Regression Spline Smoothing using the Minimum Description Length Principle*

Thomas C. M. Lee

Department of Statistics, Colorado State University, Fort Collins, CO 80523-1877, USA

April 12, 1999; revised: June 17, 1999

Abstract

One approach to estimating a function nonparametrically is to fit an rth order regression spline to the noisy observations, and one important component of this approach is the choice of the number and the locations of the knots. This article proposes a new regression spline smoothing procedure which automatically chooses: (i) the order r of the regression spline being fitted; (ii) the number of the knots; and (iii) the locations of the knots. This procedure is based on the minimum description length principle, which is rarely applied to choose the amount of smoothing in nonparametric regression problems.

Keywords: automatic knot selection; minimum description length; regression spline smoothing

1 Introduction

This article considers the problem of estimating a function nonparametrically. Many approaches to this problem have been proposed in the literature. These include kernel/local polynomial regression, smoothing spline methods, regression spline smoothing and wavelet techniques. The approach with which this article is concerned is regression spline smoothing.

An important aspect associated with regression spline smoothing is the choice of the number and the placement of the knots. Inadequate number of knots or badly placed knots would lead to oversmoothing in some regions of the underlying true function, while too many knots would inflate local variance. This article proposes an automatic procedure for simultaneously selecting the number and the placement of the knots. In addition, this procedure is capable of automatically

*To appear in *Statistics & Probability Letters*.

selecting the order (e.g., linear or cubic) of the regression spline being fitted. The procedure is based on Rissanen's minimum description length (MDL) principle (e.g., see Rissanen 1989), and consists of two components: (i) an MDL–based criterion in which the "best" function estimate is defined as its minimizer and (ii) a knot deletion algorithm which attempts to locate this minimizer.

Various non–MDL based regression spline smoothing procedures have been proposed in the literature. They are chiefly based on cross–validation or Bayesian approaches: Friedman & Silverman (1989), Smith & Kohn (1996), Luo & Wahba (1997) and Denison, Mallick & Smith (1998). Notice that most of these procedures fix the order of the spline *a priori*.

2 Nonparametric Regression as Model Selection

Suppose that n pairs of measurements $\{x_i, y_i\}_{i=1}^n$, $y_i = f(x_i) + \epsilon_i$, $\epsilon_i \sim$ iid $N(0, \sigma^2)$, are observed. The aim is to estimate f which is assumed to be "smooth". To be specific, it is assumed that f can be well approximated by a rth order regression spline with m knots:

$$f(x) \approx b_0 + b_1 x + \ldots + b_r x^r + \sum_{j=1}^{m} \beta_j (x - k_j)_+^r.$$

Here k_j is the location of the jth knot, $\{b_0, \ldots, b_r, \beta_1, \ldots, \beta_m\}$ is a set of coefficients and $(a)_+ = \max(0, a)$. It is also assumed that $\min(x_i) < k_1 < \ldots < k_m < \max(x_i)$, and that $\{k_1, \ldots, k_m\}$ is a subset of $\{x_1, \ldots, x_n\}$.

If f admits the above regression spline representation, then an estimate $\hat{f}$ of f can be obtained via estimating r, m, $\boldsymbol{k} = (k_1, \ldots, k_m)^T$, $\boldsymbol{b} = (b_0, \ldots, b_r)^T$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_m)^T$:

$$\hat{f}(x) = \hat{b}_0 + \hat{b}_1 x + \ldots + \hat{b}_{\hat{r}} x^{\hat{r}} + \sum_{j=1}^{\hat{m}} \hat{\beta}_j (x - \hat{k}_j)_+^{\hat{r}},$$

where $\hat{r}$, $\hat{m}$, $\hat{\boldsymbol{k}}$, $\hat{\boldsymbol{b}}$ and $\hat{\boldsymbol{\beta}}$ are estimates of r, m, $\boldsymbol{k}$, $\boldsymbol{b}$ and $\boldsymbol{\beta}$ respectively. Thus, one can see that by approximating f with a regression spline, the problem of estimating f can be transformed into a model selection problem, with each plausible model $\boldsymbol{\theta}$ completely specified by $\boldsymbol{\theta} = \{r, m, \boldsymbol{k}, \boldsymbol{b}, \boldsymbol{\beta}\}$. Notice that different $\boldsymbol{\theta}$'s may have different dimensions (number of parameters), and we shall use the MDL principle to pick the "best" model.

We make the following remark before we proceed. Let $\boldsymbol{x} = (x_1, \ldots, x_n)^T$, $\boldsymbol{y} = (y_1, \ldots, y_n)^T$ and $\boldsymbol{X} = (\boldsymbol{1}, \boldsymbol{x}, \ldots, \boldsymbol{x}^{\hat{r}}, (\boldsymbol{x} - \hat{k}_1 \boldsymbol{1})_+^{\hat{r}}, \ldots, (\boldsymbol{x} - \hat{k}_{\hat{m}} \boldsymbol{1})_+^{\hat{r}})$, where $\boldsymbol{1}$ is a $n \times 1$ vector of ones. If $\hat{r}$, $\hat{m}$ and $\hat{\boldsymbol{k}}$ are specified beforehand, then natural estimates of $\boldsymbol{b}$ and $\boldsymbol{\beta}$ are given by:

$$(\hat{\boldsymbol{b}}^T, \hat{\boldsymbol{\beta}}^T)^T = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{y}. \tag{1}$$

Observe that these are the maximum likelihood estimates of $\boldsymbol{b}$ and $\boldsymbol{\beta}$ (conditional on $\hat{r}$, $\hat{m}$ and $\hat{\boldsymbol{k}}$).

3 Model Selection by the MDL Principle

The MDL principle provides a powerful methodology for attacking model selection problems. Briefly, it *defines* the best model as the one that enables the best encoding (or compression) of the data, so that the data can be transmitted in the most economical way. That is, the best fitted model is the one that produces the shortest code length of the data. Typically the code length for a set of data can be split into two parts: (i) a fitted model plus (ii) the data "conditioning on" the fitted model; i.e., the residuals. In the present situation, the data are $\boldsymbol{x}$ and $\boldsymbol{y}$, and a fitted model is simply an estimated $\boldsymbol{\theta}$, to be denoted by $\hat{\boldsymbol{\theta}}$ below.

Thus, in order to apply the MDL principle to tackle the current problem, we first need to construct a code length expression which calculates the amount of space that is required to store an arbitrarily $\hat{\boldsymbol{\theta}}$ plus the corresponding residuals. Then, the best model, is defined as the minimizer of this code length expression.

4 Derivation of Code Length Expression

This section derives an code length expression for encoding the data $\boldsymbol{x}$ and $\boldsymbol{y}$. In general we use $L(z)$ to denote the code length of the object z. Since our goal is to estimate (or model) $\boldsymbol{y}$ but not $\boldsymbol{x}$, $L(\boldsymbol{x})$ can be treated as a constant, and hence is ignored. Thus our target is $L(\boldsymbol{y})$.

We follow the two–part code approach of Rissanen (1989, Section 3.1) and express $L(\boldsymbol{y})$ as:

$$L(\boldsymbol{y}) = L(\text{fitted model}) + L(\text{data given the fitted model}) = L(\hat{\boldsymbol{\theta}}) + L(\boldsymbol{y}|\hat{\boldsymbol{\theta}}),$$

and for the present problem $L(\hat{\boldsymbol{\theta}})$ can be further decomposed into:

$$L(\hat{\boldsymbol{\theta}}) = L(\hat{r}) + L(\hat{m}) + L(\hat{\boldsymbol{k}}|\hat{m}) + L(\hat{\boldsymbol{b}}, \hat{\boldsymbol{\beta}}|\hat{r}, \hat{m}, \hat{\boldsymbol{k}}). \tag{2}$$

Code Length for $\hat{r}$ and $\hat{m}$ — $L(\hat{r}) + L(\hat{m})$: Apparently when $r \geq 3$ it is hard for human eyes to detect any subtle differences between two well–fitted but with different r regression splines (e.g., Hastie & Tibshirani 1990, Section 2.9). For this reason many researchers use $r = 3$, and in this article we shall impose 3 as an upper bound for r. Therefore the number of possible choices of r is four, and hence $L(\hat{r}) = \log_2 4 = 2$ bits: a constant that will be ignored. Since $\hat{m}$ is an integer, using the results of Rissanen (1989, Section 2.2.4), $L(\hat{m})$ is given by $L^*(\hat{m})$, with $L^*(\cdot)$

defined by: $L^*(N) = \log_2 c + \log_2 N + \log_2 \log_2 N + \ldots$, where the sum only includes positive terms and c is a constant approximately equal to 2.865. When $\hat{m}$ is reasonably large, $L^*(\hat{m})$ can be well approximated by $\log_2 \hat{m}$. Thus we have

$$L(\hat{r}) + L(\hat{m}) \approx \log_2 \hat{m}. \tag{3}$$

Code Length for $\hat{\boldsymbol{k}}$ given $\hat{m}$ — $L(\hat{\boldsymbol{k}}|\hat{m})$: Since $\{\hat{k}_1, \ldots, \hat{k}_{\hat{m}}\}$ is restricted to be a subset of $\{x_1, \ldots, x_n\}$, $\hat{\boldsymbol{k}}$ can be specified by the indices of those x_i's where a knot is placed. Such a set of (sorted) indices can be compactly specified by their successive differences. For convenience, define $\hat{k}_0 = \min(x_i)$ and $\hat{k}_{\hat{m}+1} = \max(x_i)$, and let $\hat{l}_j$ be the number of x_i's which satisfy $\hat{k}_{j-1} \leq x_i < \hat{k}_j$, $j = 1, \ldots, \hat{m}$. That is, $\hat{l}_j$ is the jth successive "index difference", and complete knowledge of $\hat{l}_1, \ldots, \hat{l}_{\hat{m}}$ implies complete knowledge of $\hat{\boldsymbol{k}}$. Now as the $\hat{l}_j$'s are integers, we have

$$L(\hat{\boldsymbol{k}}|\hat{m}) = L(\hat{l}_1, \ldots, \hat{l}_{\hat{m}}|\hat{m}) = \sum_{j=1}^{\hat{m}} L^*(\hat{l}_j) \approx \sum_{j=1}^{\hat{m}} \log_2 \hat{l}_j. \tag{4}$$

Code Length for $\{\hat{\boldsymbol{b}}, \hat{\boldsymbol{\beta}}\}$ given $\{\hat{r}, \hat{m}, \hat{\boldsymbol{k}}\}$ — $L(\hat{\boldsymbol{b}}, \hat{\boldsymbol{\beta}}|\hat{r}, \hat{m}, \hat{\boldsymbol{k}})$: Given $\{\hat{r}, \hat{m}, \hat{\boldsymbol{k}}\}$, $\{\hat{\boldsymbol{b}}, \hat{\boldsymbol{\beta}}\}$ can be readily computed by (1), and the resulting computed values are the (conditional) maximum likelihood estimates of $\boldsymbol{b}$ and $\boldsymbol{\beta}$. Rissanen (1989, pp 55-56) demonstrates that, if a (conditional) maximum likelihood estimate is estimated from N data points, then it can be effectively encoded with $\frac{1}{2}\log_2 N$ bits. It is obvious to see that, when $\hat{r} \geq 1$, each of the $\hat{b}_j$'s and $\hat{\beta}_j$'s is estimated from all the n measurements. Thus $L(\hat{b}_0) = \ldots = L(\hat{b}_{\hat{r}}) = L(\hat{\beta}_1) = \ldots = L(\hat{\beta}_{\hat{m}}) = \frac{1}{2}\log_2 n$, and hence

$$L(\hat{\boldsymbol{b}}, \hat{\boldsymbol{\beta}}|\hat{r}, \hat{m}, \hat{\boldsymbol{k}}) = \frac{\hat{m} + \hat{r} + 1}{2} \log_2 n, \qquad \text{when } \hat{r} \geq 1. \tag{5}$$

When $\hat{r} = 0$, expression for $L(\hat{\boldsymbol{b}}, \hat{\boldsymbol{\beta}}|\hat{r}, \hat{m}, \hat{\boldsymbol{k}})$ is different. It is because when $\hat{r} = 0$, $\hat{f}$ is a piecewise constant function with jumps at $\hat{k}_1, \ldots, \hat{k}_m$, and the "height" of the jth $(1 \leq j \leq \hat{m}+1)$ segment is estimated by the mean of those x_i's which lie inside this segment. There are $\hat{l}_j$ such x_i's. In other words, $\hat{f}$ is specified by $\hat{m}+1$ "height" parameters, where the jth "height" parameter is estimated from $\hat{l}_j$ data points. Hence, using the same result regarding the encoding of (conditional) maximum likelihood estimates as before, we have

$$L(\hat{\boldsymbol{b}}, \hat{\boldsymbol{\beta}}|\hat{r}, \hat{m}, \hat{\boldsymbol{k}}) = \sum_{j=1}^{\hat{m}+1} L(\text{"}j\text{th estimated height"}) = \frac{1}{2} \sum_{j=1}^{\hat{m}+1} \log_2 \hat{l}_j, \ \text{ when } \hat{r} = 0. \tag{6}$$

Code Length for $\boldsymbol{y}$ given $\hat{\boldsymbol{\theta}} = \{\hat{r}, \hat{m}, \hat{\boldsymbol{k}}, \hat{\boldsymbol{b}}, \hat{\boldsymbol{\beta}}\}$ — $L(\boldsymbol{y}|\hat{\boldsymbol{\theta}})$: This last part of the overall code length is given by the negative of the log of the likelihood of $\boldsymbol{y}$ conditioning on the fitted model $\hat{\boldsymbol{\theta}}$;

see Rissanen (1989, pp 54-55). For the present problem, it simplifies to

$$L(\boldsymbol{y}|\hat{\boldsymbol{\theta}}) = \frac{n}{2}\log_2\left\{\frac{\mathrm{RSS}(\hat{\boldsymbol{\theta}})}{n}\right\} + C, \tag{7}$$

where C is a negligible term and $\mathrm{RSS}(\hat{\boldsymbol{\theta}}) = \sum\{y_i - \hat{f}(x_i)\}^2$ is the residual sum of squares.

Final Code Length — $L(\boldsymbol{y})$: Combining expressions (2) to (7) and changing $\log_2$ to the natural log, we obtain the following MDL criterion

$$\mathrm{MDL}(\hat{\boldsymbol{\theta}}) = \begin{cases} \log\hat{m} + \sum_{j=1}^{\hat{m}}\log\hat{l}_j + \frac{1}{2}\sum_{j=1}^{\hat{m}+1}\log\hat{l}_j + \frac{n}{2}\log\left\{\frac{\mathrm{RSS}(\hat{\boldsymbol{\theta}})}{n}\right\} & \text{if } \hat{r}=0, \\ \log\hat{m} + \sum_{j=1}^{\hat{m}}\log\hat{l}_j + \frac{\hat{m}+\hat{r}+1}{2}\log n + \frac{n}{2}\log\left\{\frac{\mathrm{RSS}(\hat{\boldsymbol{\theta}})}{n}\right\} & \text{if } \hat{r}\geq 1. \end{cases} \tag{8}$$

The above criterion, being an approximation of $L(\boldsymbol{y})$, is the objective function that we aim to minimize. We propose to estimate $\boldsymbol{\theta}$ (and hence f) by the minimizer of $\mathrm{MDL}(\hat{\boldsymbol{\theta}})$.

Observe that the negative of $\mathrm{MDL}(\hat{\boldsymbol{\theta}})$ can be treated as a penalized likelihood function with three penalty terms. The first one accounts for the number of knots, the second term accounts for the "distances" between knots while the third term mainly accounts for the number of parameters to be estimated.

5 Knot Deletion Algorithm

Due to the complexity of $\hat{\boldsymbol{\theta}}$, finding the global minimizer of $\mathrm{MDL}(\hat{\boldsymbol{\theta}})$ is difficult, and a global search is infeasible even if n is only of moderate size. Common approaches to overcoming similar problems include knot insertion, knot deletion, or combinations of both. This section describes a knot deletion algorithm which may miss the global minimizer of $\mathrm{MDL}(\hat{\boldsymbol{\theta}})$ but guarantees to find a local minimizer.

Firstly fix a value for $\hat{r}$, and the knot deletion algorithm starts with placing a relatively large number of initial knots and computes the corresponding value of $\mathrm{MDL}(\hat{\boldsymbol{\theta}})$. That is, the knot deletion algorithm starts with an "overfitted" model. Then, at each time step, it removes one knot and recomputes the value of $\mathrm{MDL}(\hat{\boldsymbol{\theta}})$. This knot is chosen in such a way that, when it is removed, it provides the largest reduction in the current value of $\mathrm{MDL}(\hat{\boldsymbol{\theta}})$. Such a knot deletion strategy is often called the "greedy" strategy (e.g., see Hastie 1989). The knot deletion algorithm continues until all initial knots are removed.

One typical strategy for placing initial knots is to place a knot at every s sorted values of the x_i's. As mentioned in Smith & Kohn (1996), this initial knot placement strategy permits the initial knots to follow the density of the x_i's. In our implementation s is taken as between three to five. However, one referee pointed out that this strategy often fails in capturing high–frequency signals, such as the left tail of Doppler signal tested in Section 6.3.

If there are M initial knots, then, when the algorithm finishes, $M+1$ hierarchically fitted models are produced. The one that has the smallest $\mathrm{MDL}(\hat{\boldsymbol{\theta}})$ value will be chosen as the best fitted model for that fixed value of $\hat{r}$.

To choose a $\hat{r}$ and hence a final model, apply the above knot deletion algorithm with different candidate values of $\hat{r}$. Then the fitted model which gives the grand minimum $\mathrm{MDL}(\hat{\boldsymbol{\theta}})$ value will be chosen as the final model. In our implementation, candidate values for $\hat{r}$ are 0, 1, 2, and 3.

The computing time required to complete the whole procedure depends on the number of data points and initial knots. If $n = 100$ with 25 initial knots, the procedure on average takes less than 1 second user time to finish on a Ultra–10 machine. However when $n = 1024$ with 146 initial knots, the procedure on average takes about 50 minutes to finish using the same machine.

6 Simulation Results

This section reports results of three simulation studies, which were designed for assessing the practical performance of various aspects of the proposed procedure.

6.1 Selection of Regression Spline Order

Here we are interested in the performance of the proposed procedure in choosing the order r of the regression spline being fitted. Four test functions were used, and the design density for $\boldsymbol{x}$ was uniform in the interval $[0,1]$. These four test functions were constructed in such a way that an obvious "best" r exists; see Test Functions 1 to 4 in Table 1.

We have tested each of the four test functions with three signal–to–noise ratios (SNRs, defined below) and three different sample sizes n; i.e., each test function was tested with nine different combinations of SNR and n. We define SNR as the variance ratio $\mathrm{var}(f)/\sigma^2$ and used: high SNR $= 8$, medium SNR $= 4$ and low SNR $= 2$. The three values of n used were 50, 100 and 200, and the number of repeated simulations for each combination of test function, SNR and n was 100. However, we only report results corresponding to $n = 100$, as results for $n = 50$ and $n = 200$ are similar in nature.

Table 2 reports the number of times that the proposed procedure chose a particular value of r.

Test Function	Specification	"best" r
1	$4 - 7I_{x>0.4} + 5I_{x>0.7}$	0
2	$2.5x - 4(x-0.25)_+ + 1.8(x-0.5)_+$	1
3	$1-(x-0.3)^2 + 5(x-0.6)^2_+$	2
4	$(x-0.1)(x-0.3)(0.7x-0.5) - 3.5(x-0.65)^3_+$	3
5	$2x-1$	1
6	$\sin(10\pi x)$	—
7	$\phi(x, 0.15, 0.05^2)/4 + \phi(x, 0.6, 0.2^2)/4$	—

Table 1: *Specification of test functions. Range of x is $[0,1]$, I_E is the indicator function for the event E and $\phi(x,\mu,\sigma^2)$ is the Gaussian density with mean μ and variance σ^2 evaluated at x.*

One can see that, in general, the proposed procedure performed satisfactorily in terms of choosing the order r (with the exception of Test Function 3, low SNR).

To visually evaluate the estimation quality of the proposed procedure, we did the following. For each test function, we ranked the 100 estimates $\hat{f}$ of f associated with medium SNR, according to their values of

$$\mathrm{MSE}(\hat{f}) = \frac{1}{400}\sum_{i=0}^{399}\left\{f\left(\frac{i}{399}\right) - \hat{f}\left(\frac{i}{399}\right)\right\}^2.$$

We then plotted the 51st best estimates for each of the test functions in Fig. 1, together with the corresponding observations and true functions. From a visual sense, the proposed procedure gave reasonable results.

6.2 Comparison with Bayesian Approach

In this section we compare our procedure with the Bayesian regression spline smoothing procedure proposed by Smith & Kohn (1996). This Bayesian procedure is shown be the best amongst all regression spline smoothing procedures considered and compared by Wand (1999).

The test functions used were the three test functions used by Smith & Kohn (1996), and they are listed in Table 1 as Test Functions 5 to 7. We used the same three SNRs and the same three n's (50, 100 and 200) as in the last subsection. The x_i's were from Unif[0, 1], and the number of repeated simulations for each test function, SNR and n combination was again 100. For each estimate $\hat{f}$, we computed the corresponding $\mathrm{MSE}(\hat{f})$ value. Boxplots of the log of these computed values are given

Test Function	SNR	$r=0$	$r=1$	$r=2$	$r=3$
1 (0)	low	83	16	1	0
	medium	70	30	0	0
	high	59	41	0	0
2 (1)	low	8	71	14	7
	medium	6	71	11	12
	high	1	86	4	9
3 (2)	low	2	0	14	84
	medium	0	5	45	50
	high	0	2	62	36
4 (3)	low	2	3	14	81
	medium	0	2	10	88
	high	0	0	12	88

Table 2: *Number of times that a particular value of r is selected. The bracketed number after each test function is the corresponding "best" r.*

in Fig. 2. For those results associated with medium SNR and $n = 100$, we also ranked the quality of the MDL–based $\hat{f}$'s using their MSE($\hat{f}$) values. The 51st best MDL–based estimates for each of the test functions are plotted in Fig. 3, together with the corresponding observations, Bayesian estimates and true functions.

Judging from the boxplots displayed in Fig. 2, the proposed MDL–based procedure seems to be slightly inferior to the Bayesian procedure *in terms of mean squared error* (the MDL–based procedure performed better for Test Function 5 while the Bayesian procedure performed better for Test Functions 6 and 7). This is not surprising because of (at least) two reasons. Firstly, Hall & Hannan (1988) demonstrated that, in the probability density estimation context, the "bandwidth" chosen by the MDL principle is *not* asymptotically optimal for minimizing L_2 distance between the true and the estimated densities, but is of the same order for minimizing L_∞ distance (however the "constant term" is not optimal). That is, the MDL principle tends to oversmooth in the L_2 sense. In fact, Peter Hall (in a personal communication) speculated that this oversmoothing behaviour of the MDL principle would carry over to the regression context.

The second reason is that, the MDL–based procedure has to pay an additional price for its

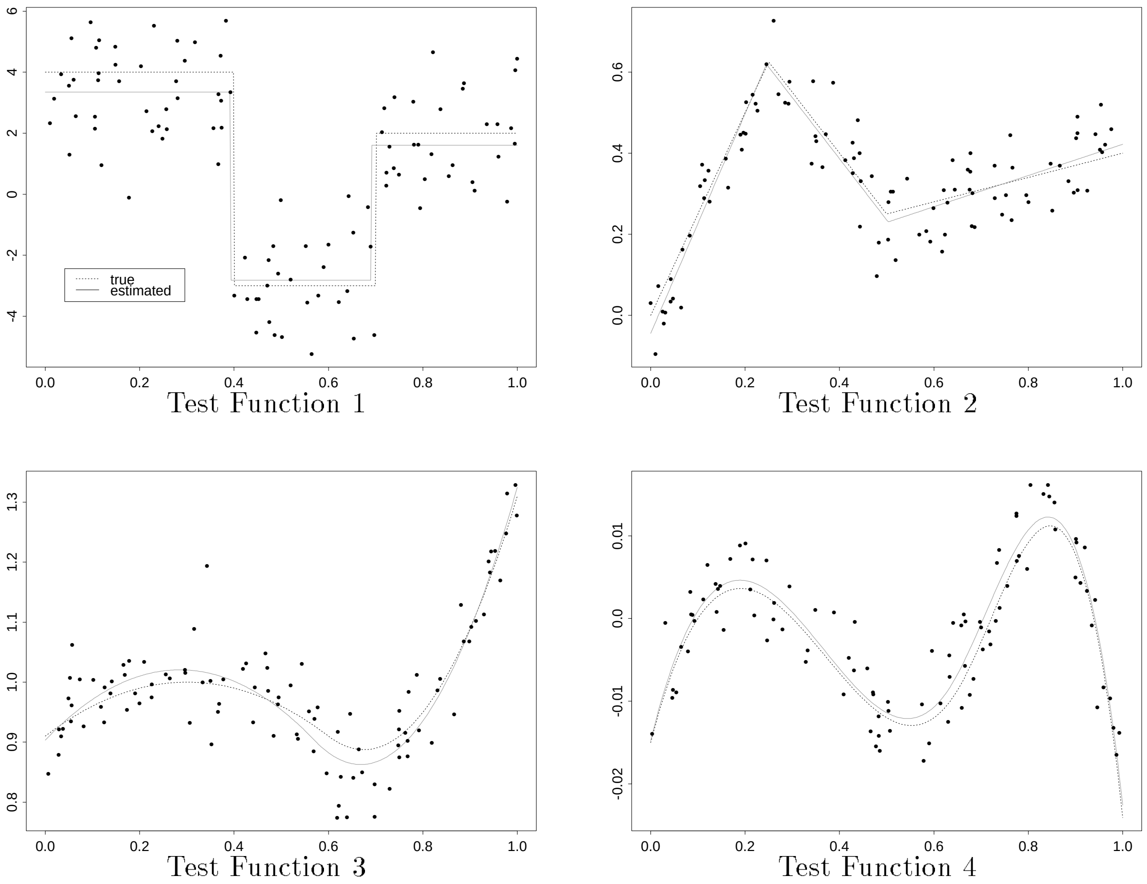

Figure 1: *Plots of true functions, estimated functions and simulated observations corresponding to the 51st smallest MSE($\hat{f}$) values for Test Functions 1 to 4, $n = 100$ and medium SNR.*

additional flexibility of the free choice of $\hat{r}$. For Test Functions 6 and 7, a good choice of $\hat{r}$ is either 2 or 3. However, due to sample randomness, the MDL–based procedure sometimes suggested $\hat{r} = 0$ or 1, which in turn produced high MSE($\hat{f}$) values. Nevertheless, we do not see the MDL–based procedure is "inadmissible" when comparing with the Bayesian procedure, and in fact we believe one can always construct test functions which are tuned to the MDL–based procedure (e.g., a piecewise linear function with suitable "hinges").

6.3 Highly Spatial Inhomogeneous Examples

The proposed procedure is also applied to the four wavelet test functions advocated by Donoho & Johnstone (1995) with the following settings: $n = 1024$, the x_i's are regularly spaced in $[0, 1]$ and SNR ≈ 49 (note that SNR is defined differently in their paper). Since n is large, we only performed 25 repeated simulations for each test functions. We indicated in the last subsection that

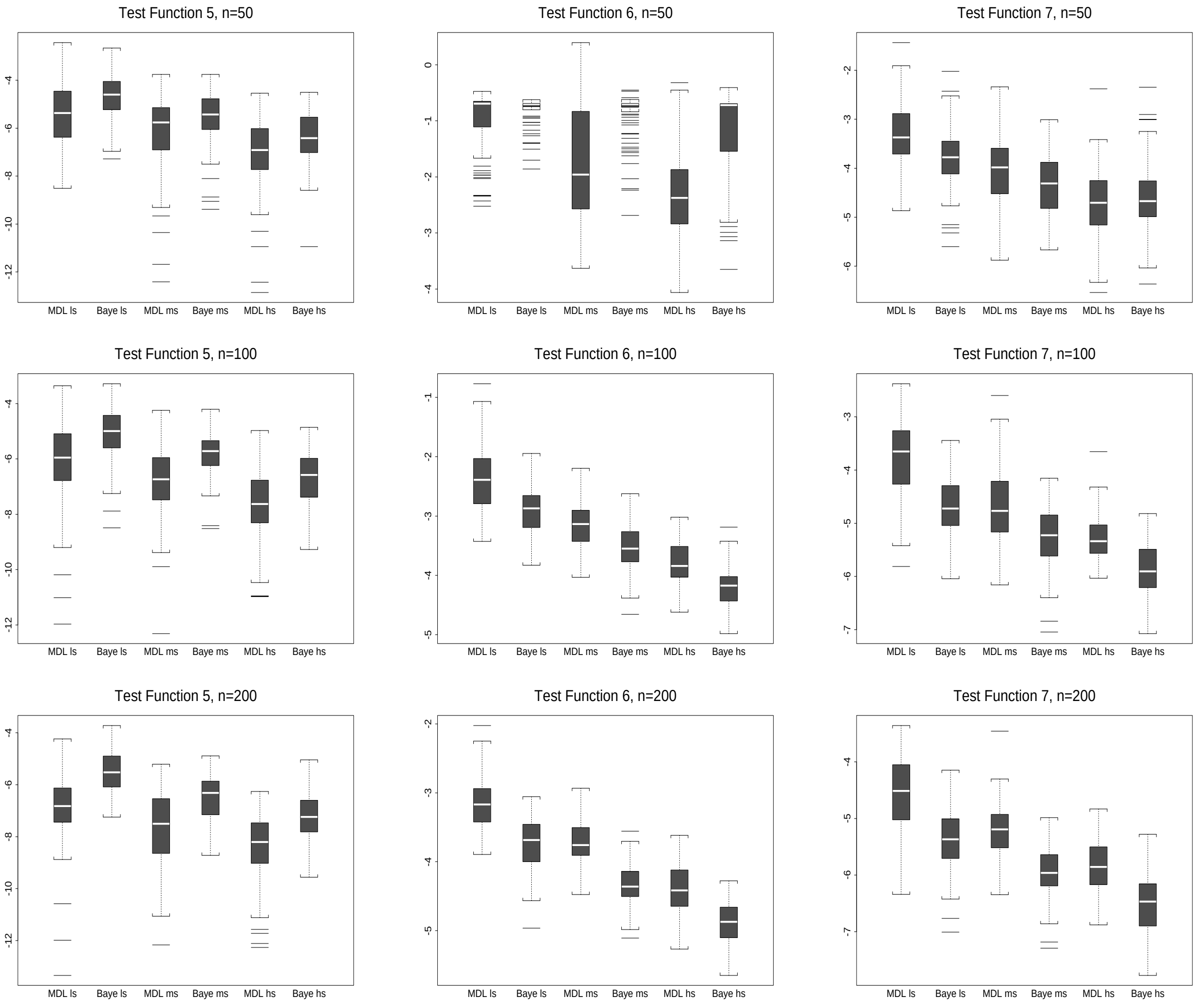

Figure 2: *Boxplots of* $\log MSE(\hat{f})$ *for Test Functions 5 to 7. Abbreviations used in boxplots: ls — low SNR, ms — medium SNR, and hs — high SNR.*

the MDL–based procedure has a tendency to oversmooth in the L_2 sense, therefore in addition to $\mathrm{MSE}(\hat{f})$ we also computed the *mean absolute deviation* for each estimated curve:

$$\mathrm{MAD}(\hat{f}) = \frac{1}{n}\sum_{i=1}^{n}\left|f(x_i) - \hat{f}(x_i)\right|.$$

Boxplots of various $\log \mathrm{MSE}(\hat{f})$ and $\log \mathrm{MAD}(\hat{f})$ values are given in Fig. 4.

As before, we ranked the MDL–based estimates by $\mathrm{MSE}(\hat{f})$, and plotted those thirteenth best estimates for the test functions in Figs. 5 and 6. Also plotted in Figs. 5 and 6 are the SureShrink estimates of Donoho & Johnstone (1995) obtained from the same noisy observations. From Figs. 4 to 6, it is hard to conclude which procedure is superior, as the two procedures performed differently with different test functions and error criteria. (We have also computed $\mathrm{MAD}(\hat{f})$ for the curve

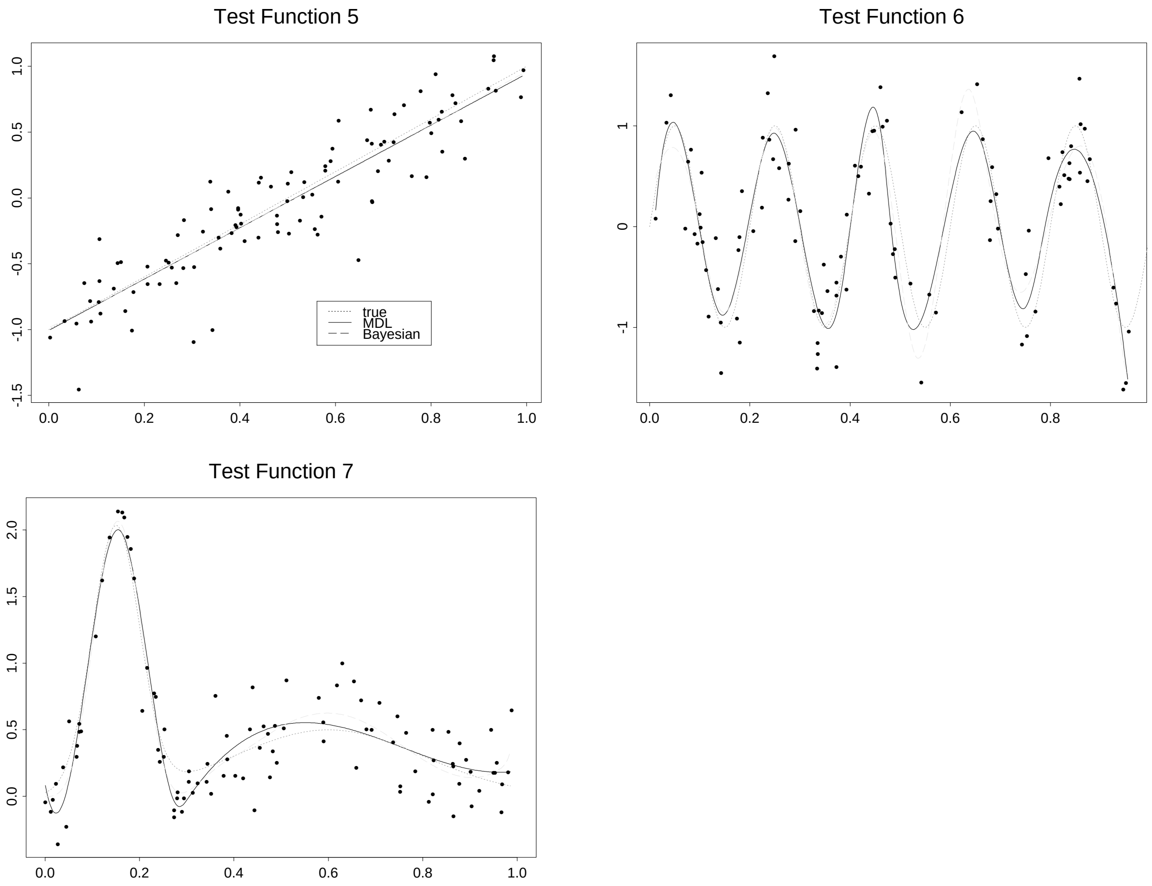

Figure 3: *Function plots corresponding to the 51st smallest MSE($\hat{f}$) values for Test Functions 5 to 7, $n = 100$ and medium SNR.*

estimates obtained in the previous subsection, but they are very similar to MSE($\hat{f}$) and so are not reported.)

Acknowledgement

Part of the work of this article was done while the author was a Ph.D. student at Macquarie University and CSIRO Mathematical and Information Sciences, Sydney, Australia. He would like to thank Victor Solo and Peter Hall for useful discussions, and a referee for helpful comments. Revision of this article was completed while the author was visiting the University of Chicago, Department of Statistics.

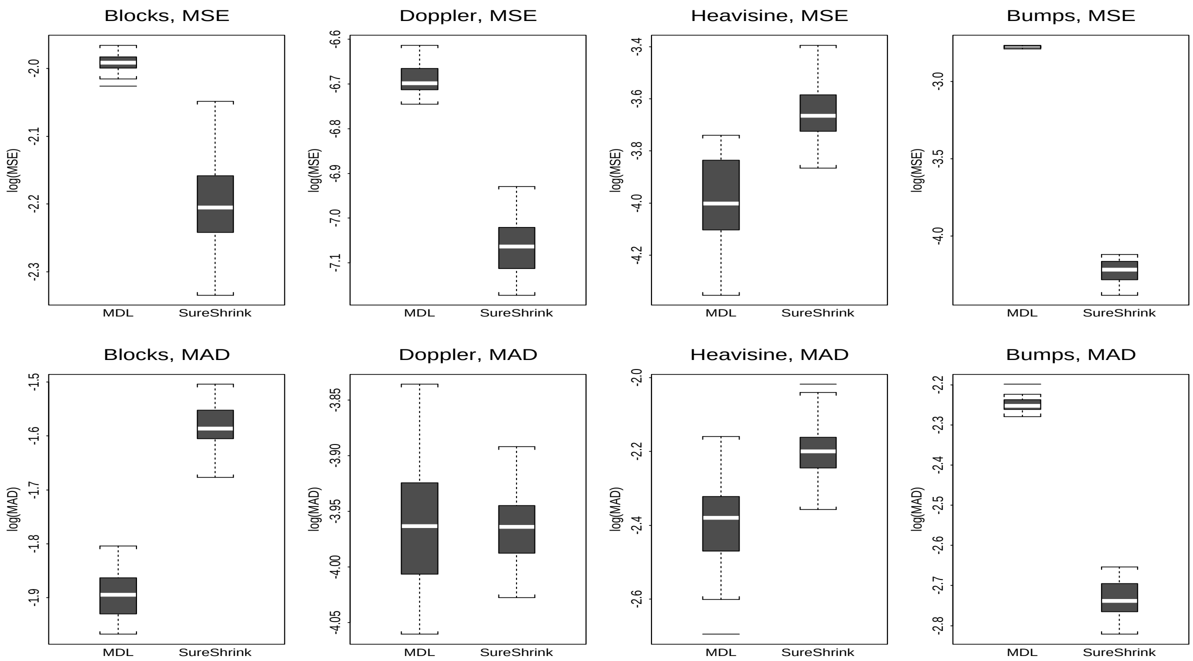

Figure 4: *Boxplots of* $\log MSE(\hat{f})$ *and* $\log MAD(\hat{f})$ *for the four wavelet testing functions.*

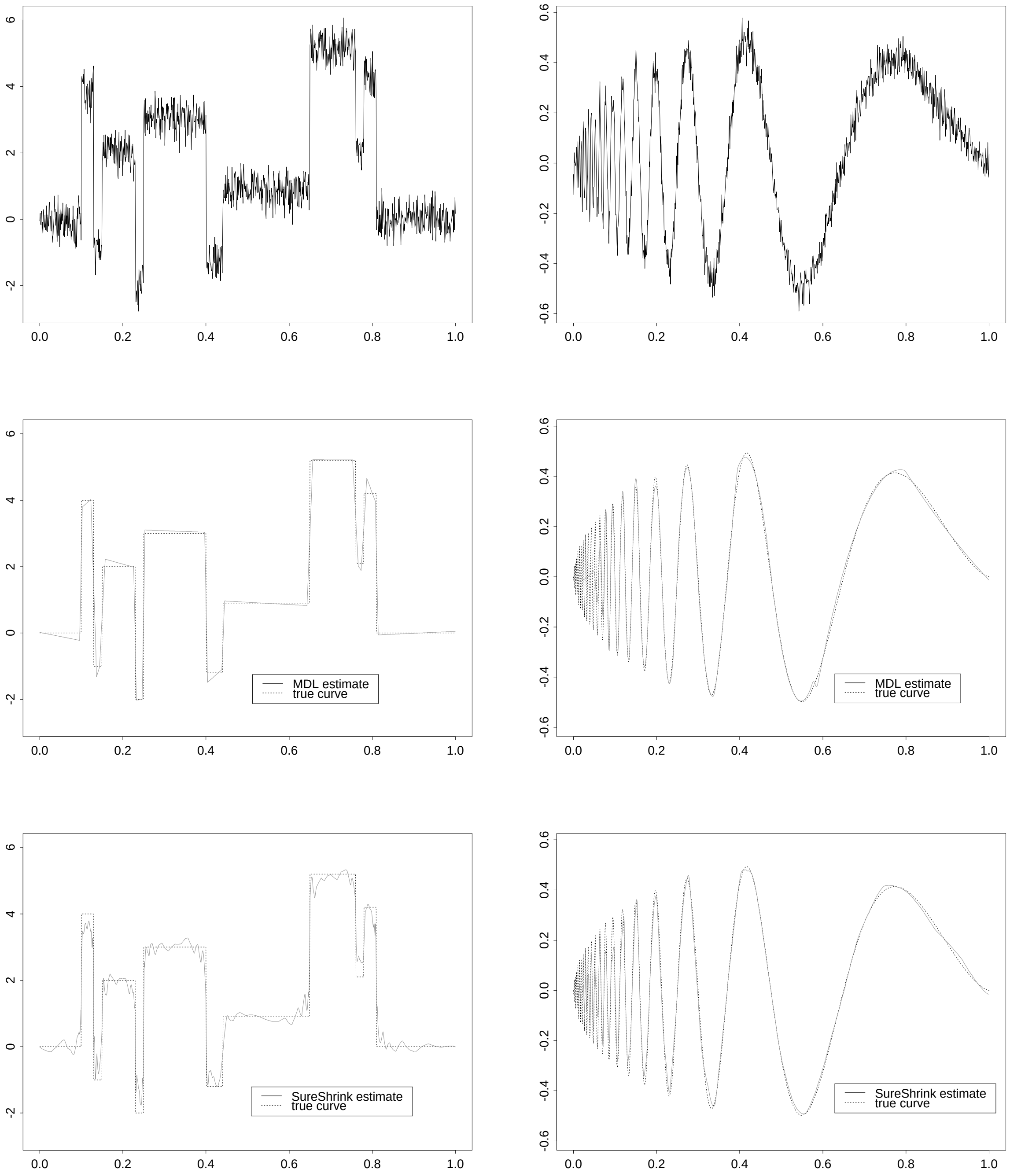

Figure 5: *Wavelet examples. Top row: simulated observations; middle row: MDL–based estimates; bottom row: SureShrink estimates. Left column: Blocks; right column: Doppler.*

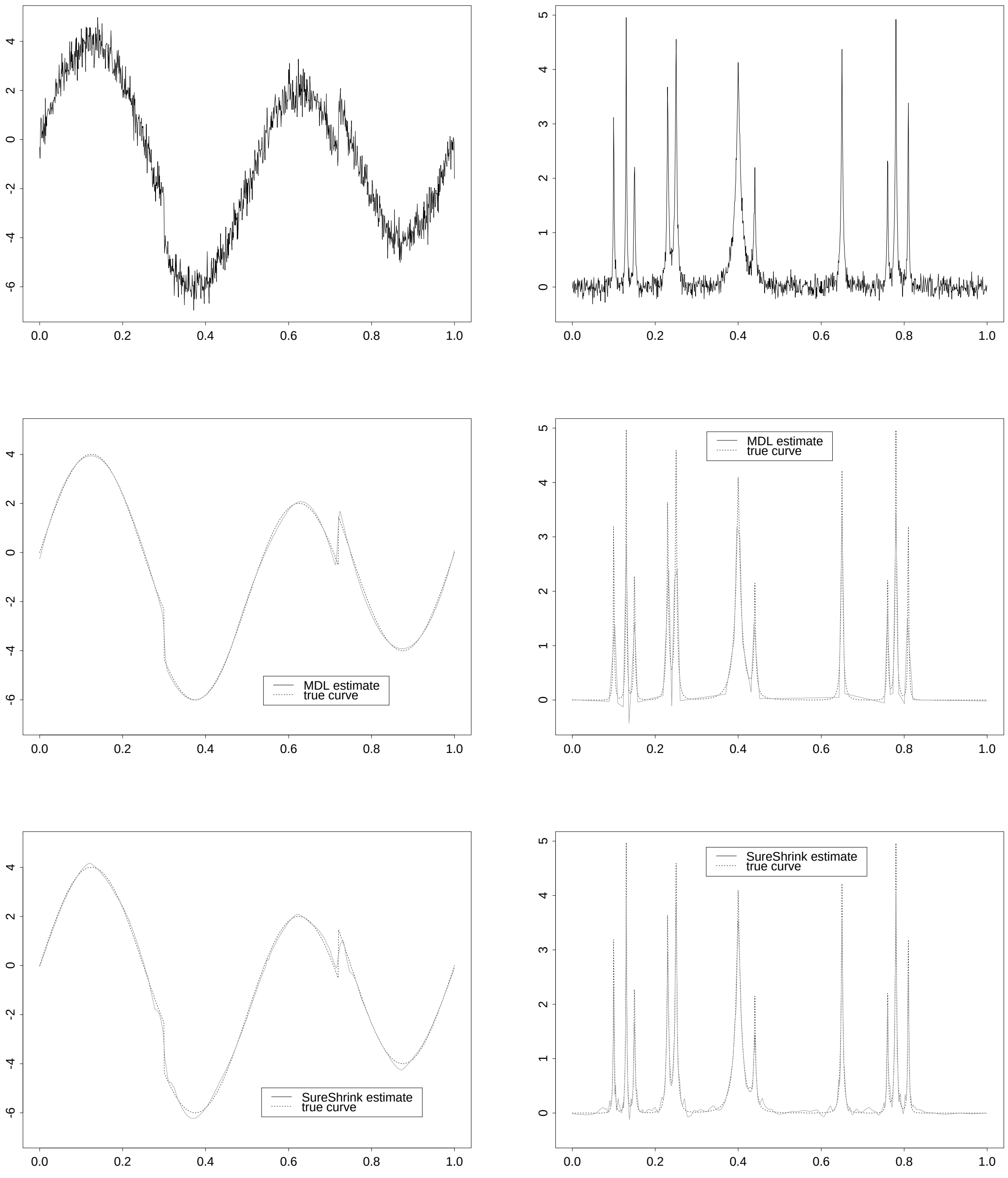

Figure 6: *Wavelet examples. Top row: simulated observations; middle row: MDL–based estimates; bottom row: SureShrink estimates. Left column: HeaviSine; right column: Bumps.*

References

Denison, D. G. T., Mallick, B. K. & Smith, A. F. M. (1998), 'Automatic Bayesian curve fitting', *Journal of the Royal Statistical Society Series B* **60**, 333–350.

Donoho, D. L. & Johnstone, I. M. (1995), 'Adapting to unknown smoothness via wavelet shrinkage', *Journal of the American Statistical Association* **90**, 1200–1224.

Friedman, J. H. & Silverman, B. W. (1989), 'Flexible parsimonious smoothing and additive modeling (with discussion)', *Technometrics* **31**, 3–21.

Hall, P. & Hannan, E. J. (1988), 'On stochastic complexity and nonparametric density estimation', *Biometrika* **75**, 705–714.

Hastie, T. (1989), 'Discussion of "Flexible parsimonious smoothing and additive modeling" by Friedman and Silverman', *Technometrics* **31**, 23–29.

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

Luo, Z. & Wahba, G. (1997), 'Hybrid adaptive splines', *Journal of the American Statistical Association* **92**, 107–116.

Rissanen, J. (1989), *Stochastic Complexity in Statistical Inquiry*, World Scientific, Singapore.

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

Wand, M. P. (1999), 'A comparison of regression spline smoothing procedures', *Computational Statistics*. To appear.