WAVELETS FOR KIDS
A Tutorial Introduction

By
Brani Vidakovic and Peter Mueller
Duke University

Strictly speaking, wavelets are topic of pure mathematics, however in only a few years of existence as a theory of their own, they have shown great potential and applicability in many fields.

There are several excellent monographs and articles talking about wavelets, and this modest tutorial does not intend to compete with any of them. Rather it is intended to serve as a very first reading, giving examples interesting for the statistical community. We also give references for further reading as well as some MATHEMATICA do-it-yourself procedures.

Key words and phrases: Wavelets, Multiresolution analysis (MRA), Haar wavelet, Thresholding.

1991 AMS Subject Classification: 42A06, 41A05, 65D05.

1 What are wavelets?

Wavelets are functions that satisfy certain requirements. The very name *wavelet* comes from the requirement that they should integrate to zero, "waving" above and below the x-axis. The diminutive connotation of *wavelet* suggest the function has to be well localized. Other requirements are technical and needed mostly to insure quick and easy calculation of the direct and inverse wavelet transform.

There are many kinds of wavelets. One can choose between smooth wavelets, compactly supported wavelets, wavelets with simple mathematical expressions, wavelets with simple associated filters, etc. The most simple is the *Haar wavelet,* and we discuss it as an introductory example in the next section. Examples of some wavelets (from the family of Daubechies wavelets) are given in Figure 1. Like sines and cosines in Fourier analysis, wavelets are used as basis functions in representing other functions. Once the wavelet (sometimes called *the mother wavelet*) $\psi(x)$ is fixed, one can form of translations and dilations of the mother wavelet $\{\psi(\frac{x-b}{a}), (a,b) \in R^+ \times R\}$. It is convenient to take special values for a and b in defining the wavelet basis: $a = 2^{-j}$ and $b = k \cdot 2^{-j}$, where k and j are integers. This choice of a and b is called *critical sampling* and will give a sparse basis. In addition, this choice naturally connects multiresolution analysis in signal processing with the world of wavelets.

Wavelet novices often ask, why not use the traditional Fourier methods? There are some important differences between Fourier analysis and wavelets. Fourier basis functions are localized in frequency but not in time. Small frequency changes in the Fourier transform will produce changes everywhere in the time domain. Wavelets are local in both frequency/scale (via dilations) and in time (via translations). This localization is an advantage in many cases.

Second, many classes of functions can be represented by wavelets in a more compact way. For example, functions with discontinuities and functions with sharp spikes usually take substantially fewer wavelet basis functions than sine-cosine basis functions to achieve a comparable approximation.

This sparse coding makes wavelets excellent tools in data compression. For example, the FBI has standardized the use of wavelets in digital fingerprint image compression. The compression ratios are on the order of 20:1, and the difference between the original image and the decompressed one can be told only by an expert. There are many more applications of wavelets, some of them very pleasing. Coifman and his Yale team used wavelets to clean noisy sound recordings, including old recordings of Brahms playing his *First Hungarian Dance* on the piano.

This already hints at how statisticians can benefit from wavelets. Large and noisy data sets can be easily and quickly transformed by the discrete wavelet transform (the counterpart of the discrete Fourier transform). The data are coded by the wavelet coefficients. In addition, the epithet "fast" for Fourier transform can, in most cases, be replaced by "faster" for the wavelets. It is well known that the computational complexity of the fast Fourier transformation is $O(n \cdot \log_2(n))$. For the fast wavelet

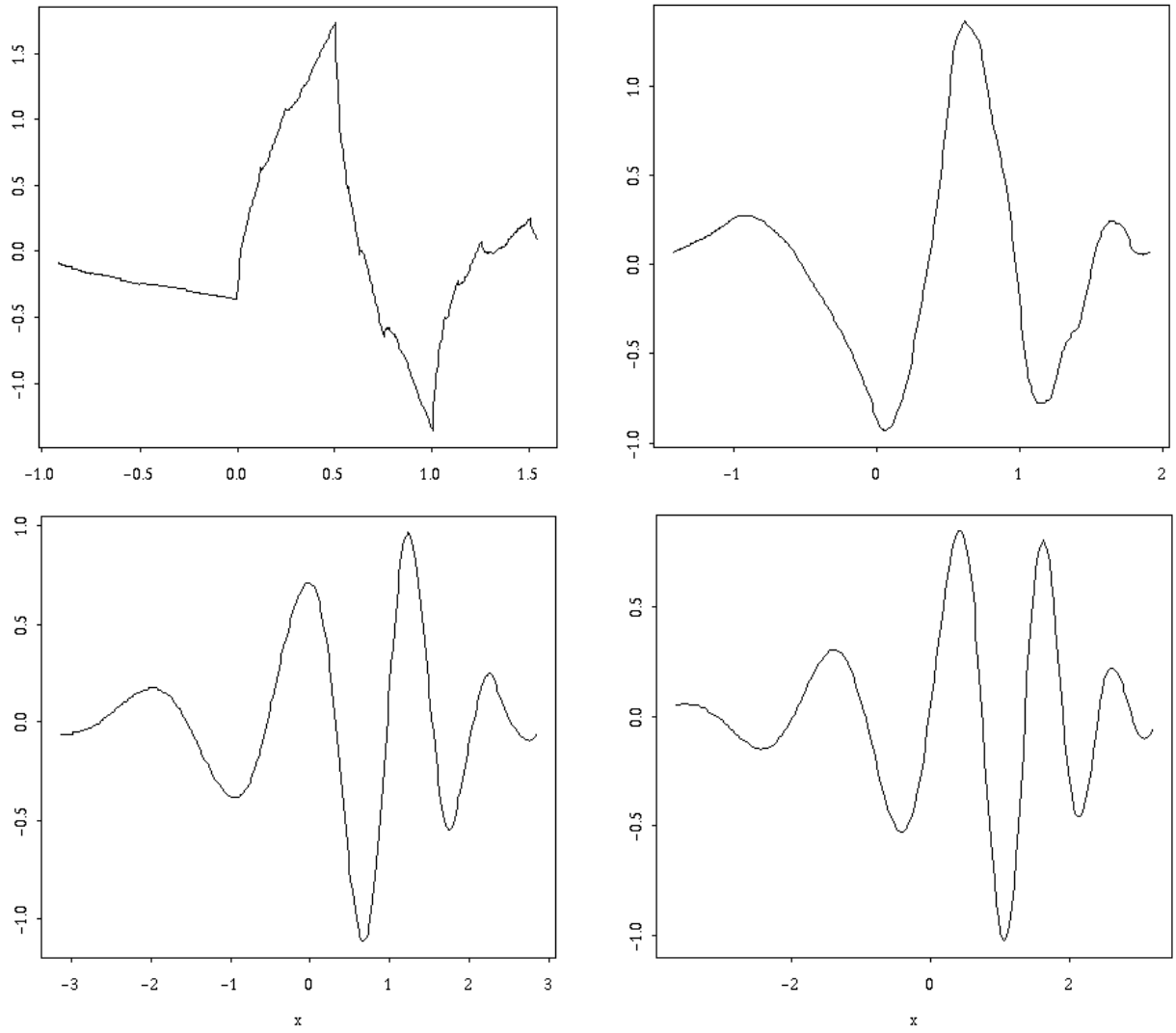


Figure 1: Wavelets from the Daubechies family

transform the computational complexity goes down to $O(n)$.

Many data operations can now be done by processing the corresponding wavelet coefficients. For instance, one can do data smoothing by thresholding the wavelet coefficients and then returning the thresholded code to the "time domain." The definition of thresholding and different thresholding methods are given in Section 3.

Figure 2: Data analysis by wavelets

2 How do the wavelets work?

2.1 The Haar wavelet

To explain how wavelets work, we start with an example. We choose the simplest and the oldest of all wavelets (we are tempted to say: mother of all wavelets!), the Haar wavelet, $\psi(x)$. It is a step function taking values 1 and -1, on $[0, \frac{1}{2})$ and $[\frac{1}{2}, 1)$, respectively. The graph of the Haar wavelet is given in Figure 3.

The Haar wavelet has been known for more than eighty years and has been used in various mathematical fields. It is known that any continuous function can be approximated uniformly by Haar functions. (Brownian motion can even be defined by using the Haar wavelet.[1]) Dilations and translations of the function ψ,

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

define an orthogonal basis in $L^2(R)$ (the space of all square integrable functions). This means that any element in $L^2(R)$ may be represented as a linear combination (possibly infinite) of these basis functions.

The orthogonality of ψ_{jk} is easy to check. It is apparent that

$$\int \psi_{jk} \cdot \psi_{j'k'} = 0, \tag{1}$$

whenever $j = j'$ and $k = k'$ is not satisfied simultaneously.

If $j \neq j'$ (say $j' < j$), then nonzero values of the wavelet $\psi_{j'k'}$ are contained in the set where the wavelet ψ_{jk} is constant. That makes integral (1) equal to zero:

If $j = j'$, but $k \neq k'$, then at least one factor in the product $\psi_{j'k'} \cdot \psi_{jk}$ is zero. Thus the functions ψ_{ij} are orthogonal.

[1] If $\xi \sim_{iid} N(0,1)$ and $S_{jk}(t) = \int_0^t \psi_{jk}(x)dx$, then $B_t =_{def} \Sigma_{j=1}^{\infty} \Sigma_{k=0}^{2^j-1} \xi_{jk} S_{jk}(t)$ (**P. Levy**).

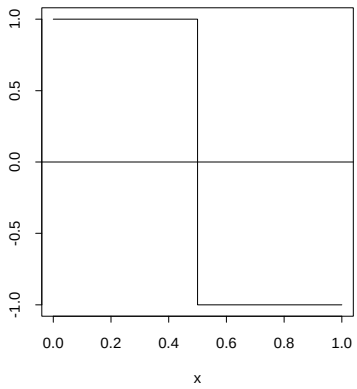


Figure 3: Haar wavelet

The constant that makes this orthogonal basis orthonormal is $2^{j/2}$. Indeed, from the definition of norm[2] in L^2 :

$$1 = (\text{const})^2 \int \psi^2(2^j x - k)dx = (\text{const})^2 \cdot 2^{-j} \int \psi^2(t)dt = (\text{const})^2 \cdot 2^{-j}.$$

The functions $\psi_{10}, \psi_{11}, \psi_{20}, \psi_{21}, \psi_{22}, \psi_{23}$ are depicted in Figure 4. The set $\{\psi_{jk}, j \in Z, k \in Z\}$ defines an orthonormal basis for L^2. Alternatively we will consider orthonormal bases of the form $\{\phi_{j_0,k}, \psi_{jk}, j \geq j_0, k \in Z\}$, where ϕ_{00} is called the *scaling function* associated with the wavelet basis ψ_{jk}. The set $\{\phi_{j_0 k}, k \in Z\}$ spans the same subspace as $\{\psi_{jk}, j < j_0, k \in Z\}$. We will later make this statement more formal and define ϕ_{jk}. For the Haar wavelet basis the scaling function is very simple. It is unity on the interval [0,1), i.e.

$$\phi(x) = \mathbf{1}(0 \leq x < 1).$$

The statistician may be interested in wavelet representations of functions generated by data sets.

Let $\underset{\sim}{y} = (y_0, y_1, \ldots, y_{2^n-1})$ be the data vector of size 2^n. The data vector can be associated with a piecewise constant function f on [0,1) generated by $\underset{\sim}{y}$ as follows,

$$f(x) = \Sigma_{k=0}^{2^n-1} \; y_k \; \cdot \; \mathbf{1}(k2^{-n} \leq x < (k+1)2^{-n}).$$

The (data) function f is obviously in the $L^2[0,1)$ space, and the wavelet decomposition of f has the form

$$f(x) = c_{00}\phi(x) + \Sigma_{j=0}^{n-1}\Sigma_{k=0}^{2^j-1} d_{jk}\psi_{jk}(x). \tag{2}$$

The sum with respect to j is finite because f is a step function, and everything can be exactly described by resolutions up to the $(n-1)$-st level. For each level the sum

[2] $||f||^2 =_{def} \langle f, f \rangle = \int f^2$.

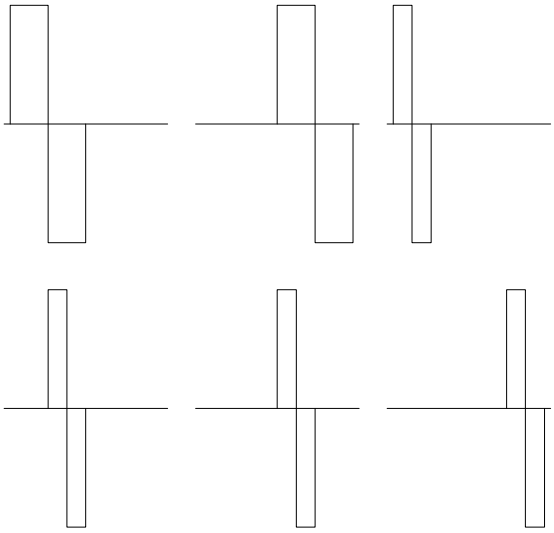


Figure 4: Dilations and translations of Haar wavelet on [0,1]

with respect to k is also finite because the domain of f is finite. In particular, no translations of the scaling function ϕ_{00} are required.

We fix the data vector y and find the wavelet decomposition (2) explicitly. Let $y = (1, 0, -3, 2, 1, 0, 1, 2)$. The corresponding function f is given in Figure 5. The following matrix equation gives the connection between y and the wavelet coefficients. Note the constants 2^j ($1, \sqrt{2}$ and 2) with Haar wavelets on the corresponding resolution levels (j=0, 1, and 2).

$$\begin{bmatrix} 1 \\ 0 \\ -3 \\ 2 \\ 1 \\ 0 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & \sqrt{2} & 0 & 2 & 0 & 0 & 0 \\ 1 & 1 & \sqrt{2} & 0 & -2 & 0 & 0 & 0 \\ 1 & 1 & -\sqrt{2} & 0 & 0 & 2 & 0 & 0 \\ 1 & 1 & -\sqrt{2} & 0 & 0 & -2 & 0 & 0 \\ 1 & -1 & 0 & \sqrt{2} & 0 & 0 & 2 & 0 \\ 1 & -1 & 0 & \sqrt{2} & 0 & 0 & -2 & 0 \\ 1 & -1 & 0 & -\sqrt{2} & 0 & 0 & 0 & 2 \\ 1 & -1 & 0 & -\sqrt{2} & 0 & 0 & 0 & -2 \end{bmatrix} \cdot \begin{bmatrix} c_{00} \\ d_{00} \\ d_{10} \\ d_{11} \\ d_{20} \\ d_{21} \\ d_{22} \\ d_{23} \end{bmatrix}$$

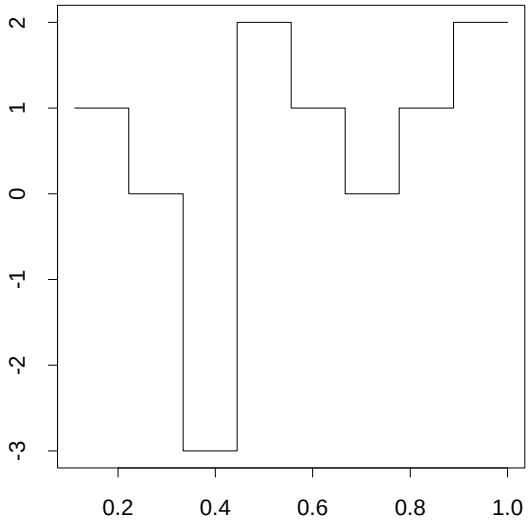


Figure 5: "Data function" on [0,1)

The solution is

$$\begin{bmatrix} c_{00} \\ d_{00} \\ d_{10} \\ d_{11} \\ d_{20} \\ d_{21} \\ d_{22} \\ d_{23} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ \frac{1}{2\sqrt{2}} \\ -\frac{1}{2\sqrt{2}} \\ \frac{1}{4} \\ -\frac{5}{4} \\ \frac{1}{4} \\ -\frac{1}{4} \end{bmatrix}.$$

Thus,

$$f = \frac{1}{2}\phi - \frac{1}{2}\psi_{00} + \frac{1}{2\sqrt{2}}\psi_{10} - \frac{1}{2\sqrt{2}}\psi_{11} + \frac{1}{4}\psi_{20} - \frac{5}{4}\psi_{21} + \frac{1}{4}\psi_{22} - \frac{1}{4}\psi_{23} \tag{3}$$

The solution is easy to check. For example, when $x \in [0, \frac{1}{8})$,

$$f(x) = \frac{1}{2} - \frac{1}{2} \cdot 1 + \frac{1}{2\sqrt{2}} \cdot \sqrt{2} + \frac{1}{4} \cdot 2 = 1.$$

The reader may already have the following question ready: "What will we do for vectors y of much bigger length?" Obviously, solving the matrix equations becomes impossible.

2.2 Mallat's multiresolution analysis, filters, and direct and inverse wavelet transformation

An obvious disadvantage of the Haar wavelet is that it is not continuous, and therefore choice of the Haar basis for representing smooth functions, for example, is not natural and economic.

2.2.1 Mallat's MRA

As a more general framework we explain Mallat's Multiresolution Analysis – (MRA). The MRA is a tool for a constructive description of different wavelet bases.

We start with the space L^2 of all square integrable functions.[3] The MRA is an increasing sequence of closed subspaces $\{V_j\}_{j\in Z}$ which approximate $L^2(R)$.

Everything starts with a clever choice of the *scaling function* ϕ. Except for the Haar wavelet basis for which ϕ is the characteristic function of the interval $[0,1)$, the scaling function is chosen to satisfy some continuity, smoothness and tail requirements. But, most importantly, the family $\{\phi(x-k), \quad k \in Z\}$ forms an orthonormal basis for the *reference* space V_0. The following relations describe the analysis.

MRA 1 $\qquad \cdots \subset V_{-1} \subset V_0 \subset V_1 \subset \cdots$

The spaces V_j are nested. The space $L^2(R)$ is a closure of the union of all V_j. In other words, $\cup_{j\in Z} V_j$ is dense in $L^2(R)$. The intersection of all V_j is empty.

MRA 2 $\qquad f(x) \in V_j \Leftrightarrow f(2x) \in V_{j+1}, \ j \in Z.$

The spaces V_j and V_{j+1} are "similar." If the space V_j is spanned by $\phi_{jk}(x), k \in Z$ then the space V_{j+1} is spanned by $\phi_{j+1,k}(x), k \in Z$. The space V_{j+1} is generated by the functions $\phi_{j+1,k}(x) = \sqrt{2}\phi_{jk}(2x)$.

We now explain how the wavelets enter the picture. Because $V_0 \subset V_1$, any function in V_0 can be written as a linear combination of the basis functions $\sqrt{2}\phi(2x-k)$ from V_1. In particular:

$$\phi(x) = \Sigma_k h(k)\sqrt{2}\phi(2x-k). \tag{4}$$

Coefficients $h(k)$ are defined as $\langle \phi(x), \sqrt{2}\phi(2x-k)\rangle$. Consider now the orthogonal complement W_j of V_j to V_{j+1} (i.e. $V_{j+1} = V_j \oplus W_j$). Define

$$\psi(x) = \sqrt{2}\Sigma_k (-1)^k h(-k+1)\phi(2x-k). \tag{5}$$

It can be shown that $\{\sqrt{2}\psi(2x-k), k \in Z\}$ is an orthonormal basis for W_1.[4]

[3] A function f is in $L^2(S)$ if $\int_S f^2$ is finite.

[4] This can also be expressed in terms of Fourier transformations as follows: Let $m_0(\omega)$ be the

Again, the similarity property of MRA gives that $\{2^{j/2}\psi(2^j x - k), k \in Z\}$ is a basis for W_j. Since $\cup_{j\in Z} V_j = \cup_{j\in Z} W_j$ is dense in $L_2(R)$, the family $\{\psi_{jk}(x) = 2^{j/2}\psi(2^j x - k), j \in Z, k \in Z\}$ is a basis for $L^2(R)$.

For a given function $f \in L^2(R)$ one can find N such that $f_N \in V_N$ approximates f up to preassigned precision (in terms of L_2 closeness). If $g_i \in W_i$ and $f_i \in V_i$, then

$$f_N = f_{N-1} + g_{N-1} = \Sigma_{i=1}^{M} g_{N-M} + f_{N-M}. \tag{6}$$

Equation (6) is the wavelet decomposition of f. For example, the data function (2.1) is in V_n, if we use the MRA corresponding to the Haar wavelet. Note that $f \equiv f_n$ and $f_0 = 0$.

2.2.2 The language of signal processing

We repeat the multiresolution analysis story in the language of signal processing theory. Mallat's multiresolution analysis is connected with so called "pyramidal" algorithms in signal processing. Also, "quadrature mirror filters" are hidden in Mallat's MRA.

Recall from the previous section that

$$\phi(x) = \Sigma_{k\in Z} h(k)\sqrt{2}\phi(2x - k), \tag{7}$$

and

$$\psi(x) = \Sigma_{k\in Z} g(k)\sqrt{2}\phi(2x - k). \tag{8}$$

The l^2 sequences[5] $\{h(k), k \in Z\}$ and $\{g(k), k \in Z\}$ are *quadrature mirror filters* in the terminology of signal analysis. The connection between h and g is given by:

$$g(n) = (-1)^n h(1 - n).$$

The sequence $h(k)$ is known as a *low pass* or *low band* filter while $g(k)$ is known as the *high pass* or *high band* filter. The following properties of $h(n), g(n)$ can be proven by using Fourier transforms and orthogonality: $\Sigma h(k) = \sqrt{2}$, $\Sigma g(k) = 0$.

The most compact way to describe the Mallat's MRA as well to give effective procedures of determining the wavelet coefficients is the *operator representation of filters.*

Fourier transformation of the sequence $h(n), n \in Z$, i.e. $m_0(\omega) = \Sigma_n h(n)e^{in\omega}$. In the 'frequency domain" the relation (4) is $\hat{\phi}(\omega) = m_0(\frac{\omega}{2})\hat{\phi}(\frac{\omega}{2})$. If we define $m_1(\omega) = e^{-i\omega} m_0(\omega + \pi)$ and $\hat{\psi}(2\omega) = m_1(\frac{\omega}{2})\hat{\phi}(\frac{\omega}{2})$, then the function ψ corresponding to $\hat{\psi}$ is *the wavelet associated with the* MRA.

[5] A sequence $\{a_n\}$ is in the Hilbert space l^2 if $\Sigma_{k\in Z} a_k^2$ is finite.

For a sequence $a = \{a_n\}$ the operators H and G are defined by the following coordinatewise relations:

$$\begin{aligned}(Ha)_k &= \Sigma_n h(n-2k)a_n \\ (Ga)_k &= \Sigma_n g(n-2k)a_n.\end{aligned}$$

The operators H and G correspond to one step in the wavelet decomposition. The only difference is that the above definitions do not include the $\sqrt{2}$ factor as in Equations (4) and (5).

Denote the original signal by $\underset{\sim}{c}^{(n)}$. If the signal is of length 2^n, then $\underset{\sim}{c}^{(n)}$ can be represented by the function $f(x) = \Sigma c_k^{(n)} \phi_{nk}$, $f \in V_n$. At each stage of the wavelet transformation we move to a coarser approximation $\underset{\sim}{c}^{(j-1)}$ by $\underset{\sim}{c}^{(j-1)} = H\underset{\sim}{c}^{(j)}$ and $\underset{\sim}{d}^{(j-1)} = G\underset{\sim}{c}^{(j)}$. Here, $\underset{\sim}{d}^{(j-1)}$ is the "detail" lost by approximating $\underset{\sim}{c}^{(j)}$ by the averaged $\underset{\sim}{c}^{(j-1)}$. The discrete wavelet transformation of a sequence $\underset{\sim}{y} = \underset{\sim}{c}^{(n)}$ of length 2^n can then be represented as another sequence of length 2^n (notice that the sequence $\underset{\sim}{c}^{(j-1)}$ has half the length of $\underset{\sim}{c}^{(j)}$):

$$(\underset{\sim}{d}^{(n-1)}, \underset{\sim}{d}^{(n-2)}, \ldots, \underset{\sim}{d}^{(1)}, \underset{\sim}{d}^{(0)}, \underset{\sim}{c}^{(0)}). \tag{9}$$

Thus the discrete wavelet transformation can be summarized as a single line:

$$\underset{\sim}{y} \longrightarrow (G\underset{\sim}{y}, GH\underset{\sim}{y}, GH^2\underset{\sim}{y}, \ldots, GH^{n-1}\underset{\sim}{y}, H^n\underset{\sim}{y}).$$

The reconstruction formula is also simple in terms of H and G; we first define adjoint operators $H^\star$ and $G^\star$ as follows:

$$\begin{aligned}(H^\star a)_n &= \Sigma_k h(n-2k)a_k \\ (G^\star a)_n &= \Sigma_k g(n-2k)a_k.\end{aligned}$$

Recursive application leads to:

$$(G\underset{\sim}{y}, GH\underset{\sim}{y}, GH^2\underset{\sim}{y}, \ldots, GH^{j-1}\underset{\sim}{y}, H^j\underset{\sim}{y}) \longrightarrow \underset{\sim}{y} = \Sigma_{j=0}^{n-1}(H^\star)^j G^\star \underset{\sim}{d}^{(j)} + (H^\star)^n \underset{\sim}{c}^{(0)}.$$

Equations (7) and (8) which generate filter coefficients (sometimes called *dilation equations*) look very simple for the Haar wavelet:

$$\begin{aligned}\phi(x) &= \phi(2x) + \phi(2x-1) = \frac{1}{\sqrt{2}}\sqrt{2}\phi(2x) + \frac{1}{\sqrt{2}}\sqrt{2}\phi(2x-1), \\ \psi(x) &= \phi(2x) - \phi(2x-1) = \frac{1}{\sqrt{2}}\sqrt{2}\phi(2x) - \frac{1}{\sqrt{2}}\sqrt{2}\phi(2x-1).\end{aligned} \tag{10}$$

The filter coefficients in (10) are

$$h(0) = h(1) = \frac{1}{\sqrt{2}} \qquad g(0) = -g(1) = \frac{1}{\sqrt{2}}$$

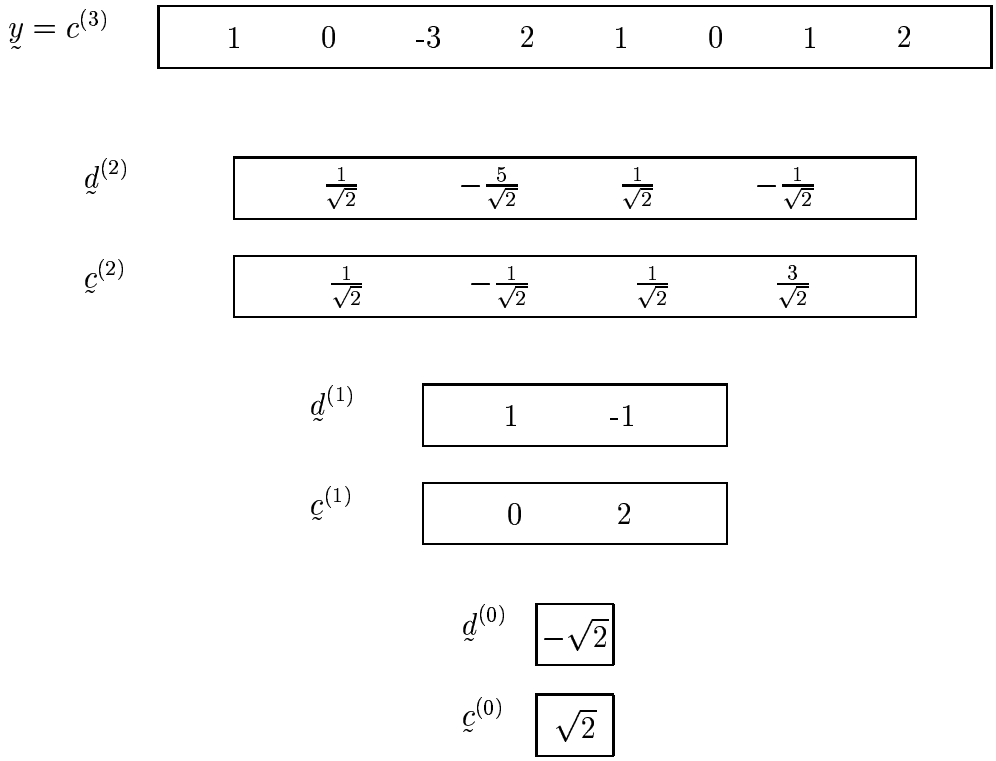


Figure 6: Decomposition procedure

Figure 6 schematically gives the decomposition algorithm applied to our data set.

To get the wavelet coefficients as in (3) we multiply components of $\underset{\sim}{d}^{(j)}$, $j = 0, 1, 2$ and $c^{(0)}$ with the factor $2^{-N/2}$. Simply,

$$d_{jk} = 2^{-N/2} d_k^{(j)}, \quad 0 \leq j < N \ (= 3).$$

It is interesting that in the Haar wavelet case $2^{-3/2} c_0^{(0)} = c_{00} = \frac{1}{2}$ is the mean of the sample $\underset{\sim}{y}$.

Figure 7 schematically gives the reconstruction algorithm for our example.

The careful reader might have already noticed that when the length of the filter is larger than 2, boundary problems occur. (There are no boundary problems with the Haar wavelet!) There are two main ways to handle the boundaries: *symmetric* and *periodic*.

3 Thresholding methods

In wavelet decomposition the filter H is an "averaging" filter while its mirror counterpart G produces details. The wavelet coefficients correspond to details. When details are small, they might be omitted without substantially affecting the "general picture." Thus the idea of thresholding wavelet coefficients is a way of cleaning out "unimportant" details considered to be noise. We illustrate the idea on our old friend, the data vector $(1, 0, -3, 2, 1, 0, 1, 2)$.

Example: The data vector $(1, 0, -3, 2, 1, 0, 1, 2)$ is transformed into the vector

$$(\frac{1}{\sqrt{2}}, -\frac{5}{\sqrt{2}}, \frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 1, -1, -\sqrt{2}, \sqrt{2}).$$

If all coefficients less than 0.9 (well, our choice) are replaced by zeroes, then the resulting ("thresholded") vector is $(0, -\frac{5}{\sqrt{2}}, 0, 0, 1, -1, -\sqrt{2}, \sqrt{2})$.

The graph of "smoothed data", after reconstruction, is given in Figure 8.

An important feature of wavelets is that they provide unconditional bases[6] for not only L^2, but variety of smoothness spaces such as Sobolev and Hölder spaces. As a consequence, wavelet shrinkage acts as a smoothing operator. The same can not be said about Fourier basis. By shrinking Fourier coefficients one can get bad results

[6]Informaly, a family $\{\psi_i\}$ is an unconditional basis for a space S if one can decide if the element $f = \Sigma_i a_i \psi_i$ belongs to S by looking only at $|a_i|$s.

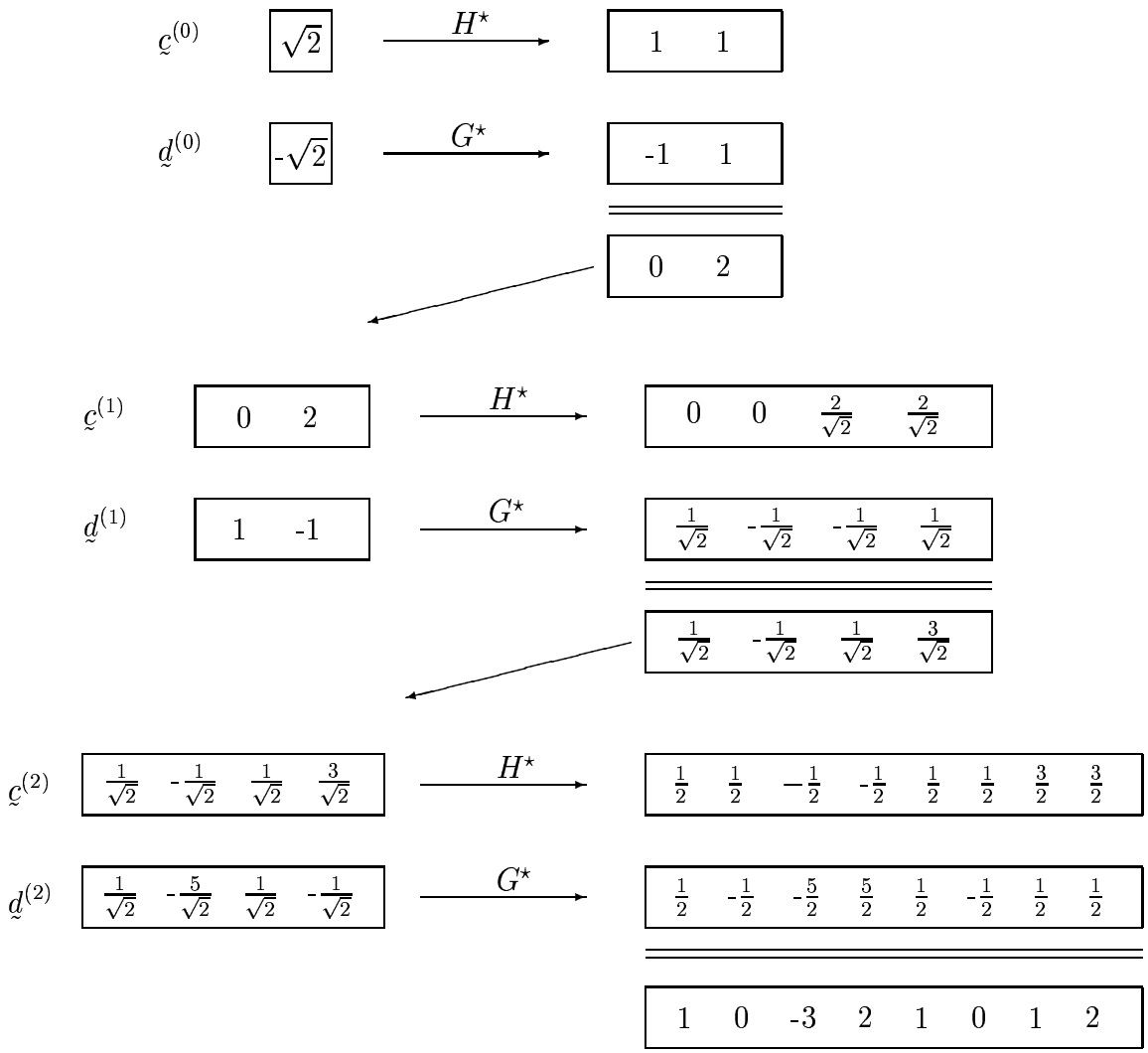


Figure 7: Reconstruction example

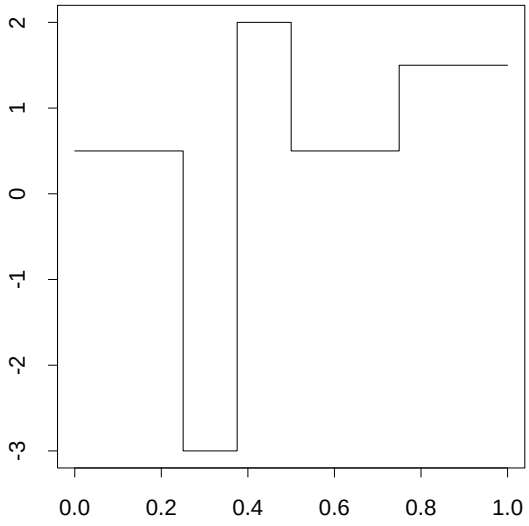


Figure 8: "Smoothed" sequence

in terms of mean square error. Also, some bad visual artifacts can be obtained, see Donoho (1993).

Why is thresholding good? The parsimony of wavelet transformations ensures that the signal of interest can be well described by a relatively small number of wavelet coefficients. A simple Taylor series argument shows that if the mother wavelet has L vanishing moments and the unknown "signal" is in C^{L-1}, then

$$|d_{jk}| \leq \text{const} \cdot 2^{-j(L-1/2)} \int |y|^L |\psi(y)| dy.$$

For j large (fine scales) this will be negligible. For a nice discussion on a compromise between regularity (number of vanishing moments) and the mother wavelet support see Daubechies (1992), page 244.

The process of thresholding wavelet coefficients can be divided into two steps. The first step is the policy choice, i.e., the choice of the threshold function T. Two standard choices are: **hard** and **soft** thresholding with corresponding transformations given by:

$$T^{hard}(d, \lambda) = d \ \mathbf{1}(|d| > \lambda), \tag{11}$$

$$T^{soft}(d, \lambda) = (d - sgn(d)\lambda) \ \mathbf{1}(|d| > \lambda). \tag{12}$$

The "hyperbola" function:

$$T^{hyper}(d, \lambda) = sgn(d)\sqrt{d^2 - \lambda^2} \ \mathbf{1}(|d| > \lambda), \tag{13}$$

is a compromise between hard and soft thresholding functions, (Vidakovic, 1994b). The function T^{hyper} is an "almost" hard thresholder with the continuity property.

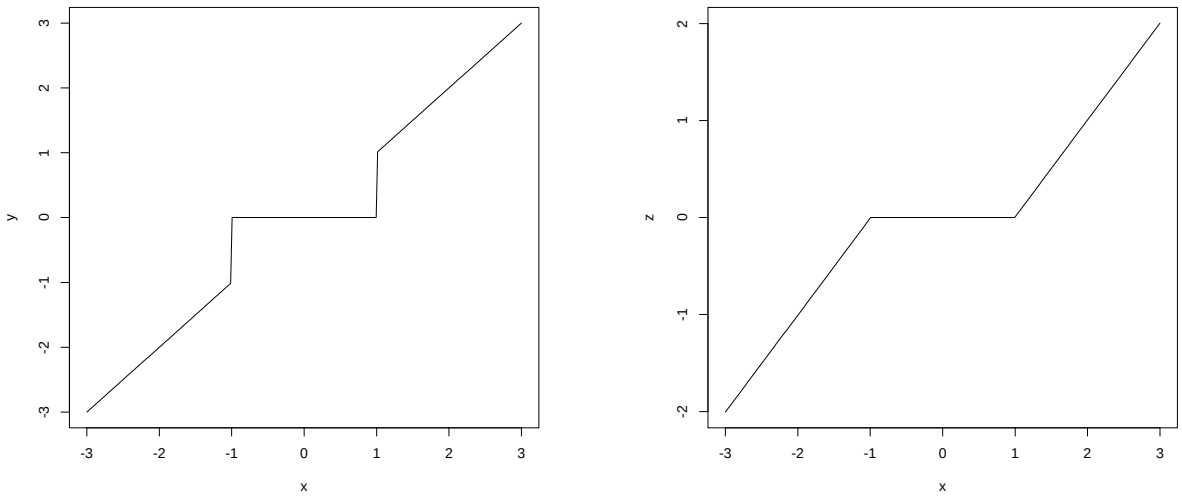


Figure 9: Hard and soft thresholding with $\lambda = 1$.

Another class of useful functions are shrinkage (tapering) functions. A function S from that class exhibits the following properties:

$$S(d) \sim 0, d \text{ small}; \quad S(d) \sim d, d \text{ large}.$$

The second step is the choice of a threshold. In the following subsections we briefly discuss some of the standard methods of selecting a threshold.

3.1 Universal threshold

Donoho and Johnstone (1993) propose a threshold λ based on the following result.

Result: Let z_i be iid standard normal random variables. Define

$$A_n = \{\max_{i=1,n} |z_i| \leq \sqrt{2 \log n}\}.$$

Then

$$\pi_n = P(A_n) \to 0, n \to \infty.$$

In addition, if

$$B_n(t) = \{\max_{i=1,n} |z_i| > t + \sqrt{2 \log n}\}.$$

then $P(B_n(t)) < e^{-\frac{t^2}{2}}$. That motivates the following threshold:

$$\lambda^U = \sqrt{2 \log n}\ \hat{\sigma}, \tag{14}$$

which Donoho and Johnstone call *universal.* This threshold is one of the first proposed and provides an easy, fast, and automatic thresholding. The rationale is to remove all wavelet coefficients that are smaller than the expected maximum of an assumed iid normal noise sequence of given size. There are several possibilities for the estimator $\hat{\sigma}$.

Almost all methods involve the wavelet coefficients of the finest scale. The signal-to-noise ratio is smallest at high resolutions in a wavelet decomposition for almost all reasonably behaved signals.

Some standard estimators are:

$$(i) \quad \hat{\sigma}^2 = \frac{1}{N/2-1} \Sigma_{i=1}^{N/2} (d_{n-1,i} - \bar{d})^2,$$

or a more robust

$$(ii) \quad \hat{\sigma}^2 = 1/0.6745 \quad MAD(\{d_{n-1,i}, i = 1, N/2\}),$$

where $n-1$ is the highest level.

In some problems, especially with (i) large data sets, and (ii) when the σ is over-estimated, the universal thresholding gives under-fitted models.

3.2 A threshold based on Stein's unbiased estimator of risk

Donoho and Johnstone (1994) developed a technique of selecting a threshold by minimizing Stein's unbiased estimator of risk.

Result: Let $x_i \overset{iid}{\sim} N(\mu_i, 1), i = 1, k$. Let $\hat{\underset{\sim}{\mu}}$ be an estimator of $\underset{\sim}{\mu} = (\mu_1, \ldots, \mu_k)$. If the function $\mathbf{g} = \{g_i\}_{i=1}^k$ in representation $\hat{\underset{\sim}{\mu}}(\underset{\sim}{x}) = \underset{\sim}{x} + \mathbf{g}(\underset{\sim}{x})$ is weakly differentiable, then

$$E^{\mu} ||\hat{\underset{\sim}{\mu}} - \underset{\sim}{\mu}||^2 = k + E^{\mu} ||\mathbf{g}(\underset{\sim}{x})||^2 + 2\nabla \mathbf{g}(\underset{\sim}{x}), \tag{15}$$

where $\nabla \mathbf{g} = \{\frac{\partial}{\partial x_i} g_i\}$. It is interesting that estimator $\hat{\underset{\sim}{\mu}}$ can be nearly arbitrary; for instance, biased and non-linear.

The application of (15) to $T^{soft}(\underset{\sim}{x}, \lambda)$ gives:

$$\text{SURE}(\underset{\sim}{x}, \lambda) = k - 2\Sigma_{i=1}^k \mathbf{1}(|x_i| \leq \lambda) + \Sigma_{i=1}^k (|x_i| \wedge \lambda)^2. \tag{16}$$

The SURE is an unbiased estimator of risk, i.e.,

$$E||T^{soft}(\underset{\sim}{x}, \lambda) - \underset{\sim}{\mu}||^2 = E\ \text{SURE}(\underset{\sim}{x}, \lambda).$$

The LLN argument motivates the following threshold selection:

$$\lambda^{sure} = arg \min_{0 \le \lambda \le \lambda^U} \text{SURE}(\underset{\sim}{x}, \lambda). \tag{17}$$

It is possible to derive a SURE-type threshold for T^{hard} and T^{hyper} but the simplicity of the representation (16) is lost.

3.3 Cross-validation

Nason (1994) proposed a very interesting cross-validatory threshold selection procedure. From the original noisy data set $y_i,\ i = 1, N(= 2^n)$, two subsequences are formed:

$$\bar{y}_i^{ODD} = \frac{y_{2i-1} + y_{2i+1}}{2}, i = 1, N/2; y_{N+1} = y_{N-1}, \tag{18}$$

and

$$\bar{y}_i^{EVEN} = \frac{y_{2i} + y_{2i+2}}{2}, i = 1, N/2; y_{N+2} = y_N. \tag{19}$$

The cross-validatory threshold λ^C is a minimizer of

$$\hat{M}(\lambda) = \Sigma_{j,k}(T^{soft}(d_{jk}^{EVEN}; \lambda) - d_{jk}^{ODD})^2 + \Sigma_{j,k}(T^{soft}(d_{jk}^{ODD}; \lambda) - d_{jk}^{EVEN})^2, \tag{20}$$

multiplied by the correction factor $(1 - \frac{\log 2}{\log N})^{-\frac{1}{2}}$, where d_{jk}^{ODD} and d_{jk}^{EVEN} are discrete wavelet transformations of the sequences $\bar{y}^{ODD}$ and $\bar{y}^{EVEN}$.

Nason (1994) showed that almost always one can find a unique minimizer of $\hat{M}(\lambda)$ and compared the performance of the cross-validatory threshold to the Donoho-Johnstone universal and SURE methods.

3.4 Other methods

At the expense of a slight increase of computational complexity (up to $O(n \log n)$), Donoho and Johnstone (1993) propose the *SUREShrink* method. The idea is to shrink wavelet coefficients level-wise. The SURE is used only if the level has a significant signal present. Otherwise universal thresholding is used. The proposed method has excellent smoothness adaptation properties. Wang (1994b) generalizes Nason's crossvalidation technique by removing more than half of the data each time. The motivation is to robustify the threshold selection procedure against the effect of a correlated noise (with a long range dependence). Saito (1994) incorporates the hard thresholding into a minimum description length procedure. Vidakovic (1994b) describes wavelet shrinkage via Bayes rules and Bayesian testing of hypothesis.

4 Example: California earthquakes

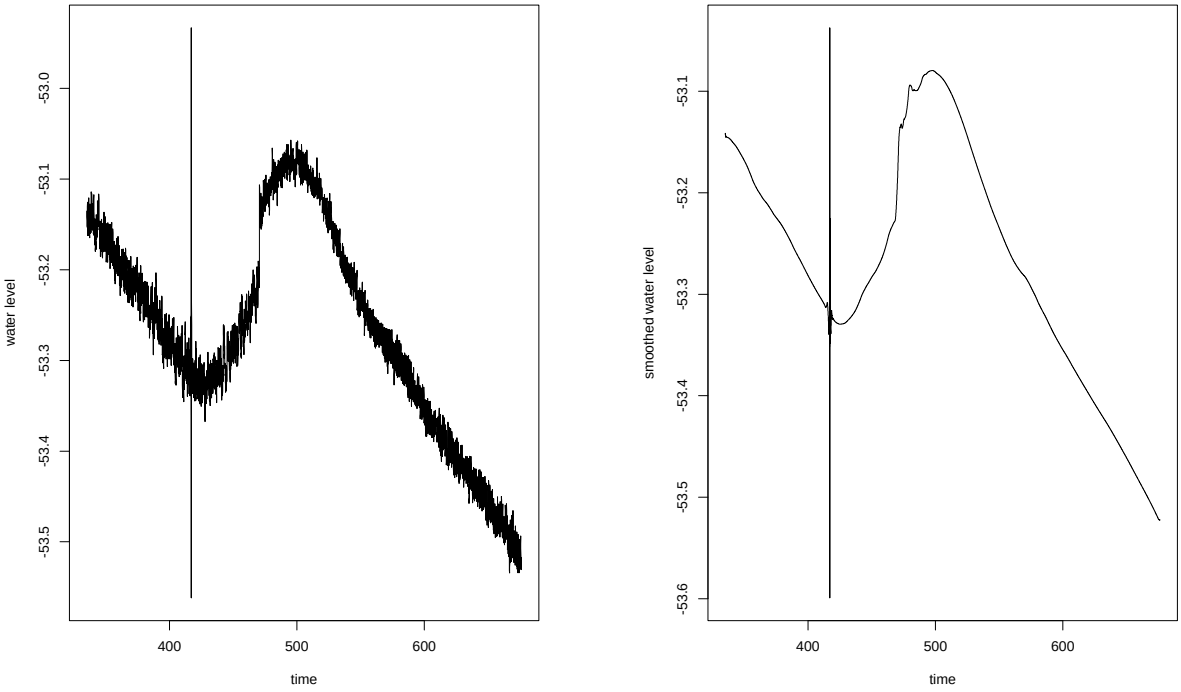
A researcher in geology was interested in predicting earthquakes by the level of water in nearby wells. She had a large ($8192 = 2^{13}$ measurements) data set of water levels taken every hour in a period of time of about one year in a California well. Here is the description of the problem.

> The ability of water wells to act as strain meters has been observed for centuries. The Chinese, for example, have records of water flowing from wells prior to earthquakes. Lab studies indicate that a seismic slip occurs along a fault prior to rupture. Recent work has attempted to quantify this response, in an effort to use water wells as sensitive indicators of volumetric strain. If this is possible, water wells could aid in earthquake prediction by sensing precursory earthquake strain.
>
> We have water level records from six wells in southern California, collected over a six year time span. At least 13 moderate size earthquakes (Magnitude 4.0 - 6.0) occurred in close proximity to the wells during this time interval. There is a a significant amount of noise in the water level record which must first be filtered out. Environmental factors such as earth tides and atmospheric pressure create noise with frequencies ranging from seasonal to semidiurnal. The amount of rainfall also affects the water level, as do surface loading, pumping, recharge (such as an increase in water level due to irrigation), and sonic booms, to name a few. Once the noise is subtracted from the signal, the record can be analyzed for changes in water level, either an increase or a decrease depending upon whether the aquifer is experiencing a tensile or compressional volume strain, just prior to an earthquake.

A plot of the raw data for hourly measurements over one year ($8192 = 2^{13}$ observations) is given in Figure 10a. After applying the DAUB #2 wavelet transformation and thresholding by the Donoho-Johnstone "universal" method, we got a very clear signal with big jumps at the earthquake time. The cleaned data are given in Figure 10b. The magnitude of the water level change at the earthquake time did not get distorted in contrast to usual smoothing techniques. This is a desirable feature of wavelet methods. Yet, a couple of things should be addressed with more care.

(i) Possible fluctuations important for the earthquake prediction are cleaned as noise. In post-analyzing the data, having information about the earthquake time, one might do time-sensitive thresholding.

(ii) Small spikes on the smoothed signal (Figure 10b) as well as 'boundary distortions" indicate that the DAUB2 wavelet is not the most fortunate choice. Compromising between smoothness and the support shortness of the mother wavelet with help of wavelet banks, one can develop ad-hoc rules for better mother wavelet (wavelet model) choice.



(a) Raw data, water level vs. time

(b) After thresholding the wavelet coefficients

Figure 10: Panel (a) shows $n = 8192$ hourly measurements of the water level for a well in an earthquake zone. Notice the wide range of water levels at the time of an earthquake around $t = 415$.

5 Wavelet image processing

We will explain briefly how wavelets may be useful in the matrix data processing. The most remarkable application is, without any doubt, image processing. Any (black and white) image can be approximated by a matrix A in which the entries a_{ij} correspond to intensities of gray in the pixel (i, j). For reasons that will be obvious later, it is assumed that A is the square matrix of dimension $2^n \times 2^n$, n integer.

The process of the image wavelet decomposition goes as follows. On the rows of the matrix A the filters H and G are applied. Two resulting matrices are obtained: H_rA and G_rA, both of dimension $2^n \times 2^{n-1}$ (Subscript r suggest that the filters are applied on rows of the matrix A). Now on the columns of matrices H_rA and G_rA, filters H and G are applied again and the four resulting matrices $H_cH_rA, G_cH_rA, H_cG_rA$ and G_cG_rA of dimension $2^{n-1} \times 2^{n-1}$ are obtained. The matrix H_cH_rA is the average, while the matrices G_cH_rA, H_cG_rA and G_cG_rA are details (Figure 11)

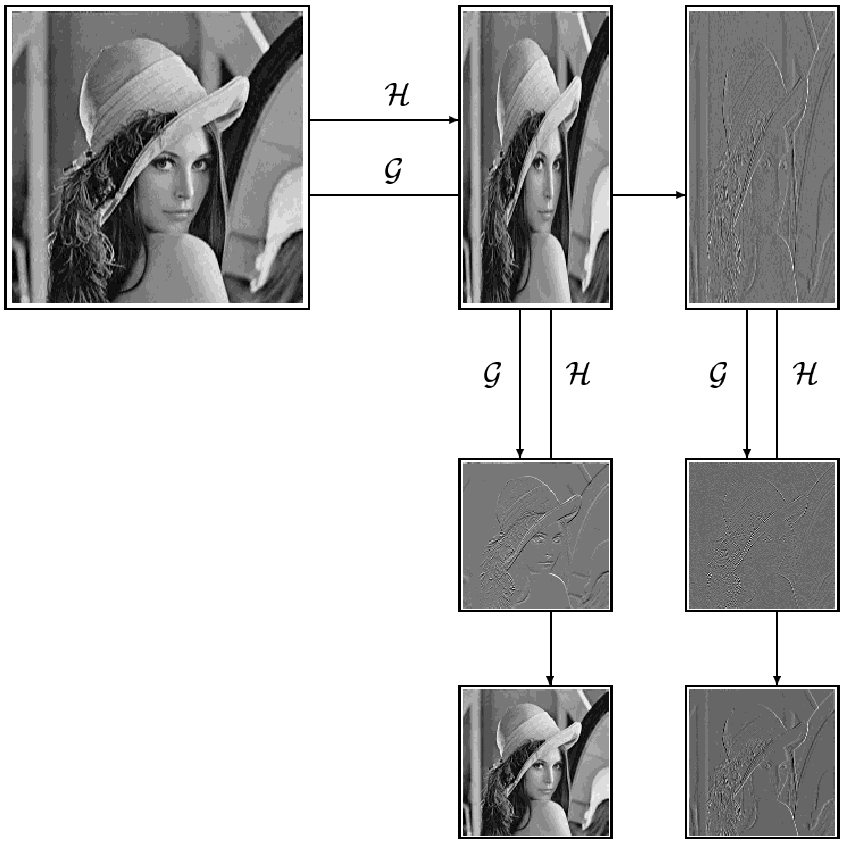


Figure 11: Lenna Image wavelet decomposition

The process can be continued with the *average* matrix $H_c H_r A$ until a single number (" an average" of the whole original matrix A) is obtained. Two examples are given below.

Example 1.

This example is borrowed from Nason and Silverman (1993). The top left panel in Figure 12 is 256×256 black and white image of John Lennon in 0-255 gray scale.

In the top-right figure each pixel is contaminated by normal $N(0, 60)$ noise. (In *Splus*: `le` $\leftarrow$ `lennon+rnorm(256*256, s=60)` where `lennon` is the pixel matrix of the original image.)

The two bottom figures are restored images. The DAUB #4 filter was used for

the first figure, while DAUB #10 was used for the second.

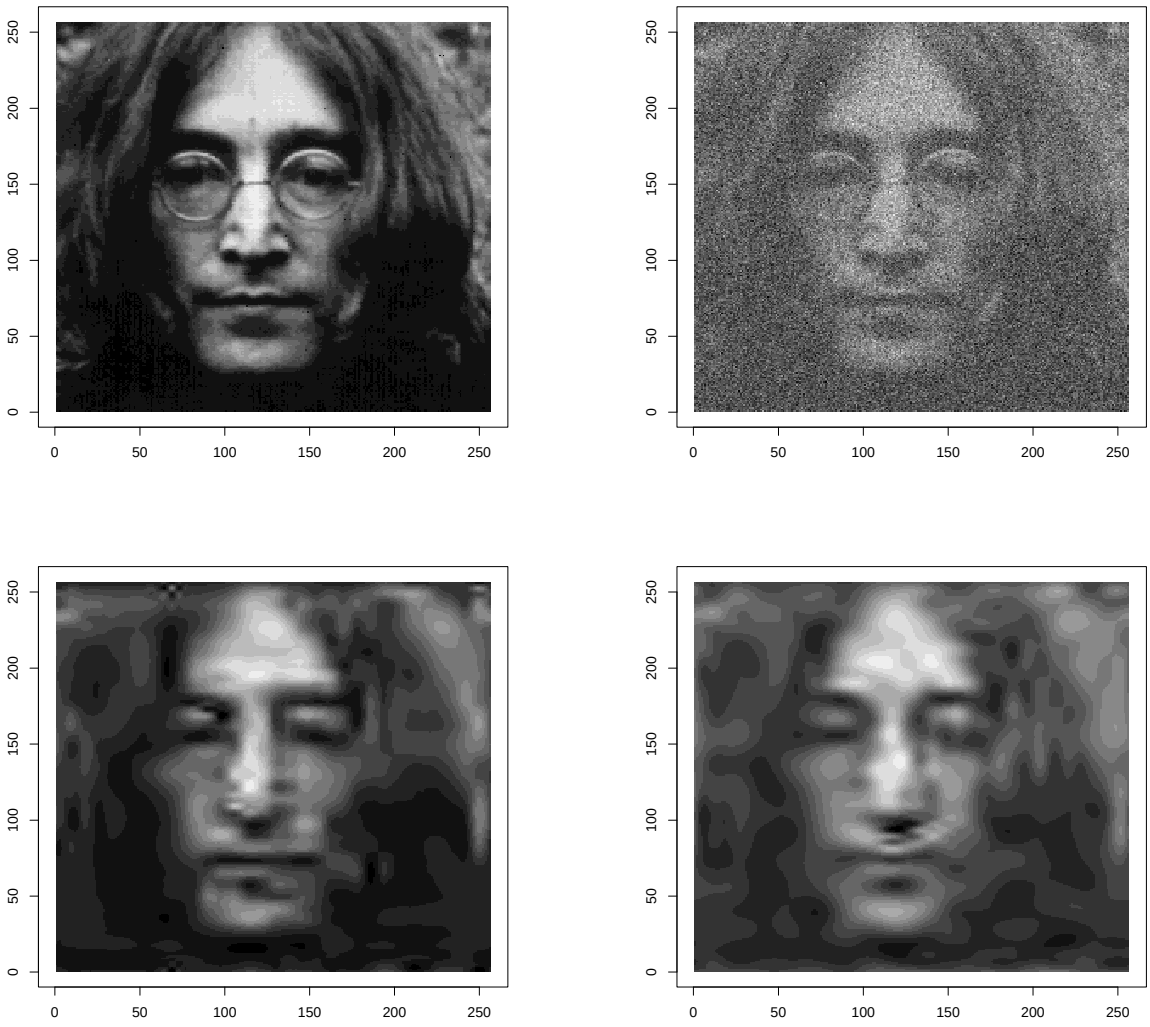


Figure 12: Wavelet image restoration example

Though the quality of the restored images may be criticized, the stunning property of wavelet image analysis shows up in this example. Both restored images use only about 1.8 % of the information contained in the "blurred" image. The compression rate is amazing: 527120 bites go to 9695 bites after the universal thresholding.

Example 2.

This is an adaptation of the data set of J. Schmert, University of Washington. The word **five** was recorded and each column on the top-right figure represents a periodogram over a short period of time (adjacent columns have half of their observations

in common). The rows represent time. The original 92 $\times$ 64 matrix was cut to 64 $\times$ 64 matrix for obvious reasons. After performing hard thresholding with $\lambda = 0.25$, a compression ratio of 1:2 is achieved. The compressed figures are shown in the two bottom panels of Figure 13.

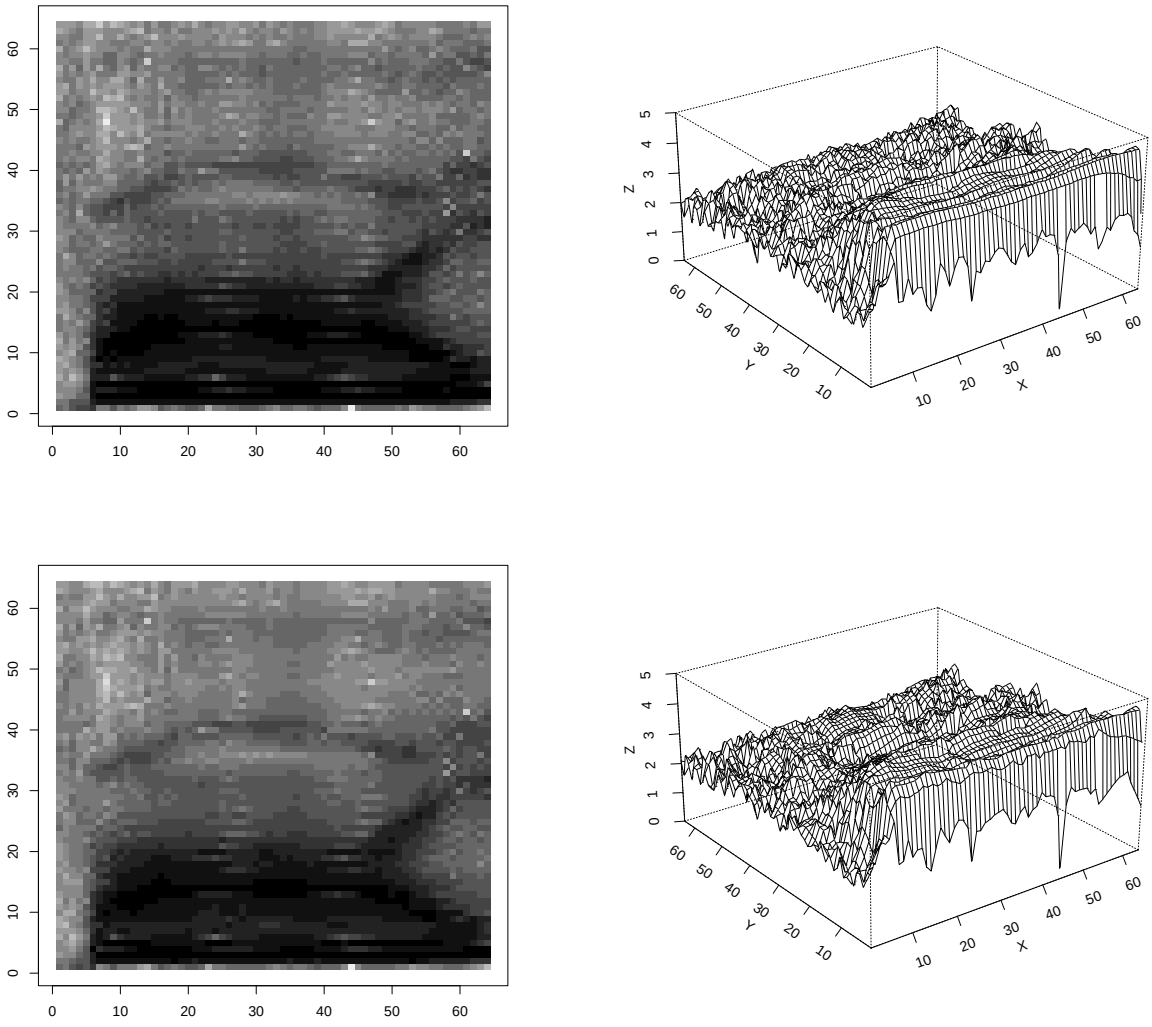


Figure 13: Word **FIVE** data. The panels in the first row show to the original data. The bottom panels show the signal after thresholding.

6 Can you do wavelets?

Yes, you can! There are several several packages that support wavelet calculations. An S-based non-commercial package is Nason and Silverman's: *The Discrete Wavelet Transform in S.* The manual [19] describes installation and use of Nason's software. The software is free and can be **ftp**ed[7] from `lib.stat.cmu.edu` or `hensa.unix.ac.uk`. The name of the package is `wavethresh`.

`WaveLab` package by Donoho and coauthors (http://playfair.Stanford.EDU:80/w̃avelab/). is a free MATLAB-based software that is very comprehensive.

Carl Taswell (`taswell@sccm.stanford.edu`) developed Wavelet Toolbox for MATLAB. The latest version is WavBox 4.0 and the software has to be registered with the author. Some other MATLAB based software are: Matlab toolbox for W-Matrix Multiresolution Analyses, by M.K. Kwong (kwong@mcs.anl.gov). The Rice Wavelet Tools are a Matlab toolbox for filter bank and wavelet design and analysis. It was developed by the DSP group at Rice University (wlet-tools@rice.edu).

Some C-based packages are:

XWPL is an X based tool to examine one-dimensional real-valued signals using wavelets and wavelet packets. It was developed by Fazal Majid (majid@math.yale.edu).

The Imager Wavelet Library (wvlt) is a small set of routines that allow the user to manipulate wavelets. It was developed by Bob Lewis (bobl@cs.ubc.ca). The Multigrid Algorithm Library of the Hamburg Multigrid Group.

There are several MATHEMATICA notebooks on wavelet omcputations. V. Wickerhauser, Jack Cohen, (`jkc@keller.mines.colorado.edu`), made theirs available to the public.

To understand how the wavelets work, we reinvented the wheel and developed a MATHEMATICA program for direct and inverse wavelet transformation and thresholding and applied it to some exemplary data sets. The algorithms are far from being effective; rather they are educational. A MATHEMATICA notebook with worked examples is available via ftp anonymous at `isds.duke.edu` in `/pub/brani/papers`.

References

[1] BARRY A. C. (1993). Wavelet applications come to the fore, *SIAM News, November 1993.*

[2] COIFMAN, R., MEYER, Y., and WICKERHAUSER, V. (1991) Wavelet analysis and signal processing. In: *Wavelets and Their Applications,* Edited by Mary Beth Ruskai, Jones and Bartlet Publishers.

[3] DAUBECHIES, I. (1988), Orthonormal bases of compactly supported wavelets. Commun. Pure Appl. Math., 41 (7), 909-996.

[7] A new verb, ha!

[4] DAUBECHIES, I. (1992), *Ten Lectures on Wavelets*, Society for Industrial and Applied Mathematics.

[5] DEVORE, R. and LUCIER, B. J. (1991). Wavelets. *Acta Numerica* **1** *1-56.*

[6] DONOHO, D. (1992). Wavelet shrinkage and WVD: A 10-minute tour. Presented on the International Conference on Wavelets and Applications, Tolouse, France, June 1992.

[7] DONOHO, D. (1993). Nonlinear Wavelet Methods for Recovery of Signals, Densities, and Spectra from Indirect and Noisy Data. *Proceedings of Symposia in Applied Mathematics,* American Mathematical Society.

[8] DONOHO, D. and JOHNSTONE, I. (1992). Minimax estimation via wavelet shrinkage. Technical Report, Department of Statistics, Stanford University.

[9] DONOHO, D. and JOHNSTONE, I. (1993a) Ideal spatial adaptation by wavelet shrinkage. *Biometrika.* to appear.

[10] DONOHO D. , and JOHNSTONE, I. (1993b). Adapting to unknown smoothness via wavelet shrinkage. *J. Am. Stat. Ass.,* to appear.

[11] DONOHO, D., JOHNSTONE, I., KERKYACHARIAN, G, and PICARD, D. (1993a). Density estimation by wavelet thresholding. Technical Report, Department of Statistics, Stanford University.

[12] DONOHO, D., JOHNSTONE, I., KERKYACHARIAN, G, and PICARD, D. (1993b). Wavelet shrinkage: Asymptopia? *J. R. Statis. Soc.* to appear.

[13] GAO, H-Y. (1993). Choice of thresholds for wavelet estimation of the log-spectrum. *Tech. Report,* Statistics, Stanford University.

[14] GAO, H-Y. (1993). Spectral density estimation via wavelet shrinkage. *Tech. Report,* Statistics, Stanford University.

[15] GROSSMANN, A. and MORLET, J. (1984). Decomposition of Hardy functions into square integrable wavelets of constant shape. SIAM J. Math., 15, 723-736.

[16] JOHNSTONE, I. (1993). Minimax-Bayes, asymptotic minimax and sparse wavelet priors. Technical Report, Department of Statistics, Stanford University.

[17] MALLAT, S. G. (1989). A theory for multiresolution signal decomposition: the wavelet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11 (7), 674-693.

[18] NASON, G. (1994). Wavelet regression by cross-validation. *Technical Report 447.* Department of Statistics, Stanford University.

[19] NASON, G. P. and SILVERMAN B. W. (1993). The discrete wavelet transform in S, Statistics Research Report 93:07, University of Bath, Bath, BA2 7AY , UK.

[20] PRESS W. H., FLANNERY, B. P., TEUKOLSKY, S. A., and VETTERLING, W. T. (1993). *Numerical Recipes in C.* Second Edition, Cambridge University Press.

[21] SAITO N. (1994). Simultaneous noise suppression and signal compression using a library of orthonormal bases and the minimum description length criterion. In: *Wavelets in Geophysics*, Foufoula-Georgiou and Kumar (eds.), Academic Press.

[22] STRANG, G. (1993). Wavelet transforms versus Fourier transforms, *BAMS,* **28***, 288-305.*

[23] VIDAKOVIC, B. (1994a). Random densities via wavelets. *Discussion Paper* **94-06.** ISDS, Duke University. Submitted.

[24] VIDAKOVIC, B. (1994b). Nonlianer wavelet shrinkage via Bayes rules and Bayes factor, *Discussion Paper* **94-24.** ISDS, Duke University. Submitted.

[25] WANG, Y. (1994). Jump and sharp cusp detection by wavelets - One dimensional case. *Tech. Report.* Department of Statistics, University of Missouri-Columbia.

[26] WANG, Y. (1994). Function estimation via wavelets for data with long-range dependence. *Tech. Report.* Department of Statistics, University of Missouri-Columbia.

[27] WANG, Z. (1993). Estimating a Holder Continuous Function from a Noisy Sample via Shrinkage and Truncation of Wavelet Coefficients. Technical Report **93-9**, Purdue University, Department of Statistics.

[28] *Wavelets and Their Applications*, Edited by Mary Beth Ruskai, Jones and Bartlett Publishers. (1991).

[29] *Wavelets: A Tutorial in Theory and Applications*, Edited by Charles K. Chui, Academic Press, Inc. (1992)

7 Appendix

```
BeginPackage["Waves`"]
(* Author: Brani Vidakovic, ISDS, Duke University ;
Functions Dec and Comp are based on M. V. Wickerhauser's
mathematica program; December 1994 *)

Mirror::usage = "Mirror[_filter_] gives the mirror  \
```

```
filter for the input _filter_. This is an adjoint  \
operator H* of the operator H corresponding to  _filter_."

WT::usage = "WT[_vector_,  _filter_]  performs the direct \
 wavelet transformation of the data vector  _vector_. \
The  wavelet  base is chosen by _filter_. The length \
of the vector  _vector_  has to be a degree of 2."

WR::usage = "WR[_vector_,  _filter_] gives the wavelet \
reconstruction algorithm. From the set of  wavelet \
coefficients _vector_ the data set is reconstructed. \
The wavelet base is chosen by _filter_."

Dec::usage = "An auxiliary function needed for the \
direct wavelet transformation. See WT."

Comp::usage = "An auxiliary function needed for the \
inverse wavelet transformation (wavelet reconstruction  \
algorithm). See WR."

 Begin["`Private`"]


Mirror[ filter_List]:= Module[{fl=Length[filter]},
Table[ -(-1)^i filter[[fl+1-i]], {i, 1, fl}]];


Dec[ vector_List, filter_List]:=  Module[
 {vl= Length[vector], fl=Length[filter]},
 Table[
      Sum[ filter[[m]] vector[[Mod[2 k+m - 3, vl]+1 ]],
         {m,1,fl}],
      {k,1,vl/2}]
 ];


Comp[ vector_List, filter_List]:= Module[
 { temp=Table[0,{i,1,2 Length[vector]}],
 vl=Length[vector], fl=Length[filter]},
 Do[ temp[[ Mod[2 j + i -3, 2 vl]+1]] +=
        vector[[j]] filter[[i]],
      {j, 1, vl}, {i, 1, fl}];
```

```
  temp];


WT[ vector_List, filter_List]:=
Module[ { wav={}, c,d, ve=vector, H=filter,
G=Mirror[filter]},
While[ Length[ve] > 1,
     lev=Log[2,Length[ve]]-1;
     c = Dec[ve, H];
     d = Dec[ve, G];
     wav= Join[ wav, d ];
     ve = c]; Join[wav, c] ];

WR[ vector_List, filter_List]:=
Module[ {i=1, vl=Length[vector], c=Take[vector,-1],
        d=Take[RotateRight[vector,1],-1],
        mirrorf=Mirror[filter], cn, dn, k=1},
While[ i <= vl/2 ,
       k += i;
       i= 2 i;
       cn=Comp[c, filter]+Comp[d, mirrorf];
       dn=Take[RotateRight[vector, k], -i ];
       c=cn;
       d=dn;
     ];
c  ];

End[ ]

EndPackage[ ]
```

Institute of Statistics
and Decision Sciences
Duke University
Durham, NC 27708-0251
brani@isds.duke.edu
pm@isds.duke.edu