Ann. Inst. Statist. Math.
Vol. 56, No. 3, 449–473 (2004)
©2004 The Institute of Statistical Mathematics

A NEW ALGORITHM FOR FIXED DESIGN REGRESSION AND DENOISING

F. COMTE[1] AND Y. ROZENHOLC[2]

[1]*MAP5, UMR CNRS 8145, Université Paris 5, 45 rue des Saints-Pères, 75270 Paris cedex 06, France,* e-mail: fabienne.comte@univ-paris5.fr
[2]*LPMA, UMR CNRS 7599, Université Paris VII, 175 rue du Chevaleret, 75013 Paris, France and Université du Maine, Le Mans, France,* e-mail: rozen@math.jussieu.fr

(Received October 4, 2002; revised September 8, 2003)

Abstract. In this paper, we present a new algorithm to estimate a regression function in a fixed design regression model, by piecewise (standard and trigonometric) polynomials computed with an automatic choice of the knots of the subdivision and of the degrees of the polynomials on each sub-interval. First we give the theoretical background underlying the method: the theoretical performances of our penalized least-squares estimator are based on non-asymptotic evaluations of a mean-square type risk. Then we explain how the algorithm is built and possibly accelerated (to face the case when the number of observations is great), how the penalty term is chosen and why it contains some constants requiring an empirical calibration. Lastly, a comparison with some well-known or recent wavelet methods is made: this brings out that our algorithm behaves in a very competitive way in term of denoising and of compression.

Key words and phrases: Least-squares regression, piecewise polynomials, adaptive estimation, model selection, dynamical programmation, algorithm for denoising.

1. Introduction

We consider in this paper the problem of estimating an unknown function f from $[0,1]$ into $\mathbb{R}$ when we observe the sequence Y_i, $i = 1, \dots, n$, satisfying

$$Y_i = f(x_i) + \sigma\varepsilon_i, \tag{1.1}$$

for fixed x_i, $i = 1, \dots, n$ in $[0,1]$ with $0 \le x_1 < x_2 < \cdots < x_n \le 1$. Most of the theoretical part of the work concerns any type of design but only the equispaced design $x_i = i/n$ is computationally considered and implemented. Here ε_i, $1 \le i \le n$ is a sequence of independent and identically distributed random variables with mean 0 and variance 1. The positive constant σ is first assumed to be known. Extensions to the case where it is unknown are proposed.

We aim at estimating the function f with a data driven procedure. In fact, we want to estimate f by piecewise standard and trigonometric polynomials in a spirit analogous but more general than e.g. Denison *et al.* (1998). We also want to choose among "all possible subsets of a large collection of pre-specified candidates knot sites" as well as among various degrees on each subinterval defined by two consecutive knots.

Our method is based on recent theoretical results obtained by Baraud (2000, 2002), Baraud *et al.* (2001*a*, 2001*b*) who adapted to the regression problem general methods of model selection and adaptive estimation initiated by Barron and Cover (1991) and developed by Birgé and Massart (1998), Barron *et al.* (1999). Results on Gaussian regression can also be found in Birgé and Massart (2001). It is worth mentioning that a similar (theoretical) solution to our regression problem, in a context of regression with random design, is studied by Kohler (1999): he proposes also piecewise smooth regression functions to estimate the regression function, and he uses a penalized least-squares criterion as well. The approach is similar to Baraud's (2000) and he uses Vapnis-Chervonenkis theory in place of Talagrand's or deviation inequalities.

All the results we have in mind about fixed design regression have the specificity of giving non asymptotic risk bounds and of dealing with adaptive estimators. The first results about adaptation in the minimax sense in that context were given by Efromovich and Pinsker (1984). Some asymptotic results have been also proved by Shibata (1981), Li (1987), Polyak and Tsybakov (1990). An overview of most nonparametric techniques is also given by Hastie and Tibshirani (1990). Lastly, it is worth mentioning that most available algorithms deal with equally spaced design; results and proposals concerning the more general case of a non necessarily equi-spaced design are quite recent. Some of them can be found for instance in Antoniadis and Pham (1998), see also the survey in Antoniadis *et al.* (2002).

An attractive feature of the method which is developed here is that, once a calibration step is done, everything is completely automatic and reasonably fast. Friedman and Silverman (1989) already gave an algorithm for optimizing the number and location of the knots of the partition in an adaptive way: this algorithm is used by Denison *et al.* (1998) but the latter calibrate a piecewise cubic fit. In other words, all their polynomials have the same degree a priori fixed to be 3. Ours have variable degrees between 0 and $r_{\max}$ (which is $r_{\max} = 75$ in most experiments) and those degrees are also automatically chosen. This provides an important flexibility for denoising step signals for instance. Note that we did not deal with splines (and associated constraints) which were studied for instance by Lindstrom (1999) using a penalized estimation method as well. Moreover, the calibration operation being done once and for all, the only input of the algorithm is the maximal number of knots to be considered and the maximal degree $r_{\max}$. Contrary to many MCMC methods, we do not have any complicated nor arbitrary stopping criterion to deal with, nor do we have any problems of initialization either. A great number of wavelet methods have also been recently proposed in the literature. For an exhaustive presentation and test of these methods, the reader is referred to Antoniadis *et al.* (2002). Therefore, we compared our method with standard toolboxes implemented by Donoho and Johnstone (1994), as well as with some more recent methods tested in Antoniadis *et al.* (2002). The performances of our algorithm prove that our method is very good, for any sample size, any type of function f, and whether σ is known or not. Let us mention also that our method seems to globally behave in a very competitive way, in term of $\mathbb{L}_2$-error performances as well as in term of sparseness of the representation of the signal. In addition, we deal with much more general frameworks. Our main drawback until now is in term of the complexity of our algorithm,which is of the order $O(n^2)$ linear operations or $O(n^3)$ elementary operations $(+, \times, <)$ when the one of wavelet algorithms is of the order $O(n \log_2(n))$ elementary operations. Actually we propose a quick but approximated version with complexity of the order $O(n)$ linear operations or $O(n^2)$ elementary operations $(+, \times, <)$. As a counterpart the analysis provided by wavelet methods includes $2^{n/2}$ bases which are constructed on dyadic partitions whereas ours includes about $(2r_{\max})^n$ different bases which are constructed on general partitions.

Section 2 gives some more details on the theoretical part of the procedure. We present first its formal principle and a theoretical result is stated. Finally, the general form of the penalty we are working with is written. In Section 3, details about how the estimate is computed are given, two relevant bases are described (one for the space of standard polynomials, the other for the space of trigonometric polynomials) and the reason for the choice of the form of the penalty term involved in the computation of the estimate is explained. Section 4 presents the algorithm: the two main ideas, namely localization and dynamical programming are developed. The scheme for accelerating the algorithm without losing its good properties is introduced. Section 5 presents the empirical results for both the complete algorithm and the accelerated algorithm. The calibration procedure is led by the complete algorithm. Then both methods are compared (in term of $\mathbb{L}_2$-error and of compression performances) with wavelet denoising developed by Donoho and Johnstone (1994) and Donoho *et al.* (1995) whose toolbox is available on the Internet together with the test functions we used. Comparison results with 8 other recent methods are also provided.

2. The general method

2.1 *General framework*

We aim at estimating the function f of model (1.1) using a data driven procedure. For that purpose we consider families of linear spaces generated by piecewise (standard or trigonometric) polynomials bases and we compute for each space (basis) the associated least-squares estimator. Our procedure chooses among the resulting collection of estimators the "best" one, in a sense that will be later specified. The procedure is the following. Let $D_{\max}$ and $r_{\max}$ be two fixed integers and D an integer such that $0 \le D \le D_{\max}$. For any D, we choose a partition of $[0,1]$, that is a sequence $a_1, \dots, a_{D-1}$ of $D-1$ real numbers in $[0,1]$ such that $0 = a_0 < a_1 < \cdots < a_{D-1} < a_D = 1$, a sequence of degrees, that is integers $r_1, \dots, r_D$, such that for any d, $1 \le d \le D$, $0 \le r_d \le r_{\max}$, and a sequence $\mathcal{C}_1, \dots, \mathcal{C}_D$ of binary variables such that if $\mathcal{C}_d = \mathcal{P}$ standard polynomials are considered on $[a_{d-1}, a_d[$ and if $\mathcal{C}_d = \mathcal{T}$ trigonometric polynomials are considered on $[a_{d-1}, a_d[$, for $d = 1, \dots, D$. Then, denoting by

$$m = (D, a_1, \dots, a_{D-1}, r_1, \dots, r_D, \mathcal{C}_1, \dots, \mathcal{C}_D) \tag{2.1}$$

we define a linear space S_m as the set of functions g defined on $[0,1]$ that admit the following kind of decomposition: let $I_d = [a_{d-1}, a_d[$ for $d = 1, \dots, D-1$, and $I_D = [a_{D-1}, a_D]$, then

$$g(x) = \sum_{d=1}^{D} P_d(x) 1\!\!1_{I_d}(x), \qquad P_d \text{ polynomials with degree } r_d, d = 1, \dots, D,$$

where the polynomial P_d is standard if $\mathcal{C}_d = \mathcal{P}$ and trigonometric if $\mathcal{C}_d = \mathcal{T}$. Note that we may consider standard polynomials exclusively and in such a case $\mathcal{C}_1, \dots, \mathcal{C}_D$ is not necessary in the definition of m. We define $\ell(I)$ as the number of x_k falling in the subinterval I and we call it "weight of I".

The space S_m generated in this way has the dimension $D_m = \sum_{i=1}^{D}(r_i + 1)$. If we call $\mathcal{M}_n \subset \{1, \dots, D_{\max}\} \times \bigcup_{D=1}^{D_{\max}} [0,1]^{D-1} \times \{0, \dots, r_{\max}\}^D \times \{\mathcal{P}, \mathcal{T}\}^D$ a finite set of all possible choices for m, the family of linear spaces of interest is then $\{S_m, m \in \mathcal{M}_n\}$.

Given some m in $\mathcal{M}_n$, we define the standard least-squares estimator $\hat{f}_m$ of f in S_m by

$$\sum_{i=1}^{n}(Y_i - \hat{f}_m(x_i))^2 = \min_{g \in S_m} \sum_{i=1}^{n}(Y_i - g(x_i))^2. \tag{2.2}$$

In other words, we compute the minimizer $\hat{f}_m$ for all g in S_m of the contrast $\gamma(g)$ where

$$\gamma(g) = \frac{1}{n}\sum_{i=1}^{n}[Y_i - g(x_i)]^2. \tag{2.3}$$

Each model m being associated with an estimator $\hat{f}_m$, we have a collection of estimators $\{\hat{f}_m, m \in \mathcal{M}_n\}$ and we look for a data driven procedure $\hat{m} = \hat{m}(Y_i, i = 1, \ldots, n)$ which selects automatically among the set of estimators the one that is defined as *the* estimator of f:

$$\tilde{f} = \hat{f}_{\hat{m}}.$$

$\hat{m}$ is a vector "number of bins, partition, degree of the polynomials on each piece, type of polynomial on each piece" with values in $\mathcal{M}_n$, $(\hat{D}, (\hat{a}_1, \ldots, \hat{a}_{\hat{D}-1}), (\hat{r}_1, \ldots, \hat{r}_{\hat{D}}), (\hat{\mathcal{B}}_1, \ldots, \hat{\mathcal{B}}_{\hat{D}}))$ based solely on the data and not on any a priori assumption on f.

Let us precise what the "best" estimator is and how to select it. We measure the risk of an estimator via the Expected Average Squared Error. Namely, if $\hat{f}$ is some estimator of f, the risk of $\hat{f}$ is defined by

$$d_n^2(f, \hat{f}) = \boldsymbol{E}\left(\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - \hat{f}(x_i))^2\right).$$

The risk of $\hat{f}_m$, where $\hat{f}_m$ is an estimator built as in relation (2.2), can in fact be proved to be equal (see equation (2) in Baraud (2000)) to

$$d_n^2(f, S_m) + \frac{\dim(S_m)}{n}\sigma^2$$

where $d_n(f, S_m) = \inf_{t \in S_m} d_n(f, t)$ and $\dim(S_m)$ denotes the dimension of S_m. Therefore an ideal selection procedure choosing $\hat{m}$ should look for an optimal trade-off between $d_n^2(f, S_m)$, the so-called bias term and $\sigma^2 \dim(S_m)/n$, the so-called variance term. In other words, we look for a model selection procedure $\hat{m}$ such that the risk of the resulting estimator $\hat{f}_{\hat{m}}$ is almost as good as the risk of the best least-squares estimator in the family. More precisely, our aim is to find $\hat{m}$ such that

$$d_n^2(f, \hat{f}_{\hat{m}}) \le C \min_{m \in \mathcal{M}_n}\left\{d_n^2(f, S_m) + \sigma^2 \frac{L_m \dim(S_m)}{n}\right\}, \tag{2.4}$$

where the L_m's are some weights related to the collection of models $\{S_m, m \in \mathcal{M}_n\}$. This inequality means that, up to a constant C (which has to be not too far from one for the result to be of some interest) our procedure chooses an optimal model and inside that model an optimal estimator in the sense that it realizes a L_m-trade-off between the bias and the variance terms.

We consider the selection procedure based on a penalized criterion of the following form

$$\hat{m} = \arg \min_{m \in \mathcal{M}_n} \left[\gamma(\hat{f}_m) + \frac{\mathrm{pen}_n(m)}{n} \right]$$

where $\mathrm{pen}_n(m)$ is a penalty function mapping $\mathcal{M}_n$ into $\boldsymbol{R}^+$. We will precise this penalty later on and just mention that it is closely related to the classical C_p criterion of Mallows (1973).

The procedure is then as follows: for each model m we compute the normalized residual sum of squares, $\gamma(\hat{f}_m)$, where γ is defined by (2.3), we choose $\hat{m}$ in order to minimize among all models $m \in \mathcal{M}_n$ the penalized residual sum of squares $\gamma(\hat{f}_m) + \mathrm{pen}_n(m)/n$ and we compute the resulting estimator, $\hat{f}_{\hat{m}}$. Mallows' C_p criterion corresponds to $\mathrm{pen}_n(m) = 2\hat{\sigma}^2 \dim(S_m)/n$ where $\hat{\sigma}^2$ denotes a suitable estimator of the unknown variance of the ε_i's. Our penalty term is similar but uses an unknown universal constant instead of 2 and the factor L_m allowing for very rich collections of models (see the further discussion on the choice of the L_m's). When σ^2 is unknown we replace it by an estimator.

2.2 *Theoretical results*

From the theoretical point of view, Kohler (1999), Baraud (2000, 2002), Baraud *et al.* (2001*a*, 2001*b*) obtained several results. We formulate in detail a result corresponding to model (1.1) satisfying the following condition:

($\boldsymbol{H}_\epsilon$) The ε_i's are i.i.d. centered variables and satisfy, $\forall u \in \boldsymbol{R}$

$$\boldsymbol{E}(\exp u\varepsilon_1) \leq \exp(u^2 s^2/2)$$

for some positive s.

This assumption allows the variables ε_i's to be Gaussian with variance s^2 or to be bounded by s. The particular case of Gaussian variables is given in Baraud (2002) and the following result is a simplified version of Theorem 1 in Baraud *et al.* (2001*a*).

THEOREM 2.1. *Consider model* (1.1) *where f is an unknown function. Assume that the ε_i's satisfy Assumption* ($\boldsymbol{H}_\varepsilon$) *and that the family of piecewise polynomials described in Subsection* 2.1 *has dimensions D_m such that*

$$\sum_{m \in \mathcal{M}_n} e^{-L_m D_m} = \Sigma < +\infty \tag{2.5}$$

where the L_m's are nonnegative numbers (to be chosen) and Σ is a constant independent of n. Then there exists a universal constant $\theta > 0$ such that if the penalty function is chosen to satisfy

$$\mathrm{pen}_n(m) \geq \theta s^2 D_m (1 + L_m)$$

then the estimator $\tilde{f} = \hat{f}_{\hat{m}}$ satisfies

$$d_n^2(f, \tilde{f}) \leq C \inf_{m \in \mathcal{M}_n} \left[d_n^2(f, S_m) + \frac{\mathrm{pen}_n(m)}{n} \right] + C' s^2 \frac{\Sigma}{n} \tag{2.6}$$

where C and C' are universal constants.

This kind of result can be extended to variables ε_i's admitting only moments of order p, provided that $p > 2$ (see Baraud (2000)) for regular collections of models only (i.e. collections with one model by dimension, as in example (RP) defined below).

It is worth mentioning that (2.6) allows to compute the rate of the estimator $\tilde{f}$ as soon as f is assumed to belong to some class of regularity authorizing an evaluation of the bias term in function of the dimension of the projection space S_m.

2.3 *Collections of models and choice of the weights*

Let us now illustrate condition (2.5) in order to better see the role of the L_m's. Roughly speaking, the final rate for estimating a function of smoothness α is the minimax rate $n^{-2\alpha/(2\alpha+1)}$ when the L_m's can be chosen constant. In most other cases, the L_m's are required to be of order $\ln(n)$ and the rate falls to $(n/\ln(n))^{-2\alpha/(2\alpha+1)}$. Let us give some (standard) examples for the choice of the spaces when the design is equispaced, namely when $x_i = i/n$:

(RP) Regular piecewise polynomials (and regular S_m's). This is typically what is meant when talking about *regular* collections of models. We work with constant degrees $r_1 = \cdots = r_D = r - 1$ and we choose $a_j = j/D$ for $j = 0, \ldots, D$ (regular partition of $[0, 1]$). Then $m = (D, a_1, \ldots, a_{D-1}, r-1, \ldots, r-1, \mathcal{P}, \ldots, \mathcal{P})$, $\dim(S_m) = rD$, we take $D = 1, \ldots, D_{\max}$ and we simply impose that $rD_{\max} \leq n$, i.e. $D_{\max} = [n/r]$. Then we look for L_m's such that

$$\sum_{m \in \mathcal{M}_n} e^{-L_m D_m} = \sum_{D=1}^{[n/r]} e^{-L_m D} \leq \Sigma < \infty.$$

Therefore $L_m = 1$ (or $L_m = 2\ln(D)/D$) suits.

(IPC) Irregular piecewise polynomials with constant degrees. This illustrates by comparison the extension from regular to *general* collections of models.

We keep all the degrees constant equal to $r - 1$. We choose the $D - 1$ values of $a_1 < \cdots < a_{D-1}$ in the set $\{j/n, j \in \{1, \ldots, n-1\}\}$ for $D = 1, \ldots, D_{\max} = [n/r]$. We then have for $L_m = L_n$

$$\sum_{m \in \mathcal{M}_n} e^{-L_m D_m} = \sum_{D=1}^{[n/r]} \binom{n-1}{D-1} e^{-rDL_n}.$$

Therefore, if we choose $L_m = L_n = \ln(n)/r$ it implies that

$$\begin{aligned}\sum_{m \in \mathcal{M}_n} e^{-L_m D_m} &\leq \sum_{k=0}^{n-1} \binom{n-1}{k} e^{-r(k+1)L_n} \\ &\leq \sum_{k=0}^{n-1} \binom{n-1}{k} \left(\frac{1}{n}\right)^{k+1} = \frac{1}{n}\left(1 + \frac{1}{n}\right)^{n-1} \\ &\leq \left(1 + \frac{1}{n}\right)^n \leq e\end{aligned}$$

and condition (2.5) is satisfied with logarithmic weights.

(ITC) Irregular trigonometric polynomials with constant degrees. The partitions and the a_j's are selected just as before. The degree in this example (but not in practice) is fixed to $2r+1$ in the sense that, on an interval I of weight ℓ we consider $\mathrm{Trig}_0^\ell(x) = \sqrt{1/\ell}\mathbb{1}_I(x)$,

$$\begin{cases} \mathrm{Trig}_{2p}^\ell(x) = \sqrt{2/\ell}\cos\left(\frac{2n\pi}{\ell}px\right)\mathbb{1}_I(x), \\ \mathrm{Trig}_{2p+1}^\ell(x) = \sqrt{2/\ell}\sin\left(\frac{2n\pi}{\ell}px\right)\mathbb{1}_I(x), \end{cases}$$

for $p=1,\dots,r$. Let us mention that the Trig polynomials would have to be multiplied by $\sqrt{n}$ to be normalized in $\boldsymbol{L}^2$. For the same reason as above, this would lead to weights L_m's of order $\ln(n)$, in order for (2.5) to be fulfilled.

The degrees of the polynomials are supposed to be fixed (to r or $2r+1$) in the previous examples for the sake of simplicity but are variable in the set $\{0,\dots,r_{\max}\}$ in the algorithm developed below. In such case, the dimensional constraint $D_{\max} = [n/r]$ becomes $\sum_{d=1}^D (r_d+1) \le n$ where $r_d \in \{0,\dots,r_{\max}\}$ is the degree of the polynomial on I_d. This implies a greater number of models.

2.4 *The aim of the calibration study*

The order of the penalty as given in the theoretical results above is only a crude approximation which technically works. One of the aims of the empirical work is to find a more precise development for the choice of the penalty and to calibrate empirically the universal constants involved. For instance, if we think of a penalty:

$$\text{(2.7)} \qquad \mathrm{pen}_n(m) = s^2\left[c_1 \ln\binom{n-1}{D-1} + c_2(\ln(D))^{c_3} + c_4\sum_{d=1}^D (r_d+1) + c_5\sum_{d=1}^D [\ln(r_d+1)]^{c_6}\right]$$

for m defined by (2.1), we need to check that it satisfies (2.5). Then since we believe the constants c_i, $i=1,\dots,6$ to be universal constants, we want to compute them using simulation experiments.

Note that complementary terms in a penalty function have been studied in a theoretical framework (but for another problem and with a penalty having a different form) by Castellan (2000). On the other hand, empirical experiments for calibrating a penalty have already been performed for density estimation with regular histograms by Birgé and Rozenholc (2002). For all degrees set to zero and regular partitions, they proposed $\mathrm{pen}_n(D) = [\ln(D)]^{2.5} + D - 1$. Here, we take $c_1 = c_4 = 2$ and $s^2 = \sigma^2$. We look for c_2, c_3, c_5 and c_6.

The choice $s^2 = \sigma^2$ appears in many theoretical results. If this variance is known, we keep it as the multiplicative factor. Otherwise it can be estimated by the least-squares residuals:

$$\hat\sigma^2 = \frac{1}{n}\sum_{i=1}^n (Y_i - \hat f_m(x_i))^2$$

for a $\hat f_m$ computed on a well chosen S_m. For instance, for equispaced design regression, we can take the space generated by $a_d = d/D$ with $D = n/\ln(n)$. This has been proved to allow an extension of the theoretical results in the case of regular subdivisions in Comte and Rozenholc (2002).

3. Computation of the estimate

3.1 *The general formula*

Given a basis $\mathcal{B} = (B_0, \dots, B_r, \dots)$ of polynomials, with degree of $B_r = r$, we denote by $\mathcal{B}^r = (B_0, \dots, B_r)$ and by $\mathcal{P}_r^{\mathcal{B}}$ the linear space spanned by $B_0, \dots, B_r$.

The first step for the computation of $\tilde{f} = \hat{f}_{\hat{m}}$ is the computation of the $\hat{f}_m$'s for m varying in $\mathcal{M}_n$ among which we choose it. Let $m = (D, a_1, \dots, a_{D-1}, r_1, \dots, r_D, C_1, \dots, C_D)$ be given and recall that $I_d = [a_{d-1}, a_d[$ for $d = 1, \dots, D-1$, and $I_D = [a_{D-1}, a_D]$, $a_0 = 0$ and $a_D = 1$. Then $\hat{f}_m$ satisfies

$$\gamma(\hat{f}_m) = \frac{1}{n} \sum_{d=1}^{D} \min_{P \in \mathcal{P}_{r_d}^{\mathcal{B}}} \sum_{x_k \in I_d} (Y_k - P(x_k))^2.$$

In other words, for some given m, we replace the global minimization of the contrast γ in S_m by D minimizations of local contrasts denoted by

$$\gamma_{I_d}(g) = \frac{1}{n} \sum_{\{k/x_k \in I_d\}} (Y_k - g(x_k))^2 \tag{3.1}$$

for g varying in $\mathcal{P}_{r_d}^{\mathcal{B}}$. Then we have to compute for any degree r, any basis $\mathcal{B}$ and any interval I, the polynomial $P_r^{I,\mathcal{B}} \in \mathcal{P}_r^{\mathcal{B}}$ such that:

$$\begin{aligned} \hat{\gamma}_I^{\mathcal{B}}(r) \overset{\text{def}}{=} \gamma_I(P_r^{I,\mathcal{B}}) &= \min_{P \in \mathcal{P}_r^{\mathcal{B}}} \sum_{\{k/x_k \in I\}} (Y_k - P(x_k))^2/n \\ &= \frac{1}{n} \left[\sum_{\{k/x_k \in I\}} Y_k^2 - \sum_{\{k/x_k \in I\}} (P_r^{I,\mathcal{B}}(x_k))^2 \right]. \end{aligned}$$

This defines the contribution of the interval I to the global contrast. This local contrast is defined only by the points x_k and Y_k for the indexes k such that $x_k \in I$; therefore, any interval I containing the same x_k leads to the same minimization procedure and to the same polynomial P_I. So there is no loss of generality to consider intervals with bounds chosen among the x_k's.

It is well known that, for any basis $\mathcal{B}^r = (B_0, \dots, B_r)$ of a linear space $\mathcal{P}_r^{\mathcal{B}}$, the contrast minimizer $P_r^I = \alpha_0 B_0 + \alpha_1 B_1 + \cdots + \alpha_r B_r$ is the solution of the system of equations $C_r^I A_r^I = D_r^I$ where (denoting by X' the transpose of the vector X),

$$A_r^I = (\alpha_0, \dots, \alpha_r)', \tag{3.2}$$

$$C_r^I = (c_{s,t})_{1 \le s,t \le r}, \qquad c_{s,t} = \sum_{k/x_k \in I} B_s(x_k) B_t(x_k), \tag{3.3}$$

$$D_r^I = (d_0, d_1, \dots, d_r), \qquad d_s = \sum_{k/x_k \in I} Y_k B_s(x_k). \tag{3.4}$$

Let us denote by $\boldsymbol{X}_r^I$ the matrix $(B_s(x_k))$, $s = 0, \dots, r$, $k \in \{j/x_j \in I\}$ with $r+1$ rows and with $\#\{k/x_k \in I\} = \ell(I)$ columns, and by Y^I the vector of the Y_k's for x_k falling in I. The minimum of the contrast satisfies

$$n\hat{\gamma}_I^{\mathcal{B}}(r) = (Y^I)'Y^I - (A_r^I)'\boldsymbol{X}_r^I(\boldsymbol{X}_r^I)'A_r^I = (Y^I)'Y^I - (D_r^I)'(C_r^I)^{-1}C_r^I(C_r^I)^{-1}D_r^I,$$

and therefore

$$\hat{\gamma}_I^{\mathcal{B}}(r) = \frac{1}{n}[(Y^I)'Y^I - (D_r^I)'(C_r^I)^{-1}D_r^I]. \tag{3.5}$$

3.2 *Choice of a relevant basis*

Since (3.5) is valid for any basis $\mathcal{B}^r$ of $\mathcal{P}_r^{\mathcal{B}}$, we look for a relevant choice of the basis of polynomials $\mathcal{B}^r = (B_0, \dots, B_r)$ on the interval I. In other words, we aim at choosing the basis such that $C_r^I = I_r$ (I_r is the $r \times r$ identity matrix), that is

$$c_{s,t} = \sum_{k/x_k \in I} B_s(x_k)B_t(x_k) = \delta_{s,t}$$

where $\delta_{s,t}$ is the Kronecker symbol such that $\delta_{s,t} = 1$ is $s = t$ and $\delta_{s,t} = 0$ otherwise.

In the case of a general design $(x_i)_{1 \leq i \leq n}$ (not necessarily equispaced), for each interval I, we can build by Gram-Schmidt orthonormalization and using a Q-R decomposition of $\boldsymbol{X}$, an orthonormal basis of polynomials of any degree r, with respect to the discrete scalar product associated to the x_k's in I. The problem here is that for each possible interval I, and degree $r_{\max}$ a specific orthonormalized basis must be computed, which is feasible but costing a lot of time from a computational point of view. Consequently, some other ideas for accelerating the method have to be found. The ideas below are relevant in the equispaced case only.

3.3 *Choice of a relevant basis in the case $x_i = i/n$*

3.3.1 *Polynomial basis*

We use the discrete Chebyshev polynomials defined as follows (see Abramowitz and Stegun (1972)). The discrete Chebyshev polynomial on $\{0, 1, \dots, \ell - 1\}$ with degree r is

$$\mathrm{Cheb}_r^\ell(x) = \frac{1}{\{\sum_{i=0}^{\ell-1}[C_r^\ell(i)]^2\}^{1/2}} C_r^\ell(x) \tag{3.6}$$

where $C_0^\ell(x) = 1$ and

$$C_r^\ell(x) = \frac{1}{(r!)^2}\Delta^r\left[\prod_{s=0}^{r} g_\ell(x-s)\right], \quad \text{where} \quad g_\ell(x) = x(x-\ell)$$

and $\Delta f(x) = f(x+1) - f(x)$, $\Delta^r = \Delta^{r-1} \mathrm{o} \Delta$. Those polynomials satisfy

$$\sum_{k=0}^{\ell-1} \mathrm{Cheb}_t^\ell(k)\mathrm{Cheb}_s^\ell(k) = \delta_{s,t}, \quad \text{for} \quad 0 \leq s, \ t \leq r.$$

Therefore, choosing on the intervals $I = [i/n, \dots, (i+\ell+1)/n[$, the basis

$$B_s^I(x) = \mathrm{Cheb}_s^\ell(nx - i),$$

will do the job. This leads to

$$\hat{\gamma}_I^{\mathrm{Cheb}}(r) = \frac{1}{n}[(Y^I)'Y^I - (D_r^I)'D_r^I], \tag{3.7}$$

where D_r^I is the vector with components

$$d_s = \sum_{k/n \in I} Y_k B_s^I(k/n) = \sum_{k=0}^{\ell(I)-1} Y_{k+i}\mathrm{Cheb}_s^\ell(k). \tag{3.8}$$

3.3.2 *Trigonometric basis*

The case of piecewise trigonometric bases is even simpler since the basis described in (**ITC**) is naturally orthonormal with respect to the discrete scalar product considered with a regular design, for $\ell = \ell(I)$:

$$\sum_{x_k \in I} \mathrm{Trig}_s^\ell(x_k)\mathrm{Trig}_t^\ell(x_k) = \delta_{s,t}.$$

Therefore, choosing on the intervals $I = [i/n, \dots, (i+\ell+1)/n[$, the basis

$$B_s^I(x) = \mathrm{Trig}_s^\ell(x),$$

will do the job. This leads to

$$\hat{\gamma}_I^{\mathrm{Trig}}(r) = \frac{1}{n}[(Y^I)'Y^I - (D_r^I)'D_r^I], \tag{3.9}$$

where D_r^I is the vector with components

$$d_s = \sum_{k/n \in I} Y_k B_s^I(k/n) = \sum_{x_k \in I} Y_k \mathrm{Trig}_s^\ell(x_k). \tag{3.10}$$

3.4 *The choice of the penalty*

Equation (2.7) is our choice for the global form of the penalty. For the results given in Theorem 2.1 to hold, we must prove that $\sum_{m \in \mathcal{M}_n} e^{-L_m D_m} < +\infty$ with $\mathrm{pen}_n(m) = s^2(1+L_m)D_m$ and m defined by (2.1).

$$\begin{aligned}
\sum_{m \in \mathcal{M}_n} e^{-L_m D_m} &= \sum_{m \in \mathcal{M}_n} e^{-\,\mathrm{pen}_n(m)/s^2 + D_m} \\
&= \sum_{1 \le D \le D_{\max}, 0 \le r_d \le r_{\max}} 2^D \exp\left\{ - \left[c_1 \ln\binom{n-1}{D-1} + c_2[\ln(D)]^{c_3} \right.\right. \\
&\qquad + c_4 \sum_{d=1}^{D} (r_d + 1) \\
&\qquad \left.\left. + c_5 \sum_{d=1}^{D} [\ln(r_d+1)]^{c_6} \right] + D \right\} \\
&= \sum_{D=1}^{D_{\max}} 2^D \binom{n-1}{D-1} \exp\left\{ - \left[c_1 \ln\binom{n-1}{D-1} + c_2[\ln(D)]^{c_3} \right] + D \right\} \\
&\quad \times \left[\sum_{r_1=0}^{r_{\max}} \cdots \sum_{r_D=0}^{r_{\max}} \exp\left\{ -c_4 \sum_{d=1}^{D} (r_d+1) - c_5 \sum_{d=1}^{D} [\ln(r_d+1)]^{c_6} \right\} \right] \\
&= \sum_{D=1}^{D_{\max}} 2^D \exp\left\{ \ln\binom{n-1}{D-1} - \left[c_1 \ln\binom{n-1}{D-1} + c_2[\ln(D)]^{c_3} \right] + D \right\} \\
&\quad \times \left(\sum_{r=0}^{r_{\max}} e^{-c_4(r+1) - c_5[\ln(r_d+1)]^{c_6}} \right)^D .
\end{aligned}$$

Therefore, if $c_1 \geq 1$, $c_2 \geq 0$ and $c_5 \geq 0$, we can give the following bound

$$\sum_{m\in\mathcal{M}_n} e^{-L_m D_m} \leq \sum_{D=1}^{D_{\max}} (2e)^D e^{-c_4 D} \left(\frac{1-e^{-c_4(r_{\max}+1)}}{1-e^{-c_4}}\right)^D \leq \sum_{D=1}^{D_{\max}} \left(\frac{2e^{1-c_4}}{1-e^{-c_4}}\right)^D,$$

and this last term is bounded provided that

$$\left|\frac{2e^{1-c_4}}{1-e^{-c_4}}\right| < 1$$

that is, if $c_4 > \ln(1+2e) \simeq 1.862$. Thus in the general case, the chosen penalty is of the form:

PROPOSITION 3.1. *The following choice of the penalty:*

$$\operatorname{pen}_n(m) = s^2 \left[c_1 \ln \binom{n-1}{D-1} + c_2(\ln(D))^{c_3} + c_4 \sum_{d=1}^{D}(r_d+1) + c_5 \sum_{d=1}^{D}[\ln(r_d+1)]^{c_6}\right]$$

is such that $\sum_{m\in\mathcal{M}_n} e^{-L_m D_m} = \sum_{m\in\mathcal{M}_n} e^{-\operatorname{pen}_n(m)/s^2 + D_m}$ *converges with exponential rate, provided that* $c_1 \geq 1$, $c_2 \geq 0$, $c_4 \geq 1.87$ *and* $c_5 \geq 0$.

It should be noted that the total number of visited bases is asymptotically (for great values of n and fixed $r_{\max}$) of the order

$$\sum_{D=1}^{n} \binom{n-1}{D-1} (2r_{\max})^D = (2r_{\max})(2r_{\max}+1)^{n-1} = O((2r_{\max})^n).$$

4. Description of the algorithm

In the sequel, both for the description of the algorithm and for the empirical results, we consider only the regular design defined by $x_i = i/n$.

4.1 *Localization*

Here we should emphasize the two basic ideas of our procedure. The first one is based on a localization of the problem. With the results and notations of Section 3.1 and the subsections following, the global value of the contrast is

$$\gamma(\hat{f}_m) = \sum_{d=1}^{D} \hat{\gamma}_{I_d}^{\mathcal{B}^{r_d}}(r_d)$$

and we look for

$$\hat{m} = \arg \min_{m\in\mathcal{M}_n} \left[\hat{\gamma}_m + \frac{\operatorname{pen}_n(m)}{n}\right]$$

where we must consider here that $\operatorname{pen}_n(m) = \operatorname{pen}_{n,c}(m)$ with $c = (c_1, c_2, c_3, c_4, c_5, c_6)$ and

$$\begin{aligned}\operatorname{pen}_{n,c}(m) &= \sigma^2 \left[c_1 \ln \binom{n-1}{D-1} + c_2 \ln(D)^{c_3}\right] + \sum_{d=1}^{D} \sigma^2[c_4(1+r_d) + c_5 \ln(1+r_d)^{c_6}] \\ &\overset{\text{def}}{=} \operatorname{pen}_{n,c_1,c_2}(D) + \sum_{d=1}^{D} \operatorname{pen}_{n,c_4,c_5,c_6}(r_d).\end{aligned}$$

Then we find a localized decomposition of the penalized contrast:

$$n\gamma(\hat{f}_m) + \mathrm{pen}_{n,c}(m) = \mathrm{pen}_{n,c_1,c_2}(D) + \sum_{d=1}^{D}\{n\hat{\gamma}_{I_d}^{\mathcal{B}^{r_d}}(r_d) + \mathrm{pen}_{n,c_4,c_5,c_6}(r_d)\},$$

where the first part of the penalty $\mathrm{pen}_{n,c_1,c_2}(D)$ is the global penalization concerning the number of sub-intervals and the second part $\mathrm{pen}_{n,c_4,c_5,c_6}(r_d)$ is the local part concerning the degree on each sub-interval. We recall that $\hat{\gamma}_I^{\mathcal{B}}(r)$ is defined by (3.5) for a basis $\mathcal{B}^r$, and more precisely given by (3.7)–(3.8) or (3.7)–(3.9).

This can also be written:

$$n\gamma(\hat{f}_m) + \mathrm{pen}_{n,c}(m) = \sum_{k=1}^{n} Y_k^2 + \mathrm{pen}_{n,c_1,c_2}(D) + \sum_{d=1}^{D}[\mathrm{pen}_{n,c_4,c_5,c_6}(r_d) - p_{I_d}^{\mathcal{B}^{r_d}}(r_d)]$$

where on the interval $I = [i/n, \ldots, (j+1)/n[$

$$p_I^{\mathcal{B}^s}(s) = \sum_{t=0}^{s}\left[\sum_{k=0}^{j-i} Y_{k+i} B_t^I(k)\right]^2 \overset{\mathrm{def}}{=} p_s^{\mathcal{B}}(i,j).$$

The quantity $p_s^{\mathcal{B}}(i,j)$ represents precisely the weight of the contrast when going from i to j (j included), so that $p_s^{\mathcal{B}}(i,i)$ is defined. Note that, for $1 \le \ell \le n$, those quantities are systematically computed by setting

$$\boldsymbol{Y}_\ell = (Y_{i+k-1})_{1\le i\le \ell, 1\le k\le n-\ell} \quad \text{and} \quad \boldsymbol{B}_\ell = (B_i^I(k))_{0\le i\le r_{\max}, 0\le k\le \ell-1}$$

and by computing and storing $(\boldsymbol{B}_\ell \boldsymbol{Y}_\ell)^{\bullet 2}$ where $A^{\bullet 2} = (a_{i,k}^2)_{1\le i\le p, 1\le k\le q}$ for $A = (a_{i,k})_{1\le i\le p, 1\le k\le q}$. Then considering different values of ℓ amounts to take into account intervals of any weight $\ell = 1, \ldots, n$.

For $j \ge i$, the procedure of minimization first computes:

$$p(i,j) \overset{\mathrm{def}}{=} \min_{1\le s\le r_{\max}} \min_{\mathcal{B}^s}[\sigma^2(c_4(1+s) + c_5 \ln(1+s)^{c_6}) - p_s^{\mathcal{B}}(i,j)],$$

so that the best basis and the best degree are chosen.

4.2 *Dynamical programming*

Here we reach the point where we need to use dynamical programming (see Kanazawa (1992)). The fundamental idea of dynamical programming is that to go to point j with d steps (i.e. pieces), we must first go to some $k < j$ with $d-1$ steps and then go from k to j in one step. A very similar idea is developed from a theoretical point of view by Kohler (1999) in order to define a procedure in the same type of regression function estimation problem.

Let $q(d,k)$ be the minimum of the contrast—penalized in degree with basis selection—to go from 1 to k with d pieces; this value is thus associated to a best partition, d best bases and a choice of d best degrees which fulfill the localization constraints.

First note that $q(1,k) = p(1,k)$ which gives an initialization; then

$$q(d+1,j) = \min_{d\le k<j}[q(d,k) + p(k+1,j)] \tag{4.1}$$

which represents $2j$ operations. Then a Q matrix can be filled in, with two possible strategies:

(1) "Off line" method: Compute the $q(1,j)$ for $j = 1, \dots, n$ and then do a recursion on d using (4.1). The drawback of the method is that the actualization (i.e. if some more observations are available and so n increases), everything must be done again whereas it is clear that only changes in the last column are useful.

(2) "In line" method: Assume that you have built $(q(d,j))_{1 \le d \le j \le n}$ and you want to increase n and compute the $q(d, n+1)$, $d = 1, \dots, n+1$. Then as $q(1, n+1) = p(1, n+1)$ and

$$q(d+1, n+1) = \min_{d \le k < n+1} [q(d,k) + p(k+1, n+1)],$$

you only need to compute the $p(k+1, n+1)$, $1 \le k \le n$, the $q(d,k)$ being already known.

The first part of the work, namely the computation of the coefficients $p(i,\ell)$ requires $O(n^3 r_{\max})$ elementary operations, and the dynamical programming part requires $O(n^2 D_{\max})$ operations. The global complexity of the algorithm is therefore of the order

$$n^3 r_{\max} + n^2 D_{\max}.$$

The implemented method is the former (off line), but for an actualization purpose, the latter method is preferable.

Now, on the last column of Q, there are the $q(d,n)$'s, $1 \le d \le n$, which are the minima of the contrast penalized in degree, to go from 1 to n with d pieces. Thus the last thing to do is to choose

$$\hat{D} = \arg \min_{d=1,\dots,n} \left[q(d,n) + c_1 \ln \begin{pmatrix} n-1 \\ d-1 \end{pmatrix} + c_2 \ln(d)^{c_3} \right].$$

Of course, the involved partitions must be stored, and not only their number of pieces.

As a summary, let us give the steps of the algorithm:

PROPOSITION 4.1. *A model is selected by the algorithm following the steps*:

1. *On any interval* $I = [i/n, (j+1)/n[$, *compute* $p_s^{\mathcal{B}}(i,j) = \sum_{t=0}^{s} [\sum_{k=0}^{j-i} Y_{k+i} B_t^I(k)]^2$ *for* $1 \le i \le j \le n$, $0 \le s \le r_{\max}$, *and for* $B_t^I = \mathrm{Cheb}_t^{\ell(I)}$ *and* $B_t^I = \mathrm{Trig}_t^{\ell(I)}$ (*see Subsections* 3.3.1 *and* 3.3.2),

2. *Compute* $p^{\mathcal{B}}(i,j) = \min_{1 \le s \le r_{\max}} (\sigma^2 (c_4 s + c_5 \ln(s)^{c_6}) - p_s^{\mathcal{B}}(i,j))$ *for* $1 \le i \le j \le n$,

3. *Compute* $p(i,j) = \min_{\mathcal{B} \in \{\mathrm{Cheb}, \mathrm{Trig}\}} p^{\mathcal{B}}(i,j)$,

4. *Initialize* $q(1,k) = p(1,k)$ *for* $1 \le k \le n$, *and compute recursively for* $1 \le d \le n-1$,

$$q(d+1, n) = \min_{d \le k < n} [q(d,k) + p(k+1, n)],$$

5. *Then choose* $\hat{D} = \arg\min_{d=1,\dots,n} [q(d,n) + c_1 \ln \binom{n-1}{d-1} + c_2 \ln(d)^{c_3}]$.

The positions of the knots of the involved partitions as well as the selected degrees in step 2 *and bases in step* 3 *must be stored.*

4.3 *A fast version of the algorithm*

We have also implemented a quick but approximated version of the algorithm, with complexity of order $D_{opt} r_{\max} n^2$. We must admit that other algorithms, like those based on wavelets for example, have a lower complexity, of the order $O(n \log_2(n))$. This is a drawback of our procedure.

Let us now describe our fast procedure. We construct iteratively a sequence of partitions defined by $0 = a_0^t < a_1^t < \cdots < a_{D_t-1}^t < a_{D_t}^t = 1$ using the following scheme.

- We start at time $t = 0$ with $D_0 = 1$, i.e. with the partition with a unique interval.
- At time t of the procedure,

– we first check, when $D_t > 3$, if the penalized contrast decreases when removing one a_j^t for $j = 1, \ldots, D_t - 1$ from the partition at time t. If some such points exist, we consider the one implying the most significant decrease.

– Next, we check if the penalized contrast decreases when adding one point a in between the a_j^t's for $j = 0, \ldots, D_t$ of the partition at time t. If some such points exist, we consider the one implying the most important decrease.

– If only one such points exist, we follow the associated strategy.

– If two such points exist, we choose the strategy which leads to the most important decrease.

– If no such point exists, the procedure stops.

During the procedure, on each involved interval, the best basis (Chebyshev or trigonometric polynomials), the best degrees and the best coefficients are chosen. The interest of such a procedure is that it requires only comparisons between

$$\hat{\gamma}^{\mathcal{B}}_{[a,b[}(r) + \mathrm{pen}_{n,c_1,c_2}(D_t) + \mathrm{pen}_{n,c_4,c_5,c_6}(r)$$

and

$$\hat{\gamma}^{\mathcal{B}'}_{[a,c[}(r') + \hat{\gamma}^{\mathcal{B}''}_{[c,b[}(r'') + \mathrm{pen}_{n,c_1,c_2}(D_t + 1) + \mathrm{pen}_{n,c_4,c_5,c_6}(r') + \mathrm{pen}_{n,c_4,c_5,c_6}(r'').$$

Without the potential "removing step", this procedure is like the expansion phase of a CART procedure, see Breiman *et al.* (1984). In that way, this procedure is related to the CART-type procedure proposed by Kohler (1999), except that he does not consider either the possibility of removing some points from the partition.

5. Empirical results

5.1 *Risks and calibration*

First we take for the penalty as defined in Proposition 3.1, $s^2 = \sigma^2$, $c_1 = c_4 = 2$. Concerning the determination of $c_1, \ldots, c_6$, we determined them as follows, using the test functions (and some others, to avoid over-fitting) described below. With the notations of formula (8), we took $c_4 = 2$ to be in accordance with standard Akaike criterion from an asymptotic point of view. For c_5 and c_6, we followed some ideas developed by Birgé and Rozenholc (2002) for the choice of the number of bins of an histogram in density estimation. This question is similar to the choice of the degree in polynomial regression. For c_1 and c_2, the problem amounts to the choice of an irregular partition when the degree is fixed, for example to zero. We used a rough optimization on the square $(c_1, c_2) \in [0, 5] \times [0, 5]$ to fix our choice. In any case, we claim that our global choice is satisfactory even if probably not optimal. This choice is the following:

$$c_1 = c_2 = c_4 = c_5 = 2, \qquad c_3 = c_6 = 2.5.$$

Secondly, we use a set of 16 test functions with very different shapes and regularity. The test functions are given in Fig. 1. Functions 1 to 4 and 7 to 14 are the same as the ones used by Antoniadis *et al.* (2002), functions 1 to 14 come from the Wavelab toolbox developed by Donoho (see Buckheit *et al.* (1995)) and functions 15 and 16 have been added in order to test also the estimators for some regular functions.

Third, we consider different levels (namely 3, 5, 7 , 10) of noise which are evaluated in terms of a signal to noise ratio, denoted by s2n, and computed as

$$\text{s2n} = \sqrt{\frac{\frac{1}{n}\sum_{i=1}^{n}(f(i/n) - \bar{f})^2}{\sigma^2}}, \qquad \bar{f} = \frac{1}{n}\sum_{i=1}^{n} f(i/n).$$

Lastly, the performances are usually compared to a reference value called an oracle. This oracle is the lowest value of the risk. It is computed by using the fact that, in the simulation study, the true function is known, and that, if all the estimates are computed, the one with the smallest risk can be found, as well as its associated risk. In other words in this case, we would have to compute all $||f - \hat{f}||_n^2$, for some known f and all possible $\hat{f}$, for all the sample paths. This can be done for regular models but would imply much too heavy computations for the general models considered here. Therefore, another reference must be found to evaluate our estimator.

Before giving the details about the wavelet methods, let us explain the reference we use. The index w_j denotes the wavelet method number j where 48 wavelet methods are considered and $\hat{f}_{w_j}$ denotes the estimate of f obtained by using the method w_j explained below. We generate K ($K = 100$) samples with length n ($n = 128, 512$) in the regression model, and denote by $\hat{f}^{(k)}$ an estimate of f (computed with any method, $\hat{f}_{w_j}^{(k)}$ with the

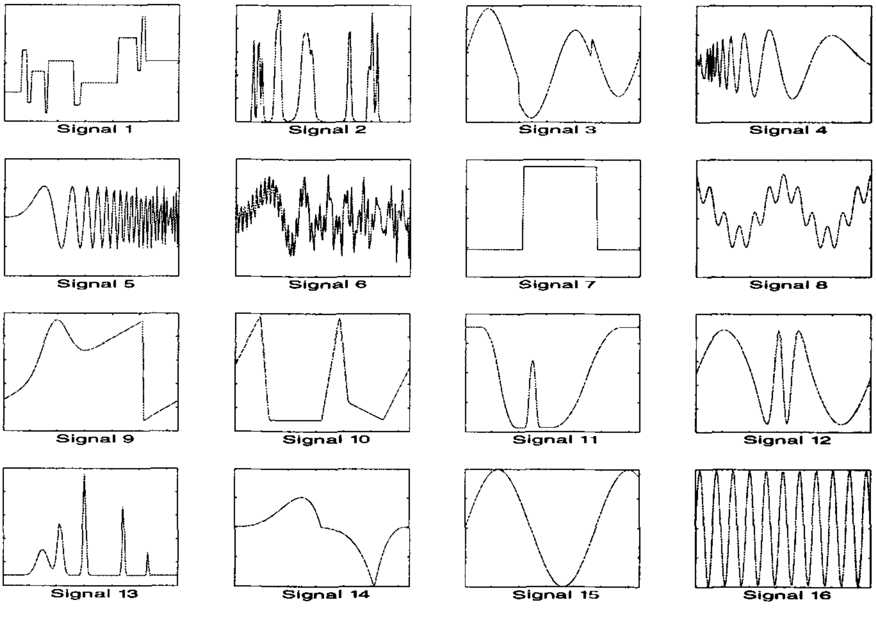

Fig. 1. Test functions.

method w_j) based on the k-th sample. Then

$$\ell_2^2(f, \hat{f}^{(k)}) = \frac{1}{n}\sum_{i=1}^{n}(f - \hat{f}^{(k)})^2(i/n),$$

and

$$\boldsymbol{E}^*[\ell_2^2(f, \hat{f})] = \frac{1}{K}\sum_{k=1}^{K}\ell_2^2(f, \hat{f}^{(k)}).$$

We use the following ratios

$$R_2(f) = \frac{\boldsymbol{E}^*[\min_{j=1,\ldots,48}\ell_2^2(f, \hat{f}_{w_j})]}{\boldsymbol{E}^*[\ell_2^2(f, \tilde{f})]}. \tag{5.1}$$

The ratios are compared to one: the higher over 1 the ratio, the better our method. If our test functions $f_1, \ldots, f_{16}$ lead to values of $R_2(f_i)$ for $i = 1, \ldots, 16$ such that $\forall i \in \{1, \ldots, 16\}$, $R_2(f_i) \geq a$, then this means that, for any $f \in \{f_1, \cdots, f_{16}\}$,

$$\boldsymbol{E}^*[\ell_2^2(f, \tilde{f})] \leq \frac{1}{a}\boldsymbol{E}^*\left[\min_{j=1,\ldots,48}\ell_2^2(f, \hat{f}_{w_j})\right].$$

We must emphasize that we chose diadic values of n ($n = 128 = 2^7$ or $n = 512 = 2^9$) in order to be able to apply all wavelet methods, but our method does not require diadic samples and can be used for any n without any change.

We present in Fig. 2 an example of data set and estimated signal as performed by our algorithm. The signal has been built with three pieces using functions 15, 8 and 6. The fourth picture gives the variation of the residuals and shows that the algorithm has found an estimator with four pieces, the first one is a standard polynomial with degree 6 corresponding to the estimation of the first function, the second one is a trigonometric polynomial with degree 26 corresponding to the estimation of the second function, the last two pieces correspond to the estimation of the third function, and are polynomials of degree 59 and 54.

5.2 *Comparison with wavelet methods*

5.2.1 *Comparison with standard wavelet methods*

Let us be more precise about the wavelets. We use both the MathWorks toolbox developed by Misiti *et al.* (1995) and the WaveLab toolbox developed by Buckheit *et al.* (1995), following the theoretical works by Donoho (1995), Donoho and Johnstone (1994), Donoho *et al.* (1995). The abbreviations below refer to the MathWorks toolbox. We use the 6 following basic wavelets: the Haar wavelet (well suited for step signals), two Daubechies DB4 and DB15 wavelets (well suited for smooth signals), two symmetric wavelets abbreviated as Symmlets, sym2 and sym8, the bi-orthogonal wavelet bior3.1 (well suited for signals with rupture). The wavelets are associated with 4 types of threshold: the threshold $\sqrt{2\log(n)}$ called "sqtwolog", the minimax threshold called "minimaxi", the SURE (Stein's Unbiased Risk Estimate) threshold called "Rigsure", an heuristical version of SURE threshold using a correcting term for small values of n, called "Heursure". Lastly, we use the two standard types of threshold, hard and soft thresholding. This explains the $6 * 4 * 2 = 48$ indexes for the wavelets methods. This is what we call in the following "standard wavelet methods", as opposed to some more recent methods described further.

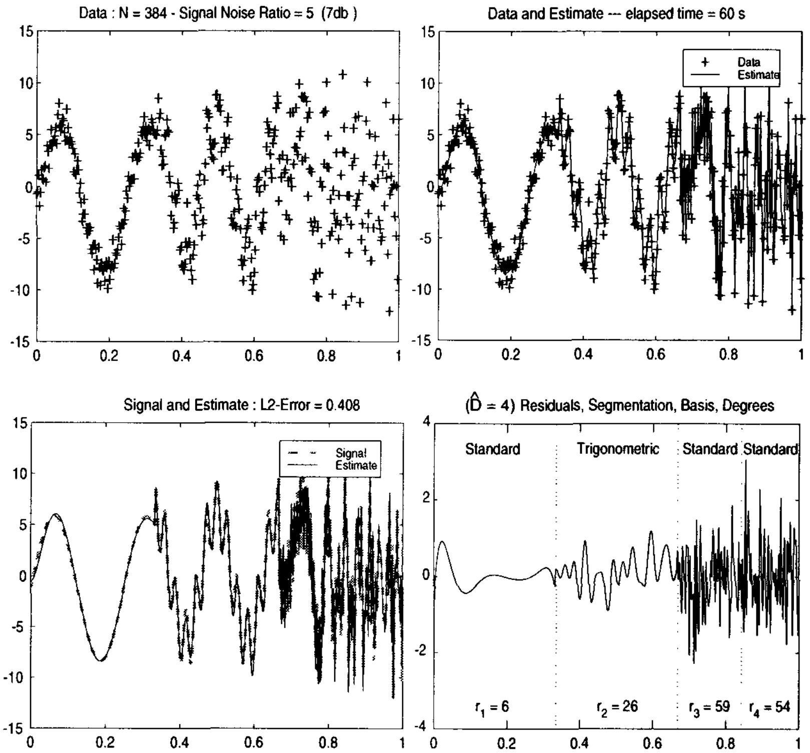

Fig. 2. An example of decomposition of a signal by the complete algorithm.

Note that the estimation is improved by using the function "wmaxlev" to select the maximum level of the wavelets instead of the standard level round$[\log_2(n)]$ and we therefore use this MathWorks function as well.

We report in Fig. 3 the performances of our estimate when the maximal degree is set to $r_{\max} = 74$ and for s2n $= 3, 5, 7, 10$, the functions f_i being as given in Fig. 1, $K = 100$ and $n = 128$. More precisely, Fig. 3 plots the values obtained for $R_2(f)$ relative to the test functions given in Fig. 1. We emphasize that the ratio we compute is very unfavorable to our method, because for each sample, we compare our risk to the best (unknown in practice) wavelet method. The ordinate of the lower point is anyway greater than 0.65, which is relatively good since we recall that it means that, for any $f \in \{f_1, \ldots, f_{16}\}$, $E^*[\ell_2^2(f, \tilde{f})] \leq 1.54 E^*[\min_{j=1,\ldots,48} \ell_2^2(f, \hat{f}_{w_j})]$.

We have also computed the risks in term of ℓ_1-type error (squares are replaced by absolute values) where we obtained the same type of results except for the functions f_8 and f_{16} where the results are even better: this can be explained by the fact that the ℓ_1-distance reduces the weights of the discontinuities which are inherent in our method. In addition, the errors are centered Gaussian with known variance, but we also considered centered uniform and Cauchy errors and the results were similar.

5.2.2 *Comparison with recent wavelet methods*

We also implemented for comparison some more recent wavelet methods, already studied in Antoniadis *et al.* (2002) and therefore quite reproducible for such a test. More precisely we considered the following methods, implemented using the Gaussian Wavelet

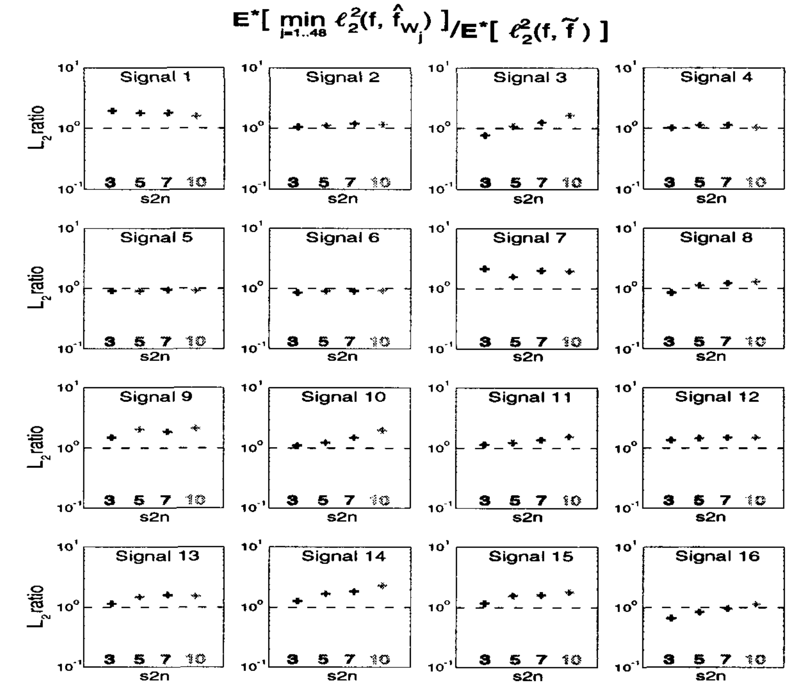

Fig. 3. Performance ratios $R_2(f)$ relative to the 16 test functions for $K = 100$ and $n = 128$. The greater than one, the better our algorithm.

Denoising Library built by Antoniadis *et al.* (2002) (see http://www.jstatsoft.org/v06/i06/), using either a Haar or a Symmlet8 filter:

W1 Coifman and Donoho (1995)'s translation invariant method using soft thresholding (TI-soft), coded with the function "recTI" in the library,

W2 Coifman and Donoho (1995)'s translation invariant method using hard thresholding (TI-hard), coded with the function "recTI" in the library,

W3 Cai (1999)'s method using a block non-overlapping thresholding estimator, reusing the first few empirical coefficients to fill the last block, coded with the function "recblockJS" in the library,

W4 Cai (1999)'s previous method, the last few remaining empirical coefficients being unused, coded with the function "recblockJS" in the library,

W5 Huang and Lu (2000)'s method based on nonparametric mixed-effect models, coded with the function "recmixed" in the library,

W6 Cai and Silverman (2001)'s method using an overlapping block thresholding estimator, coded with the function "recneighblock" in the library,

W7 Antoniadis and Fan (2001)'s hybrid method using a "keep", "shrink" or "kill" rule (SCAD),

W8 Vidakovic and Ruggeri (2001)'s bayesian adaptive multiresolution method coded with the function "recbams" in the library.

For a more precise description of those methods, we refer to Antoniadis *et al.* (2002). There are a selection of recent methods that Antoniadis *et al.* (2002) describe and test, namely methods number 5, 6, 12, 13, 20, 11, 18 and 34 respectively in their Table 3. We work first with $\sigma = 1$ assumed to be known. Moreover, in all the following of the subsection, we use the quick version of our algorithm.

Table 1 gives the L_2-errors for the 16 test functions and signal to noise ratio s2n = 5 obtained with the quick version of our method (when using Chebyshev polynomials (CP) or both Chebyshev and trigonometric polynomials (CP/TP)) and with the other methods W1 to W8. We must say that we did not succeed in making W8 work, but this may be an error of ours. Besides we found out that the method of Coifman and Donoho (1995) with hard thresholding (W2 or TI-hard) seems to be almost always better than all the other wavelet methods. Our method behaves very well and is in general better than all the other methods. Even when we do not have the lowest errors, we are close to it. Globally, the CP/PT method seems to be preferable: the losses are never very important but the gains are sometimes decisive, when compared to the wavelet methods in competition.

Since we found the method of Coifman and Donoho (1995) with hard thresholding (TI-hard) to be the better one, we present a more precise comparison of our results with theirs in Fig. 4, in order to illustrate the influence of the choice of the filter (either the Symmlet8 filter or the Haar filter) in the wavelet methods. Our method does not require such a choice, which seems to be sometimes decisive (Signals 7 and 8).

Lastly, we compared our method with others when σ is unknown. We implemented our method using a preliminary estimator of σ^2 based on the mean square residuals obtained with an estimator of the regression function computed on a regular model with $D = [n/\ln(n)]$ intervals in the subdivision and degree $r = 3$. This estimator is used for the penalization procedure. The estimate of f is then used to re-evaluate σ and to initialize a second penalization. For the test functions 5 and 6, it appears clearly that

Table 1. L_2-errors for s2n = 5, $n = 512$, CP is our method when considering piecewise Chebyshev polynomials only, CP/TP is our method when considering both Chebyshev and trigonometric (piecewise) polynomials, W1 to W8 are the wavelet methods described in Subsection 5.2.2 with Symmlet8 filter. $\sigma = 1$ is known. $\bullet$ gives the best wavelet method, * gives the best method between CP/TP and W1–W8.

Signal	CP	CP/TP	W1	W2	W3	W4	W5	W6	W7	W8
1	**0.077**	$\mathbf{0.098}^*$	1.856	$0.423_\bullet$	0.798	0.817	0.472	0.947	0.866	0.871
2	**0.313**	$\mathbf{0.361}^*$	2.421	$0.404_\bullet$	0.769	0.785	0.506	0.820	0.977	0.871
3	**0.061**	$\mathbf{0.063}^*$	0.207	$0.108_\bullet$	0.198	0.228	0.135	0.216	0.148	0.856
4	**0.197**	**0.202**	0.805	$0.174^*_\bullet$	0.251	0.282	0.238	0.197	0.335	0.857
5	**0.582**	$\mathbf{0.563}^*$	4.337	$0.651_\bullet$	0.724	0.746	0.670	0.677	1.600	0.883
6	**1.001**	$\mathbf{1.003}^*$	8.506	1.601	1.492	1.474	4.788	$1.406_\bullet$	3.816	4.050
7	**0.010**	$\mathbf{0.012}^*$	0.552	$0.133_\bullet$	0.388	0.405	0.193	0.416	0.285	0.860
8	**0.168**	$\mathbf{0.059}^*$	0.596	$0.072_\bullet$	0.239	0.256	0.273	0.253	0.398	0.864
9	**0.050**	$\mathbf{0.053}^*$	0.377	$0.080_\bullet$	0.159	0.177	0.143	0.198	0.159	0.859
10	**0.113**	**0.091**	0.313	$0.084^*_\bullet$	0.127	0.154	0.120	0.135	0.181	0.859
11	**0.087**	**0.087**	0.200	$0.052^*_\bullet$	0.108	0.140	0.088	0.082	0.091	0.856
12	**0.084**	**0.083**	0.235	$0.050^*_\bullet$	0.085	0.117	0.094	0.080	0.106	0.858
13	**0.181**	**0.141**	0.891	$0.136_\bullet$	0.258	0.269	0.234	0.246	0.365	0.867
14	**0.057**	**0.052**	0.176	$0.062^*_\bullet$	0.097	0.096	0.073	0.096	0.084	0.857
15	**0.027**	$\mathbf{0.027}^*$	0.203	$0.071_\bullet$	0.156	0.175	0.136	0.166	0.122	0.856
16	**0.156**	$\mathbf{0.076}^*$	0.735	$0.118_\bullet$	0.233	0.251	0.255	0.227	0.357	0.868

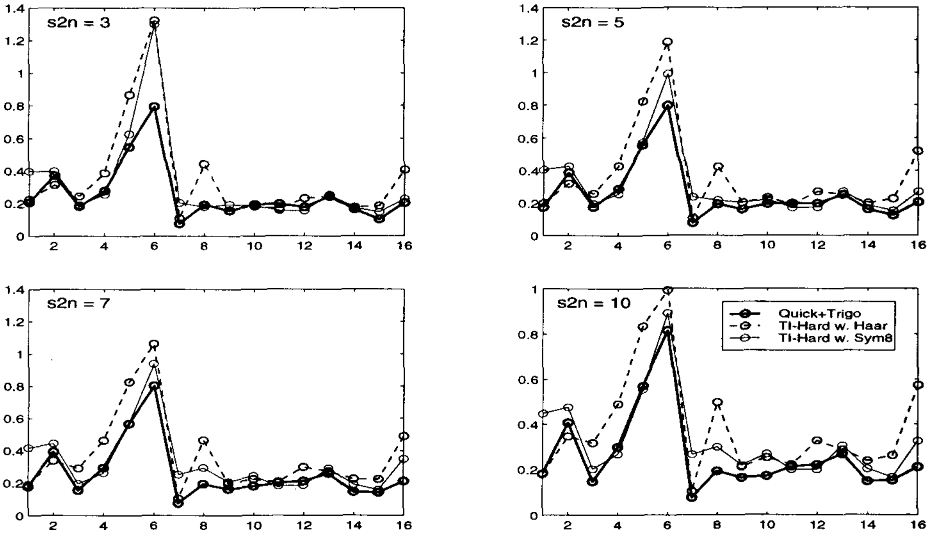

Fig. 4. Comparison of the L_2-errors for the 16 test functions and the four s2n, of our method, using both Chebyshev and trigonometric polynomials (thick curve) and Coifman and Donoho (1995)'s method for the Haar (dashed curve) and the Symmlet8 filters (thin curve). $K = 100$, $n = 512$, $\sigma = 1$ known.

almost no method makes the job in this case, neither wavelets, nor ours. The only good wavelet method is Huang and Lu (2000)'s method W5 which is never better than the other wavelet methods for the other signals. Note that the test functions 5 and 6 are not used by Antoniadis *et al.* (2002) in their experiments. In the other cases again and as shown by the results given in Table 2, one of the better wavelet methods remains Coifman and Donoho (1995)'s method, contrary to Antoniadis *et al.* (2002)'s conclusion that the best method highly depends on the type of the signal function. Note that we gave for this method the results using both the Symmlet8 and the Haar filter.

5.3 *Comparison of the complete and fast algorithms*

We have described in Subsection 4.3 an accelerated version of our complete algorithm, but we had to test if the performances of this method were indeed of the same order as the standard one, and nevertheless appreciably faster.

We give in Table 3 the estimation performances in term of $R_2(f)$ (with respect to the 48 standard wavelet methods) and in term of CPU time (for the same samples) of the accelerated algorithm compared to the standard one.

It appears that except for the first signal, which is better identified by the complete algorithm, the quick algorithm performs very well both in term of risk (which was expected) and time (which was the aim). More precisely and if we except Signal 1, there is essentially no loss in term of risk when using the quick algorithm, but it is between five and twenty times faster for a sample with size $n = 512$. This effect naturally increases with the sample size. As a conclusion, it is clear that both the standard and the accelerated algorithm work very well.

Table 2. L_2-errors for s2n = 5, n = 512, and σ = 1 is unknown. CP/TP is our method when considering both Chebyshev and trigonometric piecewise polynomials, W1 to W8 are the wavelet methods described above with Symmlet8 filter, W2H is the method W2 when using the Haar filter. $_\bullet$: best wavelet method, *: best method.

Signal	CP/TP	W1	W2	W2H	W3	W4	W5	W6	W7	W8
1	**0.092***	2.08	0.494	*0.111*$_\bullet$	0.914	0.934	0.470	1.070	0.976	0.871
2	**0.367***	2.990	0.461	*0.385*$_\bullet$	0.946	0.960	0.510	1.030	1.200	0.871
3	**0.063***	0.209	0.111$_\bullet$	*0.120*	0.201	0.230	0.136	0.217	0.149	0.856
4	**0.206**	0.873	0.187$^*_\bullet$	*0.461*	0.274	0.305	0.240	0.215	0.365	0.857
5	**4.690**	11.4	2.57	*16.1*	2.38	2.38	0.694$^*_\bullet$	2.4	5.98	0.884
6	**23.1**	23.7	23.5	*23.6*	23.6	22.8	0.986$^*_\bullet$	23.6	23.1	4.05
7	**0.013***	0.571	0.140	*0.026*$_\bullet$	0.404	0.420	0.192	0.437	0.295	0.86
8	**0.060***	0.610	0.073$_\bullet$	*0.290*	0.243	0.260	0.275	0.259	0.402	0.864
9	**0.052***	0.390	0.081	*0.064*$_\bullet$	0.167	0.184	0.144	0.208	0.164	0.859
10	**0.090**	0.319	0.083$^*_\bullet$	*0.104*	0.128	0.154	0.120	0.136	0.182	0.859
11	**0.088**	0.204	0.053$^*_\bullet$	*0.066*	0.110	0.142	0.089	0.083	0.092	0.856
12	**0.088**	0.240	0.050$^*_\bullet$	*0.123*	0.085	0.117	0.094	0.080	0.107	0.858
13	**0.150**	0.913	0.136$^*_\bullet$	*0.181*	0.265	0.274	0.233	0.250	0.373	0.867
14	**0.053***	0.179	0.063$_\bullet$	*0.072*	0.098	0.097	0.074	0.096	0.084	0.857
15	**0.028***	0.206	0.073$_\bullet$	*0.084*	0.158	0.177	0.138	0.169	0.124	0.856
16	**0.080***	0.751	0.118$_\bullet$	*0.427*	0.239	0.258	0.256	0.234	0.362	0.868

Table 3. Quick and complete algorithm comparison: Risk ratio and CPU time ratio, ratio = quick/complete, n = 512 and K = 100.

Signal/Ratio → ↓	s2n = 3		s2n = 5		s2n = 7		s2n = 10	
	Risk	Time	Risk	Time	Risk	Time	Risk	Time
1	1.55	0.24	1.53	0.25	1.37	0.25	1.46	0.25
2	0.93	0.18	0.95	0.18	0.96	0.17	0.99	0.16
3	1.05	0.10	1.00	0.11	1.01	0.12	1.00	0.14
4	0.95	0.13	0.95	0.13	0.96	0.14	0.96	0.14
5	0.99	0.19	1.00	0.19	0.99	0.18	0.98	0.18
6	0.99	0.19	1.00	0.20	0.99	0.20	1.00	0.21
7	1.00	0.13	1.00	0.14	1.00	0.14	1.00	0.14
8	0.97	0.08	0.98	0.08	0.98	0.08	1.00	0.08
9	0.96	0.11	0.96	0.13	0.96	0.13	0.95	0.13
10	0.93	0.13	1.04	0.14	1.04	0.15	1.21	0.15
11	1.03	0.10	1.11	0.12	1.10	0.13	1.03	0.14
12	1.00	0.13	1.04	0.13	1.07	0.13	1.09	0.13
13	0.93	0.17	0.97	0.17	0.98	0.16	1.01	0.16
14	0.95	0.10	0.94	0.10	0.92	0.10	0.94	0.10
15	0.96	0.05	0.98	0.05	1.00	0.06	0.99	0.07
16	1.04	0.13	1.04	0.13	1.14	0.13	1.08	0.13

5.4 *Compression performances*

We already computed the complexity of our algorithm so that it is clear that even in its quick version, it remains slower than wavelet methods. But it has two decisive advantages with respect to those methods, in addition to its completely automatic feature: first, it performs very well whatever the type of signal, and particularly when discontinuities arise, and secondly, its compression properties are reasonably good, and in particular much better in many cases than wavelets, which was not expected.

Therefore, we also made a simple comparison of the standard wavelet methods and of our algorithm in terms of their compression performances. For each estimated function, we compute three types of code lengths: an "integer length" which is a number of integers, namely the number of nonzero wavelet coefficients for the wavelet methods, $Nintw$, and twice the chosen number D of intervals in the Chebyshev piecewise polynomials method $NintCP$ (1 integer for D, D integers for each degree r_d, $D-1$ integers for the length of the intervals), a "real length" which is the number of real coefficients of the developments for each method, $Nrealw$ and $NrealCP$, and a global length, Nw and NCP, defined in both cases as $(Nint/4) + Nreal$ to take into account the fact that an integer is four times smaller than a real number in terms of code length.

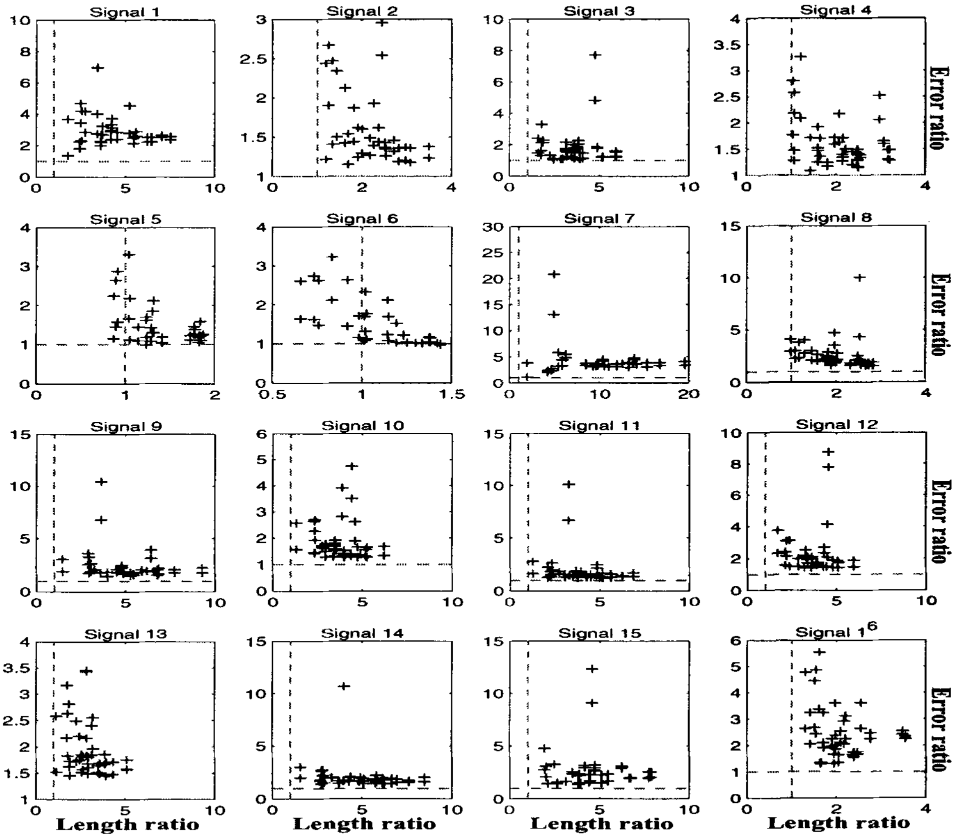

Fig. 5. Error ratio in function of the compression ratio. The ratios are our method over all wavelet methods (dotted lines correspond to levels 1). $K = 100$ samples with length $n = 128$ and s2n = 5.

Table 4. Compression performances for the three ratios in function of the s2n and of the signal for $K = 100$ samples with length $n = 128$.

	$Nrealw_j/NrealCP$				$Nintw_j/NintCP$				Nw_j/NCP			
	s2n				s2n				s2n			
Signal	3	5	7	10	3	5	7	10	3	5	7	10
1	2,57	2,05	2,09	2,22	1,61	1,20	1,24	1,26	2,30	1,79	1,84	1,93
2	2,92	2,89	1,52	1,47	4,32	4,20	2,35	2,44	3,13	3,08	1,64	1,59
3	2,85	3,57	3,33	3,24	5,25	6,46	6,31	7,30	3,14	3,92	3,68	3,64
4	2,23	2,12	2,11	1,20	7,61	8,72	9,74	6,86	2,59	2,50	2,50	1,44
5	1,47	1,01	1,17	1,06	16,74	12,58	14,98	13,64	1,80	1,23	1,43	1,29
6	1,13	1,16	1,19	1,22	28,89	33,98	35,01	36,30	1,40	1,44	1,47	1,52
7	2,35	2,26	2,40	2,36	1,32	1,33	1,35	1,34	2,03	1,98	2,08	2,05
8	1,64	2,33	2,55	2,43	9,06	18,00	21,97	22,10	1,96	2,83	3,10	2,95
9	1,35	3,71	3,55	2,80	1,80	5,56	5,69	4,94	1,42	3,97	3,84	3,06
10	3,00	4,97	5,19	5,75	4,80	6,87	6,26	6,27	3,24	5,26	5,37	5,84
11	2,18	4,72	4,85	4,68	3,54	8,07	8,89	8,84	2,36	5,15	5,33	5,17
12	4,43	3,15	4,67	4,50	8,79	7,48	11,41	11,52	4,92	3,57	5,30	5,12
13	3,50	3,67	3,68	1,85	5,10	5,40	5,63	2,93	3,73	3,92	3,95	1,99
14	3,40	2,49	2,46	6,11	5,49	4,53	4,68	12,79	3,68	2,73	2,72	6,82
15	2,09	3,45	3,14	3,21	3,42	6,41	6,04	6,85	2,26	3,80	3,47	3,59
16	1,40	1,61	1,83	1,97	9,95	13,01	15,99	18,50	1,69	1,95	2,23	2,40

Figure 5 above plots the error ratios $R^{(j)}(f) = \boldsymbol{E}^*[\ell_2^2(f, \hat{f}_{w_j})]/\boldsymbol{E}^*[\ell_2^2(f, \tilde{f})]$ in function of the global length ratios Nw_j/NCP where j is the index of the wavelet method, for f taken as each signal of Fig. 1 and for s2n = 5. We can see that both the estimation and the compression performances of our algorithm are most of the time better than wavelet methods since both ratios are higher than 1. Signals 5 and 6 are the only one for which the wavelets are better but it appears that they are better either in term of risk or in term of compression, but not both; on the contrary for the other signals, our method is better in terms of both risk and compression performance.

We give also in Table 4 the means (on the K samples) of the ratios: mean of $Nrealw_j$ divided by mean of $NrealCP$, mean of $Nintw_j$ divided by mean of $NintCP$ and mean of Nw_j divided by mean of NCP, for the index j corresponding to the best wavelet in term of approximation performance for each sample path; we also distinguish between the values s2n of the signal to noise ratios. We can see that all ratios are greater than 1, which means that our method is better in term of compression that all standard wavelets. In addition, the compression improvement of our algorithm increases when the s2n increases.

References

Abramowitz, A. and Stegun, I. A. (1972). *Handbook of Mathematical Functions*, Dover Publications, New York.

Antoniadis, A. and Fan, J. (2001). Regularization of wavelets approximations (with discussion), *Journal of the American Statistical Association*, **96**, 939–967.

Antoniadis, A. and Pham, D. T. (1998). Wavelet regression for random or irregular design, *Computational Statistics and Data Analysis*, **28**, 353–369.

Antoniadis, A., Bigot, J. and Sapatinas, T. (2002). Wavelet estimators in nonparametric regression: A comparative simulation study, *Journal of Statistical Software*, **6** (see http://www.jstatsoft.org/v06/i06).

Baraud, Y. (2000). Model selection for regression on a fixed design, *Probability Theory and Related Fields*, **117**, 467–493.

Baraud, Y. (2002). Model selection for regression on a random design, *ESAIM Probability and Statistics*, **6**, 127–146.

Baraud, Y., Comte, F. and Viennet, G. (2001*a*). Adaptive estimation in an autoregressive and a geometrical β-mixing regression framework, *The Annals of Statistics*, **39**, 839–875.

Baraud, Y., Comte, F. and Viennet, G. (2001*b*). Model Selection for (auto-)regression with dependent data, *ESAIM Probability and Statistics*, **5**, 33–49.

Barron, A. and Cover, T. M. (1991). Minimum complexity density estimation, *IEEE Transactions on Information Theory*, **37**, 1037–1054.

Barron, A., Birgé, L. and Massart, P. (1999). Risks bounds for model selection via penalization, *Probability Theory and Related Fields*, **113**, 301–413.

Birgé, L. and Massart, P. (1998). Minimum contrast estimators on sieves: Exponential bounds and rates of convergence, *Bernoulli*, **4**, 329–375.

Birgé, L. and Massart, P. (2001). Gaussian model selection, *Journal of the European Mathematical Society*, **3**, 203–268.

Birgé, L. and Rozenholc, Y. (2002). How many bins should be put in a regular histogram. Preprint du LPMA 721, http://www.proba.jussieu.fr/mathdoc/preprints/index.html.

Breimann, L., Friedman, J. H., Olshen, R. H. and Stone, C. J. (1984). *Classification and Regression Trees*, Wadsworth, Belmont, California.

Buckheit, J. B., Chen, S., Donoho, D. L., Johnstone, I. M. and Scargle, J. (1995). About WaveLab, Tech. Report, Department of Statistics, Stanford University, Stanford, California. available http://www-stat.stanford.edu/wavelab

Cai, T. T. (1999). Adaptive wavelet estimation: A block thresholding and oracle inequality approach, *The Annals of Statistics*, **27**, 898–924.

Cai, T. T. and Silverman, B. W. (2001). Incorporating information on neighboring coefficients into wavelet estimation, *Sankhya, Series B*, **63**, 127–148.

Castellan, G. (2000). Sélection d'histogrammes à l'aide d'un critère de type Akaike (Histograms selection with an Akaike type criterion), *Comptes Rendus de l'Académie des Sciences. Paris. Série I. Mathematique*, **330**, 729–732.

Coifman, R. R. and Donoho, D. L. (1995). Translation-invariant de-noising (eds. Antoniadis, A. and Oppenheim, G.), *Wavelets and Statistics*, Lecture Notes in Statistics, **103**, 125–150, Springer-Verlag, New York.

Comte, F. and Rozenholc, Y. (2002). Adaptive estimation of mean and volatility functions in (auto)-regressive models, *Stochastic Processes and Their Applications*, **97**, 111–145.

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

Donoho, D. L. (1995). Denoising by soft-thresholding, *IEEE Transactions on Information Theory*, **41**, 613–627.

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

Donoho, D. L., Johnstone, I. M., Kerkyacharian, G. and Picard, D. (1995). Wavelet shrinkage: Asymptopia (with discussion), *Journal of the Royal Statistical Society Series B*, **57**, 371–394.

Efromovich, S. and Pinsker, M. (1984). Learning algorithm for nonparametric filtering, *Automatic Remote Control*, **11**, 1434–1440.

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

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

Huang, S. Y. and Lu, H. H.-S. (2000). Bayesian wavelet shrinkage for nonparametric mixed-effects models, *Statistica Sinica*, **10**, 1021–1040.

Kanazawa, Y. (1992). An optimal variable cell histogram based on the sample spacings, *The Annals of Statistics*, **20**, 291–304.

Kohler, M. (1999). Nonparametric estimation of piecewise smooth regression functions, *Statistics and Probability Letters*, **43**, 49–55.

Li, K. C. (1987). Asymptotic optimality for c_p, c_l cross-validation and generalized cross-validation: Discrete index set, *The Annals of Statistics*, **15**, 958–975.

Lindstrom, M. J. (1999). Penalized estimation of free-knot splines, *Journal of Computational and Graphical Statistics*, **8**, 333–352.

Mallows, C. L. (1973). Some comments on C_p, *Technometrics*, **15**, 661–675.

Misiti, M., Oppenheim, G. and Poggi, J.-M. (1995). *The Wavelet Toolbox* (ed. The MathWorks).

Polyak, B. T. and Tsybakov, A. B. (1990). Asymptotic normality of the c_p test for the orthogonal series estimation of regression, *Theory of Probability and Its Applications*, **35**, 293–306.

Shibata, R. (1981). An optimal selection of regression variables, *Biometrika*, **68**, 45–54.

Vidakovic, B. and Ruggeri, F. (2001). BAMS method: Theory and simulations, Special Issue on Wavelets, *Sankhya Series B*, **63**(2), 234–249.