An Introduction to Wavelets and some Applications

Milan, May 2003

Anestis Antoniadis

Laboratoire IMAG-LMC
University Joseph Fourier
Grenoble, France

Outline

- **General facts on wavelet decompositions**

Outline

- **General facts on wavelet decompositions**
- Multiresolution analyses

Outline

- **General facts on wavelet decompositions**
- Multiresolution analyses
- Computational algorithms

Outline

- **General facts on wavelet decompositions**
- Multiresolution analyses
- Computational algorithms
- **Approximation and Compression**

Outline

- **General facts on wavelet decompositions**
- Multiresolution analyses
- Computational algorithms
- **Approximation and Compression**
- Linear

Outline

- **General facts on wavelet decompositions**
- Multiresolution analyses
- Computational algorithms
- **Approximation and Compression**
- Linear
- Nonlinear

Outline

- **General facts on wavelet decompositions**
- Multiresolution analyses
- Computational algorithms
- **Approximation and Compression**
- Linear
- Nonlinear
- **Estimation and Denoising**

Outline

- **General facts on wavelet decompositions**
- Multiresolution analyses
- Computational algorithms
- **Approximation and Compression**
- Linear
- Nonlinear
- **Estimation and Denoising**
- Regularization

Outline

- **General facts on wavelet decompositions**
- Multiresolution analyses
- Computational algorithms
- **Approximation and Compression**
- Linear
- Nonlinear
- **Estimation and Denoising**
- Regularization
- Asymptotics and Applications

Generalities

- Interest in representing an "observed" signal as an appropriate superposition of "elementary" functions in an adapted way.

Generalities

- Interest in representing an "observed" signal as an appropriate superposition of "elementary" functions in an adapted way.
- To obtain such representations that can be useful in practice, one needs fast computational algorithms.

Generalities

- Interest in representing an "observed" signal as an appropriate superposition of "elementary" functions in an adapted way.
- To obtain such representations that can be useful in practice, one needs fast computational algorithms.
- Once such representations are derived, one would like to simplify them in a efficient way by choosing appropriately only few elementary components. This may be seen as an *approximation* or *compression* task.

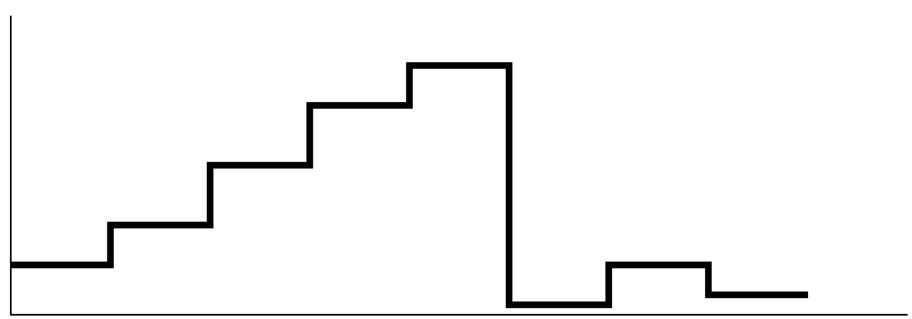
The Haar basis

The Haar basis is the simplest example of a wavelet basis. It allows us to introduce in a clear and simple way the wavelet idea, without going to far into mathematical details.
Suppose we are given a discretized version of an integrable function f on $[0, 1]$ on an equidistant grid of 8 values.

The Haar basis

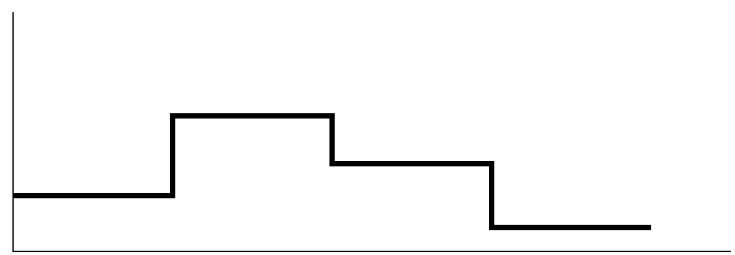
The Haar basis is the simplest example of a wavelet basis. It allows us to introduce in a clear and simple way the wavelet idea, without going to far into mathematical details.
Suppose we are given a discretized version of an integrable function f on $[0,1]$ on an equidistant grid of 8 values. Let the discrete signal be:

$$[2\ \ 4\ \ 8\ \ 12\ \ 14\ \ 0\ \ 2\ \ 1]$$



A digital signal on $[0,1]$

We could represent the above digital signal in a different way in order to exploit a possible correlation between adjacent points. To this end, for every pair of neighbors we compute their averages to obtain: $[3 \ \ 10 \ \ 7 \ \ 1.5]$

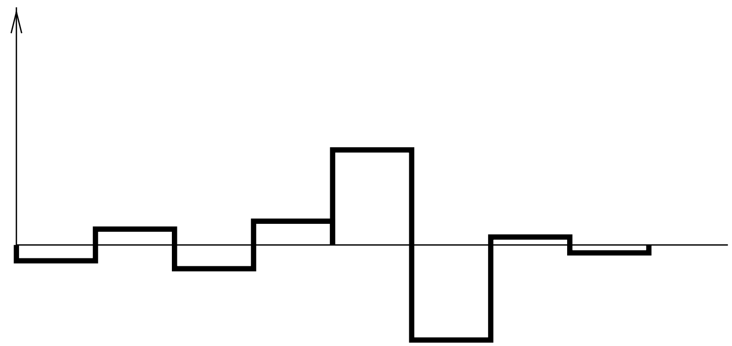


Averages of neighbor values

To avoid any loss of information we also need to record some other values that represent the loss of information when going from the finer grid to the coarser.

We choose

$$[-1 \quad -2 \;\; 7 \;\; 0.5]$$

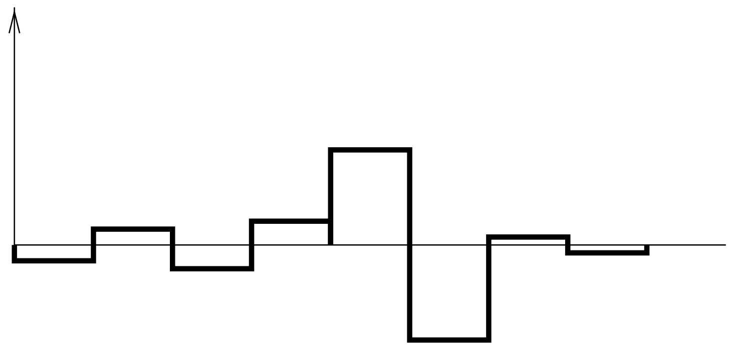


Differences of Input and its approximation

Indeed, 3+(-1)=2, 3-(-1)=4, 10+(-2)=8, . . .

We choose

$$[-1 \quad -2 \;\; 7 \;\; 0.5]$$



Differences of Input and its approximation

Indeed, 3+(-1)=2, 3-(-1)=4, 10+(-2)=8, ...

input signal = signal with lower resolution (4 values) and a quadruple of differences (the details).

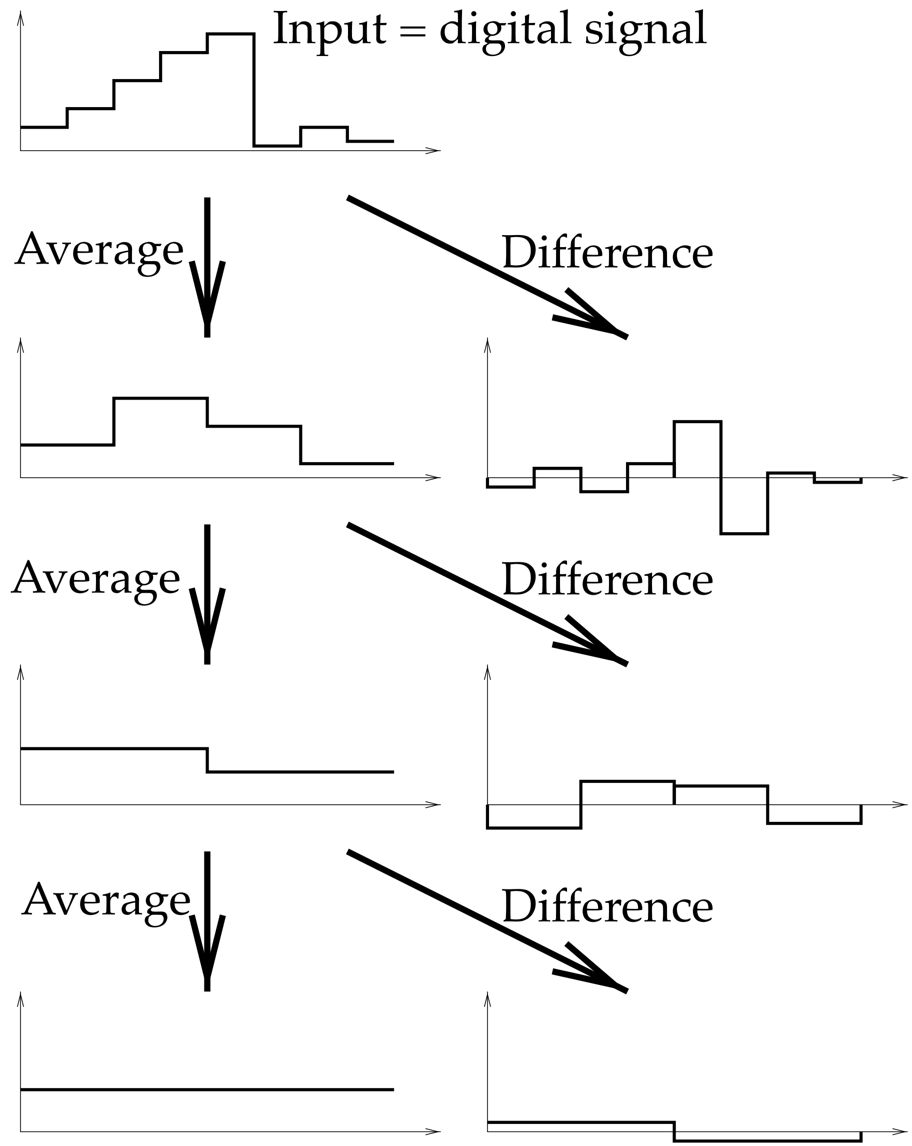
The Haar transform

We can repeat the procedure over the averages again and again to obtain:

Resolution	Averages	Details
8	[2 4 8 12 14 0 2 1]	
4	[3 10 7 1.5]	[-1 -2 7 0.5]
2	[6.5 4.25]	[-3.5 2.75]
1	[5.375]	[1.125]

thus representing the input as:

$$[5.3751.125 - 3.52.75 - 1 - 270.5]$$



The Haar transform

The digital input may be considered as a piecewise constant function on $[0,1]$ on the intervals $I_{3,k} = [2^{-3}k, 2^{-3}(k+1)[$, $k = 0, \ldots, 2^3 - 1$. If $\phi(x) = \mathbf{I}_{[0,1[}(x)$ and $\phi_{j,k}(x) = \phi(2^j x - k)$, the function may be written as

$$
\begin{aligned}
f(x) &= 2\phi_{3,0}(x) + 4\phi_{3,1}(x) + 8\phi_{3,2}(x) + 12\phi_{3,3}(x) + \cdots \\
&\quad 14\phi_{3,4}(x) + 0\phi_{3,5}(x) + 2\phi_{3,6}(x) + 1\phi_{3,7}(x).
\end{aligned}
$$

We then may re-write:

$$
\begin{aligned}
f(x) &= 3\phi_{2,0}(x) + 10\phi_{2,1}(x) + 7\phi_{2,2}(x) + 1.5\phi_{2,3}(x) + \cdots \\
&\quad (-1)\psi_{2,0}(x) + (-2)\psi_{2,1}(x) + 7\psi_{2,2}(x) + 0.5\psi_{2,3}(x),
\end{aligned}
$$

where $\psi(x) = \mathbf{I}_{[0,1/2[}(x) - \mathbf{I}_{[1/2,1[}(x)$.

- V_0 subspace of $L^2([0,1[)$ spanned by the constant functions on $[0,1[$.

- V_0 subspace of $L^2([0,1[)$ spanned by the constant functions on $[0,1[$.
- V_1 subspace spanned by the piecewise constant functions on $[0,1/2[$ and $[1/2,1[$

- V_0 subspace of $L^2([0,1[)$ spanned by the constant functions on $[0,1[$.
- V_1 subspace spanned by the piecewise constant functions on $[0,1/2[$ and $[1/2,1[$
- V_j subspace spanned by the piecewise constant functions on the intervals $I_{j,k}$, $k=0,2^j-1$.

- V_0 subspace of $L^2([0,1[)$ spanned by the constant functions on $[0,1[$.
- V_1 subspace spanned by the piecewise constant functions on $[0,1/2[$ and $[1/2,1[$
- V_j subspace spanned by the piecewise constant functions on the intervals $I_{j,k}$, $k = 0, 2^j - 1$.

They are nested. Moreover, for each of them, the families $\{\phi_{j,k},\ k = 0, \ldots, 2^j - 1\}$ form a basis. For the usual inner product $\langle f, g\rangle = \int_0^1 f(x)\bar{g}(x)dx$, these families are orthogonal and $\{\psi_{j,k}, k = 0, \ldots, 2^j - 1\}$ is a basis of the vector space W_j, orthogonal complement of V_j in V_{j+1}.

Multiresolution analysis on the $\mathbb{R}$

A *multiresolution analysis* of $L^2(\mathbb{R})$ is a nested sequence of closed subspaces V_j, $j \in \mathbb{Z}$, of $L^2(\mathbb{R})$,

$$\cdots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots,$$

such that

$$\cap_j V_j = \{0\}, \quad \overline{\cup_j V_j} = L^2(\mathbb{R}), \quad f(x) \in V_j \Leftrightarrow f(2x) \in V_{j+1},$$

There exists a function $\phi \in V_0$ such that

$$V_0 = \left\{ f \in L^2(\mathbb{R}) : \; f(x) = \sum_{k \in \mathbb{Z}} \alpha_k \phi(x-k) \right\},$$

$\{\phi_k, k \in \mathbb{Z}\}$ is a "stable" basis of V_0, i.e. $0 < m \leq \|\phi_k\| \leq M < \infty$ and $A\|f\|^2 \leq \sum_k \alpha_k^2 \leq B\|f\|^2$.

The function ϕ is called the *scaling function* of the MRA. Let $\phi_{j,k}(x) = 2^{j/2}\phi(2^j x - k)$.
For an orthogonal MRA, an orthonormal basis of V_j is $\{\phi_{j,k} : k \in \mathbb{Z}\}$ and

$$P_j f = \sum_{k \in \mathbb{Z}} < f, \phi_{j,k} > \phi_{j,k} \, ,$$

is the approximation of f at resolution 2^{-j}.
If W_j is the orthogonal complement of V_j in V_{j+1}, we obtain another sequence $\{W_j : j \in \mathbb{Z}\}$ of closed orthogonal subspaces of $L^2(\mathbb{R})$, such that each W_j is a refinement of W_0, and their direct sum is $L^2(\mathbb{R})$.

One can show that there exists a function ψ such that W_0 is spanned by its integer translates. Then ψ is called the wavelet associated to φ. For each integer j,the family

$$\{\psi_{j,k}, k \in \mathbb{Z}\}$$

is an orthonormal basis of W_j.

If $g \in L^2(\mathbb{R})$ we have:

$$g = \sum_{k\in\mathbb{Z}} c_{j_0,k}\varphi_{j_0,k} + \sum_{j\geq j_0}\sum_{k\in\mathbb{Z}} d_{j,k}\psi_{j,k}$$

where j_0 is a level of coarse approximation.

The first part in the right hand side is the orthogonal projection $P_{j_0}g$ of g on V_{j_0}, and the second part represents the details. The coefficients are defined by

$$c_{j,k} = \langle g, \phi_{j,k} \rangle$$

and

$$d_{j,k} = \langle g, \psi_{j,k} \rangle.$$

The c's are called the scaling coefficients while the d's are the wavelet coefficients.

Filter banks

Since $V_0 \subset V_1$, any function of V_0 has an expansion in terms of the basis $\{\phi_{1,k}, k \in \mathbb{Z}\}$ of V_1. For $\phi(x) = \phi_0(x) = \phi_{0,0}(x)$ we have

$$\phi(x) = \sum_{k\in\mathbb{Z}} a_k \phi_{1,k}(x) = \sqrt{2} \sum_{k\in\mathbb{Z}} a_k \phi(2x-k)$$

with

$$a_k = \langle \phi, \phi_{1,k} \rangle \in \ell^2(\mathbb{Z}).$$

When the scaling functions are compactly supported, there is only a finite number of non zero coefficients among the a_k's and

$$\phi(x) = \sqrt{2} \sum_{k=0}^{D-1} a_k \phi(2x - k)$$

The coefficients $a = \{a_k\}$ define the filter corresponding to ϕ.

The scaling function is compactly supported if and only if a has a finite number of non zero coefficients.

By analogy, and since $W_0 \subset V_1$ we also have

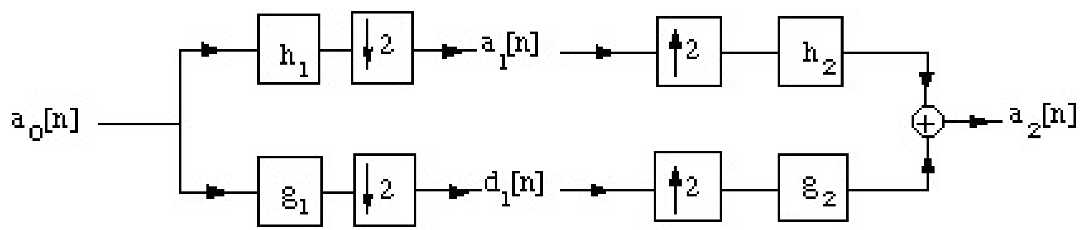
$$\psi(x) = \sqrt{2} \sum_{k=0}^{D-1} b_k \phi(2x - k)$$

with

$$b_k = \langle \psi, \phi_{1,k} \rangle \in \ell^2(\mathbb{Z}).$$

The filters $\{a_k\}$ and $\{b_k\}$ are conjugate mirror filter banks.

A perfect reconstruction filter bank decomposes a signal by filtering and subsampling. It reconstructs it by inserting zeroes, filtering and summation.



The filter bank is said to be a perfect reconstruction filter bank when $a_2 = a_0$. If, additionally, $h = h_2$ and $g = g_2$, the filters are called conjugate mirror filters.

Numerical evaluation

No analytic formulas for evaluating numerically ϕ and ψ.

For compactly supported wavelets we have two algorithms :

- An iterative algorithm (Daubechies and Lagarias cascade algorithm).
- An iterative method for solving the dilatation equations

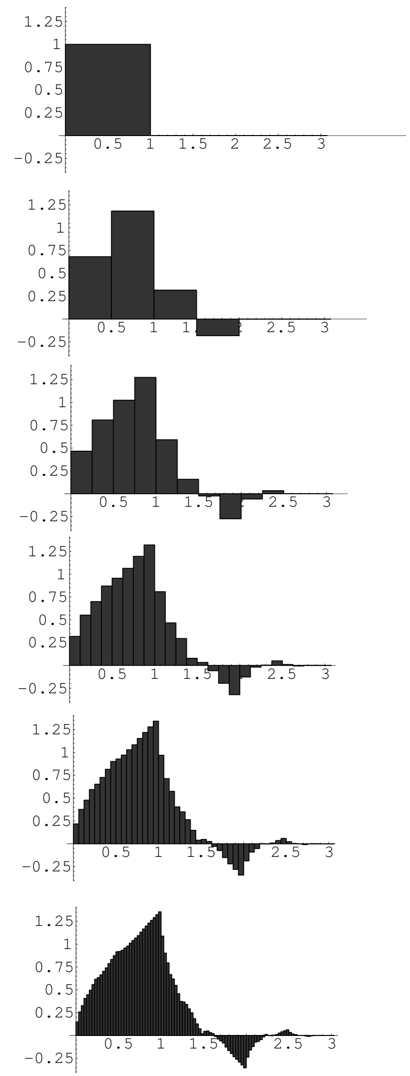
The Cascade Algorithm

Build ϕ using the filter $a_0, \ldots, a_{D-1}$ with D even and ≥ 4. Start with $\phi_0(x) = \mathbf{I}_{[0,1[}(x)$. Compute then recursively the successive approximations of $\phi(x)$ using $a_0, \ldots, a_{D-1}$, i.e.

$$\phi_m(x) = \sum_{k=0}^{D-1} a_k \phi_{m-1}(2x - k).$$

One can show that this algorithm converges towards ϕ, when the iterations m converge to ∞. In practice, 8 iterations suffice for a good discretization of ϕ.

Example: ϕ Daubechies $D = 4$



Building ϕ by the cascade algorithm

Periodic wavelets

Until now the functions were defined on $\mathbb{R}$.
While this seems reasonable for some applications, in practice most functions are observed over bounded domains .

There exist many ways to define a MRA adapted to a bounded domain.

One of these, the most simple and direct way, is by using *periodic wavelets*.

Let the scaling function $\phi \in L^2(\mathbb{R})$ and the associated wavelet $\psi \in L^2(\mathbb{R})$ be given.

For all $j, l \in \mathbb{Z}$, we define the periodic scaling function of period 1, by

$$\tilde{\phi}_{j,l}(x) = \sum_{n=-\infty}^{+\infty} \phi_{j,l}(x+n)$$

and the associated periodic wavelet by

$$\tilde{\psi}_{j,l}(x) = \sum_{n=-\infty}^{+\infty} \psi_{j,l}(x+n).$$

Note that, for $j \leq 0$ and $l \in \mathbb{Z}$,

$$\tilde{\phi}_{j,l}(x) = 2^{j/2} \sum_n \phi(2^j(x + n - 2^{-j}l)) = \tilde{\phi}_{j,0}(x)$$

Therefore

$$j \leq 0 \quad \tilde{\phi}_{j,l}(x) = 2^{-j/2}.$$

Similarly we can show that

$$\forall j \leq -1, \quad \tilde{\psi}_{j,l} = 0.$$

For ϕ and ψ generated with a filter of length D:

- $\tilde{\phi}_{j,l}(x) = 2^{-j/2}$ for all $j \leq 0$ and all $l \in \mathbb{Z}$.

For ϕ and ψ generated with a filter of length D:

- $\tilde{\phi}_{j,l}(x) = 2^{-j/2}$ for all $j \leq 0$ and all $l \in \mathbb{Z}$.
- $\tilde{\psi}_{j,l}(x) = 0$ for all $j \leq -1$ and all $l \in \mathbb{Z}$.

For ϕ and ψ generated with a filter of length D:

- $\tilde{\phi}_{j,l}(x) = 2^{-j/2}$ for all $j \leq 0$ and all $l \in \mathbb{Z}$.
- $\tilde{\psi}_{j,l}(x) = 0$ for all $j \leq -1$ and all $l \in \mathbb{Z}$.
- for all $j > J_0 \geq \log_2(D-1)$ and $x \in [0,1]$,

$$\tilde{\phi}_{j,l}(x) = \phi_{j,l}(x)\mathbf{I}_{I_{j,l}}(x) + \phi_{j,l}(x+1)\mathbf{I}_{I^c_{j,l}}(x)$$

with an identical relation for $\tilde{\psi}$.

For ϕ and ψ generated with a filter of length D:

- $\tilde{\phi}_{j,l}(x) = 2^{-j/2}$ for all $j \leq 0$ and all $l \in \mathbb{Z}$.
- $\tilde{\psi}_{j,l}(x) = 0$ for all $j \leq -1$ and all $l \in \mathbb{Z}$.
- for all $j > J_0 \geq \log_2(D-1)$ and $x \in [0,1]$,

$$\tilde{\phi}_{j,l}(x) = \phi_{j,l}(x)\mathbf{I}_{I_{j,l}}(x) + \phi_{j,l}(x+1)\mathbf{I}_{I^c_{j,l}}(x)$$

with an identical relation for $\tilde{\psi}$.

$$\int_0^1 f(x)\tilde{\phi}_{j,l}(x)dx = \int_{-\infty}^{\infty} \tilde{f}(x)\phi_{j,l}(x)dx$$

(similarly for ψ) where $\tilde{f}(x) = f(x-[x])$, $x \in \mathbb{R}$.

The periodic scaling functions and corresponding periodic wavelets define an orthonormal MRA of $L^2([0,1])$.

The approximation and details spaces are given by:

$$\tilde{V}_j = \text{span}\left\{\tilde{\phi}_{j,l}, l = 0, \ldots, 2^j - 1\right\}$$

and

$$\tilde{W}_j = \text{span}\left\{\tilde{\psi}_{j,l}, l = 0, \ldots, 2^j - 1\right\}$$

We obtain, for $J_0 \geq 0$ the following decomposition of $L^2([0,1])$:

$$L^2([0,1]) = \tilde{V}_{J_0} \oplus \left(\oplus_{j \geq J_0} \tilde{W}_j\right).$$

Fast discrete wavelet transform

Orthogonality of the scaling functions and the associated wavelets lead to a fast computational algorithm for decomposing or synthetizing a function of V_j.

Let $f \in L^2(\mathbb{R})$. We have

$$(P_{V_j}f)(x) = (P_{V_{j-1}}f)(x) + (P_{W_{j-1}}f)(x),$$

formulated as

$$(P_{V_j}f)(x) = \sum_{\ell \in \mathbb{Z}} c_{j-1,\ell}\phi_{j-1,\ell}(x) + \sum_{\ell \in \mathbb{Z}} d_{j-1,\ell}\psi_{j-1,\ell}(x).$$

Aim: find a relation between the sequence of coefficients $c_{j,\ell}$ and the sequences $c_{j-1,\ell}$ and $d_{j-1,\ell}$.

The key is

$$\phi_{j-1,\ell}(x) = \sum_{k=0}^{D-1} a_k \phi_{j,2\ell+k}(x) \text{ and } \psi_{j-1,\ell}(x) = \sum_{k=0}^{D-1} b_k \phi_{j,2\ell+k}(x)$$

We have

$$\begin{aligned} c_{j-1,\ell} &= \int f(x)\phi_{j-1,\ell}(x)dx = \int f(x) \sum_{k=0}^{D-1} a_k \phi_{j,2\ell+k}(x)dx \\ &= \sum_{k=0}^{D-1} a_k \int f(x)\phi_{j,2\ell+k}(x)dx = \sum_{k=0}^{D-1} a_k c_{j,2\ell+k} \end{aligned}$$

and $d_{j-1,\ell} = \sum_{k=0}^{D-1} b_k c_{j,2\ell+k}$.

Conversely, noting that

$$\phi_{j,\ell}(x) = \sum_k a_{\ell-2k}\phi_{j-1,k}(x) + \sum_k b_{\ell-2k}\psi_{j-1,k}(x)$$

we obtain

$$c_{j,\ell} = \sum_k a_{\ell-2k}c_{j-1,k} + \sum_k b_{\ell-2k}d_{j-1,k}(x)$$

These two above equations lead to the discrete fast wavelet transform.

An example

As an example, let

$$\mathbf{c} = \begin{pmatrix} c_{3,0} \\ c_{3,1} \\ c_{3,2} \\ c_{3,3} \\ c_{3,4} \\ c_{3,5} \\ c_{3,6} \\ c_{3,7} \end{pmatrix}.$$

The decomposition steps are:

$$
\begin{pmatrix} c_{3,0} \\ c_{3,1} \\ c_{3,2} \\ c_{3,3} \\ c_{3,4} \\ c_{3,5} \\ c_{3,6} \\ c_{3,7} \end{pmatrix}
\longrightarrow
\begin{pmatrix} c_{2,0} \\ c_{2,1} \\ c_{2,2} \\ \underline{c_{2,3}} \\ d_{2,0} \\ d_{2,1} \\ d_{2,3} \\ d_{2,4} \end{pmatrix}
\longrightarrow
\begin{pmatrix} c_{1,0} \\ \underline{c_{1,1}} \\ d_{1,0} \\ d_{1,1} \\ d_{2,0} \\ d_{2,1} \\ d_{2,3} \\ d_{2,4} \end{pmatrix}
\longrightarrow
\begin{pmatrix} \underline{c_{0,0}} \\ d_{0,0} \\ d_{1,0} \\ d_{1,1} \\ d_{2,0} \\ d_{2,1} \\ d_{2,2} \\ d_{2,4} \end{pmatrix}.
$$

The final vector is the results of the DWT of the initial values.

Matrix Representation of the DWT

Let

$$\mathbf{c}_j = (c_{j,0}, \ldots, c_{j,2^j-1})^T$$

and

$$\mathbf{d}_j = (d_{j,0}, \ldots, d_{j,2^j-1})^T$$

The DWT equations define linear maps from $\mathbb{R}^{2^j}$ to $\mathbb{R}^{2^j-1}$ and may be written as:

$$\begin{aligned} \mathbf{c}_{j-1} &= \mathbf{A}_j \mathbf{c}_j \\ \mathbf{d}_{j-1} &= \mathbf{B}_j \mathbf{c}_j \end{aligned}$$

where $\mathbf{A}_j$ and $\mathbf{B}_j$ are $2^{j-1} \times 2^j$ matrices defined via the filters.

Approximation properties

A linear approximation of square integrable function f on an orthonormal basis $\mathcal{B} = \{e_n\}_{n\in\mathbb{Z}}$ projects f on the space spanned by M vectors chosen *a priori* in $\mathcal{B}$:

$$f_M = \sum_{n=0}^{M-1} \langle f, e_n \rangle e_n.$$

The quality of the approximation

$$\|f - f_M\|_2 = \sum_{n \geq M} |\langle f, e_n \rangle|_2$$

depends on the properties of f with respect to $\mathcal{B}$.

Fourier analysis provides efficient approximations for global smooth functions by projecting them on the space spanned by sinusoidal waves of frequencies the first M low frequencies.

In a wavelet basis the signal is projected on V_M.

Here too, the quality of approximation depends on the global regularity of the function lf.

Nonlinear approximation

A linear approximation of f may be enhanced if the M vectors are chosen *a posteriori*, in a way that depends on f.

For M fixed, the approximation error is minimized by taking the M vectors for which the coefficients $|\langle f, e_n \rangle|$ are the largest.

If $\mathcal{B}$ is a wavelet basis,the amplitude of the coefficients is related to the local regularity of f and a nonlinear approximation amounts in using an adaptative sampling whose resolution increases locally where the function is irregular.

First application: Compression

Consider a function f expanded in a basis as

$$f(x) = \sum_{k=1}^{m} c_k e_k(x).$$

The input data are the coefficients $c_1, \ldots, c_m$. The aim of compression is to define an approximation of f using much less coefficients (possible in a different basis) with a minimum loss of information.
Given an upper bound for the error, say $\epsilon > 0$, we seek

$$\tilde{f}(x) = \sum_{k=1}^{\tilde{m}} \tilde{c}_k \tilde{e}_k(x)$$

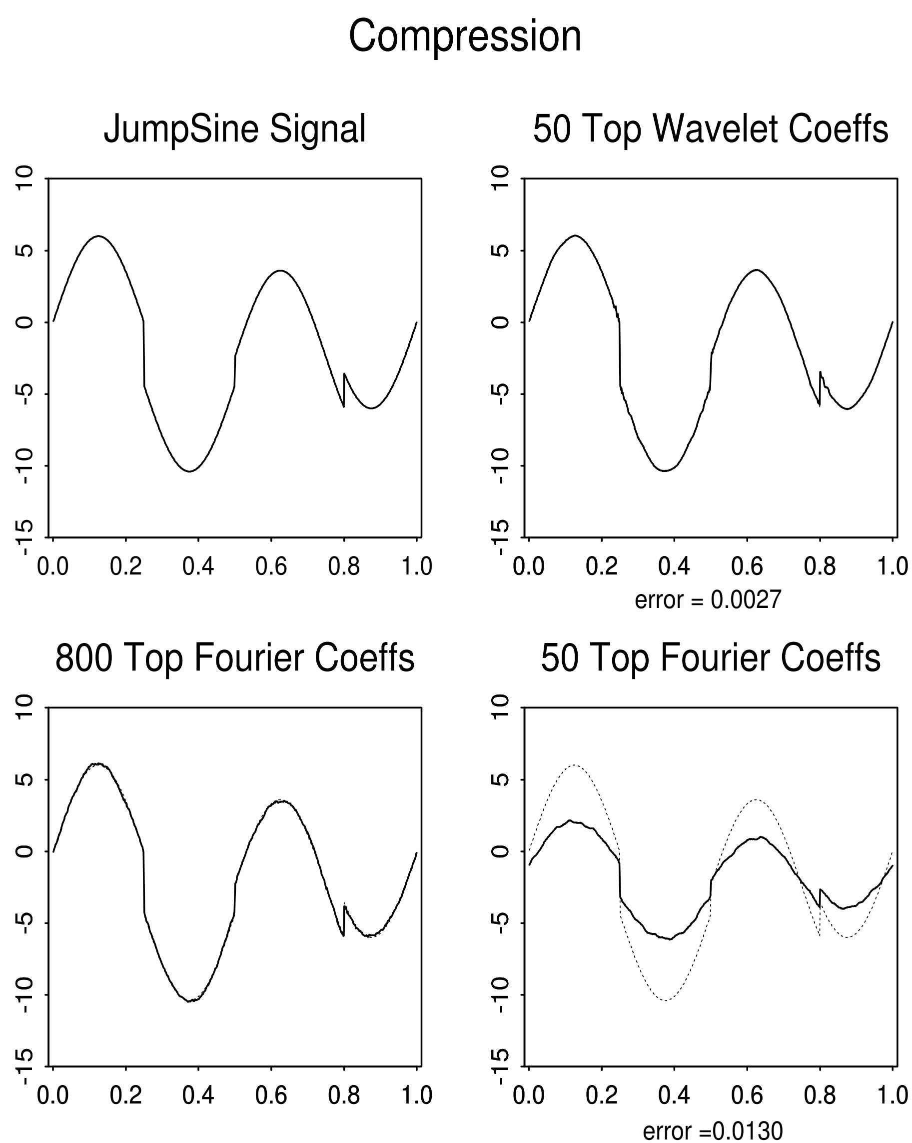
with $\tilde{m} < m$ and $\|f - \tilde{f}\| \leq \epsilon$.

For simplicity consider that the basis is fixed one for all and that it is orthonormal.
Let σ be a permutation of $\{1, 2, \ldots, m\}$ and let $\tilde{f}$ the approximation using the first $\tilde{m}$ coefficients of the permutation σ :

$$\tilde{f}(x) = \sum_{k=1}^{\tilde{m}} c_{\sigma(k)} e_{\sigma(k)}(x).$$

The L^2 error of this approximation is $\|f - \tilde{f}\|_2^2 = \sum_{k=\tilde{m}+1}^{m} |c_{\sigma(k)}|^2$. To minimize it σ must rank the coefficients in a decreasing order of their absolute magnitude.

Example



Compression example

Second application: denoising

Data :

$$Y_j = f(t_j) + \epsilon_j, \quad j = 1, \ldots, n$$

Usually : t_j equidistant, $n = 2^m$, and ϵ_j i.i.d mean 0 and variance σ^2. In practice,

- nonequispaced design;

Second application: denoising

Data :

$$Y_j = f(t_j) + \epsilon_j, \quad j = 1, \ldots, n$$

Usually : t_j equidistant, $n = 2^m$, and ϵ_j i.i.d mean 0 and variance σ^2. In practice,

- nonequispaced design;
- number of data points not a power of 2.

Second application: denoising

Data :

$$Y_j = f(t_j) + \epsilon_j, \quad j = 1, \ldots, n$$

Usually : t_j equidistant, $n = 2^m$, and ϵ_j i.i.d mean 0 and variance σ^2. In practice,

- nonequispaced design;
- number of data points not a power of 2.
- design may be random;

Second application: denoising

Data :

$$Y_j = f(t_j) + \epsilon_j, \quad j = 1, \ldots, n$$

Usually : t_j equidistant, $n = 2^m$, and ϵ_j i.i.d mean 0 and variance σ^2. In practice,

- nonequispaced design;
- number of data points not a power of 2.
- design may be random;
- the errors may not be of constant variance;

Linear methods

- Extension of penalized least-squares methods

Linear methods

- Extension of penalized least-squares methods
- choice of the smoothing parameter by GCV.

Linear methods

- Extension of penalized least-squares methods
- choice of the smoothing parameter by GCV.
- An example.

Simple idea:

Approximate $2^{m/2}\langle f, \phi_{m,k}\rangle \sim f(k/2^m)$ and replace the raw data $\{Y_i\}$ by its interpolation in V_m:

$$\hat{f}_m(t) = 2^{-m/2} \sum_{k\in\mathbb{Z}} Y_k \phi_{m,k}(t)$$

Minimize

$$\|\hat{f}_m - f\|^2_{L^2([0,1])} + \lambda J^p_{spp}(P_{V_{J_0}} f)$$

where J_0 is given et J_{spp} is norm on $B^s_{pp}([0,1])$.

The norm of $g \in B^s_{pq}([0,1])$ is equivalent to

$$J_{spq}(\alpha, \beta) = \|\alpha_{j_0 \cdot}\|_p + \left(\sum_{j=0}^{\infty} (2^{j(s+(1/2)-(1/p))} \|\beta_{j\cdot}\|_p)^q \right)^{1/q} .$$

The solution

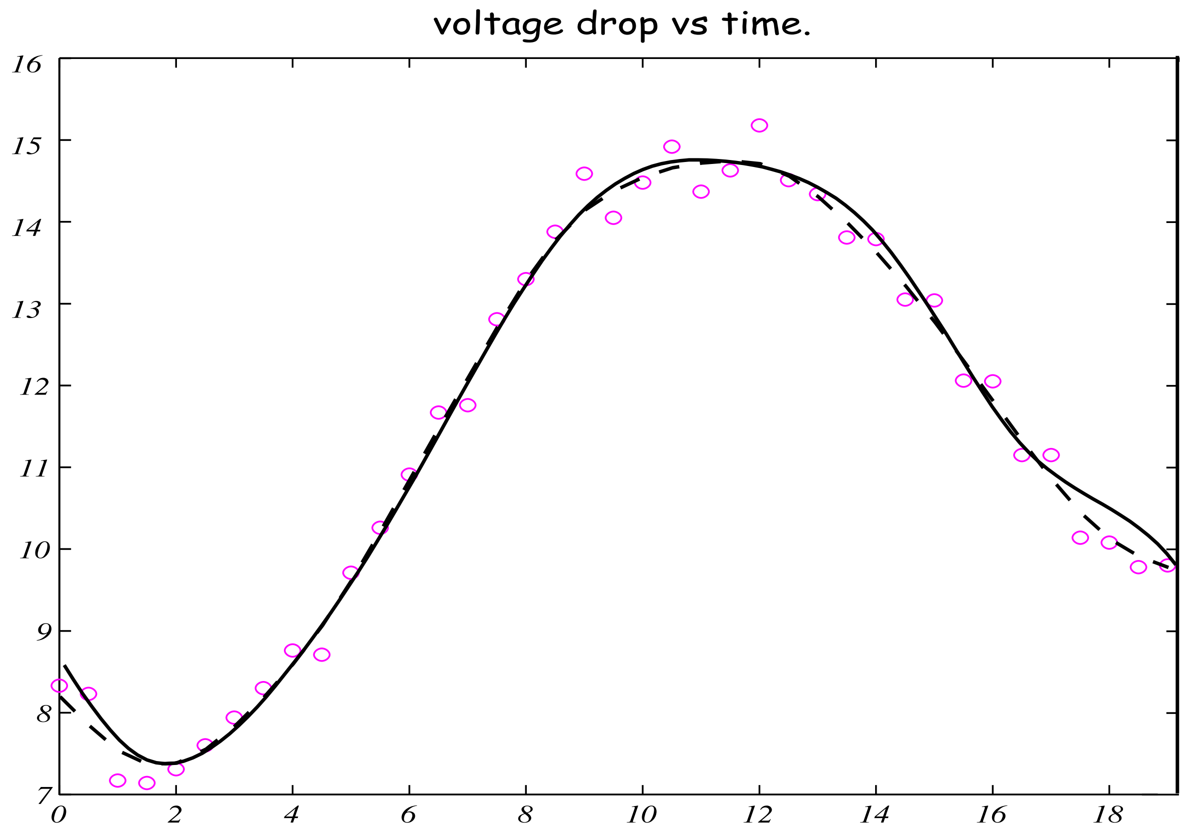
$$f_\lambda = \sum_{k=0}^{2^{j_0}-1} c_{j_0,k}\varphi_{J_0,k} + \sum_{j=J_0}^{m}\sum_{k=0}^{2^j-1} \hat{\beta}_{j,k}\psi_{j,k},$$

• $c_{j_0,k}, \quad k = 0,\ldots,2^{j_0}-1$ are the scaling coefficients of the DWT of $\hat{f}_m$.

• $\hat{\beta}_{j,k} = \frac{d_{j,k}}{1+\lambda 2^{2sj}}, \quad j \geq j_0, k = 0,\ldots,2^j-1,$ with $d_{j,k}$ the wavelet coefficients of $W\hat{f}_m$.

Cross validation

One may choose J_0 and λ given the data and show that the resulting estimation has good properties (at least asymptotically). One popular method is cross validation.



A real data example with D8 compared to spline smoothing

Nonlinear methods

- Decompose the signal into its wavelet coefficients

Nonlinear methods

- Decompose the signal into its wavelet coefficients
- Extract the most significant coefficients by shrinkage or thresholding.

Nonlinear methods

- Decompose the signal into its wavelet coefficients
- Extract the most significant coefficients by shrinkage or thresholding.
- Denoise by applying the inverse wavelet transform on the resulting coefficients.

Almost all existing nonlinear methods are of the form:

$$2^{-1}\sum_{i=1}^{n}(z_i-\theta_i)^2+\lambda\sum_{i\geq i_0}p(|\theta_i|),$$

where z_i is the ith row of $\mathbf{z}=\mathbf{WY}_n$ and p_λ is an appropriate penalty function.

Hard thresholding

1. Transform the data via DWT : $\tilde{\Theta} = W \cdot \mathbf{Y}$.
2. To separate the signal form its noise, threshold the coefficients: set $\hat{\theta}_{j,k} = \tilde{\theta}_{j,k}$ if $|\tilde{\theta}_{j,k}|$ is large, 0 otherwise.
3. Estimate the signal by $\hat{f} = W^{-1} \cdot \hat{\Theta}$.

Soft thresholding

1. Transform the data via DWT : $\tilde{\Theta} = W \cdot \mathbf{Y}$.
2. To separate the signal form its noise, threshold the coefficients: set

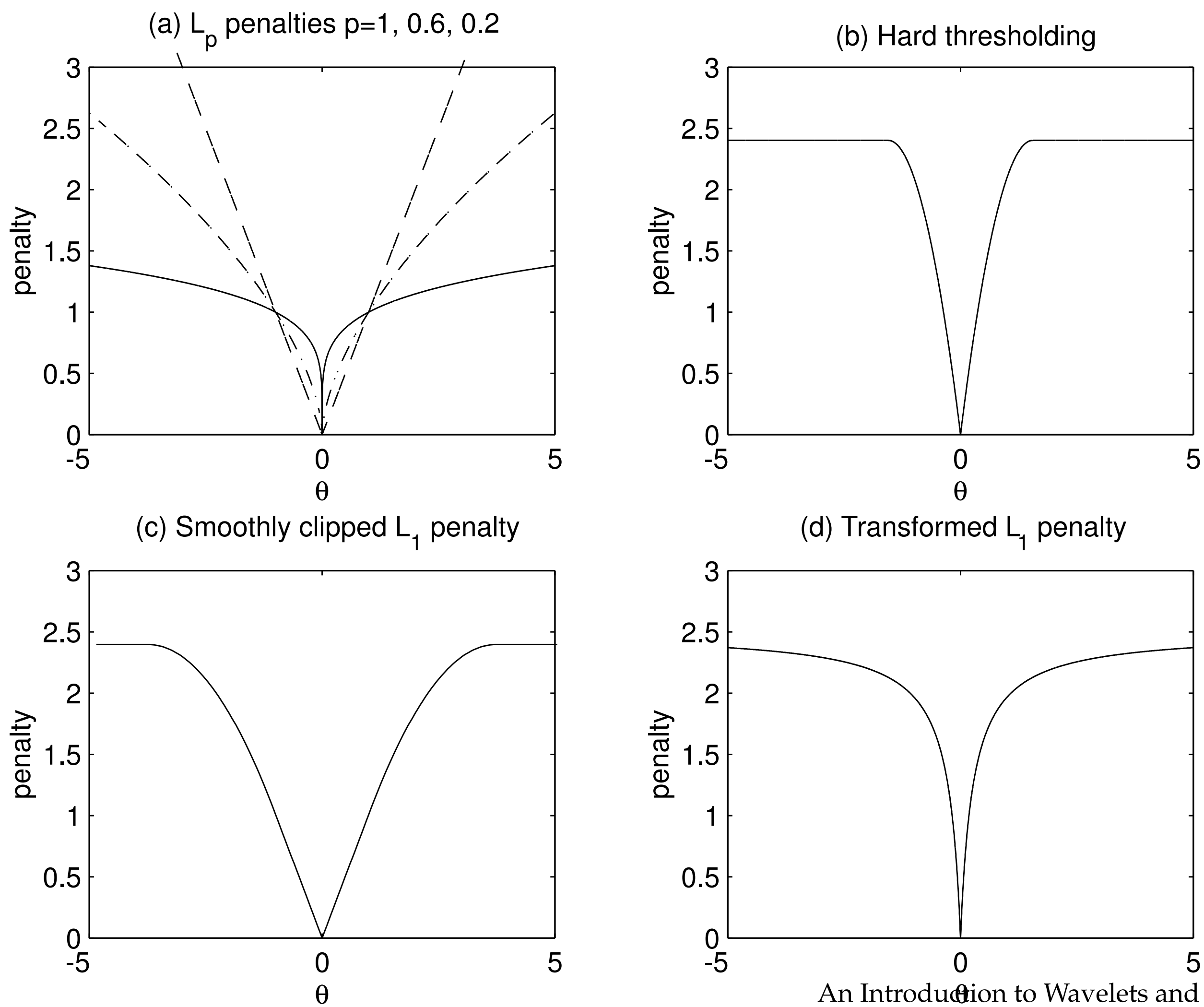
$$\hat{\theta}_{j,k} = \begin{cases} \tilde{\theta}_{j,k} - \lambda, & \text{si } |\tilde{\theta}_{j,k}| > \lambda, \\ 0, & \text{si } |\tilde{\theta}_{j,k}| \leq \lambda, \end{cases}$$

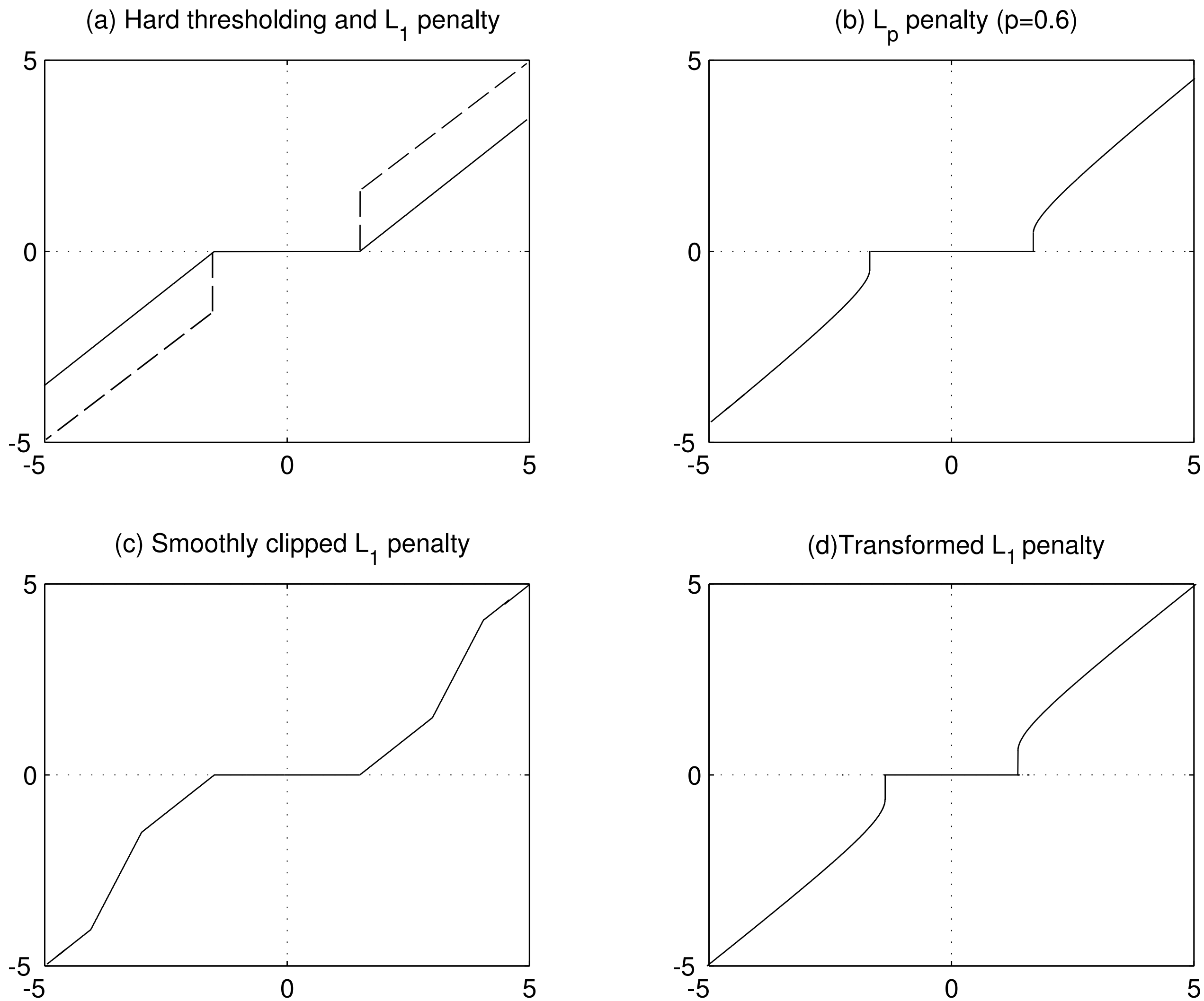
3. Estimate the signal by $\hat{f} = W^{-1} \cdot \hat{\Theta}$.

Many ways for choosing the thresholds.The initial resolution j_0 is usually set to $(\log_2 N)/2$.

Examples

(a) L_p penalty with $p = 1$ (soft (p=1)), $p = 0.6$ (short dash) and $p = 0.2$ (solid); (b) hard; (c) SCAD (robust); (d) transformed soft.

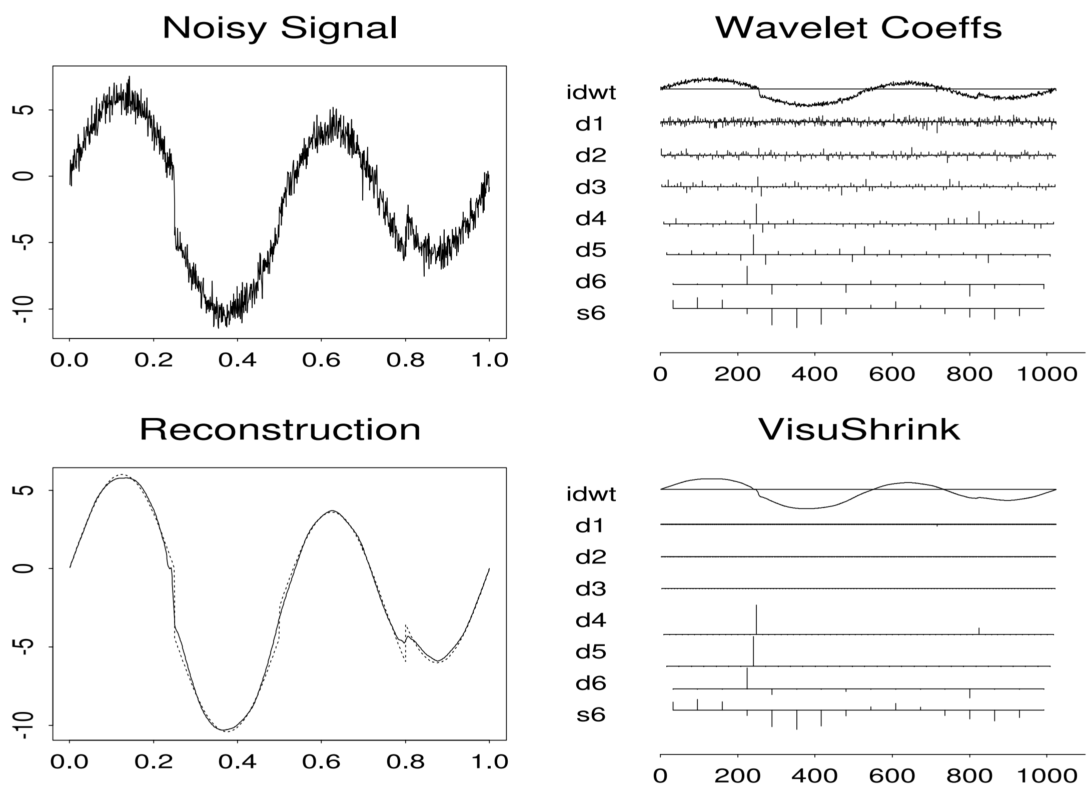




Corresponding estimates

Universal thresholding (VisuShrink)

When the noise is Gaussina, most of the coefficients (normalized as: $\sqrt{N} d_{j,k}/\sigma$) are essentially white noise. This suggests to take $\lambda = \sqrt{2 \log N}$, producing the principle VisuShrink implemented in the package wavethresh of R.



VisuShrink

Properties

When compared, it seems that hard thresholding produces estimates with larger variance while soft thresholding is more biased.
Some authors propose a robust version:

$$\hat{\theta}_{j,k}^{\lambda_1,\lambda_2} = \begin{cases} 0 & \text{si } |\tilde{\theta}_{j,k}| \leq \lambda_1 \\ \operatorname{sgn}(\tilde{\theta}_{j,k}) \frac{\lambda_2(|\tilde{\theta}_{j,k}|-\lambda_1)}{\lambda_2-\lambda_1} & \text{si } \lambda_1 < |\tilde{\theta}_{j,k}| \leq \lambda_2 \\ \tilde{\theta}_{j,k} & \text{si } |\tilde{\theta}_{j,k}| > \lambda_2, \end{cases}$$

which groups the advantages of both thresholding methods

References

Newland D. E (1994) *An Introduction to Random Vibrations, Spectral and Wavelet Analysis*, Third Edition, (Wiley).

Koornwinder, T H (1993) *Wavelets: An Elementary Treatment in Theory and Applications*, (World Scientific, Singapore).

Strang, G. (1989) *Wavelets and Dilation equations: A Brief Introduction*, dans *SIAM Review*, **31**(4) 614-627.

Daubechies, I. (1992) *Ten Lectures on Wavelets*, CBMS-NSF Series in Applied Mathematics (SIAM Publications, Philadelphia).

Chui, C. K. (1992) *An Introduction to Wavelets*, (Academic Press).

Software

- **WaveLab** for Matlab.
- **Wavelet Toolbox** for Matlab.
- **Wavelab** for Scilab.
- **S+Wavelets** for S-plus.
- **Wavetresh 2.2** for S-plus and R.

Many new denoising algorithms are implemented in MATLAB and are available at

`www-lmc.imag.fr/SMS/software`