Wavelet shrinkage using cross-validation

By G. P. NASON†

University of Bristol, Bristol, UK

summary

Wavelets are orthonormal basis functions with special properties that show potential in many areas of mathematics and statistics. This article concentrates on the estimation of functions and images from noisy data using wavelet shrinkage. A modified form of twofold cross-validation is introduced to choose a threshold for wavelet shrinkage estimators operating on data sets of length a power of two. The cross-validation algorithm is then extended to data sets of any length and to multi-dimensional data sets. The algorithms are compared to established threshold choosers using simulation. An application to a real data set arising from anaesthesia is presented.

Keywords: adaptive estimation; nonparametric regression; spatial adaptation; smoothing parameter; threshold; anaesthetics

Journal of the Royal Statistical Society, Series B. (1996), **58**, 463–479.

1 Introduction

Recent work by Donoho *et al.* (1995a) has introduced the method of wavelet shrinkage for general curve estimation problems. There are several good reasons why wavelet shrinkage can be used for function estimation. The main reasons are that wavelet shrinkage estimators are: nearly minimax for a wide range of loss functions and for general function classes; simple, practical and fast; adaptable to spatial and frequency inhomogeneities; readily extendable to high dimensions; applicable to various other problems such as density estimation and inverse problems. A review of these reasons and justification for them appears in Donoho *et al.* (1995a).

This paper introduces two cross-validation methods for choosing the threshold parameter in wavelet shrinkage. Wavelet shrinkage is briefly reviewed in Section 2. Section 3 introduces the cross-validation algorithms and Section 4 illustrates the algorithms in one- and two-dimensions using simulations and by application to some real data collected on breathing patterns.

For further information on wavelets see Strang (1993), who provides an accessible introduction, and Nason and Silverman (1994) who discuss wavelets in a statistical context. Meyer (1992) and Daubechies (1992) both give detailed expositions of the mathematical aspects of wavelets.

2 Wavelet Function Estimation

2.1 Wavelets

Wavelet estimators may be used as a special kind of orthogonal series estimator because wavelets can form orthonormal bases for various function spaces. For example, a function $f \in L^2(\mathbb{R})$ may

†*Address for correspondence:* Department of Mathematics, University of Bristol, University Walk, Bristol, BS8 1TW, UK.
E-mail: G.P.Nason@bristol.ac.uk

be represented in terms of one of Daubechies' (1988) families of orthonormal wavelets $\{\psi_{jk}(x)\}$ by

$$f(x) = \sum_{j\in\mathbb{Z}}\sum_{k\in\mathbb{Z}} d_{jk}\psi_{jk}(x), \tag{1}$$

where

$$d_{jk} = \int_{\mathbb{R}} f(x)\psi_{jk}(x)\,dx$$

are the *wavelet coefficients* of f. Note that wavelets are doubly-subscripted. Roughly speaking the j subscript localizes analysis of f in frequency and the k subscript localizes analysis in time. This simultaneous time-frequency localization of information in f is the key to understanding why wavelets are attractive for function approximation and estimation.

The basis function wavelets are usually of the form

$$\psi_{jk}(x) = 2^{\frac{j}{2}}\psi(2^j x - k),$$

where j and k are integers and $\psi(x)$ is a function, called a *wavelet of class m*, specially constructed so that: $\{\psi_{jk}(x)\}$ forms an orthonormal basis for the function space under consideration; $\psi(x)$ and all its derivatives up to order m exist and decrease rapidly as $x \to \pm\infty$ and $\psi(x)$ is orthogonal to all polynomials of degree $(m-1)$. These properties and the wavelet series given in (1) all relate to a continuous domain. There is also a discrete version of the wavelet transform which is described next in the context of function estimation.

2.2 Wavelet shrinkage

Given data $g_1, \ldots g_n$ assume the model

$$g_i = f(t_i) + \epsilon_i, \tag{2}$$

where the $\{\epsilon_i\}$ is some noise process with variance σ^2, $t_i = i/n$ and f is the function to be estimated. The discrete wavelet transform can be represented by an orthogonal matrix W. Then

$$w = Wg \tag{3}$$

performs the wavelet transform on the noisy data. The wavelet coefficients are then modified by some procedure to form an array of coefficients $\hat{w}$ and then the inverse transform, W^T, is applied to obtain:

$$\hat{f} = W^T\hat{w}, \tag{4}$$

where $\hat{f}$ is the estimate of f at the points $\{t_i\}$. In practice, a fast algorithm developed by Mallat (1989) is used to perform the transform in $O(n)$ operations and matrix multiplication is not used. However, use of the fast algorithm limits n to be a power of 2. In practice one is unlikely to receive a real data set with $n = 2^M$ points and there are various ways around this limitation:

1. truncate or extend the series in some way and pretend that you have 2^M points;
2. devise some method of using wavelet shrinkage estimators for any number of points.

This article uses Daubechies' (1988) wavelets with periodic boundary correction. Further details of this transform can be found in Nason and Silverman (1994).

The key question for wavelet shrinkage is how should the wavelet coefficients, w, be modified to form $\hat{w}$? Donoho *et al.* (1995a) advise that thresholding wavelet coefficients produces estimates

that possess the desirable properties in the list at the beginning of the introduction. Given a wavelet coefficient w and a threshold $t > 0$ the *hard-thresholded* value is given by

$$T_{\text{hard}}(w;t) = w\, I(|w| > t),$$

and the *soft-thresholded* value by

$$T_{\text{soft}}(w;t) = \text{sgn}(w)\,(|w| - t)\, I(|w| > t),$$

where I is the usual indicator function. This article considers soft thresholding although in many situations hard thresholding is a suitable alternative. The question of how coefficients should be modified then reduces to the numerical choice of the threshold t. The choice is critical: if the threshold is too small/large then wavelet shrinkage estimators will tend to over/underfit the data. Donoho and Johnstone (1994) proposed various policies for choosing a threshold value. One policy used thresholds precomputed to minimize a constant term in the upper bound for the minimax risk of estimating a function using a shrinkage estimator. Donoho and Johnstone (1994) also propose the *universal* threshold that is incorporated into their *VisuShrink* procedure. The universal threshold is

$$T_{\text{UV}} = \sqrt{(2\log n)}\hat{\sigma}, \tag{5}$$

where n is the number of data points and $\hat{\sigma}$ is an estimate of the noise level σ. An important feature of *VisuShrink* is that it "guarantees" a noise-free reconstruction although by doing so it usually underfits the data (see also Fan *et al.* (1993)). Another threshold chooser based on Stein's (1981) unbiased risk estimation was proposed by Donoho and Johnstone (1995) and called *SureShrink*. The *SureShrink* chooser specifies a threshold value t_j for each resolution level j in a wavelet transform. This article introduces another threshold chooser based on cross-validation. At the present time the cross-validation algorithm chooses one threshold applicable to all resolution levels in the wavelet transform, although it could be modified to select a threshold for each level. For comparison purposes this article uses a modified version of *SureShrink* called *GlobalSure* that fixes one threshold by using Stein estimation on all applicable coefficients.

We stress that the goal of *VisuShrink* is not the minimization of mean squared error. Instead *VisuShrink* thresholds may be viewed as general purpose threshold selectors that exhibit "near-optimal" minimax error properties and ensure, with high probability, that the estimates are as smooth as the true underlying functions. Thus allowing increased bias to reduce variance is a design goal of *VisuShrink*. This contrasts with *Sure* and cross-validation methods that have the single goal of minimizing mean squared error. This difference should be remembered when comparing the simulation results of the two methods that appear later in this article.

Given a particular thresholding method T the *wavelet shrinkage estimator* at threshold t, $\hat{f}_t(x)$, is given by (4) with

$$\hat{w}_{jk} = T(w_{jk};t),$$

where w_{jk} are the wavelet coefficients of the data as in (3).

3 Cross-validation for wavelet regression

The aim of function estimation in this article is the minimization of the mean integrated square error (MISE) between the wavelet shrinkage estimator $\hat{f}_t(x)$ and the true function $f(x)$. In symbols the threshold t should minimize

$$M(t) = E\int \left\{\hat{f}_t(x) - f(x)\right\}^2 dx. \tag{6}$$

In practice the function f is not known and so an estimate of M has to be devised. It is often desirable that a loss function other than MISE be used and this can be easily achieved by replacing MISE by the appropriate loss in the estimate of M.

Cross-validation is widely used as an automatic procedure to choose a smoothing parameter in many statistical settings (for reviews in the context of nonparametric regression see Green and Silverman (1994); for density estimation see Silverman (1986)).

The classic cross-validation method is performed by systematically expelling a data point from the construction of an estimate, predicting what the removed value would have been and comparing the prediction to the value of the expelled point. This simple leave-one-out procedure cannot be directly applied to wavelet shrinkage estimation because the discrete wavelet transform using Mallat's fast algorithm only operates on data sets of size a power of 2.

Section 3.1 describes a "leave-half-out" cross-validation method which can make use of Mallat's algorithm directly. Section 3.2 describes a method for extending a data set of any size to one containing 2^M points.

3.1 Two-fold cross-validation

This section describes a cross-validation procedure that can be used to automatically select a threshold for a wavelet shrinkage estimator based on 2^M points.

The procedure works by leaving out half of the data points. This leaves 2^{M-1} data points that are then used to form a wavelet shrinkage estimator using a particular threshold. The values of the expelled points can then be compared with the shrinkage estimator to form an estimate of prediction error at a particular threshold. This quantity can be then numerically minimized over values of the threshold.

Two-fold cross-validation algorithm

Given data $g_1, \ldots, g_n$ where $n = 2^M$, remove all the odd-indexed g_i from the set. This leaves 2^{M-1} evenly indexed g_i which are reindexed from $j = 1, \ldots, 2^{M-1}$. A function estimate $\hat{f}_t^{\mathrm{E}}$ is then constructed using a particular threshold t from the re-indexed g_j. To compare the function estimator with the left-out noisy data an interpolated version of $\hat{f}_t^{\mathrm{E}}$ is formed:

$$\bar{f}_{t,j}^{\mathrm{E}} = \frac{1}{2}\left(\hat{f}_{t,j+1}^{\mathrm{E}} + \hat{f}_{t,j}^{\mathrm{E}}\right), \quad j = 1, \ldots, n/2, \tag{7}$$

setting $\hat{f}_{t,n/2+1}^{\mathrm{E}} = \hat{f}_{t,1}^{\mathrm{E}}$ because f is assumed to be periodic. The $\hat{f}_t^{\mathrm{O}}$ is computed for the odd indexed points and the interpolant $\bar{f}_t^{\mathrm{O}}$ computed as above. The full estimate for $M(t)$ compares the interpolated wavelet estimators and the left-out points:

$$\hat{M}(t) = \sum_{j=1}^{\frac{n}{2}} \left\{ \left(\bar{f}_{t,j}^{\mathrm{E}} - g_{2j+1}\right)^2 + \left(\bar{f}_{t,j}^{\mathrm{O}} - g_{2j}\right)^2 \right\}. \tag{8}$$

Note that the estimate $\hat{M}$ relies on two estimates of f_t based upon $n/2$ data points. We can use Donoho and Johnstone's (1994) universal threshold (5) to supply a heuristic method for obtaining a cross-validated threshold for n data points. If the threshold for n points is $T_{\mathrm{UV}}(n)$ then the threshold for $n/2$ points will be $T_{\mathrm{UV}}(n/2)$ and therefore

$$T_{\mathrm{UV}}(n) \approx \left(1 - \frac{\log 2}{\log n}\right)^{-1/2} T_{\mathrm{UV}}(n/2). \tag{9}$$

After the estimate $\hat{M}(t)$ has been minimized the correction (9) is applied to obtain the final cross-validated threshold.

This correction can be extended to a 2^k-fold cross-validation procedure, where 2^k estimates $\hat{f}$ are obtained, each based on $n/2^k$ data points selected in a regular way from the original data. Each estimate is interpolated to the original grid and validated by comparing to the remaining $n(1-2^{-k})$ data points. The correction term (9) is easily extended to this case and will give a correction factor of

$$\left(1-\frac{k\log 2}{\log n}\right)^{-1/2}.$$

This extension is not considered for the one-dimensional case in this paper, but we will need to use it when we consider the multivariate case in Section 3.3 below.

Our terminology is not the same as that used, for example, by Burman (1989), who uses ν-fold cross-validation – a procedure where each training set is of size $n(\nu-1)/\nu$ and each test set is of size n/ν. Another difference is that Burman considers regression problems where the design points t_i are random (and identically distributed) and the cross-validation test sets are randomly selected. In Stone's (1974) terminology our method is a case of *uncontrollable* cross-validation.

3.2 Leave-one-out cross-validation

This section develops a leave-one-out cross-validation method that works for *any* number of data points removing the previous algorithm's restriction of 2^M points.

Leave-one-out cross-validation algorithm

Given the data set $G=\{g_1,\ldots,g_n\}$ where $n>1$ choose i such that $1<i<n$. Remove g_i from G and split the remaining points into two groups:

$$G_{\mathrm{L}}=\{g_1,\ldots,g_{i-1}\}$$
$$G_{\mathrm{R}}=\{g_{i+1},\ldots,g_n\}.$$

Form G_{LRE} and G_{RRE} by reflection at the left and right ends of G_{L} and G_{R} respectively and then extend each set to the next largest power of two by filling with g_{i-1} for G_{LRE} and g_{i+1} for G_{RRE} to obtain

$$G_{\mathrm{LRE}}=\{g_{i-1},\ldots,g_{i-1},g_{i-2},\ldots,g_2,g_1,g_1,g_2,\ldots,g_{i-2},g_{i-1}\}$$
$$G_{\mathrm{RRE}}=\{g_{i+1},g_{i+2},\ldots,g_{n-1},g_n,g_n,g_{n-1},\ldots,g_{i+2},g_{i+1},g_{i+1},\ldots,g_{i+1}\}.$$

The sets G_{LRE} and G_{RRE} are illustrated in Figure 1. Denote the number of points in G_{LRE} by n_{L} and in G_{RRE} by n_{R}. Then n_{L} is the smallest power of two greater than or equal to $2(i-1)$ and n_{R} is the smallest power of two greater than or equal to $2(n-i)$. Now form two wavelet shrinkage estimators $\hat{f}_{\mathrm{L},t}$ and $\hat{f}_{\mathrm{R},t}$ using G_{LRE} and G_{RRE} and threshold value t. The removed point g_i is predicted by

$$\hat{g}_{t,-i}=\tfrac{1}{2}\left(\hat{f}_{\mathrm{L},t,n_{\mathrm{L}}}+\hat{f}_{\mathrm{R},t,1}\right),$$

where $\hat{f}_{\mathrm{L},t,n_{\mathrm{L}}}$ is the rightmost point of $\hat{f}_{\mathrm{L},t}$ and $\hat{f}_{\mathrm{R},t,1}$ the leftmost point of $\hat{f}_{\mathrm{R},t}$. The cross-validation score is given by

$$\hat{M}(t)=\sum_{i=2}^{n-1}\left(g_i-\hat{g}_{t,-i}\right)^2.$$

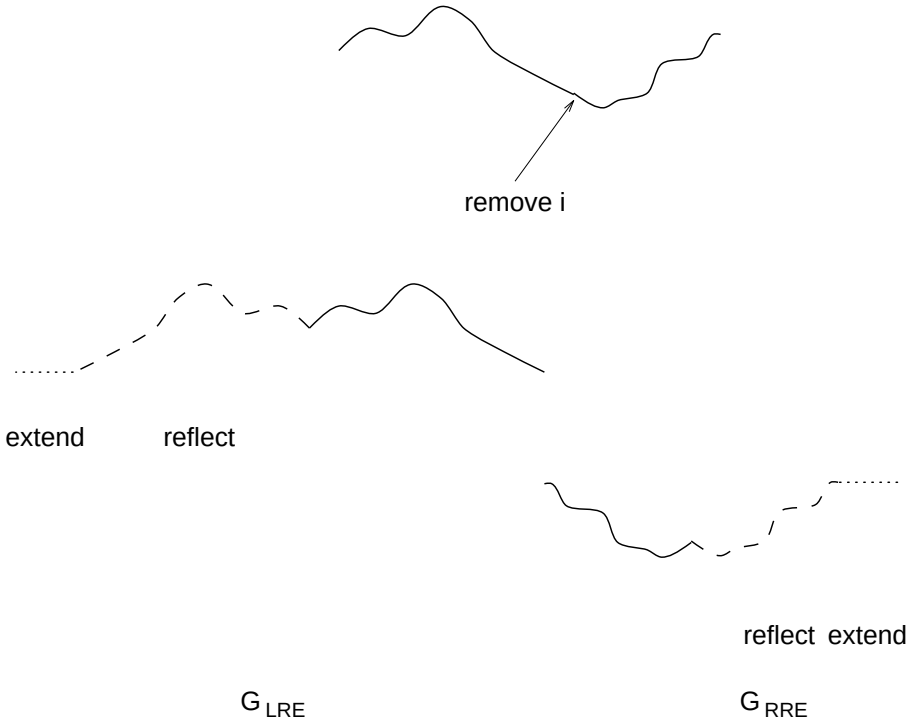


Figure 1: Reflection and extension of left and right series generated by removal of a point to form sets G_{LRE} and G_{RRE}. The reflected parts are indicated by dashed lines and the extended parts by dotted lines.

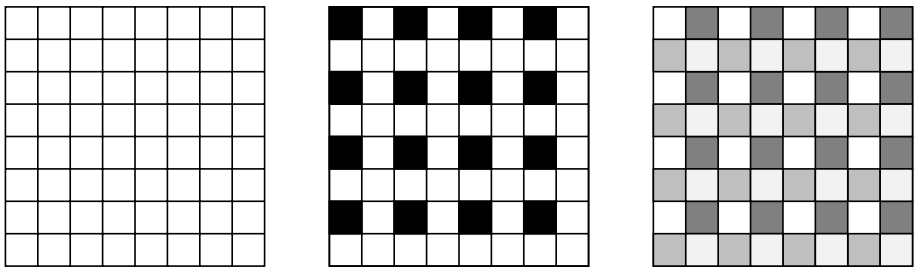
3.3 Cross-validation in more dimensions

The extension of the two-fold cross-validation of Section 3.1 to k dimensions is achieved by using the multidimensional DWT of Mallat (1989). As in Section 3 the cross-validation algorithm will minimize an estimate of the k-dimensional MISE (the x in equation (6) is now a vector in k-dimensional space). The next section develops an estimate of the k-dimensional MISE.

2^k-fold cross-validation algorithm

Assume now that the k-dimensional data may be denoted by $g_{i_1,\ldots,i_k}$ with $i_j \in \{1,\ldots,2^M\}$ for $j = 1,\ldots,k$. Suppose the data are arranged on a fixed equally spaced k-dimensional hypergrid H. For each of the k subscripts of g it is possible to select either the odd or evenly subscripted observations. Denote the selection of an even subscript by 0 and an odd subscript by 1. Then a 0/1 selection for each subscript provides a subset of H that is 2^{-k} times the size of H and equally spaced on a subgrid of H. We will denote a particular subgrid by $g_{[b]}$ where b is the binary number formed by concatenating the 0/1 selections in dimension order. For example, the selection $g_{[101]}$ would select all the odd-indexed observations on subscripts 1 and 3 and the even-indexed observations on subscript 2. Let the subgrid defined by $g_{[i]}$ be denoted H_i.

Denote the k-dimensional wavelet shrinkage estimator with threshold t based on data $g_{[i]}$ by $\hat{f}_{[i]}(t)$. Denote the quantity $\bar{f}^j_{[i]}(t)$ to be the interpolant of $\hat{f}_{[i]}(t)$ to the grid defined by $g_{[j]}$ by multiple repeats of the univariate interpolation scheme (7). This interpolation scheme is invariant with respect to the order in which each univariate interpolant is applied. Then the k-dimensional



a. b. c.

Figure 2: Organization for 2^2-fold cross-validation. a. The 8×8 pixel grid H. b. The data $g_{[11]}$ forming the first subgrid H_3. c. The other three subgrids ■= H_2, □=H_1 and □=H_0 containing data $g_{[10]}$, $g_{[01]}$ and $g_{[00]}$ respectively.

cross-validation score is given by:

$$\hat{M}(t) = \sum_{i=0}^{2^k-1} \sum_{j=0, j\neq i}^{2^k-1} \sum \left\{ \bar{f}^j_{[i]}(m;t) - g_{[j]}(m) \right\}^2,$$

where the final sum is over all indices m in the subgrid H_j.

3.3.1 Cross-validation for images

Images are two-dimensional objects and therefore 2^2-fold cross-validation can be used. This section illustrates the above algorithm using a 8×8 image on the pixel grid in Figure 2a. The grid H_3 is illustrated in Figure 2b. Mallat's two-dimensional DWT will be applied to the data in H_3 and a wavelet shrinkage estimator constructed from it at threshold t. The estimator is then interpolated:

right to match the H_2 grid;

down to match the H_1 grid.

To match H_0 it is possible to either interpolate the H_2 grid downwards or the H_1 grid to the right — this demonstrates the invariance with respect to the ordering of the univariate interpolating procedure. Each of the interpolates ($\bar{f}^{01}_{[11]}(t)$, $\bar{f}^{10}_{[11]}(t)$, and $\bar{f}^{00}_{[11]}(t)$) is compared to g_{01}, g_{10} and g_{00} using quadratic loss and each component summed to form the part of the estimate of $\hat{M}$ using H_3 as a starting point for constructing a wavelet estimator. This procedure is then repeated using H_0, H_1 and H_2 as starting points and the contributions are summed to form the final $\hat{M}$.

3.4 Computational effort and optimization

The leave-one-out cross-validation algorithm requires approximately $O(n^2)$ operations for each evaluation of $\hat{M}$. The twofold algorithm requires $O(n)$ operations. The 2^k-fold requires $O\{(2n)^k\}$ operations where n is the length of each side of the hypercube H.

The optimization algorithm that is used in all cases is the simple golden section search as mentioned in Press *et al.* (1992). The algorithm works extremely well in practice. This is mainly because the function $\hat{M}$ is very nearly convex (to the eye on a large scale it looks convincingly

convex). Detailed investigation of $\hat{M}$ by Nason (1994) shows that the first derivative of $\hat{M}(t)$ is continuous and linear increasing on intervals defined by increasing $\{|w_{jk}|\}$ where $\{w_{jk}\}$ are the noisy wavelet coefficients formed from the transform of $g_1, \ldots, g_n$. At the points $t = |w_{jk}|$ the derivative may experience a discontinuity. Nason (1994) provides heuristics that indicate that although these jumps may be negative they are usually small (only negative jumps cause non-convexity of $\hat{M}$) and therefore the zero-derivative point of $\hat{M}$ is usually well-determined. Since the first derivative is known it would be possible to use a gradient-based algorithm to minimize $\hat{M}$ or better still only the points $t = |w_{jk}|$ would need to be checked (following Donoho and Johnstone (1995)).

4 Some Examples

Some of the examples given here are discussed in much greater detail in Nason (1994). Nason (1994) also presents some other examples.

4.1 Piecewise polynomial

The first example uses the piecewise polynomial with discontinuity function $y(x)$ that appeared in Nason and Silverman (1994). The function definition was:

$$y(x) = \begin{cases} 4x^2(3-4x) & \text{for } x \in [0, \frac{1}{2}] \\ \frac{4}{3}x(4x^2 - 10x + 7) - \frac{3}{2} & \text{for } x \in [\frac{1}{2}, \frac{3}{4}] \\ \frac{16}{3}x(x-1)^2 & \text{for } x \in [\frac{3}{4}, 1] \end{cases} \tag{10}$$

and was sampled 512 times in the interval $[0, 1]$. Figure 3 shows y plotted against x. The figure was generated by the SPlus function `example.1()` that comes with the `WaveThresh` software developed by Nason (1993). A noisy version was created by adding independent pseudo-random normal deviates with standard deviation of 0.1 to y. A particular noisy version is shown in Figure 4a. The cross-validation algorithm, *VisuShrink* and *GlobalSure* reconstruction methods were applied to the noisy data and the reconstruction results appear in Figure 4b, c and d. Note that the *VisuShrink* estimate is noise-free, *GlobalSure* overfits slightly and the cross-validated reconstruction minimizes the integrated squared error (ISE) with this example. Although *GlobalSure* is a valid estimation procedure it is not necessarily a good one for this function. The reason why is that *SureShrink* forms level-dependent thresholds and uses *universal* thresholding when it deems that there are few true coefficients at a level (sparsity). In contrast we apply the *Sure* technology on nearly all coefficients and the piecewise polynomial is simple and sparse. There are some signals that even wavelets do not represent sparsely and then the performance of *GlobalSure* is much better (see the chirp example in Nason (1994)).

Naturally, one example is not of much use on its own. Table 1 gives results of 100 simulations. The "true" threshold referred to in Table 1 refers to the threshold that minimizes the ISE when the true function is known. The mean squared errors and their standard deviations appear in Table 2. This table shows that the cross-validation method performs best under the normal independent and Student's t noise conditions, but fails to cope with serial correlation.

One aspect of the simulation results that Tables 1 and 2 do not show is any correlation of the estimates with the "true" threshold (see Hall and Johnstone (1992) for discussion of this phenomenon). The correlations for the experiments in Table 1 are displayed in Table 3. It is difficult to draw any general conclusions for Table 3 except that for experiment A the correlations are negligible apart from the leave-one-out method. The correlation for the leave-one-out method

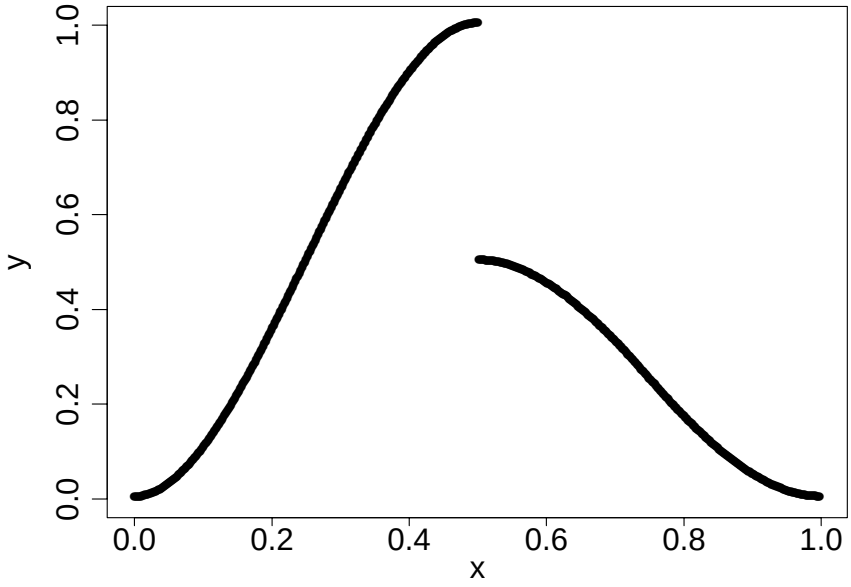


Figure 3: Piecewise polynomial with discontinuity function of Nason and Silverman (1994) sampled 512 times on the interval $[0, 1]$.

Table 1: The mean and standard deviations ($\times 1000$) of the "true" and estimated thresholds that minimize $M(t)$ (true) and $\hat{M}$. The summary statistics were computed from 100 simulations with: A. independent normally distributed deviates, B. independent Student's t on 3 d.f. distributed deviates, C. correlated normally distributed deviates with autocorrelation of 0.5 added to the true function y as defined in (10). In each case the standard deviation of the noise was 0.1. Daubechies' least asymmetric $N = 8$ wavelets were used.

			Mean Estimated Threshold (s.d.)							
Simulation	*True*		*VisuShrink*		*GlobalSure*		*twofold*		*Leave-1-out*	
A.	187	(12)	362	(20)	104	(12)	182	(33)	209	(22)
B.	274	(189)	293	(18)	87	(12)	206	(94)	244	(44)
C.	228	(33)	298	(17)	90	(13)	47	(8)	30	(17)

Table 2: The mean squared error ($\times 1000$) and standard deviations of the simulation runs given in Table 1. The summary statistics were computed from 100 simulations with: A. independent normally distributed deviates, B. independent Student's t on 3 d.f. distributed deviates, C. correlated normally distributed deviates with autocorrelation of 0.5 added to the true function y as defined in (10). In each case the standard deviation of the noise was 0.1. Daubechies' least asymmetric $N = 8$ wavelets were used.

			Mean Squared Error (s.d.)							
Simulation	*True*		*VisuShrink*		*GlobalSure*		*twofold*		*Leave-1-out*	
A.	593	(93)	904	(120)	1020	(213)	634	(103)	617	(98)
B.	857	(376)	1030	(728)	1760	(1200)	968	(540)	980	(760)
C.	860	(177)	937	(206)	1600	(356)	2650	(435)	3430	(819)

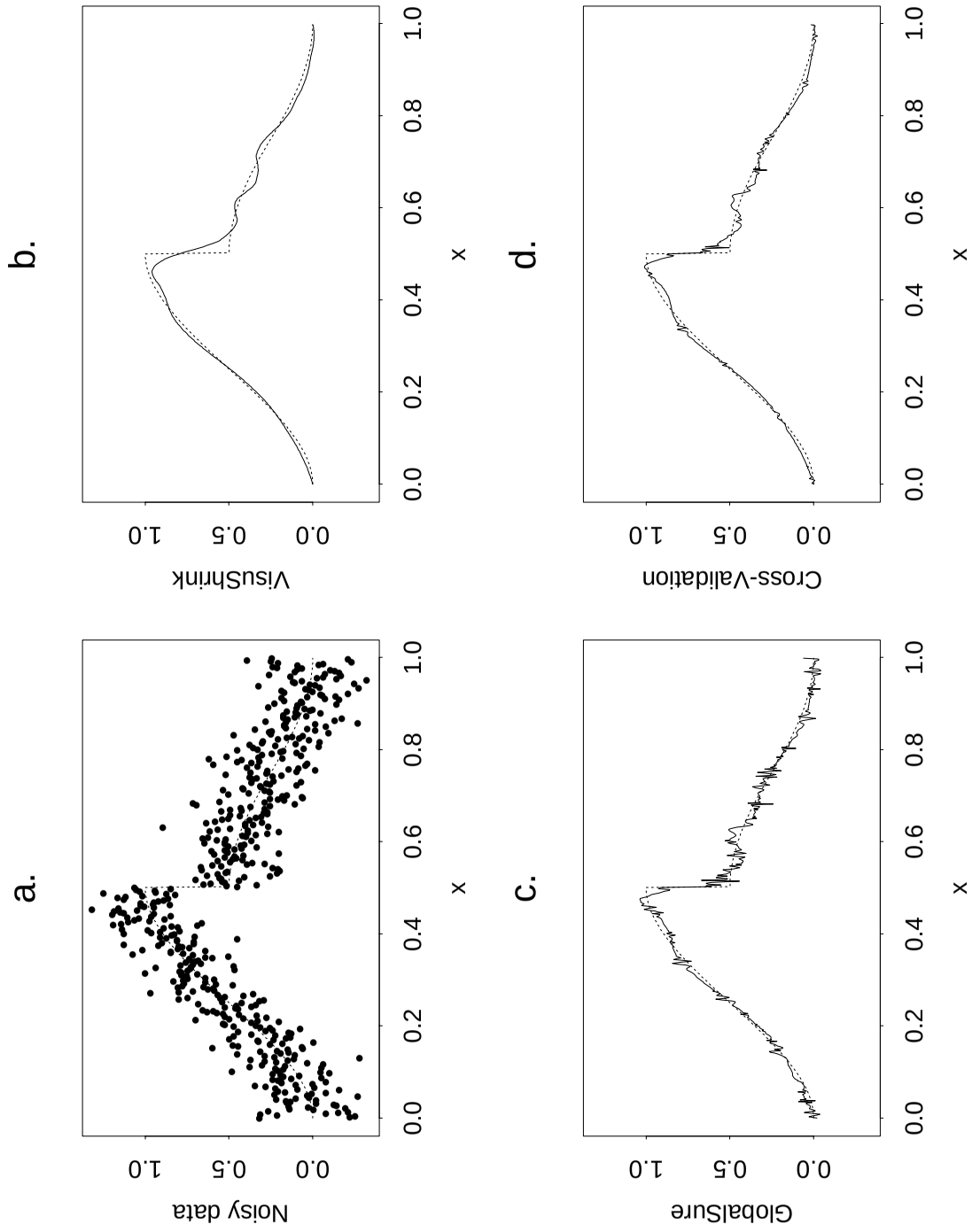


Figure 4: a. Piecewise polynomial y with added independent pseudo-normal deviates with standard deviation of 0.1. Reconstructions using Daubechies' extremal phase wavelet $N = 6$ with thresholds t and integrated squared error, R, between original and reconstruction: b. *VisuShrink*: $t = 0.35, R = 0.95$; c. *GlobalSure*: $t = 0.14, R = 0.83$; d. Cross-validation: $t = 0.20, R = 0.77$. The dotted line is the true function.

Table 3: Correlation between 100 "true" threshold and estimates. The summary statistics were computed from 100 simulations with: A. independent normally distributed deviates, B. independent Student's t on 3 d.f. distributed deviates, C. correlated normally distributed deviates with autocorrelation of 0.5 added to the true function y as defined in (10). In each case the standard deviation of the noise was 0.1. As a rough guide a correlation of 0.196 or above would be significant at the 5% level (0.256 at the 1% level) given a bivariate normal model (see Chatfield (1983)).

	Sample Correlation			
Simulation	*VisuShrink*	*GlobalSure*	*twofold*	*Leave-1-out*
A.	0.071	0.134	-0.027	0.355
B.	0.311	0.477	0.813	0.501
C.	0.060	0.354	-0.287	-0.278

is positive whereas the corresponding correlation in Hall and Johnstone (1992) was negative. It is also interesting, but perhaps not surprising, that the *VisuShrink* algorithm tends to exhibit little correlation.

Serially-correlated data

It is clear from Table 2 that the cross-validation method does not do very well with serially correlated data. This is a well known problem with cross-validation methods (see Diggle (1990), Altman (1990) and Hart (1994)). Nason (1995) gives an example using real exchange rate data where serial correlation causes wavelet cross-validation to choose too small a threshold. It is possible that the problem may be alleviated by leaving out other groups within the cross-validation and then summing over the prediction error contributions from each of these groups.

4.2 Inductance Plethysmography Data

The following example arises from data collected by inductance plethysmography. A plethysmograph is an apparatus for measuring variations in the size of parts of the body. In this experiment the inductance plethysmograph consists of a coil of wire encapsulated in a belt. A radio-frequency carrier signal is passed through the wire and size variations change the inductance of the coil that can be detected as a change in voltage. When properly calibrated the output voltage of the inductance plethysmograph is proportional to the change in volume of the part of the body under examination.

It is of both clinical and scientific interest to discover how anaesthetics or analgesics may alter normal breathing patterns post-operatively. Sensors exist that measure blood oxygen saturation but by the time they indicate critically low levels the patient is often apnoeic (cease breathing) and in considerable danger. It is possible for a nurse to continually observe a patient but this is expensive, prone to error and requires training. In this example 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 recordings below were made by the Department of Anaesthesia at the Bristol Royal Infirmary after the patients had undergone surgery under general anaesthetic. Figure 5a shows a section of plethysmograph recording lasting approximately 80 seconds (4096 data points). The two main sets of regular oscillations correspond to normal breathing. The disturbed behaviour in the centre of the plot where the normal breathing pattern disappears corresponds to the patient vomiting.

Twofold cross-validation

Figures 5b, c and d show the reconstructions via *VisuShrink*, *GlobalSure* and cross-validation. At present signals such as the one in Figure 5a are classified using neural networks. Wavelet shrinkage can act as a dimension reducer, compressing the signal into few coefficients, so that maybe:

- the subsequent neural networks may be smaller and simpler;
- the classification rates will be better as the noise is removed.

For the purposes of dimension reduction it is possible that any of the shrinkage procedures here will be adequate. Although the cross-validation method best retains the sharpness of the peaks, but is still noisy in places whereas the *VisuShrink* procedure effectively suppresses the noise but has slightly more rounded peaks.

Leave-one-out cross-validation

We apply the leave-one-out cross-validation method to a 30 second section of the data in Figure 5a. This section contains 1500 observations, not a power of two. The reconstruction is shown in Figure 6 as the top trace. The bottom trace shows the (translated) cross-validation reconstruction of the corresponding 1500 observations from Figure 5d. The leave-one-out method has chosen a similar threshold to the two-fold algorithm albeit on a reduced set of data. In extensive simulations on data sets that are a power of 2 in length the leave-one-out method performs similarly to the two-fold algorithm.

4.3 Cross-validation using images

The Lennon image of Nason and Silverman (1994) is used to illustrate the two-dimensional cross-validation algorithm. Figure 7 shows the original Lennon image which consists of 256×256 pixels. Figure 8 shows a noisy Lennon image. The noisy image was composed by adding the original image to independent pseudo-random deviates generated by the SPlus function `rnorm` using a noise level of twice the standard deviation of the signal (image pixels). Figure 9 shows the *VisuShrink* thresholded reconstruction from the noisy image. Figure 10 shows the cross-validated reconstruction. As mentioned in Section 2 the *VisuShrink* reconstruction is free of noise, but also appears to be underfitting the data. The cross-validated method obtains a smaller sum of squares with respect to the original image compared to *VisuShrink* and indeed more features of the original appear in the reconstruction and are better defined than those in the *VisuShrink* reconstruction. As with the one-dimensional examples the first four levels of wavelet coefficients (low-frequencies) are not touched by the thresholding procedure. The drastic underfitting in Figures 9 and 10 could be improved if fewer levels were thresholded. For example, in Figure 11 levels 5,6 and 7 have been thresholded and levels zero to four have been left alone. Compare this to the previous two reconstructions in Figures 9 and 10 where levels three and four were thresholded as well.

5 Conclusion

This article has introduced cross-validation to the estimation of functions using wavelets. In particular, twofold cross-validation appears to be particularly adept at selecting a threshold for normally distributed independent data, but suffers from the usual affliction of overfitting in the presence of serial correlation. The cross-validation method extends to higher dimensions in a straightforward manner.

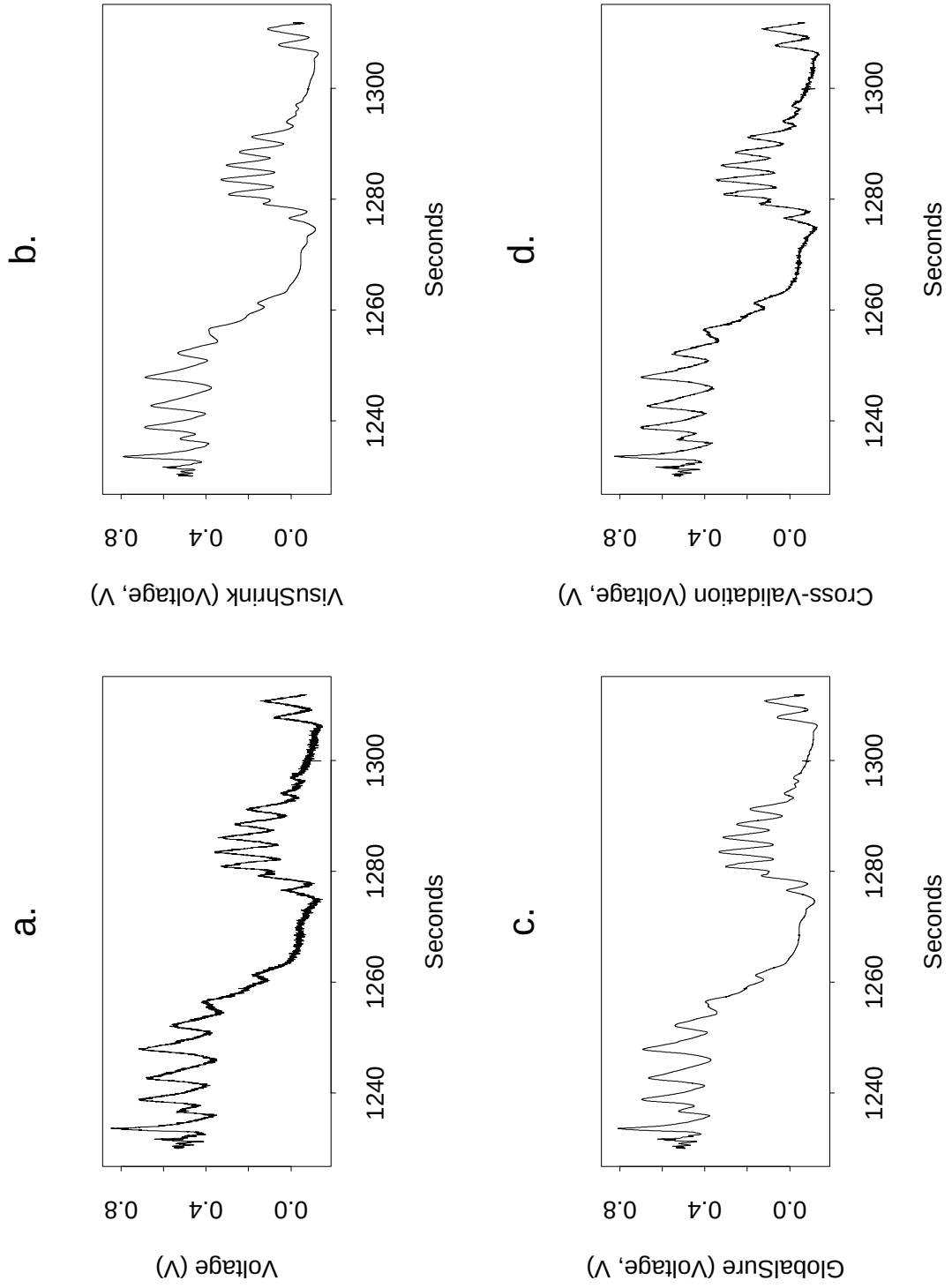


Figure 5: a. Section of inductance plethysmograph recording. Reconstructions using Daubechies' least asymmetric wavelet $N = 6$ with thresholds t: b. *VisuShrink*: $t = 0.048$; c. *GlobalSure*: $t = 0.025$; d. Cross-validation: $t = 0.010$.

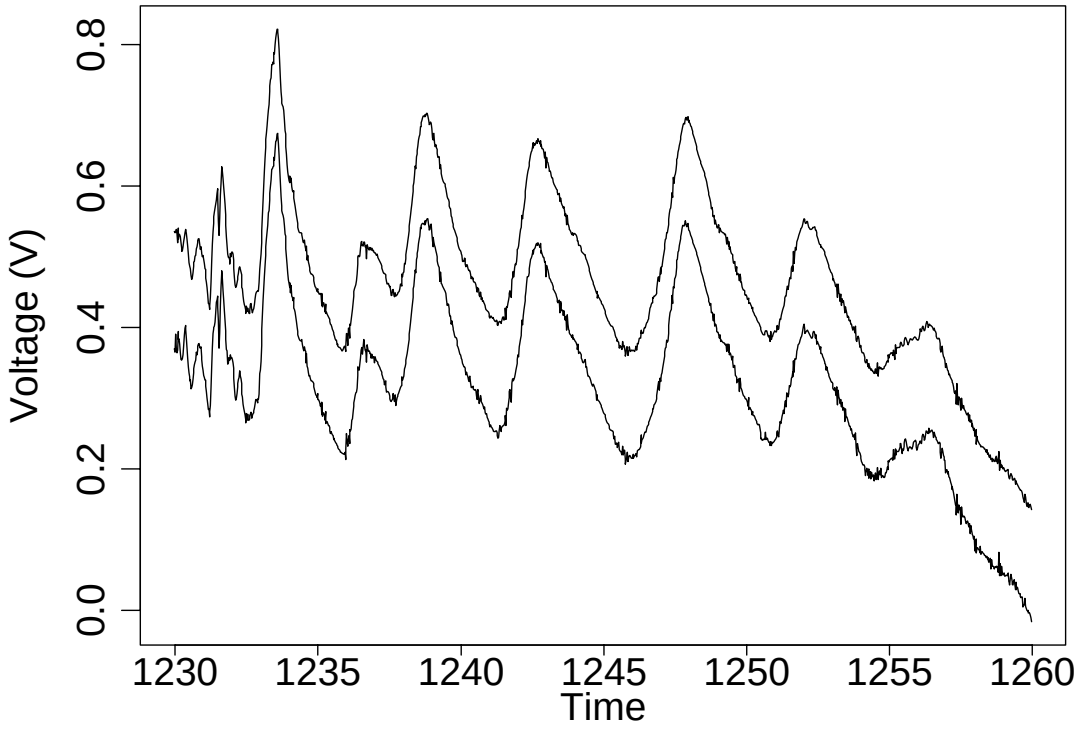


Figure 6: Top trace: leave-one-out cross validation reconstruction ($t = 0.012$). Bottom trace: corresponding 1500 observations of the two-fold cross-validation reconstruction from Figure 5d translated down by 0.15 ($t = 0.010$). The two-fold cross-validation reconstruction has been translated down so it may be compared to the upper trace.

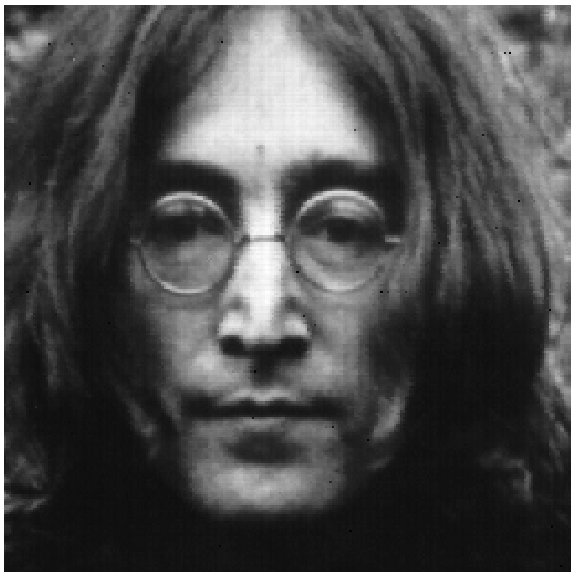


Figure 7: Original Lennon image.

Figure 8: Noisy Lennon image (signal to noise ratio is $\frac{1}{2}$).

Figure 9: *VisuShrink* reconstruction of noisy image in Figure 8 using Daubechies' extremal phase wavelets $N = 8$, soft thresholding and thresholding above level 3. The *VisuShrink* threshold was 534 and the integrated squared error between this reconstruction and the original was 6543.

Figure 10: Cross-validated reconstruction of noisy image in Figure 8 using Daubechies' extremal phase wavelets $N = 8$, soft thresholding and thresholding above level 3. The cross-validated threshold was 222 and the integrated squared error between this reconstruction and the original was 5660.

Figure 11: Cross-validated reconstruction of noisy image in Figure 8 using Daubechies' extremal phase wavelets $N = 8$, soft thresholding and thresholding above level 5. The cross-validated threshold was 514 and the integrated squared error between this reconstruction and the original was 4772.

A leave-one-out cross-validation algorithm has been devised and in simulation experiments on power of two data sets has performed comparably to twofold cross-validation. This leave-one-out algorithm has the advantage that it is applicable to data sets of any length. In common with many statistical methods, especially wavelet methods, the leave-one-out algorithm works best with large data sets.

We fully intend to deposit the Splus code that performs all the above analyses onto the Statlib public archive with the next release of `WaveThresh` (see Nason (1993)).

Further Developments

Further developments include modification to level-dependent thresholding which would improve performance on non-normal and correlated data sets; improving the optimization algorithm to take account of derivative information and theoretical developments that study the asymptotic and other properties of estimation by cross-validation with wavelets.

Level-dependent thresholding in cross-validation for wavelet shrinkage has already been undertaken successfully by Wang (1994) and Weyrich and Warhola (1994). In our view, it would not be too computationally demanding to extend our methods to choose a different threshold at each level by cross-validation. Hopefully it would be possible to choose a threshold for one level at a time whilst holding the other thresholds fixed and then repeat for all levels until convergence. Remarkably the computational effort of this procedure is still $O(N)$ because the number of coefficients halves at each level. In practice though the computational effort is likely to be more because there is an optimization step at each level and probably cycling through the levels will occur more than once. Also, the cross-validation method will not work well at low resolution levels where there are few coefficients. However, this is just the situation where *SureShrink* performs well. Therefore a possible hybrid procedure would be to use *SureShrink* for low resolution levels and cross-validation for medium and high resolution levels.

We should also mention that further improvements may occur when the "first-generation" wavelet transforms used in this paper are replaced by more flexible systems such as wavelet packets (Wickerhauser (1994)) and the stationary wavelet transform (Pesquet *et al.* (1994), Silverman (1995)) Indeed preliminary investigations indicate that the stationary transform will be a promising method for regression problems (Donoho *et al.* (1995b) and Lang *et al.* (1995)).

6 Acknowledgments

I would like to thank Andrew Black and David Moshal of the Department of Anaesthesia, University of Bristol for supplying and explaining the inductance plethysmography data. I am grateful to Bernard Silverman for suggesting the bias correction factor (9) given in Section 3.1.

I would like to thank the Division of Mathematics and Statistics, CSIRO; Department of Mathematics, University of Queensland; AGSM, University of NSW; Centre for Mathematics and its Applications, ANU, Australia and Department of Statistics, Stanford University for support and encouragement whilst this work was developed. In particular I would like to extend special thanks to Mark Berman, Geoff Eagleson, Peter Hall and Iain Johnstone. Finally, I would like to thank Bernard Silverman, the associate editor and referees who helped improve the paper drastically.

This work was supported by a grant under the Complex Stochastic Systems Initiative of the UK Science and Engineering Research Council.

References

Altman, N. S. (1990) Kernel smoothing of data with correlated errors. *J. Am. Statist. Ass.*, **85**, 749–759.

Burman, P. (1989) A comparative study of ordinary cross-validation, ν-fold cross-validation and the repeated learning-testing methods. *Biometrika*, **76**, 503–514.

Chatfield, C. (1983) *Statistics for Technology*, 3rd edn. London: Chapman and Hall.

Cohen, A., Daubechies, I., Jawerth, B. and Vial, P. (1992) Multiresolution analysis, wavelets, and fast algorithms on an interval. *Compt. Rend. Acad. Sci. Paris* A, **316**, 417–421.

Daubechies, I. (1988) Orthonormal bases of compactly supported wavelets. *Communs Pure Appl. Math.*, **41**, 909–996.

Daubechies, I. (1992) *Ten Lectures on Wavelets*. Philadelphia: Society for Industrial and Applied Mathematics.

Diggle, P. J. (1990) *Time Series: a Biostatistical Approach*. Oxford: Oxford University Press.

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. (1995) Adapting to unknown smoothness via wavelet shrinkage. *J. Am. Statist. Ass.*, to be published.

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

Donoho, D. L., Johnstone, I. M., Kerkyacharian, G. and Picard, D. (1995b) Reply to the discussion on Wavelet shrinkage: asymptopia? (by D. L. Donoho, I. M. Johnstone, G. Kerkyacharian and D. Picard). *J. R. Statist. Soc.* B, **57**, 362–366.

Fan, J., Hall, P., Martin, M. and Patil, P. (1993) Adaption to high spatial inhomogeneity based on wavelets and on local linear smoothing. *Technical Report CMA-SR18-93*. Centre for Mathematics and Its Applications, Australian National University, Canberra.

Green, P. J. and Silverman, B.W. (1994) *Nonparametric Regression and Generalized Linear Models*. London: Chapman and Hall.

Hall, P. and Johnstone, I. (1992) Empirical functions and efficient smoothing parameter selection. *J. R. Statist. Soc.* B, **54**, 475–530.

Hart, J. D. (1994) Automated kernel smoothing of dependent data by using time series cross-validation. *J. R. Statist. Soc.* B, **56**, 529–542.

Lang, M., Guo, H., Odegard, J. E., Burrus, C. S. and Wells, R. O. (1995) Nonlinear processing of a shift invariant DWT for noise reduction. *Technical Report CML9503*. Computational Mathematics Laboratory, Rice University, Houston.

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

Meyer, Y. (1992) *Wavelets and Operators*. Cambridge: Cambridge University Press.

Nason, G. P. (1993) *The* `WaveThresh` *package; wavelet transform and thresholding software for S.* Available from the StatLib archive.

Nason, G. P. (1994) Wavelet regression by cross-validation. *Technical Report 447.* Department of Statistics, Stanford University, Stanford.

Nason, G. P. (1995) Choice of the threshold parameter in wavelet function estimation. In *Wavelets and Statistics: Proc XVth Recontres Franco-Belges des Statisticiens* (ed. A. Antoniadis and G. Oppenheim). Springer-Verlag (to be published).

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

Nason, G. P. and Silverman, B. W. (1995) The stationary wavelet transform and some statistical applications. *In preparation.*

Pesquet, J. C., Krim, H. and Carfantan, H. (1994) Time-invariant orthonormal wavelet representations. *IEEE Trans. Signal Process.*, to be published.

Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P. (1992) *Numerical Recipes in C, the Art of Scientific Computing*, 2nd edn. Cambridge: Cambridge University Press.

Silverman, B. W. (1986) *Density Estimation*. London: Chapman and Hall.

Silverman, B. W. (1995) Discussion on Wavelet shrinkage: asymptopia? (by D. L. Donoho, I. M. Johnstone, G. Kerkyacharian and D. Picard). *J. R. Statist. Soc.* B, **57**, 339–341.

Smith, M. J. T. and Barnwell, T. P. (1986) Exact reconstruction techniques for tree-structured subband coders. *IEEE Trans. Acoust. Spch Signal Process.*, **34**, 434–441.

Stein, C. (1981) Estimation of the mean of a multivariate normal distribution. *Ann. Statist.*, **9**, 1135–1151.

Stone, M. (1974) Cross-validatory choice and assessment of statistical predictions (with discussion). *J. R. Statist. Soc.* B, **36**, 111–47.

Strang, G. (1993) Wavelet transforms versus Fourier transforms. *Bull. (New Series) Am. Math. Soc.*, **28**, 288–305.

Wang, Y. (1994) Function estimation via wavelets for data with long-range dependence. *Technical Report.* Department of Statistics, University of Missouri, Columbia.

Weyrich, N. and Warhola, G. T. (1994) De-noising using wavelets and cross-validation. *Technical Report AFIT/ENC 2950 P ST.* Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio.

Wickerhauser, M. V. (1994) Adapted wavelet analysis from theory to software. Wellesley, Massachusetts: A. K. Peters.