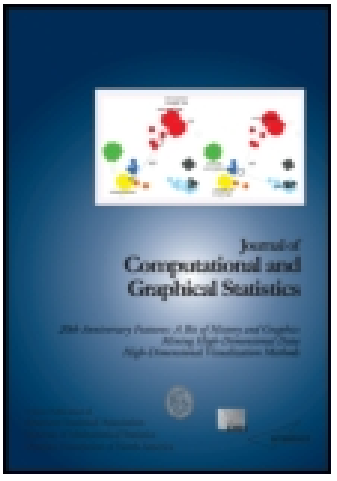

Journal of Computational and Graphical Statistics

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/ucgs20

Selecting the Number of Knots for Penalized Splines

David Ruppert[a]

[a] David Ruppert is Professor, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY 14853-3801

Published online: 01 Jan 2012.

To cite this article: David Ruppert (2002) Selecting the Number of Knots for Penalized Splines, Journal of Computational and Graphical Statistics, 11:4, 735-757, DOI: 10.1198/106186002853

To link to this article: http://dx.doi.org/10.1198/106186002853

Selecting the Number of Knots for Penalized Splines

David RUPPERT

Penalized splines, or P-splines, are regression splines fit by least-squares with a roughness penalty. P-splines have much in common with smoothing splines, but the type of penalty used with a P-spline is somewhat more general than for a smoothing spline. Also, the number and location of the knots of a P-spline is not fixed as with a smoothing spline. Generally, the knots of a P-spline are at fixed quantiles of the independent variable and the only tuning parameters to choose are the number of knots and the penalty parameter. In this article, the effects of the number of knots on the performance of P-splines are studied. Two algorithms are proposed for the automatic selection of the number of knots. The myopic algorithm stops when no improvement in the generalized cross-validation statistic (GCV) is noticed with the last increase in the number of knots. The full search examines all candidates in a fixed sequence of possible numbers of knots and chooses the candidate that minimizes GCV. The myopic algorithm works well in many cases but can stop prematurely. The full-search algorithm worked well in all examples examined. A Demmler–Reinsch type diagonalization for computing univariate and additive P-splines is described. The Demmler–Reinsch basis is not effective for smoothing splines because smoothing splines have too many knots. For P-splines, however, the Demmler–Reinsch basis is very useful for super-fast generalized cross-validation.

Key Words: Additive models; Full search; Myopic search; P-spline; Smoothing.

1. INTRODUCTION

In this article variants of smoothing splines are studied that will be called penalized splines or, following Eilers and Marx (1996), P-splines. The knots for a P-spline are generally on a grid of equally spaced sample quantiles and the only tuning parameters are the number of knots and the penalty parameter. This article discusses the choice of the number of knots and penalty parameters jointly by generalized cross-validation.

David Ruppert is Professor, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY 14853-3801 (E-mail: davidr@orie.cornell.edu).

Journal of Computational and Graphical Statistics, Volume 11, Number 4, Pages 735–757
DOI: 10.1198/106186002853

Suppose that one has data (x_i, y_i) where for now x_i is univariate,

$$y_i = m(x_i) + \epsilon_i, \tag{1.1}$$

m is a smooth function giving the conditional mean of y_i given x_i, and $\{\epsilon_i\}_{i=1}^n$ are independent, mean zero errors with a constant variance, σ^2. To estimate m one uses a regression spline model

$$m(x;\boldsymbol{\beta}) = \beta_0 + \beta_1 x + \cdots + \beta_p x^p + \sum_{k=1}^{K} \beta_{p+k}(x-\kappa_k)_+^p, \tag{1.2}$$

where $p \geq 1$ is an integer, $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_{p+K})^\top$ is a vector of regression coefficients, $(u)_+^p = u^p I(u \geq 0)$, and $\kappa_1 < \cdots < \kappa_K$ are fixed knots. It is not difficult to see that m given by (1.2) is a pth degree polynomial on each interval between two consecutive knots and has $p-1$ continuous derivatives everywhere. The pth derivative of m takes a jump of size b_k at the kth knot, κ_k. As discussed in Section 2.2, the power basis used in (1.1) could be replaced by another spline basis.

When fitting model (1.2) to noisy data, one must prevent overfitting which can cause near interpolation of the data. Methods for obtaining a smooth spline estimate include knot selection—for example, Friedman and Silverman (1989), Friedman (1991), and Stone, Hansen, Kooperberg, and Truong (1997)—and smoothing splines (Wahba 1990; Eubank 1988). With the first set of methods, the knots are selected from a set of candidate knots by a technique similar to stepwise regression and then, given the selected knots, the coefficients are estimated by ordinary least squares. Smoothing splines have a knot at each unique value of x and control overfitting by using least-squares estimation with a roughness penalty. The penalty is on the integral of the square of a specified derivative, usually the second. Luo and Wahba's (1997) hybrid adaptive spline (HAS) combines knots selection and a roughness penalty.

In this article a penalty approach is used that is similar to smoothing splines but with fewer knots and a somewhat more general roughness penalty. I allow K in (1.2) to be large but typically far less than n. Once K has been chosen, the knots are placed at fixed quantiles of the the $\{x_i\}$. Unlike knot-selection techniques, the penalty approach retains all these candidate knots. Our approach allows any penalty which is a quadratic function of the spline coefficients. However, in the numerical study I focus on a roughness penalty on the squares of the jumps in the pth derivative of $m(x;\boldsymbol{\beta})$. One could view this as a penalty on the $(p+1)$th derivative of $m(x;\boldsymbol{\beta})$ where that derivative is a generalized function. Eilers and Marx (1996) developed this method of "P-splines," though they traced the original idea to O'Sullivan (1986, 1988). Also, P-splines are low dimensional smoothers, that is, their smoother matrices have rank equal to $K+p+1$ which is typically far less than n, and thus P-splines are similar in spirit to the low-rank *pseudosplines* proposed by Hastie (1996).

Section 2 discusses penalized least-squares estimation of univariate P-splines and introduces an algorithm for generalized cross-validation (GCV) based on the Demmler–Reinsch basis. The Demmler–Reinsch basis is not effective for smoothing splines because smoothing splines have too many knots. For P-splines, however, the Demmler–Reinsch basis is very

useful for super-fast generalized cross-validation. Section 3 introduces two algorithms for selecting the number of knots. Section 4 presents some simulation results. The extension to additive models is discussed in Section 6, and Section 7 contains further discussion and a summary of the conclusions.

It has generally been believed that a P-spline can have too few knots but not too many knots. The reason for this belief is that once there are enough knots to fit features in the data, overfitting is controlled by the penalty. This belief is somewhat supported by the data, but I found examples where having too many knots degrades the performance of the spline estimator. This finding was unexpected.

2. THE PENALIZED LEAST-SQUARES ESTIMATOR

Define $\widehat{\boldsymbol{\beta}}(\alpha)$ to be the minimizer of

$$\sum_{i=1}^{n} \{y_i - m(x;\boldsymbol{\beta})\}^2 + \alpha \sum_{k=1}^{K} \beta_{p+k}^2, \tag{2.1}$$

where α is a smoothing parameter. The larger the value of α, the more the spline fit is shrunk towards a global polynomial fit where $\beta_{p+k} = 0$ for $k = 1, \ldots, K$. Selection of α by generalized cross-validation (GCV) will be discussed in the following. The penalty in (2.1) can be generalized to $\alpha \boldsymbol{\beta}^\top \mathbf{D} \boldsymbol{\beta}$, where $\mathbf{D}$ is any symmetric, nonnegative definite matrix; see below.

Let $\mathbf{Y} = (y_1, \ldots, y_n)^T$ and $\mathbf{X}$ be the "design matrix" for the regression spline so that the ith row of $\mathbf{X}$ is

$$\mathbf{X}_i = (\,1, \quad x_i, \quad \cdots \quad x_i^p, \quad (x_i - \kappa_1)_+^p, \quad \cdots \quad (x_i - \kappa_K)_+^p\,). \tag{2.2}$$

Also, let $\mathbf{D}$ be a diagonal matrix whose first $(1 + p)$ diagonal elements are 0 and whose remaining diagonal elements are 1. Then simple calculations show that $\widehat{\boldsymbol{\beta}}(\alpha)$ is given by

$$\widehat{\boldsymbol{\beta}}(\alpha) = \left(\mathbf{X}^T\mathbf{X} + \alpha\mathbf{D}\right)^{-1} \mathbf{X}^T\mathbf{Y}. \tag{2.3}$$

This is a ridge regression estimator that shrinks the regression spline towards the least-squares fit to a pth degree polynomial model (Hastie and Tibshirani 1990, sec. 9.3.6).

Computing (2.3) is extremely quick, even for a relatively large number, say 100, values of α, especially if one uses the diagonalization algorithm discussed Section 2.1. For a fixed number of knots, K, the computational time for the matrices $\mathbf{X}^T\mathbf{X}$ and $\mathbf{X}^T\mathbf{Y}$ is linear in the sample size n, but these matrices need only be computed once. As Eilers and Marx (1996) mentioned, after these matrices are computed, only $K \times K$ matrices need to be manipulated. This allows rapid selection of α minimizing the GCV statistic when $\widehat{\boldsymbol{\beta}}(\alpha)$ is calculated over a grid of values of α.

Using a suitable value of α is crucial to obtaining a satisfactory curve estimate. Here I follow Hastie and Tibshirani (1990) closely. Let

$$\text{ASR}(\alpha) = n^{-1} \sum_{i=1}^{n} \left\{ y_i - m(X_i; \widehat{\boldsymbol{\beta}}\,(\alpha)) \right\}^2$$

be the average squared residuals using α. Let

$$\mathbf{S}(\alpha) = \mathbf{X}\left(\mathbf{X}^T\mathbf{X} + \alpha\mathbf{D}\right)^{-1}\mathbf{X}^T$$

be the "smoother" or "hat" matrix. Then

$$\text{GCV}(\alpha) = \frac{\text{ASR}(\alpha)}{\left[1 - \lambda n^{-1}\text{tr}\{\mathbf{S}(\alpha)\}\right]^2} \tag{2.4}$$

is the generalized cross-validation statistic. Here $\text{tr}\{\mathbf{S}(\alpha)\}$ is the "effective degrees of freedom" of the fit and λ is a tuning parameter; ordinary GCV uses $\lambda = 1$ but $\lambda > 1$ is used as an extra penalty for model searching in algorithms such as MARS. The algorithms studied in this article do *not* search across a large number of models and therefore use $\lambda = 1$. As mentioned in Section 4, some simulations with $\lambda = 1.5$ or 2 were tried but using $\lambda > 1$ did not improve mean squared errors.

One chooses α by computing $\text{GCV}(\alpha)$ for a grid of α values and choosing the minimizer of that criterion over the grid. As a default, I use a grid of 100 values of α such that the values of $\log_{10}(\alpha)$ are equally spaced between -10 and 12. (If GCV is minimized at either endpoint of this grid, then the grid should be expanded at that end.)

2.1 Diagonalization Algorithm

Computation can be sped up and numerically stabilized with the following diagonalization method that is a variation on the Demmler-Reinsch algorithm used to compute smoothing splines; see Eubank (1988) and Nychka (2000). Suppose that $\widehat{\boldsymbol{\beta}}$ is defined by

$$(\mathbf{X}^\top\mathbf{X} + \alpha\mathbf{D})\widehat{\boldsymbol{\beta}} = \mathbf{X}^\top\mathbf{Y}, \tag{2.5}$$

where $\mathbf{D}$ is a nonnegative definite symmetric matrix. ($\mathbf{D}$ need not be diagonal, though it would be for the penalty in (2.1).) Define $A^{-T} = (A^{-1})^\top$. Then the following result holds.

Theorem 1. *Let* $\mathbf{B}$ *be a square matrix satisfying* $\mathbf{B}^{-1}\mathbf{B}^{-\top} = \mathbf{X}^\top\mathbf{X}$*, for example,* $\mathbf{B}^{-1}$ *is a Cholesky factor of* $\mathbf{X}^\top\mathbf{X}$*. Let* $\mathbf{U}$ *be orthogonal and let* $\mathbf{C}$ *be diagonal such that* $\mathbf{UCU}^\top = \mathbf{BDB}^\top$.

Finally, define $\mathbf{Z} = \mathbf{X}(\mathbf{B}^\top\mathbf{U})$ *so that* $\mathbf{Z}^\top = \mathbf{U}^\top\mathbf{B}\mathbf{X}^\top$*, and let* $\widehat{\boldsymbol{\lambda}} = \mathbf{U}^\top\mathbf{B}^{-\top}\widehat{\boldsymbol{\beta}} = (\mathbf{B}^\top\mathbf{U})^{-1}\widehat{\boldsymbol{\beta}}$.

Then $\widehat{\boldsymbol{\lambda}}$ *solves the diagonal system*

$$(\mathbf{I} + \alpha\mathbf{C})\widehat{\boldsymbol{\lambda}} = \mathbf{Z}^\top\mathbf{Y} = (\mathbf{U}^\top\mathbf{B})\mathbf{X}^\top\mathbf{Y}. \tag{2.6}$$

Moreover, $\mathbf{X}\widehat{\boldsymbol{\beta}} = \mathbf{Z}\widehat{\boldsymbol{\lambda}}$ *so the hat matrix is* $\mathbf{S}(\alpha) = \mathbf{Z}(\mathbf{I} + \alpha\mathbf{C})^{-1}\mathbf{Z}^\top$ *and the degrees of freedom for the fit is*

$$\text{tr}\left\{\mathbf{S}(\alpha)\right\} = \text{tr}\left\{(\mathbf{I} + \alpha\mathbf{C})^{-1}\mathbf{Z}^\top\mathbf{Z}\right\} = \text{tr}(\mathbf{I} + \alpha\mathbf{C})^{-1} = \sum_i (1 + \alpha C_i)^{-1}, \tag{2.7}$$

where C_i *is the* i*th diagonal element of* $\mathbf{C}$.

The beauty of this method of calculating the P-Spline is that the work of calculating $\mathbf{B}$, $\mathbf{B}^{-1}$, $(\mathbf{U}, \mathbf{C})$ and $\mathbf{Z}$ needs to be done only once and then these quantities can be used for all values of α. For each value of α, $\widehat{\boldsymbol{\lambda}}$ is computed by solving the *diagonal* system (2.6) and then the spline fit is $\mathbf{Z}\widehat{\boldsymbol{\lambda}}$. Moreover, computing $\text{tr}(\mathbf{S}(\alpha))$ by (2.7) is also very fast. With 500 observations and 30 knots, computing the estimate for 100 values of α takes only 10% more time than for only one value of α.

The diagonalization algorithm takes advantage of the Cholesky algorithm that is numerically stable (Gill, Murray, and Wright 1981). However, the power basis functions are ill-conditioned,especially when the number of knots is close to the number of unique values of the x_i. The result is that with round-off errors, $\mathbf{X}^\top\mathbf{X}$ may not be numerically positive definite. To increase stability, I add $10^{-10}\mathbf{D}$ to $\mathbf{X}^\top\mathbf{X}$. The effect from an estimation viewpoint is quite small—by increasing the penalty from α to $\alpha + 10^{-10}$ the minimum penalty becomes 10^{-10} rather than 0.

Because $\mathbf{D}$ is not of full rank, $\mathbf{C}$ will contain some diagonal elements that are zero. Nonetheless, $\mathbf{U}$ is orthogonal and hence invertible.

2.2 OTHER BASES AND OTHER PENALTIES

The methodology in this article for selecting the number of knots is applicable to other bases, for example, B-splines. Also, other penalties such as quadratic integral penalties used in the smoothing spline literature can replace the ridge-type penalty in (2.1). Thus, one can generalize the spline model by assuming that

$$m(x;\boldsymbol{\beta}) = \sum_{j=0}^{p+K} B_j(x)\beta_j,$$

where $B_0(x), \ldots, B_{p+K}(x)$ is any basis for the vector spline of pth degree splines with knots $\kappa_1, \ldots, \kappa_K$. One could also use a basis for a subspace of such splines, for example, natural cubic splines. With this change of basis, $\mathbf{X}$ is the matrix whose i, jth element is $B_j(x_i)$.

The penalized least-squares estimator can be generalized as the minimizer of

$$\sum_{i=1}^{n} \{y_i - m(x;\boldsymbol{\beta})\}^2 + \alpha\boldsymbol{\beta}^\top\mathbf{D}\boldsymbol{\beta}, \tag{2.8}$$

where $\mathbf{D}$ is an appropriately chosen symmetric, nonnegative definite matrix. For example, if the penalty is

$$\int_{\min(x_i)}^{\max(x_i)} \left\{m''(x)\right\}^2 dx,$$

then the i, jth element of $\mathbf{D}$ would be

$$\int_{\min(x_i)}^{\max(x_i)} B_i''(x)\, B_j''(x)dx. \tag{2.9}$$

With these changes, the penalized least-squares estimator continues to be given by (2.5) and the diagonalization algorithm remains applicable.

With P-splines defined at this degree of generality, cubic smoothing splines become a special case of P-splines where the basis is that of natural cubic splines and $\mathbf{D}$ is defined by (2.9). Although, strictly speaking, smoothing splines require a knot at every unique value of the x_i, when the sample size is large some algorithms such as `smooth.spline` in S-Plus select a smaller set of knots as the default (Chambers and Hastie 1993). The results of this article shed light on this practice and the algorithms I propose could be applied to smoothing splines or B-splines with penalty (2.8).

To keep the article to a reasonable length, the numerical study includes only the power basis functions and the penalty in (2.1).

Proof of Theorem 1:

$$\mathbf{X}^\top\mathbf{X} + \alpha\mathbf{D} = \mathbf{B}^{-1}\mathbf{B}^{-\top} + \alpha\mathbf{D}$$
$$= \mathbf{B}^{-1}(\mathbf{I} + \alpha\mathbf{B}\mathbf{D}\mathbf{B}^\top)\mathbf{B}^{-\top} = \mathbf{B}^{-1}\mathbf{U}(\mathbf{I} + \alpha\mathbf{C})\mathbf{U}^\top\mathbf{B}^{-\top},$$

so that

$$\mathbf{U}^\top\mathbf{B}\left\{\mathbf{X}^\top\mathbf{X} + \alpha\mathbf{D}\right\}\mathbf{B}^\top\mathbf{U} = \mathbf{I} + \alpha\mathbf{C}.$$

Thus,

$$\mathbf{U}^\top\mathbf{B}\left\{(\mathbf{X}^\top\mathbf{X} + \alpha\mathbf{D})\mathbf{B}^\top\mathbf{U}(\mathbf{U}^\top\mathbf{B}^{-\top}\widehat{\boldsymbol{\beta}})\right\} = \mathbf{U}^\top\mathbf{B}(\mathbf{X}^\top\mathbf{Y})$$

or

$$(\mathbf{I} + \alpha\mathbf{C})\widehat{\boldsymbol{\lambda}} = \mathbf{Z}^\top\mathbf{Y}.$$

Also, $\mathbf{X} = \mathbf{Z}\mathbf{U}^\top\mathbf{B}^{-\top}$ so that

$$\mathbf{X}\widehat{\boldsymbol{\beta}} = \mathbf{Z}\mathbf{U}^\top\mathbf{B}^{-\top}\widehat{\boldsymbol{\beta}} = \mathbf{Z}\widehat{\boldsymbol{\lambda}}$$

by definition of $\widehat{\boldsymbol{\lambda}}$. The remainder of the proof is straightforward algebra. □

3. CHOOSING THE NUMBER OF KNOTS

Because smoothing is controlled by the penalty parameter, α, the number of knots, K, is not a crucial parameter. Monte Carlo evidence in Section 4 shows that there must be enough knots to fit features in the data, but after this minimum necessary number of knots has been reached, further increases in K often have little effect on the fit. However, there are examples where increasing K above a minimum necessary value increases the mean square error by a moderate amount; see Figures 1 and 3 where increasing K above five knots leads to as much as an 20%, respectively 33%, increase in MSE over the value at five knots and the myopic algorithm described below outperforms a fixed choice of 20 or more knots.

Thus, the first goal for any algorithm for selecting K is to make certain that K is sufficiently large to fit the data. The second goal is to choose K not so large that computation time is excessive or MSE is larger than necessary.

The first algorithm was proposed by Ruppert and Carroll (2000) for P-splines with a spatially adaptive penalty, but it has not yet been studied in much detail. Here it is applied to the global-penalty (not spatially adaptive) P-splines discussed in Section 2. A sequence of trial values of K is selected. The trial values are 5, 10, 20, 40, 80, and 120, except that only values of K in this sequence that are less than $n - p - 1$ are used, so that the number of parameters is less than the number of observations. (If there are repeats among the x_i, then n would be replaced by the number of unique values among the x_i.) The knots are at "equally spaced" sample quantiles of $\{x_i\}$. More precisely, the kth knot is the jth order statistic of $\{x_i\}_{i=1}^n$ where j is $nk/(n+1)$ rounded to the nearest integer.

The algorithm for selecting the number of knots is as follows. First, the P-spline fit is computed for K equal to 5 and 10. In each case α is chosen to minimize GCV(α) for that number of knots. If GCV at $K = 10$ is greater than .98 times GCV at $K = 5$, then one concludes that further increases in K are unlikely to decrease GCV and one uses $K = 5$ or 10, whichever has the smallest GCV. Otherwise, one computes the P-spline fit with $K = 20$ and compares GCV for $K = 10$ with GCV for $K = 20$ in the same way one compared GCV for $K = 5$ and 10. One stops and uses $K = 10$ or 20 (whichever gives the smaller GCV) if GCV at $K = 20$ exceeds .98 times GCV at $K = 10$. Otherwise, one computes the P-spline at $K = 40$, and so on. The algorithm is called " myopic" since it never looks beyond the value of K where it stops.

The second algorithm, called the full-search algorithm, computes GCV, minimized over α, at all values of K in our trial sequence. The value of K in that sequence that minimizes GCV is selected.

The myopic algorithm usually takes far less computation than the full-search algorithm. However, P-splines can be computed so rapidly that this advantage is not compelling.

The one drawback to the myopic algorithm is that it can "stop before it really gets started." More precisely, for regression functions with enough complexity, neither $K = 5$ or 10 will fit the data satisfactorily and it may happen that 5 knots is just as good as 10. In this case, the myopic policy will stop at 10 knots whereas the full-search policy will select a much greater number of knots and achieve a much better fit. An example where this phenomenon occurs is a sine wave with 12 cycles; see Section 4. This problem with the myopic policy may be occurring in the fossil data example given in Section 5.2, though with real data it is impossible to know the "right answer."

4. SIMULATIONS

This study uses seven examples where m, σ, and n vary as shown in Table 1. A number of other examples were also studied but are not included here since they support the same conclusions as the seven reported examples. In all cases, the x_i are equally spaced on [0, 1]. In this study, only quadratic splines ($p = 2$) are used. The reason for this restriction

Table 1. Parameter Values for the Seven Case Studies

Name	*m*	σ	*n*	*SN ratio*
Logit	(4.1)	0.2	75	5.08
Bump	(4.2)	0.3	100	3.89
Sine3	$\sin(2\pi\theta)$, $\theta = 3$	1	150	0.50
Sine6	$\sin(2\pi\theta)$, $\theta = 6$	0.5	150	2
Sine12	$\sin(2\pi\theta)$, $\theta = 12$	0.25	150	8
SpaHet3	(4.3) with $j = 3$	0.3	200	0.88
SpaHet3LS	(4.3) with $j = 3$	0.3	2,000	0.88

is that quadratic splines work very well in practice when m is smooth. For functions with discontinuities or "kinks" where the first derivative is discontinuous, $p = 0$ or 1 is preferred to $p = 2$, but such functions are not included in this study. I have seen little or no evidence that $p > 2$ outperforms $p = 2$ though estimates with $p = 3$ are usually similar to those with $p = 2$. Thus, all quantitative conclusions in this study about selecting the number of knots apply only to quadratic splines. A similar study using linear splines and regression functions with less smoothness might be a useful future research project.

The first example, called "Logit," use a logistic function

$$m(x) = \frac{1}{1 + \exp\{-20(x - .5)\}}. \tag{4.1}$$

The second example, called "Bump," uses

$$m(x) = x + 2\exp[-\{16(x - .5)\}^2]. \tag{4.2}$$

The next three examples, "Sine3," "Sine6," and "Sine12" are sine waves with 3, 6, and 12 cycles, respectively. The final examples, "SpaHet3" and "SpaHet3LS," have a spatially heterogeneous function used by Wand (2000):

$$m(x; j) = \sqrt{x(1 - x)}\sin\left\{\frac{2\pi(1 + 2^{(9-4j)/5})}{x + 2^{(9-4j)/5}}\right\}. \tag{4.3}$$

The parameter j controls the amount of spatially heterogeneity and in this example $j = 3$ which give only a slight amount of heterogeneity. Spahet3LS has a larger value of n than SpaHet3; LS = "large sample." In Table 1 the SN ratio is the ratio of the sample variance of $\{m(x_i)\}_{i=1}^n$ to σ^2.

For each example 300 datasets were simulated. P-splines with different fixed values of K were compared to the myopic and full-search algorithms using MASE (mean average squared error) defined as the mean over the 300 datasets of the average squared error,

$$\text{ASE} = n^{-1}\sum_{i=1}^{n}\{m(x_i; \widehat{\boldsymbol{\beta}}) - m(x_i)\}^2.$$

Figures 1–7 show a typical dataset in panel (a), MASE comparisons in panel (b), and histograms of K as selected by the myopic and full-search algorithms in panel (c). Also,

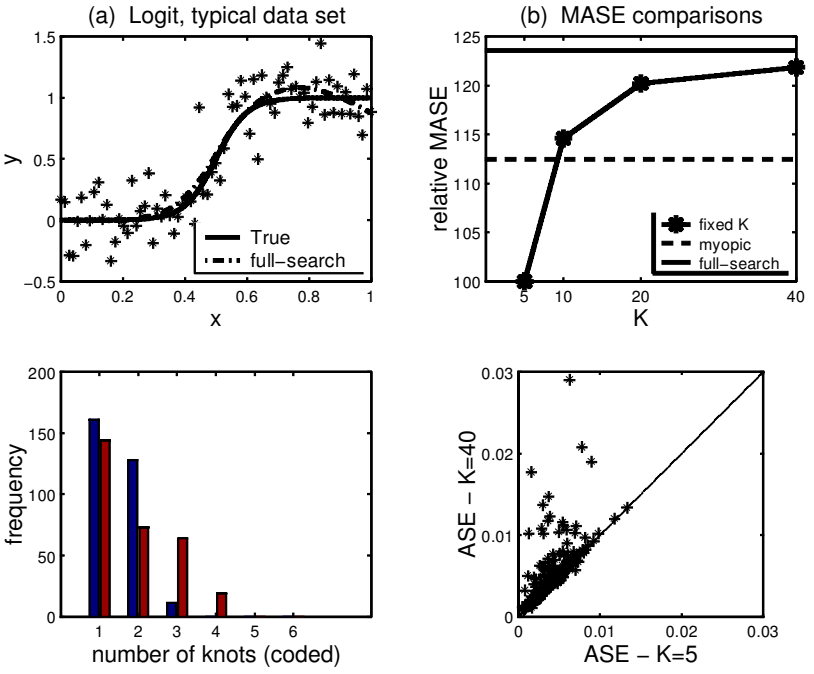

Figure 1. Logit example. Simulation of 300 datasets. (a) Typical dataset, true regression function, and estimate from the full-search algorithm. (b) Mean average squared error (MASE) as a function of K with horizontal lines through the values of MASE for the myopic and full-search algorithms. (c) Histograms of K as chosen by the myopic (on left) and full-search (on right) algorithms, respectively. The number of knots is coded: $1 = 5$ knots, $2 = 10$ knots, $3 = 20$ knots, $4 = 40$ knots, $5 = 80$ knots, and $6 = 120$ knots. (d) Plot of average squared errors for the 300 samples for a small and for a large value of K

panel (d) plots the 300 pairs of ASE for a large value of K and ASE for a small value of K. Define the "oracle estimator" to be the fixed-K estimator which uses that value of K that minimizes MASE among the values of K searched by the myopic and full-search estimators. In each panel (b), relative MASE is the MASE divided by MASE of the oracle estimator.

4.1 RESULTS

Logit

Since $n = 75$ in this case and only trial values of K less than $n - p - 1$ are used, 80 and 120 are automatically removed from the trial sequence.

One can see from Figure 1(b) that MASE rises monotonically as K increases from 5 to 40 and is about 20% higher at 40 than at 5. The myopic and full-search algorithms choose $K = 5$ in at least half of all samples. The myopic algorithm tends to choose smaller values of K than the full-search algorithm and the myopic algorithm has a smaller MASE than the full-search algorithm. In panel (d) we see that in about 90% to 95% of all samples, the ASE of the 5-knot fit is similar to that of the 40-knot fit. However, in the remaining samples the

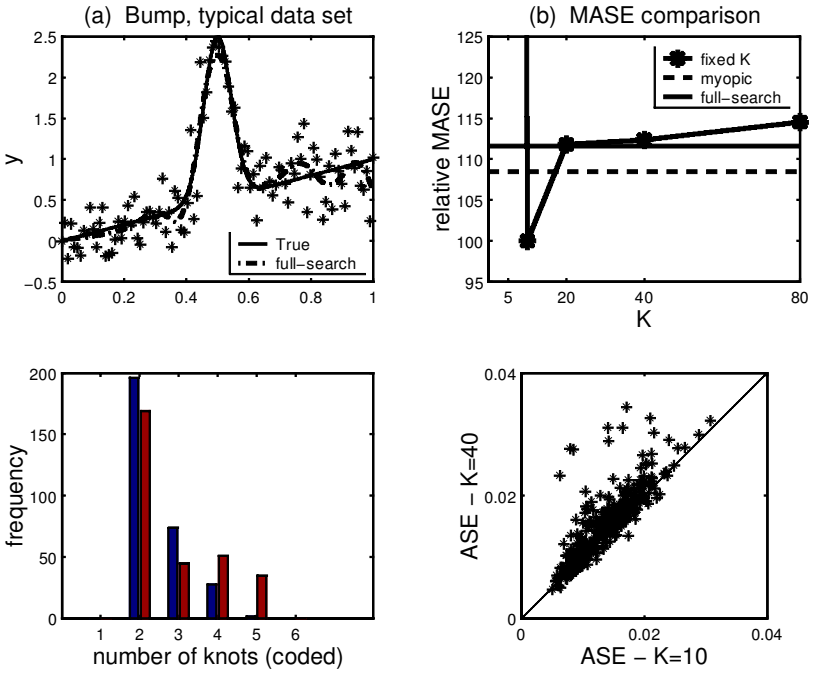

Figure 2. Bump function example. (a)–(d) as in Figure 1.

5-knot fit has a much smaller ASE than the 40-knot fit. Unfortunately, the GCV values of these poor fits is small and the full search tends to choose a large number of knots precisely when it should not. The full-search algorithm has a larger MSE than any of the fixed-knot estimators, even though the full-search algorithm is attempting to choose the best of the fixed-knot estimators.

These examples are similar to many found in practice, where the regression function is monotonic and there is a fair amount of noise. In such situations, the number of knots is not very important as long as there are at least five and a fixed number of knots in the range of 5 to 20 seems as sensible as using a GCV-driven search. However, there is some penalty for using more knots than necessary.

Bump

For Bump shown in Figure 2, 5 knots is clearly not enough. For Bump, 10, 20, 40, and 80 knots all have similar MASE values, though MASE increases somewhat with K. Both the myopic and full-search algorithm select enough knots to have MASE values near optimal. Neither algorithm, of course, does as well as the oracle estimator but the myopic algorithm has a MASE only about 9% larger than the oracle estimator. Notice in panel (d) that the 10- and 40-knot fits give similar ASE values for most samples, and the higher MASE value of the 40-knot fit is due to poor ASE values in about 5% of the samples.

This example should be similar to those in practice with a unimodal regression function. There is minimum adequate number of knots and both search algorithms select a value of

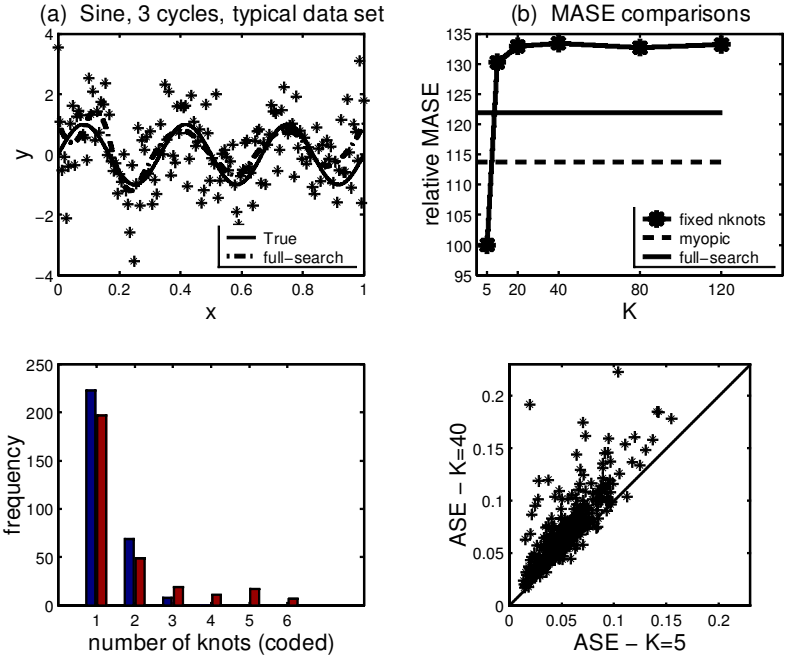

Figure 3. Sine3 example. (a)–(d) as in Figure 1.

K above this minimum. Experimentation not reported here with other values of σ shows that the minimum adequate number of knots increase with the signal-to-noise ratio of the data.

Sine3

In Sine3 (Figure 3), MASE increases monotonically as K increases through 5, 10, and 20 and then levels off. Both the myopic algorithm and full-search algorithm select low numbers of knots and have lower MASE values than fits with a fixed number of knots equal to 10, 20, 40, 80, or 120. It may seem surprising that five knots works well for a function with this much complexity. However, five knots neatly divides [0, 1] into the six subintervals where m'' has a constant sign. Therefore, m can be approximated reasonably well, at least relative to the amount of noise in the data, by a 5-knot quadratic spline which necessarily has a constant value of m'' between knots.

In a further simulation, σ was decreased by a factor of ten and then the bias of the 5-knot fit was noticeable and the 5-knot fit had the largest MASE. The full-search algorithms chose 20 or more knots in almost all samples, but the myopic algorithm often chose 5 or 10 knots and performed poorly.

Sine6

For a six-cycle sine wave (Figure 4), at least 20 knots are needed for a satisfactory MASE and the full-search algorithm always selects at least 20 knots. The myopic search

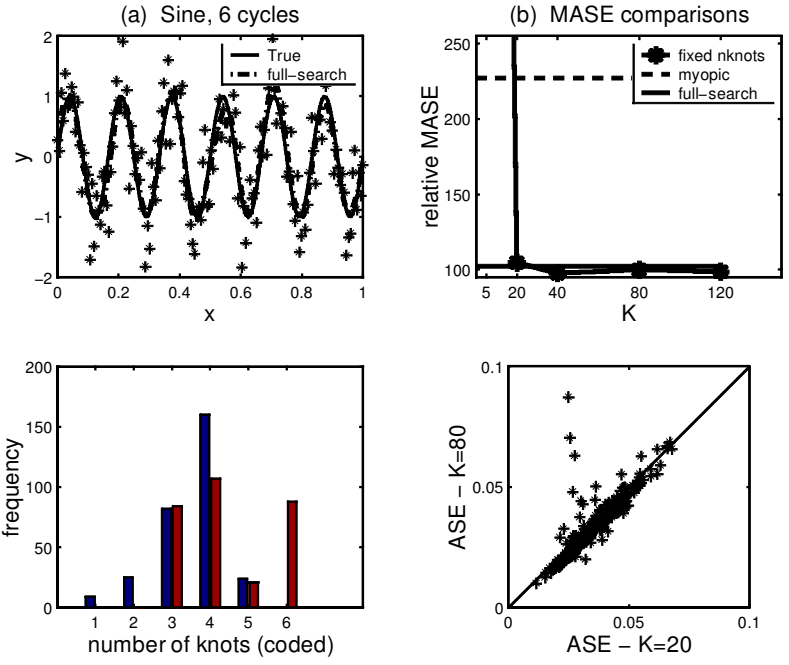

Figure 4. Sine6 example. (a)–(d) as in Figure 1.

occasionally stops prematurely at 5 or 10 knots and has a much higher MASE than the full-search algorithm.

Sine12

For a 12-cycle sine wave (Figure 5) at least 40 knots are needed for a satisfactory MASE; MASE is nearly constant for 5, 10, or 20 knots and then drops by more than a factor of ten as the number of knots increases from 20 to 40. MASE is also nearly constant for 40, 80, or 120 knots. The full-search algorithm always selects at least 40 knots. The myopic search stops prematurely at 5 knots in every one of the 300 Monte Carlo samples.

This is a nice illustration of the potential pitfall of the myopic algorithm. Clearly, this algorithm cannot be used as a black box. However, cyclic data of this type usually arise in practice when there is a known cause for the periodicity, for example, one has collected hourly data for 12 days. In such cases, the investigator would know not to use the myopic algorithm unless started with considerably more than 5 knots—starting at 23 or 47 knots (to divide the data into 24 or 48 intervals) would be sensible if one suspected 12 periods.

The plot in panel (d) we see that the 40-knot and 80-knot fits have highly correlated ASE values, but that the 40-knot fits have noticeably higher ASE values because of bias.

SpaHet3 and SpaHet3LS

For SpaHet3 (Figure 6) the number of knots has relatively little effect and any fixed

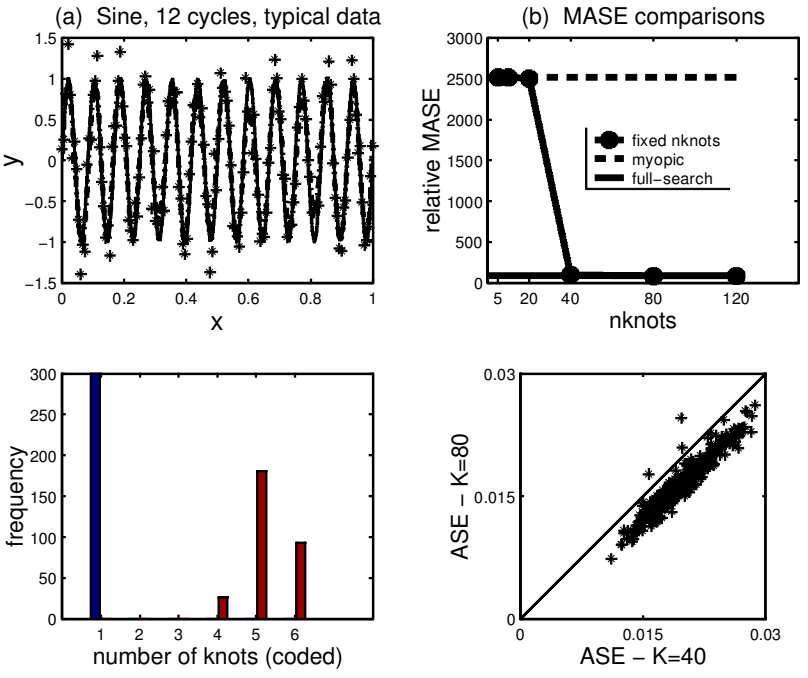

Figure 5. Sine12 example. (a)–(d) as in Figure 1.

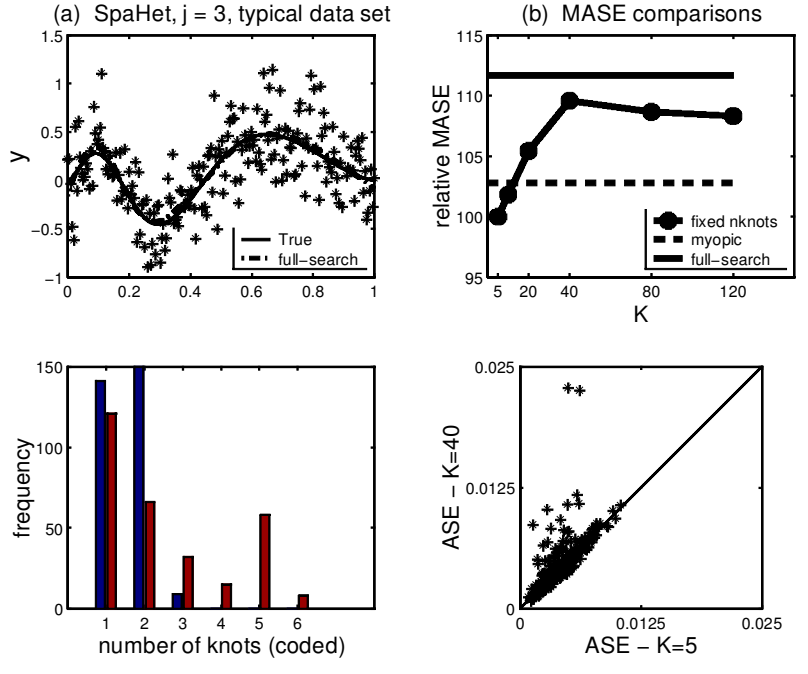

Figure 6. Spatial heterogeneity example. (a)–(d) as in Figure 1.

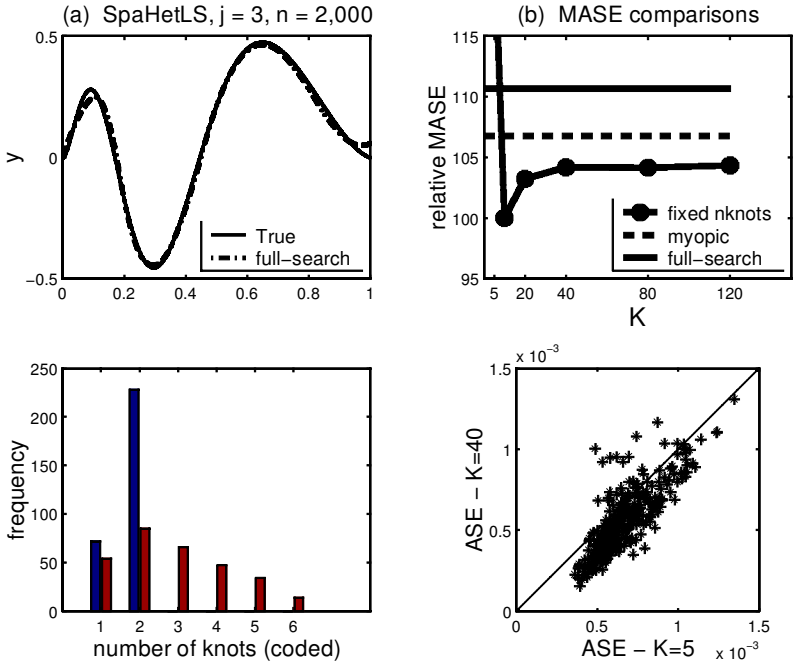

Figure 7. Spatial heterogeneity, large sample, example. (a)–(d) as in Figure 1.

number of knots five or greater works reasonably well, as do the two automatic algorithms. However, less knots is generally better than more, and the myopic algorithm does a good job of selecting a low number of knots. One can see in panel (d) that the higher MASE of 40 knots is due to a few outlying datasets with large values of ASE. Unfortunately, the full-search algorithm tends to select a poor value of K precisely for such datasets and the full-search algorithm has a slightly higher MASE value than *all* of the fixed choices of knots.

As discussed in Section 2, the GCV parameter λ is equal to 1 in the simulations reported here. However, I also experimented with $\lambda = 1.5$ and 2 for the SpaHet3 example. I found that, when $\lambda = 2$, then the number of knots had much less effect—all fixed numbers of knots and the two search algorithms gave very similar fits for all samples. This could be seen in the plot of ASE for $K = 40$ and for $K = 5$; all points fell close to the diagonal line. However, MASE was slightly higher with $\lambda = 2$ than when λ was 1 and when $\lambda = 2$ the fits showed signs of bias with peaks and troughs being rounded off. Not surprisingly, using $\lambda = 1.5$ gave results intermediate between $\lambda = 1$ and $\lambda = 2$. In summary, using $\lambda > 1$ somewhat stabilizes fits with larger values of K but at a cost in terms of MASE.

SpaHet3LS shown in Figure 7 is the "large sample" version of SpaHet3 where all parameters are the same except that n of 200 is raised to 2,000. One can see that for this larger sample size, five knots is not enough, but ten knots is adequate. I also experimented with low noise levels for SpaHet3. If the SN ratio is raised, then more than five knots are needed. For example, if σ is lowered from 0.3 to 0.1, then at least 10 knots are needed for

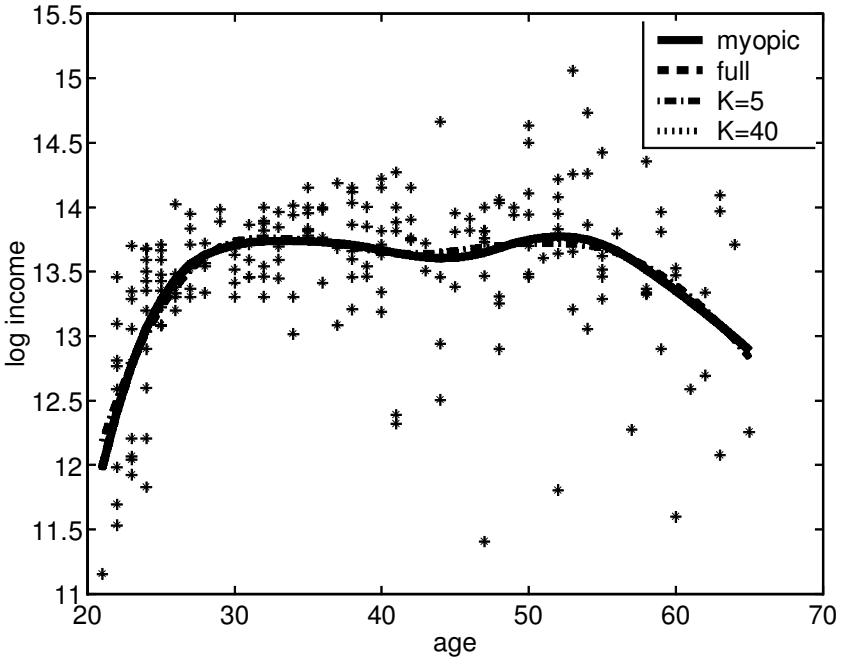

Figure 8. Myopic and full search fits (both use 10 knots) to the age and income data, plus fits with 5 and with 40 knots.

MASE to be near its minimum. If $\sigma = 0.03$ then using 20 knots is about 30% more efficient than 10 knots but using more than 20 knots does not improve upon 20 knots. Both the myopic and the full-search algorithms select enough knots under each of these three values of σ. Since this regression function is at least as complex as most found in biology and social sciences, I feel that 20 knots can be recommended for routine use in such disciplines, even for very large sample sizes. Of course, there will be exceptions—long periodic time series, for example. Also, like all conclusions in this study, this result applies only to smooth functions, that is, functions with at least one continuous derivative.

5. EXAMPLES

In this section, two real examples are given. In the first, the number of knots has little effect on the fit, but this is not true of the second example.

5.1 AGE AND INCOME DATA

Ullah (1985) collected data on the age and income of 205 Canadian workers. Figure 8 shows the myopic and full-search fits of the log of income to age. Both algorithms choose 10 knots so that the fits are identical. For contrast, the fits with 5 and with 40 knots are also shown. The fit with 40 knots nearly overlays the 10-knot fit—the same would be true of the 20-knot fit if it were shown. The 5-knot fit is slightly different, but the different is small

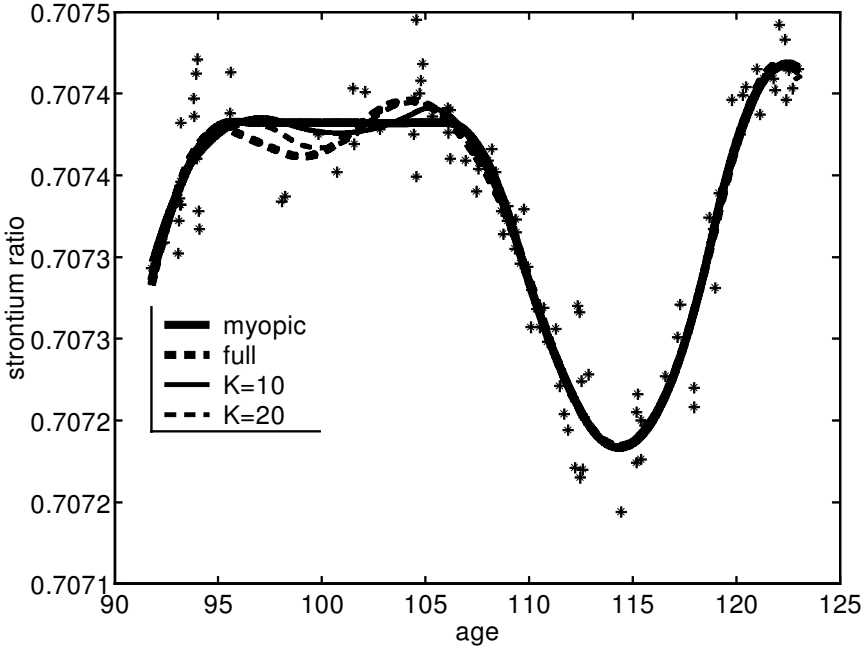

Figure 9. Myopic (5 knot) and full search (80 knot) fits to the fossil data, plus fits with 10 and 20 knots.

compared to the scatter in the data. It is, however, interesting that the search algorithms do not choose this fit

5.2 FOSSIL DATA

This example is discussed in Chaudhuri and Marron (1999). The global climate millions of years ago is reflected in ratios of strontium isotopes in fossils from that period. Figure 9 is a scatterplot of the age of 106 fossils (in millions of years) determined by biostratigraphic methods and their strontium ratios. The myopic algorithm chooses 5 knots and the full-search algorithm chooses 80 knots. The fits with 10 and 20 knots are also shown. The 40-knot fit is similar to the full-search fit with 80 knots. Clearly the number of knots does affect the fit in the region between 95 and 105 million years. A dip in this region is seen in all fits except the 5-knot fit. However, Chaudhuri and Marron used their SiZer method to conclude that the data do *not* support the hypothesis of a dip about 97 million years. Thus, there is no evidence that the 5-knot fit without the dip selected by the myopic algorithm is inferior to the other fits; if the dip is spurious then the 5-knot fit could be considered better than the other estimators.

6. ADDITIVE MODELS

An attractive generalization of the multiple linear regression model is the additive model (Hastie and Tibshirani 1990). Suppose that for the ith observation one observes

L predictor variables, $x_{1,i}, \ldots, x_{L,i}$. The additive model is

$$y_i = \beta_0 + m_1(x_{1,i}) + \cdots + m_L(x_{L,i}) + \epsilon_i. \tag{6.1}$$

I will use a spline model for each m_l:

$$m_l(x_l; \boldsymbol{\beta}_l) = \beta_{l,1}x_l + \cdots + \beta_{l,p}x_l^p + \sum_{k=1}^{K_l} b_{l,k}(x_l - \kappa_k)_+^p. \tag{6.2}$$

When additive models are fit using local polynomial or similar smoothers, there is a need to impose constraints to ensure identifiability. Moreover, additive models cannot be fit directly by local polynomial regression, but rather a backfitting algorithm is used. Because, there is no intercept in (6.2), model (6.1) with m_l given by (6.2) is identifiable. Moreover, direct fitting of additive spline models is straightforward (Marx and Eilers 1998).

The parameter vector $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^\top, \ldots, \boldsymbol{\beta}_L^\top)^\top$ can be estimated by penalized least-squares by minimizing

$$\sum_{i=1}^{n} \left\{ y_i - m(x; \boldsymbol{\beta}) \right\}^2 + \sum_{l=1}^{L} \alpha_l \sum_{k=1}^{K_l} \beta_{l,p+k}^2, \tag{6.3}$$

where $\alpha_1, \ldots, \alpha_L$ are smoothing parameters. The component functions $(\{m_l\}_{l=1}^L)$ may require different amounts of smoothing and this can be accomplished by allowing the α_l values to vary independently rather than assuming a common value. Ruppert and Carroll (2000) discussed an algorithm for choosing $\{\alpha_l\}_{l=1}^L$ by GCV, and they compared that algorithm to one using a common α. The separate α algorithm generally outperformed the common α algorithm. The diagonalization method given in Section 2 for computing univariate P-splines is extended to additive P-spline models in Section 6.1.

I recommend using a common value of K_l, which I will call K as before. The reason for this recommendation, is that the value of K_l is not too important, provided it is large enough and it is relatively easy to choose a sufficiently large value of K by GCV. If one wished to choose both a different number of knots and a different value of α for each variable, this would result in a rather complex and computationally intensive algorithm.

I have experimented with the following "full-search" algorithm. The additive model is fit using the Ruppert and Carroll (2000) algorithm with separate α_l values for K equal to each of 5, 10, 20, and 40. Then the value of K minimizing GCV is used. I did not try more than 40 knots, since the number of parameters of the additive spline model is $L(p+K)+1$ and is rather large for K much bigger than 40.

To test the algorithm, data were simulated with $L = 3$, $n = 150$, x_1, x_2, and x_3 are independent uniform(0,1) random variables,

$$m_1(x_1) = \sin(2\pi\theta x_1),$$

with $\theta = 3$ or 6, $m_2(x) = 1/(1 + x_2)$, and $m_3(x_3) = x_3^4$. I used $\sigma = 1$ when $\theta = 3$ and $\sigma = 0.25$ when $\theta = 12$. These two cases are similar to Sine3 and Sine12, except that they have two additional components. Since m_2 and m_3 are monotonic, the value of K needed to get a good fit depends largely on the value of θ, at least for $\theta \geq 3$.

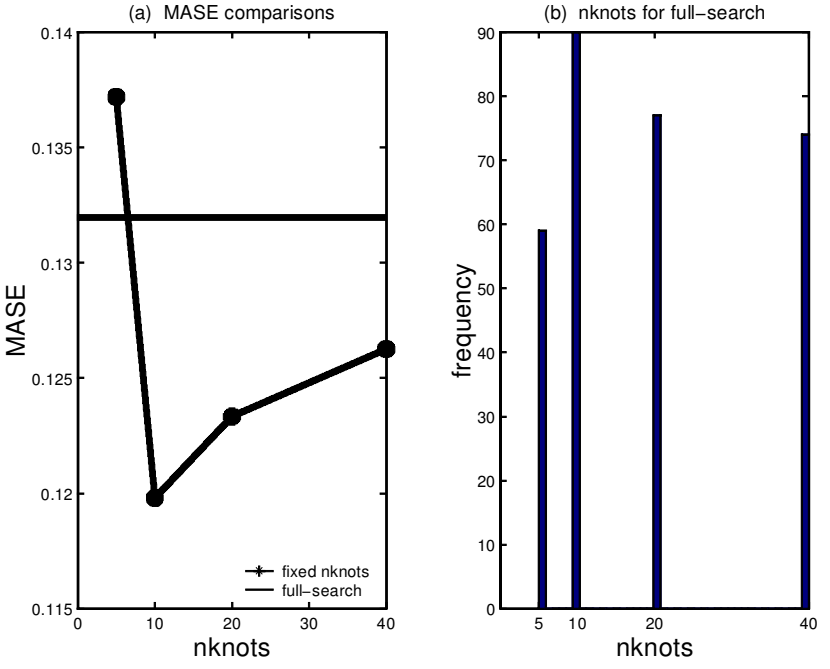

Figure 10. Additive model with Sine3 as first component and $\sigma = 1$. (a) MASE. (b) Histogram of values of K selected by the full-search algorithm.

Figure 10 shows the results when $\theta = 3$. One can see from panel (a) that MASE is minimized at 0.12 when $K = 10$. This is the value of K mostly commonly selected; see panel (b). However, the algorithm has difficulty selecting the best value of K. This difficulty exists for the same reason that the difficulty is not serious: MASE does not depend much upon whether 5, 10, 20, or 40 is used.

When $\theta = 12$, it is crucial than 40 rather than 5, 10, or 20 knots be used since MASE is approximately 0.47 for 5, 10, or 20 knots and about 0.05 for 40 knots. The full-search algorithm chooses 40 knots in every one of the 300 simulations.

6.1 COMPUTING ADDITIVE P-SPLINES

In the algorithm of Ruppert and Carroll (2000), $\alpha_1, \ldots, \alpha_L$ are chosen by GCV in two steps. In the first step, GCV is minimized with a common smoothing parameter, that is, with $\alpha_1 = \cdots = \alpha_L = \alpha$, say. In Step 2, starting with this common smoothing parameter, $\alpha_1, \ldots, \alpha_L$ are selected one at a time by minimizing the GCV criterion. More precisely, α_1 is set equal to its minimum GCV value with the other α_l fixed, then α_2 is set to its minimum GCV value with the other α_l fixed, and so on. One cycles in this way through $\alpha_1, \ldots, \alpha_L$ a fixed number of iterations. Two iterations generally works well in practice.

To use the diagonalization technique of Theorem 1, let $\mathbf{X}$ be the $n \times \{1 + L(p + K)\}$ matrix whose nth row is the set of basis functions for model (6.1) and (6.2) evaluated at

$x_{1,i}, \ldots, x_{L,i}$. Similarly, let $\boldsymbol{\beta}$ be the vector of all coefficients in this model. For $l = 1, \ldots, L$, let $\mathbf{D}_l$ be the diagonal matrix with diagonals elements equal to either zero or one such that $\sum_{k=1}^{K} b_{l,k}^2 = \boldsymbol{\beta}^\top \mathbf{D}_l \boldsymbol{\beta}$.

In Step 1, to find the common value of $\{\alpha_l\}_{l=1}^{L}$, call it α, that minimizes GCV, one can apply directly the diagonalization technique of Theorem 1 with $\mathbf{D} = \mathbf{D}_1 + \cdots + \mathbf{D}_L$. In Step 2 suppose that one wants to find the value of α_{l^*} that minimizes GCV with the other α_l fixed. Then we find a square $\mathbf{B}$ such that $\mathbf{B}^{-1}\mathbf{B}^{-\top} = \{\mathbf{X}^\top\mathbf{X} + \sum_{l \neq l^*} \alpha_l \mathbf{D}_l\}$ so that $\widehat{\boldsymbol{\beta}}$ solves $(\mathbf{B}^{-1}\mathbf{B}^{-\top} + \alpha_{l^*}\mathbf{D}_{l^*})\widehat{\boldsymbol{\beta}} = \mathbf{X}^\top\mathbf{Y}$. With $\mathbf{B}$ chosen in this manner, the computation of spline fits and GCV over a grid of values of α_{l^*} is done as in Section 2 for a univariate spline, but with α and $\mathbf{D}$ of the univariate spline replaced by α_{l^*} and $\mathbf{D}_{l^*}$.

The total number of diagonalizations is L times the number of iterations. Since diagonalization is rapid, this amount of computation is not burdensome.

7. DISCUSSION AND CONCLUSIONS

The following general conclusions emerge from the results presented here:

1. There is a minimum adequate value of K, the number of knots. Fits using less than this minimum K have high bias and MASE. Fits using more than this number of knots give satisfactory fits.
2. When using more than the minimum number of knots, often there is a slight MASE penalty. However, this larger MASE is due to a few outlying datasets. For a typical dataset ASE is very similar for any two values of K above the minimum. Also, the cost of using more knots that necessary is much less than the cost of using too few knots.
3. Both the myopic and full-search algorithms are successful in most cases in selecting a value of K above the minimum needed for a good fit.
4. An exception to 3 is that the myopic algorithm may select a value of K below the minimum because both of the first two trial values are too small and the increase in K from the first to the second trial value does not improve GCV.

A researcher with some knowledge of the shape of m may very well be able to select a suitable value of K without using an automatic algorithm. When an automatic algorithm is desired, the full-search algorithm is closer to foolproof than the myopic algorithm. The myopic algorithm is suitable for use by an investigator with some sense of when it might fail.

Besides using GCV or some other statistical criterion to choose the number of knots, one might use a simple default. For example, Matt Wand (personal communication) stated that "my current default is a knot between every d observations, where $d = \max(4, \lfloor n/35 \rfloor)$. ($\lfloor r \rfloor$ is the floor of r, that is, the greatest integer less than or equal to r.) Wand's default chooses roughly $\min(n/4, 35)$ knots. Since the algorithms in this article cannot choose 35 knots but can choose 40 knots, consider a default similar to Wand's where $d = \max(4, \lfloor n/40 \rfloor)$, so that approximately $\min(n/4, 40)$ knots are used.

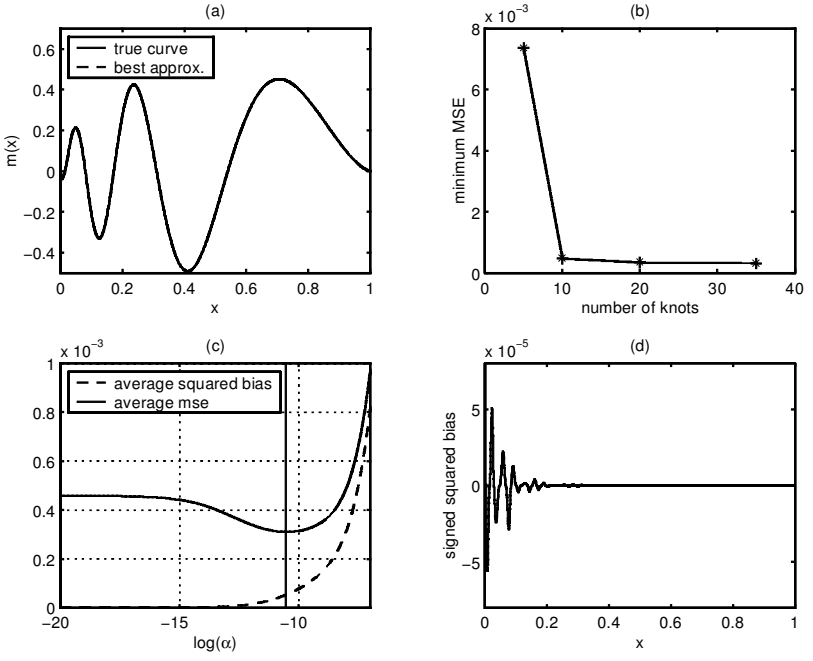

Figure 11. Study of bias with $n = 25{,}000$, $\sigma^2 = 0.3$, and the regression function given by (4.3). (a) Regression and best least-squares approximation by a 35-knot quadratic spline. The two curves are too close to be easily distinguished. (b) Average MSE minimized over α as function of the number of knots of a quadratic spline. (c) Squared bias and MSE as a function of α for a 35-knot quadratic spline. (d) Signed squared bias for a 35-knot quadratic spline with $\alpha = 0$.

For the cases in Table 1 (p. 742) this default will use 18 knots when $n = 75$, 25 knots when $n = 150$, and 40 knots in all other cases. Comparing these values of K with the results in Section 4.1, one sees that the default will choose an effective number of knots in all the cases studied. Of course, the default will fail in more extreme cases such as a long periodic time series. The default will often choose many more knots than necessary. In the Sine3 example, the default will choose 40 knots and have about 40% greater MASE than a 5-knot spline. In this example, the myopic algorithm is most likely to choose 5 knots and has a MASE only 14% greater than a 5-knot spline; see Figure 3 (b) (p. 745).

It may seem surprising that a default that uses at most 35 (or 40) knots could be recommended *for effectively all sample sizes* and for all smooth regression functions without too many oscillations. To see that this is recommendation is reasonable, at least for regression functions without too many oscillations, consider an example with n extremely large. The mean function in this example is (4.3) with $j = 4$, the variance is $\sigma^2 = 0.3$, and the sample size is $n = 25{,}000$. The x values are equally spaced on $[0, 1]$. With this large a value of n, m can be estimated extremely accurately. One might expect that the bias due to using only 35 knots would be bothersome. In fact, as will be seen, this bias is negligible. When α is fixed as here, a P-spline is a linear estimator so that biases and variances can be computed exactly and there is no need for simulations. Some results for this example

with n extremely large are found in Figure 11. In panel (a) one sees the regression function m and the best approximation of m by a 35-knot quadratic spline. Here "best" means in a least-squares sense, i.e., the best approximation is the spline $m(x;\boldsymbol{\beta})$ that $\boldsymbol{\beta}$ minimizes $\sum_{i=1}^{n}\{m(x_i) - m(x_i;\boldsymbol{\beta})\}^2$. Visually, the two curves are impossible to distinguish. One can see that a 35-knot spline approximates this rather complex regression function nearly perfectly. However, one must be careful if one recommends any single value of K as a default, No default will work for all datasets for two reasons. First, one might have a regression function with many oscillations and therefore need more knots than the default. Second, the examples have shown that often using less knots is better.

In panel (b) we see the average (over x) MSE of a quadratic spline, minimized over α, as a function of K, Clearly there is little improvement when K increase from 20 to 35, even for this extreme case of n very large and σ relatively small.

In panel (c) one sees the average squared bias and average mean squared error. There is a vertical line through the value of α that minimizes the MSE. One can see that the squared bias there is much larger than when α is 0. Most of the bias is due to smoothing with a positive value of α; little of the bias is due to the approximation of m by a 35-knot spline. In fact, the squared bias due to spline approximation (the bias at $\alpha = 0$) is only about 4% of the squared bias at the value of α that minimizes the MSE. Moreover, the squared bias is a relatively small portion of the total MSE. The squared bias due to the spline approximation is only about 0.7 % of the minimum MSE.

Panel (d) shows the signed squared bias, that is, the squared bias times the sign of the bias, when $\alpha = 0$. The bias is the difference between the two curves in panel (a). Signed squared bias was used instead of the bias, since the magnitude of the signed squared bias is comparable to the quantities in panel (c). Note that the vertical axis units are 10^{-3} and 10^{-5} in (c) and (d), respectively. This fact shows again that the bias due to spline approximation is negligible. Of course, if σ^2 is made very small, say shrunk by a factor of 100 with all else held constant, then the bias due to spline approximation will be a relatively large portion of the total MSE. However, in that case the MSE itself is negligible—the spline estimator will virtually identical to m.

Although simulations were reported in Section 4 only for quadratic splines, the power basis functions, and the ridge penalty in (2.1), other degrees of the splines, bases, and penalties can be used. As a small experiment, the myopic and full-search algorithms were tried on the Logit case with cubic splines and the penalty (2.8). As with quadratic splines and penalty (2.1), all choices of K were reasonable and both algorithms tended to choose small values of K. For $K = 5$, 10, or 10 ($K = 40$), the MASE of the quadratic spline and ridge penalty estimator was somewhat smaller (larger) than the MASE of the cubic spline and quadratic integral penalty on m''.

Luo and Wahba's (1997) hybrid adaptive spline (HAS) is somewhat related to the algorithms of this article. HAS has two stages. In the first stage, a subset of knots is chosen by a stepwise procedure similar to Friedman's (1991) MARS. In a second stage, a smoothing spline is fit with only these knots. The difference between HAS and the algorithms of this paper is that in HAS one can select from among the 2^K possible models using a subset of K

potential knots. In contrast, my algorithms only chose between a few, more precisely, six, models indexed by K. HAS is "spatially adaptive" since it can put a large number of knots in a region with large changes in curvature. The penalized splines proposed here have their knots at equally spaced quantile, so are not spatially adaptive. However, spatially adaptivity can be achieved, as in Ruppert and Carroll (2000), by replacing a global penalty, α, by a local penalty $\alpha(x)$ depending on spatial location. In Ruppert and Carroll (2000) there is a small study comparing global and local penalty splines with results from Luo and Wahba for smoothing splines, HAS, MARS, and the wavelet estimator SureShrink. The study was too small to reach any strong conclusions but the local penalty spline was the best estimator in that study and HAS, smoothing splines, and global penalty splines were about equal.

The spline estimators in this article are intended for smooth regression functions and only such regression functions were included in the simulation study. Spline algorithms for discontinous regression functions have been developed, for example, by Denison, Mallick, and Smith (1998), but are quite different from the methods here.

ACKNOWLEDGMENTS

Doug Nychka showed me the diagonalization method described by Theorem 1. I thank Matt Wand and John Staudenmayer for their careful reading of an earlier draft and their helpful comments. The very perceptive comments of the associate editor and three referees greatly improved this article. This research was supported NSF grant DMS-9804058.

[Received September 2000. Revised August 2001.]

REFERENCES

Chambers, J. M., and Hastie, T. J. (1993), *Statistical Models in S*, New York and London: Chapman & Hall.

Chaudhuri, P., and Marron, J. S. (1999), "SiZer for Exploration of Structures in Curves," *Journal of the American Statistical Association*, 94, 807–823.

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

Eilers, P. H. C., and Marx, B. D. (1996), "Flexible Smoothing With B-splines and Penalties" (with discussion), *Statistical Science*, 11, 89–121.

Eubank, R. L. (1988), *Spline Smoothing and Nonparametric Regression*, New York and Basil: Marcel Dekker.

Friedman, J. H. (1991), "Multivariate Adaptive Regression Splines" (with discussion), *The Annals of Statistics*, 19, 1–141.

Friedman, J. H., and Silverman, B. W. (1989), "Flexible Parsimonious Smoothing and Additive Modeling" (with discussion), *Technometrics*, 31, 3–39.

Gill, P. E., Murray, W., and Wright, M. H. (1981), *Practical Optimization*, New York, Academic Press.

Hastie, T. (1996), "Pseudosplines," *Journal of the Royal Statistical Society*, Series B, 58, 379–396.

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

Luo, Z., and Wahba, G. (1997), "Hybrid Adaptive Splines," *Journal of the American Statistical Association*, 92, 107–116.

Marx B. D., and Eilers P. H. C. (1998), "Direct Generalized Additive Modeling with Penalized Likelihood," *Computational Statistics and Data Analysis*, 28, 193–209.

Nychka, D. (2000), "Spatial Process Estimates as Smoothers," *Smoothing and Regression. Approaches, Computation and Application*, ed. M. G. Schimek, New York: Wiley, pp. 393–424.

O'Sullivan, F. (1986), "A Statistical Perspective on Ill-Posed Inverse Problems" (with discussion), *Statistical Science*, 1, 505–527.

——— (1988), "Fast Computation of Fully Automated Log-Density and Log-Hazard Estimators," *SIAM Journal of Scientific and Statistical Computation*, 9, 363–379.

Ruppert, D., and Carroll, R. J. (2000), "Spatially-Adaptive Penalties for Spline Fitting," *Australian and New Zealand Journal of Statistics*, 42, 205–223.

Stone, C. J., Hansen, M., Kooperberg, C., and Truong, Y. K. (1997), "Polynomial Splines and Their Tensor Products in Extended Linear Modeling" (with discussion), *The Annals of Statistics*, 25, 1371–1470.

Ullah, A. (1985), "Specification Analysis of Econometric Models," *Journal of Quantitative Economics*, 2, 187–209.

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

Wand, M. P. (2000), "A Comparison of Regression Spline Smoothing Procedures," *Computational Statistics*, 15, 443–462.