	99	31
slight bimodal	200	6	22	13	0	1	18	96	19
slight bimodal	1000	5	17	19	0	7	6	45	16
slight bimodal	10000	4	12	4	1	3	4	2	10
sharp peak	50	24	38	1	0	9	56	99	13
sharp peak	200	0	1	0	0	0	0	89	1
sharp peak	1000	0	0	0	0	0	0	0	0
sharp peak	10000	0	0	0	0	0	0	0	0
	total	84	142	41	1	41	158	430	90

parameters. Kooperberg and Stone (1992) argue that the Logspline method is not very sensitive to the exact value of the parameter, possibly explaining the behavior of version (viii). In Kooperberg and Stone (2001) a double penalty is also employed in the context of free knot Logspline density estimation.

2.3 Income data.

We applied the nine versions of Logspline used for the simulation study to the income data discussed in Stone *et al.* (1997), and the results are displayed in Figure 2. For the computations on the income data we ran the MCMC chain for 5000 iterations in which a new model was proposed, after discarding the first 500 iterations for burn-in. For the versions with priors on the parameters we alternated these iterations with updates of the parameters. The estimates for versions (ii), which was indistinguishable from version (iii), and versions (viii) which was indistinguishable from version (v) are not shown. In Kooperberg and Stone (1992) it was argued that the height of the peak should be at least about 1. Thus, it appears that versions (vi) and (vii) have oversmoothed the peak. On the other hand, version (iv) seems to have too many small peaks.

It is interesting to compare the number of knots for the various schemes. The greedy estimate (version i) has 8 knots, and the simulated annealing estimate (version ii) has 7 knots. The Bayesian versions (iii), (v) and (viii) have an average number of knots between 5 and 8, while the three versions that produced unsatisfactory results (iv, vii, and viii) have an average number of knots between 14 and 17.

The MCMC iterations can also give us information about the uncertainty in the knot locations. To study this further, we ran a chain for version (iii) with 500,000 iterations. Since the knots are highly correlated from one iteration to the next (at most one knot moves at each step), we only considered every 250th iteration. The autocorrelation function of the fitted log-likelihood suggested that this was well beyond the time over which iterations are correlated. This yielded 2000 sets of knot locations: 1128 with five knots, 783 with six knots, 84 with seven knots, and 5 with eight knots. When there were five knots, the first three were always located close to the mode, the fourth one was virtually always between 0.5 and 1.25, and the last knot between 1 and 2. The locations of the first three knots overlap considerably. When there are six knots, the extra knot can either be a fourth knot in the peak, or it is beyond the fifth knot.

TABLE 4

Number of times out of 100 simulations that a Logspline density estimate had too many modes.

	Version	(i)	(ii)	(iii)	(iv)	(v)	(vi)	(vii)	(viii)
distribution	n								
normal	50	18	11	94	100	49	5	0	28
normal	200	34	9	38	100	81	21	0	24
normal	1000	26	4	15	91	68	54	32	32
normal	10000	4	1	7	61	31	29	1	17
slight bimodal	50	4	1	84	99	6	0	0	4
slight bimodal	200	16	1	19	99	55	4	0	5
slight bimodal	1000	15	1	13	93	51	31	1	17
slight bimodal	10000	6	1	8	68	33	39	0	6
sharp peak	50	15	8	90	93	3	1	0	2
sharp peak	200	36	19	46	94	43	5	0	5
sharp peak	1000	28	14	30	77	32	12	1	9
sharp peak	10000	25	12	15	31	20	30	11	7
	total	227	82	459	1006	472	231	46	156

TABLE 5

Number of times that a Logspline density estimate had an incorrect number of modes.

	(i)	(ii)	(iii)	(iv)	(v)	(vi)	(vii)	(viii)
too few modes	84	142	41	1	41	158	430	90
too many modes	227	82	459	1006	472	231	46	156
total	311	224	500	1007	513	389	476	246

3. Triogram regression

When estimating a univariate function ϕ, our "pieces" in a piecewise polynomial model were intervals of the form (t_k, t_{k+1}). Through knot selection, we adjusted these intervals to capture the major features in ϕ. When ϕ is a function of two variables, we have more freedom in how we define a piecewise polynomial model. In this section we take our separate pieces to be triangles in the plane, and consider data-drive techniques that adapt these pieces to best fit ϕ. Our starting point is the Triogram methodology of Hansen *et al.* (1998) which employs continuous, piecewise linear (planar) bivariate splines. Triograms are based on a greedy, stepwise algorithm that builds on the ideas in Section 1 and can be applied in the context of any ELM where ϕ is a function of two variables. After reviewing some notation, we present a Bayesian version of Triograms for ordinary regression. An alternative approach to piecewise linear modeling was proposed in Brieman (1993) and given a Bayesian extension in Holmes and Mallick (2001).

Let $\triangle$ be a collection of triangles δ (having disjoint interiors) that partition a bounded, polygonal region in the plane $\mathcal{X} = \cup_{\delta \in \triangle} \delta$. The set $\triangle$ is said to be a *triangulation* of $\mathcal{X}$. Furthermore, $\triangle$ is *conforming* if the nonempty intersection between pairs of triangles in the collection consists of either a single, shared vertex or an entire common edge. Let $\mathbf{v}_1, \ldots, \mathbf{v}_K$ represent the collection of (unique) vertices of the triangles in $\triangle$.

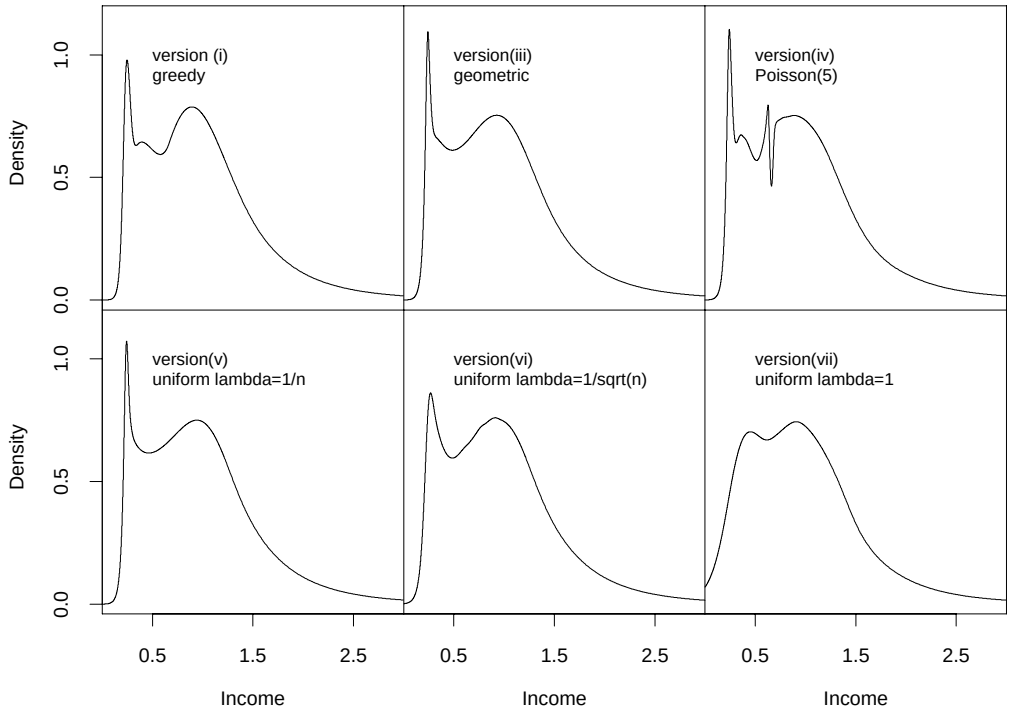


FIG. 2. *Logspline density estimates for the income data.*

Over $\mathcal{X}$, we consider the collection G of continuous, piecewise linear functions which are allowed to break (or hinge) along the edges in $\triangle$. It is not hard to show that G is a linear space having dimension equal to the number of vertices K. A simple basis composed of "tent functions" was derived in Courant (1943): For each $j = 1, \ldots, K$, we define $B_j(\mathbf{x}; \triangle)$ to be the unique function that is linear on each of the triangles in $\triangle$ and takes on the value 1 at $\mathbf{v}_j$ and 0 at the remaining vertices in the triangulation. It is not hard to show that $B_1(\mathbf{x}; \triangle), \ldots, B_K(\mathbf{x}; \triangle)$ is a basis for G. Also notice that each function $B_j(\mathbf{x}; \triangle)$ is associated with a single vertex $\mathbf{v}_j$, and in fact each $g \in G$

$$g(\mathbf{x}; \boldsymbol{\beta}, \triangle) = \sum_{i=1}^{J} \beta_i B_i(\mathbf{x}; \triangle), \tag{11}$$

interpolates the coefficients $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_K)$ at the points $\mathbf{v}_1, \ldots, \mathbf{v}_K$.

We now apply the space of linear splines to estimate an unknown regression function. In the notation of an ELM, we let $W = (\mathbf{X}, Y)$, where $\mathbf{X} \in \mathcal{X}$ is a two-dimensional predictor and Y is a univariate response. We are interested in exploring the dependence of Y on $\mathbf{X}$ by estimating the regression function $\phi(\mathbf{x}) = E(Y|\mathbf{X} = \mathbf{x})$. Given a triangulation $\triangle$, we employ linear splines over $\triangle$ of the form (11). For a collection of (possibly random) design points $\mathbf{X}_1, \ldots, \mathbf{X}_n$ taken from $\mathcal{X}$ and corresponding observations $Y_1, \ldots, Y_n$, we apply ordinary least squares to estimate $\boldsymbol{\beta}$. That is, we take $\widehat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} \sum_i [Y_i - g(\mathbf{X}_i; \boldsymbol{\beta}, \triangle)]^2$, and use $\widehat{g}(\mathbf{x}) = g(\mathbf{x}; \widehat{\boldsymbol{\beta}}, \triangle)$ as an estimate for ϕ.

As with the univariate spline models, we now consider stepwise alterations to the space G. Following Hansen *et al.* (1998), the one-to-one correspondence between vertices and the "tent" basis functions suggests a direct implementation of the greedy schemes in Section 1. Stepwise addition involves introducing a new vertex into an existing triangulation, thereby adding one new basis function to the original spline space. This

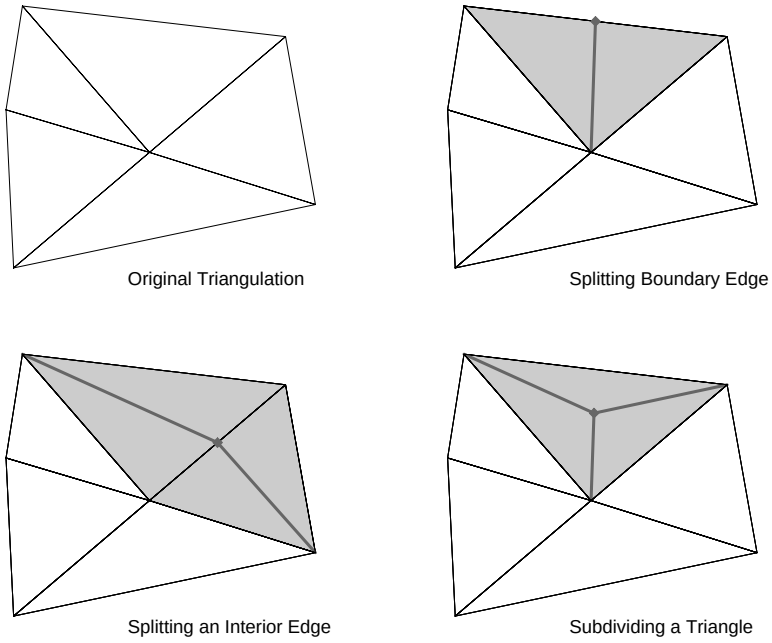


FIG. 3. *Three "moves" that add a new vertex to an existing triangulation. Each addition represents the introduction of a single basis function, the support of which is colored gray.*

operation requires a rule for connecting the new point to the vertices in $\triangle$ so that the new mesh is again a conforming triangulation. In Figure 3, we illustrate three options for vertex addition: we can place a new vertex on either a boundary or an interior edge, splitting the edge, or we an add a point to the interior of one of the triangles in $\triangle$. Given a triangulation $\triangle$, candidate vertices are selected from a regular triangular grid in each of the existing triangles, as well as a number of locations on each of the existing edges (for details see Hansen *et al.*, 1998). We impose constraints on our search by limiting, say, the area of the triangles in a mesh, their aspect ratio, or perhaps the number of data points they contain. As with Logspline, spaces satisfying these restrictions are referred to as allowable. At each step in the addition process, we select from the set of candidate vertices (that result in an allowable space), the point that maximizes the decrease in residual sum of squares when the Triogram model (11) is fitted to sample data. (In regression, the Rao and Wald statistics are the same and reduce to the change in the residual sum of squares between two nested models.)

Deleting a knot from an existing triangulation can be accomplished most easily by simply reversing one of the steps in Figure 3. Observe that removing a vertex in one of these three settings is equivalent to enforcing continuity of the first partial derivatives across any of the grey edges in this figure. Such continuity conditions are simple linear constraint on the coefficients of the fitted model, allowing us to once again apply a Wald test to evaluate the rise in the residual sum of squares after the vertex is deleted.

3.1 A Bayesian framework

Priors on model space As with univariate spline models, a prior on the space of Triograms is most easily defined by first specifying the structure of the approximation space, which in this case is a triangulation $\triangle$.

For any $\triangle$, we need to select the number of vertices K, their placement $\mathbf{v}$, and the triangles that connect them. Each set $\mathbf{v}$ can be joined by a number of different triangulations (assuming $\mathbf{v}$ has more than 3 points). Sibson (1978) shows that by starting from one triangulation of $\mathbf{v}$, we can generate any other by a sequence of "edge swaps." (This operation is given in Figure 4 and will come up later when we discuss MCMC for bivariate splines.) Unfortunately, a closed-form expression for the number of triangulations associated with a given set of vertices does not exist. Computing this number for even moderately sized configurations is difficult because two sets each with K vertices can have different numbers of triangulations.

To see how this complicates matters, suppose we follow the strategy for Logspline and propose a hierarchical prior of the form

$$p(\triangle|\mathbf{v}, K)\, p(\mathbf{v}|K)\, p(K), \tag{12}$$

where $\triangle$ is a triangulation of the vertices $\mathbf{v} = \{\mathbf{v}_1, \ldots, \mathbf{v}_K\}$. Assigning any proper distribution to $\triangle$ given $\mathbf{v}$ introduces a normalizing constant in $p(\triangle|\mathbf{v}, K)$ that involves enumerating the different triangulations of $\mathbf{v}$. Therefore, when taking ratios of (12) for two different sets of vertices, we are usually left with a prohibitively expensive computational problem. MCMC methods for exploring the model space are not possible.

To avoid this problem, we will use a tractable prior on triangulations developed by Nicholls (1998). This distribution depends on a pair of Poisson point processes, one that generates vertices on the interior of $\mathcal{X}$ and one for the boundary. As constructed, there is one parameter β that controls the intensity of this process, where larger values of β produce triangulations with more vertices. Nicholls (1998) avoids counting triangulations by normalizing across all triangulations obtainable from all vertex sets generated by this point process, and produces a distribution $p(\triangle)$. Bounds on the number of triangulations obtainable from a given vertex set are used to show that this kind of normalization is possible. This construction also has the advantage that restrictions on the size and shape of triangles are easily enforced and only change the (global) normalization constant in $p(\triangle)$. In our experiments, we set β so that the expected number of vertices for this base process is 5. We then adapted Nicholls' approach, so that the underlying point process produces a geometric (with parameter $1 - 1/\sqrt{n}$) or a uniform (on $K_{\min}, \ldots, K_{\max}$) number of vertices, following the simulation setup in the previous section.

Priors on splines in a given space Unlike the Logspline example, we do not have a single obvious choice for the smoothing prior for linear splines $g \in G$ defined relative to a triangulation $\triangle$. Dyn *et al.* (1990ab) propose several criteria of the form

$$\sum_e s^2(g, e) \quad \text{for} \quad g \in G\,,$$

where the summation is over all edges in $\triangle$. Their cost function $s(g, e)$ evaluates the behavior of g along an edge, assigning greater weight when the hinged linear pieces are farther from a single plane. Koenker and Mizera (2001) elegantly motivate a cost function $s(g, e) = \| \bigtriangledown g_e^+ - \bigtriangledown g_e^- \| \cdot \|e\|$, where $\bigtriangledown g_e^+$ and $\bigtriangledown g_e^-$ are the gradients of g computed over the triangles that share the common edge e having length $\|e\|$. This is similar to the approach taken by Nicholls (1998) who derived an edge-based smoothness penalty for piecewise constant functions defined over triangulations.

We choose to work with the cost function of Koenker and Mizera (2001). It is not hard to show that this gives rise to a quadratic penalty on the coefficient vector $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_K)$ which can be written $\boldsymbol{\beta}^t A \boldsymbol{\beta}$ for a positive-semidefinite matrix A. Since constant and linear functions have zero roughness by this measure, A

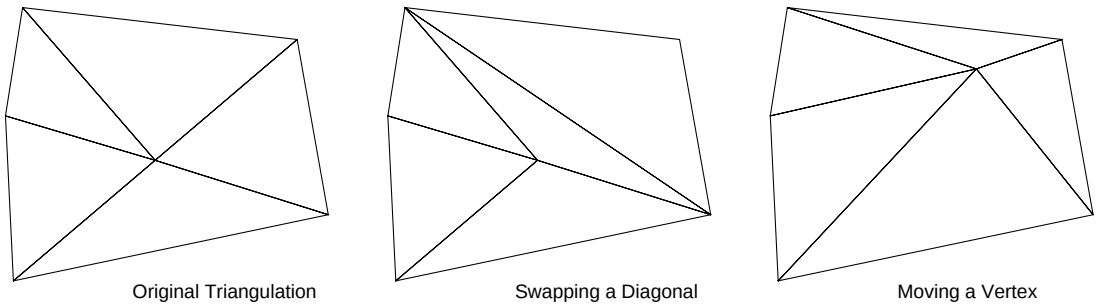


Fig. 4. Additional structural moves for the reversible jump MCMC scheme. Note that these two proposals result in a non-nested sequence of spaces.

has two zero eigenvalues. As was done for Logspline, we use A to generate a partially improper normal prior on $\boldsymbol{\beta}$ (with prior variance $\lambda\sigma^2$, where σ^2 is the error variance). Following Denison *et al.* (1998a), we assign a proper, inverse-gamma distribution to σ, and experiment with various fixed choices for λ that depend on sample size.

Markov chain Monte Carlo Our approach to MCMC for Triograms is similar to that with Logspline except that we need to augment our set of structural changes to $\triangle$ to include more moves than simple vertex addition and deletion. In Figure 4, we present two additional moves that maintain the dimension of the space G but change its structure. The middle panel illustrates swapping an edge, an operation that we have already noted is capable of generating all triangulations of a given vertex set $\mathbf{v}$. Quak and Schumaker (1991) use random swaps of this kind to come up with a good triangulation for a fixed set of vertices. In in the final panel of Figure 4, we demonstrate moving a vertex inside the union of triangles that contain it. These changes to $\triangle$ are non-nested in the sense that they produce spline spaces that do not differ by the presence or absence of a single basis function. For Triograms, the notion of an allowable space can appear through size or aspect ratio restrictions on the triangulations, and serves to limit the region in which we can place new vertices or to which we can move existing vertices. For example, given a triangle, the set into which we can insert a new vertex and still maintain a minimum area condition is a subtriangle, easily computable in terms of barycentric coordinates (see Hansen et al. 1998). As with Logspline, we alternate between these structural moves and updating the model parameters. Because we are working with regression, we can integrate out $\boldsymbol{\beta}$ and only have to update σ^2 at each pass. This approach allows us to focus on structural changes as was done by Smith and Kohn (1996) for univariate regression.

3.2 Simulations

In Figure 5, we present a series of three fits to a simulated surface plotted in the upper lefthand corner. A data set consisting of 100 observations was generated by first sampling 100 design points uniformly in the unit square. The actual surface is described by the function

$$f(\mathbf{x}) \quad = \quad \frac{40\exp\{8[(x_1-0.5)^2+(x_2-0.5)^2]\}}{\exp\{8[(x_1-0.2)^2+(x_2-0.7)^2]\}+\exp\{8[(x_1-0.7)^2+(x_2-0.2)^2]\}},$$

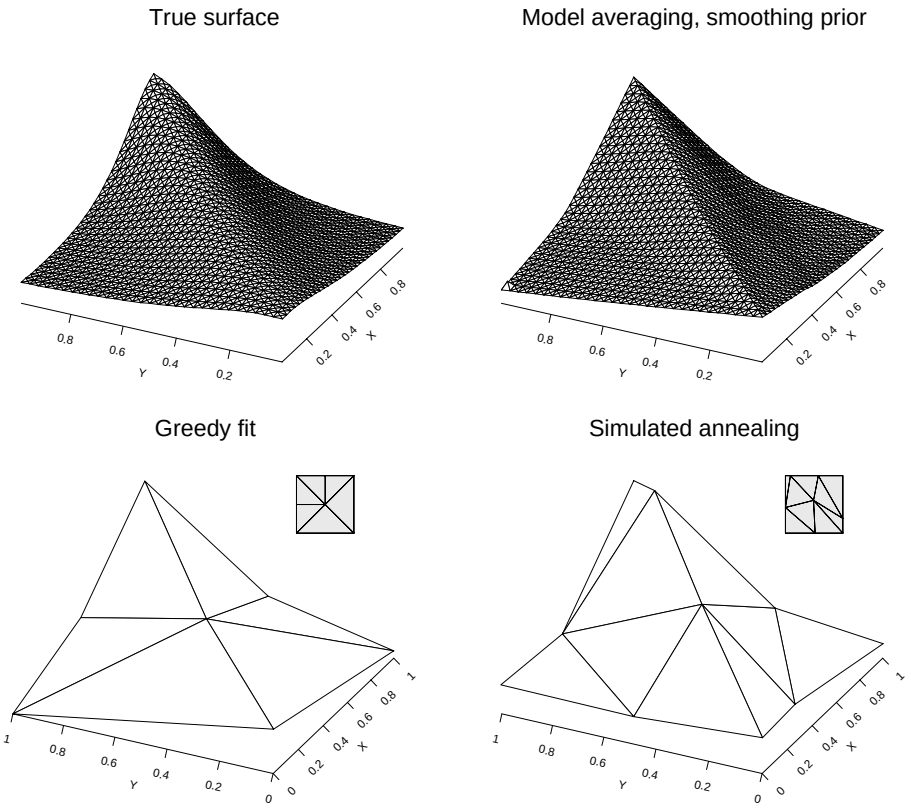


Fig. 5. In the top row we have the true surface (left) and the fit resulting from model averaging (right). In the bottom row we have two isolated fits, each a "minimal" BIC model, the leftmost coming from a greedy search, and the rightmost produced by simulated annealing (the triangulations appear at the top of each panel).

to which we add standard Gaussian errors. This function first appeared in Gu *et al.* (1990), and it will be hereafter referred to as simply GBCW. The signal-to-noise ratio in this setup is about 3. In the lower lefthand panel in Figure 5, we present the result of applying the greedy, Triogram algorithm. As is typical, the procedure has found a fairly regular, low-dimensional mesh describing the surface (the MISE is 0.31). For the fit plotted in the lower righthand panel, we employed a simulated annealing scheme similar to that described for Logspline. The geometric prior for $\triangle$ is used to guide the sampler through triangulations, and in each corresponding spline space G we consider $\widehat{g}$, the MLE (or in this case the ordinary least squares fit). In this way, the objective function matches that of the greedy search, the generalized AIC criterion (8). The scheme alternates between (randomly selected) structural changes (edge swaps and vertex moves, additions and deletions) and updating the estimate $\widehat{\sigma}^2$ of the noise variance. After 6,000 iterations, the sampler has managed to find a less regular, and marginally poorer-fitting model (the MISE is 0.32). In the context of triangulations, the greedy search is subject to a certain regularity that prevents configurations like the one in Figure 5. We can recapture this in the MCMC simulations either by placing restrictions on the triangulations in each mesh (say, imposing a smallest allowable size or aspect ratio) or by increasing the penalty on dimension, specified through our geometric prior.

In the last panel, we present the result of model averaging using a uniform prior on model size and a

TABLE 6.
Versions of Triogram used in the simulation study.

	model size	parameters
(i)	greedy optimization of BIC	
(ii)	simulated annealing optimization of BIC	
(iii)	Poisson (5)	ML
(iv)	geometric	ML
(v)	uniform	$\lambda = 1/n$

smoothing prior on the coefficients ($\lambda = 1/n$). The sampler is run for a total of 6,000 iterations, of which 1,000 are discarded as burn-in. We then estimate the posterior mode as a pointwise average of the sampled surfaces. The final fit is smoother in part because we are combining many piecewise-planar surfaces. We still see sharp effects, however, where features like the central ridge are present. The model in the lower righthand panel is not unlike the surfaces visited by this chain. As spaces G are generated, the central spine (along the line $y = x$) of this surface is always present. The same is true for the hinged portions of the surface along the lines $x = 0$ and $y = 0$. With these caveats in mind, the MISE of the averaged surface is about half of the other two estimates (0.15).

TABLE 7.
Mean Integrated Squared Error (MISE) for two smooth test functions.

	version	(i)	(ii)	(iii)	(iv)	(v)
distribution	n	MISE	ratio of MISE over (i)			
GBCW (high snr)	100	0.31	1.35	0.85	0.78	0.77
GBCW (high snr)	500	0.10	1.0	0.64	0.76	0.80
GBCW (high snr)	1000	0.08	0.91	0.82	0.94	0.79
Exp (low snr)	100	0.15	0.90	0.52	0.51	0.49
Exp (low snr)	500	0.04	0.85	0.46	0.50	0.47
Exp (low snr)	1000	0.03	0.51	0.32	0.40	0.46

We repeated these simulations for several sample sizes, taking $n = 100$, 500 and 1000 (100 repetitions for each value of n). In Table 6, we present several variations in the prior specification and search procedure. In addition to GBCW, we also borrow a test function from Breiman (1991), which we will refer to as Exp. Here, points $\mathbf{X} = (X_1, X_2)$ are selected uniformly from the square $[-1, 1]^2$. The response is given by $\exp(x_1 \sin(\pi x_2))$ to which normal noise is added ($\sigma = 0.5$). The signal-to-noise ratio in this setup is much lower, 0.9. The results are presented in Table 7. It seems reasonably clear that the simulated annealing approach can go very wrong, especially when the sample size is small. Again, this argues for the use of greater constraints in terms of allowable spaces when n is moderate. It seems that model averaging with the smoothing prior ($\lambda = 1/n$) and the Poisson prior of Denison *et al.* (1998a) perform the best. A closer examination of the fitted surfaces reveals the same kinds of secondary structure as we saw in Figure 5. To be sure, smoother basis functions would eliminate this behavior. It is not clear at present, however, if a different smoothing prior on the coefficients might serve to "unkink" these fits.

TABLE 8

Mean Integrated Squared Error (MISE) for two piecewise-planar test functions

	version	(i)	(ii)	(iii)	(iv)	(v)
distribution	n	MISE	ratio of MISE over (i)			
Model 1	50	0.16	0.97	0.70	0.35	0.80
Model 1	200	0.04	0.82	0.95	0.52	0.62
Model 1	1000	0.01	0.63	0.72	0.76	0.40
Model 3	50	0.70	1.40	0.86	0.51	0.50
Model 3	200	0.17	0.85	0.63	0.27	0.30
Model 3	1000	0.03	0.34	0.45	0.21	0.20

The performance of the Poisson (5) distribution is somewhat surprising. While for Logspline this choice led to undersmoothed densities, it would appear that the Triogram scheme benefits from slightly larger models. We believe that this is because of the bias involved in estimating a smooth function by a piecewise-linear surface. In general, these experiments indicate that tuning the Bayesian schemes in the context of a Triogram model is much more difficult than univariate set-ups. One comforting conclusion, however, is that essentially each of the schemes considered outperform the simple greedy search.

As a final test, we repeated the simulations from Hansen *et al.* (1998). We took as our trial functions two piecewise planar surfaces, one that the greedy scheme can jump to in a single move (Model 1), and one that requires several moves (Model 3). In this case, the model averaged fits (iv) were better than both simulated annealing and the greedy procedure. The estimate built from the Poisson prior tends to spend too much time in larger models, leading to its slightly poorer MISE results, while the geometric prior extracts a heavy price for stepping off of the "true" model. (Unlike the smooth cases examined above, the extra degrees of freedom do not help the Poisson scheme.) The simulations are summarized in Table 8. One message from this suite of simulations, therefore, is that a posterior mean does not over-smooth edges, and in fact preserves them better than the greedy alternatives.

4. Discussion

Early applications of splines were focused mainly on curve estimation. In recent years, these tools have proved effective for multivariate problems as well. By extending the concepts of "main effects" and "interactions" familiar in traditional $d-$way analysis of variance (ANOVA), techniques have been developed that produce so-called functional ANOVA's. Here, spline basis elements and their tensor products are used to construct the main effects and interactions, respectively. In these problems, one must determine which knot sequence to employ for each covariate, as well as what interactions are present.

In this paper we have discussed a general framework for adaptation in the context of an extended linear model. Traditionally, model-selection for these problems is accomplished through greedy, stepwise algorithms. While these approaches appear to perform reasonably well in practice, they visit a relatively small number of candidate configurations. By casting knot selection into a Bayesian framework, we have discussed an MCMC algorithm that samples many more promising models. We have examined various techniques for calibrating the prior specifications in this setup to more easily compare the greedy searches

and the MCMC schemes. An effective penalty on model size can be imposed either explicitly (through a prior distribution on dimension), or through the smoothness prior assigned to the coefficient vector. In general, we have demonstrated a gain in final mean squared error when appealing to the more elaborate sampling schemes.

We have also gone to great lengths to map out connections between this Bayesian method and other approaches to the knot placement problem. For example, a geometric prior distribution on model size, has a natural link to (stepwise) model selection with BIC, while we can choose a multivariate normal prior on the coefficients to connect us with the penalized likelihood methods employed in classical smoothing splines. In addition, the Bayesian formalism allows us to account for the uncertainty in both the structural aspects of our estimates (knot configurations and triangulations) as well as the coefficients in any given expansion. Model averaging in this context seems to provide improvement over simply selecting a single "optimal" model in terms of say BIC. The disadvantage of this approach is that we do not end up with a model based on one set of knots (or one triangulation).

While running our experiments, we quickly reached the conclusion that the priors play an important role: an inappropriate prior can easily lead to results that are much worse than the greedy algorithms. However, in our experiments we found out that when the priors are in the right ballpark, Bayesian procedures do perform somewhat better than greedy schemes in a mean squared error sense. This improvement in performance is larger for a relatively "unstable" procedures such as Triogram, while the improvement for a "stable" procedure such as Logspline is smaller.

For the Triogram methodology there is an additional effect of model averaging: the average of many piecewise planar surfaces will give the impression of being smoother. Whether this is an advantage or not probably depends on the individual user and her/his application: when we gave seminars about the original Triogram paper, there were people who saw the piecewise planar approach as a major strength and as a major weakness of the methodology.

Acknowledgements

Charles Kooperberg was supported in part by National Institutes of Health grant R29 CA 74841. The authors wish to thank Merlise Clyde, David Denison, Ed George, Peter Green, Robert Kohn and Bin Yu for many helpful discussions.

References

Akaike, H. (1974) A new look at the statistical model identification. *IEEE Trans. AC*, **19** 716–723.

Besag, J. and Higdon, D. (1999) Bayesian inference for agricultural field experiments (with discussion). *J. R. Statist. Soc.* B, **61**, 691–746.

Breiman, L. (1991) The Π-method for estimating multivariate functions from noisy data. *Technometrics*, **33**, 125–143.

Breiman, L. (1993) Hinging hyperplanes for regression, classification and function approximation, *IEEE Trans. Info. Theory*, **3**(3), 999–1013.

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

Courant, R. (1943) Variational methods for the solution of problems of equilibrium and vibrations. *Bull. Am. Math. Soc.*, **49**, 1–23.

de Boor, C. (1978) *A Practical Guide to Splines*. New York: Springer.

Denison, D. G. T., Mallick, B. K. and Smith, A. F. M. (1998a) Automatic Bayesian curve fitting. *J. R. Statist. Soc.* B, **60**, 333–350.

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

Dyn, N., Levin, D. and Rippa, S. (1990a) Data dependent triangulations for piecewise linear interpolation. *IMA J. Num. Anal.*, **10**, 137–154.

Dyn, N., Levin, D. and Rippa, S. (1990b) Algorithms for the construction of data dependent triangulations. *Alg. Approx.* (eds. J. C. Mason and M. G. Cox, M. G), vol II, pp. 185–192. New York: Chapman and Hall.

Friedman, J. H. (1991) Multivariate adaptive regression splines (with discussion). *Ann Statist*, **19**, 1–141.

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

Green, P. J. (1995) Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, **82**, 711–732.

Green, P. J. and Silverman, B. W. (1994) *Nonparametric Regression and Generalized Linear Models*. London: Chapman and Hall.

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

Halpern, E. F. (1973) Bayesian spline regression when the number of knots is unknown. *J. R. Statist. Soc.* B, **35**, 347–360.

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

Hansen, M., Kooperberg, C. and Sardy, S. (1998) Triogram Models. *J. Am. Statist. Assoc.*, **93**, 101–119.

Holmes, C.C. and Mallick, B.K. (2001) Bayesian regression with multivariate linear splines, *J. Roy. Statist. Soc. Series B.*, **63**, 3–18.

Huang, J. Z. (1998) Projection estimation in multiple regression with application to functional ANOVA models. *Ann. Statist.*, **26**, 242–272.

Huang, J. Z. (2001) Concave extended linear modeling: a theoretical synthesis. *Statistica Sinica*, **11**, 173–197.

Jupp, D. L. B. (1978). "Approximation to Data by Splines with Free Knots," *SIAM J. Num. Anal.*, **15**, 328–343.

Koenker, R. and Mizera, I. (2001) Penalized Triograms: Total variation regularization for bivariate smoothing, *Technical Report*.

Kooperberg, C., Bose, S. and Stone, C. J. (1997) Polychotomous Regression. *J. Am. Statist. Assoc.*, **92**, 117–127.

Kooperberg, C., and Stone, C. J. (1991) Logspline density estimation. *Comp. Statist. Data Anal.*, **12**, 327–347.

Kooperberg, C., and Stone, C. J. (1992) Logspline density estimation for censored data. *J. Comp. Graph. Statis.*, **1**, 301–328.

Kooperberg, C., and Stone, C. J. (2001) Logspline density with free knot splines. Manuscript.

Lindstrom, M (1999) Penalized estimation of free-knot splines. *J. Comp. Graph. Statis.*, **8**, 333–352.

Nicholls, G. (1998) Bayesian image analysis with Markov chain Monte Carlo and colored continuum triangulation models. *J. R. Statist. Soc.* B, in press.

Quak, E. and Schumaker, L. L. (1991) Least squares fitting by linear splines on data dependent triangulations. In *Curves and Surfaces* (eds. P.J. Laurent, A. Le Méhauté and L. L. Schumaker), pp. 387–390. San Diego: Academic Press.

Schumaker, L. L. (1993) *Spline Functions: Basic Theory.* New York: Wiley.

Schwarz, G. (1978) Estimating the dimension of a model. *Anni. Statist.*, **6**, 461–464.

Silverman, B. W. (1985) Some aspects of the spline smoothing approach to nonparametric regression curve fitting (with discussion). *J. R. Statist. Soc.* B, **47**, 1–52.

Sibson, R. (1978) Locally equiangular triangulations. *Computer Journal*, **21**, 243–245.

Smith, M. (1996) Nonparametric Regression: A Markov chain Monte Carlo Approach. Ph. D. Dissertation. Australia: University of New South Wales.

Smith, M. and Kohn, R. (1996) Nonparametric regression using Bayesian variable selection. *J. Econometrics*, **75**, 317–344.

Smith, M. and Kohn, R. (1998) Nonparametric estimation of irregular functions with independent or autocorrelated errors. In *Practical Nonparametric and Semiparametric Bayesian Statistics* (eds. D. Dey, P. Mller and D. Sinha), pp. 133–150. New York: Springer-Verlag.

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

Smith, P.L. (1982b) Hypothesis testing in B-spline regression. *Communications in Statistics, Part B–Simulation and Computation*, **11**, 143–157.

Stone, C. J. (1985) Additive regression and other nonparametric models. *Ann. Statist.*, **13**, 689–705.

Stone, C. J. (1994) The use of polynomial splines and their tensor products in multivariate function estimation (with discussion). *Ann. Statist.*, **22**, 118–184.

Stone, C. J., Hansen M., Kooperberg, C. and Truong, Y. K. (1997) Polynomial splines and their tensor products in extended linear modeling (with discussion). *Ann. Statist.* **25**, 1371–1470.

Stone, C. J. and Huang, J. Z. (2001) Free knot splines in concave extended linear modeling . *J. Statist. Plan. Inf.*, to appear.

Stone, C. J. and Koo, C.-Y. (1986) Logspline density estimation. *AMS Cont. Math. Ser.*, **59**, 1–15.

Wahba, G. (1990) *Spline Models for Observational Data.* Philadelphia: Society for Industrial and Applied Mathematics.