Filtering

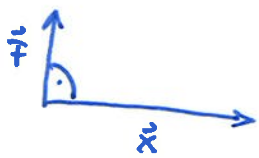
1st April 2019

- Common aims of filtering:
  * finding a signal in the data
  * separating noise and signal
  * compressing the data
  * filters can also be <u>optimized</u>
  * Neural nets essentially have „filters" stored in their layers.

- <u>Linear</u> filters: the general filtering equation
  * data stream $x(t)$
  * Filtering function $F(t)$

$$\boxed{\rightsquigarrow \quad \underbrace{x_F(t)}_{\substack{\text{the filtered data} \\ \text{„filter output"}}} = \int_a^b \underbrace{F(t')}_{\text{the filter}} \; x(t-t') \, \underbrace{dt'}_{\text{sifting through the data}}}$$

- Integrals are <u>linear</u> operators; and most data are discrete $\Rightarrow$ one often sees filters written in terms of linear algebra:
  - $\underbrace{\vec{x}_F}_{\text{Filtered data}} = \langle \vec{F}, \vec{x} \rangle$ — could also be a matrix ($\vec{F}$); not a scalar product (else the output would not be a vector)
  - $S/N = \langle \vec{F} \cdot \vec{x} \rangle^2$ is an often found „signal to noise"



$\Rightarrow \vec{F} \cdot \vec{x} = 0 \rightsquigarrow$ signal $\vec{F}$ not in data $\vec{x}$

$\Rightarrow \vec{F} \cdot \vec{x} > 0 \rightsquigarrow$ signal $\vec{F}$ and data $\vec{x}$ display some similarity

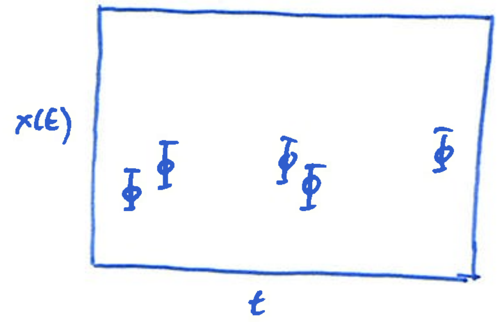
$\rightsquigarrow$ We continue to discuss <u>linear</u> filters.

- Filters can improve (or facilitate) an analysis, if they are sensibly constructed and used.

  * Wiener filter: is a maximum-a-posteriori solution to Gaussian fields under Gaussian shot noise (see script)
  * Sparsity filtering: adds prior information into the analysis, of what „sensible signal shapes" are.

- Sparsity and filtering:

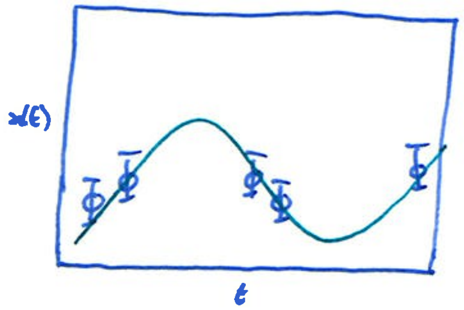
  Imagine 5 data points



→ How much do these 5 points tell us about a potential signal?

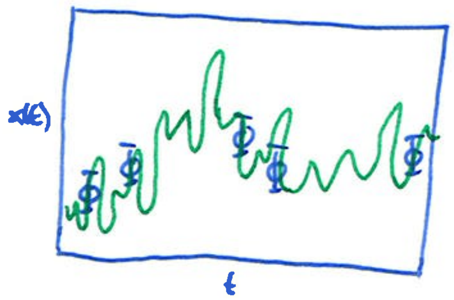
⇒ It depends on how smooth the signal is!

Case I



→ the green signal is very smooth, 5 data points measure it reliably

Case II



→ the green signal is wildly variable and the 5 data points are insufficient to constrain this signal.

→ If we have a priori a sensible reason to believe that there are no high frequencies in our signal, i.e. it looks more like Case I, then putting in this prior belief will improve the analysis.

→ „Wavelet" filters allow to do this.

Example: 'wavelet filtering' as done in gravitational wave research

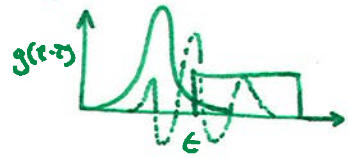
- The usual Fourier transform:

$$S_f(t) = \int_{-\infty}^{+\infty} s(\omega)\, e^{-i\omega t}\, d\omega$$

⇝ a bit extreme: either in ω-space **or** in t-space

- The short-time Fourier transform:

$$S(\tau,\omega) = \int_{-\infty}^{+\infty} s(t)\, \underbrace{g(t-\tau)}\, e^{-i\omega t}\, dt$$



⇝ now we start keeping information in both t- and ω-space.

⇝ Before $e^{\pm i\omega t}$ was our Basis: the Fourier basis.

⇝ Now declare $g(t-\tau)\, e^{-i\omega t}$ our new „basis".

„Basis": a set of functions which projects out <u>interesting</u> elements from the data.

* „over complete basis": one that is not orthonormal, and strictly speaking contains too many basis elements (often adapted to real situations).

- Wavelet transform:

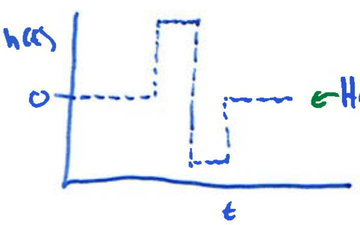
$$\underset{\uparrow\ \text{wavelet } \psi}{f_\psi}(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(t)\, \psi\left(\frac{t-b}{a}\right) dt$$

b is a positional parameter (a location)

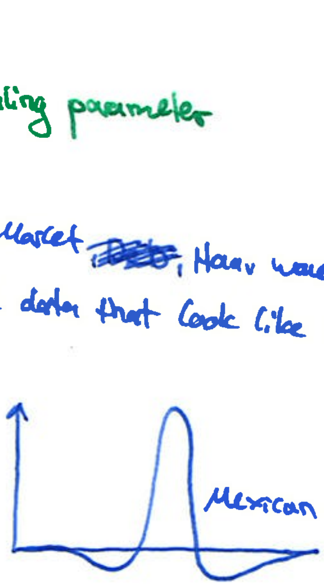
a is a scaling parameter

⇝ there is a multitude of famous wavelets (Gabor, Morlet, ~~[illegible]~~, Haar wavelets...)

⇝ all have in common that they filter patterns from the data that look like the wavelet (because that is what filters do)



← Haar wavelet ⇝ edge detection



Mexican hat filter

← Gabor-Morlet wavelets

Neural Nets
⇝ $\vec{w}_\ell \cdot \vec{\text{input}}_\ell + \vec{b}_\ell$
⇝ a very general filter

The matched filter

* Typically astronomers and physicists already know which signal they are searching for in the data. ⇝ It is then unnecessary to use any ready-made filter such as wavelet filters, since we can as well create our own optimal filter. The most famous of these is the matched filter.

* „optimal": if the sought signal is in the data, then this filter is the most sensitive with respect to it („picks it up the fastest")

* $\vec{f}^T \cdot \vec{x}$ = „apply filter $\vec{f}$ to data $\vec{x}$"

Derivation, matched filter:

- 2 Prerequisites: 1) for optimization: the Cauchy-Schwarz inequality:

$$|\vec{x}^T \vec{y}|^2 \leq (\vec{x}^T \vec{x})(\vec{y}^T \vec{y})$$

2) symmetric positive matrices have a unique matrix root:

$$M = M^{1/2} M^{1/2} \text{ and } M^{1/2} \text{ invertible, such that } M^{1/2} M^{-1/2} = \mathbb{1}$$

- Aim: maximize signal to noise $\langle S/N \rangle$ for additive noise:

$$\underset{\text{data stream}}{\vec{x}} = \vec{s} + \underset{\text{noise}}{\vec{n}} \quad , \quad \langle \vec{n} \rangle = 0 \quad , \quad \langle \vec{n} \vec{n}^T \rangle = \underset{\text{Covariance of noise}}{C}$$

- filtering: $\vec{f}^T \vec{x} = \underbrace{\vec{f}^T \vec{s}}_{\text{correct detections}} + \underbrace{\vec{f}^T \vec{n}}_{\text{false detections (interpreting noise patterns as signal)}}$

⇝ the noise of the filter (which we want to be as low as possible) is then

$$\langle \vec{f}^T \vec{n} \; \vec{n}^T \vec{f} \rangle = \vec{f}^T C \vec{f} =: N \quad \text{(expected false detections)}$$

⇝ the corresponding signal S to this N is then $|\vec{f}^T \vec{s}|^2$ (mind the square).

Our signal to noise is then expected to be:

$$\langle S/N \rangle = \frac{|\vec{f}^T \vec{s}|^2}{\vec{f}^T C \vec{f}}$$

top and bottom measure in different units, adapt those

writing out the metric from the top

$$= \frac{|\vec{f}^T \mathbb{1} \vec{s}|^2}{\vec{f}^T C \vec{f}}$$

use that C is symm pos. def

$$= \frac{|\vec{f}^T C^{1/2} \, C^{-1/2} \vec{s}|^2}{\vec{f}^T C^{1/2} \, C^{1/2} \vec{f}}$$

$$= \frac{|(C^{1/2}\vec{f})^T \mathbb{1} (C^{-1/2}\vec{s})|^2}{(C^{1/2}\vec{f})^T \mathbb{1} (C^{1/2}\vec{f})}$$

now both use the same units

Now we are greedy, and demand that the signal to noise ratio shall satisfy the Cauchy-Schwarz inequality. This is the optimization step.

$$\stackrel{!}{=} \frac{\cancel{(C^{1/2}\vec{f})^T (C^{1/2}\vec{f})} \quad (C^{-1/2}\vec{s})^T (C^{-1/2}\vec{s})}{\cancel{(C^{1/2}\vec{f})^T (C^{1/2}\vec{f})}}$$

$$= (C^{-1/2}\vec{s})^T (C^{-1/2}\vec{s})$$

$$= \vec{s}^T C^{-1} \vec{s} \quad \Rightarrow \quad \boxed{\langle S/N \rangle \stackrel{!}{=} \vec{s}^T C \vec{s} \text{ after optimization}}$$

Now we have to find a filter which satisfies our demands:

$$\boxed{\langle S/N \rangle = \frac{|\vec{f}^T \vec{s}|^2}{\vec{f}^T C \vec{f}} \stackrel{!}{=} \vec{s}^T C^{-1} \vec{s}}$$

we see: if $C^{-1} = \mathbb{1}$, then we have $\vec{s}^T \vec{s}$ on the right hand side. Hence we need $\vec{f} = A\vec{s}$, otherwise the left side cannot produce a $\vec{s}^T \vec{s}$.

$$= \frac{(\vec{s}^T A)\vec{s} \quad (\vec{s}^T A \vec{s})}{\vec{s}^T A C A \vec{s}} \stackrel{!}{=} \vec{s}^T C^{-1} \vec{s}$$

$\Rightarrow$ $A = C^{-1}$ solves this

$$\boxed{\begin{array}{l} \text{So our sought matched filter is } \vec{f} = C^{-1}\vec{s}, \\ \text{the inverse-variance weighted signal.} \\ \text{also known as „correlating with a inverse variance weighted template"} \end{array}}$$

Matched filtering in the Ligo tutorial

$$\vec{x} = \underset{\uparrow \, GW}{\vec{s}} + \vec{n} \quad , \quad \text{or} \quad \vec{x} = \underset{\uparrow \, \text{no } GW}{0} + \vec{n} \quad \Bigg\} \quad \vec{x}^T = \left(\hat{h}(t_1), \hat{h}(t_2), \dots \hat{h}(t_n) \right)$$

⚠ in the tutorial sheet I use a variable x : $M = x M_\odot$, here I write $M/M_\odot$, since the data are $\vec{x}$ already.

$\rightarrow \vec{f}^T = \left(h(\tau_1, M/M_\odot, \phi_c), h(\tau_2, M/M_\odot, \phi_c), \dots \right)$ searching for correct collapse tim

$\rightarrow \vec{f}^T = \left(h(\tau_1, M'/M_\odot, \phi_c), h(\tau_2, M'/M_\odot, \phi_c) \dots \right)$ searching for correct mass M

$\rightarrow \vec{f}^T = \left(h(\tau_1, M'/M_\odot, \phi_c'), \dots \right)$ searching for correct ϕ_c

→