Selecting the Number of Knots For Penalized Splines

David Ruppert [*]

October 9, 2000

Abstract

Penalized splines, or P-splines, are regression splines fit by least-squares with a roughness penaly. 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 parameter to choose is the number of knots. 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 myoptic 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 myoptic 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.

Key words and phrases. Additive models, Full search, Myoptic search, P-spline, Smoothing.

[*] David Ruppert is Professor, School of Operations Research & Industrial Engineering, Cornell University, Ithaca, New York 14853-3801 (E-mail: davidr@orie.cornell.edu). Ruppert's research was supported a NSF grant DMS-9804058.

1 Introduction

In this paper we study a variant of smoothing splines that we call 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 parameters. In this article, wee discuss the choice of the number of knots and penalty parameters jointly by generalized cross validation.

Suppose that we have data (x_i, y_i) where for now x_i is univariate,

$$y_i = m(x_i) + \epsilon_i, \tag{1}$$

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 we use a regression spline model

$$m(x;\boldsymbol{\beta}) = \beta_0 + \beta_1 x + \cdots + \beta_p x^p + \sum_{k=1}^{K} b_k (x - \kappa_k)_+^p, \tag{2}$$

where $p \geq 1$ is an integer, $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_p, b_1, \ldots, b_K)^{\mathsf{T}}$ 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 (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.

When fitting model (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, e.g., 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.

In this paper a penalty approach is used that is similar to smoothing splines but with fewer knots and a somewhat different roughness penalty. We allow K in (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. A roughness penalty is placed on $\{b_k\}_{k=1}^{K}$ which is the set of 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 have traced the original idea to O'Sullivan (1986, 1988). Also, P-splines are low dimensional smoothers, i.e., 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).

In Section 2 penalized least-squares estimation of univariate P-splines is discussed. In Section 3 two algorithms for selecting the number of knots are introduced. Section 4 presents some simulation result. The extension to additive models is discussed in Section 5, and Section 6 contains further discussion and a summary of the conclusions.

2 The penalized leasts-squares estimator

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

$$\sum_{i=1}^{n} \Big\{ y_i - m(x;\boldsymbol{\beta}) \Big\}^2 + \alpha \sum_{k=1}^{K} b_k^2, \tag{3}$$

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 $b_k = 0$ for $k = 1, \ldots, K$. Selection of α by generalized cross validation (GCV) will be discussed below.

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

$$\boldsymbol{X}_i = (\, 1, \quad x_i, \quad \cdots \quad x_i^p, \quad (x_i - \kappa_1)_+^p, \quad \cdots \quad (x_i - \kappa_K)_+^p \,). \tag{4}$$

Also, let $\boldsymbol{D}$ 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

$$\widehat{\boldsymbol{\beta}}(\alpha) = \Big(\boldsymbol{X}^T\boldsymbol{X} + \alpha \boldsymbol{D} \Big)^{-1} \boldsymbol{X}^T\boldsymbol{Y}. \tag{5}$$

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, Section 9.3.6).

Computing (5) is extremely quick, even for a relatively large number, say 100, values of α, especially if one uses the diagonalize algorithm discussed below. For a fixed number of knots, K, the computational time for the matrices $\boldsymbol{X}^T\boldsymbol{X}$ and $\boldsymbol{X}^T\boldsymbol{Y}$ is linear in the sample size n, but these matrices need only be computed once. As Eilers and Marx (1996) mention, 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

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

be the average squared residuals using α. Let

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

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

$$\mathrm{GCV}(\alpha) = \frac{\mathrm{ASR}(\alpha)}{\left[1 - n^{-1}\mathrm{tr}\{\boldsymbol{S}(\alpha)\}\right]^2} \tag{6}$$

is the generalized cross validation statistic. Here $\mathrm{tr}\{\boldsymbol{S}(\alpha)\}$ is the "effective degrees of freedom" of the fit.

One chooses α by computing $\mathrm{GCV}(\alpha)$ 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 $\log(\alpha)$ are equally spaced between -12 and 12. (If GCV is minimized at either endpoint of this grid, then clearly the grid should be expanded at that end.)

Computation can be sped up and stabilized numerically 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

$$(\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X} + \alpha\boldsymbol{D})\widehat{\boldsymbol{\beta}} = \boldsymbol{X}^{\mathsf{T}}\boldsymbol{Y}.$$

where $\boldsymbol{D}$ is a non-negative definite symmetric matrix. ($\boldsymbol{D}$ need not be diagonal, though it would be for a P-spline.) We use the notation $A^{-T} = (A^{-1})^{\mathsf{T}}$. Then we have the following result.

Theorem 1: *Let $\boldsymbol{B}$ be a square matrix satisfying $\boldsymbol{B}^{-1}\boldsymbol{B}^{-\mathsf{T}} = \boldsymbol{X}^{\mathsf{T}}\boldsymbol{X}$, e. g., $\boldsymbol{B}^{-1}$ is a Cholesky factor of $\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X}$. Let $\boldsymbol{U}$ be orthogonal and let $\boldsymbol{C}$ be diagonal such that $\boldsymbol{U}\boldsymbol{C}\boldsymbol{U}^{\mathsf{T}} = \boldsymbol{B}\boldsymbol{D}\boldsymbol{B}^{\mathsf{T}}$.*

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

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

$$(\mathbf{I} + \alpha\boldsymbol{C})\widehat{\boldsymbol{\lambda}} = \boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Y} = (\boldsymbol{U}^{\mathsf{T}}\boldsymbol{B})\boldsymbol{X}^{\mathsf{T}}\boldsymbol{Y}. \tag{7}$$

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

$$\mathrm{tr}(\boldsymbol{S}(\alpha)) = \mathrm{tr}[(\mathbf{I} + \alpha\boldsymbol{C})^{-1}\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z}] = \mathrm{tr}(\mathbf{I} + \alpha\boldsymbol{C})^{-1} = \sum_i (1 + \alpha C_i)^{-1}, \tag{8}$$

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

The beauty of this method of calculating the P-Spline is that the work of calculating $\boldsymbol{B}$, $\boldsymbol{B}^{-1}$, $(\boldsymbol{U}, \boldsymbol{C})$ and $\boldsymbol{Z}$ 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 (7) and then the spline fit is $\boldsymbol{Z}\widehat{\boldsymbol{\lambda}}$. Moreover, computing $\mathrm{tr}(\boldsymbol{S}(\alpha))$ by (8) 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.

Proof of Theorem 1:

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

so that

$$\boldsymbol{U}^{\mathsf{T}}\boldsymbol{B}[\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X} + \alpha\boldsymbol{D}]\boldsymbol{B}^{\mathsf{T}}\boldsymbol{U} = \mathbf{I} + \alpha\boldsymbol{C}.$$

Thus,

$$\boldsymbol{U}^{\mathsf{T}}\boldsymbol{B}[(\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X} + \alpha\boldsymbol{D})\boldsymbol{B}^{\mathsf{T}}\boldsymbol{U}(\boldsymbol{U}^{\mathsf{T}}\boldsymbol{B}^{-1}\widehat{\boldsymbol{\beta}})] = \boldsymbol{U}^{\mathsf{T}}\boldsymbol{B}(\boldsymbol{X}^{\mathsf{T}}\boldsymbol{Y})$$

or

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

Also, $\boldsymbol{X} = \boldsymbol{Z}\boldsymbol{U}^{\mathsf{T}}\boldsymbol{B}^{-\mathsf{T}}$ so that

$$\boldsymbol{X}\widehat{\boldsymbol{\beta}} = \boldsymbol{Z}\boldsymbol{U}^{\mathsf{T}}\boldsymbol{B}^{-\mathsf{T}}\widehat{\boldsymbol{\beta}} = \boldsymbol{Z}\widehat{\boldsymbol{\lambda}}$$

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 4 where increasing K above five knots leads to as much as an 18%, respectively 40%, increase in MSE over the value at five knots.

Thus, the goal for any algorithm for selecting K is to make certain that K is sufficiently large to fit the data and not so large that computation time is unnecessarily large.

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. We use 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 $\{x_i\}$ 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 decreases GCV and we use $K = 10$. Otherwise, one computes the P-spline fit with $K = 20$ and compares GCV for $K = 10$ and with GCV for $K = 20$ in the same way one compared GCV for $K = 5$ and 10. One stops and uses $K = 20$ if GCV at $K = 20$ exceeds .98 times GCV at $K = 10$. Otherwise, one computes the P-spline at $K = 40$, etc. The algorithm is called "myoptic" 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 myoptic algorithm has the advantage that it 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 myoptic 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 myoptic 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.

4 Simulations

We consider eleven examples where m, σ, and n vary as shown in Table 1. In all cases, the x_i are equally spaced on [0, 1]. In this study, we only use quadratic splines ($p = 2$). The reason for this restriction is that quadratic splines work very well in practice when m is smooth.

For functions with discontinuities or "kinks" where the first derivative is continuous, $p = 0$ or 1 is preferred to $p = 2$, but we have not included such functions 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 two examples, called "Logit" and "LogitLN," use a logistic function

$$m(x) = \frac{1}{1 + \exp\{-20(x - .5)\}} \tag{9}$$

and differ only in the value of σ. (LN = "Low Noise").

The third and fourth examples, called the "Bump" and "BumpLN," use

$$m(x) = x + 2\exp[-\{16(x - .5)\}^2], \tag{10}$$

and again differ in that BumpLN has σ only one tenth as large as Bump.

The next four examples, "Sine3," "Sine3LN," "Sine6," and "Sine12," are sine waves with θ cycles where θ is 3, 3, 6, and 12, respectively. Sine3LN is a low noise version of Sine3.

The final example, "SpaHet3," has a spatially heterogeneous function used in Wand (1997):

$$m(x; j) = \sqrt{x(1 - x)}\sin\left\{\frac{2\pi(1 + 2^{(9-4j)/5)}}{x + 2^{(9-4j)/5}}\right\}. \tag{11}$$

The parameter j controls the amount of spatially heterogeneity and in this example $j = 3$ which give only a slight amount of heterogeneity.

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 we simulated 300 data sets. We compare P-splines with different fixed values of K to the myoptic and full-search algorithms using MASE (mean average squared error) defined as the mean over the 300 data sets of the average squared error,

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

Figures 1–8 show a typical data set in panel (a), MASE comparisons in panel (b), and histograms of K as selected by the myoptic and full-search algorithms in panels (c) and (d) for eight of the eleven examples. To save space, examples LogitLN, Sine3LN, and Sine12, were not included in the figures but are discussed below. 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 myoptic and full-search estimators. In each panel (b), relative MASE is the MASE divided by MASE of the oracle estimator.

4.1 Results

Logit and LogitLN:

Since $n = 75$ in these two cases and only values of 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 for Logit, MASE rises monotonically as K increases from 5 to 40 and is about 20% higher at 40 than at 5. The myoptic algorithm cannot choose $K = 5$ because of the way it is defined but it does choose $K = 10$ about 90% of the time. The full search algorithm chooses K equal to 5 more than half the time, but surprisingly the full search algorithm has a 6% higher MASE than the myoptic algorithm, perhaps because the full search algorithm chooses 40 knots more than the myoptic algorithm does (compare Figures 1(c) and 1(d)).

For LogitLN, as for Logit, 5 knots is optimal, but now 20 or 40 knots is better than 10; $10^5\times$ MASE equals 5.1, 8.0, 6.6, and 6.9 for 5, 10, 20, and 40 knots, respectively. The myoptic policy selects 10 knots, the worst choice in terms of MASE, in about 92% of the samples, 20 knots about 7% of the time, and 40 knots for about 1% of the samples. The full search algorithm selects 5 knots almost two thirds of the time and has a lower MASE than either the myoptic algorithm or using a fixed number, 10, 20, or 40, knots.

These examples are similar to many found in practice where the regression function is monotonic. In such situations, the number of knots is not very important as long as there are at least five and a fixed but nearly arbitrary number of knots seems as sensible as using a GCV-driven search.

Bump and BumpLN:

For both Bump and BumpLN shown in Figures 2 and 3, 5 knots is clearly not enough. For Bump, 10, 20, 40 and 80 knots all have similar MASE values. In the low noise case of BumpLN, 10 knots has a far higher MASE value than 20 or more knots and MASE is minimized by 40 or more knots. Both the myoptic and full search algorithm select enough knots to have MASE values near optimal.

These examples should be similar to those in practice with a unimodal regression function. The minimum correct number of knots depends on the SN ratio and both search algorithms select a value of K above this minimum.

Sine3 and Sine3LN:

In the higher noise case of Sine3 (Figure 4), MASE increases monotonically as K increases through 5, 10, 20, 40, 80, and finally 120. Both the myoptic and full search algorithm select low numbers of knots and the full search algorithm selects five knots more that two thirds of the time. 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.

For low noise at least 20 knots are needed; $10^3\times$ MASE is about 2.3 for 5 or 10 knots and drops to about 1.2 for 40, 80, or 120 knots. Unfortunately, the myoptic algorithm prematurely stops at least two thirds of the time at 10 knots, because 10 knots does not improve over 5 knots. The full search algorithm performs much better than the myoptic algorithm, though it too occasionally chooses 5 or 10 knots. $10^3\times$ MASE is about 1.9 for the myoptic policy and 1.2 for the full search policy.

Sine6:

For a six-cycle sine wave (Figure 5), at least 20 knots are needed for a satisfactory MASE and the full search algorithm always selects at least 20 knots. The myoptic search occasionally stops prematurely at 10 knots.

Sine12:

For a twelve-cycle sine wave 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 myoptic search stops prematurely at 10 knots in every one of the 300 Monte Carlo samples.

This is a nice illustration of the potential pitfall of the myoptic 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, e.g., one has collected hourly data for 12 days. In such cases, the investigator would know not to use the myoptic 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 twelve periods.

SpaHet3, SpaHet3LS, SpaHe3VLS:

For SpaHet3 (Figure 6) the number of knots has little effect and any fixed number of knots greater than five works well, as do the two automatic algorithms. SpaHet3LS and SpaHet3VLS, shown in Figures 7 and 8, are “large sample” and “very large sample” versions of SpaHet3 where all parameters are the same except that n of 200 is raised to 2,000 and 10,000, respectively. One can see that for larger sample sizes, five knots is not quite enough, but ten knots is adequate for even $n = 10,000$. 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 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 myoptic 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, we feel that twenty knots can be recommended for routine use in such disciplines, even for very large sample sizes. Of course, there will be exceptions, e.g., long periodic time series.

5 Additive Models

An attractive generalization of the multiple linear regression model is the additive model (Hastie and Tibshirani, 1990). Suppose that we for the ith observation we observe L predictor variables, $x_{1,i}, \ldots, x_{L,i}$. The additive model is

$$y_i = \beta_0 + m_1(x_{1,i}) + \cdots + m_L(x_{L,i}) + \epsilon_i. \tag{12}$$

We will use a spline model for each m_l:

$$m_l(x_l; \boldsymbol{\beta}_l) = \beta_{l,1} x_l + \cdots + \beta_{l,p} x_l^p + \sum_{k=1}^{K_l} b_{l,k}(x_l - \kappa_k)_+^p. \tag{13}$$

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 (13), model (12) with m_l given by (13) is identifiable. Moreover, direct fitting of additive spline models is straightforward (Marx and Eilers, 1998).

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

$$\sum_{i=1}^{n} \Big\{ y_i - m(x; \boldsymbol{\beta}) \Big\}^2 + \sum_{l=1}^{L} \alpha_l \sum_{k=1}^{K_l} \beta_{l,p+k}^2, \tag{14}$$

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) discuss an algorithm for chosing $\{\alpha_l\}_{l=1}^L$ by GCV and they compare 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 5.1.

I recommend using a common value of K_l, which we will call K as before. The reason we make 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 chose 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, we simulated data where $L = 3$, $n = 150$, 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 $\sigma = .25$ 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 9 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.

5.1 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, i.e., 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, etc. 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 $\boldsymbol{X}$ be the $n \times \{1 + L(p + K)\}$ matrix whose nth row is the set of basis functions for model (12) and (13) 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 $\boldsymbol{D}_l$ be the diagonal matrix with diagonals elements equal to either zero or one such that $\sum_{k=1}^{K} b_{l,k}^2 = \boldsymbol{\beta}^{\mathsf{T}} \boldsymbol{D}_l \boldsymbol{\beta}$.

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 $\boldsymbol{D} = \boldsymbol{D}_1 + \cdots + \boldsymbol{D}_L$. In step 2 suppose that we want to find the value of α_{l^*} that minimizes GCV with the other α_l fixed. Then we find a square $\boldsymbol{B}$ such that $\boldsymbol{B}^{-1}\boldsymbol{B}^{-\mathsf{T}} = \{\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X} + \sum_{l \neq l^*} \alpha_l \boldsymbol{D}_l\}$ so that $\widehat{\boldsymbol{\beta}}$ solves $(\boldsymbol{B}^{-1}\boldsymbol{B}^{-\mathsf{T}} + \alpha_{l^*}\boldsymbol{D}_{l^*})\widehat{\boldsymbol{\beta}} = \boldsymbol{X}^{\mathsf{T}}\boldsymbol{Y}$. With $\boldsymbol{B}$ 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 $\boldsymbol{D}$ of the univariate spline replaced by α_{l^*} and $\boldsymbol{D}_{l^*}$.

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

6 Discussion and Conclusions

In summary, in some cases, e.g., the monotonic Logit example, the number of knots has little effect. In other cases, there is a miminal number of knots and a low MASE value is achieved whenever the number of knots exceeds this value. The minimal value of K needed for a good fit depends on the SN ratio and is higher for higher SN ratios.

In the experiments reported here, the full search algorithm selected reasonable values of K without fail. The myoptic algorithm generally works well but can stop prematurely in some examples.

A researcher with some knowledge of the shape of m should 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 myoptic algorithm. The myoptic 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) states that "my current defult is a knot between every d observations, where $d = \max(4, \lfloor n/35 \rfloor)$. ($\lfloor r \rfloor$ is the floor of r, i.e., the greatest integer less than or equal to r.) Wand's default chooses roughly $\min(n/4, 35)$ knots. Since the algorithms in this paper 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.

For the cases in Table 1 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 full search algorithm is most likely to choose 5 knots and has a MASE only 25% greater than a 5-knot spline; see Figure 4 (b).

It may seem surprising that a default that uses at most 35 (or 40) knots could be recommended *for effectively all sample sizes.* To see that this is true, consider an example with n extremely large. The mean function in this example is (11) 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 we will see, 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 10. 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 virtually impossible to distinguish. One can see that a 35-knot spline approximates this rather complex regression function nearly perfectly.

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, i.e., 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.

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 his helpful comments.

REFERENCES

Eilers, P.H.C., & 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., & Silverman, B.W. (1989), "Flexible parsimonious smoothing and additive modeling (with discussion)," *Technometric*, 31, 3–39.

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

Marx B.D., & 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, Wiley, New York 393–424.

O'Sullivan, F. (1986), "A statistical perspective on ill-posed inverse problems (with discussion)," *Statistical Science*, 1, 505–527.

O'Sullivan, F. (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. (1999). Spatially-adaptive penalties for spline fitting. *Australian and New Zealand Journal of Statistics*, 42, 205–223.

Stone, C.J., Hansen, M., Kooperberg, C., & Truong, Y. K. (1997). "Polynomial splines and their tensor products in extended linear modeling (with discussion)," *The Annals of Statistics*, 25, 1371–1470.

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

Table 1: Parameter values for the nine case studies.

Name	m	σ	n	SN ratio
Logit	(9)	0.2	75	5.08
LogitLN	(9)	0.02	75	508
Bump	(10)	0.3	100	3.89
BumpLN	(10)	0.03	100	389
Sine3	$\sin(2\pi f)$, $\theta = 3$	1	150	0.50
Sine3LN	$\sin(2\pi f)$, $\theta = 3$	0.1	150	50
Sine6	$\sin(2\pi f)$, $\theta = 6$	0.5	150	2
Sine12	$\sin(2\pi f)$, $\theta = 12$	0.25	150	8
SpaHet3	(11) with $j = 3$	0.3	200	0.88
SpaHet3LS	(11) with $j = 3$	0.3	2,000	0.88
SpaHet3VLS	(11) with $j = 3$	0.3	10,000	0.88

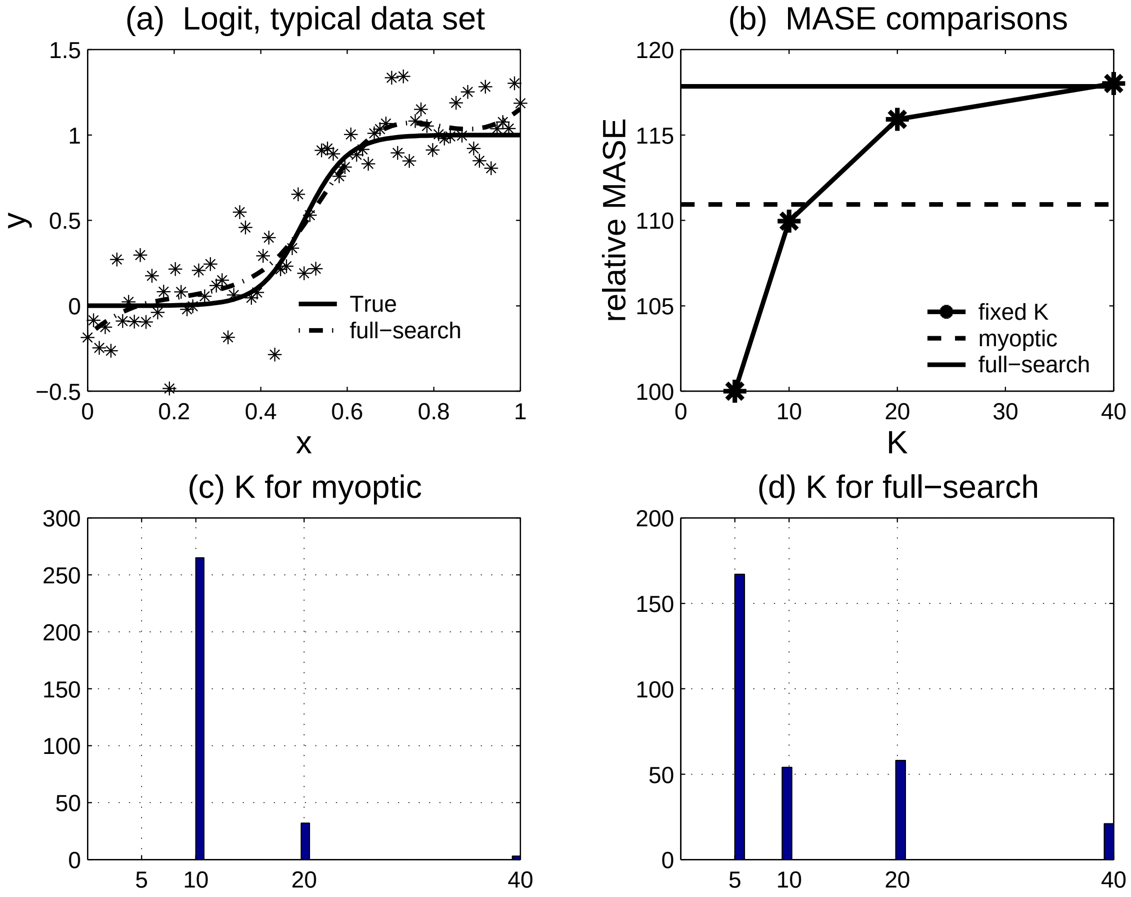

Figure 1: *Logit example. Simulation of 300 data sets. (a) Typical data set, true regression function, and estimate from the full-search algorithm. (b) Mean average squared error (MASE) as a function of K with horizontal lines through the values of MASE for the myoptic and full-search algorithms. (c)–(d) Histograms of K as chosen by the myoptic and full-search algorithms, respectively.*

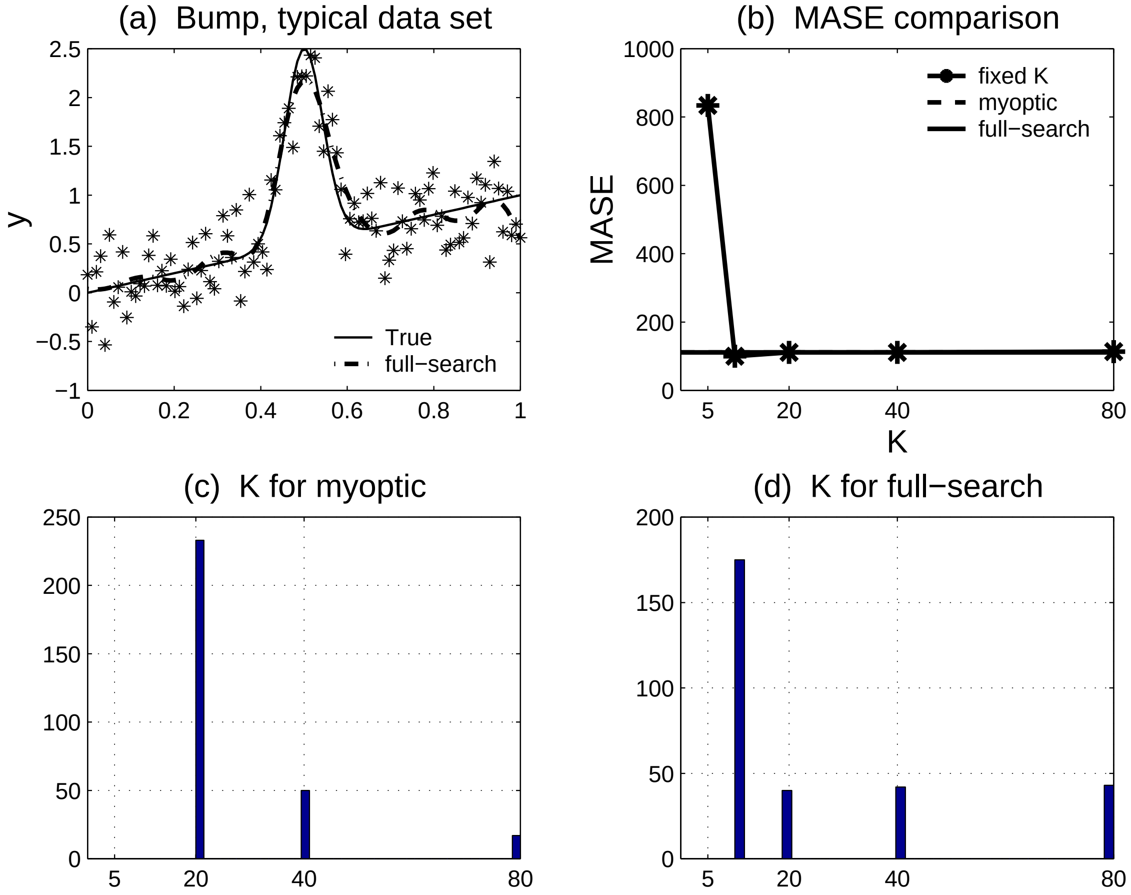

Figure 2: *Bump function example. (a)–(d) as in Figure 1.*

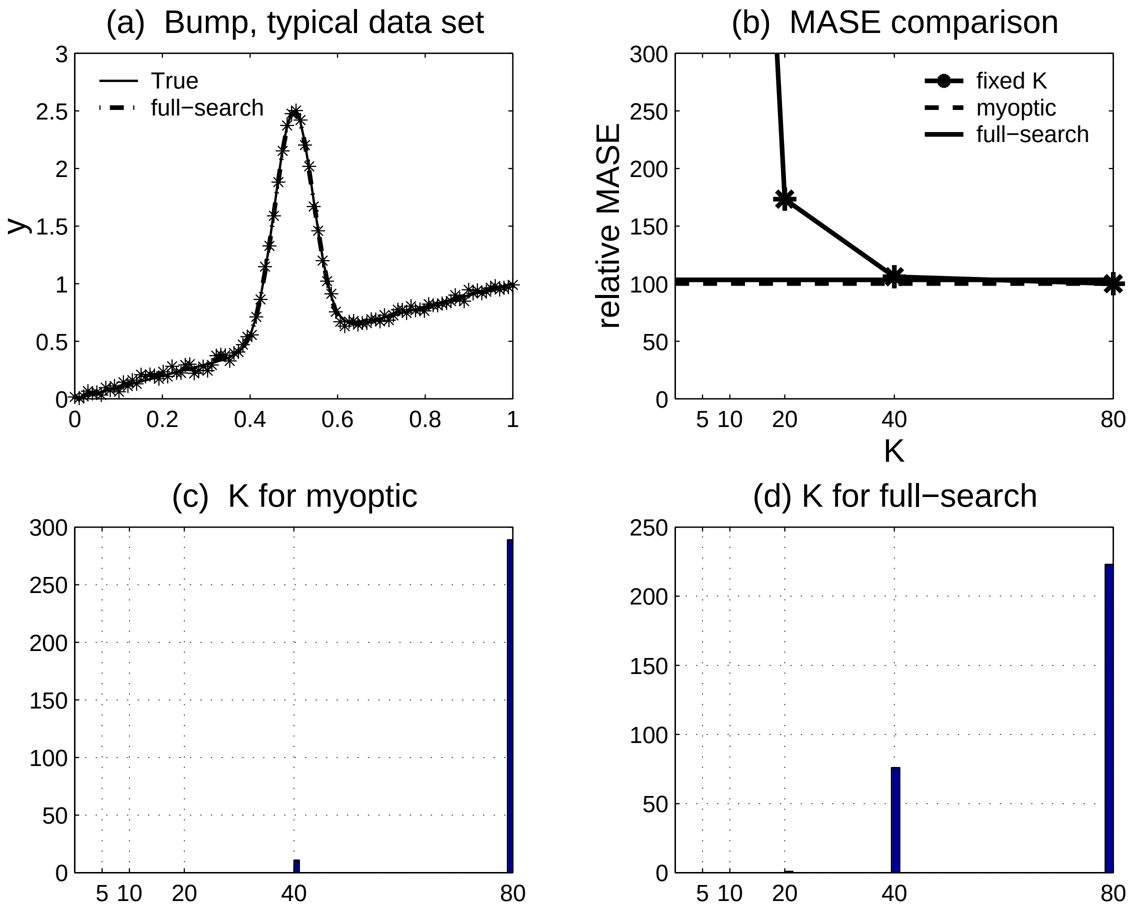

Figure 3: *Bump function, low noise, example. (a)–(d) as in Figure 1.*

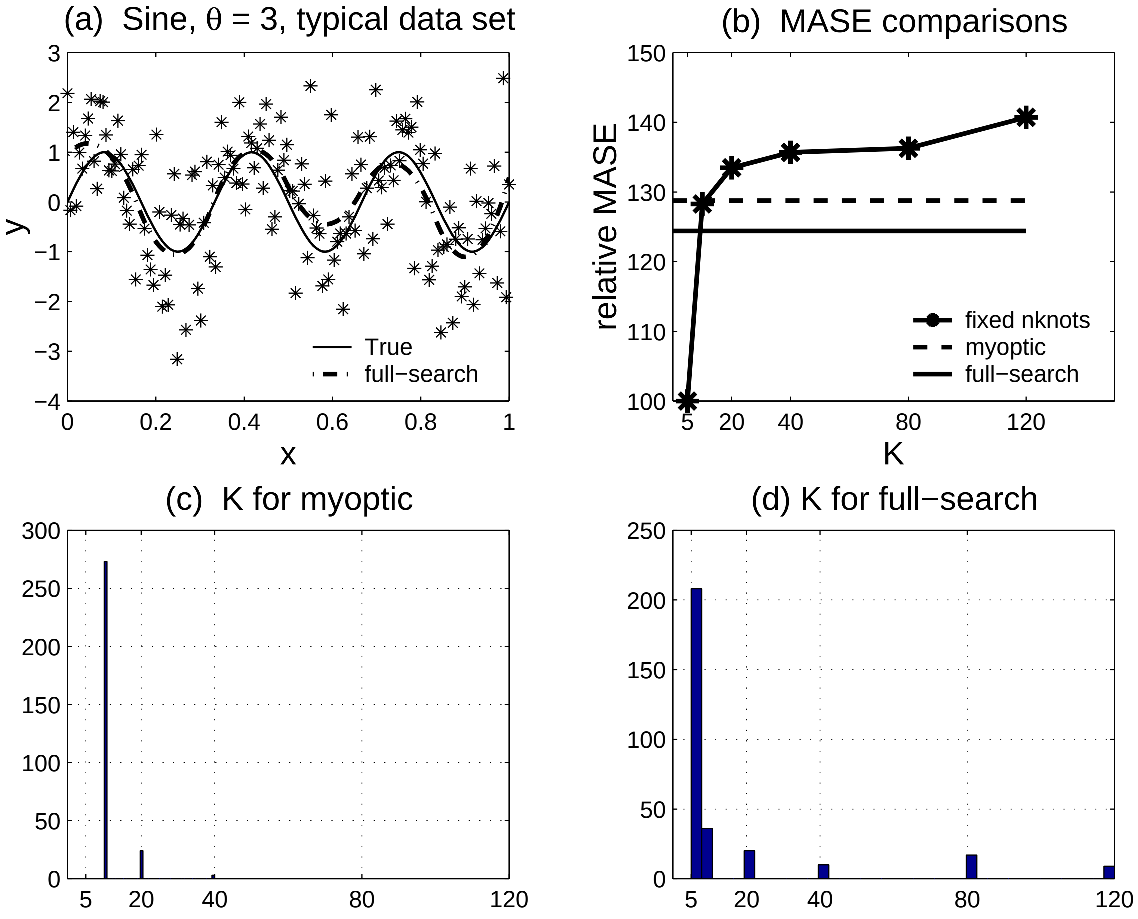

Figure 4: *Sine3 example. (a)–(d) as in Figure 1.*

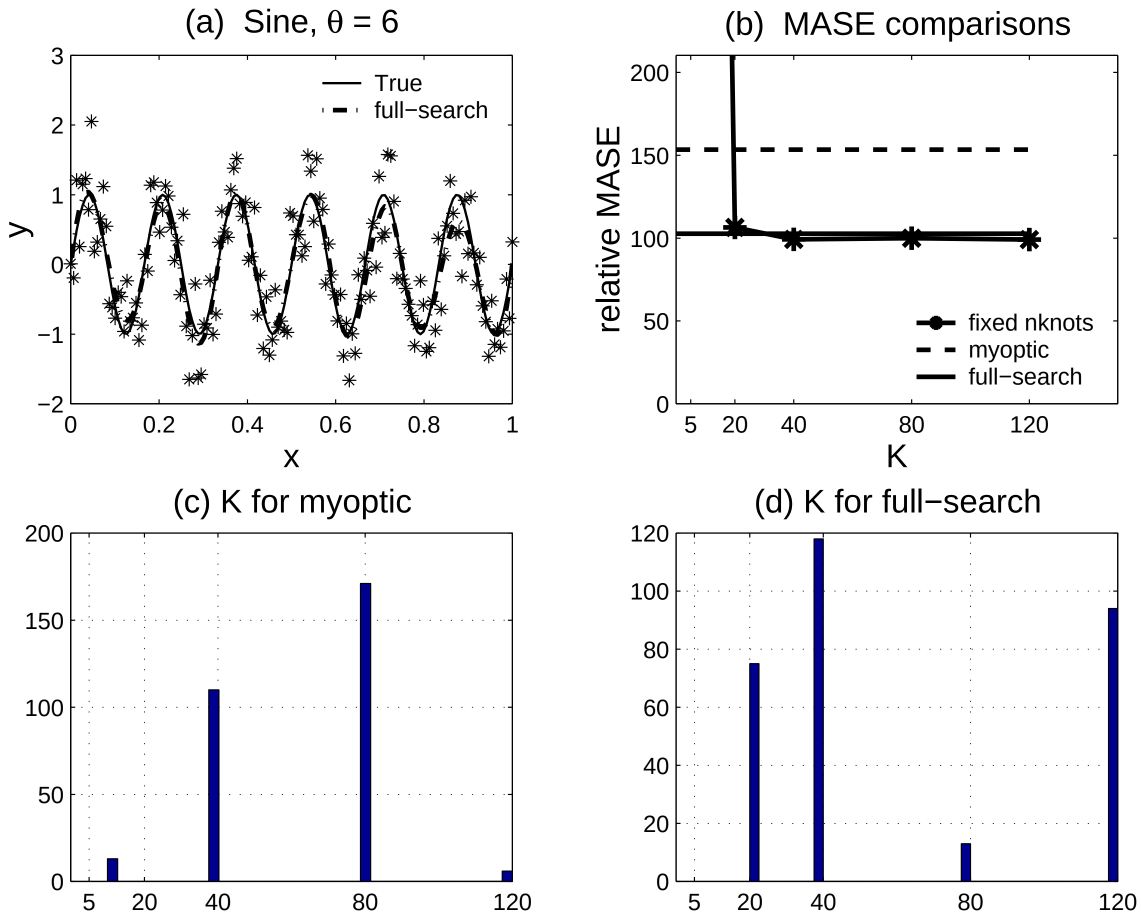

Figure 5: *Sine6 example. (a)–(d) as in Figure 1.*

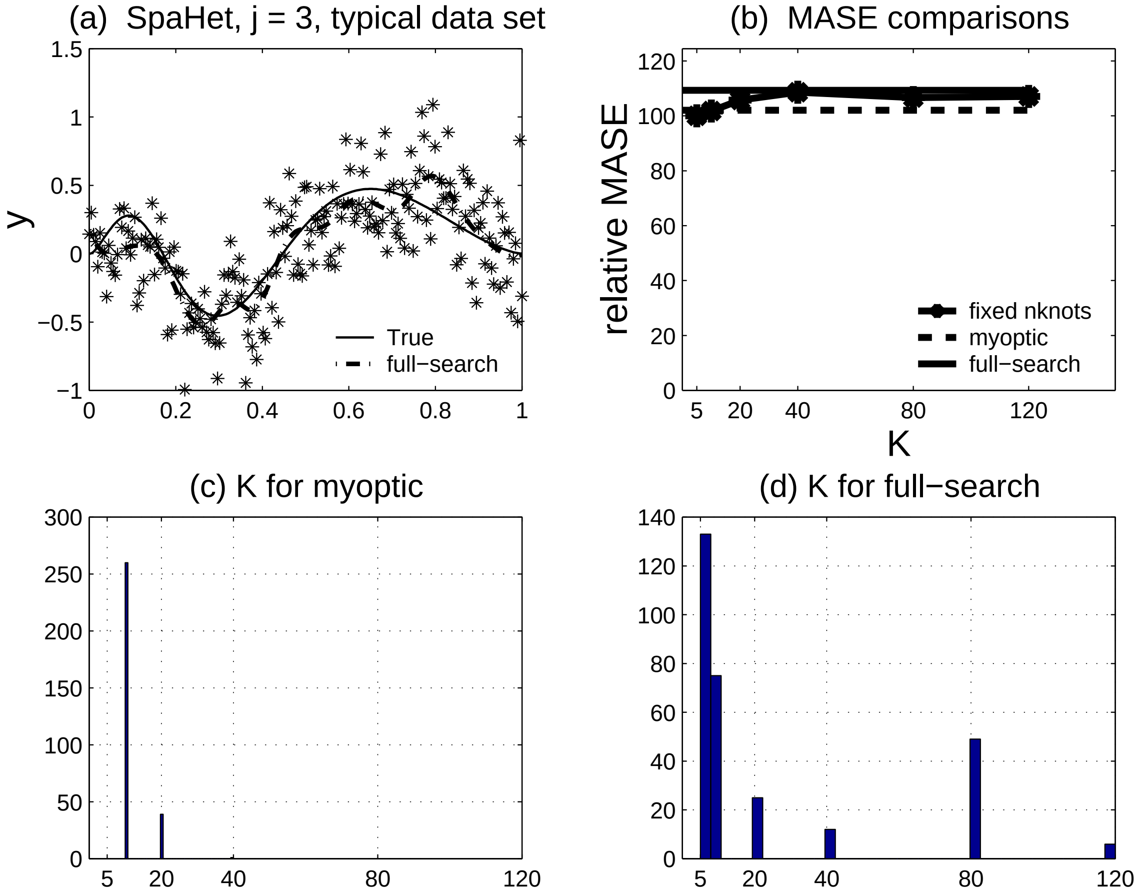

Figure 6: *Spatial heterogeneity example. (a)–(d) as in Figure 1.*

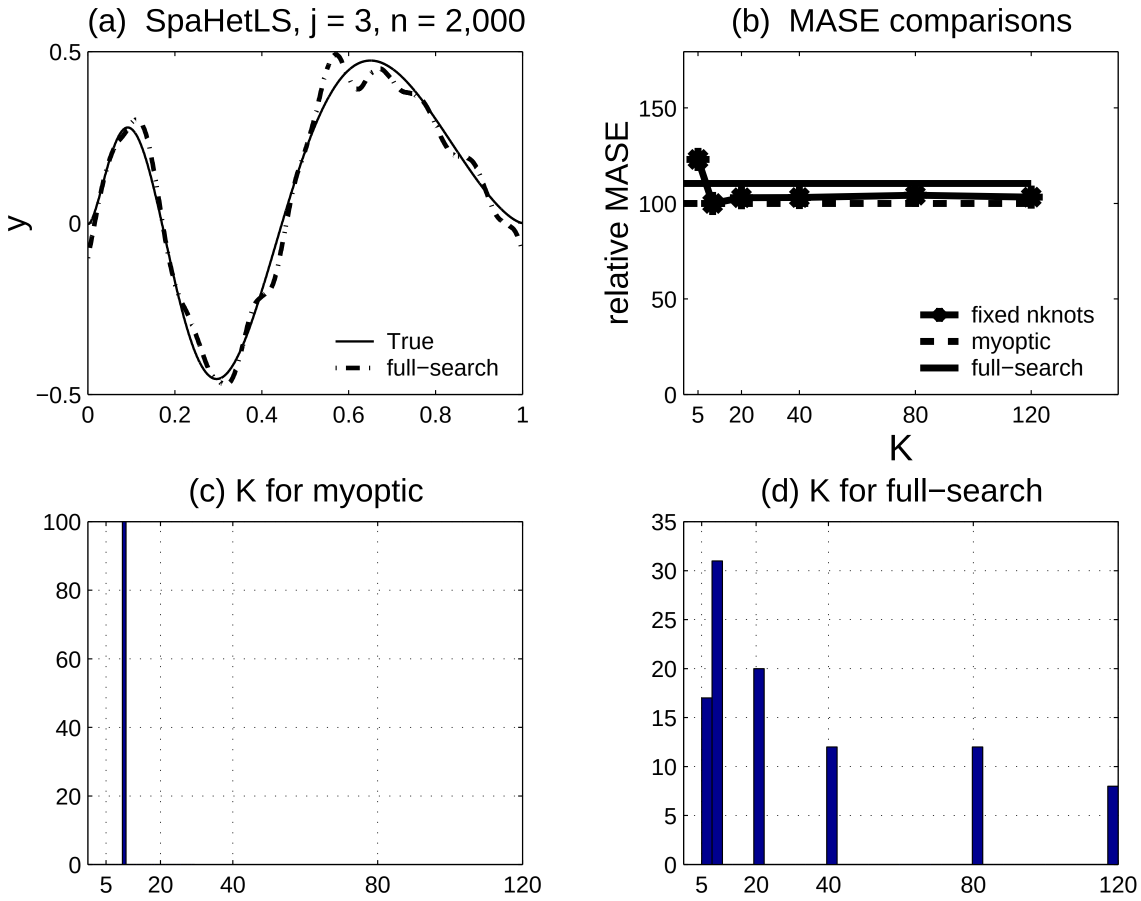

Figure 7: *Spatial heterogeneity, large sample, example. (a)–(d) as in Figure 1.*

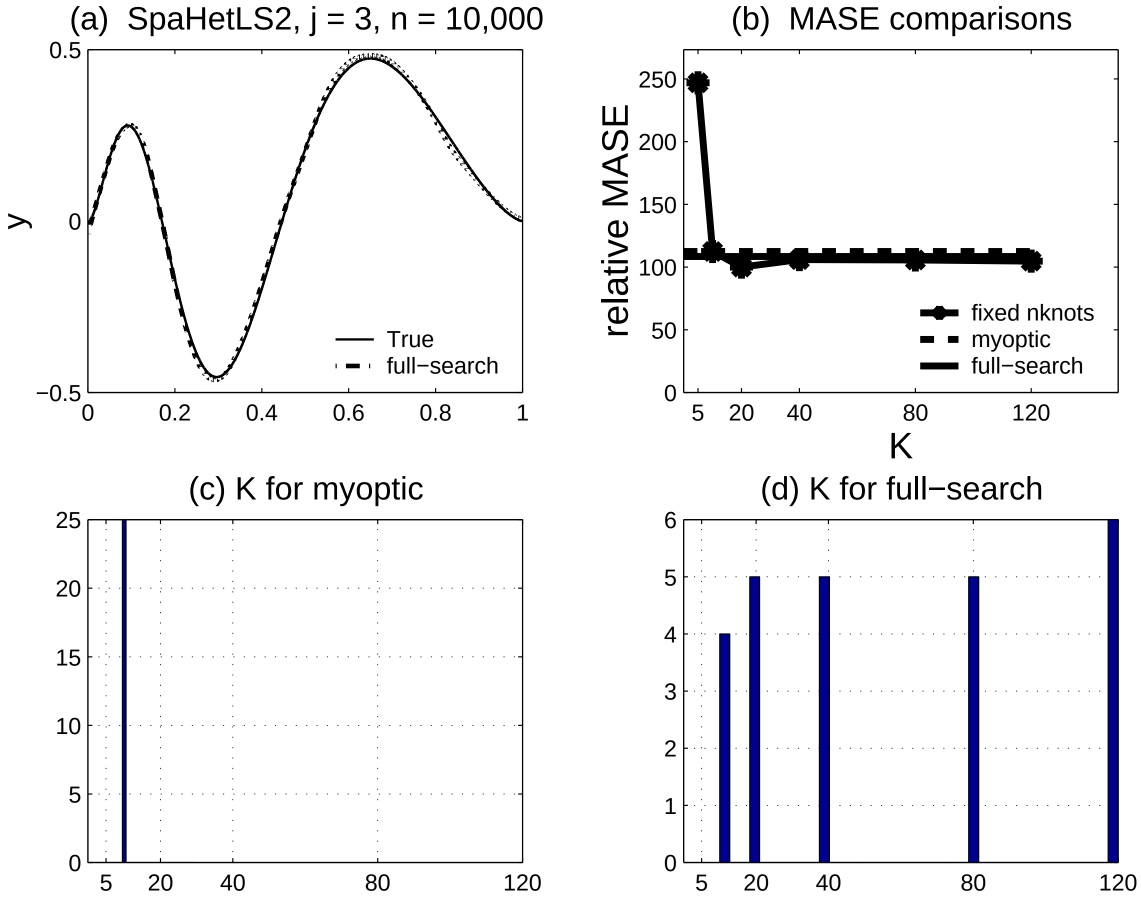

Figure 8: *Spatial heterogeneity, very large sample, example. (a)–(d) as in Figure 1.*

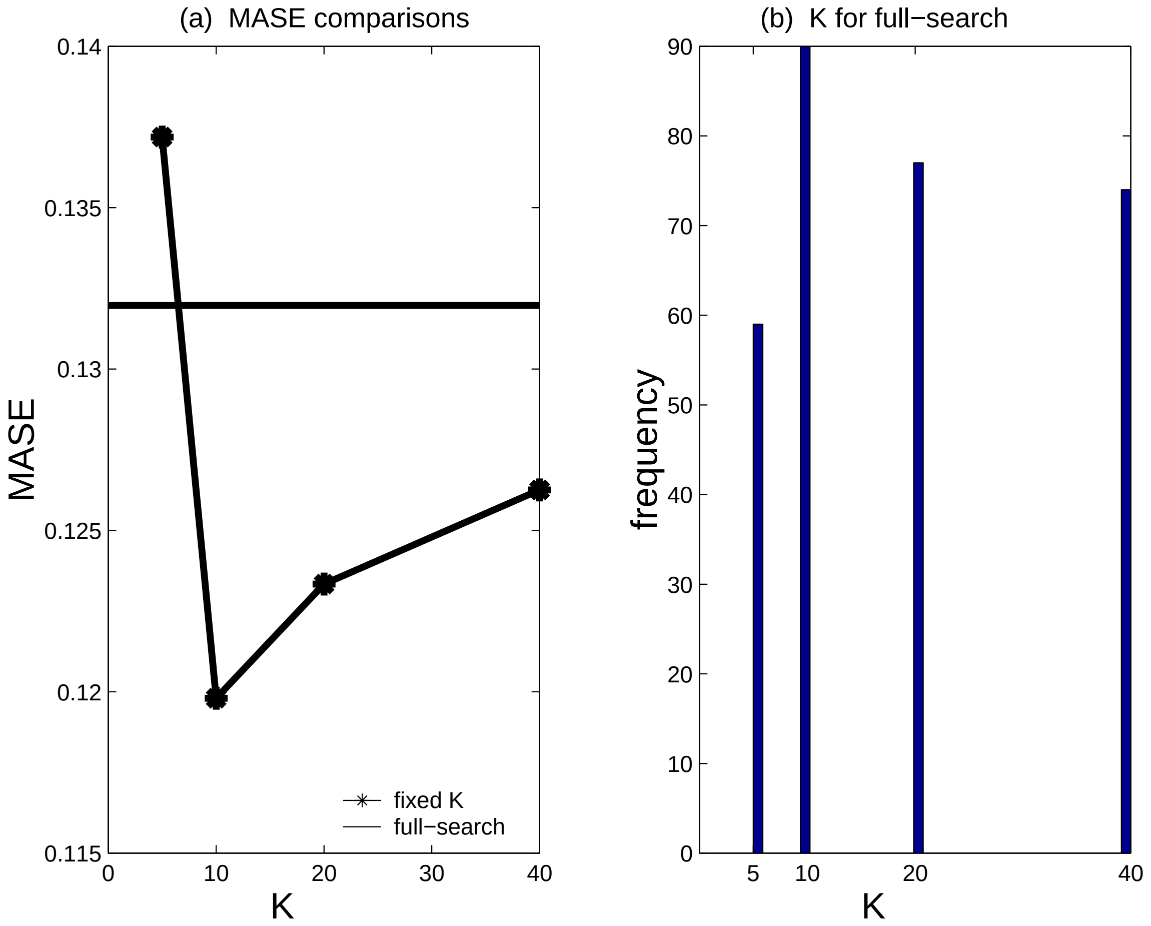

Figure 9: *Additive model with Sine3 as first component and $\sigma = 1$. (a) MASE. (b) Histogram of values of K selected by the full search algorithm.*

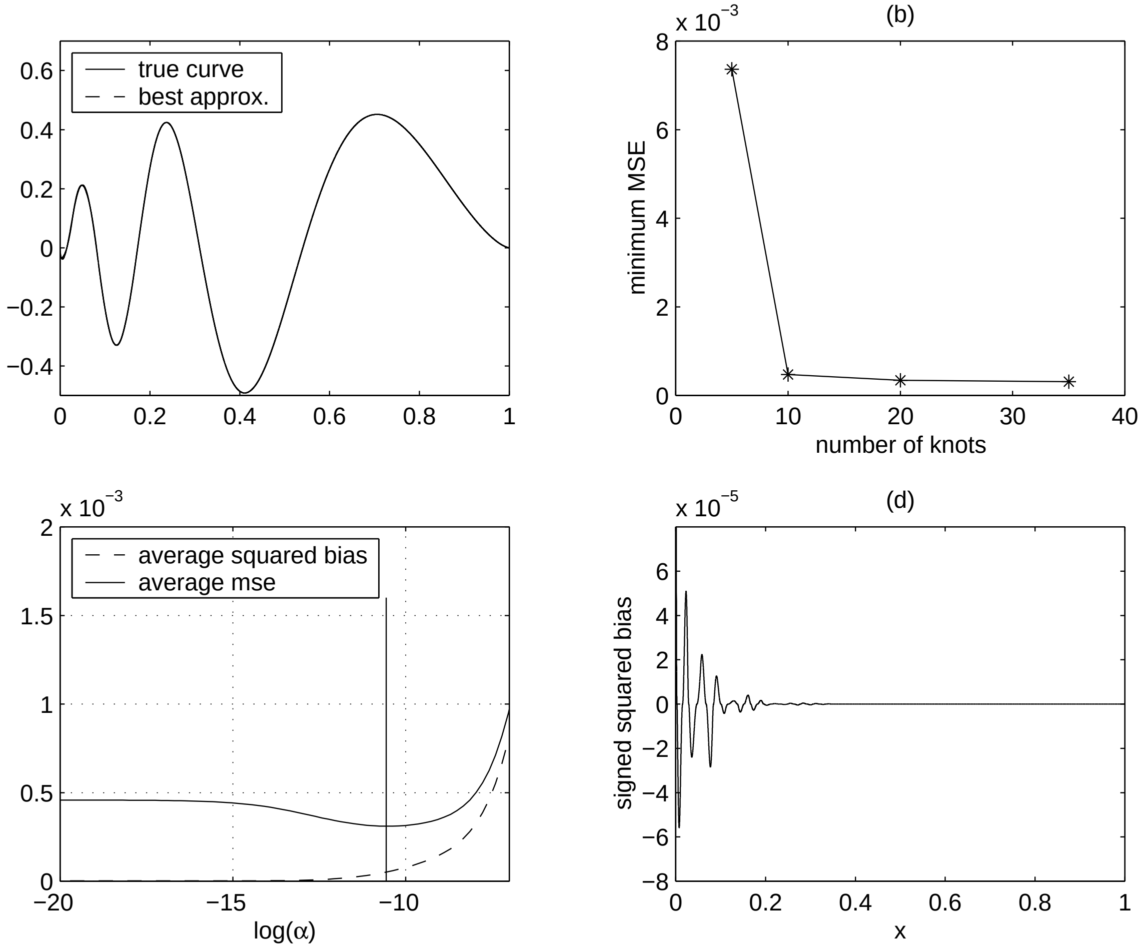

Figure 10: *Study of bias with* $n = 25,000$*,* $\sigma^2 = 0.3$*, and the regression function given by (11). (a) Regression and best least-squares approximation by a 35-knot quadratic spline. The two curves are too close to be easily distinguished. (b) Average MSE minimized over* α *as function of the 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* $\alpha = 0$*.*