Random tree Besov priors for detail detection

Hanne Kekkonen

Delft Institute of Applied Mathematics
Delft University of Technology

November 22, 2023

Outline

Outline

Bayesian approach to inverse problems

We want to recover the unknown f from a noisy measurement M;

$$M = Af + noise,$$

where A is a forward operator that usually causes loss of information.

- Consider observing data M drawn at random from some unknown probability distribution $P^M_{f^\dagger}$, and sample size n.
- Specify a prior distribution Π for the unknown f and assume

$$M \,|\, f \sim P^M_f.$$

- Using Bayes' theorem the prior distribution can be updated to a posterior distribution

$$\pi_M(f) = \pi(f \,|\, M) \propto p(M - Af)\pi(f).$$

Bayesian approach to inverse problems

We want to recover the unknown f from a noisy measurement M;

$$M = Af + noise,$$

where A is a forward operator that usually causes loss of information.

- Consider observing data M drawn at random from some unknown probability distribution $P^M_{f^\dagger}$, and sample size n.
- Specify a prior distribution Π for the unknown f and assume

$$M \,|\, f \sim P^M_f.$$

- Using Bayes' theorem the prior distribution can be updated to a posterior distribution

$$f \,|\, M \sim \Pi(\cdot \,|\, M).$$

Gaussian priors are often used for inverse problems

- Assume measurement model

$$M = Af + \delta \mathbb{W},$$

  where A is a linear forward operator and $\mathbb{W} \sim \mathcal{N}(0, I)$.

- If we assume $f \sim \mathcal{N}(0, C_f)$ the posterior is also Gaussian and CM coincides with MAP estimate and is given by

$$\widehat{f}(M) = (A^*A + \delta^2 C_f^{-1})^{-1} A^* M.$$

- Standard Gaussian priors are often used in practice due to their fast computational properties.

Many applications require edge preservation

Noisy image

ℓ^2-regularised solution

TV-regularised solution

$$\pi_{TV}(f) \propto \exp\left(\alpha \sum_{i,j} |f_{i+1,j} - f_{i,j}| + |f_{i,j+1} - f_{i,j}|\right)$$

Is total variation prior consistent?

When discretisation gets finer the discrete total variation prior either diverges or the posterior distribution converges to a Gaussian distribution.
$\Rightarrow$ Not edge preserving with fine discretisation, Lassas and Siltanen 2004.

The widely used formal total variation prior

$$\pi_{pr}(f) \underset{formally}{\approx} \exp(-\alpha \|\nabla f\|_{L^1}), \quad f \in L^2.$$

is not known to correspond to any well defined random variable.

We want to

- Have similar edge preserving properties than total variation priors.
- Correspond to well defined infinite dimensional random variables.
- Can be approximated by finite dimensional random variables.

Outline

Replacing TV prior by a Besov prior

We can replace the formal prior

$$\pi(f) \underset{\text{formally}}{\propto} \exp\left(-\|\nabla f\|_{L^1}\right)$$

by a well defined Besov prior

$$\pi(f) \underset{\text{formally}}{\propto} \exp\left(-\|\nabla f\|_{B^0_{11}}^p\right),$$

that was first introduced by Lassas, Saksman and Siltanen 2009, and further studied by Dashti, Harris and Stuart 2012.

How to form a random function?

Remember that $\sqrt{2}\sin(kt)$ and $\sqrt{2}\cos(kt)$ form an orthonormal basis for $L^2[-\pi,\pi]$. A periodic signal $u(t)$, $t \in [-\pi,\pi]$, can be written as

$$u(t) = \sum_{k=1}^{\infty} a_k \sin(kt) + b_k \cos(kt).$$

Extension of this idea for random functions is given by

$$U(t) = \sum_{k=1}^{\infty} Z_k \psi_k(t),$$

where Z_k's are pairwise uncorrelated random variables and ψ_k is an orthonormal basis on $L^2[-\pi,\pi]$.

How to form a random function?

Remember that $\sqrt{2}\sin(kt)$ and $\sqrt{2}\cos(kt)$ form an orthonormal basis for $L^2[-\pi,\pi]$. A periodic signal $u(t)$, $t \in [-\pi,\pi]$, can be written as

$$u(t) = \sum_{k=1}^{\infty} a_k \sin(kt) + b_k \cos(kt).$$

Extension of this idea for random functions is given by

$$U(t) = \sum_{k=1}^{\infty} h_k Z_k \psi_k(t),$$

where Z_k's are i.i.d. random variables and ψ_k is an orthonormal basis on $L^2[-\pi,\pi]$.

Karhunen–Loève expansion

We can construct random draws from a Gaussian measure;

- Let $\{\psi_k, \lambda_k\}$ be an orthonormal set of eigenvectors and eigenvalues for the covariance operator Σ.
- Take $\{\xi_k\}_{k=1}^{\infty}$ to be a sequence of independent random variables with $\xi_k \sim \mathcal{N}(0,1)$.

Then the random variable U given by the Karhunen–Loève expansion

$$U(t) = \sum_{k=1}^{\infty} \sqrt{\lambda_k}\xi_k\psi_k(t)$$

is distributed according to $\mathcal{N}(0,\Sigma)$.

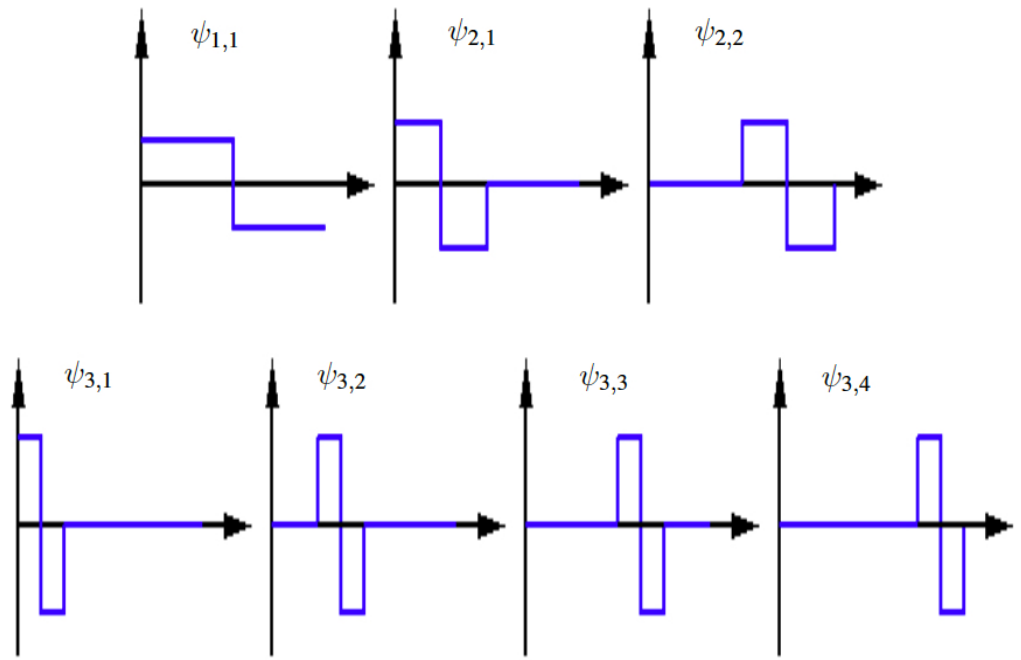
Example: If Σ^{-1} is a Laplace type operator the eigenvalues will grow like k^{-2}.

Wavelet basis

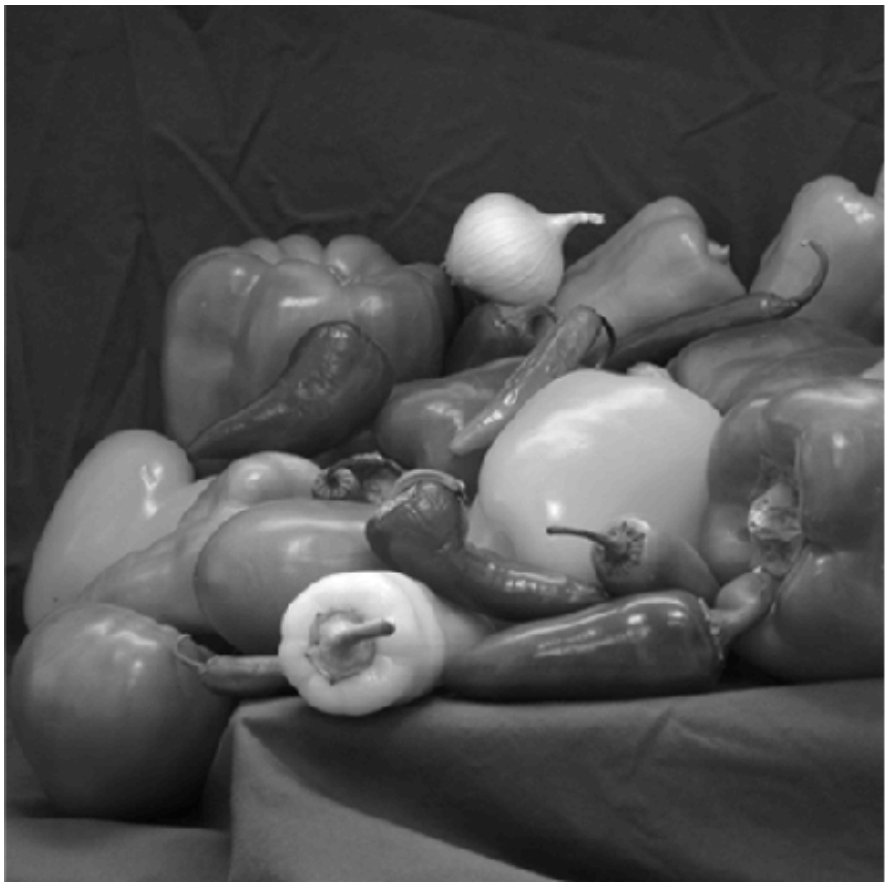
Let Ψ be the mother wavelet suitable for multi-resolution analysis of smoothness C^r and define wavelets

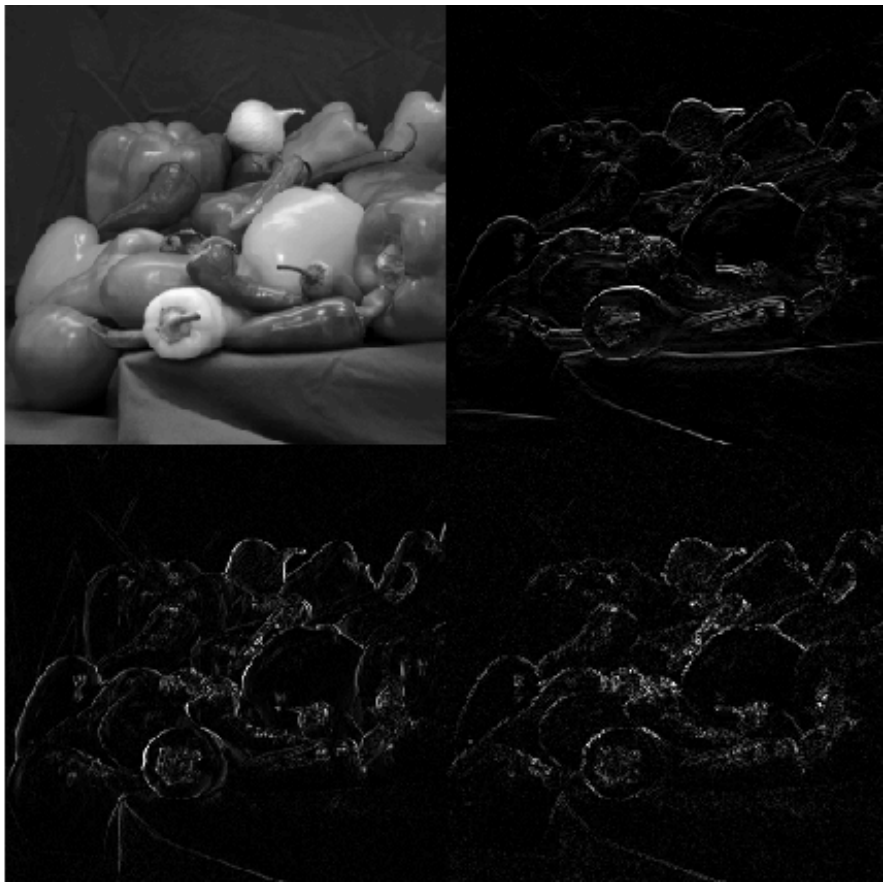
$$\psi_{j,k}(x) = 2^{j/2}\Psi(2^j x_1 - k_1, \ldots, 2^j x_d - k_d), \quad j \in \mathbb{N},\ k \in \mathbb{Z}^d.$$

We consider $f(x) = \sum_{j\in\mathbb{N},k\in\mathbb{Z}^d} f_{j,k}\psi_{j,k}(x), \quad f_{j,k} = \langle f, \psi_{j,k}\rangle.$

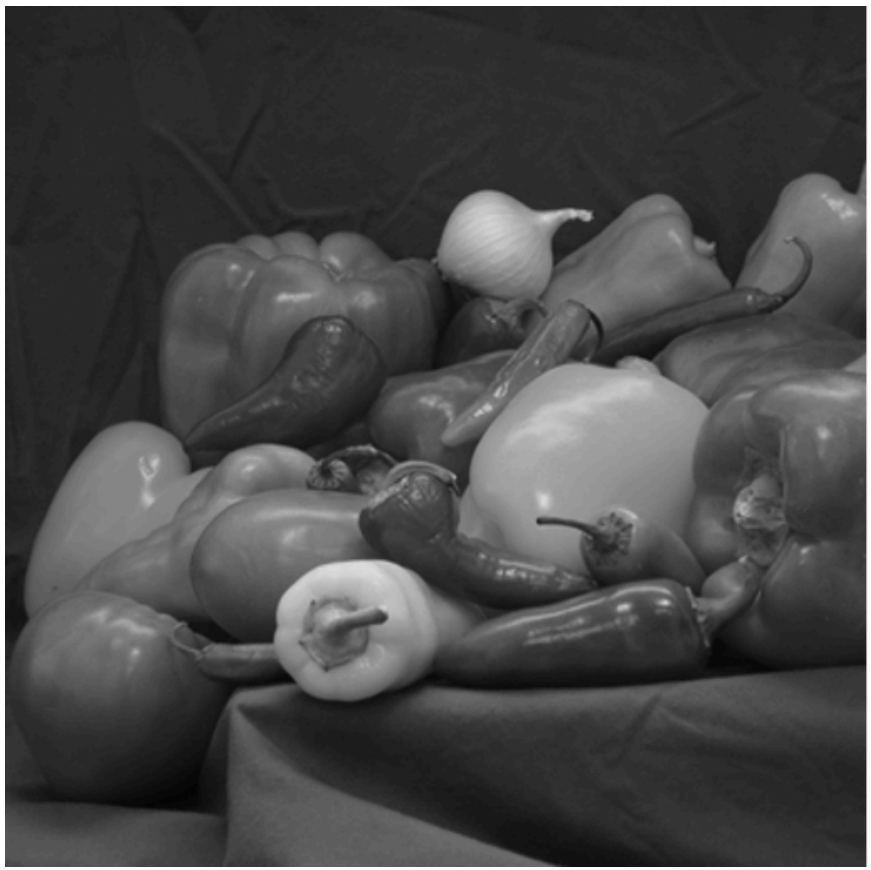


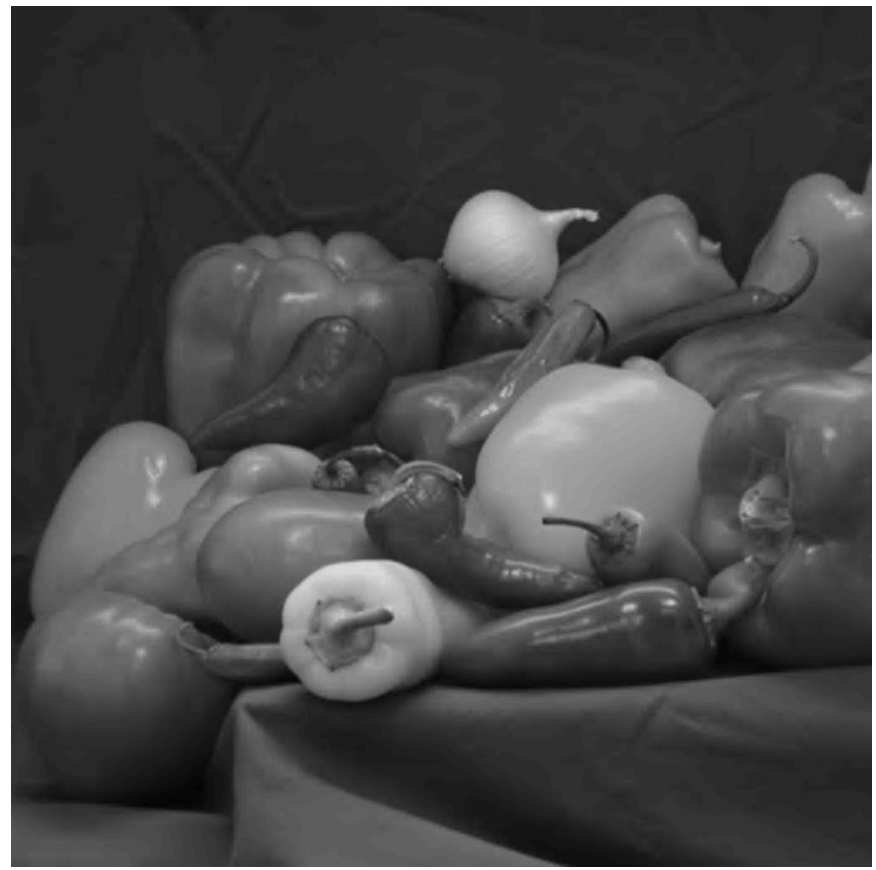
Discrete wavelet transform





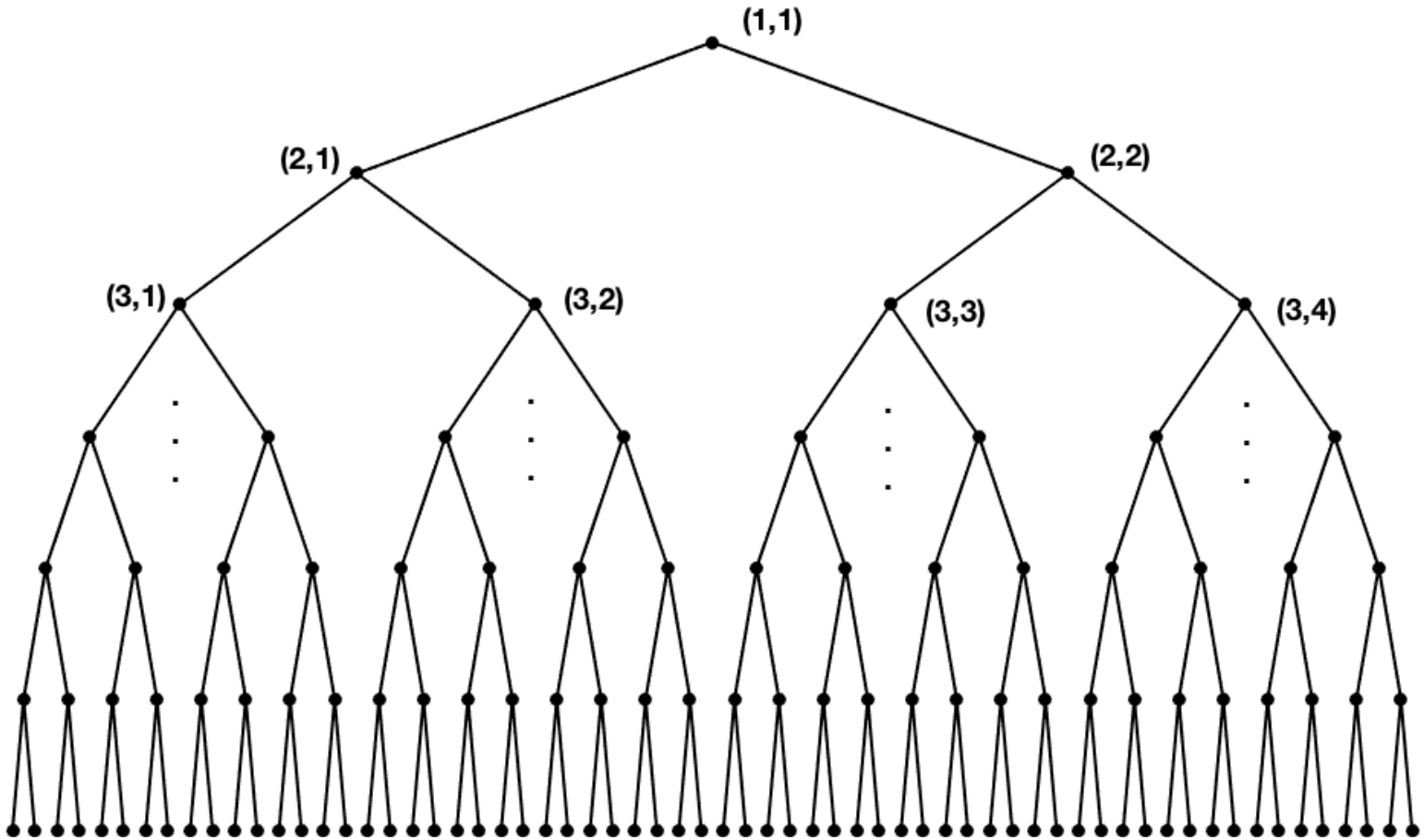
Thresholded peppers





Left: the original image. Right: 95% of the wavelet coefficients are set to zero using hard thresholding.

The wavelet coefficients can be placed into a tree ($d = 1$)



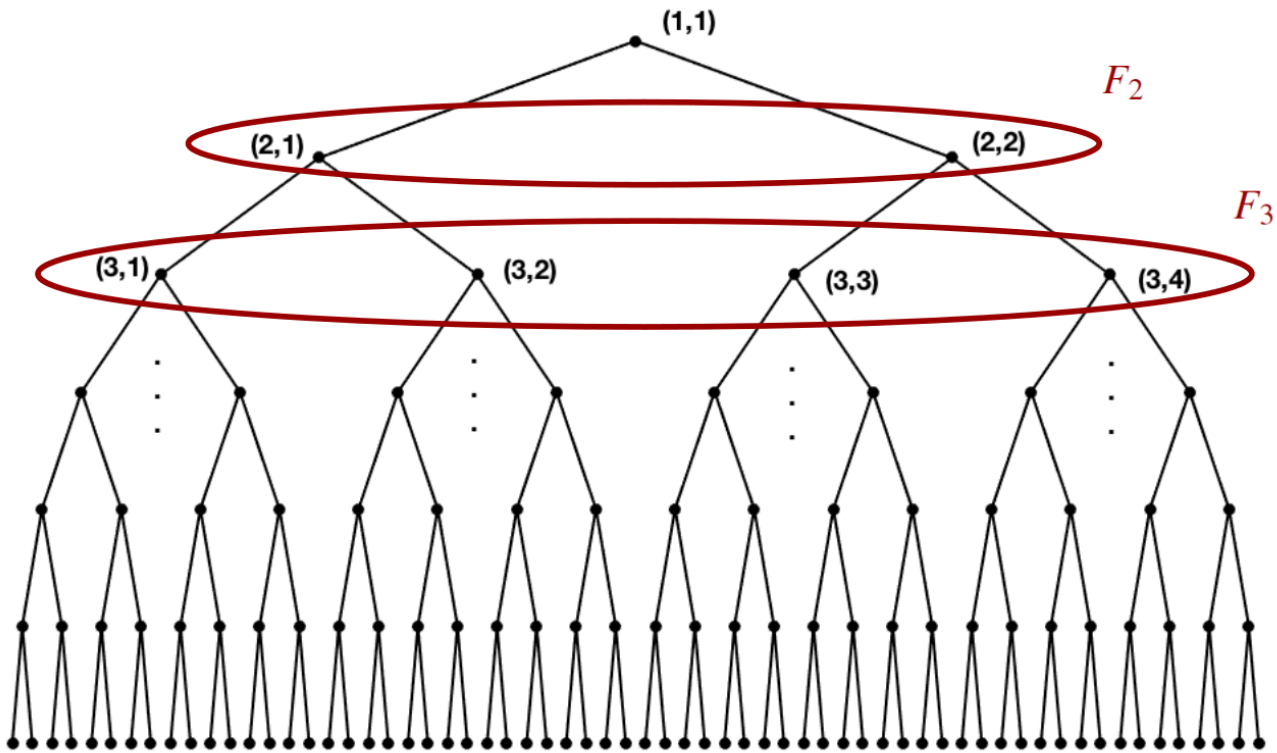
An entire tree is defined as a set of indices

$$\mathbf{T} = \{(j,k) \in \mathbb{N} \times \mathbb{N}^d \mid j \in \mathbb{N}_{\geq 1}, k = (k_1, \cdots, k_d),\ 1 \leq k_\ell \leq 2^{j-1}\},$$

Besov spaces B^s_{pp}

For $s < r$, the Besov norm can be defined as

$$\|f\|^p_{B^s_{pp}(\mathbb{R}^d)} = \sum_{j=0}^{\infty} 2^{jp(s+d(\frac{1}{2}-\frac{1}{p}))} \|F_j\|^p_{\ell p} \qquad F_j = (f_{j,k})_{k\in\mathbb{Z}^d}.$$



Besov spaces B^s_{pp}

For $s < r$, the Besov norm can be defined as

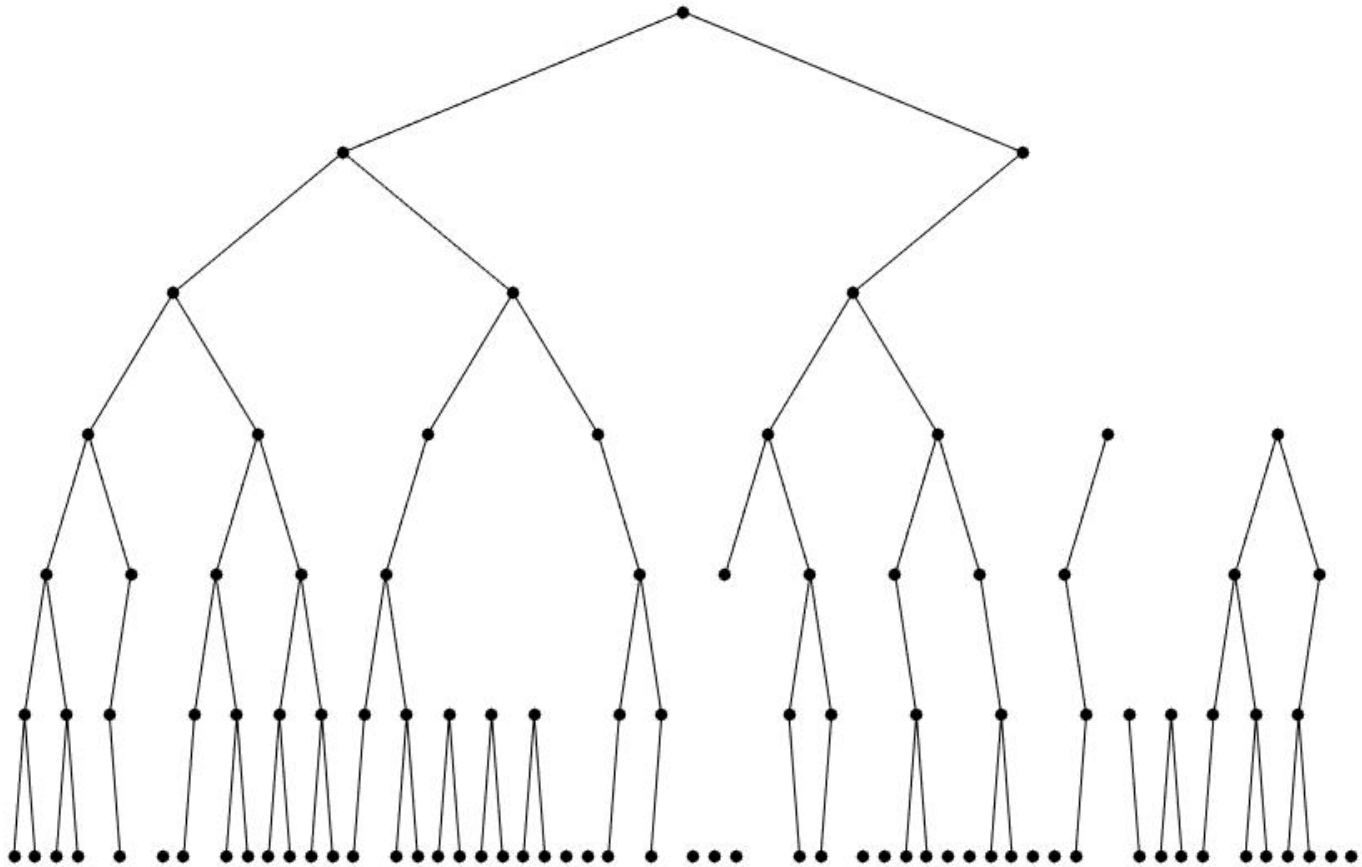
$$\|f\|^p_{B^s_{pp}(\mathbb{R}^d)} = \sum_{j=0}^{\infty} 2^{jp(s+d(\frac{1}{2}-\frac{1}{p}))} \|F_j\|^p_{\ell p} \qquad F_j = (f_{j,k})_{k\in\mathbb{Z}^d}.$$

- Besov spaces $B^s_{22}(\mathbb{R}^d)$ coincide with the Sobolev spaces $H^s(\mathbb{R}^d)$.
- $B^1_{11}(\mathbb{R}^d)$ space is relatively close to space of functions with bounded variations, $\|\nabla u\|_{L^1} < \infty$.

We can show that:

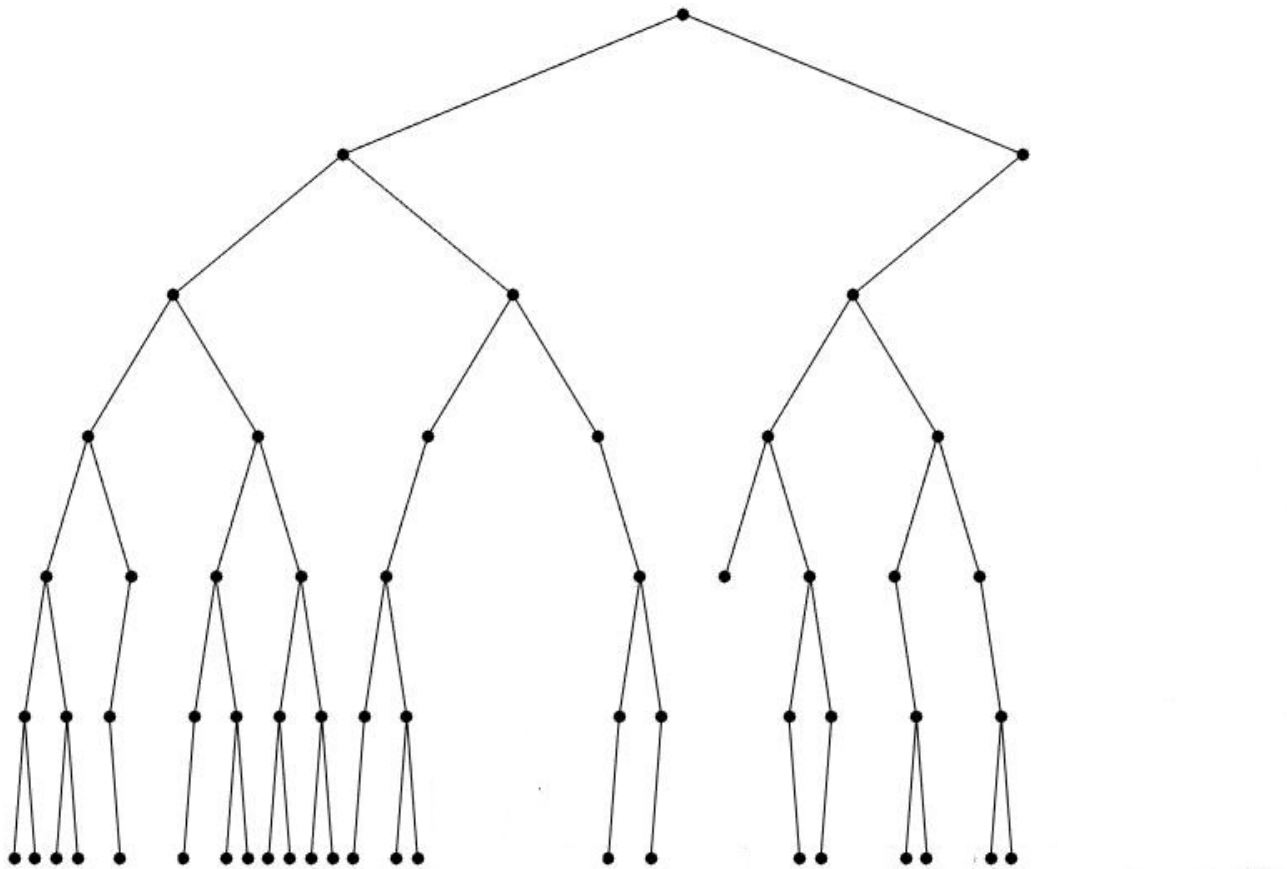
$$B^1_{11}(\mathbb{R}^d) \subset W^{1,1}_{loc}(\mathbb{R}^d) \subset B^{1-\varepsilon}_{11}(\mathbb{R}^d), \qquad \text{for all} \quad \varepsilon > 0.$$

Creating a proper subtree



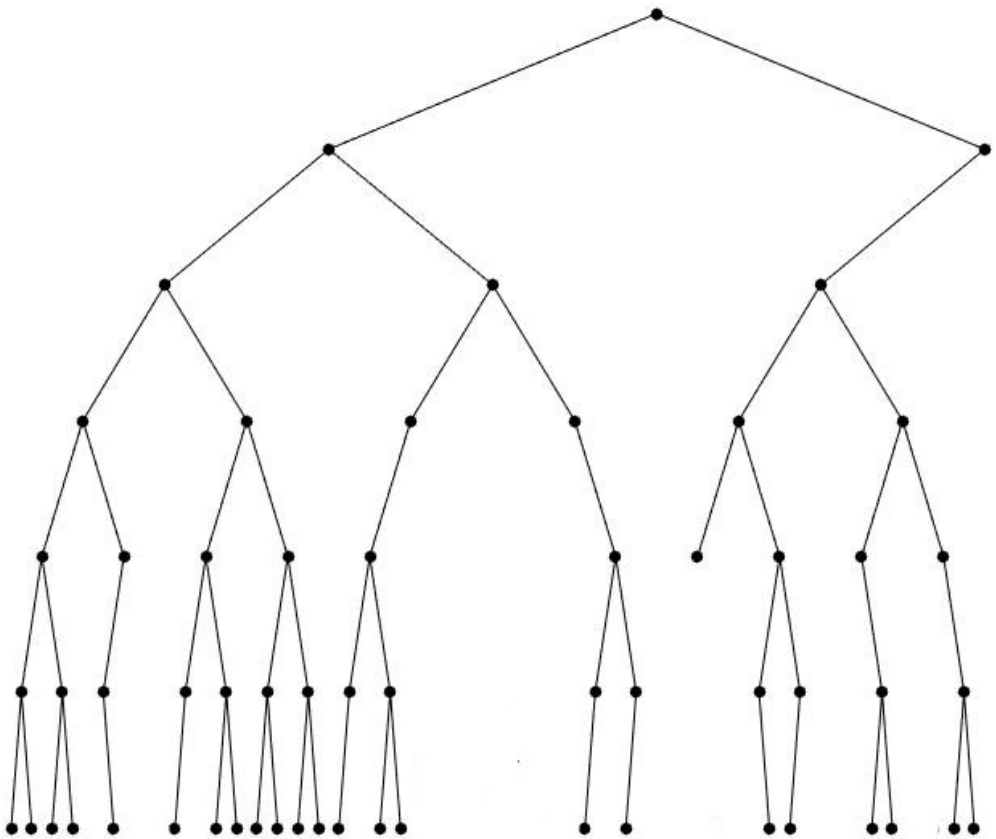
Draw $t_{j,k} \sim \mathcal{U}[0,1]$ and set a node 1 if $t_{j,k} \leq \beta$, $\beta \in [0,1]$, and 0 otherwise.

Creating a proper subtree



Only choose nodes that are connected to the root node.

Random tree Besov priors ($d = 1$)



$$X_{j,k} \sim \mathcal{N}(0,1)$$

or

$$X_{j,k} \sim \text{Laplace}(0,a)$$

$$f(x) = \sum_{(j,k)\in T} h_j X_{j,k} \psi_{j,k}(x).$$

If we set $h_j = 2^{-j(s+\frac{1}{2}-\frac{1}{p})}$ then f takes values in $B^{\tilde{s}}_{pp}$, $\tilde{s} < s - \frac{\log_2(\beta)+1}{p}$.

K., Lassas, Saksman, Siltanen, Random tree Besov priors - Towards fractal imaging, 2022

Outline

Signal denoising

Consider the denoising problem

$$M = f + W,$$

where $W = \sum w_{j,k}\psi_{j,k}$ is white noise, independent of f. We choose prior

$$f(x) = \sum_{(j,k)\in T} f_{j,k}\psi_{j,k}(x) = \sum_{(j,k)\in \mathbf{T}} \tilde{t}_{j,k} g_{j,k}\psi_{j,k}(x),$$

where $g_{j,k} \sim \mathcal{N}(0,1)$ and $\tilde{t}_{j,k} \in \{0,1\}$ defines if a node $(j,k) \in \mathbf{T}$ is chosen. Denote $t_{j,k}$ an independent node, assume $\mathbb{P}(t_{j,k} = 1) = \beta_j$, with $\beta_j \sim \pi$, and

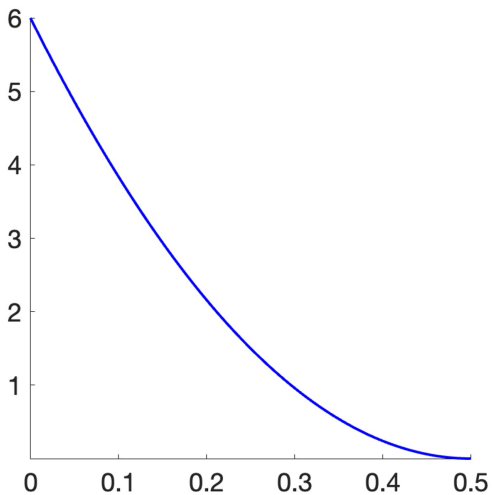
$$\tilde{t}_{j,k} = \prod_{(j',k')\trianglerighteq(j,k)} t_{j,k}.$$

Calculating the MAP estimator

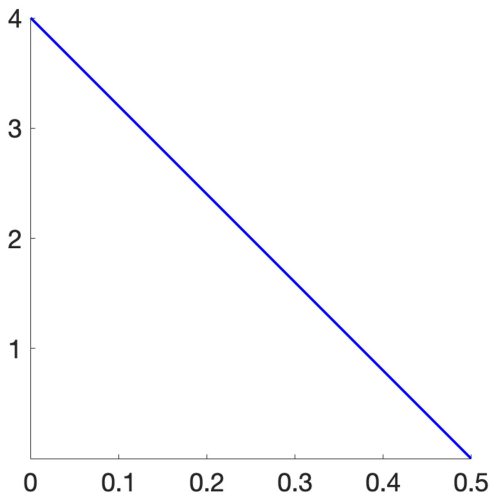
The posterior distribution can be written in form

$$\pi(g,t,\beta \mid m) \propto \pi(m \mid g,t)\pi(g)\pi(t \mid \beta)\pi(\beta)$$
$$= \prod_{(j,k)\in \mathbf{T}} \pi(m_{j,k} \mid g_{j,k}, \tilde{t}_{j,k})\pi(g_{j,k})\pi(t_{j,k} \mid \beta_j)\pi(\beta_j)$$

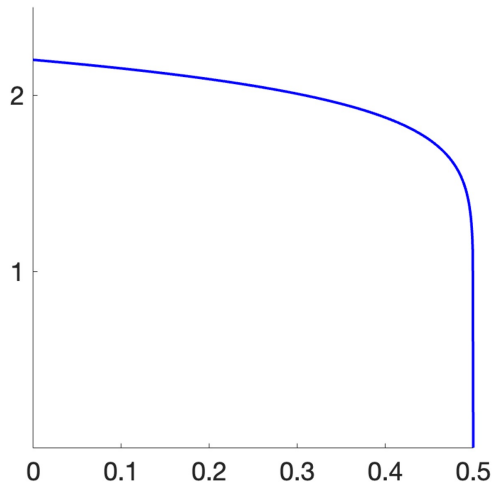
We consider priors of the form $\pi(\beta) = 2^{1+\alpha}(1+\alpha)(\frac{1}{2}-\beta)^{\alpha}$, $\alpha > 0$.



$\alpha = 2$



$\alpha = 1$



$\alpha = 0.1$

Every node is either chosen to the tree or not

We consider maximising

$$z^1_{jk} = \exp\Big(-\frac{1}{2}(m_{jk}-g_{jk})^2 - \frac{g^2_{jk}}{2}\Big) c_\alpha \beta_j (\frac{1}{2}-\beta_j)^\alpha$$

$$z^0_{jk} = \exp\Big(-\frac{1}{2}m^2_{jk} - \frac{g^2_{jk}}{2}\Big) c_\alpha (1-\beta_j)(\frac{1}{2}-\beta_j)^\alpha.$$

or equivalently minimising

$$-\log(z^1_{jk}) = \frac{1}{2}(m_{jk}-g_{jk})^2 + \frac{g^2_{jk}}{2} - \log(c_\alpha \beta_j (\frac{1}{2}-\beta_j)^\alpha)$$

$$-\log z^0_{jk} = \frac{1}{2}m^2_{jk} + \frac{g^2_{jk}}{2} - \log(c_\alpha (1-\beta_j)(\frac{1}{2}-\beta_j)^\alpha).$$

Start from the bottom of the tree

At the bottom leaves we set $\hat{g}_{jk} = m_{jk}^2/2$ to attain the optimal weights $m_{jk}^2/4$. If a node is not selected $\hat{g}_{jk} = 0$ the weight is $m_{jk}^2/2$.

We want to find β_j so that the selected bottom row has an optimal weight, that is, we want to minimise

$$B_{j,A} = \sum_{k \in A} \frac{1}{4} m_{jk}^2 - \log(c_\alpha \beta_j (\frac{1}{2} - \beta_j)^\alpha) + \sum_{k \notin A} \frac{1}{2} m_{jk}^2 - \log(c_\alpha (1 - \beta_j)(\frac{1}{2} - \beta_j)^\alpha),$$

where A is the set of nodes that are selected.

We only need to consider finite number of values for β_j

A bottom row node is selected if

$$\frac{1}{4}m_{jk}^2 - \log\left(c_\alpha \beta_j (\frac{1}{2} - \beta_j)^\alpha\right) \leq \frac{1}{2}m_{jk}^2 - \log\left(c_\alpha (1-\beta_j)(\frac{1}{2} - \beta_j)^\alpha\right)$$

$$m_{jk} \geq 4\log\left(\frac{1-\beta_j}{\beta_j}\right)$$

When minimising $B_{j,A}$ we only consider grid points $\beta_{jk} \geq (1+e^{\frac{1}{4}m_{jk}})^{-1}$.

Order the data $m_{jk_i} \geq m_{jk_{i+1}}$. We can then consider

$$B_{j,i} = \sum_{k \leq k_i} \frac{1}{4}m_{jk_i}^2 + \sum_{k > k_i} \frac{1}{2}m_{jk_i}^2 + k_i \log\left(\frac{1-\beta_{jk_i}}{\beta_{jk_i}}\right)$$

$$- n_j \log\left(c_\alpha (1-\beta_{jk_i})(\frac{1}{2} - \beta_{jk_i})^\alpha\right)$$

where n_j is the number of nodes on the row j.

Optimising for a general level

Denote the weight of an optimised sub-tree with root node (j,k) by G_{jk}. The sub-tree is included in the tree if

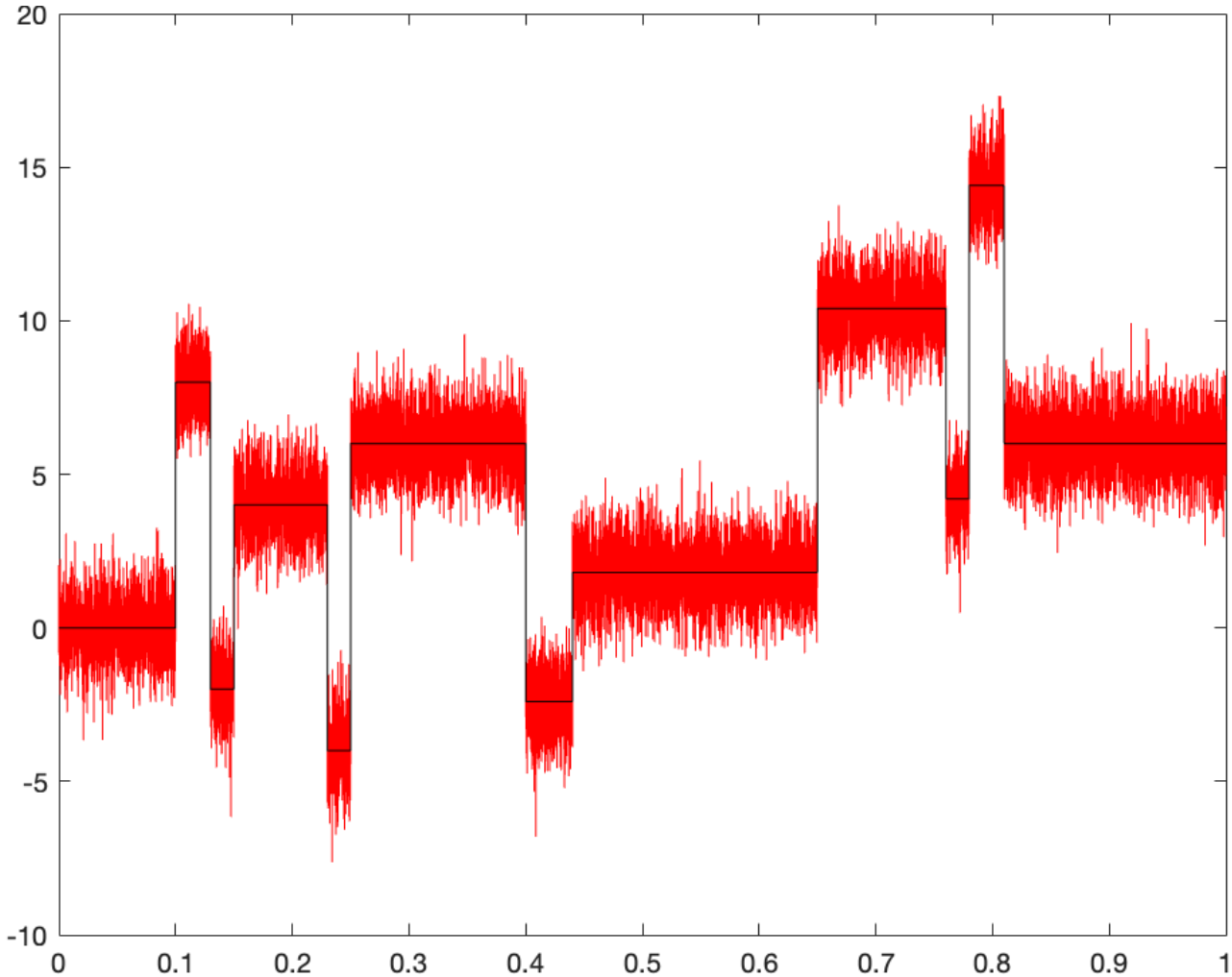
$$G_{jk} - \log(\beta_j) \leq \frac{1}{2}\|m|_{T_{jk}}\|^2 - \log(1-\beta_j) - \sum_{\ell=1+j}^{\ell_{\max}} r_\ell \log(c_\alpha(1-\beta_\ell)(\frac{1}{2}-\beta_\ell)^\alpha),$$

where r_ℓ is the number of nodes of the sub-tree on the row ℓ. We get grid points

$$\beta_{jk} \geq \frac{1}{1+e^{M_{jk}}},$$

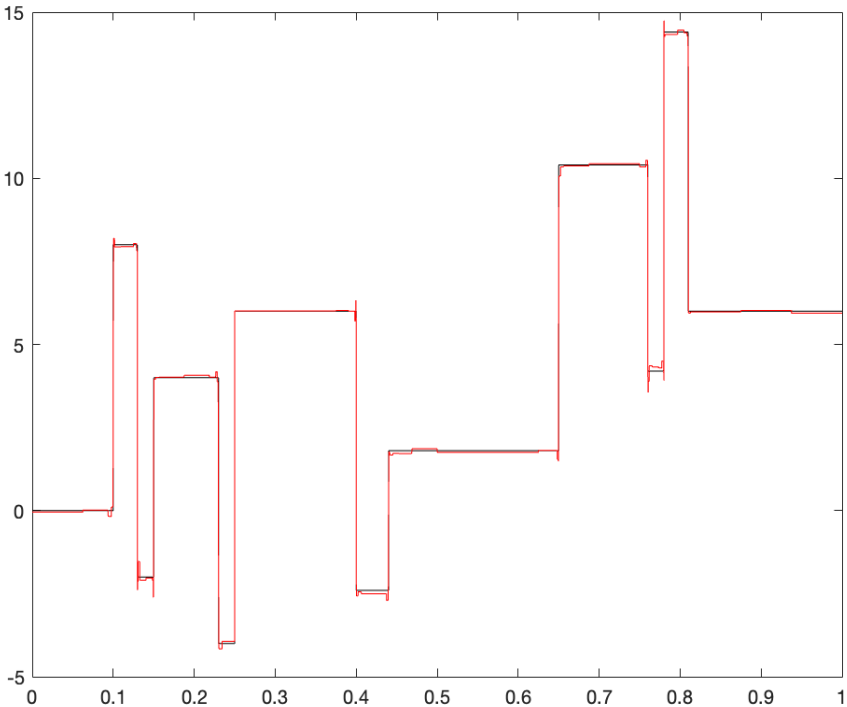
where $M_{jk} = \frac{1}{2}\|m|_{T_{jk}}\|^2 - G_{jk} - \sum_{\ell=1+j}^{\ell_{\max}} r_\ell \log(c_\alpha(1-\beta_\ell)(\frac{1}{2}-\beta_\ell)^\alpha)$.

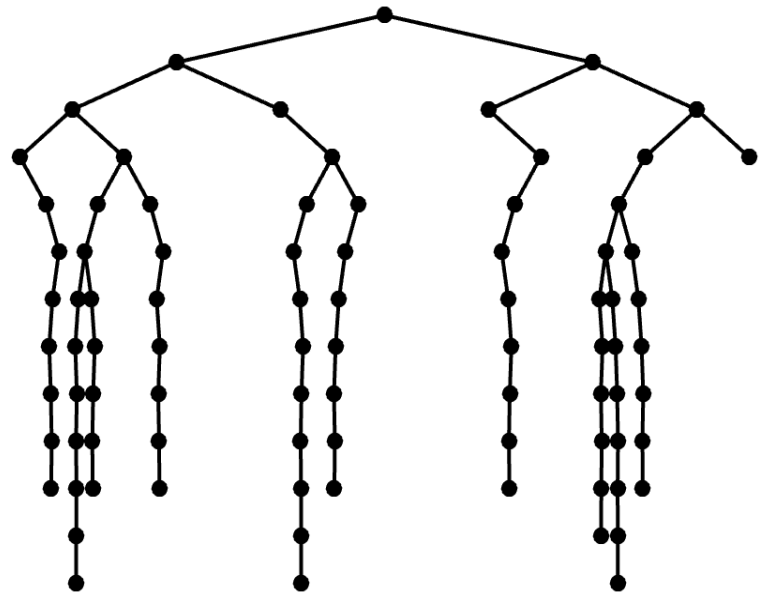
Blocks data



The original signal (black) and the noisy data (red)

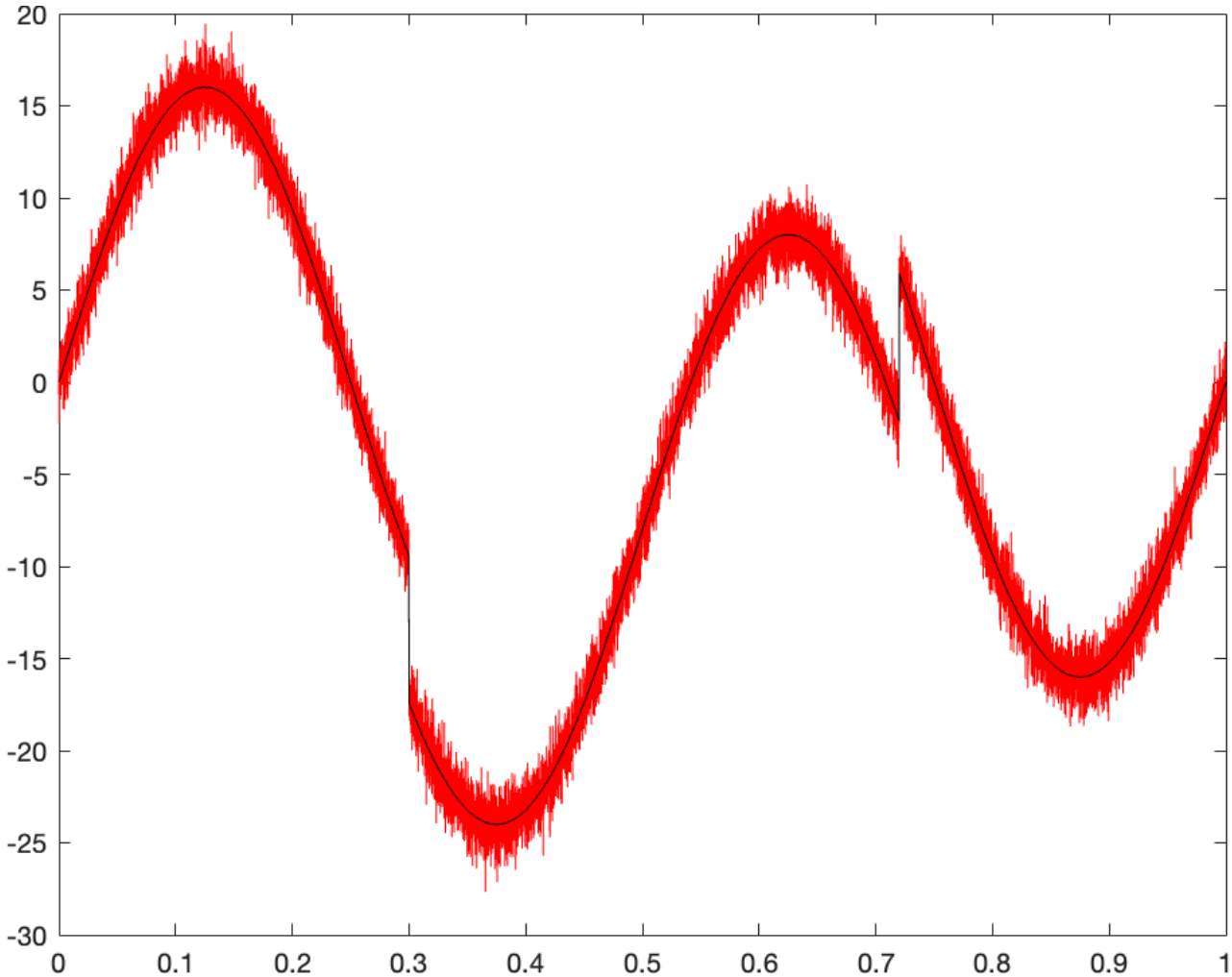
Wavelet pruning with automatic β selection





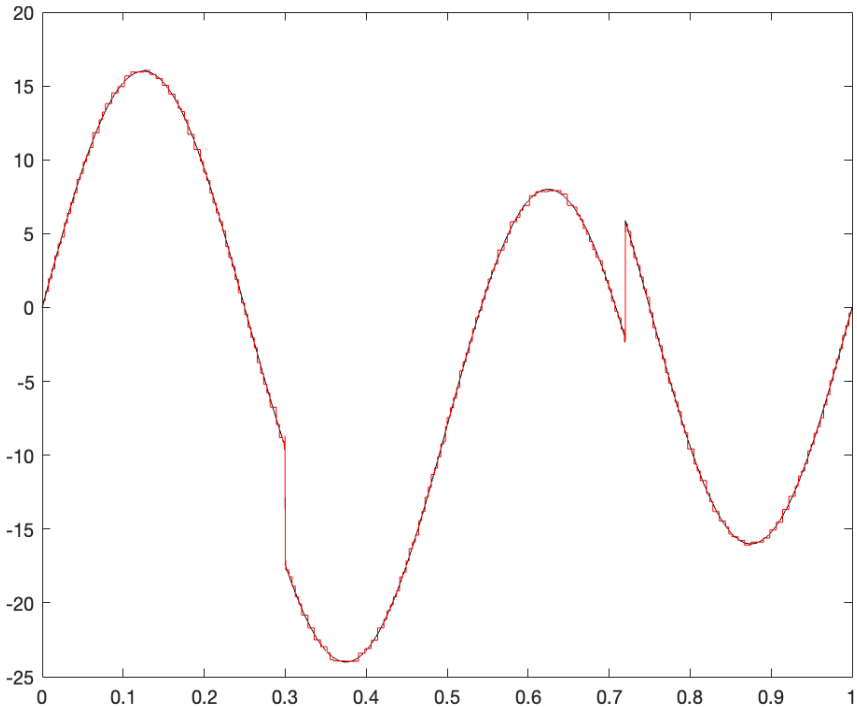
$$\beta = 0.0000,\ 0.0000,\ 0.2894,\ 0.2289,\ 0.1863,\ 0.1959,\ 0.0481,\ 0.1359,$$
$$0.0029,\ 0.0005,\ 0.0000,\ 0.0002,\ 0.0000$$

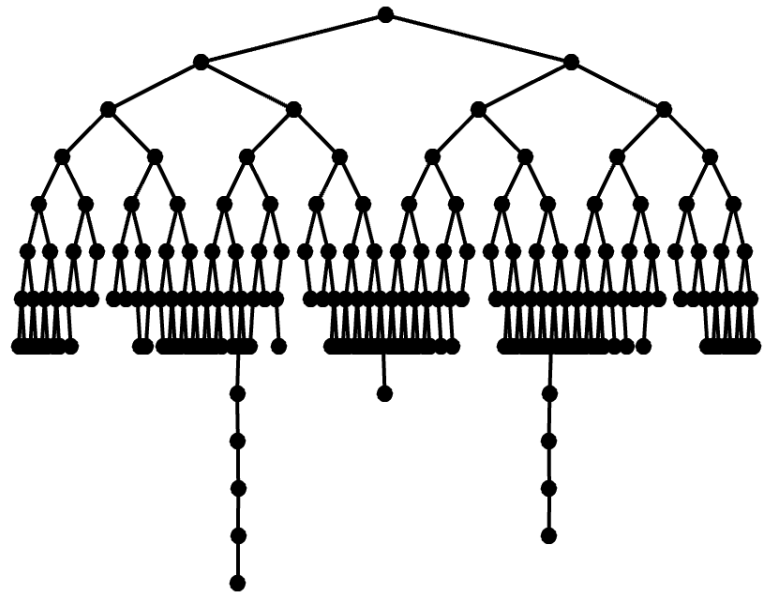
Smooth data with jumps



The original signal (black) and the noisy data (red)

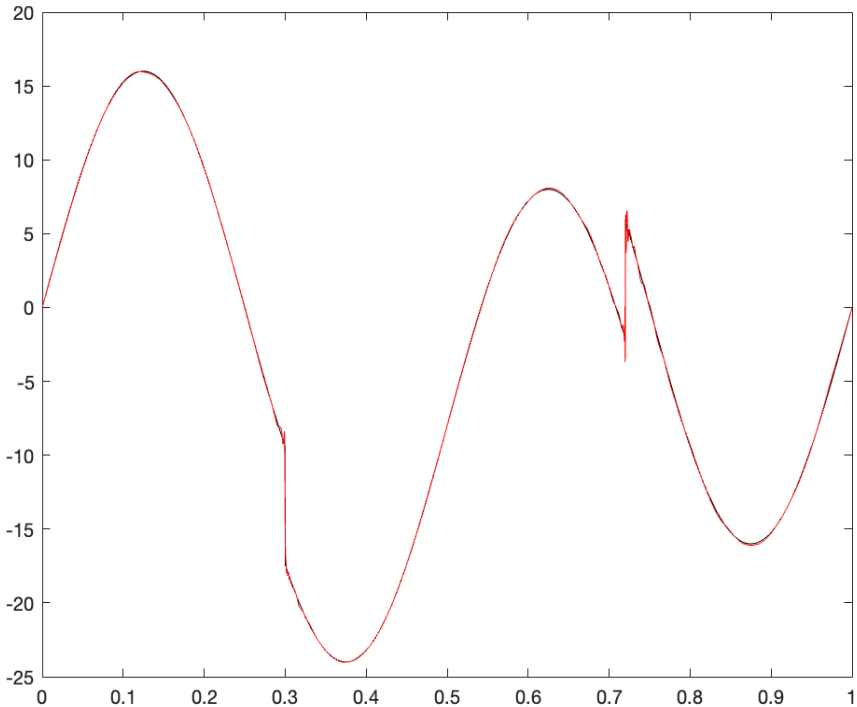
Wavelet pruning with automatic β selection (Haar)

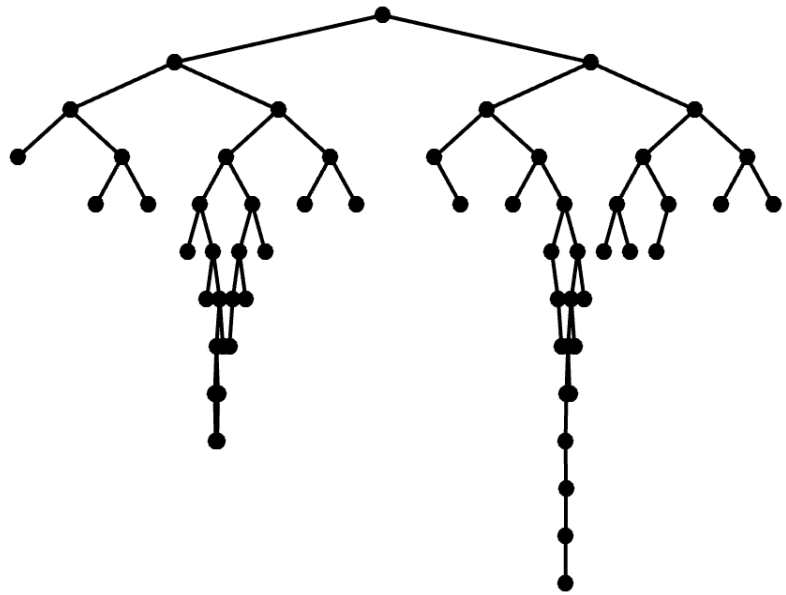




$$\beta = 0.0000,\ 0.0000,\ 0.0000,\ 0.0000,\ 0.0000,\ 0.2374,\ 0.1734,\ 0.0171,$$
$$0.0560,\ 0.0003,\ 0.0000,\ 0.0023,\ 0.0005$$

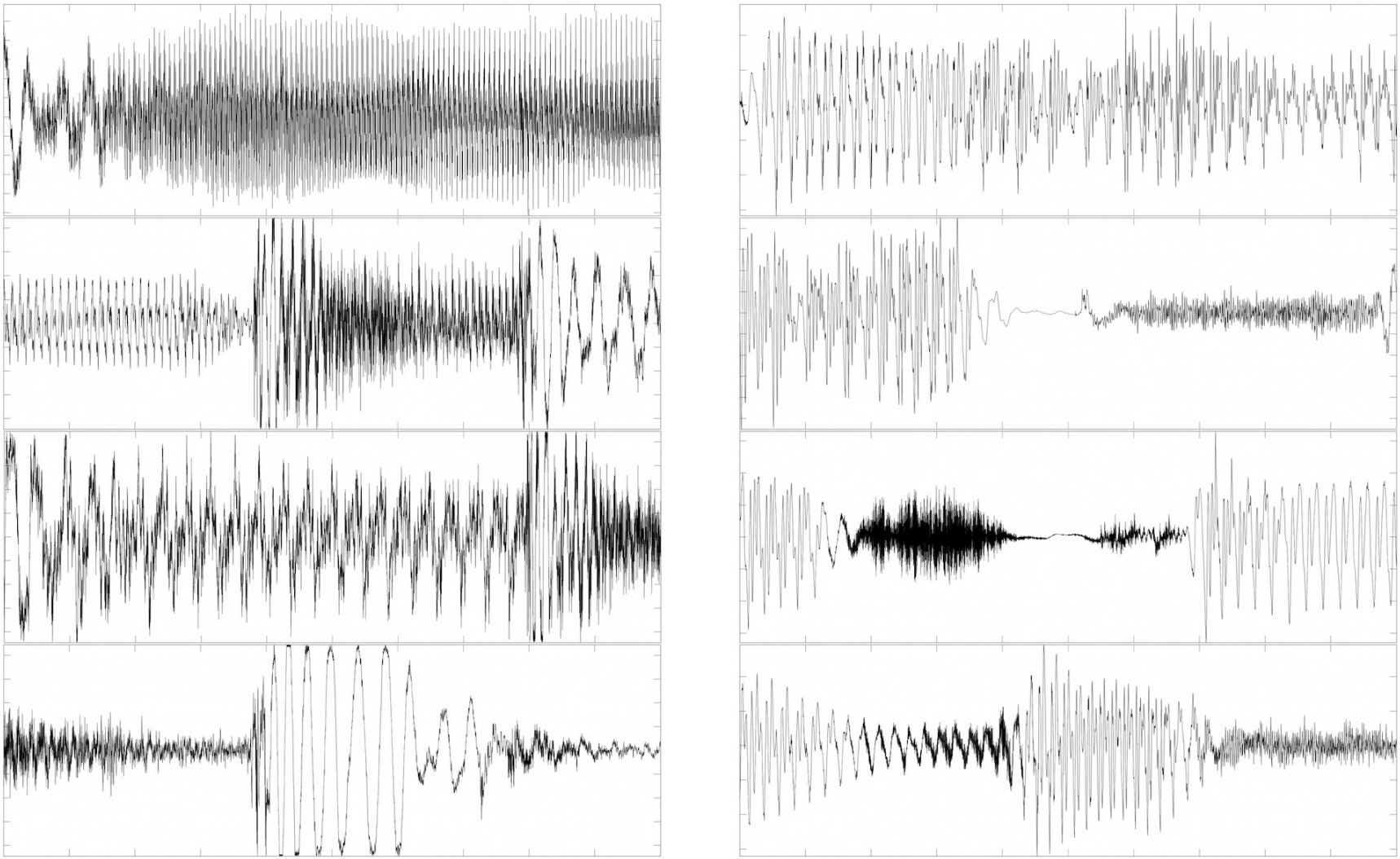
Wavelet pruning with automatic β selection (db4)





$$\beta = 0.0000,\ 0.0000,\ 0.0000,\ 0.3238,\ 0.2268,\ 0.0308,\ 0.0116,\ 0.0025,$$
$$0.0026,\ 0.0008,\ 0.0000,\ 0.0007,\ 0.0559$$

Separating music from speech using optimised β values



Test signals of length 2^{14} of music (left) and speech (right).

Optimised β values (db2)

0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0002	0.0054	0.2334	0.4961	0.4995	0.0011	0.0925
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.2548	0.3121	0.4981	0.4994	0.0023	0.2454
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0002	0.0558	0.4324	0.4900	0.4995	0.0018	0.1123
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0203	0.4732	0.4987	0.4994	0.0103	0.2720
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.2884	0.2391	0.4951	0.4995	0.1230	0.0010	0.3125
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1457	0.1031	0.2878	0.4963	0.4995	0.0888	0.1680
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.4304	0.4938	0.4995	0.0907	0.0014	0.0442

0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.4550	0.4966	0.4995	0.0337	0.3763	0.3754	0.3998
0.0000	0.0000	0.0000	0.2395	0.4138	0.4943	0.4998	0.4995	0.4993	0.0374	0.1518	0.2262	0.3015
0.0000	0.0000	0.0000	0.0000	0.2995	0.4286	0.4936	0.4999	0.4994	0.2561	0.2266	0.0682	0.0549
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1270	0.4642	0.4994	0.0018	0.2399	0.2194	0.2420
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1178	0.4525	0.4995	0.0241	0.2464	0.2684	0.3285
0.0000	0.0000	0.0000	0.0007	0.4624	0.4907	0.4984	0.4993	0.4992	0.0038	0.0044	0.0361	0.3746
0.0000	0.0000	0.0000	0.0652	0.4539	0.4981	0.4994	0.4993	0.4992	0.1375	0.1161	0.0554	0.0244

Optimised β values for music (top) and speech (bottom).

Classification using support vector machine

- Signal length $2^{14} \approx 0.37$s.
- Training set 450 signals.
- Test set 150 signals.
- Classification error $< 5\%$.