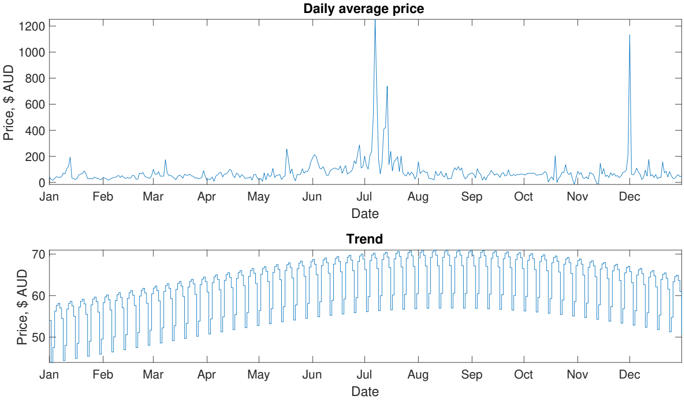
Maths is trendy
Trend estimation and signal processing
(Data science and functional analysis)

Me

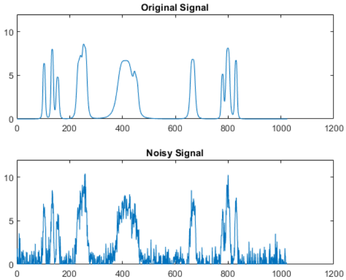
Disclaimer: most of the material here is not my own

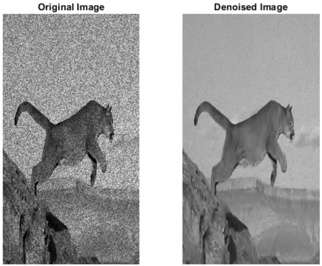
PG seminar, 2019

Motivation


Daily average price
Price, $ AUD
1200
1000
800
600
400
200
0
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Date
Trend
Price, $ AUD
70
60
50
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Date

Motivation


Original Signal
10
5
0
0
200
400
600
800
1000
1200
Noisy Signal
10
5
0
0
200
400
600
800
1000
1200


Original Image
Denoised Image

Motivation

Original

5:1

10:1

27:1

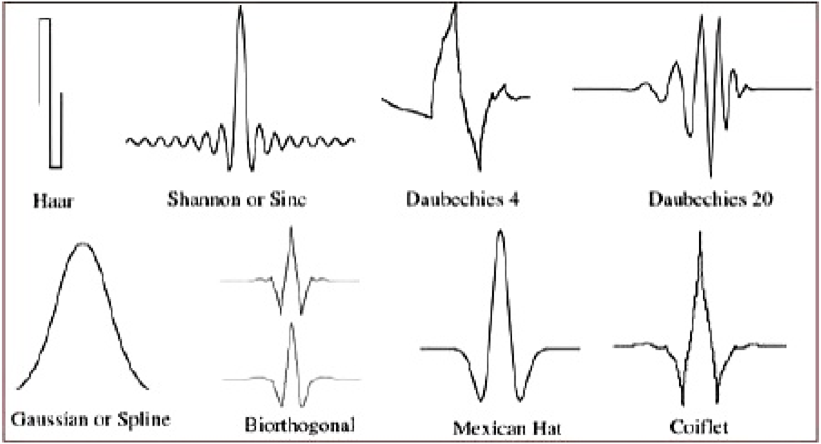
General idea

- Data is represented as a vector (or array) of values.
- Which we represent as a projection onto a basis $\{\boldsymbol{v}_n\}$.

$$\boldsymbol{u} = a_1 \boldsymbol{v}_1 + a_2 \boldsymbol{v}_2 + \ldots + a_N \boldsymbol{v}_N.$$

  The signal is encoded in the coefficients a_i, $i = 1, \ldots, N$.
- For smoothing/denoising: remove high frequency components.
- For compression: discard all the small a_i's and keep the large ones.
- Wavelets provide are one class of basis functions with nice properties which enable computations.

Wavelets and scaling functions



Wavelets and scaling functions

A wavelet $\psi(t)$;

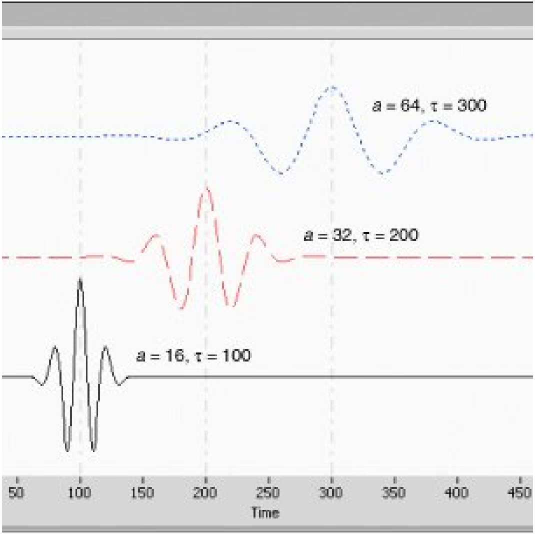
- $\int_{\mathbb{R}} \psi(t) dt = 0$
- other technical conditions

Define shifted and stretched versions of $\psi(t)$ by

$$\psi_{j,n}(t) = \sqrt{2}^{-j} \psi(2^{-j} t - n),$$

$j, n \in \mathbb{Z}$. Forms a basis for $L^2(\mathbb{R})$.

E.g. Morelet wavelet



Multiresolution analysis

$\{V_j, j \in \mathbb{Z}\}$ a family of subspaces

- $\{0\} \subset \cdots \subset V_{-1} \subset V_0 \subset V_1 \cdots \subset L^2(\mathbb{R})$,
- Closure under shifting.

$$g(x) \in V_k \iff g(x - m2^k) \in V_k$$

- For $g(x) \in V_k$ there is $f(x) \in V_\ell$ with

$$g(x) = f(2^{k-\ell}x) \text{ for all } x \in \mathbb{R}.$$

- $\varphi(t-k) \in V_0$ for $k \in \mathbb{Z}$ an orthogonal basis for V_0.

We shift and stretch them too

$$\varphi_{j,n}(t) = \sqrt{2}^{-j}\varphi(\sqrt{2}^{-j}t - n).$$

Wavelets and scaling functions

Relating $\varphi(t)$ and $\psi(t)$

$$\psi(t) = \sum_{k=-N}^{N} (-1)^k a_{1-k} \varphi(2t-k)$$

Two-scale relation

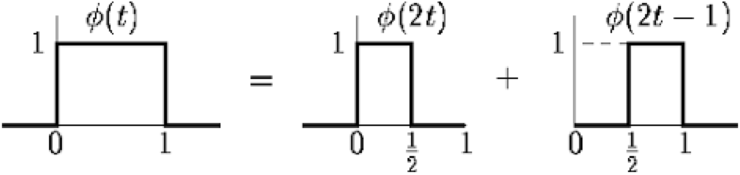
$$\varphi(t) = \sum_{k=-N}^{N} a_k \varphi(2t-k). \tag{1}$$

- These equations, along with other conditions, give the coefficients a_k.
- The sequences a_k and $(-1)^k a_{1-k}$ form *filters*.
- The functions $\psi(t)$ and $\varphi(t)$ are rarely known analytically.

Wavelets and scaling functions

Example: Haar scaling function.

$$\varphi(t) = \begin{cases} 1 & 0 \leq t \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

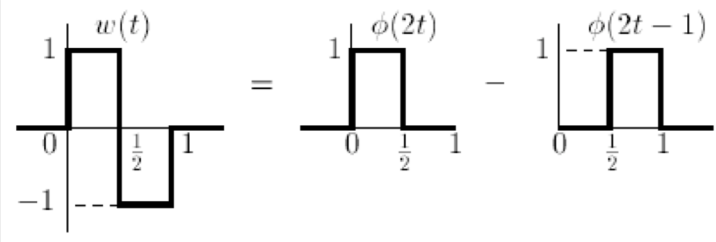


$$\varphi(t) = \sum_{k=-N}^{N} a_k \varphi(2t-k). \qquad (2)$$

Wavelets and scaling functions

Haar wavelet.

$$\psi(t) = \begin{cases} 1 & 0 \leq t \leq 0.5 \\ -1 & 0.5 < t \leq 1 \\ 0 & \text{otherwise} \end{cases}$$



$$\psi(t) = \sum_{k=-N}^{N} (-1)^k a_{1-k} \varphi(2t-k)$$

Projection onto wavelets

Any function $f(t) \in L^2(\mathbb{R})$ can be represented as

$$f(t) = \sum_{j=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} d_{j,n}\psi_{j,n}(t).$$

Data can be viewed as a piecewise constant function so we need only

$$f(t) = \sum_{j=-\infty}^{J} \sum_{n=-\infty}^{\infty} d_{j,n}\psi_{j,n}(t).$$

where

$$d_{j,n} = \int_{\mathbb{R}} f(t)\psi_{j,n}(t)dt,$$

since $\psi_{j,n}$'s are orthonormal. Ultimately, we want to compute $d_{j,n}$?

Projection onto scaling functions

The data can also be written as

$$\begin{aligned} f(t) &= \sum_{j=-\infty}^{J} \sum_{n=-\infty}^{\infty} d_{j,n} \psi_{j,n}(t) \\ &= \sum_{n=-\infty}^{\infty} c_{J+1,n} \varphi_{J+1,n}(t), \end{aligned}$$

where

$$c_{J+1,n} = \int_{\mathbb{R}} f(t) \varphi_{J+1,n}(t) dt.$$

How to compute $c_{j,n}$, $d_{j,n}$?

Suppose (for now) we have $c_{J+1,n}$ for all n. From the two-scale relation we get

$$
\begin{aligned}
c_{J,n} &:= \int_{\mathbb{R}} \varphi_{J,n}(t) f(t) dt \\
&= \int_{\mathbb{R}} \varphi_{J,n}(t) \sum_{n=-N}^{N} c_{J+1,n} \varphi_{J+1,n}(t) \\
&= \int_{\mathbb{R}} \sum_{\ell=-N}^{N} a_\ell \varphi_{J+1,\ell-2n}(t) \sum_{n=-N}^{N} c_{J+1,n} \varphi_{J+1,n}(t) \\
&= \sum_{\ell=-N}^{N} a_{\ell-2n} c_{J+1,\ell}.
\end{aligned}
$$

No integration required!

How to compute $c_{j,n}$, $d_{j,n}$?

Similarly,

$$\begin{aligned} d_{J,n} &:= \int_{\mathbb{R}} \psi_{J,n}(t) f(t) dt \\ &= \int_{\mathbb{R}} \psi_{J,n}(t) \sum_{n=-N}^{N} c_{J+1,n} \varphi_{J+1,n}(t) \\ &= \int_{\mathbb{R}} \sum_{\ell=-N}^{N} (-1)^{\ell} a_{1-\ell} \varphi_{J+1,\ell-2n}(t) \sum_{n=-N}^{N} c_{J+1,n} \varphi_{J+1,n}(t) \\ &= \sum_{\ell=-N}^{N} (-1)^{\ell} a_{1-\ell+2n} c_{J+1,n}. \end{aligned}$$

No integration required!

How to get $c_{J+1,n}$?

Consider

$$f(t) = \sum_{n=-\infty}^{\infty} x_n \varphi_{J+1,n}(t).$$

Then

$$c_{J+1,m} = \int_{\mathbb{R}} f(t)\varphi_{J+1,m}(t)dt = \int_{\mathbb{R}} \sum_{n=-\infty}^{\infty} x_n \varphi_{J+1,n}(t)\varphi_{J+1,m}(t)dt = x_m.$$

Assuming f is nicely behaved

$$c_{J+1,m} = \int_{\mathbb{R}} f(t)\varphi_{J+1,m}(t)dt$$

can be interpreted as a weighted average of f over a small interval, and so $x_m \approx f(m)$.

Back to the top

We can compute

$$f(t) = \sum_{j=-\infty}^{J} \sum_{n=-\infty}^{\infty} d_{j,n}\psi_{j,n}(t).$$

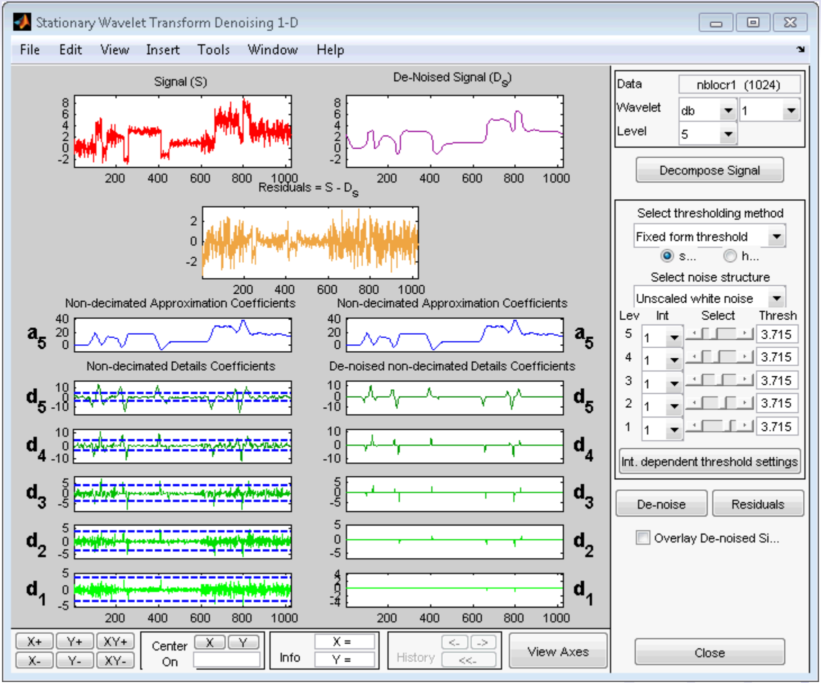
Smoothed/trend estimate is

$$\widehat{f}(t) = \sum_{j=-\infty}^{J-L} \sum_{n=-\infty}^{\infty} d_{j,n}\psi_{j,n}(t).$$

Image compression is

$$\widehat{f}(t) = \sum_{j=-\infty}^{J} \sum_{n=-\infty}^{\infty} d_{j,n}1(d_{j,n} > z)\psi_{j,n}(t).$$

for some threshold z.


Stationary Wavelet Transform Denoising 1-D
File Edit View Insert Tools Window Help
Signal (S)
De-Noised Signal (D$_S$)
Residuals = S - D$_S$
Non-decimated Approximation Coefficients
Non-decimated Approximation Coefficients
Non-decimated Details Coefficients
De-noised non-decimated Details Coefficients
a$_5$
d$_5$
d$_4$
d$_3$
d$_2$
d$_1$
Data
nblocr1 (1024)
Wavelet
db
1
Level
5
Decompose Signal
Select thresholding method
Fixed form threshold
s...
h...
Select noise structure
Unscaled white noise
Lev Int Select Thresh
5 1 3.715
4 1 3.715
3 1 3.715
2 1 3.715
1 1 3.715
Int. dependent threshold settings
De-noise
Residuals
Overlay De-noised Si...
X+ Y+ XY+
X- Y- XY-
Center On
X Y
Info
X =
Y =
History
View Axes
Close

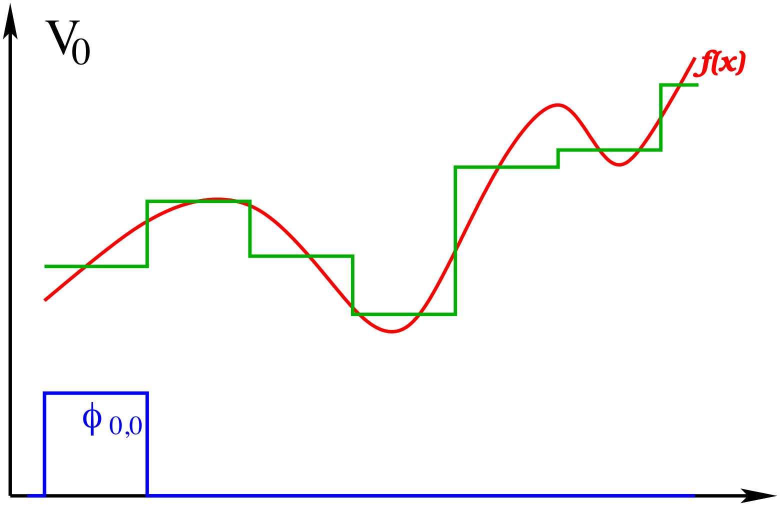
Comments

- Projection on to wavelets and scaling functions can be viewed as an OLS problem. But computations are much faster with this methodology.
- The design matrix is full of convolutions of a_k with itself.
- Computational complexity here is $\mathcal{O}(N)$ c.f. to $\mathcal{O}(N^3)$ for OLS regression.
- In practice data is finite which screws everything up – the maths is prettier than the application. Known as edge effects.

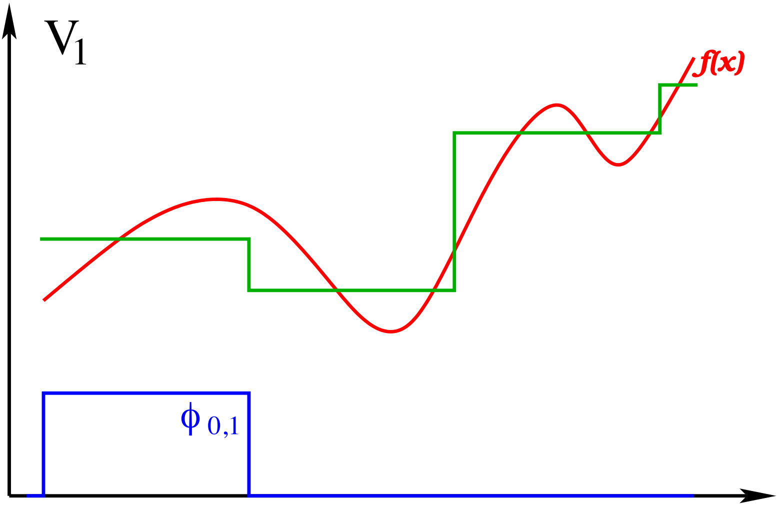
Read this:

https://hal.archives-ouvertes.fr/hal-01311772/document

MRA example



MRA example



MRA example