Filters

- FIR (Finite Impulse Response) IIR (Infinite Impulse Response)

LTI (Linear time Invariant) filters and morphological filters such as median filter

- **LTI** $\Rightarrow$ $a_1 * x(t_i) + a_2 * x_2(t_i) \rightarrow a_1 * y_1(t_i) + a_2 * y_2(t_i)$
- **FIR** $\Rightarrow$ output view of FIR filters, represented by: $y(n) = \sum_{i=0}^{k} w_i * x(n-i) = w_0 * x(n) + w_1 * x(n-1) + w_2 * x(n-2) + \cdots w_k x(n-k)$ (5.6)

this can be seen as moving window filter

5.6 also called as convolution of signal x w/ convolution kernel w, and is then written as

$$y(n) = w(k) * x(n)$$

Application of an FIR filter w/ weights w is equivalent to a convolution of the signal w/ w.

- Infinite Impulse Response (IIR)

While the output of a FIR filter only depends on the incoming signal (5.6), the general output of a filter may also depend on the m most recent values of the output signal:

$$y(n) + a_1 y(n-1) + \cdots + a_m y(n-m) = b_0 x(n) + b_1 * x(n-1) + \cdots + b_k x(n-k)$$

$$\sum_{j}^{m} a_j y(n-j) = \sum_{i=0}^{k} b_i * x(n-i) \quad (5.12) \quad \boxed{a_0 = 1}$$

Ex) Exponential averaging filter

$y_n = \alpha * y_{n-1}$ for $\alpha > 1$ $\uparrow$ exp $\alpha < 1$ $\downarrow$ exp. (colab - Signal.ipynb)

- Difference between FIR and IIR Filters

FIR $\rightarrow$ simple, IIR $\rightarrow$ sharper freq response,

- morphological filters

FIR and IIR are both linear filters. Good at eliminating noise for Gaussian dist, it fails for other tasks e.g. for removing extreme outliers. For such tasks "morphological filters" can be used. Uses data features such as the min, max, median, range of the elements w/ a data window. For removing extreme outliers, a median filter can be used.

Filter Characteristics

to characterize the effect of an LTI filter on a given input signal: Impulse, step, freq (response)

- Freq response

Important property of LTI $\rightarrow$ you give sine input. it gives sine input w/ same frequency w/ only phase and/or amplitude modified.

Amp and phase can be expressed as a single complex #: Amplitude $\rightarrow r$ phase $\rightarrow \theta$

- Artifacts in Casual and Non-Casual Filters

— Casual Filters → depends only past and present. → real time data

— Noncasual filters → depends past, present and future (predicts) data → we already have if we collected

  - Centered Analysis window

$$y_n = \sum_{m=-k}^{k} w_m x_{n+m}$$

[Applications] Smoothing, differentiation and integration of signals

- Savitzky - Golay Filter

Smooths data w/ best fit q order poly for ~~...~~ specified window length.

~~...~~ We can use it as n^{th} order differentiator. Adv and disadv. are in Signal.ipynb.

- Smoothing of Regularly Sampled data.

We saw → Moving avg, Exp avg, Median, Savitzky-Golay filters

  - Butterworth Low-pass Filter

designed to have as flat a freq response as possible in the pass band. ("maximally flat magnitude filter")

— Differentiation

  - 1st difference differentiation

$y(n) = \Delta x / \Delta t = [x(n) - x(n-1)] / \Delta t$ this gives the filter weights for an FIR-filter $w = [1, -1]/\Delta t$

  - Central-Difference diff (for offline analysis)

$w = [1, 0, -1] \times 1/2 \times \Delta t$

  - Cubic diff.

$w = [1, -8, 0, 8, -1] \times 1/12 \times \Delta t$

Others: Lanczos, Parks-McClellan

— Integration

$x(t_n) = x_0 + \Delta x_1 + \Delta x_2 + \dots + \Delta x_n$ $\quad vel(t_i+1) \approx vel(t_i) + acc(t_i) \times \Delta t$

$x(t_{i+1}) = x(t_i) + vel(t_i) \times \Delta t + acc(t_i) \times \Delta t^2/2$

[Smoothing of Irregularly Sampled Data] (non-equal spacing)

— Lowess and ~~Loess~~ Smoothing

1D data → "Loess" filters and "Lowess". → k-nearest neighbour based meta model

In short, one specifies the percentage of adjacent data to be included for these data, a weighted linear regression is applied. The weight funct used for these filters is tri-cube weight function

$$\rightarrow w(x) = (1 - |x|^3)^3 \; I[|x| < 1]$$

$I[\dots]$ is indicator function, indicating the range over which the funct. argument is True.

Lowess uses a linearly poly, loess uses quadratic poly.

— Splines

Def: ~ spline funct is piecewise poly funct of degree < q in a variable x.

q : degree of the spline $q+1$: order of the spline

• B-Splines, poly 2-D and 3-D trajectories are so called B-splines. (Basis splines)

For a given trajectory, the spline knots seperate the piecewise poly parts of the trajectory. Knots are equidistant: cardinal B spline, of B-spline of degree p ($p \in \mathbb{N}_0$), the convolution operator $*$, and the indicator funct $b^0 = I_{[0,1)}$ of the half open interval, the corresponding cardinal B splines is given by

$$b^p : \underbrace{I_{[0,1)} * \cdots * I_{[0,1]}}_{p+1\text{-times}}$$

Note that B-splines have minimal support.

-Kernel Density ~~Function~~ Estimation

You have discrete data but wanna obtain smooth prob density funct, use this.

For a KDE of 10 data, each sample is multiplied w/ Gaussian funct = $g(x) = \exp[-\frac{1}{2}(\frac{x-\mu}{\sigma})^2] / \sigma\sqrt{2\pi}$

{Filtering ~~2D~~ images} (2D filters)

• 2D filtering $y(n,m) = \sum_{i=-2}^{2} \sum_{j=-2}^{2} w_{ij} * x(n+i, m+j)$

The moving window interpolation still holds.

• Morphological Filters for 2D

- Erosion and Dilation of Images

for linear filters → order does not matter.

for morphological operations we have to set structural element SE which is an area w/ a well defined shape around the point M.

def of action erosion $\mathcal{E}$ is:

$$\mathcal{E}(M) = \begin{cases} 0, & \text{if } \sum_{i=0}^{n} \sum_{j=0}^{m} se_{ij} < 9, \text{ w/ } se_{ij} \in SE(M) \\ 1, & \text{else} \end{cases}$$

n and m height and width of the structural element, respectively.

for dilation

$$D(M) = \{ 1, \text{ if } \sum_{i=0}^{n} \sum_{j=0}^{m} se_{ij} \geq 1, \text{ with } se_{ij} \in SE(M)$$

• Opening and Closing

opening = dilation ∘ erosion   closing = erosion ∘ dilation