Adaptive lifting for nonparametric regression

Matthew A. Nunes[*], Marina I. Knight and Guy P. Nason
Department of Mathematics, University of Bristol, UK

20th January 2006

Abstract

Many wavelet shrinkage methods assume that the data are observed on an equally spaced grid of length of the form 2^J for some J. These methods require serious modification or preprocessed data to cope with irregularly spaced data. The lifting scheme is a recent mathematical innovation that obtains a multiscale analysis for irregularly spaced data.

A key lifting component is the “predict” step where a prediction of a data point is made. The residual from the prediction is stored and can be thought of as a wavelet coefficient. This article exploits the flexibility of lifting by adaptively choosing the kind of prediction according to a criterion. In this way the smoothness of the underlying ‘wavelet’ can be adapted to the local properties of the function. Multiple observations at a point can readily be handled by lifting through a suitable choice of prediction. We adapt existing shrinkage rules to work with our adaptive lifting methods.

We use simulation to demonstrate the improved sparsity of our techniques and improved regression performance when compared to both wavelet and non-wavelet methods suitable for irregular data. We also exhibit the benefits of our adaptive lifting on the real inductance plethysmography and motorcycle data.

Keywords: Curve estimation, lifting, nonparametric regression, wavelets.

1 Introduction

Wavelet shrinkage is used extensively for estimating functions which have been corrupted by noise. We assume a basic knowledge of wavelet shrinkage but reiterate key points throughout the paper. Recall that the general idea is to wavelet transform the observed data, for example using the discrete wavelet transform (DWT), threshold the wavelet coefficients, and then invert the transform to form an estimate.

This paper proposes *adaptive lifting*, a wavelet-like decomposition motivated by the aim of adapting the smoothness of *each* wavelet basis function to the local features of the given data. We will demonstrate that our adaptive ‘wavelets’ possess good compression and denoising properties for irregularly spaced data. Many real data sets contain multiple observations at single data points which our adaptive lifting methodology can readily handle (and which classical wavelet shrinkage usually cannot).

[*] Corresponding author: `Matt.Nunes@bristol.ac.uk`

We have found that modified versions of empirical Bayes methods described by Johnstone and Silverman (2004a,b, 2005) work well for shrinkage on our adaptive lifting transforms. We shall show through simulations that our two adaptive lifting transforms perform well against other smoothing methods, such as *Locfit* (Loader, 1997, 1999), smoothing splines and the wavelet method introduced by Kovac and Silverman (2000) for irregular data.

Section 2 reviews wavelet shrinkage methods for both regularly and irregularly spaced data and reviews the lifting scheme in general terms. Section 3 introduces our adaptive versions of the coefficient-by-coefficient lifting scheme and discusses methods of shrinkage for nonparametric regression. Section 4 gives some simulation preliminaries. Section 5 investigates the sparsity of adaptive lifting coefficient sequences on simulated data and compares it to other wavelet methods. Section 6 presents a comprehensive simulation study that compares our new methods to other wavelet and non-wavelet techniques and also illustrates our methods on real data. Section 7 concludes and outlines ideas for further work.

2 Multiscale methods for regression

Multiscale methods for nonparametric regression have become increasingly popular over the last decade. See Vidakovic (1999); Abramovich *et al.* (2000); Percival and Walden (2000) for reviews.

A popular model that dominates the wavelet shrinkage literature is given by

$$f_i = g(x_i) + \varepsilon_i, \tag{1}$$

for $i = 1, \ldots, n$. The goal of wavelet shrinkage is to estimate g from the f_i sequence. Most of the work in this area is univariate and many methods, implicitly or explicitly, cope only with situations that further assume:

1. The regression ordinates x_i are equally spaced. Typically $x_i = i/n$ for integers $i = 1, \ldots, n$;
2. n is a power of two;
3. The ε_i are independent and identically distributed and sometimes assumed to be Gaussian;
4. For each i there is one (and only one) f_i.

These assumptions are not satisfied by many real data sets: for example, the well-known `mcycle` motorcycle data from Silverman (1985). The methods we develop here are designed to work with irregularly spaced data sets of any length, with the possibility of multiple f_i for each i. We also consider departures from the above third assumption later in Section 3.3.

2.1 Review of wavelet methods for regular designs

By now wavelet shrinkage with the classical DWT is well known. To help establish our notation for what comes later we shall briefly outline the procedure.

A multiresolution analysis of a function $g(x) \in L^2(\mathbb{R})$ is an expansion given by

$$g(x) = \sum_{k \in \mathbb{Z}} c^*_{0,k} \varphi_{0,k}(x) + \sum_{j \geq 0} \sum_{k \in \mathbb{Z}} d^*_{j,k} \psi_{j,k}(x), \tag{2}$$

where $\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k)$ and similarly for $\varphi_{j,k}(x)$. The functions $\psi(x)$ and $\varphi(x)$ are called the mother wavelet and scaling function respectively. The name wavelet arises from $\psi(x)$ being an oscillatory function of short extent, ideally of compact support. For certain choices of $\psi \in L^2(\mathbb{R})$, the family $\{\psi_{j,k}(x)\}_{j,k\in\mathbb{Z}}$ forms an orthonormal wavelet basis of $L^2(\mathbb{R})$, in which case $\{d^*_{j,k}\}_{j,k\in\mathbb{Z}}$ is given by $d^*_{j,k} =< g, \psi_{j,k} >$ and provides information about g at scale 2^{-j} near position $2^{-j}k$.

For an equally spaced sequence $\mathbf{g} = \{g(i/n)\}_{i=1}^n$ for $n = 2^J$ the DWT computes discrete (periodic) wavelet coefficients $\mathbf{d}^* = \{d^*_{j,k}\}_{j=0,\ldots,J-1;k=0,\ldots,2^j-1}$, where the $d^*_{j,k}$ have the same scale and location interpretation as before, but now about the sequence $\mathbf{g}$ rather than the function $g(x)$. Note that we abuse notation by keeping the same symbols for both the discrete and continuous wavelet coefficients. It is also possible to decompose the sequence $\mathbf{g}$ down to some primary resolution level, L, so that the discrete wavelet coefficients are $\mathbf{d}^* = \{c^*_{L,\ell}, d^*_{j,k}\}$ for $\ell = 0, \ldots, 2^L - 1$ and $j = L+1, \ldots, J-1$ and $k = 0, \ldots, 2^{j-1}$. In other words, the coefficients are divided into scaling function coefficients $c^*_{L,\ell}$ (which carry information about the average of the sequence $\mathbf{g}$ at a given scale L and location $2^{-L}\ell$) and wavelet coefficients $d^*_{j,k}$ (as before). The DWT can be expressed as a matrix multiplication $\mathbf{d}^* = W\mathbf{g}$, where W is an orthogonal matrix derived from the $\psi(x)$ wavelet. However, in most implementations the fast DWT of Mallat (1989) is used for computing the DWT.

Applying the DWT to model (1) yields:

$$\mathbf{d} = \mathbf{d}^* + \mathbf{e}, \tag{3}$$

where $\mathbf{d}$ is the DWT of $\mathbf{f} = \{f_i\}_{i=1}^n$ and $\mathbf{e}$ is the DWT of $\{\varepsilon_i\}_{i=1}^n$. The wavelet shrinkage problem then becomes how can one estimate $\mathbf{d}^*$ from the $\mathbf{d}$? For many real signals the wavelet transform of g is sparse and hence a thresholding approach is often adopted and successful. For further information on carrying out the DWT and basic thresholding in a statistical context see Nason and Silverman (1994); for early work on wavelet shrinkage see Donoho and Johnstone (1994, 1995); Donoho *et al.* (1995); see the above mentioned references for further reviews.

Here we have only mentioned the most popular orthonormal transform linked to an orthonormal basis of wavelets. More general and flexible schemes include Riesz bases or bases where there are biorthogonal analysis and synthesis wavelets.

There are several families of wavelets that one might use. Popular families include the Daubechies (1992) compactly supported wavelets which offer wavelets with widely varying degrees of smoothness. A common question, not well answered by the existing wavelet shrinkage literature is “which wavelet should I use?” General advice is to “use a wavelet that matches or exceeds the smoothness characteristics of $g(x)$”. Of course, in practice, $g(x)$ is unknown and so the advice is not always of much help. Some work (e.g. Nason (2002)) suggests using cross-validation to choose the wavelet smoothness, which can help, but not for functions that possess varying degrees of smoothness in different locations.

One of the philosophies of our *adaptive lifting* is that not only do the (effective) smoothing parameters change over the domain of the data but *also* the (smoothness) class of the basis functions, and so the precise choice of wavelet no longer rests with the user.

2.2 Review of wavelet methods for irregular designs

Various methods have been proposed to adapt irregularly spaced data to the “equally-spaced” DWT. We discuss the advantages and disadvantages of some of these.

Cai and Brown (1998) proposed taking into account the irregularity by using the correspondence $x_i = H^{-1}(i/n)$, where H is a strictly increasing function which usually needs to be estimated. This allows the mapping of the function g collected on an irregular grid into the function $g \circ H^{-1}$ collected on a regular grid, which hence can be estimated. By composing this estimator with H, an estimator of g can be obtained. Yet this proves not to be a good estimator, especially when g is much smoother than H. In these conditions, for piecewise Hölder functions, an estimator of g is constructed, based on the wavelet decomposition of $\mathrm{Proj}_{V_J} n^{-1/2} \sum_{i=1}^{n} f_i \varphi_{J,i}^{per}(x)$. A new threshold is obtained by a generalization of the Donoho and Johnstone (1994) VisuShrink. The final estimator is proved to enjoy nice theoretical properties.

Sardy *et al.* (1999) proposed four ways (comparable in performance) of extending the Haar wavelet transform to irregularly spaced data, and then adapted VisuShrink to the modified transform. Out of the proposed transforms, the isometric Haar wavelets are computationally simplest, and the authors point out that they can be generalized to wavelets of higher order than Haar.

Cai and Brown (1999) show that when the x_i are distributed independently on [0,1] the wavelet method with universal threshold can be applied directly, as if the grid were regular. The motivation is the use of the approximation $x_{(i)} \sim E(x_{(i)}) = i/(n+1)$, and so the observations $(i/(n+1), f_i)$ are considered instead of (x_i, f_i). This estimator is also within a logarithmic factor of the minimax risk over a range of Hölder functions.

Kovac and Silverman (2000) map irregularly spaced data, $\mathbf{f}$, to a regular grid, $\tilde{\mathbf{f}}$, by a linear interpolation of the original noisy values: $\tilde{\mathbf{f}} = R\mathbf{f}$ where the matrix R describes the interpolation. The usual wavelet shrinkage can be applied to $\tilde{\mathbf{f}}$ and, additionally, to permit accurate thresholding, the variance of the wavelet coefficients can be computed using a fast wavelet algorithm akin to the 2D DWT. To simultaneously handle the choice of wavelet, primary resolution level and threshold for estimating the true function, Nason (2002) developed a fast cross-validation algorithm, able to work on irregular grids using the Kovac-Silverman (KS) procedure.

Antoniadis and Fan (2001) formulated a penalized least squares problem in terms of the unknown wavelet coefficients of $g(i/n)$. Initially assuming a regular grid and $n = 2^J$, and under certain conditions on the penalty function, they proved that a solution exists and is unique. They also introduced a new universal threshold, proved to produce estimators with smaller risk than by using the classical one. The procedure was extended to irregular data by constructing non-linear regularized Sobolev interpolators as estimators and then improving them by constructing a regularized one-step estimator.

In Pensky and Vidakovic (2001) the x_i's are endowed with a probability space: X is considered to be a random variable, with density function say h, to be estimated. The regression function $E(f|X = x)$ is then estimated by its projection on the space V_J of a multiresolution analysis, i.e. $\sum_k \hat{c}_{J,k} \varphi_{J,k}$, where $\hat{c}_{J,k}$ is an estimator of $c_{J,k}$ based on $\varphi_{J,k}$ and the estimator of h. The final estimator has good properties, provided that h is reasonably smooth.

All the above methods enable wavelet shrinkage to be carried out for irregularly spaced data. However, some of them assume models for the grid values that either might not apply (e.g. uniformly distributed x_i), or require the estimation of additional quantities (e.g. the function H or the density h) that might be unreliable for small sample sizes. For interpolation methods, choices need to be made, such as location and spacing of the regular grid or interpolation method, which will influence performance. For some of the other methods, unreasonable assumptions on the smoothness of g are made which might not hold in practice.

In our simulation study later we also compare our new methods with two non-wavelet methods: *Locfit* and `smooth.spline()`. *Locfit* is a technique that implements smoothing by using local regression. A polynomial model is fitted to the data within a sliding window whose bandwidth controls the smoothness of the fit. The S-Plus function `smooth.spline()` finds the function $\mu(x)$ which minimizes $\sum_i \{f_i - \mu(x_i)\}^2 + \lambda \int \mu''(x)^2 \, dx$. The solution to this optimization problem is a piecewise cubic spline function. The parameter λ controls the balance between the fidelity to the data and the 'wiggliness' of the curve. A detailed discussion of both these methods is beyond the scope of this paper so we refer the reader to Loader (1999, 1997) for details on *Locfit* and Green and Silverman

(1994) for details on smoothing splines.

2.3 Lifting

Recently, a new mathematical technique called *lifting* was developed, which enabled wavelet-like multiresolution analyses to be applied to very general data situations. In particular, lifting can be applied to irregularly spaced data.

We will first provide an informal introduction to the lifting scheme as introduced by Sweldens (1996, 1997). Lifting a signal consists of three main steps: split, predict (dual lifting) and update (primal lifting) which we now describe. Split: the points f_i are separated into odd- and evenly-indexed sets. The next task is to *predict* the odd-indexed f-values by using the information contained in the evenly-indexed f-values. Of course, good prediction is only possible if the function f possesses some degree of local smoothness. The prediction error (the difference between the true function values on the odd positions and the predicted values on the same positions) is then quantified in a vector, called a *detail vector*. The final update step consists of *updating the f-values on the even positions* by using a linear combination of the (old) evenly-indexed f-values and the detail vector. The purpose of this update stage is to preserve in the updated values some quantity of the initial signal (such as the mean value of the signal) over successive repetitions of this procedure. After iterating the split-predict-update procedure on the updated values, the initial data, $\mathbf{f}$, is replaced by the remaining updated subsample (which reproduces the coarse scale features of the signal) and the detail coefficients which accumulate throughout the process. This is similar to the DWT which replaces $\mathbf{f}$ by a set of father and mother wavelet coefficients.

The above construction can be put in a formal biorthogonal multiresolution analysis (MRA) framework, that parallels the one that introduces the classical wavelets, with a few important differences (see Sweldens, 1997). One very important difference, is that the basis functions are no longer necessarily dilations and translations of the same functions. Also, the lifting construction generates bases that are not guaranteed to be Riesz, and hence they can exhibit stability problems, which we investigate for our construction below. (Stability issues with odd/even splits and suggestions for improvements appear in Simoens and Vandewalle (2003). Also Vanraes *et al.* (2002) show that increasing grid irregularity can result in stability problems and propose two stabilization schemes).

Delouille *et al.* (2001) and Delouille *et al.* (2004) both describe some of the first lifting techniques for nonparametric regression of irregularly spaced data. The first of these nice papers describe the unbalanced Haar transform, a way of adapting Haar wavelets to an irregular design. The second builds on this work and enables smoother 'wavelets' using interpolation in the predict step. Both of these techniques make use of a dyadic-like partitioning of the interval on which they are working.

In more than one dimension it is not possible to use the even/odd split, but there are various alternatives. In our work, described below, we adopt the 'one coefficient at a time' split proposed by Jansen *et al.* (2001, 2004), where the current scaling coefficients are split into two groups, but one group contains *only one coefficient* that gets predicted by its neighbours from the other group.

The 'one coefficient at a time' approach also makes it extremely easy to introduce adaptivity to the lifting: every time a coefficient gets removed we can make a choice about how we lift it to achieve the 'best' wavelet coefficient at that particular point.

3 Adaptive lifting

We introduce an *adaptive* lifting algorithm which forms a key component of our nonparametric regression algorithm. The advantages are:

1. The algorithm is highly adaptive. This means that, *at each step*, a choice of the type of prediction is made: this permits a local choice of vanishing moments (wavelet smoothness). At each step we try a number of possibilities for prediction and choose the one which gives maximum compression (measured as the smallest absolute value of the wavelet coefficient).

2. Lifting coefficients resemble wavelet coefficients and so many techniques previously designed for 'denoising' wavelet coefficients can also be used here.

3. The algorithm is computationally efficient $\mathcal{O}(n)$.

4. It is simple to handle the situation of multiple y values for a given x value.

Like many manifestations of lifting that have gone before, our technique is entirely computational. However, comprehensive simulations, shown later, demonstrate that adaptive lifting is competitive. Of course, a statistical theory for its asymptotic or even a mathematical theory for the lifting smoothness properties would be extremely interesting, but challenging.

3.1 Previous adaptive lifting schemes

The ideal of adaptivity in lifting is clearly appealing. Most adaptive lifting has been used in a 2D context (mostly for image compression) although some 1D studies exist.

Claypoole *et al.* (2003) proposed an adaptive lifting scheme for image compression. Their prediction step consists of adaptively choosing from a set of linear predictors (the $(1, N)$ branch of the Cohen-Daubechies-Feauveau family), in such a way that if an edge is detected in the image, then the wavelet is chosen such that its support does not overlap the edge. In order not to send information on the predictor being chosen when applying the algorithm for lossy coding, and to ensure that the update stage preserves the frequency localization, the update stage is applied first. In this way all the 'scaling function' coefficients (down to the coarsest scale) are obtained through updating, and then quantized. Then the prediction stage is applied to the quantized coefficients, and the detail coefficients are computed, quantized and transmitted. The "update first" approach originates in the paper Claypoole *et al.* (1998), where two adaptive algorithms are proposed: the scale-adapted transform in which the predictor gets adapted to match the signal structure at each scale, and the space-adapted transform, which chooses from a family of predictors the one that minimizes each detail value. A small simulation study on regular grids is provided, which shows that the proposed algorithms give very similar results, sometimes slightly better than those obtained if using the Daubechies wavelets on denoising the classical Donoho-Johnstone signals.

Piella and Heijmans (2002) also follow an update first strategy, followed by the prediction step, but unlike Claypoole *et al.* (1998) they introduce adaptiveness into the update stage, leaving the predictor of a fixed form. The algorithm's behaviour is briefly investigated by denoising a few signals, and only compared to the results produced by its fixed linear version, no other comparisons being made.

Trappe and Liu (2000) built adaptiveness into the prediction step which has the goal of minimizing the l_2-norm of the signal by using Wiener filtering. This adaptive algorithm was used for decorrelating the low-pass and high-pass subbands of an AR(2) process, and then for the shrinkage of the same process, corrupted by Gaussian noise. Boulgouris *et al.* (2001) develop an adaptive lifting scheme for the still image lossless compression.

In all the above work the philosophy is to choose the 'wavelet functions' locally to represent the signal in the most efficient way. The above adaptive lifting techniques use the usual odd/even splitting, whereas we augment the 'one coefficient at a time' methodology of Jansen *et al.* (2001) with adaptiveness. Further, we address the statistically important case of multiple y values for each given x.

3.2 Adaptive lifting one coefficient at a time

First we summarize a slightly simplified 1D version of the multivariate lifting 'one coefficient at a time' algorithm of Jansen *et al.* (2001). Then we explain how to make the algorithm adaptive.

3.2.1 Lifting one coefficient at a time

Suppose we have a function f, sampled at n irregularly-spaced points, x_i, on the real line. Our aim is to transform the sampled function values by means of lifting into a set of detail and scaling coefficients. Since we work in 1D, we can order the x-values and associate intervals to each point. A simple way of doing this is to construct intervals having the endpoints as the midpoints between the initial grid points.

In order to express the initial (input) function as a linear combination of scaling functions, we take the initial scaling functions to be the characteristic functions of the intervals associated with each point. Hence, we then have the property that $\varphi_{n,k}(x_i) = \delta_{i,k}$, for $k, i \in \{1, \dots, n\}$, and f can be expressed as

$$f(x) = \sum_{k=1}^{n} c_{n,k}\varphi_{n,k}(x), \tag{4}$$

where $f(x_i) = \sum_{k=1}^{n} c_{n,k}\delta_{i,k} = c_{n,i}$. In this way, the function values on the irregular grid are used as the initial scaling coefficients.

For the first lifting step (say, stage n) a point to be lifted must be chosen. We choose the point to be lifted, j_n, such that $\int \varphi_{n,j_n}(x)\,dx = \min_{k\in\{1,\dots,n\}} \int \varphi_{n,k}(x)\,dx$. By using the minimum scaling function integral, we choose the point with the finest detail. Since we use the interval construction to represent the scaling function integrals, the smaller integral values correspond to regions where the function has been densely sampled, and thus removal of a point will only cause small information loss in the signal. So the first coefficients to be obtained are the ones corresponding to the finest detail, with further steps eliciting progressively coarser detail.

After choosing the point to be removed, j_n, we identify its set of neighbours, I_n. Since there is a one-to-one correspondence between the point to be removed and its removal stage, we index each set of neighbours by the stage (here, n). We use the neighbours to predict the value of the function at j_n using simple regression techniques. Both neighbourhood definition and regression method are crucial to our installation of adaptivity, and so we defer their description to the next section.

The prediction phase yields an estimate of the form $\sum_{i\in I_n} a_i^n c_{n,i}$, where a^n are the weights resulting from the regression procedure over I_n. If j_n has only one neighbour, i, then the prediction is $f(x_i)$. The detail coefficient will be obtained from

$$d_{j_n} := c_{n,j_n} - \sum_{i\in I_n} a_i^n c_{n,i}, \tag{5}$$

or in the one neighbour case,

$$d_{j_n} := c_{n,j_n} - c_{n,i}. \tag{6}$$

The update phase only affects the scaling coefficients associated with the neighbouring points:

$$c_{n-1,i} := c_{n,i} + b_i^n d_{j_n},\ \forall i \in I_n, i \neq j_n. \tag{7}$$

For any $i \notin I_n$ ($i \neq j_n$) the scaling coefficients are unaffected, $c_{n-1,i} := c_{n,i}$. The aim of the update stage is to keep $\sum_{i \in I_n} c_{n,i} \int \varphi_{n,i}(x)\, dx$ constant across the scales, where the sum is indexed by the unlifted coefficients. In other words:

$$c_{n,j_n} \int \phi_{n,j_n}(x)\, dx + \sum_{i \in I_n} c_{n,i} \int \phi_{n,i}(x)\, dx = \sum_{i \in I_n} c_{n-1,i} \int \phi_{n-1,i}(x)\, dx, \tag{8}$$

and the b^n are obtained using this condition. Further, the integral associated with the removed point gets redistributed to its neighbours, see Jansen *et al.* (2001, 2004) for further details on these issues.

At this point, Jansen *et al.* (2004) state that the signal can be represented as

$$f(x) = d_{j_n} \psi_{j_n}(x) + \sum_{i \in \{1,\dots,n\} \setminus \{j_n\}} c_{n-1,i} \varphi_{n-1,i}(x), \tag{9}$$

where ψ_{j_n} and $(\varphi_{n-1,i})_i$ are the analogues of the usual wavelet and scaling functions. To summarize, we started with representation (4), a point, j_n, is identified, the scaling function ϕ_{n,j_n} is destroyed and a wavelet ψ_{j_n} is created with new coefficient d_{j_n}. All neighbouring scaling function coefficients of point j_n get updated. All this results in representation (9).

However, unlike the usual discrete wavelet case there are no neat analytical formulae for the scaling and wavelet functions. These functions are recursively constructed as the algorithm proceeds and depend on the locations of the input points $\{x_i\}$. It is possible to construct the wavelet functions by performing a forward transform on a zero function at the locations $\{x_i\}$, then inserting the value 1 at the location of the wavelet coefficient whose wavelet function you want to construct and applying the inverse lifting transform. This same method is often used to construct pictures of mother wavelets in the regular case.

More details on the precise interpretation and construction can be found in Jansen *et al.* (2001, 2004). Orthogonality of the wavelet and scaling functions is a desirable feature since it would ensure the stability of the transform, but it does not hold in this context.

After obtaining the predicted and updated values, the grid point j_n is removed, and the process repeated: a new point is chosen based on the minimum of the updated integrals, a neighbourhood structure is determined out of the remaining grid points (the ones which have not been removed yet nor chosen at the current stage), and the prediction and update steps are performed.

As a result, at the end of stage r (after points $j_n, j_{n-1}, \dots, j_r$ have been removed), the signal f will be represented as a linear combination of the $n - r + 1$ wavelet functions generated through the transform and of the remaining (updated) scaling functions, with the corresponding coefficients consisting of details and "low frequency" coefficients:

$$f(x) = \sum_{k \in \{n,n-1,\dots,r\}} d_{j_k} \psi_{j_k}(x) + \sum_{i \in \{1,\dots,n\} \setminus \{j_n, j_{n-1},\dots,j_r\}} c_{r-1,i} \varphi_{r-1,i}(x). \tag{10}$$

3.2.2 Adding adaptivity

Let us return to the issue of *neighbour choice*. We can employ prediction based on *symmetrical neighbours*: the same number of neighbours on the left and right of the removed point or choose *the closest neighbours* to the removed point irrespective of which side they lie. Our software allows for prediction with any number of neighbours.

For prediction we use regression of up to order three using the neighbours as explanatory variables (that is linear, quadratic or cubic regression over the specified neighbourhood). In wavelet language this corresponds to locally using more vanishing moments, which is of great utility when the function is locally smooth, but not when there are discontinuities present.

We want our transform to adjust itself to the local signal structure. Hence for each lifting step we permit two sources of adaptiveness: *order of the regression* and *neighbourhood size and configuration*. The two methods we introduce are:

AdaptPred. At each step the order of regression (linear, quadratic, cubic; with or without intercept) is chosen that generates the smallest detail coefficient in absolute value. The neighbourhood size and configuration are specified by the user. The ‘wavelet bases’ adapt themselves to the signal smoothness.

AdaptNeigh. In addition to *AdaptPred*, a choice is made over several possibilities of neighbourhood size and configuration, such that the smallest detail coefficient (in absolute value) is obtained. The considered neighbourhood configurations are symmetric neighbours up to and including a pre-specified number each side, and of closest neighbours up to twice the specified number.

At this point one might like to refer to Figures 6 and 7 which show two test signals decomposed with an AdaptNeigh algorithm. The plots show where linear, quadratic and cubic basis functions get placed.

Let us now make some remarks on the implications of the above constructions.

- In Jansen *et al.* (2001), for each stage r the prediction weights $(a_i^r)_{i \in I_r}$ are designed to sum to one. When regression with an intercept is used the sum is one, and areas where the function is constant yield exactly zero detail coefficients. When an intercept is not used the weights are not guaranteed to sum to one.

- A consequence of using closest neighbours is that the prediction weights can take negative values. This in turn will imply that when updating the corresponding integrals, they will become smaller rather than larger, contradicting the intuition that since we remove a point, each of the remaining points should span a larger set to account for the removed point. Hence note that for this situation, the observation in Jansen *et al.* (2004) that the scales of the wavelet functions are a monotonic function of the index, does not hold.

- The number of neighbours to be used in either of the two configurations (closest or symmetrical) is pre-specified by the user, and the choice can be made based on prior knowledge of the signal. When the point to be removed is on the boundary, rather than using the requested number of neighbours (coming from only one side), we use only its closest neighbour to prevent using artificial boundary neighbours. Also, after reiterating the algorithm several times, the requested number of neighbours might not be available, and in this case we decrease the number of used neighbours to the maximum available.

- To ensure a stable transform, non-degenerate regression curves are desired. Yet the higher the order of prediction we use, the more neighbours we have to request in order to get non-degenerate curves. For instance, in order to obtain a line with a slope, we need at least two points (neighbours), but for a parabola we will need at least three, while for a cubic at least four. At certain steps of the transform we will be in situations of having fewer neighbours than requested, so we decrease the order of the prediction curve as necessary. Hence the final transform will be a mixture of regression orders even when requiring a fixed order of prediction (linear, quadratic or cubic) all way through the lifting scheme, and so the final order of our MRA will not be exactly 2, 3 or 4, respectively.

- When using the lifting scheme with a fixed prediction strategy (linear, quadratic or cubic), the prediction and update weights will depend on the grid structure and on the type of prediction, hence the transform is a linear one. However, when using an adaptive approach, the prediction weights will be a function of the signal, since we build filters which depend on f. Hence the signal influences the prediction weights, and it also influences the updated integrals and the choice of the point to be removed next. As a consequence, the matrix associated to the transform, $\tilde{W}$, will be a function of the signal, f, which means that the adaptive transform is no longer linear. Thus, we cannot simply characterize the stability of our adaptive transform in terms of matrix condition numbers (see Section 5), although some form of expected condition number might be of some use.

3.3 Statistical shrinkage for adaptive lifting

We now examine how to denoise signals. We assume the following well-established model for our observation data $(f_i)_{i=1}^n$

$$f_i = g_i + \varepsilon_i, \tag{11}$$

for $i \in \{1, \dots, n\}$, where g_i is the population value to be estimated and ε_i is identically distributed, independent noise, assumed here to follow a $N(0, \sigma^2)$ distribution. This classical problem has been thoroughly addressed in the statistical literature. With wavelet shrinkage, denoising is achieved by taking the wavelet transform of (11). Theoretically the wavelet coefficients of g_i can be shown to form a sparse set, and it can be shown that diagonal coefficient shrinkage is an optimal strategy (see, e.g., Donoho and Johnstone, 1994). We demonstrate computationally (in Section 5) that our adaptive lifting produces sparse coefficient sets, so it makes sense to adopt recent wavelet shrinkage techniques for our coefficient processing. A review of wavelet shrinkage appears in Abramovich *et al.* (2000) and some more recent techniques are described and compared in Barber and Nason (2004). Our shrinkage is based on Johnstone and Silverman (2004a,b, 2005), which also contains further detailed references to related and earlier work in this large body of literature.

After transformation, model (11) is converted to

$$d_{j,k} = d^*_{j,k} + e_{j,k}, \tag{12}$$

where $d_{j,k}$ are the observed wavelet coefficients, $d^*_{j,k}$ are the true coefficients and $e_{j,k}$ is the DWT of the noise $\varepsilon_{n,i}$. For the classical DWT, $e_{j,k}$ is itself distributed as independent $N(0, \sigma^2)$, but for our lifting which lacks orthogonality, the noise will be correlated and different coefficients have different variances. This phenomenon is carefully considered below so that a suitable shrinkage algorithm can be devised.

3.3.1 The empirical Bayesian wavelet shrinkage approach

First let us review how empirical Bayes wavelet shrinkage works for the DWT and then we shall describe the modifications necessary for adaptive lifting.

For wide classes of functions we know that their DWT coefficients are sparsely populated (i.e. most wavelet coefficients are zero and a few are non-zero). Hence a good choice of prior for a DWT coefficient is:

$$d^*_{j,k} \sim (1-\pi)\delta_0 + \pi\gamma, \tag{13}$$

where π is the prior probability of a DWT coefficient being non-zero, and conditioned on it being non-zero it has density function given by γ. Recent work by Johnstone and Silverman (2004a, 2005) shows that a heavy-tailed choice of γ demonstrates excellent theoretical and practical advantages: here we use their "quasi-Cauchy" prior. In the DWT it is assumed that wavelet coefficients at the same scale, j, all have the same prior probability, denoted π_j, of being non-zero. From the signal plus noise model in (12) we know that the likelihood of $d_{j\cdot}|d^*_{j\cdot}$ is given by $d_{j\cdot} \sim N(d^*_{j\cdot}, \sigma^2)$ independently conditional on the $d^*_{j\cdot}$. The posterior distribution of $d^*_{j\cdot}$ given $d_{j\cdot}$ can be calculated from the prior and likelihood in the usual way. The hyperparameters are estimated as follows: π_j is estimated using a level-wise marginal-maximum likelihood (MML) and σ is estimated from the median absolute deviation (MAD) from zero of the finest observed details (as in Donoho and Johnstone, 1994). The "true" wavelet coefficients can then be estimated by the median of the posterior distribution (this operation acts as a true thresholding operation on the noisy wavelet coefficients $d^*_{j\cdot}$).

3.3.2 Modifications for adaptive lifting

Lack of discrete dyadic scales. In the DWT scale is a discrete dyadic quantity. In 'one coefficient at a time' lifting, scale becomes more of a continuous concept, and we define the scale of the detail coefficient associated to ψ_{j_r} to be the integral I_{r,j_r}. Then we mimic the Bayesian model above by introducing *artificial scale levels* through partitioning the detail coefficients according to their scale I_{r,j_r}. That is, we find the median, the upper quartile, the 87.5th quantile, etc. so that the "finest scale" half of the coefficients are put into the finest scale level, the next "finest" half goes into the next finest level, and so on and π_j for each artificial scale level is estimated as before by MML. This definition of (continuous) scale and artificial levels is borrowed from Jansen *et al.* (2001, 2004).

Correlation structure.

The structure of the adaptive lifting transform is dependent on the input function, f. Hence the following results are obtained by, and only valid for, conditioning on the local structure.

The first step of the lifting transform in (5) is: $d_{j_n} = c_{n,j_n} - \sum_{i \in I_n} a_i^n c_{n,i}$. Since the initial observations are assumed independent, we have

$$\operatorname{var}(d_{j_n}) = \sigma^2 \left\{ 1 + \sum_{i \in I_n} (a_i^n)^2 \right\}. \tag{14}$$

The update step in (7) gives, for all $i \in I_n, i \neq j_n$,

$$\operatorname{var}(c_{n-1,i}) = \operatorname{var}(c_{n,i}) + (b_i^n)^2 \operatorname{var}(d_{j_n}) + 2b_i^n \operatorname{cov}(c_{n,i}, d_{j_n}), \tag{15}$$

where $\operatorname{cov}(c_{n,i}, d_{j_n}) = -a_i^n \sigma^2$ from (5).

Also, for $i, j \in I_n, i \neq j, \ i, j \neq j_n$,

$$\operatorname{cov}(c_{n-1,i}, c_{n-1,j}) = (-a_i^n b_j^n - a_j^n b_i^n)\sigma^2 + b_i^n b_j^n \operatorname{var}(d_{j_n}) \tag{16}$$

and for $i \in I_n, j \notin I_n, \ i, j \neq j_n$ we have

$$\mathrm{cov}(c_{n-1,i}, c_{n-1,j}) = 0. \tag{17}$$

For any $i, j \notin I_n$, $\mathrm{cov}(c_{n-1,i}, c_{n-1,j}) = 0$.

The above argument shows that the update step induces correlations between the coarser coefficients as the algorithm proceeds, with the correlations propagating through the collection of coefficients. If $\tilde{W}$ is the matrix associated with the transform, then the resulting vector of coefficients can be written as $\tilde{W}f$. Hence $\mathrm{var}(\tilde{W}f) = \tilde{W}\,\mathrm{var}(f)\tilde{W}^t = \sigma^2\tilde{W}\tilde{W}^t$ under model (11).

Following Jansen *et al.* (2001, 2004) we ignore these correlations in our empirical Bayes procedure. On the other hand, given the normality assumptions in (11) the detail coefficients will be normally distributed.

From the above it is also clear that even though each f_i has variance σ^2, the resulting details will have different variances. To overcome this we apply the empirical Bayes method to the normalized detail coefficients $d_{j_r}\{\mathrm{diag}(\tilde{W}\tilde{W}^t)_r\}^{-1/2}$ which all have the same variance. The final thresholded coefficients are the medians of the posterior distributions of the normalized details, multiplied by $\{\mathrm{diag}(\tilde{W}\tilde{W}^t)_r\}^{1/2}$. The thresholded coefficients are then inverted to obtain the denoised signal.

We have also developed procedures to take into account situations when the initial signal observations are subject to heteroscedastic noise. Assuming the initial observations variances are known up to proportionality, we get $\mathrm{var}(f_i) = \sigma^2\gamma_i^2$, where γ_i is the known proportionality factor and σ^2 is unknown. After applying the lifting transform (in one of its linear variants), the variances of the detail coefficients are described by $\mathrm{var}(d_j) = \sigma^2\sum_{i\in\{1,\ldots,n\}}\gamma_i^2\tilde{W}_{j,i}^2$, where $\tilde{W}_{j,i}$ is the (j,i)th entry in $\tilde{W}$. To estimate σ we normalize the wavelet coefficients by dividing by $\sqrt{\sum_{i\in\{1,\ldots,n\}}\gamma_i^2\tilde{W}_{j,i}^2}$ and then use the MAD of the normalized details belonging to the first artificial level. After normalizing, thresholding and “un-normalizing”, we can invert the transform to form an estimate.

If the variance is heteroscedastic without any structural knowledge (like known up to a constant), then we estimate the variance σ_j associated with the detail coefficient d_j as suggested by Kovac and Silverman (2000). First identify any other detail coefficients within a window centered on the x_j in the data domain. Then we estimate σ_j by the MAD of the identified coefficients lying in the finest artificial level. Then we can threshold and invert the transform.

Multiple observations at a single gridpoint. For example, section 6.2 uses the famous motorcycle data described by Silverman (1985) which contains such multiple observations. There are several issues with multiple observations that arise in our lifting algorithm:

Problems with no modifications. If nothing in our algorithm is changed to take account of multiple observations then we obtain the situation where some points receive zero integrals (since the “distance” from one multiple point to another is zero). Hence, the points that get removed first are the multiple ones. This is clearly a degenerate situation and so multiple points are considered as having *one* x value, not many.

Removed point’s neighbours have multiple values. In this situation, the prediction regression curve will be estimated using all of the extra information contained in the repeated observations at neighbouring points. Since the regressions we use are simply polynomial ones this is easy to do.

In the update step, all multiple neighbours get updated in the same way by the function of the detail coefficient. So, if a neighbour point is multiple before the update step, it remains so afterwards.

Removed point is itself multiple. In this situation there are many possibilities. Each of the repeated observations can form individual distinct detail coefficients (just the difference between the datum and the prediction curve). To form just one detail coefficient we take the mean of the distinct individual detail coefficients (although other quantities such as the minimum might be an interesting alternative).

Multiple points after all lifting steps. If any of the scaling coefficients are multiple after all lifting steps, then their mean is used in the inverse transform.

4 Simulation techniques

In Sections 5 and 6, we compare our lifting algorithms to several other methods. In these sections we refer to these techniques using the following abbreviations:

- KS – the Kovac-Silverman method Kovac and Silverman (2000);
- *Locfit* – the local polynomial fitting method by Loader (1997, 1999);
- SSCV – the S-Plus function `smooth.spline()` with a cross-validatory choice of the smoothing parameter.

In our plots and tables we also abbreviate the names of our methods with a two-letter code. So LP=Linear Prediction, QP=Quadratic Prediction, CP=Cubic Prediction, AN=AdaptNeigh (above) and AP=AdaptPred (above). Then this code is followed by a number, N, which indicates the number of neighbours used in the prediction step. Finally, except for AN where both nearest and symmetric neighbours are considered, a single letter, N or S indicates that either the nearest neighbours were used or neighbours were obtained symmetrically either side of the point to predict (note that when the code is S then the actual number of neighbours is twice N).

5 Sparsity demonstration

A successful classical wavelet shrinkage algorithm relies on how efficient the transform is at sparsely representing functions. This section examines the sparsity of our adaptive lifting transforms and compares it to established methods. Our tests will be performed on the *Doppler*, *Bumps*, *Blocks* and *HeaviSine* functions devised by Donoho and Johnstone (1994), and on the piecewise polynomial (*Ppoly*) from Nason and Silverman (1994). We will investigate sparsity both on regular grids (to facilitate comparison with classical Daubechies wavelets) and irregular (jittered) grids with varying degrees of regularity.

Jittered grids. Start with a regular division of the interval [0,1] consisting of n points, where n is the number of observations of the function f. The irregularity will be generated by shifting each point around its location with a random value generated from a uniform distribution on the interval $(-d/(n-1), d/(n-1))$. We use d to denote the degree of jitter. Three possible values were used, $d_1 = 0.01, d_2 = 0.1$ and $d_3 = 1$. The reason for using jittered grids over, say, complete uniformly distributed points in the interval is that for very small jitter, d_1, we can compare results with classical wavelets on regular grids, and then we can see how our techniques perform as we progressively move towards more random locations. When the jitter value d_3 is chosen the data locations look uniform and have lost all semblance of regularity.

We perform simulations with data sets containing $n = 256$ points. For each jitter value we measure the average performance of our transforms over 50 generated sets.

Sparsity plot construction. We shall construct a diagnostic tool called a *sparsity plot*, as follows. We decompose the test signal on its irregular locations down to two scaling coefficients (all the rest are detail coefficients). We then arrange the detail coefficients in ascending order of their absolute value. Then, one by one, starting from the smallest, we replace each detail coefficient by zero, inverting the transform each time. The number of detail coefficients which remain at each step (the ones not transformed into zeroes) is plotted on the abscissa, and on the vertical axis we compute (at each step) the integrated squared error after performing the transform inverse, $ISE(i) = \sum_j (f_j - \hat{f}_j(i))^2$. Here i indexes the number of non-zero detail coefficients and $\hat{f}_j(i)$ denotes the reconstructed value of f_j based on these details and the scaling coefficients. So $i = 0$ means that we perform the reconstruction containing only the scaling coefficients. Then as i increases with each step we bring in one more detail, starting from the largest one, in decreasing order of their absolute size. The 50 ISE curves are then averaged to form the sparsity plot.

Stability. Unlike classical wavelets, adding an extra coefficient does not necessarily reduce the ISE. Our adaptive lifting transforms do not even satisfy the conditions required of a Riesz basis: the set $(\xi_k)_k$ of $L^2(\mathbb{R})$ is a Riesz basis if, $\forall f \in L^2(\mathbb{R})$, $\exists\ (c_k)_k$ such that $f(x) = \sum_k c_k \xi_k(x)$ and

$$m\|c\|_{l_2} \leq \|f\|_{L^2} \leq M\|c\|_{l_2},$$

where m, M are finite and depend only on the wavelet basis. The ratio $k = Mm^{-1}$ is called the *condition number* of the basis. For an orthonormal basis $k = 1$, whereas large values of k indicate potential problems.

We have numerically evaluated the condition number of our adaptive lifting transforms. Of course one needs to take care because the precise form of the transform depends both on the exact type of irregularity in the data set locations, and *also* on the characteristics of the test function. However, some general conclusions emerge: (i) when larger neighbourhoods in the prediction step occur the condition number is higher; (ii) higher orders of the prediction error curve also tend to give rise to high condition numbers.

Sparsity plot discussion. LP (fixed linear prediction) with 2 neighbours gives the best compression out of all the linear algorithms, and it is also associated with small condition numbers across the different data set locations. An interesting feature is that increasing irregularity did not seem to decrease sparsity or unduly increase the condition number. Over our simulations we found that AP2N, AP1S and AN1 have small condition numbers (all of them in the same range across the grids), indicating stability. All adaptive transforms give better compression than the non-adaptive ones, and increasing the adaptiveness increases the sparsity of the wavelet coefficients. However, because of potential stability issues, we would recommend using AN1, as it provides a good balance between sparsity and stability, followed by AP2N and AP1S. These general conclusions were valid for all our test signals although with *HeaviSine* there was not a lot of difference between AP2N, AN1 and LP2S. As the irregularity of our input data x_i increases (jitter increases) we have not noticed any major differences in sparsity, see, for example, Figure 1.

When faced with not having enough neighbours to completely determine the required curve (e.g. at the boundary), we tried another type of prediction step: the highest possible order curve forced through the origin. However, we found that there was a tendency for this type of prediction step to cause instability, though asymmetric neighbour selection did not adversely affect stability as long as there were enough neighbours.

Figure 2 shows the sparsity plot for the *Blocks* signal that compares two of our new methods (AP1S and AN1), with a similar construction for the KS algorithm and also sparsity plots constructed

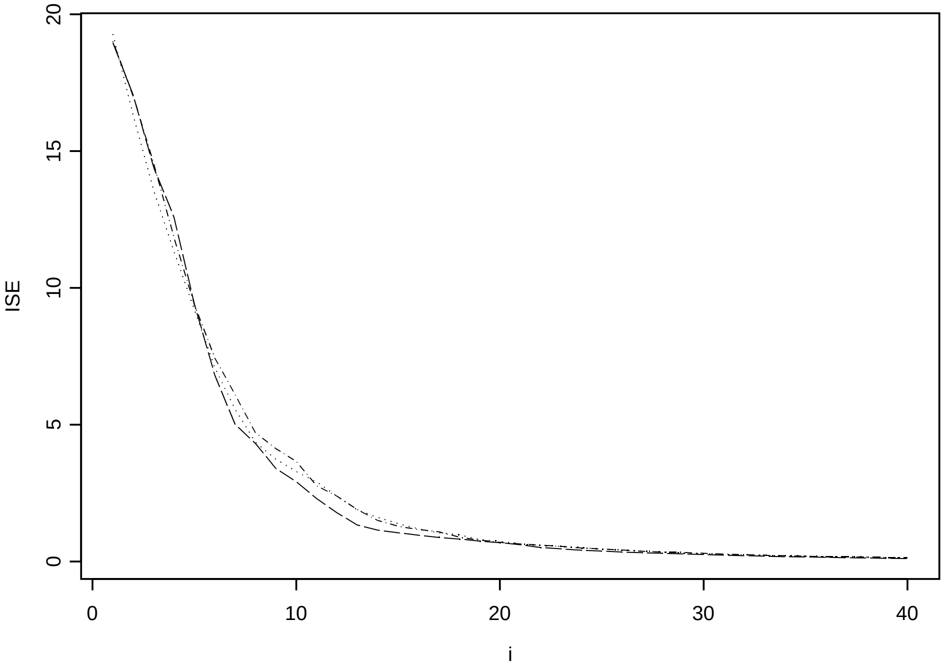


Figure 1: Blowup of sparsity plot for *Doppler* signal using AN1 over the three different jitter values: d_1 (dotted); d_2 (dot-dashed); d_3 (dashed).

for Daubechies' Extremal Phase wavelets (Haar and D7) on a regular grid. Figure 3 shows a blowup of Figure 2 in the region where few non-zero coefficients are inserted. Our AN1 method seems competitive with the regular wavelet methods (remember the curves for the irregular methods are average curves, not a single ISE curve which occur for the methods that work on a regular grid). The KS method suffers from the problem that it can only estimate an interpolated version of the true signal. If the interpolation of the 'true' signal is not a good approximation to the sampled signal on the KS equally spaced grid then the ISE of KS will not tend to zero. Figures 4 and 5 present the same plots as Figures 2 and 3 but for *Doppler*.

Figures 6 and 7 show which kind of basis function is chosen at each point that gets removed for the *Blocks* and *Bumps* signal. For the *Blocks* signal, an efficient linear fit is used for nearly every point. For the *Bumps* signal, the type of basis function is much more varied and more quadratic and cubic functions are used at certain points of the signal.

6 Real data examples and simulations

6.1 Inductance plethysmography data

Figure 8 shows the inductance plethysmography data introduced by Nason (1996). In this experiment the plethysmograph is arranged around the chest and abdomen of a set of patients and is used to measure the flow of air during breathing. The study in Nason (1996) commented on how well the wavelet methods (for the data on a regular grid) preserved the peaks whilst removing the noise. Here a similar phenomenon is observed.

All of the methods in Figure 8 do a good job as following the peaks and removing the noise.

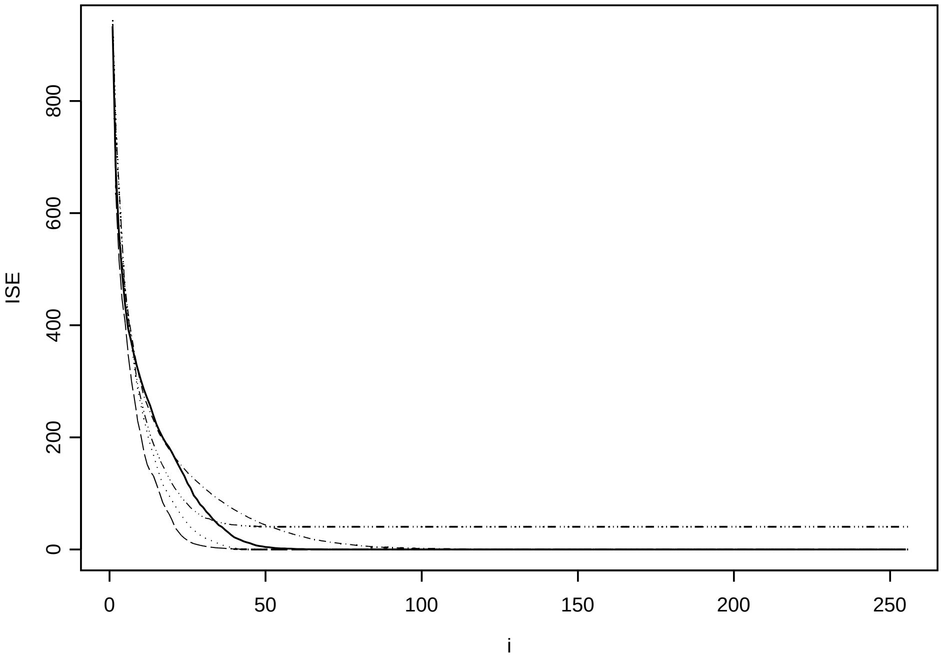


Figure 2: Sparsity plot for *Blocks* signal using different algorithms. Jitter was $d_3 = 1$ for irregular methods: AP1S (solid); AN1 (dashed); KS using Haar wavelets (3 dots dash). Regular grid for Daubechies' Extremal Phase wavelets: best sparsity was Haar wavelets (dots); worst sparsity was D7 wavelet (dot dash).

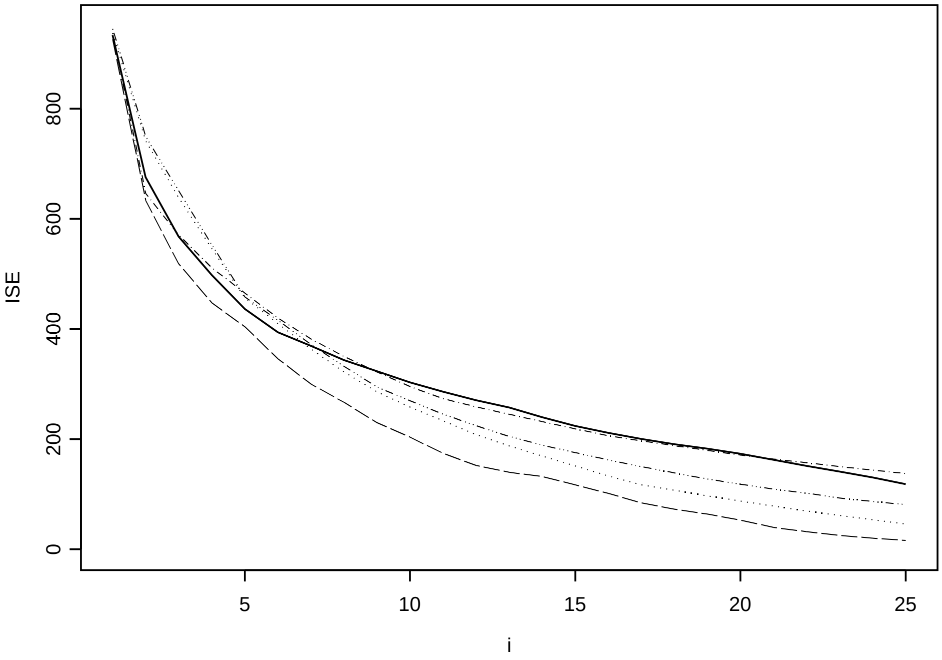


Figure 3: Blowup of Figure 2.

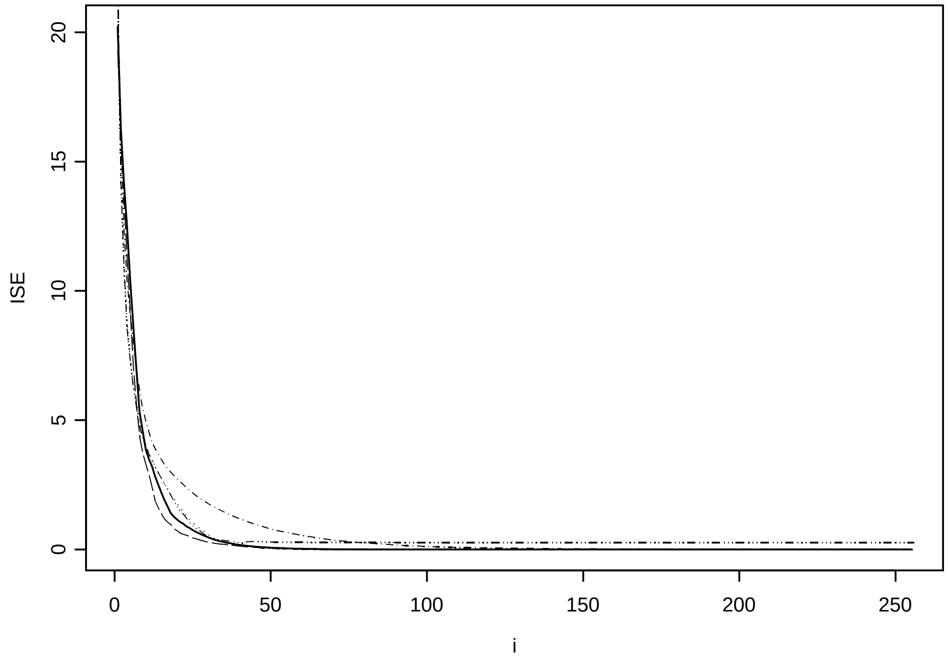


Figure 4: Sparsity plot for *Doppler* signal using different algorithms. Jitter was $d_3 = 1$ for irregular methods: AP1S (solid); AN1 (dashed); KS using D4 wavelets (3 dots dash). Regular grid for Daubechies' Extremal Phase wavelets: best sparsity was D4 wavelets (dots); worst sparsity was Haar wavelets (dot dash).

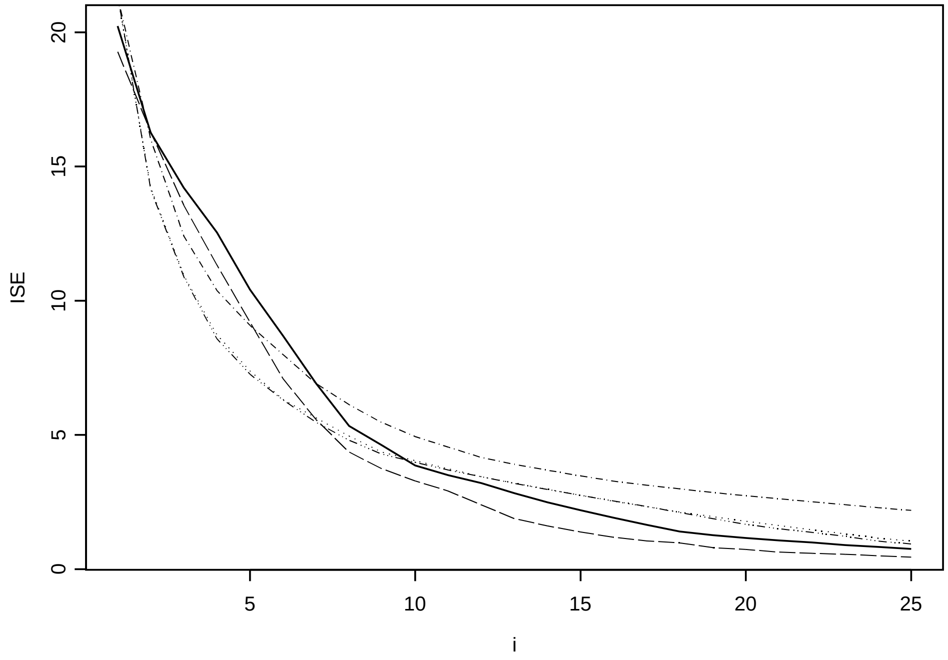


Figure 5: Blowup of Figure 4. Initially D4 wavelets and KS do well for up to 7 coefficients but then the adaptive methods do better.

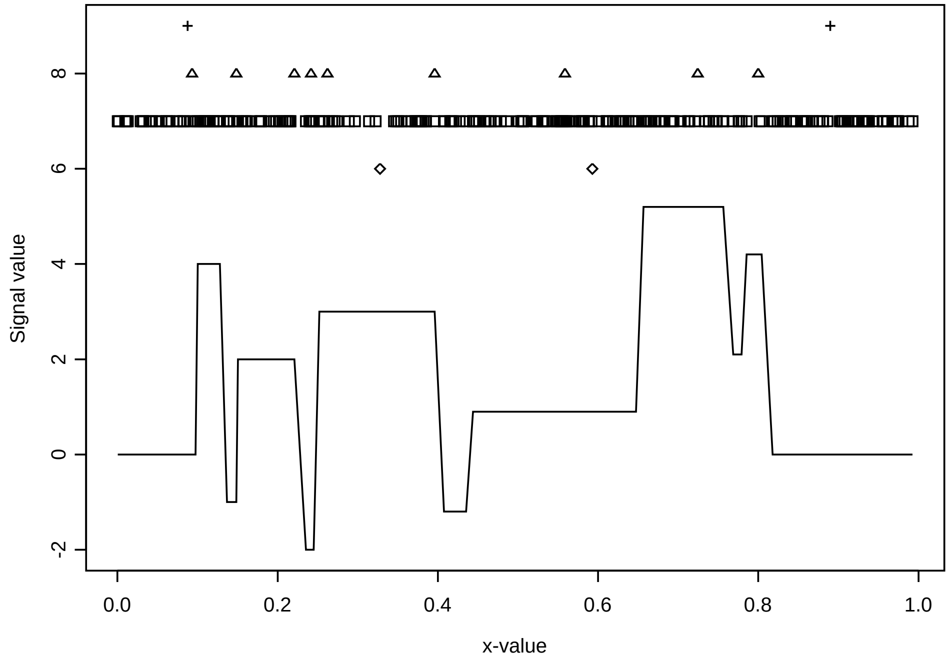


Figure 6: Plot showing choice of prediction scheme for the *Blocks* test signal decomposed with AN2 on an irregular grid (d=1). Horizontal placement of symbol indicates location of following kinds of prediction: linear (□); quadratic (△); cubic (+); scaling functions (◇).

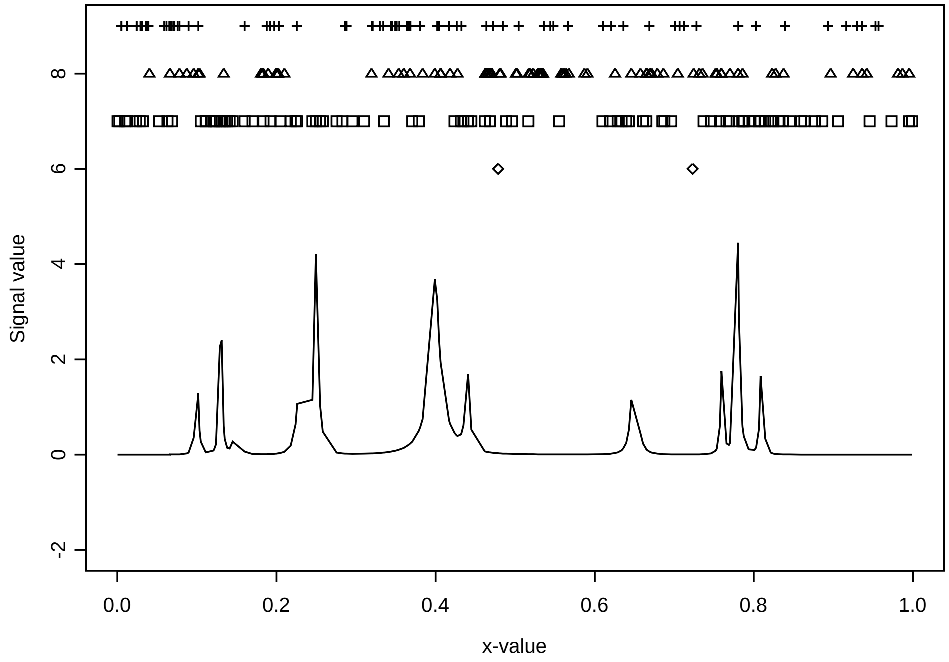
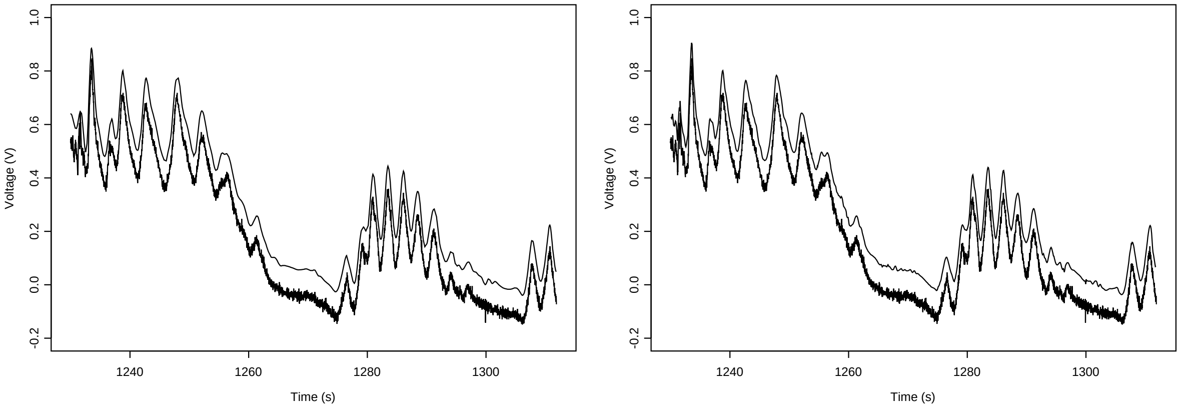


Figure 7: Plot showing choice of prediction scheme for the *Bumps* test signal decomposed with AN2 on an irregular grid (d=1). Horizontal placement of symbol indicates location of following kinds of prediction: linear (□); quadratic (△); cubic (+); scaling functions (◇).



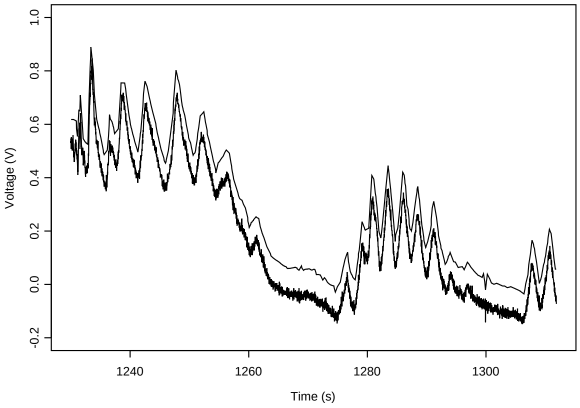


Figure 8: Estimators applied to 700 irregularly spaced samples from the inductance plethysmography data. Noisy data shown with estimate shifted up by 0.1 for visibility. Top left: smoothing spline with cross-validated smoothing parameter; Top right: KS estimate, D6 Daubechies' wavelet, primary resolution 3, using *SureShrink* thresholding; Right: adaptive lifting using AP1S and *EbayesThresh* posterior median thresholding.

However, we would suggest that our adaptive lifting gives sharper peaks than the smoothing spline and KS and it also removes noise a bit better than the KS method e.g. the "double peak" at around 1257 on the KS plot. Our method also seems to be able to "select" between rounded and sharp peaks (e.g. the first 6 peaks in our method are sharp, the 7th is more rounded).

6.2 Motorcycle data (example of multiple observations at a point)

Figure 9 shows the motorcycle data analyzed by Silverman (1985). The data contains 133 samples (at 94 time points) of head acceleration in simulated motorcycle crashes versus time in an experiment to determine the efficacy of crash helmets. No recipe is supplied in Kovac and Silverman (2000) for handling multiple observations at one time point, so we supplied KS with the mean of the multiple observations at those points.

The KS estimate in Figure 9 is very noisy. However, this particular KS estimate uses the basic algorithm, no allowance has been made for the changing variance. It should be noted that Kovac (1998) uses a further procedure to remove outliers and gets a much better looking estimate. Our adaptive lifting estimate seems quite similar to the smoothing spline one except for (a) ours looks less smooth, this is due to the basis functions being linear (b) our "main peak" after the dip occurs later than the one in the smoothing spline, but this "lateness" has also been observed in, e.g. some of the plots in Kovac (1998) (c) our estimate has a 'glitch' near the bottom of the dip which we think is unlikely to be a true feature.

We believe that the true motorcycle curve is actually most likely to be smooth and as such the smoothing spline is maybe the best estimate here. The wavelet methods (KS and ours) really come into their own on irregular sets (e.g. with sharp peaks, or jumps). However, it is pleasing that our

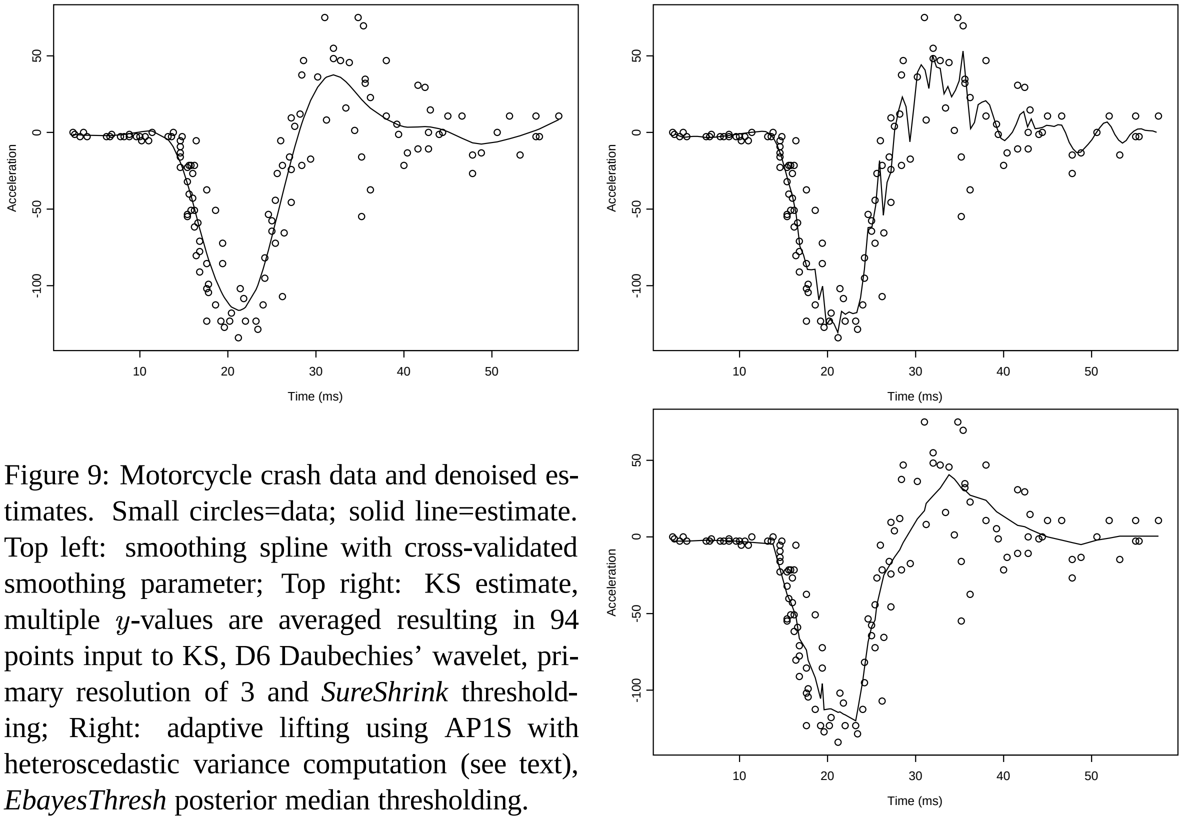


Figure 9: Motorcycle crash data and denoised estimates. Small circles=data; solid line=estimate. Top left: smoothing spline with cross-validated smoothing parameter; Top right: KS estimate, multiple y-values are averaged resulting in 94 points input to KS, D6 Daubechies' wavelet, primary resolution of 3 and *SureShrink* thresholding; Right: adaptive lifting using AP1S with heteroscedastic variance computation (see text), *EbayesThresh* posterior median thresholding.

lifting estimate in Figure 9 does a reasonable job in this case.

6.3 Further examples

Several further examples of our methodology outperforming existing wavelet methods can be found in Knight and Nason (2004). In the task of predicting transmembrane protein segments, improvements of up to 13% appear through the use of adaptively constructed wavelets over the classical Daubechies wavelets.

6.4 Simulation results

This section compares our adaptive lifting methodology with *Locfit* (Loader, 1999, 1997), the smoothing spline function in S-Plus, `smooth.spline()` and the wavelet algorithm for irregular data introduced by KS.

Tables 1, 2 and 3 show our simulation results for signal-to-noise ratios of 3, 5 and 7, where this ratio is given by $SNR = \sqrt{\mathrm{var}(g)}/\sigma$ using the notation of model (11). Each simulation is carried out on the jittered grid described in the previous section with three jitter values $d_1 = 0.01, d_2 = 0.1, d_3 = 1$. For each simulation, $k = 1, \ldots, K = 100$, we obtain an irregularly spaced set of $n = 256$ data points, evaluate the test function, g^k, at these points and add zero mean normal noise of an appropriate variance to achieve the correct SNR for f^k. We then denoise f^k to obtain our estimate $\hat{g}^k$ and compare this to the truth, g^k. A measure of the overall accuracy of the estimates is the *average mean square*

Table 1: AMSE ($\times 10^3$) simulation results for test signals with SNR=3 with three levels of jitter, d_ℓ, for various denoising methods described in the text.

	Blocks			*Bumps*			*HeaviSine*			*Doppler*			*Ppoly*		
Method	d_1	d_2	d_3	d_1	d_2	d_3	d_1	d_2	d_3	d_1	d_2	d_3	d_1	d_2	d_3
LP1S	72	71	68	81	80	73	20	20	21	54	53	52	16	16	18
LP2N	70	73	67	84	83	73	20	20	22	55	56	51	16	16	17
AP1S	72	68	59	77	77	62	20	20	23	52	50	48	16	17	18
AP2N	69	70	59	78	75	64	21	21	22	53	52	48	15	16	17
AP3N	69	68	68	76	74	73	46	44	41	64	65	61	42	39	36
AN1	55	54	52	66	67	61	36	39	37	61	61	59	38	33	32
Locfit	73	72	64	110	108	101	11	11	11	58	58	54	21	20	19
SSCV	74	74	67	307	315	250	12	11	12	61	60	53	20	20	19
KS	79	78	87	179	181	259	13	12	15	51	52	57	18	17	18

Table 2: AMSE ($\times 10^3$) simulation results for test signals with SNR=5 with three levels of jitter, d_ℓ, for various denoising methods described in the text.

	Blocks			*Bumps*			*HeaviSine*			*Doppler*			*Ppoly*		
Method	d_1	d_2	d_3	d_1	d_2	d_3	d_1	d_2	d_3	d_1	d_2	d_3	d_1	d_2	d_3
LP1S	24	25	22	31	28	27	10	10	10	23	23	23	6	6	7
LP2N	23	23	22	30	30	27	10	10	11	23	23	22	6	6	6
AP1S	22	23	20	30	29	23	10	10	10	22	22	21	6	6	7
AP2N	23	23	20	30	29	23	10	10	11	22	21	21	6	6	7
AP3N	27	27	26	30	30	29	18	18	16	26	26	26	16	15	14
AN1	19	20	18	26	26	24	15	16	16	25	24	24	13	13	12
Locfit	35	35	34	40	40	39	7	7	7	25	26	25	12	12	11
SSCV	51	51	46	277	285	227	7	7	7	37	37	30	11	12	11
KS	52	52	59	130	134	213	8	7	8	29	28	33	9	9	10

error defined by

$$\mathrm{AMSE} = (nK)^{-1} \sum_{k=1}^{K} \sum_{i=1}^{n} (g_i^k - \hat{g}_i^k)^2. \tag{18}$$

We tested several linear, quadratic, cubic and adaptive lifting methods as well as *Locfit* and `smooth.spline` with cross-validation (SSCV) and KS. We always chose to use EBayesThresh with posterior median shrinkage for our adaptive lifting methods. We also report the AMSE values obtained by using the KS algorithm (although these values are computed on the regular grid that KS provides estimates on, and so they are only approximately comparable to the ones in (18) which are computed at the data points).

With KS, as with all classical wavelet-based transforms, a decomposing wavelet has to be chosen. For a comprehensive study, we used *all combinations of* the Daubechies' Extremal Phase wavelet family with vanishing moments ranging from 1 to 10 (D1,. . . , D10), all primary resolution levels and thresholding techniques *SureShrink* and *EbayesThresh*. However, we only report the results corresponding to the combination (wavelet, primary level, threshold method) that yields the *best* estimate — *Blocks*: (D1, 2, *SureShrink*); *Bumps*: (D2, 0, *EbayesThresh*); *HeaviSine*: (D4, 4, *SureShrink*);

Table 3: AMSE ($\times 10^3$) simulation results for test signals with SNR=7 with three levels of jitter, d_ℓ, for various denoising methods described in the text.

	Blocks			*Bumps*			*HeaviSine*			*Doppler*			*Ppoly*		
Method	d_1	d_2	d_3	d_1	d_2	d_3	d_1	d_2	d_3	d_1	d_2	d_3	d_1	d_2	d_3
LP1S	11	11	11	15	15	14	6	6	6	12	12	13	3	3	3
LP2N	11	11	10	16	15	13	6	6	6	13	12	12	3	3	3
AP1S	10	10	10	15	14	12	6	6	6	12	12	11	3	3	4
AP2N	11	10	10	15	14	12	6	6	6	12	12	12	3	3	4
AP3N	14	14	14	16	16	16	10	10	10	14	14	14	8	8	7
AN1	10	10	9	14	14	13	9	9	8	14	14	13	7	7	6
Locfit	20	20	19	21	20	20	5	5	5	13	13	16	9	9	8
SSCV	44	44	39	269	273	220	5	5	5	30	30	23	8	8	8
KS	45	45	52	119	122	195	6	6	5	22	22	25	5	5	6

Doppler: (D4, 5, *SureShrink*); *Ppoly*: (D5, 4, *SureShrink*). Note that, in practice, the *best* combination *would not be known* and so, in general, KS's performance would be much worse than described here.

For our adaptive methods it is not necessary to choose a wavelet. However, there is a choice to be made, akin to the usual primary resolution, in how many points get removed in the lifting algorithm. Simulations in Nunes and Nason (2005) have shown that the exact choice of resolution level in the adaptive lifting algorithms is not crucial as long as the level is low. Hence with the adaptive algorithms, a full decomposition is made, and only the thresholding technique is subject to choice.

The standard quadratic and cubic methods and AdaptNeigh with neighbourhoods larger than 2 did not work so well so our tables do not include these results.

Overall, our adaptive lifting methods perform very well. Examination of the simulation results shows that, in particular, AN1 works extremely well on the *Blocks* and *Bumps* functions, outperforming the three competitors. AP methods with two neighbours are suitable on smoother signals such as the test functions *HeaviSine*, *Doppler* and *Ppoly*. On *HeaviSine* our method is outperformed by all competitors when SNR=3 and 5, while on SNR=7 the methods have similar denoising capability. On *Doppler* with SNR=3, the KS algorithm is the only competitor that approaches our results, whereas with SNR=5, 7 our method outperforms all the other procedures, with the closest one being *Locfit*. On *Ppoly*, our method is better than the competitors, with the closest being KS.

Modified *HeaviSine* comparison. As an extra comparison we compared our AN1 method to the results obtained in Delouille *et al.* (2004). In their study the true function was the *HeaviSine* function modified so that the jumps were 4 in size rather than 2, the $\{x_i\}_{i\in\overline{1,100}}$ are distributed as $N(0.5,(0.2)^2)$ and 500 simulations were performed. The other methods listed are ANTO/FAN which is Antoniadis and Fan (2001) and SUPSMO is the "super smoother" of Friedman (1984). Table 4 shows that our AN1 method appears to give a slight improvement over their method. However, from the simulations reported above remember that our method performed worst overall on *HeaviSine*. At this point one may wonder why our adaptive lifting methods work well compared to KS in Table 4 on *HeaviSine* but not well in Tables 1 to 2. The reason appears to be that the *HeaviSine* function in Delouille *et al.* (2004) is a modified version of the original *HeaviSine*. The modification makes the jumps twice as extreme, and although the SNR is roughly the same, the increased jumps will make the associated wavelet coefficients relatively much bigger. Hence, the relative signal around the important locations (the points of discontinuity) is much higher and it is as if we locally are in a higher SNR regime which is where our methods become more competitive for *HeaviSine*.

Table 4: Results of the Simulation Study, $n = 100$, SNR=4. AN1 result computed here, all other results as computed by Delouille *et al.* (2004). First row: square root of median MSE value; Second row: interval shows square root of 1st and 3rd quartiles of the MSE results over 500 simulations. All results $\times 10^3$.

	Delouille *et al.*				
AN1	With Update	No update	ANTO/FAN	KS	SUPSMO
588	610	792	819	775	706
$[517, 654]$	$[526, 675]$	$[661, 989]$	$[759, 875]$	$[688, 856]$	$[629, 807]$

7 Conclusions and further work

This article has introduced two nonparametric regression methods based on introducing adaptivity into 'one coefficient at a time' lifting. Simulation results show that these new methods perform extremely well when compared to other existing regression techniques. The technique can report what kind of basis was used in different areas of the signal, giving information on the local character of the signal. Real data applications show that our adaptive lifting transforms exhibit the benefits of classical wavelet algorithms, but can do so on irregularly spaced data. Our techniques are designed to handle the case where multiple observations exist at given x points. All of these factors point to adaptive lifting being a useful new tool in the regression toolkit.

There are several aspects that could be studied in further work. These include: considering how to shrink in the face of coefficient correlations, thinking about fast computation of multiple predict step competitors and alternative ways of handling multiple data points. A significant challenge would be to devise an appropriate and useful theoretical framework from which to further study lifting and adaptive lifting.

8 Appendix and Acknowledgements

Software that implements our adaptive lifting techniques is freely available at the CRAN R software archive as an R package. It can also be found at

`http://www.stats.bris.ac.uk/`
`~maman/computerstuff/Adlifthelp/Adlifthelp.html`

Nunes gratefully acknowledges support from an EPSRC DTA award; Knight gratefully acknowledges support of an ORS award and Bristol University Research Scholarship. Nason was partially supported by EPSRC Advanced Research Fellowship AF/001664. Nason would like to thank again Andrew Black and David Moshal of the Department of Anaesthesia, University of Bristol for supplying and explaining the inductance plethysmography data.

References

Abramovich, F., Bailey, T. and Sapatinas, T. (2000) Wavelet analysis and its statistical applications. *J. Roy. Statist. Soc.* D, **49**, 1–29.

Antoniadis, A. and Fan, J. (2001) Regularization of wavelet approximations. *J. Am. Statist. Ass.*, **96**, 939–967.

Barber, S. and Nason, G.P. (2004) Real nonparametric regression using complex wavelets. *J. Roy. Statist. Soc.* Series B, **66**, 927–939.

Boulgouris, N.V., Tzovaras, D. and Strintzis, M.G. (2001) Lossless image compression based on optimal prediction, adaptive lifting and conditional arithmetic coding. *IEEE Trans. Im. Proc.*, **10**, 1–14.

Cai, T. and Brown, L. (1998) Wavelet shrinkage for non-equispaced samples. *Ann. Stat.*, **26**, 1783–1799.

Cai, T. and Brown, L. (1999) Wavelet estimation for samples with random uniform design. *Stat. Prob. Lett.*, **42**, 313–321.

Claypoole, R.L., Baraniuk, R.G. and Nowak, R.D. (1998) Adaptive wavelet transforms via lifting. In Transactions of the International Conference on Acoustics, Speech and Signal Processing. *IEEE Trans. Im. Proc.*, **12**, 1513–1516.

Claypoole, R.L., Davis, G.M., Sweldens, W. and Baraniuk, R.G. (2003) Nonlinear wavelet transforms for image coding via lifting. *IEEE Trans. Im. Proc.*, **12**, 1449–1459.

Daubechies, I. (1992) *Ten Lectures on Wavelets*. Philadelphia: SIAM.

Delouille, V., Franke, J. and von Sachs, R. (2004) Nonparametric stochastic regression with design-adapted wavelets. *Sankhya*, A, **63**, 328–366.

Delouille, V., Simoens, J. and von Sachs, R. (2004) Smooth design-adapted wavelets for nonparametric stochastic regression. *J. Am. Statist. Soc.*, **99**, 643–658.

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

Donoho, D.L. and Johnstone, I.M. (1995a) Adapting to unknown smoothness via wavelet shrinkage. *J. Am. Statist. Soc.*, **90**, 1200–1224.

Donoho, D.L., Johnstone, I.M., Kerkyacharian, G. and Picard, D. (1995b) Wavelet shrinkage: asymptopia? (with discussion) *J. Roy. Statist. Soc.* B, **57**, 301–337.

Friedman, J.H. (1984) A variable span scatterplot smoother. *Technical Report,* No. 5, Laboratory for Computational Statistics, Stanford University, Stanford, CA, USA.

Green, P.J. and Silverman, B.W. (1994) *Nonparametric regression and generalized linear models.* Chapman and Hall: London.

Jansen, M., Nason, G.P. and Silverman, B.W. (2001) Scattered data smoothing by empirical Bayesian shrinkage of second generation wavelet coefficients. In Unser, M. and Aldroubi, A. (eds) *Wavelet applications in signal and image processing,* Proceedings of SPIE, **4478**, 87–97.

Jansen, M., Nason, G.P. and Silverman, B.W. (2004) Multivariate nonparametric regression using lifting. *Technical Report* 04:17, Statistics Group, Department of Mathematics, University of Bristol, UK.

Johnstone, I.M. and Silverman, B.W. (2004a) Needles and hay in haystacks: Empirical Bayes estimates of possibly sparse sequences. *Ann. Statist.*, **32**, 1594–1649.

Johnstone, I.M. and Silverman, B.W. (2004b) EBayesThresh: R and S-PLUS programs for Empirical Bayes thresholding. Unpublished manuscript (available from the CRAN archive).

Johnstone, I.M. and Silverman, B.W. (2005) Empirical Bayes selection of wavelet thresholds. *Ann. Statist.*, **33**, (to appear).

Knight, M.I. and Nason, G.P. (2004) Improving prediction of hydrophobic segments along a transmembrane protein sequence using adaptive multiscale lifting. *Technical Report* 04:19, Statistics Group, Department of Mathematics, University of Bristol, UK.

Kovac, A. (1998) Wavelet Thresholding for Unequally Time-Spaced Data. *PhD Thesis*, University of Bristol.

Kovac, A. and Silverman, B.W. (2000) Extending the scope of wavelet regression methods by coefficient-dependent thresholding. *J. Am. Statist. Ass.*, **95**, 172–183.

Loader, C. (1997) Locfit: an introduction. *Stat. Comput. Graph. News.*, **8**, 11-17.

Loader, C. (1999) *Local regression and likelihood.* Springer: New York.

Mallat, S.G. (1989) A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattn. Anal. Mach. Intell.*, **11**, 674–693.

Nason, G.P. (1996) Wavelet shrinkage using cross-validation. *J. Roy. Statist. Soc.* B, **58**, 463–479.

Nason, G.P. (2002) Choice of wavelet smoothness, primary resolution and threshold in wavelet shrinkage. *Stat. Comput.*, **12**, 219–227.

Nason, G.P. and Silverman, B.W. (1994)The discrete wavelet transform in S. *J. Comp. Graph. Statist.*, **3**, 163–191.

Nunes, M.A. and Nason, G.P. (2005) Stopping time in adaptive lifting. *Technical Report* 05:15, Statistics Group, Department of Mathematics, University of Bristol, UK.

Pensky, M. and Vidakovic, B. (2001) On non-equally spaced wavelet regression. *Ann. Inst. Statist. Math.*, **53**, 681–690.

Percival, D.B. and Walden, A.T. (2000) *Wavelet methods for time series analysis.* Cambridge University Press: Cambridge.

Piella, G. and Heijmans, H.J.A.M. (2002) Adaptive lifting schemes with perfect reconstruction. *IEEE Trans. Sig. Proc.*, **50**, 1620–1630.

Sardy, S., Percival, D.B., Bruce, A.G., Gao, H.-Y. and Stuetzle, W. (1999) Wavelet de-noising for unequally spaced data. *Stat. Comput.*, **9**, 65–75.

Silverman, B.W. (1985) Some aspects of the spline smoothing approach to nonparametric curve fitting. *J. Roy. Statist. Soc.* B, **47**, 1–52.

Simoens, J. and Vandewalle, S. (2003) A stabilized construction of wavelets on irregular meshes on the interval. *SIAM J. Scientific Computing*, **24**, 1356–1378.

Sweldens, W. (1996) Wavelets and the lifting scheme: A 5 minute tour. *Z. Angew. Math. Mech.*, **76**, 41–44.

Sweldens, W. (1997) The lifting scheme: a construction of second generation wavelets. *SIAM J. Math. Anal.*, **29**, 511–546.

Trappe, W. and Liu, K.J.R. (2000) Denoising via adaptive lifting schemes. In *Proceedings of SPIE, Wavelet applications in signal and image processing VIII*, Aldroubi, A., Laine, M.A. and Unser, M.A. (eds), **4119**, 302–312.

Vanraes, E., Jansen, M. and Bultheel, A. (2002) Stabilised wavelet transforms for non-equispaced data smoothing. *Sig. Proc.*, **82**, 1979–1990.

Vidakovic, B. (1999) *Statistical modeling by wavelets.* Wiley: New York.