Davies, Paul Lyndon; Gather, Ursula; Weinert, Henrike

Working Paper

Nonparametric regression as an example of model choice

Technical Report / Universität Dortmund, SFB 475 Komplexitätsreduktion in Multivariaten Datenstrukturen, No. 2006,24

Provided in cooperation with:
Technische Universität Dortmund

Nonparametric regression as an example of model choice

March 7, 2006

P. L. Davies[1], U. Gather[2] and H. Weinert[2]

Abstract

Nonparametric regression can be considered as a problem of model choice. In this paper we present the results of a simulation study in which several nonparametric regression techniques including wavelets and kernel methods are compared with respect to their behaviour on different test beds. We also include the taut-string method whose aim is not to minimize the distance of an estimator to some "true" generating function f but to provide a simple adequate approximation to the data. Test beds are situations where a "true" generating f exists and in this situation it is possible to compare the estimates of f with f itself. The measures of performance we use are the L^2 and the L^∞ norms and the ability to identify peaks.

1 Introduction

Consider paired data $\mathcal{Y}_n = \{(t_i, y(t_i))\}_{i=1}^n$ where the design points are ordered $0 \le t_1 < \cdots < t_n \le 1$ but not necessarily equidistant. The problem is to use the data to derive a function f_n which can be regarded as an adequate denoised representation of the data. The model we assume for the data is

$$Y(t_i) = f(t_i) + \sigma\epsilon(t_i), \qquad i = 1, \ldots, n, \tag{1}$$

which represents a *signal* f corrupted by *noise* ϵ which we take to be standard Gaussian white noise. Given the data $\mathcal{Y}_n$ and the model (1) the problem of specifying a function f_n based on the data $\mathcal{Y}_n$ becomes a problem of model choice (where σ is treated as a nuisance

[1] Department of Mathematics, University Duisburg-Essen; Department of Mathematics, Technical University Eindhoven.

[2] Department of Statistics, University of Dortmund.

parameter). Traditional model selection operates within the model by assuming that the data $\mathcal{Y}_n$ were generated by the model (1). In the context of nonparametric regression the problem of model choice becomes: estimate f by a function $f_n^* \in \mathcal{F}$ that minimizes an expected distance or risk:

$$\mathrm{E}\left[d\left(f, f_n^*\right)\right] = \inf_{f_n \in \mathcal{F}} \mathrm{E}\left[d\left(f, f_n\right)\right], \tag{2}$$

where $\mathcal{F}$ is some specified class of functions and $d(\cdot, \cdot)$ is an appropriate loss function. In addition some model selection rules require the optimization in (2) to be conducted under constraints, whereby some measure of the complexity of a model is included in the term to be minimized. In chapter 2 we shall briefly review some well known methods for signal approximation including wavelet regression (Donoho and Johnstone, 1994), kernel estimation methods (Herrmann, 1997; Polzehl and Spokoiny, 2003, 2000) and minimum description length (MDL)-denoising (Rissanen, 2000).

Another approach to nonparametric regression is based on the concept of *data approximation* described in Davies (1995, 2003). Although the concept makes use of properties of the model it does not operate solely within it but poses the question as to whether the model can be regarded as an adequate approximation to the data. Risk minimization such as (2) is not involved nor does it make assumptions about the existence of a true underlying function f. The model with parameters (f_n, σ_n) is regarded as an adequate approximation if typical data generated under the model *look like* the observed data $\mathcal{Y}_n$. Within the set of parameter values (f_n, σ_n) which give an adequate approximation we then select an f_n which minimizes one or more measures of complexity. The "taut-string" nonparametric regression method of Davies and Kovac (2001) and the corresponding nonparametric density procedure (Davies and Kovac, 2004a) are examples of this idea. In both cases the measure of complexity is the number of peaks. In the regression problem the definition of approximation is based on the residuals (see below) whilst in nonparametric density estimation Kuiper metrics of high order are used, see Davies and Kovac (2004a).

2 Methods

2.1 Wavelet methods (WH and WS)

Wavelets are defined by

$$\Psi_{j,k}(t) = 2^{\frac{j}{2}} \Psi(2^j t - k), \quad j, k \geq 0.$$

where Ψ is the so-called mother wavelet which is often chosen to have compact support. For a suitable Ψ the Ψ_{jk} form an orthonormal basis of $L^2(\mathbb{R})$ so that every function $f \in L^2(\mathbb{R})$ can be expressed as

$$f(t) = \sum_{j,k} \bar{w}_{j,k} \Psi_{j,k}(t), \qquad \bar{w}_{j,k} = \int_{-\infty}^{\infty} f(t) \Psi_{jk}(t) dt,$$

with wavelet coefficients $\bar{w}_{j,k}$ (Daubechies, 1992).
If the design points $t_i = i/n$ are equidistant and $n = 2^{J+1}$ is a power of 2 then wavelet representations can be used to construct a signal estimate f_n as follows (see Donoho and Johnstone (1994)). First the finite wavelet transform matrix $\mathcal{W}$ is used to produce a vector w of empirical wavelet coefficients via $w = \mathcal{W}\mathbf{y}_n$ with $\mathbf{y}_n = (y(1), \ldots, y(n))^\top \in \mathbb{R}^n$ (see Nason and Silverman (1994), Donoho and Johnstone (1994) and Daubechies (1992)). The $n = 2^{J+1}$ elements of w can be labelled $w_{j,k}, j = 0, \ldots, J; k = 1, \ldots, 2^j - 1$ and $w_{-1,0}$ to express a wavelet approximation

$$f_{n,\Delta}(t_i) = \sum_{(j,k)\in\Delta} w_{j,k}\Psi_{j,k}(t_i), \quad i = 1, \ldots, n, \tag{3}$$

where $\Psi_{-1,0}(t_i) \equiv 1$ and Δ is a subset of pairs (j,k) The optimal subset Δ^* can be taken as the one which minimizes the empirical L^2 risk

$$\mathrm{E}\left[d_2(f, f_{n,\Delta^*})\right] = \inf_{\Delta} \mathrm{E}\left[d_2(f, f_{n,\Delta})\right]$$

where $d_2(f, f_{n,\Delta}) = \sum_{i=1}^n |f(t_i) - f_{n,\Delta}(t_i)|^2/n$. Minimal risk considerations indicate that Δ^* can be estimated by including only those wavelets whose coefficients $w_{j,k}$ in (3) exceed a certain threshold. Donoho and Johnstone (1994) suggest a *hard* thresholding rule where the subset Δ in (3) corresponds to those coefficients satisfying

$$|w_{j,k}| > \hat{\sigma} \cdot \sqrt{2\log(n)}, \tag{4}$$

using the median absolute deviation $\hat{\sigma}$ of the wavelet coefficients. The resulting signal construction is the Wavelet Hard Thresholding Estimator (WH). Donoho and Johnstone (1994) also propose a wavelet signal approximation based on shrinking the wavelet coefficients to zero using so-called *soft* thresholding. The VisuShrink (WS) method gives a signal construction $f_{n,VS}$ at the points $\{t_i\}_{i=1}^n$ by

$$(f_{n,VS}(t_1), \ldots, f_{n,VS}(t_n))^\top = \mathcal{W}^\top \cdot \hat{\theta}_{VS} \cdot \mathcal{W}$$

with a transformed version $\hat{\theta}_{VS}$ of the wavelet coefficient vector w given by

$$\hat{\theta}_{VS} = \left\{ \begin{array}{cc} w_{j,k} & j < j_0 \\ \mathrm{sgn}(w_{j,k})(|w_{j,k}| - \hat{\sigma} \cdot \sqrt{2\log(n)})_+ & j_0 \le j \le J \end{array} \right. ,$$

for j_0 as in Donoho and Johnstone (1994) chosen to prevent extreme cases in the wavelet transform.

2.2 Minimum Description Length Denoising

The idea of minimum description length (MDL) is that a prefix coding scheme corresponds to a probability model and that probability models can therefore be compared by the lengths of the code required to encode the data. A good model describes regularities in the data which can be exploited to reduce the code length. The naive MDL principle involves finding that model, from a collection of candidate models, that provides the shortest encoding of the data (Rissanen, 1989). This naive formulation is too simple and the complexity of the model must also be taken into account which involves encoding the model in some efficient manner. There is no objective manner of doing this and MDL remains necessarily vague and arbitrary.

MDL for signal extraction by wavelets can be seen as a special case of the variable selection problem in multiple linear regression. Let $\mathcal{Z}$ be a given $n \times n$ design matrix for linear regression. Any subset $\gamma = \{h_1, \ldots, h_k\}$ of columns of $\mathcal{Z}$ defines an $n \times k$ matrix $\mathbf{Z}_\gamma$ and the corresponding linear regression model may be written as

$$\mathbf{Y}_n = \mathbf{Z}_\gamma \beta_k + \sigma\varepsilon, \quad \beta_k \in \mathbb{R}^k$$

where the $\varepsilon = (\varepsilon(t_1), \ldots, \varepsilon(t_n))^\top \in \mathbb{R}^n$ are taken to be standard Gaussian white noise. The corresponding probability model for $\mathbf{Y}_n$ has parameters $(\gamma, \beta_k, \sigma)$ and density

$$M_\gamma = \left\{ p_\gamma(\mathbf{y}_n|\beta_k, \sigma) = (2\pi\sigma^2)^{-n/2} \exp\left[-\frac{1}{2\sigma^2}(\mathbf{y}_n - \mathbf{Z}_\gamma\beta_k)^\top(\mathbf{y}_n - \mathbf{Z}_\gamma\beta_k) \right] \right\}.$$

The length of the code required for encoding $\mathbf{y}_n$ is $-\log p_\gamma(\mathbf{y}_n|\beta_k, \sigma)$ which is nothing more than the encoding of the residuals. For a given γ the shortest encoding is attained when the maximum likelihood estimates $\hat{\beta}_k \equiv \hat{\beta}_k(\mathbf{y}_n)$ and $\sigma_n \equiv \hat{\sigma}(\mathbf{y}_n)$ for β and σ are used. If the $\mathbf{Z}_\gamma\beta_k$ and σ come for free so to speak then nothing more is to be said but the complexity of the model has not been taken into account. This can be done by encoding the $\hat{\beta}_k$ and σ_n but this requires a model for the model (see Rissanen (1987)). One way of overcoming the problem of encoding the parameters is to use a universal code or probability model p^*_γ for the class of models. One such universal code is based on the Normalized Maximum Likelihood (NML) distribution

$$p^*_\gamma(\mathbf{y}_n) = p_{NML,\gamma}(\mathbf{y}_n) = \frac{p_\gamma(\mathbf{y}_n|\hat{\beta}_k(\mathbf{y}_n), \sigma_n)}{C_\gamma}, \tag{5}$$

where

$$C_\gamma = \int_{\mathbb{R}^n} p_\gamma(x_n|\hat{\beta}_k(x_n), \hat{\sigma}(x_n))dx_n$$

Even in some of the simplest cases (including the present one of a normal distribution) this requires some adjustment to the range of integration to make the latter integral finite. If it is well-defined the NML–distribution is a minimax encoding in that it minimizes the maximum regret

$$-\log p^*_\gamma(\mathbf{y}_n) - \Big(-\log p_\gamma(\mathbf{y}_n|\hat{\beta}_k(\mathbf{y}_n), \sigma_n) \Big)$$

over all samples $\mathbf{y}_n$. Given data $\mathbf{y}_n$ the MDL linear model selection chooses a probability model $p_{\hat{\gamma}}(\cdot|\hat{\beta}_k, \sigma_n)$ by minimizing the negative logarithm of (5) over γ:

$$-\log p_\gamma(\mathbf{y}_n|\hat{\beta}_k(\mathbf{y}_n), \sigma_n) + \log C_\gamma \;\equiv\; \mathit{Fit} + \mathit{Complexity} \tag{6}$$

The above criterion can be interpreted as a penalized likelihood approach to model selection with a model fit component and a model complexity penalty term (Grünwald, 2000; Rissanen, 1996). We are again left with the problem of whether or not we have to encode the optimal γ or not, and if so, how.

This may be applied to nonparametric regression by taking the design matrix $\mathcal{Z}^\top$ to be $\mathcal{W}$. Simplifications are available as in this particular situation it can be shown that the optimal subset γ of size $n-k$ corresponds either to eliminating the wavelets with the k largest or the k smallest coefficients. After some manipulation and approximations Rissanen (2000) ends up with the following procedure. Order the absolute components in the wavelet coefficient vector $|w|_{(1)} \leq \cdots \leq |w|_{(n)}$, let $S_{(k)} = \sum_{i=1}^{k} |w|_{(i)}^2$ and then choose k by minimizing

$$(n-k)\log\frac{S_{(n)}-S_{(k)}}{n-k} + k\log\frac{S_{(k)}}{k} + \log k(n-k). \tag{7}$$

The MDL principle for wavelet model selection hence becomes a form of *hard* thresholding where the subset Δ of wavelet coefficients used in (3) corresponds to the largest $\hat{k}$ coefficients of w in absolute value or those wavelet coefficients satisfying

$$|w_{j,k}| \geq \lambda = |w|_{(\hat{k})}$$

We refer to Rissanen (2000) for more details. Risssanen (2000) showed that the MDL wavelet threshold λ is asymptotically smaller than the hard threshold (4) of Donoho and Johnstone (1994).

2.3 Kernel estimators (Plug-in and AWS)

2.3.1 Local Plug-in approach

We consider the locally adaptive kernel regression estimator

$$\hat{f}(t_j; h_{t_j}) = \sum_{i=1}^{n} y(t_i) \int_{s_{i-1}}^{s_i} \frac{1}{h_{t_j}} K\left(\frac{t-u}{h_{t_j}}\right) du, \quad j = 1, \ldots, n, \tag{8}$$

with local bandwidths h_t and $s_i = (t_i + t_{i+1})/2, i = 1, \ldots, n-1, s_0 = 0, s_n = 1$. The kernel K has order $k \geq 2$ which means that the first $k-1$ moments of K vanish but the kth moment is non-zero. Expressions for the h_t can be obtained by minimizing the pointwise mean squared error

$$MSE(f_n(t; h_t)) = \mathrm{E}(f_n(t; h_t) - f(t))^2.$$

These depend on the unknown function f and in a first step an initial estimate f_n of f based on a global bandwidth $\hat{h}$ is obtained which is plugged into the expressions for the local bandwidths. These in turn are plugged into (8). The complete algorithm is as follows (see Brockmann et al. (1993) and Herrmann (1997));

1. Set $\hat{h}_0 = (k-1)/n$;
2. Iterate for $i = 1, \ldots, (k+1)(2k+1)$: $\hat{g}_i = c\hat{h}_{i-1} n^{2/\{(2k+1)(2k+3)\}}$ and
$$\hat{h}_i = \left(\frac{\sigma_n^2 C(K)}{n \hat{I}_k(f; \hat{g}_i)} \right)^{\frac{1}{2k+1}};$$
3. Set $\hat{h} = \hat{h}_{(k+1)(2k+1)}$, the global plug-in bandwidth;
4. Set the pilot bandwidths $g_i = c_i \hat{h} n^{2/\{(2k+1)(2k+3)\}}$ for $i = 1, 2$ and the local plug-in estimator in (8) to
$$\hat{h}_{t_j} = \psi(t_j; g_1) \left(\frac{\hat{S}(t_j) C(K)}{n \widetilde{r}_k 2(t_j; g_1, g_2)} \right)^{\frac{1}{2k+1}} + (1 - \psi(t_j; g_1))\hat{h}, \quad j = 1, \ldots, n;$$

for which the following must be specified or computed: a kernel constant $C(K)$, a certain variance estimator σ_n^2 for σ^2, a weight function $w(t)$, an estimator $\hat{I}_k(f; g)$ depending on a further kernel density $M^{(k)}$ and bandwidth g of an integral functional $\int w(t)\{f^{(k)}(t)\}^2 dt$ involving the kth derivative $f^{(k)}$ of the signal, a local variance estimator $\hat{S}(t)$, a local estimator $\widetilde{r}_k^2(t; g_1, g_2)$ of $\{f^{(k)}(t)\}^2$ that depends on a kernel $\tilde{K}$ and bandwidths g_1 and g_2, a weight function $\psi(t; g_1)$, and iteration constants c, c_1 and c_2.
See Herrmann (1997) for details.

2.3.2 Adaptive weights smoothing (AWS)

The second kernel estimator we consider is Adaptive Weights Smoothing (AWS) (see Polzehl and Spokoiny (2003, 2000)). Suppose a parametric family
$$\left\{ f_\theta(x) = \sum_{i=1}^{p} \theta_i \psi_i(x), \quad \theta \in \mathbb{R}^p \right\}$$
is specified with basis functions $\{\psi_i(x)\}_{i=1}^p$ (e.g. polynomials $\psi_i(x) = x^{i-1}$). The idea is to find the largest neighborhood of every design point t_i in which the underlying signal function $f(t_i)$ can be well approximated by a function f_θ.
Fix $i \in \{1, \ldots, n\}$. The estimate $f_n(t_i)$ of f at t_i is defined as a weighted mean of the observations $y(t_j)$ by nonnegative weights w_{ij}, $j = 1, \ldots, n$:
$$f_n(t_i) = \sum_{j=1}^{n} w_{ij} \cdot y(t_j) / \sum_{j=1}^{n} w_{ij}. \qquad (9)$$

The final estimator in (9) is found by iteratively computing estimators
$$f_n^{(k)}(t_i) = \sum_{j=1}^{n} w_{ij}^{(k)} \cdot y(t_j) / \sum_{j=1}^{n} w_{ij}^{(k)} \qquad (10)$$

based on updated weights $w_{ij}^{(k)}$ determined by three kernel density functions K_l, K_s and K_e on the positive half-axis with $K_l(0) = K_s(0) = K_e(0) = 1$. Initial weights are set as $w_{ij}^{(0)} = K_l(|(t_i - t_j)/h^{(0)}|^2)$ using an initial kernel bandwidth estimate $h^{(0)}$. Then, with a memory parameter $\eta \in (0, 1)$ and a scaling factor $a > 1$, weights are iteratively defined by

$$w_{ij}^{(k)} = \eta \cdot w_{ij}^{(k-1)} + (1 - \eta) \cdot K_l \left(\frac{(t_i - t_j)^2}{(a^{k-1} h^{(0)})^2} \right) K_s(s_{ij}^{(k)}) K_e(e_{ij}^{(k)}),$$

using computed penalties $s_{ij}^{(k)}$ and $e_{ij}^{(k)}$, $j = 1, \ldots, n$. Comparing estimates $f_n^{(k-1)}(t_i)$ and $f_n^{(k-1)}(t_j)$ from a previous iteration in (10), the statistical penalty $s_{ij}^{(k)}$ is evaluated as a localized maximum likelihood contrast divided by a numerical tuning parameter. The penalty $e_{ij}^{(k)}$ aims to limit the influence of an observation when t_j is deemed to be a high level point to the local fit $f_n(t_i)$; this involves another tuning parameter. For a specified maximal bandwidth h_{max}, (10) is computed over $k = 1 + \log(h_{max}/a)/\log(h^{(0)})$ iterations. See Polzehl and Spokoiny (2003) for details.

2.4 Data approximation methods (TS and TV)

The philosophy behind the taut-string method of Davies and Kovac (2001) is the following. Firstly a precise definition of adequate approximation for the model (1) is given and then, within the class of adequate approximations, a simplest approximation is required. This is approximation followed by regularization. An arbitrary function f_n is regarded as an adequate approximation for the data $\mathcal{Y}_n$ if the residuals $r_n(t_i) = y_n(t_i) - f_n(t_i)$ "look like" white noise with variance σ^2. The property of white noise which is used is that normalized sums of white noise are again Gaussian random variables with the same variance. This leads to the following condition

$$\max_{I \in \mathcal{I}} \frac{1}{\sqrt{|I|}} \left| \sum_{t_j \in I} (y(t_j) - f_n(t_j)) \right| \leq \sigma \sqrt{\tau \log n} \tag{11}$$

where $|I|$ is the number of points $t_j \in I$, $\mathcal{I}$ is a collection of subintervals of [0, 1] and $\tau > 2$ is some constant. The justification of (11) is the following. The particular form is chosen because it forces the function f_n to be close to the data whilst taking the noise level into account. It is a multiresolution scheme as it looks at the deviations over all scales of intervals, from single points to the whole interval. If $\mathcal{I}$ is the set of all intervals and the $y(t_i)$ are generated by (1) then (11) is satisfied asymptotically with $f_n = f$ for $\tau > 2$. This follows from a result on the uniform modulus of continuity of the Brownian motion on [0, 1] due to Dümbgen and Spokoiny (2001). To make (11) operational we must estimate the noise level σ and specify a constant τ. The default value we use is $\tau = 2.5$ and the estimate

$$\sigma_n = \frac{1.48}{\sqrt{2}} \text{median}\{|y(t_2) - y(t_1)|, \ldots, |y(t_n) - y(t_{n-1})|\}. \tag{12}$$

Another possibility of quantifying the noise for equally spaced design points is the following (see (Davies and Kovac, 2004b)). Using the FFT we first compute the periodogram $I_y(\omega)$ of the data

$$I_y(\omega) = \frac{1}{2\pi n}\left|\sum_{i=1}^{n} y(t_i)e^{-\omega i\sqrt{-1}}\right|^2 .$$

If the signal does not contain very high frequencies then $I_y(\omega)$ is almost constant $\frac{\sigma^2}{2\pi}$ at such frequencies and the noise parameter σ^2 can then be estimated from $I_y(\omega)$. Which ever version of σ_n is used we are lead to the following definition of approximation

$$\max_{I\in\mathcal{I}} \frac{1}{\sqrt{|I|}}\left|\sum_{t_j\in I}(y(t_j)-f_n(t_j))\right| \le \sigma_n\sqrt{\tau\log n}. \tag{13}$$

In practice there is little to be gained by taking all intervals and dyadic multiresolution schemes as for wavelets can be used which are much faster (see Davies and Kovac (2001)). There is no restriction either to equally spaced design points or to powers of two.
There are many functions f_n which are adequate approximations in the sense of (13). An extreme example is any function which interpolates the $y(t_i)$. The second step consists of a regularization in that we wish to calculate the simplest approximating function. The two definitions of simple we use are the number of local extremes and the total variation of f_n. In other words in one case we wish to minimize the number of local extremes of f_n subject to (13) whereas in the second case we wish to minimize

$$\sum_{i=2}^{n} |f(t_i) - f(t_{i-1})|. \tag{14}$$

To minimize the number of local extremes subject to (13) Davies and Kovac (2001) developed the taut string algorithm. Although this does not guarantee the exact solution it is a fast O(n) algorithm and very often does give the correct solution. This can be checked post facto. Minimizing (14) subject to (13) is a linear programming problem.

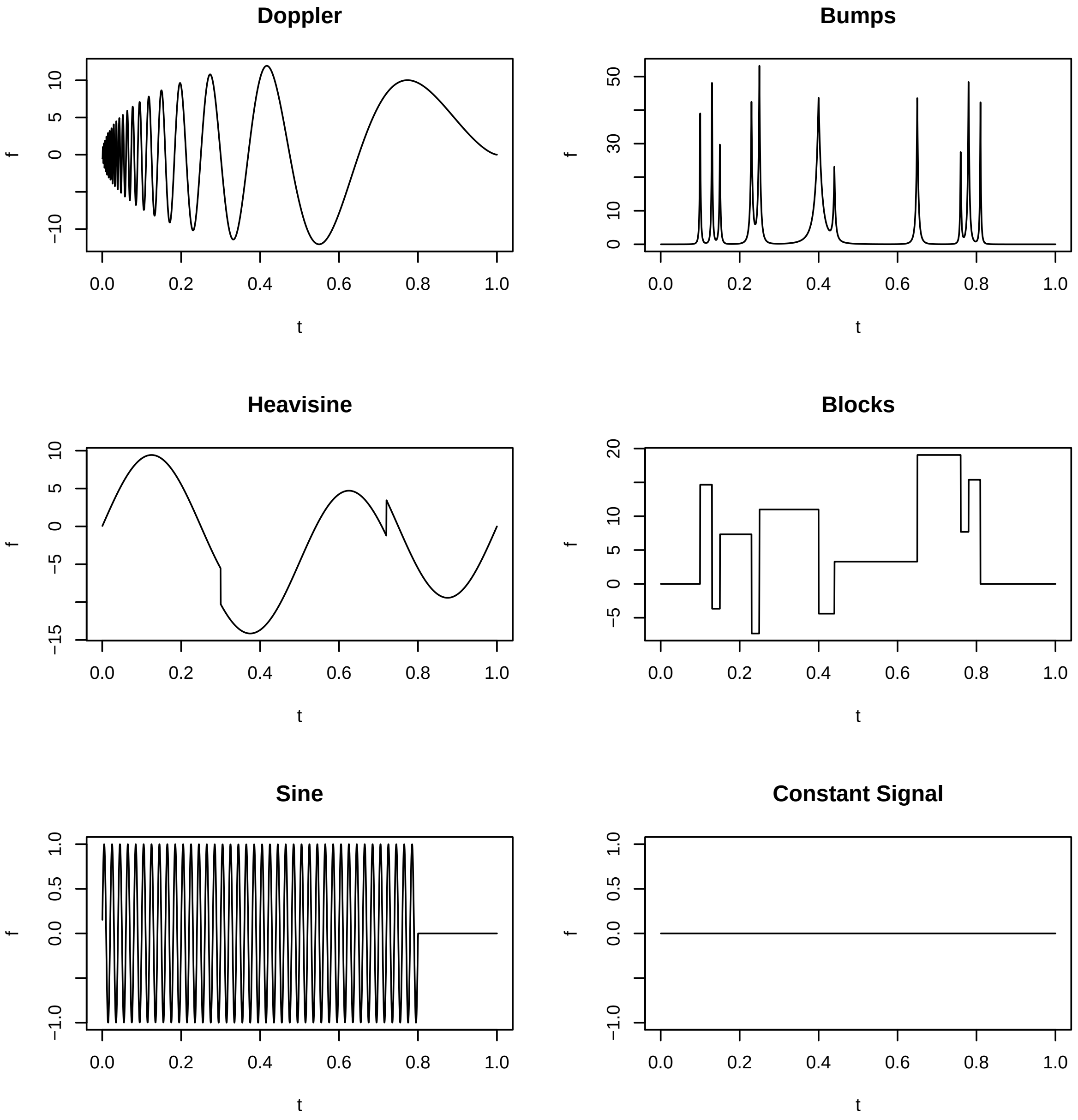
3 Test beds and loss functions

3.1 Test beds

The test beds we use are those introduced in Donoho and Johnstone (1994) and which are known as Blocks, Bumps, Heavisine and Doppler. We also include a constant signal as well as a heavily oscillating sine-function which terminates with a constant. The signals are defined below and depicted in Figure 1.

- Doppler:

$$f(t) = (t\,(1-t))^{\frac{1}{2}}\sin\left(\frac{2\pi\,(1+\delta)}{(t+\delta)}\right),$$



Doppler

Bumps

Heavisine

Blocks

Sine

Constant Signal

Figure 1: The regression functions used as test beds

with $\delta = 0.05$.

- Bumps:
$$f(t) = \sum h_j K\left(\frac{t-t_j}{w_j}\right), \quad K(t) = (1+|t|)^{-4},$$
with $(t_j) = (10, 13, 15, 23, 25, 40, 44, 65, 76, 78, 81)/100$,
$(h_j) = (40, 50, 30, 40, 50, 42, 21, 43, 31, 51, 42)/10$,
and $(w_j) = (5, 5, 6, 10, 10, 30, 10, 10, 5, 8, 5)/1000$.

- HeaviSine:
$$f(t) = 4\sin 4\pi t - \mathrm{sgn}\,(t-0.3) - \mathrm{sgn}\,(0.72-t)$$

- Blocks:
$$f(t) = \sum h_j K(t-t_j), \quad K(t) = \frac{1+\mathrm{sgn}(t)}{2}$$
with $(t_j) = (10, 13, 15, 23, 25, 40, 44, 65, 76, 78, 81)/100$,
and $(h_j) = (40, -50, 30, -40, 50, -42, 21, 43, -31, 21, -42))/10$.

- Sine:
$$f(t) = \begin{cases} \sin(100\pi t) & t \le 0.8 \\ 0 & t > 0.8 \end{cases}$$

3.2 L^p-norms

The loss functions we consider are the empirical versions of the L^2- and L^∞–norms defined by

$$d_2(f,g) = \left(\frac{1}{n}\sum_{i=1}^{n}|f(t_i) - g(t_i)|^2\right)^{1/2} \tag{15}$$

$$d_\infty(f,g) = \max_{1\le i\le n}|f(t_i) - g(t_i)|, \tag{16}$$

(see Donoho and Johnstone (1994) and Rissanen (2000)). For any given test bed with function f and for any given procedure resulting in some f_n the measures of performance are the average values of $d_2(f, f_n)$ and $d_\infty(f, f_n)$ over the simulations.

3.3 Identification of extremes

We introduce a new loss which measures how well the extremes (e.g., peaks and troughs) of an estimate f_n match those of the test signal f. There are two possible errors. The reconstruction f_n can fail to have a local extreme of the correct type at a point where a target signal f exhibits one. The second type of error is that f_n exhibits a local extreme at a point where the test bed function does not have one. Both of these types of error must allow for a certain degree of error in the position of the local extreme. With this in mind, we propose a peak identification loss (PID) as follows.

Let n_{extr} denote the number of local extremes of the signal function f at the design points and let n_{est} denote the number of local extremes of a reconstruction f_n of f. The number of local extremes of f that are correctly identified by f_n will be denoted by n_{id}. The peak identification loss is now defined by

$$\begin{aligned} d_{id}(f, f_n) &= \operatorname{sgn}(n_{est} - n_{extr})\left((n_{extr} - n_{id}) + (n_{est} - n_{id})\right) \\ &= \operatorname{sgn}(n_{est} - n_{extr})\left(n_{extr} + n_{est} - 2n_{id}\right), \end{aligned} \tag{17}$$

where the counts $(n_{extr} - n_{id})$ and $(n_{est} - n_{id})$ measure the extent of the two errors described above. We use $\operatorname{sgn}(n_{est} - n_{etr})$ so that it is possible to see if too many (positive sign) or too few (negative sign) local extremes are identified.
We still require a definition of the number n_{id} of *correctly identified* local extremes of f. We use tolerances as follows. Let $t_1^{pk} < \ldots < t_{n_{pk}}^{pk} < 1$ and $t_1^{tr} < \ldots < t_{n_{tr}}^{tr} < 1$ denote respectively the positions of the peaks and troughs of the target signal f at the design points so that $n_{pk} + n_{tr} = n_{extr}$. For identification purposes, each local extreme of f is assigned a corresponding tolerance given by

$$tol(t_i^{\ell}) = \frac{\min\left\{t_i^{\ell} - t_{i-1}^{\ell}, t_{i+1}^{\ell} - t_i^{\ell}\right\}}{2}, \quad i = 1, \ldots, n_{\ell}; \quad \ell \equiv pk, tr, \tag{18}$$

where $t_0^{\ell} = 0$, $t_{n_{\ell}+1}^{\ell} = 1$ and the label ℓ above represents peaks (pk) or troughs (tr). A peak of f at a position t_i^{pk} is said to be correctly identified by an reconstruction f_n if the position of a peak of f_n is within a distance $tol(t_i^{pk})$ from t_i^{pk}, $1, \ldots, n_{pk}$. An analogous definition holds for troughs. The count n_{id} is the number of correctly identified local extremes $\{t_i^{pk}\}_{i=1}^{n_{pk}}$ and $\{t_i^{tr}\}_{i=1}^{n_{tr}}$ by a reconstruction f_n. The tolerances specified in (18) are not severe.

4 Simulations

We conducted two simulation studies to compare the performances of the seven methods for nonparametric regression described in Section 2: the wavelet methods WH and WS, the MDL method, the kernel methods Plug-in (PL) and AWS, and the data approximation methods: taut-string (TS) and total variation (TV). The sample sizes used are $n = 2^7 = 128, 2^8 = 256, 2^{10} = 1024, 2^{11} = 2048$ and $2^{12} = 4096$. In the first study, the design points are taken equidistant: $t_i = i/n$, $i = 1, \ldots, n$. In the second study we use non-equidistant design points, so the t_i, $i = 1, \ldots, n$ are generated randomly from a standard uniform distribution. The standard deviations σ in (1) were taken to be $\sigma = 0.4, 0.8, 1.0$ and 1.4. To measure the performance of the signal approximations we compute risks based on the L^2, L^{∞} and PID losses described in Sections 3.2 and 3.3. For each method, test signal function f, σ-value, and sample size n, we generated 1000 independent samples $\mathcal{Y}_{j,f,n,\sigma}$ of size n to approximate the risk:

$$\hat{R}_n(f, f_n, d) = \frac{1}{1000}\sum_{j=1}^{1000} d(f, f_{nj}),$$

using the reconstruction f_{nj} for f based on each sample $\mathcal{Y}_{j,f,n,\sigma}$, $j = 1, \ldots, 1000$, and losses $d = d_2^2, d_\infty, d_{id}$ given in (15) and (16). For the PID given in (17) it is not meaningful to take a mean of the 1000 single values because of the sign. We use the mean of the absolute value of the PID and indicate with (+) or (-) if on average too many or too few extreme values were found. This gives an indication of how often the PID is negative or positive. If the PID is 0 for every of the 1000 simulations this is noted with (*).
To calculate the reconstructions for the non-equidistant data points we use the same methods as for equidistant. Although there exist versions for non-equidistant data for WH, WS, AWS and PL we used the standard versions because they performed better on our test beds. For the MDL method there are no versions for non-equidistant design points. The methods TS and TV can be applied without change to non-equidistant data.

4.1 Results and Summary

In Tables 1 to 6 the ranks of all methods for all studies are shown. Simulation results for the first study (equidistant design points) are listed in Tables 7 to 9 and for the second study (non-equidistant design points) in Tables 10 to 12. The values of the two best methods are depicted in **bold**. As the simulation results show no large differences in the ranking of methods over different values for σ we report only the results for $\sigma = 0.8$.
Figures 2-5 show signal reconstructions based on samples with several test bed signals. We can summarize the results of the simulation studies as follows:

- With the exception of TS and TV the performance with respect to the peak identification deteriorates as the sample size increases.
- The results for equidistant and non-equidistant design points do not differ greatly.
- The MDL method often produces too many local extremes (see Section 2.2 for an explanation).
- The reconstructions produced by kernel and wavelet methods (WH, WS, AWS and PL) often failed to reproduce the magnitudes of the peaks for the Bumps-function.
- As the Blocks function is piecewise constant all the methods apart from TS and TV performed poorly as they are designed to give smooth reconstructions.
- TS is better than TV.
- There are two types of behaviour for the sine function. Either the signal is not recognized at all or the reconstruction is reasonable. The reconstructions improved for large sample size n and smaller σ.
- The MDL method performs extremely badly on the white noise test bed. Calculations show that in this situation about 60% of the wavelet coefficients will be retained. The other wavelet methods and the kernel methods show a wave-line whilst the TS and TV often give a constant.

- Overall TS and TV perform the best. Of the 756 possible ranks in the tables TS is ranked either 1 or 2 in 724 cases. Of the remaining 32 cases it is ranked 3 seventeen times. It was ranked 5 or 6 seven times, six of which on the sine test bed.

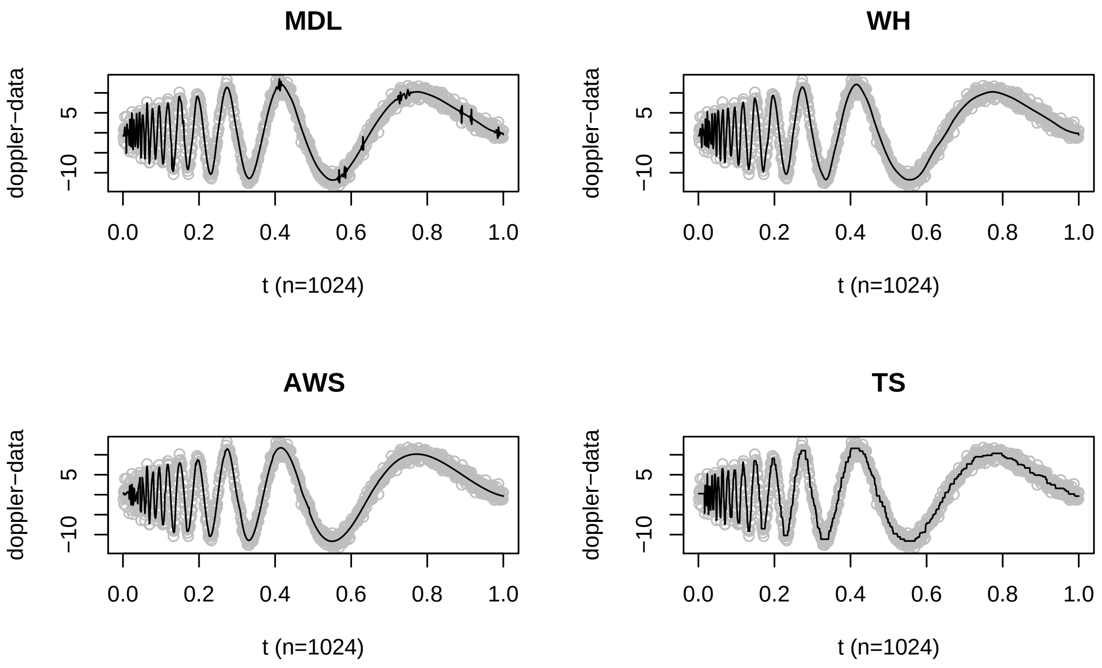


Figure 2: Reconstructions of the Doppler function with $n = 1024$ and $\sigma = 1.0$

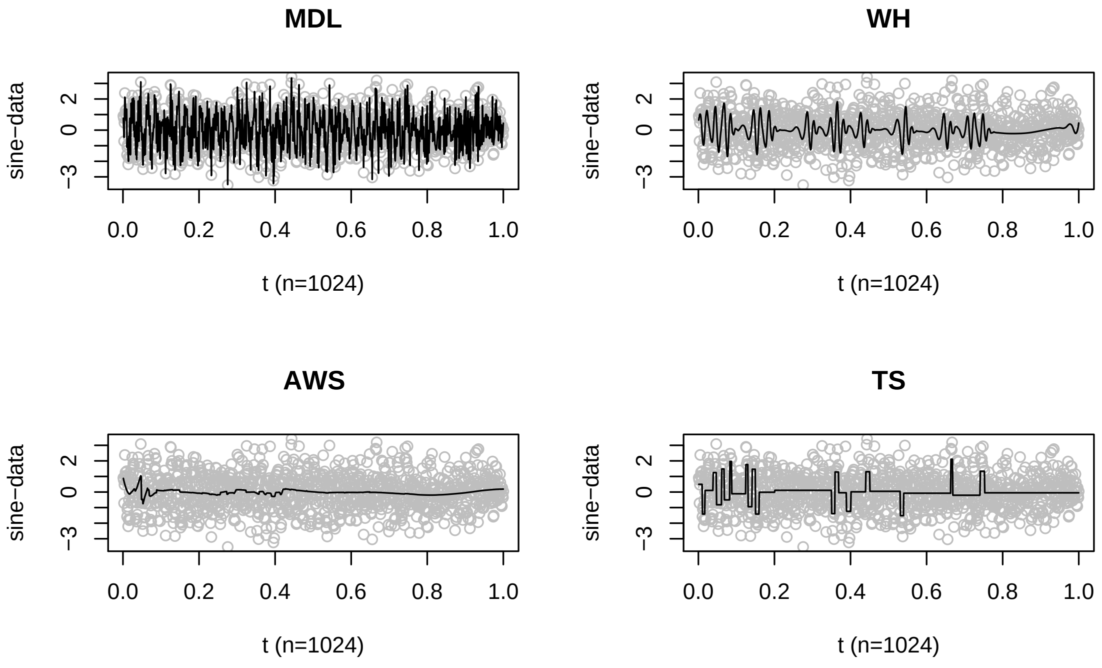


Figure 3: Reconstructions of the Sine function with $n = 1024$ and $\sigma = 1.0$

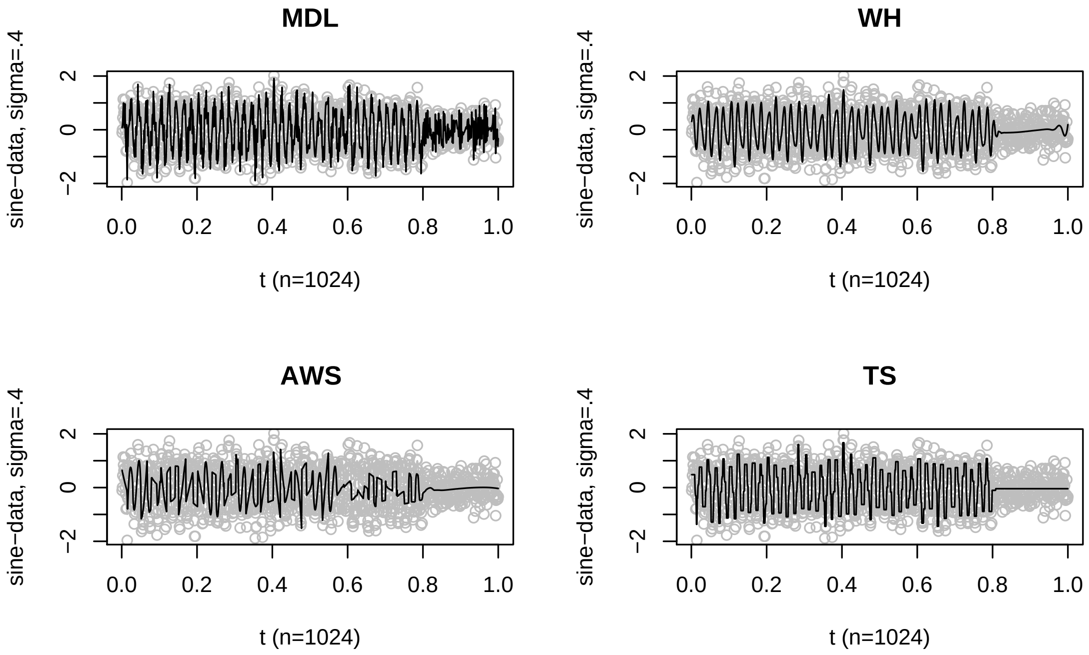


Figure 4: Reconstructions of the Sine function with $n = 1024$ and $\sigma = 0.4$

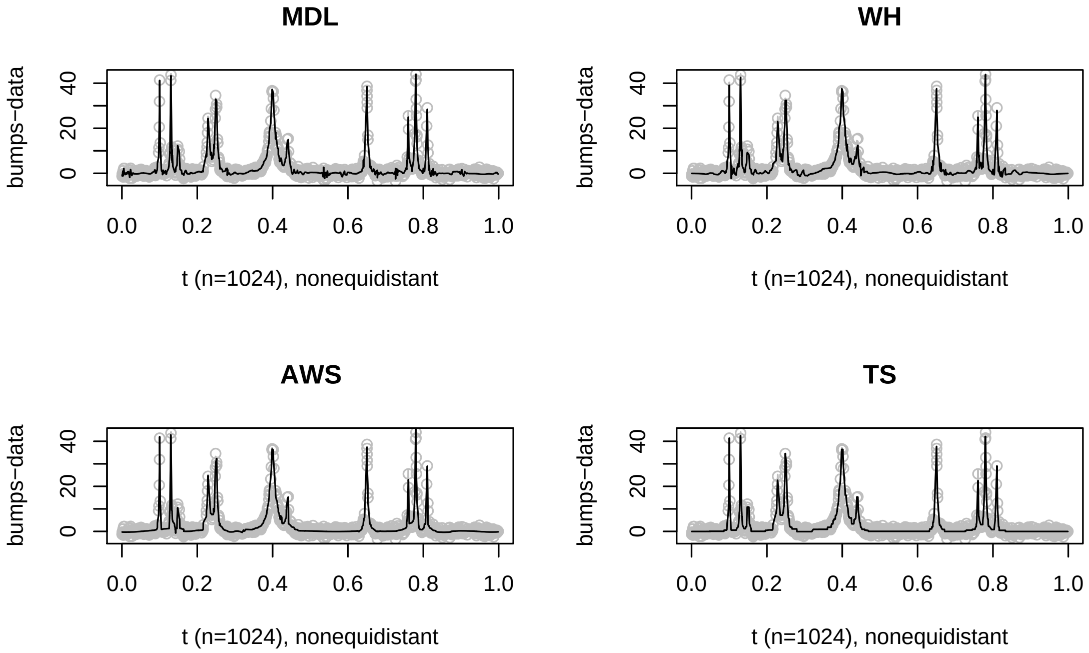


Figure 5: Reconstructions of the Bumps function with $n = 1024$ (non-equidistant) and $\sigma = 1.0$

Method	MDL	WH	WS	AWS	TS	TV	PL
Average Rank	4.48	4.44	4.73	4.10	2.63	3.33	4.29

Table 1: Average rank of the seven methods w.r.t. L_∞-norm for equidistant design points

Method	MDL	WH	WS	AWS	TS	TV	PL
Average Rank	5.15	3.90	5.58	2.78	2.42	4.93	3.24

Table 2: Average rank of the seven methods w.r.t. L_2-norm for equidistant design points

Method	MDL	WH	WS	AWS	TS	TV	PL
Average Rank	6.40	4.61	3.56	4.06	1.85	2.25	5.27

Table 3: Average rank of the seven methods w.r.t. peak identification risk (absolute value) for equidistant design points

Method	MDL	WH	WS	AWS	TS	TV	PL
Average Rank	4.25	4.98	5.23	3.59	2.78	2.72	4.47

Table 4: Average rank of the seven methods w.r.t. L_∞-norm for non-equidistant design points

Method	MDL	WH	WS	AWS	TS	TV	PL
Average Rank	4.48	4.53	5.96	2.58	2.16	4.81	3.48

Table 5: Average rank of the seven methods w.r.t. L_2-norm for non-equidistant design points

Method	MDL	WH	WS	AWS	TS	TV	PL
Average Rank	6.68	4.76	3.55	4.04	1.97	2.01	4.99

Table 6: Average rank of the seven methods w.r.t. peak identification risk (absolute value) for non-equidistant design points

Table 7: Simulation results for L_∞-norm, $\sigma = 0.8$ (equidistant design points)

function	n	MDL	WH	WS	AWS	TS	TV	LPI
Doppler	128	**2.94**	**3.67**	5.28	8.05	4.65	6.23	7.36
	1024	**3.07**	**2.83**	3.41	4.41	3.17	3.69	4.32
	4096	2.86	**1.95**	2.48	2.43	**2.22**	2.36	3.52
Bumps	128	**2.57**	7.75	17.25	7.43	**5.02**	5.08	36.58
	1024	**2.9**	4.28	11.56	6.84	**2.91**	4.89	8.60
	4096	**3.14**	3.87	9.68	3.90	**3.20**	4.72	5.44
Heavisine	128	2.32	2.46	2.41	2.40	**2.14**	2.75	**2.14**
	1024	2.48	2.67	2.45	**1.60**	**1.30**	1.62	2.40
	4096	2.51	2.71	2.46	1.63	**1.11**	**1.23**	2.42
Blocks	128	**2.05**	7.67	9.38	3.43	**1.56**	2.50	5.09
	1024	2.19	4.10	6.60	**1.92**	**1.63**	1.96	6.06
	4096	2.32	4.19	6.15	2.07	**1.65**	**2.01**	7.06
Sine	128	2.24	1.45	1.33	**1.23**	1.24	1.29	**1.19**
	1024	2.76	1.39	**1.14**	1.24	1.76	1.38	**1.01**
	4096	3.03	1.00	**0.92**	1.11	1.24	1.11	**0.60**
Constant	128	2.24	0.76	0.41	0.37	**0.31**	**0.20**	0.43
	1024	2.73	0.49	**0.14**	0.21	0.21	**0.12**	**0.14**
	4096	3.01	0.44	**0.07**	0.23	0.09	0.09	**0.07**

Table 8: Simulation results for L_2-norm, $\sigma = 0.8$ (equidistant design points)

function	n	MDL	WH	WS	AWS	TS	TV	LPI
Doppler	128	**1.03**	**0.88**	3.50	4.18	1.55	4.95	2.71
	1024	0.51	**0.17**	0.66	**0.28**	0.33	0.59	0.29
	4096	0.25	**0.05**	0.22	**0.04**	0.13	0.19	0.14
Bumps	128	**0.51**	5.56	14.83	2.94	**1.14**	3.02	27.38
	1024	**0.32**	0.52	2.12	0.58	**0.21**	0.73	0.43
	4096	0.23	0.13	0.65	**0.06**	**0.11**	0.25	0.14
Heavisine	128	0.87	0.45	0.53	**0.27**	0.43	0.80	**0.24**
	1024	0.23	0.11	0.22	**0.02**	0.10	0.15	**0.07**
	4096	0.12	**0.04**	0.10	**0.01**	**0.04**	0.06	**0.04**
Blocks	128	**0.26**	4.82	9.39	0.50	**0.16**	0.75	2.32
	1024	0.10	0.48	1.73	**0.04**	**0.03**	0.11	0.52
	4096	0.06	0.15	0.63	**0.01**	**0.01**	0.03	0.23
Sine	128	0.63	0.45	0.44	**0.42**	**0.41**	0.49	**0.42**
	1024	0.62	**0.24**	0.34	0.40	0.39	0.39	**0.10**
	4096	0.60	0.08	0.17	**0.04**	0.08	0.20	**0.03**
Constant	128	0.62	0.06	0.04	**0.03**	**0.02**	**0.03**	**0.03**
	1024	0.62	0.01	**0.00**	**0.00**	**0.00**	0.01	**0.00**
	4096	0.62	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**

Table 9: Simulation results for peak identification loss (absolute value), $\sigma = 0.8$ (equidistant design points)

function	n	n.extr	MDL	WH	WS	AWS	TS	TV	LPI
Doppler	128	20	4.22 (+)	**2.72** (-)	**2.63** (-)	7.5 (-)	3.92 (-)	3.77 (-)	10.09 (-)
	1024	39	44.98 (+)	**5.13** (-)	**4.8** (-)	16.94 (+)	9.43 (-)	10.53 (-)	59.48 (+)
	4096	40	175.12 (+)	6.00 (+)	5.53 (+)	20.25 (+)	**4.68** (-)	**4.66** (-)	85.85 (+)
Bumps	128	21	33.33 (+)	28.66 (+)	24.84 (+)	13.68 (+)	**8.73** (-)	**6.61** (-)	19.33 (-)
	1024	21	150.32 (+)	106.97 (+)	82.37 (+)	37.55 (+)	**0.11** (+)	**0.07** (+)	178.38 (+)
	4096	21	356.34 (+)	136.02 (+)	114.83 (+)	62.24 (+)	**0.07** (+)	**0.14** (+)	605.35 (+)
Heavisine	128	6	6.57 (+)	3.02 (+)	1.86 (-)	**1.13** (+)	1.59 (-)	1.68 (-)	**1.26** (+)
	1024	6	50.15 (+)	12.86 (+)	2.79 (+)	2.08 (+)	**0.00** (*)	**0.00** (+)	6.75 (+)
	4096	6	180.09 (+)	24.69 (+)	10.87 (+)	2.71 (+)	**0.01** (+)	**0.01** (+)	19.22 (+)
Blocks	128	9	50.12 (+)	30.07 (+)	15.03 (+)	11.03 (+)	**0.20** (+)	**0.05** (+)	23.39 (+)
	1024	9	223.75 (+)	158.53 (+)	108.15 (+)	27.5 (+)	**0.04** (+)	**0.17** (+)	195.42 (+)
	4096	9	546.01 (+)	294.18 (+)	217.25 (+)	59.68 (+)	**0.06** (+)	**0.33** (+)	585.82 (+)
Sine	128	79	**51.79** (+)	80.34 (-)	80.79 (-)	79.65 (-)	**78.87** (-)	79 (-)	80 (-)
	1024	80	550.52 (+)	77.41 (+)	78.37 (+)	124.83 (+)	**68.74** (-)	**7**6.25 (-)	**37.73** (+)
	4096	80	2506.86 (+)	27.37 (+)	22.39 (+)	84.82 (+)	**2.04** (-)	**1.29** (-)	80.26 (+)
Constant	128	0	82.04 (+)	9.35 (+)	8.34 (+)	2.58 (+)	**0.38** (+)	**0.31** (+)	8.59 (+)
	1024	0	664.78 (+)	9.39 (+)	8.83 (+)	12.94 (+)	**0.23** (+)	**0.35** (+)	21.21 (+)
	4096	0	2666.8 (+)	9.65 (+)	9.18 (+)	53.59 (+)	**0.11** (+)	**0.42** (+)	51.73 (+)

Table 10: Simulation results for L_∞-norm, $\sigma = 0.8$ (non-equidistant design points)

function	n	MDL	WH	WS	AWS	TS	TV	LPI
Doppler	128	**2.87**	5.40	7.53	6.04	**4.04**	5.07	6.18
	1024	**2.94**	3.98	4.88	4.23	**3.31**	3.76	4.61
	4096	2.54	2.14	2.59	**2.09**	2.16	**2.11**	3.20
Bumps	128	**2.64**	7.18	15.27	6.17	**3.11**	4.60	20.48
	1024	**2.88**	4.63	11.63	5.96	**3.04**	4.72	10.98
	4096	**3.06**	4.13	9.15	3.78	**2.76**	4.36	7.33
Heavisine	128	2.52	2.93	3.35	2.63	**2.04**	2.75	**2.36**
	1024	2.48	2.67	2.62	1.68	**1.34**	**1.67**	2.43
	4096	2.65	2.70	2.54	1.43	**1.08**	**1.13**	2.43
Blocks	128	**2.6**	6.9	9.96	2.65	**1.63**	2.77	7.48
	1024	2.74	4.26	6.68	**1.24**	**1.61**	2.00	6.29
	4096	2.92	4.12	6.18	**0.64**	**1.59**	1.92	8.04
Sine	128	2.25	1.52	1.41	**1.30**	1.31	**1.22**	1.38
	1024	2.75	1.58	**1.18**	1.37	1.82	1.39	**1.29**
	4096	3.15	1.00	0.91	**0.77**	1.06	0.96	**0.52**
Constant	128	2.26	0.79	0.42	**0.38**	0.47	**0.20**	0.42
	1024	2.74	0.52	**0.15**	0.23	0.33	**0.12**	0.14
	4096	3.02	0.43	**0.08**	0.24	0.20	0.09	**0.07**

Table 11: Simulation results for L_2-norm, $\sigma = 0.8$ (non-equidistant design points)

function	n	MDL	WH	WS	AWS	TS	TV	LPI
Doppler	128	**0.81**	2.44	5.71	2.22	**1.27**	3.05	2.21
	1024	**0.3**	0.41	1.00	**0.34**	0.37	0.54	0.36
	4096	0.09	**0.06**	0.17	**0.04**	0.09	0.11	0.09
Bumps	128	**0.68**	4.29	12.12	1.65	**0.50**	2.05	14.09
	1024	**0.32**	0.61	2.44	0.49	**0.19**	0.63	0.91
	4096	0.12	0.12	0.44	**0.07**	**0.07**	0.13	0.13
Heavisine	128	0.55	0.71	1.04	0.51	**0.39**	0.79	**0.36**
	1024	0.13	0.13	0.25	**0.05**	0.10	0.16	**0.08**
	4096	0.06	**0.02**	0.07	0.01	**0.02**	0.03	0.03
Blocks	128	0.68	4.09	9.14	**0.37**	**0.16**	0.70	3.22
	1024	0.27	0.46	1.66	**0.03**	**0.03**	0.11	0.58
	4096	0.12	0.09	0.38	**0.01**	**0.01**	0.02	0.20
Sine	128	0.63	0.42	**0.41**	**0.41**	**0.41**	0.43	**0.40**
	1024	0.61	**0.31**	0.37	0.37	0.37	0.38	**0.13**
	4096	0.59	0.06	0.13	**0.02**	0.05	0.12	**0.02**
Constant	128	0.62	0.06	0.04	**0.02**	**0.03**	**0.03**	**0.03**
	1024	0.62	0.01	**0.00**	0.01	**0.00**	0.01	**0.00**
	4096	0.62	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**

Table 12: Simulation results for peak identification loss (absolute value), $\sigma = 0.8$ (non-equidistant design points)

function	n	n.extr	MDL	WH	WS	AWS	TS	TV	LPI
Doppler	128	18.77	12.53 (+)	8.4 (+)	6.63 (-)	7.69 (-)	**6.06** (-)	**6.06** (-)	**5.48** (-)
	1024	36.96	64.17 (+)	16.61 (+)	10.7 (-)	18.37 (+)	**9.7** (-)	**9.74** (-)	62.55 (+)
	4096	39.88	204.78 (+)	13.36 (+)	7.86 (+)	23.22 (+)	**5.28** (-)	**5.24** (-)	91.57 (+)
Bumps	128	17.26	35.26 (+)	27.69 (+)	21.51 (+)	11.32 (+)	**5.55** (-)	**5.46** (-)	14.52 (+)
	1024	21	177.68 (+)	121.51 (+)	91.19 (+)	42.78 (+)	**0.24** (-)	**0.15** (-)	146.54 (+)
	4096	21	434.76 (+)	168.46 (+)	126.69 (+)	76.22 (+)	**0.08** (+)	**0.12** (+)	561.3 (+)
Heavisine	128	5.95	10.65 (+)	4.72 (+)	2.12 (-)	2.38 (+)	**1.62** (-)	**1.7** (-)	3.9 (+)
	1024	6	51.55 (+)	13.3 (+)	3.4 (+)	4.93 (+)	**0.00** (*)	**0.00** (+)	8.96 (+)
	4096	6	177.29 (+)	25.07 (+)	12.13 (+)	4.29 (+)	**0.00** (+)	**0.01** (+)	21.12 (+)
Blocks	128	8.46	49.71 (+)	32.52 (+)	16.51 (+)	11.74 (+)	**0.13** (-)	**0.07** (+)	21.95 (+)
	1024	9	224.64 (+)	162.01 (+)	105.09 (+)	28.22 (+)	**0.06** (+)	**0.20** (+)	191.81 (+)
	4096	9	545.27 (+)	291.22 (+)	214.25 (+)	60.04 (+)	**0.06** (+)	**0.31** (+)	588.27 (+)
Sine	128	55.92	**32.43** (+)	50.46 (-)	51.2 (-)	54.42 (-)	55.37 (-)	55.55 (-)	**49.29** (-)
	1024	79.99	550.78 (+)	**43.68** (-)	44.11 (-)	65.85 (+)	63.29 (-)	60.8 (-)	**33.41** (+)
	4096	80	2506.11 (+)	29.59 (+)	25.49 (+)	89.36 (+)	**3.34** (-)	**1.86** (-)	83.59 (+)
Constant	128	0	81.84 (-)	8.92 (-)	8.02 (-)	2.51 (-)	**0.37** (-)	**0.25** (-)	8.57 (-)
	1024	0	665.75 (-)	9.32 (-)	8.79 (-)	12.46 (-)	**0.22** (-)	**0.36** (-)	21.39 (-)
	4096	0	2665.38 (-)	9.38 (-)	8.94 (-)	54.07 (-)	**0.17** (-)	**0.51** (-)	51.36 (-)

Acknowledgement

This work has been supported by the Collaborative Research Center 'Reduction of Complexity in Multivariate Data Structures' (SFB 475) of the German Research Foundation (DFG).

References

Brockmann, M., T. Gasser, and E. Herrmann (1993). Locally adaptive bandwidth choice for kernelregression estimators. *Journal of the American Statistical Association 88*(424), 1302–1309.

Daubechies, I. (1992). *Ten lectures on wavelets.* Philadelphia, Pennsylvania: Society for Industrial and Applied Mathematics.

Davies, P. and A. Kovac (2001). Local extremes, runs, strings and multiresolution (with discussion and rejoinder). *Annals of Statistics 29*, 1–65.

Davies, P. L. (1995). Data features. *Statistica Neerlandica 49*(2), 185–245.

Davies, P. L. (2003). Approximating data and Statistical procedures – I. Approximating data. Technical Report 7/2003, SFB 475, Departement of Statistics, University of Dortmund, Germany.

Davies, P. L. and A. Kovac (2004a). Densities, spectral densities and modality. *Annals of Statistics 32*(3), 1093–1136.

Davies, P. L. and A. Kovac (2004b). Quantifiying the cost of simultaneous non-parametric approximation of several samples. *Preprint*.

Donoho, D. L. and I. M. Johnstone (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika 81*(3), 425–455.

Dümbgen, L. and V. Spokoiny (2001). Multiscale testing of qualitative hypotheses. *Annals of Statistics 29*(1), 124–152.

Grünwald, P. (2000). Model selection based on minimum description length. *Journal of Mathematical Psychology 44*, 133–152.

Herrmann, E. (1997). Local bandwidth choice in kernel regression estimation. *Journal of Computational and Graphical Statistics 6*(1), 35–54.

Nason, G. P. and B. W. Silverman (1994). The discrete wavelet transform in s. *Journal of Computational and Graphical Statistics 3*, 163–191.

Polzehl, J. and G. Spokoiny (2003). Varying coefficient regression modeling. Preprint.

Polzehl, J. and V. Spokoiny (2000). Adaptive weights smoothing with applications to image restoration. *Journal of the Royal Statistical Society B 62*, 335–354.

Rissanen, J. (1987). Stochastic complexity. *Journal of the Royal Statistical Society B 49*(3), 223–239.

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

Rissanen, J. (1996). Fisher information and stochastic complexity. *IEEE Transactions on Information Theory 42*, 40–47.

Rissanen, J. (2000). MDL denoising. *IEEE Transactions on Information Theory 46*(7), 2537–2760.