Signal Processing
from Fourier to Machine Learning
Part 4 : Signal Representation

R. Flamary

November 21, 2024

Full course overview

Course overview

Signal representation

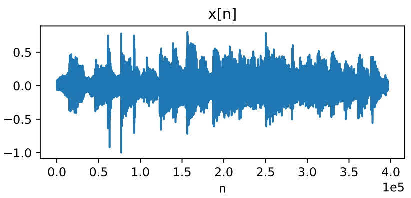
How to look at the signal

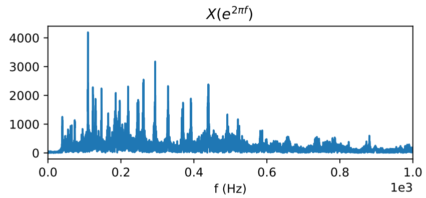
- The raw signal is a function of time (or space, or both).
- But temporal representation can be limited $\rightarrow$ Fourier domain.
- Fourier frequency representation is often pertinent but loses all temporal information.
- Other representations (linear of non-linear) can allow for better interpretation/processing.

Signal representations

- Change of bases (Fourier Domain).
- Global VS local representations (Short Time FT, wavelets).
- Linear decomposition or approximation of the signals.
- Non linearity (energy with a square, kernels, neural networks).

Non stationary signals





Stationarity

- Stationary stochastic processes have probabilistic properties that do not depend on time.
- Reasonable assumption for noise, or some structure/regular signals in telecommunications.
- Most real life signals are NOT stationary (voice, images).

Solution : locality

- Use a representation that focuses on local properties of the signal.
- Locally one can suppose the signal is stationary.
- For temporal signal this means focus on a temporal windows.
- For images it means focus on a small patch of the image.

Window function

Definition

- A window function (or apodization function) is a function used to reweight a signal in order to focus on a given time interval of the signal.
- The signal x windowed by w can be expressed as

$$x_w(t) = x(t)w(t) \quad (1)$$

Properties of a window function

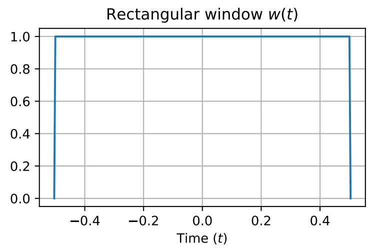
- Window functions are symmetric (real FT) and we suppose that $w \in L_2(\mathbb{R})$.
- Window functions are centered in 0:

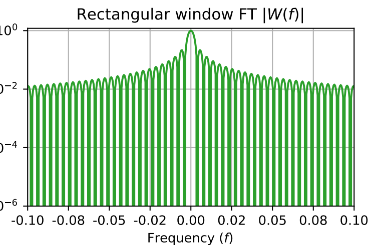
$$\int_{-\infty}^{\infty} t|w(t)|^2 dt = 0 \quad (2)$$

- For a window function $w(t)$ of support $[-1/2, 1/2]$ we can recover a window function for a finite signal of N samples:

$$w[n] = w\left(\frac{(n-(N-1)/2)}{N}\right) \quad (3)$$

Common window functions (1)



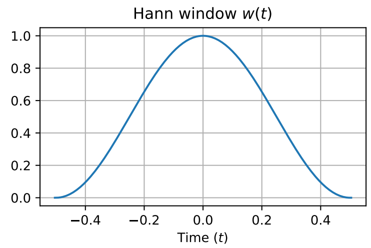


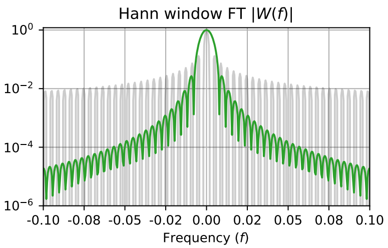
Rectangular window

$$w(t) = \begin{cases} 1 & \text{for } |t| < \frac{1}{2} \\ 0 & \text{else} \end{cases}, \qquad w[n] = \begin{cases} 1 & \text{for } 0 \leq n < N \\ 0 & \text{else} \end{cases}$$

- Corresponds to a selection of a signal on $[-1/2, 1/2]$ (or $0, \ldots, N-1$).
- Can be used to model a finite time recording of a signal.
- In the Fourier domain, it means that the FT of the signal is convolved by a cardinal sine (loss of frequency resolution).

Common window functions (2)



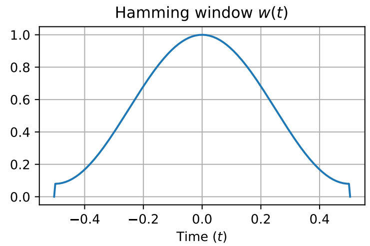


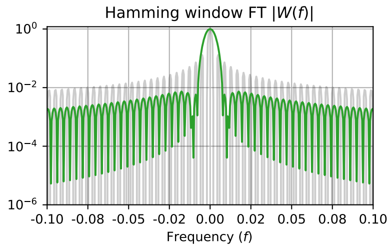
Hann window

$$w(t) = \begin{cases} \frac{1}{2}\left(1+\cos\left(2\pi t\right)\right) = \cos^2\left(\pi t\right), & |t| \leq 1/2 \\ 0, & |t| > 1/2 \end{cases}$$

- Named after meteorologist Julius von Hann.
- Erroneously named "Hanning" due to its use as a verb in some references.
- Far quicker decrease of the lobes in frequencies, but larger principal lobe.

Common window functions (3)



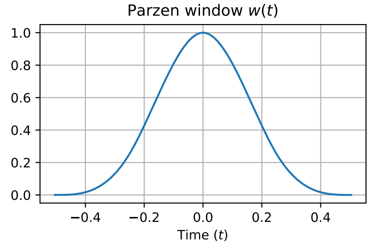


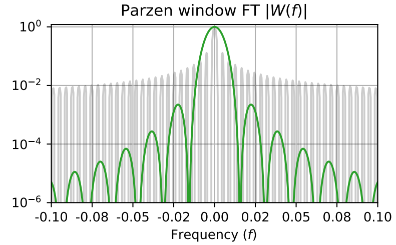
Hamming window

$$w(t) = \begin{cases} \frac{25}{46} + \frac{21}{46}\cos(2\pi t), & |t| \leq 1/2 \\ 0, & |t| > 1/2 \end{cases}$$

- Proposed by Richard W. Hamming to cancel the first sidelobe.
- Similar shape than the Hann window but with a bias (non-zero borders).
- Also called the Hamming blip when used for sound effects.
- Far quicker decrease after principal lobe then slow decrease (near equiripple).

Common window functions (4)



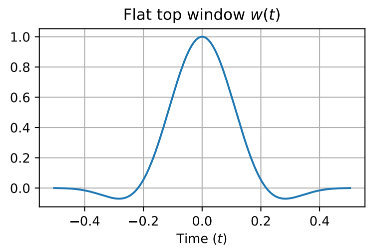


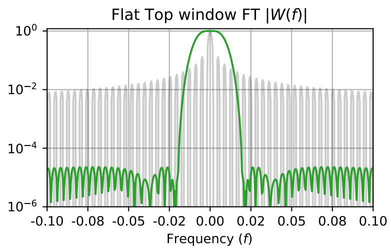
Parzen window

$$w(t) = \begin{cases} 1 - 24t^2(1 - 2|t|), & 0 \leq |t| \leq \frac{1}{4} \\ 2(1 - 2|t|)^3 & \frac{1}{4} < |t| \leq \frac{1}{2} \end{cases}$$

- Also called Parzen (de la Vallée Poussin).
- Approximation of a Gaussian with Spline of order 4.
- Quick decrease in frequency and larger sidelobes than other windows.

Common window functions (5)





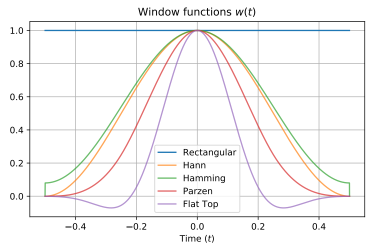
Flat Top window

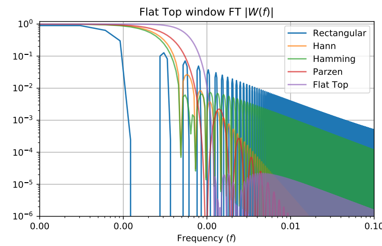
$$w[n] = \begin{cases} a_0 - a_1\cos\left(\frac{2\pi n}{N}\right) + a_2\cos\left(\frac{4\pi n}{N}\right) - a_3\cos\left(\frac{6\pi n}{N}\right) + a_4\cos\left(\frac{8\pi n}{N}\right) & 0 \leq n < N \\ 0 & \text{else} \end{cases}$$

with coefficients: $a_0 = 0.21557895$; $a_1 = 0.41663158$; $a_2 = 0.277263158$; $a_3 = 0.083578947$; $a_4 = 0.006947368$.

- Very large main lobe but very attenuated and equiripples sidelobs.
- Good estimation of frequency components magnitude but low frequency resolution.
- Several other formulations designed from ideal low pass filter approximation.

When to use window function?





Applications of window functions

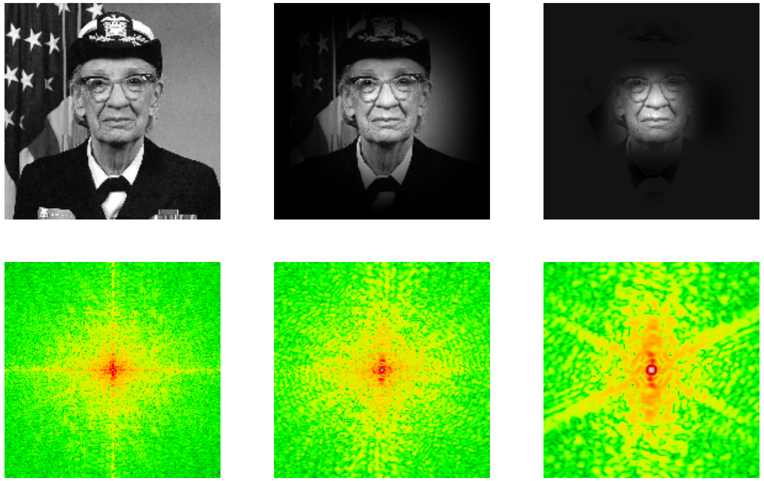
- Focus one one given temporal window centered on u of the signal:

$$x(t)w(t-u)$$

- Minimizing Border effects on finite signals (FFT demo).
- Analog apodization for canceling sidelobs (astronomy).

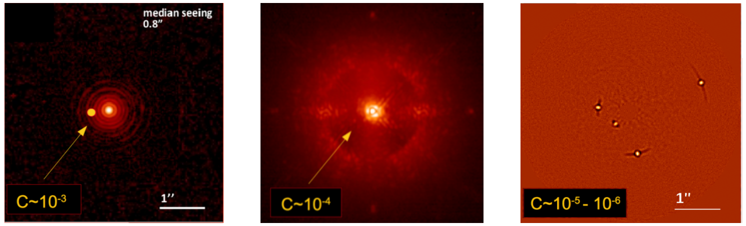
Border effects in images

Orig. image | Hann window | Flat top window



Windowing removes border effect but leads to a loss in frequency resolution.

Apodization in astronomy

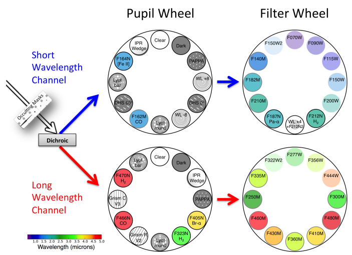


Windowing for a telescope

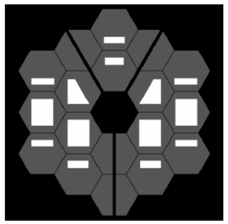
- Apodization literally stands for "removing the foot" in reference to the side lobs of classical apertures.
- Especially important for exoplanet imaging where the exoplanet might be lost in the lobes of its star (10^{-5} relative magnitude).
- Estimation of optimal window function for circular aperture telescope [Soummer et al., 2003].
- Optimal apodization can be done for any aperture shape [Carlotti et al., 2011].

Images courtesy of F. Cantalloube and M. N'Diaye

Apodization for the James Webb Space Telescope

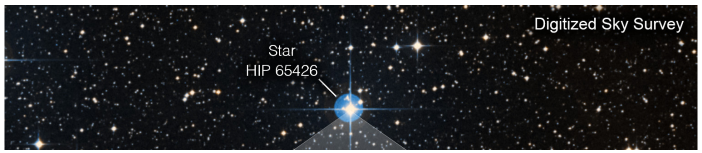


(c) Round occulter Lyot stop



(d) Bar occulter Lyot stop

- The James Webb Space Telescope (JWST) is a space telescope that will be launched in 2022.
- It includes a coronagraph for exoplanet imaging that can be selected by a wheel of different masks.
- Two shapes of masks are available: Round and Bar occulter.



Short Time Fourier Transform (STFT)

Definition

The short time Fourier transform associated to the window function w can be expressed as

$$X_w(u,f) = \mathcal{STF}_w[x(t)] = \int_{-\infty}^{\infty} x(t)w(t-u)e^{-2i\pi ft}dt = \mathcal{F}[x(t)w(t-u)] \tag{4}$$

- We define the basis function $w_{u,f}$ as

$$w_{u,f}(t) = w(t-u)e^{2i\pi ft}$$

- It is localized both in frequency f and time u.
- The STFT can be expressed as a scalar product

$$X_w(u,f) =< x, w_{u,f} >= \int_{-\infty}^{\infty} x(t)w^*_{u,f}(t)dt$$

Temporal and frequency variance

We investigate the time and frequency resolution of the STFT.

Temporal variance

The temporal variance of the basis function $w_{u,f}$ can be expressed as

$$\sigma_t^2 = \frac{1}{\|w\|^2}\int_{-\infty}^{\infty}(t-u)^2|w_{u,f}(t)|^2dt = \frac{1}{\|w\|^2}\int_{-\infty}^{\infty}t^2|w(t)|^2dt \tag{5}$$

It does not depend on time u or frequency f.

Frequency variance

The FT of of the basis function w_{u,f_0} can be expressed as

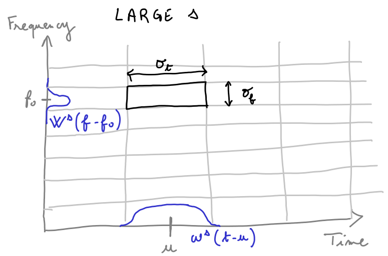
$$W_{u,f_0}(f) = \mathcal{F}[w(t-u)e^{2i\pi f_0 t}] = e^{-2i\pi f u}W(f)\star\delta(f-f_0) = e^{-2i\pi(f-f_0)u}W(f-f_0) \tag{6}$$

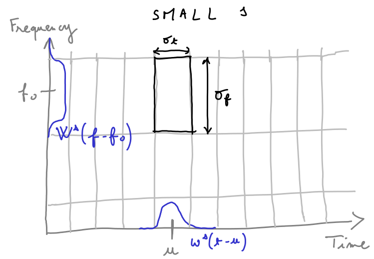
This means that the frequency variance of W_{u,f_0} is

$$\sigma_f^2 = \frac{1}{\|W\|^2}\int_{-\infty}^{\infty}(f-f_0)^2|e^{-2i\pi(f-f_0)u}W(f-f_0)|^2df = \frac{1}{\|W\|^2}\int_{-\infty}^{\infty}t^2|W(f)|^2df \tag{7}$$

Which again does not depend on u or f_0.

Uncertainty principle (1)



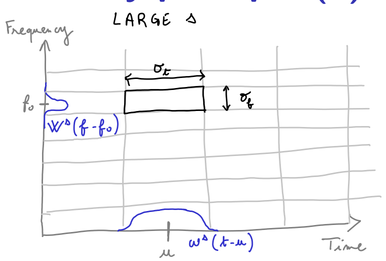


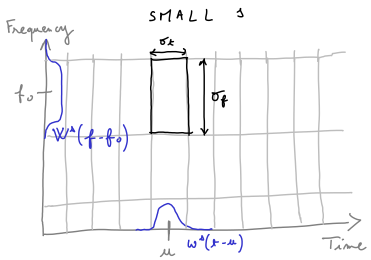
Scaling the window function with $s > 0$

$$w^s(t) = \frac{1}{\sqrt{s}}w\left(\frac{t}{s}\right), \quad \|w\|^2 = \|w^s\|^2$$

- The TF of w^s is : $W^s(f) = \sqrt{s}W(sf)$.
- Small values of s leads to small support of w_s but with large support for W^s (and vice versa).
- The time/frequency is sampled regularly (σ_t and σ_f are independent from u, f_0)
- One cannot have simultaneously a good precision in time and frequency!

Uncertainty principle (2)





Heisenberg-Gabor uncertainty (discussed in [Ricaud and Torrésani, 2014])

Let $w \in L_2(\mathbb{R})$ be a window function with both the function and its FT centered in 0:

$$\int_{-\infty}^{\infty}t|w(t)|^2dt = \int_{-\infty}^{\infty}f|W(f)|^2df = 0$$

then the variances σ_t and σ_f satisfy the following

$$\sigma_t^2\sigma_f^2 \geq \frac{1}{16\pi^2}. \tag{8}$$

The inequality above becomes an equality only for a Gaussian window function of the form

$$w(t) = ae^{-bt^2}$$

Inverse Short Time Fourier Transform

Inverse STFT

The signal x can be reconstructed for $w(t)$ such that $\|w\|^2 = \int_{-\infty}^{\infty}w(t)^2dt = 1$ with:

$$x(t) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}X_w(u,f)w(t-u)e^{2i\pi ft}dudf \tag{9}$$

- For a window function that is not normalized the inverse is scaled by $\frac{1}{\|w\|^2}$.
- Note that the basis functions are NOT orthogonal in this case.
- There also exists an energy preservation formula such that for $\|w\|^2 = 1$ we have

$$\int_{-\infty}^{\infty}x(t)^2dt = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}|X_w(u,f)|^2dudf$$

- This formula justifies that one looks at $|X_w(u,f)|^2$ as a spectral energy density (see spectrogram later).

STFT on discrete signals

Discrete STFT

For a finite signal $x[n]$ of N samples supposed periodic the DSTFT can be computed as

$$X_w[m,k] = \sum_{n=0}^{N-1} x[n]w[n-m]e^{\frac{-i2\pi kn}{N}} \quad (10)$$

- The matrix $X_w[m,k]$ can be computed with N FFT of size N with a complexity $\mathcal{O}(N^2 \log_2(N))$.
- For a window $w[n]$ of small support $M < \log_2(N)$ direct computation can be more efficient.
- For a rectangular window the DSTFT can be computed in $\mathcal{O}(N^2)$.
- In practice reconstruction can be done with a larger temporal sampling of $X_w[m,k]$ as long as the "nonzero overlap add" (NOLA) condition si respected.

Scipy `scipy.signal.stft` function

- `window` is the type of window function.
- `nperseg` is the length of the window M.
- `overlap` is overlap between windows ($M-1$ for DSTFT above).
- `nfft` is the size of the FFT (0 padding if `nfft>nperseg`).

Spectrogram

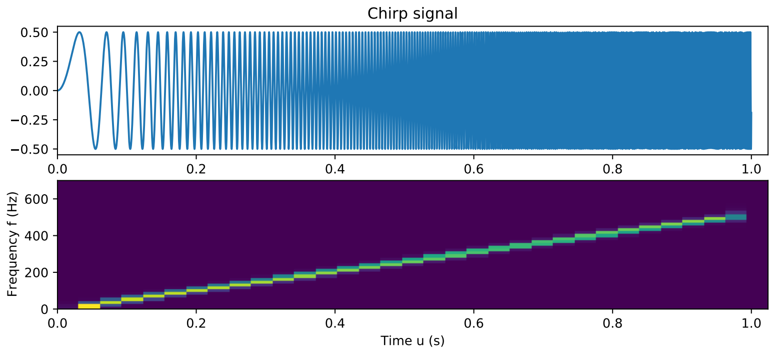
Definition

The spectrogram of a signal is the squared modulus of its STFT. For a signal $x(t)$ of STFT $X_w(u,f)$ the spectrogram can be expressed as

$$S_w(u,f) = |X_w(u,f)|^2$$

- The spectrogram represent the distribution of energy in the time/frequency domain.
- It can be used to visualize (as an image) the evolution of the frequency content of a signal.
- Good tool for interpretation of non-stationary signal.
- Due to the modulus, the phase information is partly lost and one cannot reconstruct a signal from the spectrogram only.
- Methods that perform processing of the spectrogram usually use the Phase of the STFT for reconstruction.

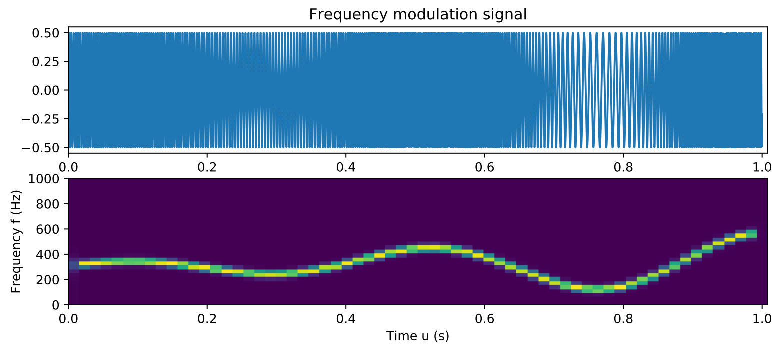
Examples of spectrograms (1)



Chirp signal

- One second signal, sampled at 8KHz.
- Starts at frequency 0 and ends at frequency $500Hz$.
- Window size of $M = 512$, overlap at 50%.

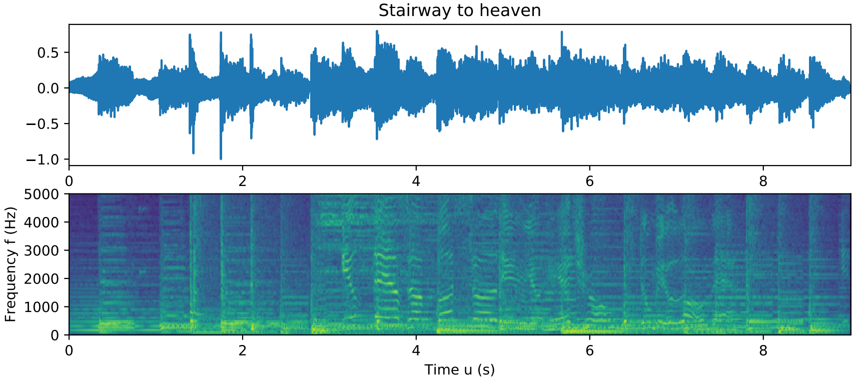
Examples of spectrograms (2)



Frequency modulation signal

- One second signal, sampled at 8KHz.
- Signal instantaneous frequency changes between 100 and $600Hz$
- Window size of $M = 256$, overlap at 50%.
- The peaks in the spectrogram follow the frequencies along time.

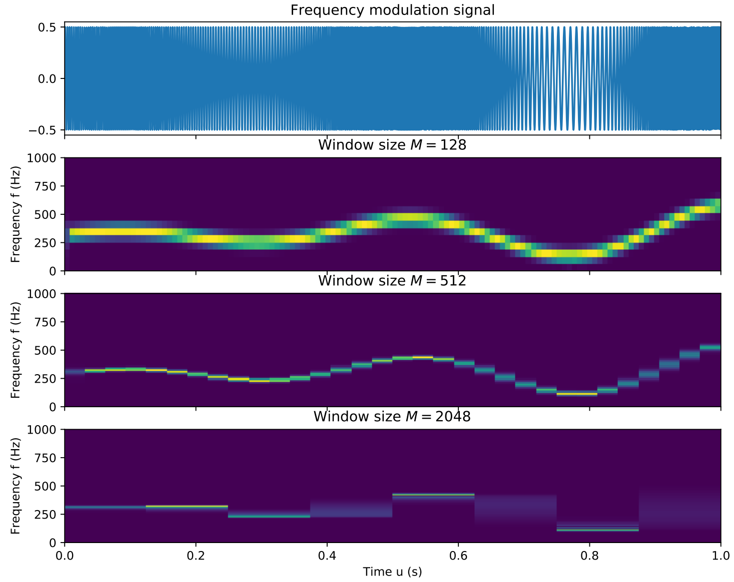
Examples of spectrograms (3)



Real music signal

- Excerpt from "Stairway to heaven".
- 9 sec signal with 44100Hz sampling, window of size $M = 1024$.
- The peaks in the spectrogram follow the frequencies along time.
- Regular harmonics are notes from the guitar, vertical lines are drum, harmonics with variation along time are due to the voice of the singer.

Effect of the window size (uncertainty)



Demos

Periodogram method for PSD estimation

Principle

- PSD estimation can be done for finite random signal realizations from empirical autocorrelation and square of FFT of the signal.
- Those estimations are noisy and sometimes hard to interpret.
- Periodogram method estimate a PSD from the spectrogram
- Welch's method [Welch, 1967] propose to average the spectrogram :

$$\hat{S}_x(f) = \int |X_w(u,f)|^2 du$$

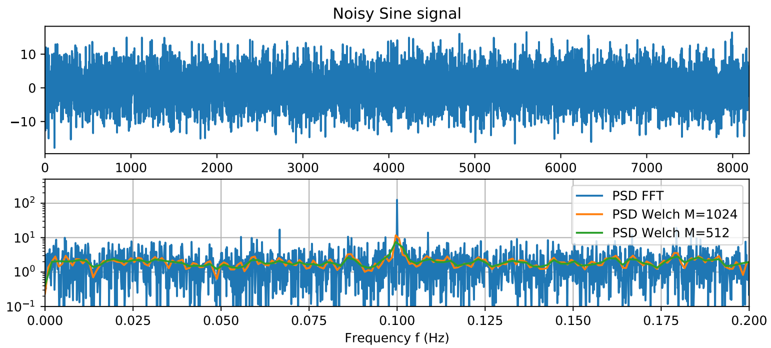
- It reduces the estimation noise of estimation of the D in exchange for a loss in frequency resolution.

Scipy `scipy.signal.welch` periodogram function

- `window` is the type of window function.
- `nperseg` is the length of the window M.
- `overlap` is overlap between windows ($M/2$ by default).
- `nfft` is the size of the FFT (0 padding if `nfft>nperseg`).

One can also use `scipy.signal.periodogram` (Bartlett method with `overlap=0`).

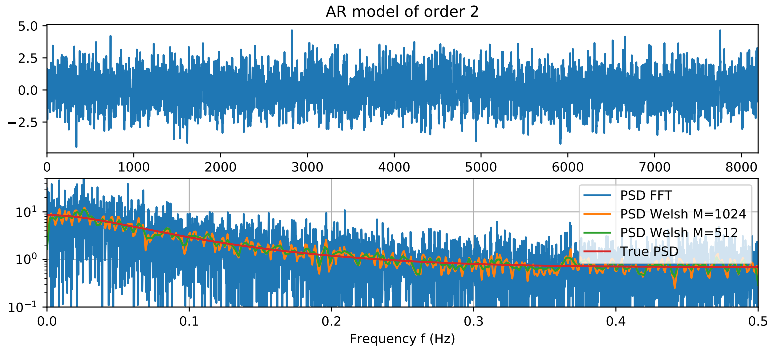
Examples of periodogram (1)



Noisy sine

- Signal containing a sine at frequency $f_0 = 0.1$ with Gaussian IID noise.
- FFT PSD estimation and Welsh periodogram estimation for $M = 1024$ and $M = 512$.
- The noise density is less noisy (near constant).
- The magnitude of the peak at f_0 is smaller (energy is spread due to windowing).

Examples of periodogram (2)



AR model

- Simulate an AR model of order 2.
- FFT PSD estimation and Welsh periodogram estimation for $M = 1024$ and $M = 512$.
- The smoothed Welch periodogram estimation is much closer to the true PSD.

Filtering the STFT with spectral subtraction

Noise suppression in the time frequency domain [Boll, 1979]

1. Compute the STFT $X_w[m,k]$ of the signal $x(t)$.
2. Apply a thresholding operator with $\lambda > 0$ to its magnitude:

$$|\hat{X}_w[m,k]| = \max(0, |X_w[m,k]| - \lambda)$$

3. Reconstruct the denoised signal with

$$\hat{x}(t) = \mathcal{STF}_w^{-1}[|\hat{X}_w[m,k]|e^{iArg(X_w[m,k])}]$$

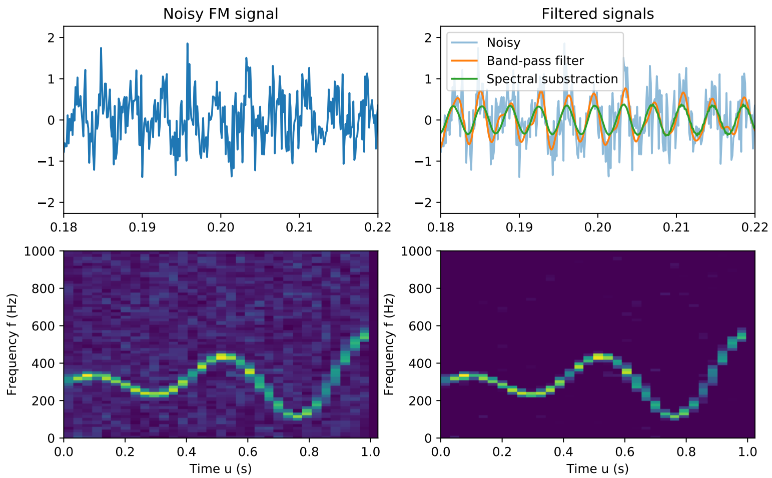
Discussion

- Use the thresholded magnitude and original phase and perform inverse STFT..
- When PSD of noise $P_n[k]$ available one can use is as an adaptive threshold:

$$|\hat{X}_w[m,k]| = \max\left(0, |X_w[m,k]| - \sqrt{P_n[k]}\right)$$

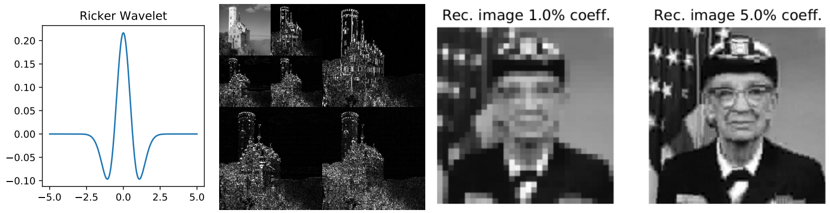
- Thresholding can be done on blocks of STFT coefficients instead of individual [Yu et al., 2008].

Example of spectral subtraction



- FM signal with additive gaussian noise.
- Comparison of bandpass filter and spectral subtraction.

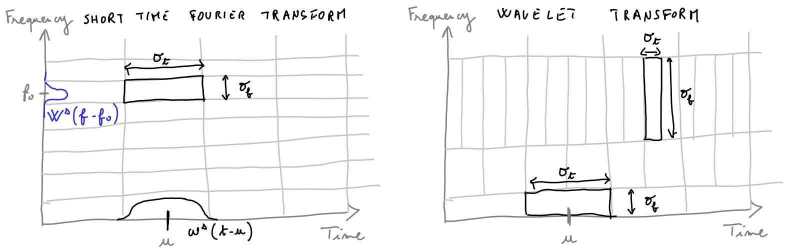
Common signal representations



Signal representation

- Basis of function to represent the signal as a linear combination.
- Wavelets allow spatial/frequency representation with an adaptive time/frequency resolution.
- Discrete Cosine Transform is a non local orthogonal basis used for image compression.
- Sparsity of the signals is used for compressing and signal denoising/reconstruction.

Continuous Wavelet Transform (1)



Definition [Mallat, 1999]

Let $\psi \in L_2(\mathbb{R})$ be the normed ($\|\psi\| = 1$) "mother" wavelet. The Continuous Wavelet Transform (CWT) of the signal $x(t)$ can be expressed as

$$X_\psi(u,s) = \frac{1}{|s|^{1/2}} \int_{-\infty}^{\infty} x(t)\psi^*\left(\frac{t-u}{s}\right) dt \tag{11}$$

- Coefficient u correspond to the time (equivalent to u in STFT).
- Coefficient s is the scale coefficient (indirect equivalence to frequency).
- "Adaptive" resolution in the time frequency representation (uncertainty remains).
- The CWT can be reformulated as a convolution.

Continuous Wavelet Transform (2)

Properties of CWT

- **Shifting** $y(t) = x(t-\tau) : Y_\psi(u,s) = X_\psi(u-\tau,s)$
- **Scaling** $y(t) = \frac{1}{\sqrt{a}}x(\frac{t}{a}) : Y_\psi(u,s) = X_\psi(\frac{u}{a},\frac{s}{a})$
- **Localization** $x(t) = \delta(t-t_0) : X_\psi(u,s) = \frac{1}{\sqrt{s}}\psi\left(\frac{u-t_0}{s}\right)$

Reconstructing the signal

- The real mother wavelet ψ is assumed to respect the *admissibility condition* :

$$C_\psi = \int_0^\infty \frac{|\Psi(f)|^2}{|f|} df < \infty$$

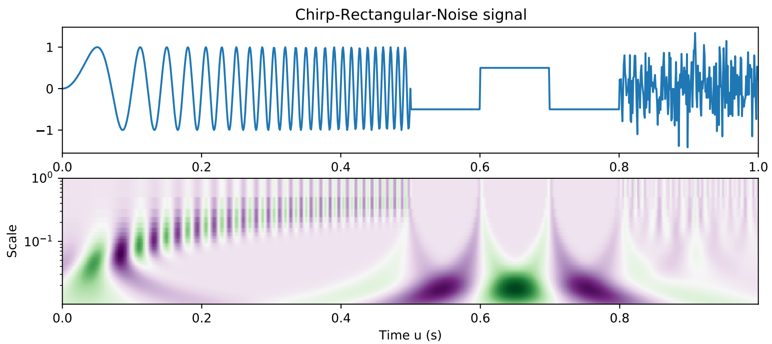
where $\Psi(f) = \mathcal{F}[\psi(t)]$ This condition implies that

$$\Psi(0) = \int_{-\infty}^{\infty} \psi(t)dt = 0$$

- The signal can be reconstructed by using Calderón's reproducing identity:

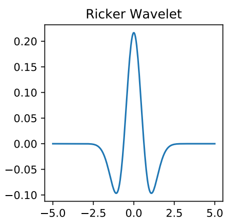
$$x(t) = \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_0^\infty X_\psi(u,s)\Phi\left(\frac{t-u}{s}\right)\frac{1}{s^2} ds du \tag{12}$$

Ricker Wavelet example



Ricker Wavelet also called Mexican Hat

$$\psi(t) = \frac{2}{\pi^{1/4}\sqrt{3}}\left(1-t^2\right)e^{-\frac{t^2}{2}} \tag{13}$$



- Used in Computer vision to detect multiscale edges in images.
- Slow components can be seen at small scale but edges are detected at large scale (quick changes).

Discrete Wavelet Transform

- For a finite sampled signal $x[n]$ with N samples, one can use a discrete version of the wavelet Transform.
- A sufficient sampling to allow reconstruction is the log space of $[-1/N, 1]$ with $s = a_0^k$ for $k \in \mathbb{Z}$ with usually $a_0 = 2$.
- The discrete scaled wavelet can be expressed as

$$\psi_k[n] = \frac{1}{\sqrt{a_0^k}}\psi\left(\frac{n}{a_0^k}\right)$$

- The Discrete Wavelet Transform can be computed as a convolution:

$$X_\psi[m,k] = \sum_{n=0}^{N-1} x[n]\psi_k^*[n-m] = x \star \psi_k^{*-}[m] \tag{14}$$

- When the signal is supposed to be periodic, one can used Fast Convolution with FFT and can compute the $\log_2(N)$ scales on the signals with complexity $\mathcal{O}(N(\log_2(N)^2))$.
- Temporal sampling can also be adapted to the resolution with a decimation depending on the scale leading to a transform of size N.
- Fast computation base on filtering/decimation can be done : Fast DWT [Mallat, 1989].

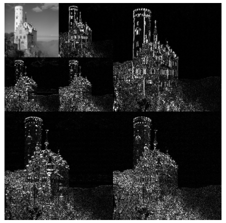
Applications of Wavelet Transforms

Wavelet transform as data transformation

- Data representation : natural signal and images are sparse in the Wavelet domain so easier to interpret.
- Sparsity can also be used for compression and denoising (noise is not sparse).
- Alternative to (Short Time) Fourier Transform in numerous applications (less sensible to Gibbs phenomenon).

Some applications

- JPEG2000 image standard [Group et al., 2000].
- Alternative to (Short Time) Fourier Transform in EEG Analysis [Adeli et al., 2003].
- Image deconvolution and reconstruction (see sparsity in the next part).



Discrete Cosine Transform

- Decomposition of discrete signals in Fourier require the use of complex number.
- Complex numbers comes with a price in memory and complexity.
- We want a similar real transform that remain interpretable in terms of frequency.
- We also want to limit the border effects for non periodic signals.

$\rightarrow$ Discrete Cosine Transform (DCT) [Ahmed et al., 1974]

Symmetrization of the signal (for variant DCT-II)

- Let $x[n]$ be a finite signal with N samples.
- We use a symmetric version (around -1/2) of signal x of size $2N$ such that

$$\tilde{x}[n] = \begin{cases} x[n] & \text{for } 0 \leq n < N \\ x[-n-1] & \text{for } -N \leq n < 0 \end{cases} \tag{15}$$

- This symmetrization of the signal allows for a decomposition of the signal of the form

$$\tilde{x}[n] = \sum_{k=0}^{N-1} a_k \cos\left(\frac{2k\pi}{2N}\left(n + \frac{1}{2}\right)\right) \tag{16}$$

Discrete Cosine Transform (2)

Basis of discrete cosines

The family of discrete cosine

$$\left\{ c_k[n] = \lambda_k \sqrt{\frac{2}{N}} \cos\left(\frac{k\pi}{N}\left(n + \frac{1}{2}\right)\right) \right\}_{k=0,\dots,N-1} \quad \text{with} \quad \lambda_k = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } k = 0 \\ 1 & \text{else} \end{cases}$$

is an orthonormal basis of $\mathbb{R}^N$.

Discret Cosine Transform

The discrete cosine transform (DCT) of signal $x[n]$ is

$$X_c[k] = < x[n], c_k[n] > = \sum_{n=0}^{N-1} \lambda_k \sqrt{\frac{2}{N}} \cos\left(\frac{k\pi}{N}\left(n + \frac{1}{2}\right)\right) x[n] \tag{17}$$

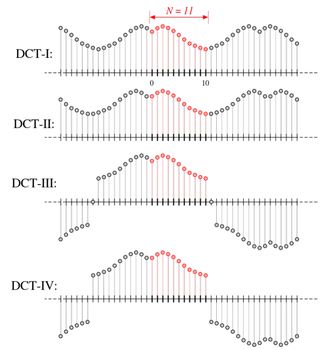
and the signal $x[n]$ can be recovered with

$$x[n] = \sum_{k=0}^{N-1} X_c[k] c_k[n] \tag{18}$$

Discrete Cosine Transform in practice

Implementations

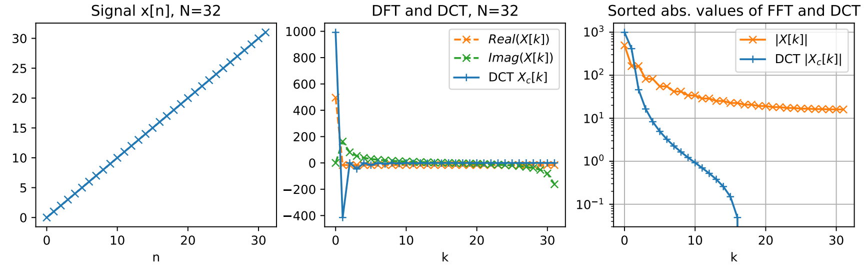
- Several variants of DCT exist with slight differences in the symmetrization process (we saw DCT-II in the course).
- All variants can be computed with an adaptation of the FFT algorithm in $\mathcal{O}(N \log_2(N))$ [Vetterli and Kovacevic, 1995].



DCT in practice

- Extension to 2D bases as product of 1D bases recover Fast transforms.
- Very common in signal/image processing and compression.
- In practice one uses a windowing of the signal in order to get space/frequency representations of the images (multiple DCT on small signal/images).
- Provided in Scipy with function `scipy.fft.dct` (not normalized by default like `fft`) and its inverse `scipy.fft.idct`.

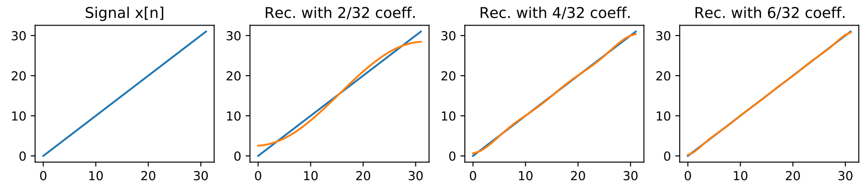
Discrete Cosine Transform 1D example (1)



Range signal example

- FFT supposes that the signal is periodic so it has a large discontinuity in $n = 0$.
- Transformation coefficients are provided in the center of the figure above.
- The right part shows the sorted (decreasing value) modulus values of the coefficients for FFT and DCT.
- We can see that thanks to the symmetrization, the DCT is sparse around 50% (contains 0 components) while FFT representation is not.

Discrete Cosine Transform 1D example (2)

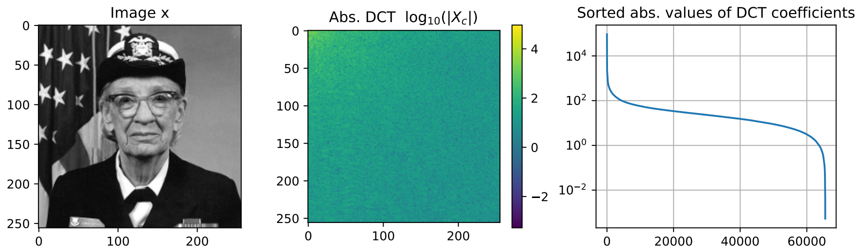


Range signal compression

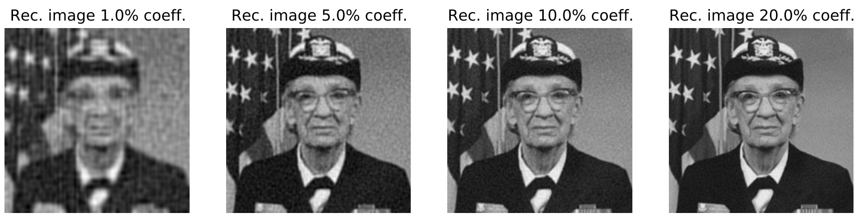
- Compute DCT of the signal.
- Threshold coefficients in order to keep only the largest.
- Reconstruction of the signal after threshold.
- Very good reconstruction from few coefficients.
- Principle used for DCT compression in JPEG.

DCT for JPEG compression (1)

Image representation (Global DCT)

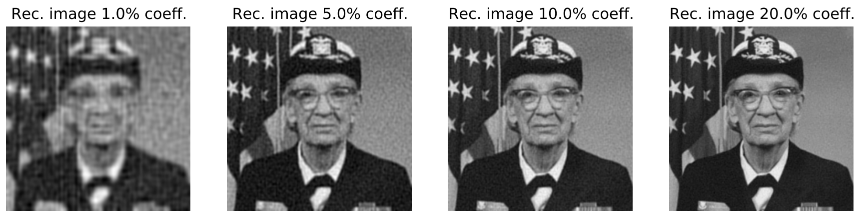


Thresholding + reconstruction (Global DCT)

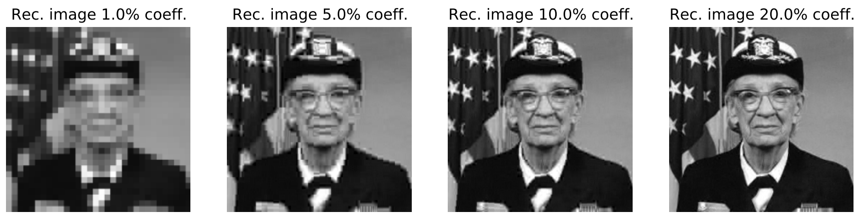


DCT for JPEG compression (2)

Thresholding + reconstruction (Global DCT)



Thresholding + reconstruction (JPEG local 8×8 DCT)



Linear model for finite signals

Finite signal as vector

- A finite signal $x[n]$ can of N samples be represented as a vector

$$\mathbf{x} = [x[0], x[1], \dots, x[N-1]]^T$$

- We suppose that the signals have a finite energy : $\|\mathbf{x}\| < \infty$

Linear model

We supposed in all the previous signal representations that the signal $\mathbf{x} \in \mathbb{R}^n$ can be represented as a weighted sum of basis signals:

$$\mathbf{x} = \mathbf{D}\mathbf{a} = \sum_{j=1}^{m} a_j \mathbf{d}_j \quad (19)$$

- $\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_m] \in \mathbb{R}^{n \times m}$ is the dictionary and the $\mathbf{d}_k$ are the basis vectors.
- $\mathbf{a} \in \mathbb{R}^m$ is the representation of the signal on the dictionary $\mathbf{D}$.
- Note that the discrete Fourier and Cosine Transforms representation have $m = n$ and the basis vectors are orthogonal.

Linear model and approximation

$$\mathbf{x} = \mathbf{D}\mathbf{a} = \sum_{j=1}^{m} a_j \mathbf{d}_j$$

Case $m < n$: Approximation

- The equality is true only when $\mathbf{x}$ is in the span of $\mathbf{D}$.
- When this is not the case one can only approximate the signal.
- Classical way is to find a representation $\mathbf{a}$ that minimizes an error $L(\cdot, \cdot)$ between $\mathbf{x}$ and its reconstruction $\mathbf{D}\mathbf{a}$:

$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{\text{argmin}} \quad L(\mathbf{x}, \mathbf{D}\mathbf{a}) \quad (20)$$

Case $m = n$: Change of basis

- When $\mathbf{D}$ is full rank the change in representation is a change of basis in $\mathbb{R}^n$.
- In this case there is a unique $\mathbf{a}$ such that the equality is true.

Case $m > n$: overcomplete dictionary

- In this case there is a possibly infinite number of $\mathbf{a}$ such that the equality is true.
- Representation used in conjunction with sparsity.

Least square estimation

$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{\text{argmin}} \quad \|\mathbf{x} - \mathbf{D}\mathbf{a}\|^2 \quad (21)$$

Solving the least square estimation when $L(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|^2$

- The solution is a projection on the span of $\mathbf{D}$ such that:

$$\mathbf{D}^T\mathbf{D}\hat{\mathbf{a}} = \mathbf{D}^T\mathbf{x}, \quad \rightarrow \hat{\mathbf{a}} = (\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\mathbf{x} \quad (22)$$

- Already seen for Wiener filtering and in MAP 535 (Regression)
- Requires $\mathbf{D}^T\mathbf{D}$ to be invertible (strictly positive definite) for a unique solution.

Special cases

- **$\mathbf{D}^T\mathbf{D}$ non strictly positive definite** : Add regularization term to find the minimal norm solution by minimizing with $\lambda > 0$:

$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{\text{argmin}} \quad \|\mathbf{x} - \mathbf{D}\mathbf{a}\|^2 + \lambda\|\mathbf{a}\|^2 \quad (23)$$

  with solution $\hat{\mathbf{a}} = (\mathbf{D}^T\mathbf{D} + \lambda\mathbf{I})^{-1}\mathbf{D}^T\mathbf{x}$ where $\mathbf{I}$ is the identity matrix (similar to noise in Wiener filtering).
- **$\mathbf{D}$ is orthonormal basis (Fourier, Cosine)** : $\hat{\mathbf{a}} = (\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\mathbf{x} = \mathbf{D}^T\mathbf{x}$

Sparsity and sparsity promoting regularization

Sparsity

- A sparse vector is a vector that contain a proportion of values exactly 0.
- Most natural signal are not sparse in the time domain but can be sparse (or near sparse) in a given dictionary.
- Usually the presence of noise comes with a loss of sparsity.
- Examples: DCT of images, Wavelet representation.

Sparsity for signal processing

- Can be used to denoise or reconstruct signals with $\mathbf{D}\hat{\mathbf{a}}$ where $\hat{\mathbf{a}}$ is sparse.
- Sparse data is handled efficiently on computers (memory, complexity).
- Better estimation of the few active coefficients (the rest are 0).
- How to use sparsity is signal processing:
  - The easy way: hard thresholding (used in spectrograms and DCT compression).
  - The subtle way : add a regularization term that will promote sparsity.

The Lasso optimization problem

$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{\text{argmin}} \quad \frac{1}{2}\|\mathbf{x} - \mathbf{D}\mathbf{a}\|^2 + \lambda\|\mathbf{a}\|_1 \tag{24}$$

where $\|\mathbf{a}\|_1 = \sum_j |a_j|$ is the L_1 norm of the vector.

- Non smooth objective function (absolute value is non differentiable).
- The non-differentiability in 0 will attract the minimum toward sparse solutions.
- No closed form for solving the problem (except for $\mathbf{D}$ orthogonal).
- Several existing algorithms of complexity $\mathcal{O}(m^3)$.

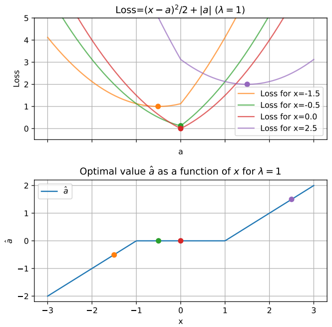
Absolute value and sparsity (in 1D)

$$\hat{a} = \underset{a}{\text{argmin}} \quad (a - x)^2/2 + \lambda|a|$$

The solution is the soft thresholding operator

$$\hat{a} = \max(0, |x| - \lambda)\text{sign}(x)$$

The function above is called the proximal operator of the absolute value.



Signal and image reconstruction with sparsity

Denoising with additive noise (Basis Pursuit [Chen and Donoho, 1994])

$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{\text{argmin}} \quad \frac{1}{2}\|\mathbf{x} - \mathbf{D}\mathbf{a}\|^2 + \lambda\|\mathbf{a}\|_1 \tag{25}$$

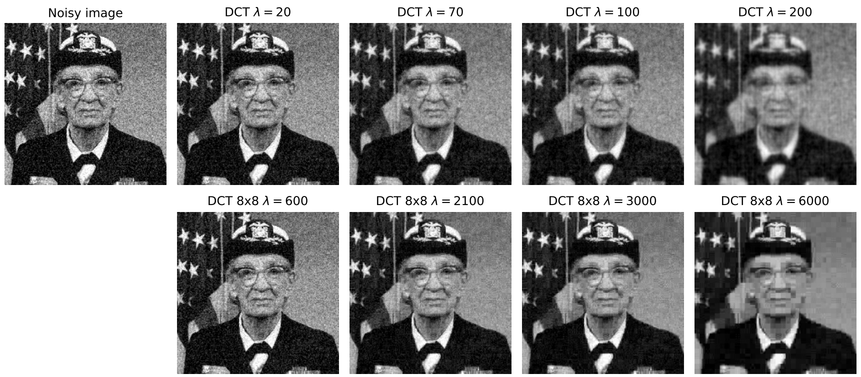
- Original signal $\mathbf{y}$ is sparse, additive IID noise $\mathbf{w}$ is not and $\mathbf{x} = \mathbf{y} + \mathbf{w}$.
- λ has to be chosen *w.r.t.* the noise level.
- Estimate the signal with $\hat{\mathbf{y}} = \mathbf{D}\hat{\mathbf{a}}$.

Signal reconstruction

$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{\text{argmin}} \quad \frac{1}{2}\|\mathbf{x} - \mathbf{H}\mathbf{D}\mathbf{a}\|^2 + \lambda\|\mathbf{a}\|_1 \tag{26}$$

- Original signal $\mathbf{y}$ is sparse, additive IID noise $\mathbf{w}$ is not and $\mathbf{x} = \mathbf{H}\mathbf{y} + \mathbf{w}$.
- $\mathbf{H}$ is a known linear operator (LTI system, convolution, ...).
- When $\mathbf{H}$ is a convolution operator it is a Toeplitz matrix (block-Toeplitz in 2D).
- Estimate the signal with $\hat{\mathbf{y}} = \mathbf{D}\hat{\mathbf{a}}$.

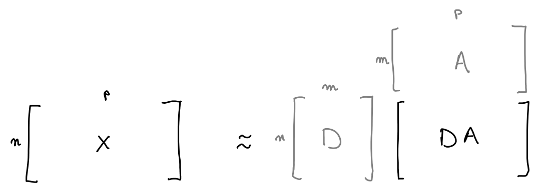
Denoising images with sparsity



Denoising with sparsity in the DCT decomposition

- Noisy image with IID Gaussian noise.
- Reconstructed by solving Equation (25) with different values of λ.
- Comparison between $\mathbf{D}_{DCT}$ corresponding to the Full DCT decomposition (top) and $\mathbf{D}_{DCT8\times8}$ for a local decomposition on 8×8 patches (bottom).

Source separation and dictionary learning

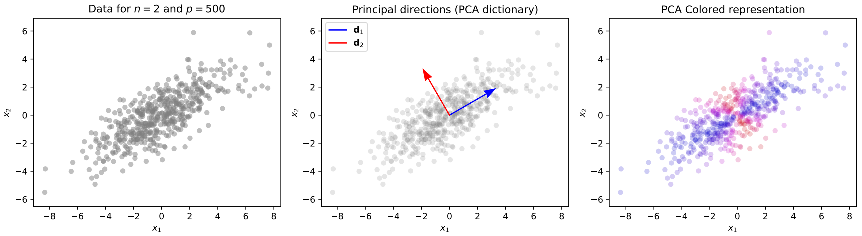


Estimate simultaneously the dictionary $\mathbf{D}$ and the representation $\mathbf{A}$ from the data:

$$\min_{\mathbf{A}\in\mathcal{C}_A, \mathbf{D}\in\mathcal{C}_D} \quad L(\mathbf{X}, \mathbf{D}\mathbf{A}) \tag{27}$$

- $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_p] \in \mathbb{R}^{n\times p}$ is a dataset of (usually centered) p signals $\mathbf{x}_i \in \mathbb{R}^n$.
- $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_p] \in \mathbb{R}^{m\times p}$ contains the representations of all the samples.
- $L(\cdot, \cdot)$ measure the discrepancy between the signals $\mathbf{x}_i$ and their model $\mathbf{D}\mathbf{a}_i$.
- $\mathcal{C}_A$ and $\mathcal{C}_D$ are constraint sets that encode prior knowledge about the data.
- This general approach is know under several names depending on the constraints on the dictionary and coefficients and the loss L.

Principal Component Analysis



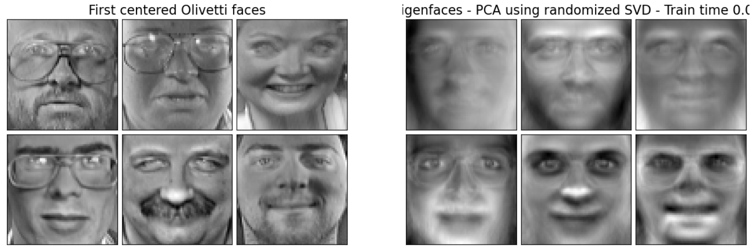
Principle

$$\min_{\mathbf{A}\in\mathbb{R}^{m\times p},\mathbf{D}\in\mathbb{R}^{n\times m},\mathbf{D}^T\mathbf{D}=I_m} \|\mathbf{X}-\mathbf{D}\mathbf{A}\|_F^2 \tag{28}$$

where $\|\mathbf{M}\|_F^2 = \sum_{i,j} M_{i,j}^2$ is the squared Frobenius norm.

- With $m < n$ we seeks for the subspace of $\mathbf{R}^n$ such that $\mathbf{D}$ is orthonormal.
- Solving the problem can be done with a SVD decomposition of matrix $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{W}^T$ and keeping the m largest singular values. The solution is $\mathbf{D} = \mathbf{U}_m$ and $\mathbf{A} = \boldsymbol{\Sigma}_m\mathbf{W}_m^T$.
- Can also be computed from the eigendecomposition of the matrix $\mathbf{X}^T\mathbf{X}$.
- Used to perform Dimensionality Reduction from n to m.
- Denoising of the signals $\hat{\mathbf{x}} = \mathbf{D}\mathbf{a}$ can be done for IID noise (isotropic).

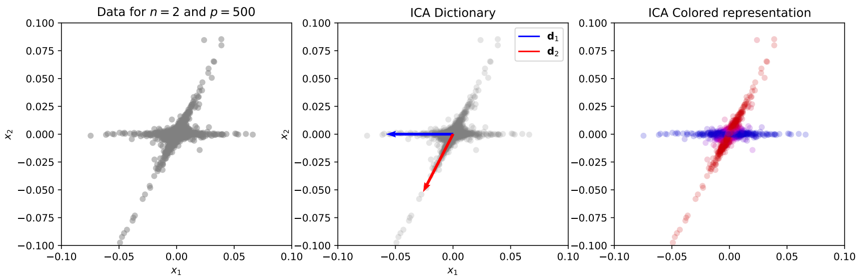
Application of PCA : Eigenfaces



Principle [Sirovich and Kirby, 1987]

- Use dataset of human faces (centered).
- PCA is performed in order to recover the eigenvector of the faces dataset.
- Can be used for representation (face recognition) or for reconstructing missing data [Turk and Pentland, 1991] or data generation.
- Original GAN : "This person does not exist" .

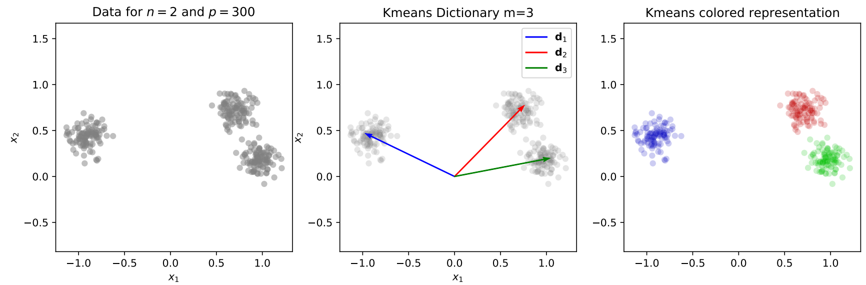
Independent Component Analysis



Principle [Herault and Jutten, 1986]

- Find a decomposition of the signal that is independent (as opposed to orthogonal for PCA).
- Not expressed as the general optimization problem (27) but still linear model.
- Works particularly well on non Gaussian data (or else PCA is optimal).
- Efficient algorithm : FastICA [Hyvärinen and Oja, 2000].
- Applied with success to several source separation problems (biomedical signal processing).

Vector Quantization (K-means)

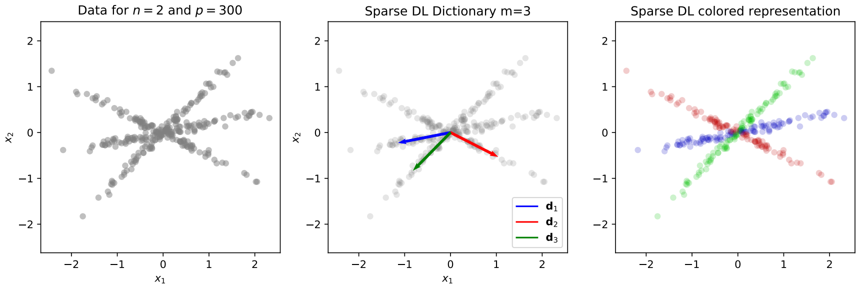


Principle [MacQueen et al., 1967]

$$\min_{\mathbf{A}\in\{0,1\}^{m\times p},\mathbf{D}\in\mathbb{R}^{n\times m},\sum_j A_{j,i}=1,\forall i} \|\mathbf{X}-\mathbf{D}\mathbf{A}\|_F^2 \tag{29}$$

- Find m dictionary element (clusters) that represent the dataset.
- The representation $\mathbf{a}_i$ for one signal an be only binary with a unique active component at one (each signal is represented only by its closest $\mathbf{d}_j$).
- Solved classically with the K-means (block coordinate descent):
  1. Update $\mathbf{D}$ by computing an average of the signals assigned to each cluster.
  2. Update $\mathbf{A}$ by finding the closest cluster for each signal.

Sparse Dictionary Learning

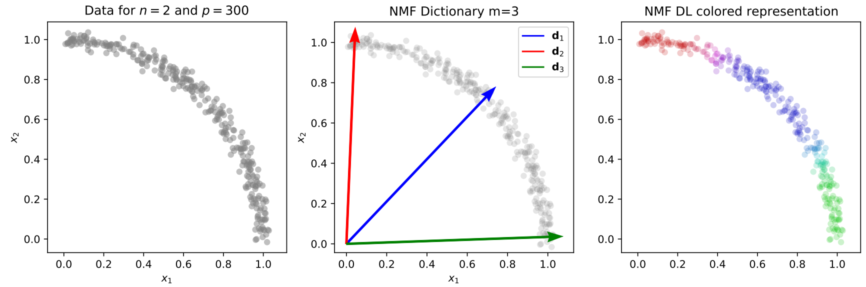


Principle

$$\min_{\mathbf{A}\in\mathbb{R}^{m\times p},\mathbf{D}\in\mathbb{R}^{n\times m},\|\mathbf{d}_i\|=1,\forall i} \quad \|\mathbf{X}-\mathbf{D}\mathbf{A}\|_F^2 + \lambda\sum_i \|\mathbf{a}_i\|_1 \tag{30}$$

- Constraints on the norm of $\mathbf{d}_i$ ensure normalized basis (not orthogonal).
- Sparsity regularization on the representations $\mathbf{a}_i$ promotes samples in linear subspaces of the span of $\mathbf{D}$.
- Can be generalized to other losses L .
- Can be solved efficiently with stochastic optimization [Mairal et al., 2009].

Non Negative Matrix Factorization (NMF)

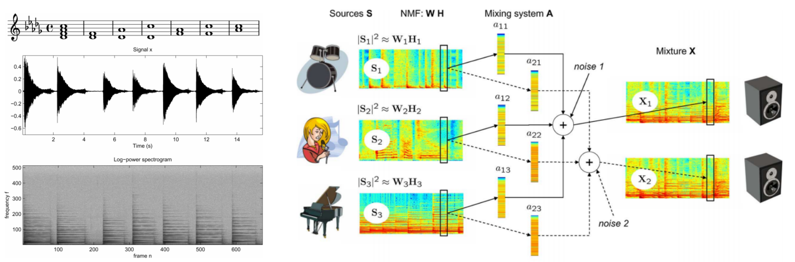


Principle [Lee and Seung, 2000]

$$\min_{\mathbf{A}\in\mathbb{R}_+^{m\times p},\mathbf{D}\in\mathbb{R}_+^{n\times m},\|\mathbf{d}_i\|=1,\forall i} \quad \|\mathbf{X}-\mathbf{D}\mathbf{A}\|_F^2 + \lambda\sum_i \|\mathbf{a}_i\|_1 \tag{31}$$

- For positive data (for instance power densities) it makes sens to have both dictionary elements $\mathbf{d}_j$ and representations $\mathbf{a}_j$ positive.
- Other losses can be used to better adapt to the data (Kullback–Leibler divergence, Itakura-Saito [Févotte et al., 2009]).
- Sparsity can sometimes be used for regularization.

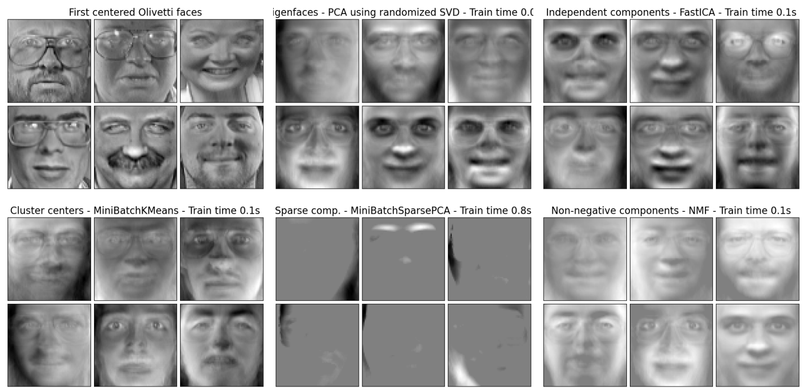
NMF for audio source separation



NMF on the spectrogram

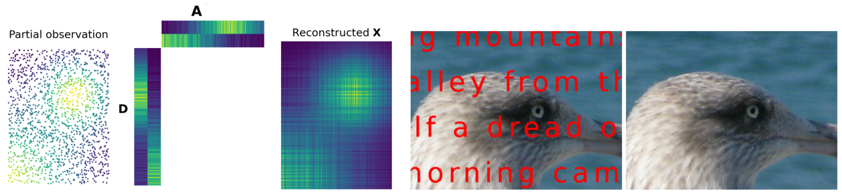
- Factorize the spectrogram of audio sequence as a low rank matrix and perform NMF to separate the sources with different spectra [Févotte et al., 2009].
- Reconstruction of individual sources can be done in for the STFT by keeping the phase and scaling wrt to the sources proportions (similar to spectral substraction).
- Can be extended to multiple channel recordings for instance to separate instruments and voice from stereo recordings [Ozerov and Févotte, 2009].

Dictionary learning comparison on faces



- Comparison of different variants of DL/marix factorization on the faces dataset.
- Results from `https://scikit-learn.org/stable/auto_examples/decomposition/plot_faces_decomposition.html`

Dictionary learning with missing data



Principle

$$\min_{\mathbf{A}\in\mathbb{R}^{m\times p},\mathbf{D}\in\mathbb{R}^{n\times m},\|\mathbf{d}_i\|=1,\forall i} \quad \|\mathbf{M}\odot(\mathbf{X}-\mathbf{DA})\|_F^2 \tag{32}$$

- $\odot$ is the pointwise multiplication and $\mathbf{M}\in\{0,1\}^{n\times p}$ is a binary mask denoting which features that are observed in the matrix $\mathbf{X}$.
- Data is only partially observed but one wants to predict the values for all components of the matrix $\mathbf{X}$ (observed values are stored in a sparse matrix).
- Solved using truncated Singular Vector Decomposition that return a low rank $p<\min(d,n)$ factorization $\mathbf{X}\approx\mathbf{AD}^T$.
- Used in recommender systems and for data imputation.
- Example for image inpainting in [Mairal et al., 2009].

WaveNet

Principle [Oord et al., 2016a]

$$p(\mathbf{x})=\prod_{t=1}^{N}p(x_t|x_1,\dots,x_{t-1}) \tag{33}$$

- The model suppose a factorization of the probability of a whole signal.
- The value x_t depends only on values of the past.
- Model the conditional probabilities are modeled as a DNN with staking of convolutional layers (Non-linear AR).
- Train the model by maximizing the log-likelihood wrt the parameters (separable thanks to factorization above).
- Variants of the model can include conditional variables and signals (for speaker selection and Text-To-Speech applications)

μ-law quantization and categorical prediction

Quantization of the signal

- Using the classical μ-law transformation (standard PCM encoding in the US)

$$f(x_t)=\text{sign}(x_t)\frac{\log(1+\mu|x_t|)}{\log(1+\mu)}$$

with $\mu=256$ and $-1<x_t<1$

- Transformed signal is quantized on 256 levels.
- Known as a good quantization for speech signals that have high dynamic.

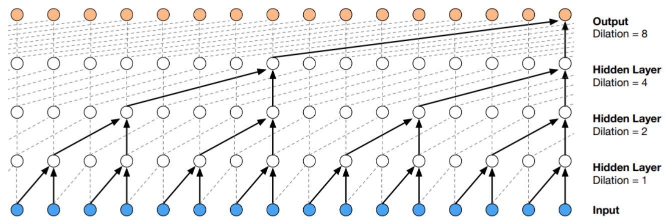
Categorical prediction and softmax

- Predicting the value of x_t cast as a classification problem instead of a regression.
- The output of the neural network has $K=256$ score functions $f(\mathbf{x})_k$ that go through the softmax operator to ensure a discrete probability distribution :

$$\text{Softmax}(f_k(\mathbf{x}))_k=\frac{\exp(f_k(\mathbf{x}))}{\sum_j\exp(f_j(\mathbf{x}))}$$

- Prediction error is measures with the categorical cross entropy that is a classical loss for multi-class classification equivalent to likelihood maximization.

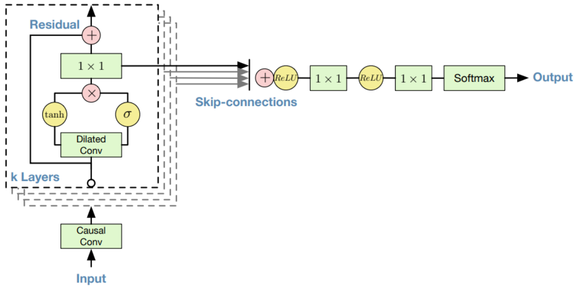
Dilated convolution



Principle [Combes et al., 2012]

- WaveNet uses causal convolutions and non-linear activations for modeling.
- Good modeling of a high frequency signal requires a long "receptive field" (equivalet of the size N of the AR model).
- Dilated convolution performs a convolution of two samples separated by a factor several dilatation layers ensuring that the whole window is used.
- Better factorization into small filters and more changes to add non linearity.

Residual net and gated activations



- Each layer k in the NN contains a dilated convolution followed by a gated activation [Oord et al., 2016b] of the form

$$\mathbf{z} = \tanh(\mathbf{W}_{f,k} \star \mathbf{x}) \odot \sigma(\mathbf{W}_{g,k} \star \mathbf{x})$$

  where σ is the sigmoid that reweights the output of the tanh activation focusing on some temporal areas .
- The output of each layer is a residual net [He et al., 2016]: $\mathbf{x} + \alpha\mathbf{z}$
- The final prediction is a weighted sum of all the output of the layers (skip connections).

Conditional WaveNet

Generative model

- Model will provide probabilities for the values of the next sample from the past observations.
- Can be used for generic signal generation (speech is meaningless).
- Practical application might require more control such as a selection of speaker or a sequence of musical notes et phonemes.

Conditional model

- Main idea is to condition the model *w.r.t.* the variables provided in the training dataset.
- Conditional representation *w.r.t.* a latent variable $\mathbf{h} \in \mathbb{R}^d$:

$$\mathbf{z} = \tanh(\mathbf{W}_{f,k} \star \mathbf{x} + \mathbf{V}_{f,k}^\top \mathbf{h}) \odot \sigma(\mathbf{W}_{g,k} \star \mathbf{x} + \mathbf{V}_{g,k}^\top \mathbf{h})$$

- Conditional representation *w.r.t.* a latent signal $\mathbf{y} \in \mathbb{R}^N$:

$$\mathbf{z} = \tanh(\mathbf{W}_{f,k} \star \mathbf{x} + \mathbf{V}_{f,k} \star \mathbf{y}) \odot \sigma(\mathbf{W}_{g,k} \star \mathbf{x} + \mathbf{V}_{g,k} \star \mathbf{y})$$

- $\mathbf{y}$ can be the (learned) upsampling of a low temporal resolution time series.

Applications of WaveNet

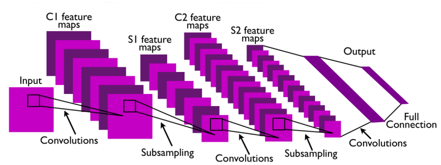
History of Wavenet

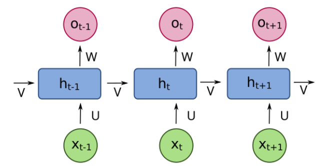
- Proposed originally in [Oord et al., 2016a] to generate realistic signals at 16KHz.
- Made more efficient and integrated in Google Assistant in 2017.

Applications

- Original applications in [Oord et al., 2016a]
  - Multi-speaker speech generation
  - Text-To-Speech (TTS)
  - Music generation
- Provided in Google cloud as a TTS service conditioned by text and speakers.
- Used for signal representation and speaker swapping [Chorowski et al., 2019].

Deep learning on signal and images





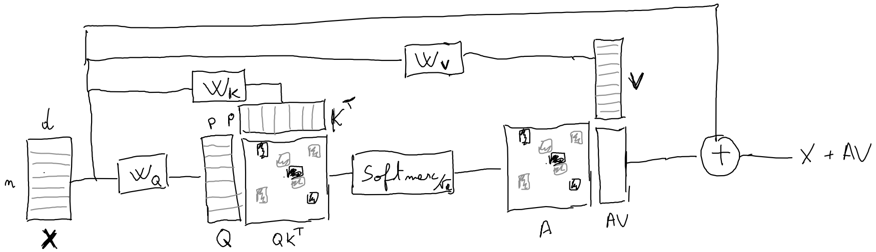
Deep learning on sequences and images

- Convolution neural networks (CNN) are non-linear filters learned on data but limited expressivity.
- Recurrent neural networks (RNN) and more recently Long Short-Term Memory models work well on sequences but harder to train.

Attention models: Transformers

- Attention mechanism is a way to focus on specific parts of the input sequence.
- Transformer model [Vaswani et al., 2017] is a sequence-to-sequence model that uses attention mechanism.
- Used for machine translation, image captioning, speech recognition.

Attention mechanism

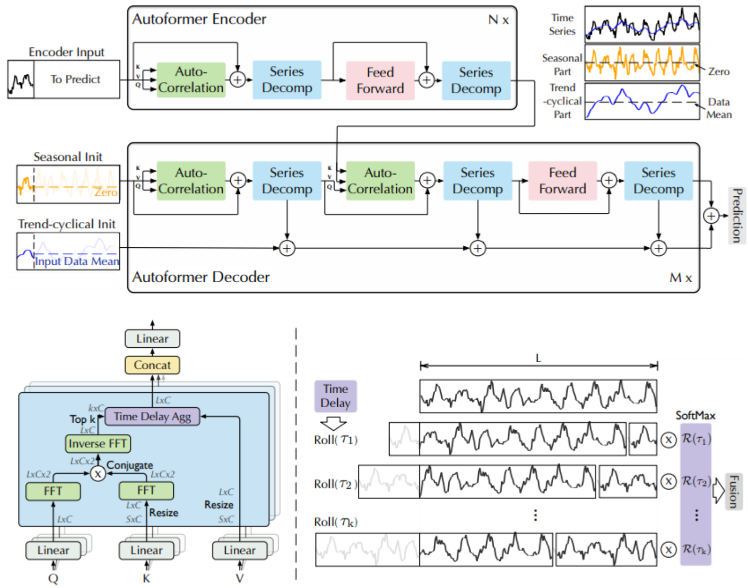


Principle [Vaswani et al., 2017]

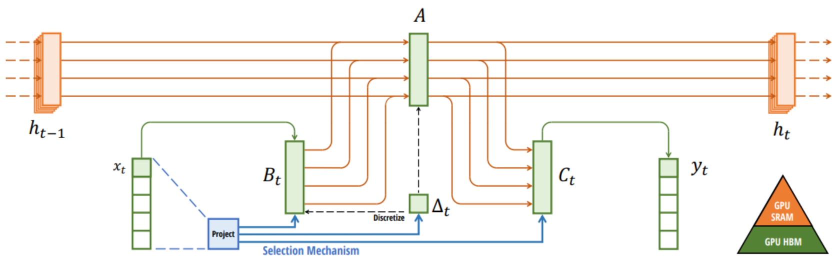
$$\text{AttLayer}(\mathbf{X}) = \mathbf{X} + \text{Softmax}_h(\underbrace{\mathbf{XW}_Q}_{\mathbf{Q}}\underbrace{(\mathbf{XW}_K)^T}_{\mathbf{K}^\top}/\sqrt{p})\underbrace{\mathbf{XW}_V}_{\mathbf{V}} \tag{34}$$

- Parameters $\mathbf{W}_Q \in \mathbb{R}^{d\times p}, \mathbf{W}_K \in \mathbb{R}^{d\times p}, \mathbf{W}_V \in \mathbb{R}^{d\times d}$ are learned from the data and when d is large $\mathbf{W}_V$ is a rank p matrix.
- Horiz. softmax $\text{Softmax}(\mathbf{x}) = \exp(x_i)/\sum_j \exp(x_j)$ is a way to focus on specific parts of the input sequence (quadratic memory w.r.t. sequence size).
- The Transformer model is a stack of several layers of attention followed by normalization and feed forward layers.
- Warning: Ordering of the "tokens" done by positional encoding.

Transformers for time series



State-Space Models for time series



Mamba: Linear-Time Sequence Modeling with Selective State-Spaces [Gu and Dao, 2023]

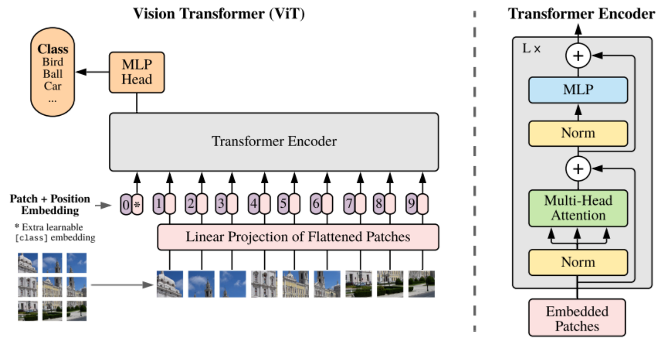
- Use a (time discretized) state space model to model the time series:

$$\mathbf{h}_{k+1} = \mathbf{A}_k\mathbf{h}_k + \mathbf{B}_k\mathbf{x}_k \tag{35}$$

$$\mathbf{y}_{k+1} = \mathbf{C}_k\mathbf{h}_{k+1} \tag{36}$$

- Implemented as a global (fast) convolution for training, but recurrently for predicting (IIR filter can be approximated by FIR filter).
- Selection mechanism is done by a gating mechanism to select the relevant state space, efficient memory implementation on GPU.
- Similar perf. to Transformer but faster to train and predict (linear complexity).

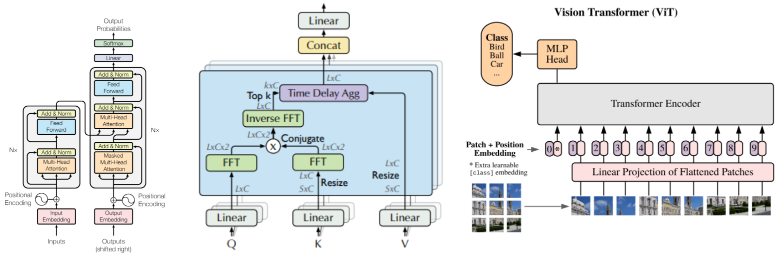
Transformer for images



ViT : Vision Transformer [Dosovitskiy et al., 2020]

- Use of the transformer model for image classification.
- The image is divided into patches that are processed by the transformer model.
- Very large models, require large datasets at least for pre-training.
- Basis for recent Generative Diffusion models [Peebles and Xie, 2023]
- Joint image/text modeling with cross attention [Xu et al., 2015].

Conclusion on Transformers



Transformers in signal processing

- Attention mechanism is a powerful tool for focusing on specific parts of structured data.
- Transformer model are applied on tokens: tokenization is necessary sometimes with positional encoding.
- Used for machine translation, image captioning, speech recognition.
- Very important computational/energy cost and required extremely large dataset.

Graph Signal Processing (GSP)

Principle (Tutorial [Ortega et al., 2018])

- Time and space signals have a regular and very specific structure.
- In some applications, the relation between the samples might be more complex.
- Graphs can be used to model this relation between samples (nodes of the graph).
- The signal on the graph is plotted through the color of the nodes.
- Illustrations in this course are done using
  - PyGSP Python GSP toolbox [Defferrard et al., 2017]
  - Strong inspiration by the awesome notebooks from `https://github.com/mdeff/pygsp_tutorial_graphsip`.

Graphs and matrices

Graph and signal

- We define a Signal on graph as
  - A graph $\mathcal{G}$ described through its **adjacency matrix** $\mathbf{A} \in \{0,1\}^{N \times N}$.
  - $\mathbf{x} \in \mathbb{R}^N$ the signal where x_i is the samples/signal at node i in the graph.
- The adjacency matrix define the existence of edges between two node: $A_{i,j} = 1$ is there exist an edge from node i to j.
- A graph is said to be symmetric if $A_{i,j} = A_{j,i},\ \forall i,j$ (often the case in GSP).

Graph matrices

- The **adjacency matrix** $\mathbf{A} \in \{0,1\}^{N \times N}$ describes the connections between nodes.
- The **Laplacian matrix** is defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{A}, \qquad \text{with} \qquad \mathbf{D} = \text{diag}(\mathbf{A}\mathbf{1}_N) \tag{37}$$

  where $\mathbf{D}$ is the diagonal degree matrix.
- Sometime the adjacency matrix can be weighted $\mathbf{A} \in \mathbb{R}_+^{N \times N}$, in this case it is often denoted as $\mathbf{W}$.

Notion of shift

Shift in 1D signals

- In a discrete 1D signal a temporal shift is a convolution by a dirac

$$x^s[n] = x[n] \star \delta[n-1]$$

- From a matrix point of view a circular temporal shift can be done with the following linear operation

$$\mathbf{x}^s = \mathbf{A}\mathbf{x}, \qquad \mathbf{A} = \begin{bmatrix} 0 & 0 & \dots & 1 \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

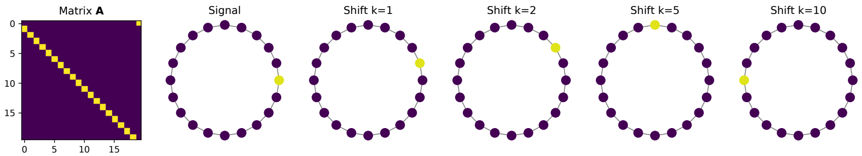
- The matrix $\mathbf{A}$ is both the adjacency matrix of the graph for a circular signal and its shift operator.
- A shift of k can be expressed as $\mathbf{A}^k\mathbf{x}$.

Shift in a graph

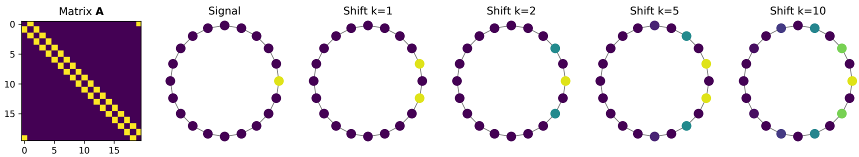
- Shift $\mathbf{A}\mathbf{x}$ is the propagation of the signal for general graphs.
- Similarly to time signal we can define the property of an operator f as shift invariant when $f(\mathbf{x}^s) = f(\mathbf{x})^s$.

Example of shifts

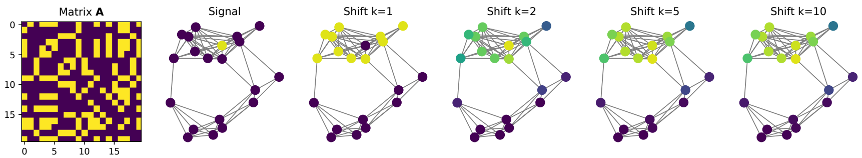
Circular 1D signal



Symmetric Circular 1D signal



Sensor graph



Spectral decomposition of a graph

Decomposition of the Laplacian

- The Laplacian matrix of a graph can be factorized as

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$$

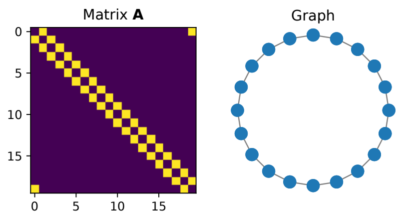
  where the columns of $\mathbf{U}$ are an orthonormal basis and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_N)$ are the eigenvalues .
- For a symmetric graph, the Laplacian is SPD and $\mathbf{U}$ is real.
- The basis vector $\mathbf{u}_k$ are sorted by increasing λ_k where λ_k can be seen as frequencies in the graph (spatial variance of the basis function $\mathbf{u}_k$).
- For non-symmetric graphs one can decompose the adjacency matrix but the basis will be complex (for a 1D circular graph, it recovers the discrete Fourier basis).

Fourier transform on graph

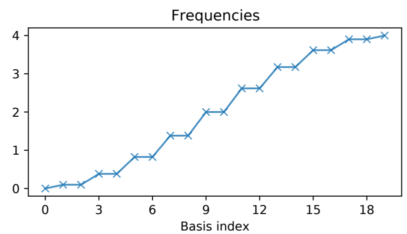
- The operator $\mathbf{U}^\top$ is called the Graph Fourier Transform.
- A shift invariant operator $\mathbf{V}$ can be diagonalized by $\mathbf{U}$.
- Similarly to a convolution it can be applied by a pointwise product in the Fourier domain/

Fourier basis : 1D perodic signal

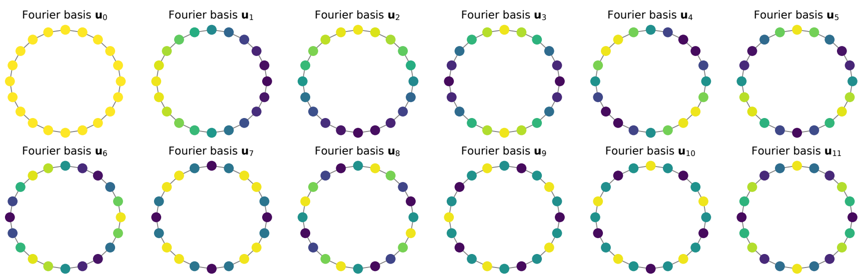
Adjacency matrix and graph



Fourier Basis frequencies

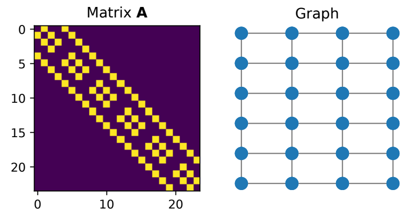


Fourier Basis

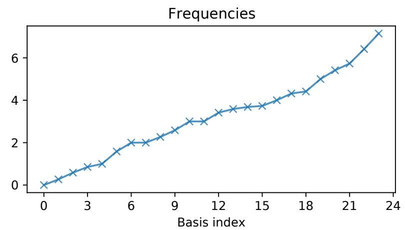


Fourier basis : regular 2D grid

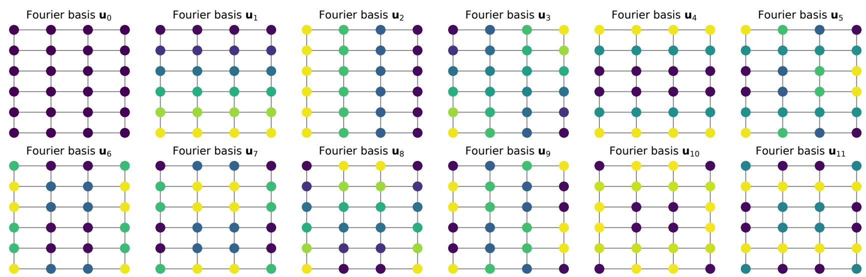
Adjacency matrix and graph



Fourier Basis frequencies

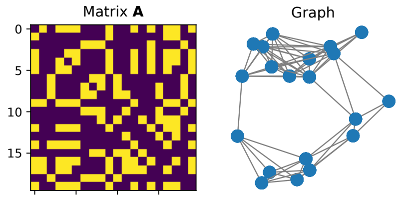


Fourier Basis

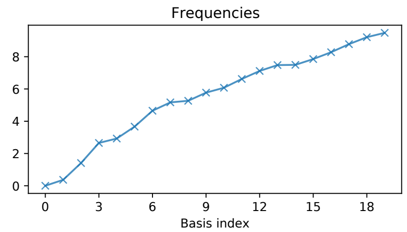


Fourier basis : Sensor graph

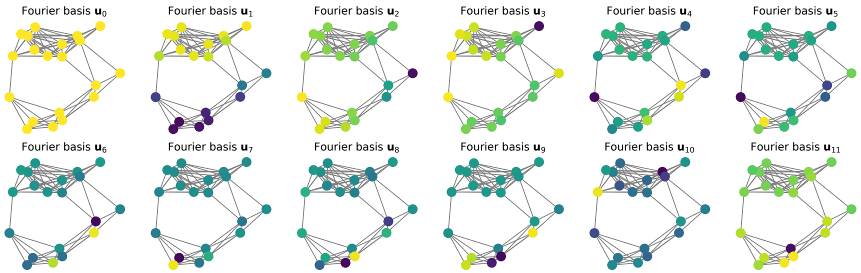
Adjacency matrix and graph



Fourier Basis frequencies

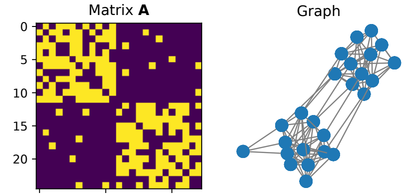


Fourier Basis

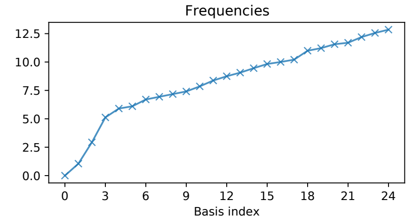


Fourier basis : Stochastic Block Model

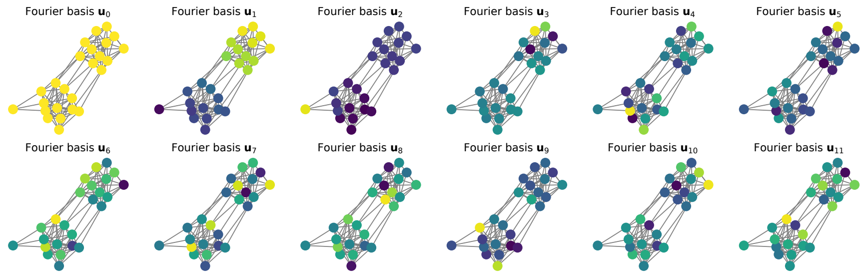
Adjacency matrix and graph



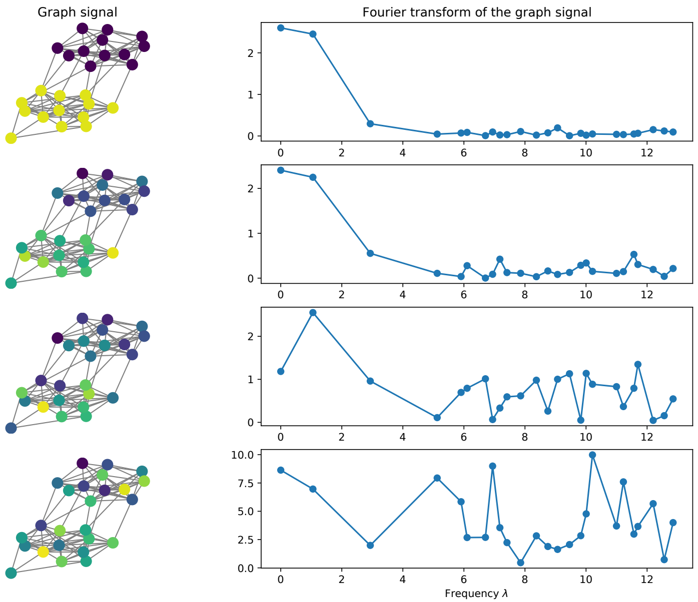
Fourier Basis frequencies



Fourier Basis



Graph Fourier Transform



Filtering a signal on graph

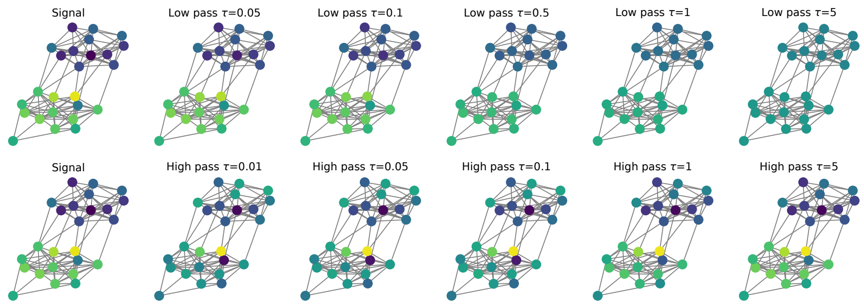
Principle

- Filtering is done with a point-wise product in the frequency domain by a frequency response function $H(\lambda)$.
- Let $\mathbf{h}$ be the frequency response $h_i = H(\lambda_i)$ as a function of the frequencies in the graph. The filtered signal is:

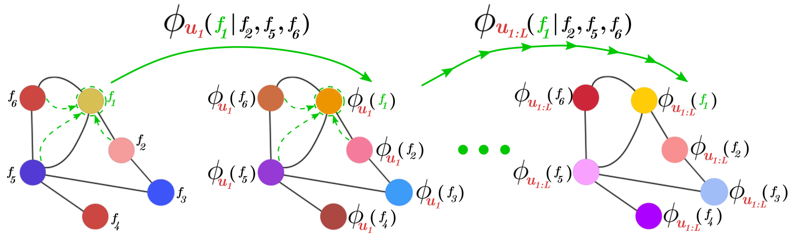
$$\mathbf{x}^f = \mathcal{GFT}^{-1}[\mathcal{GFT}[\mathbf{x}] \odot \mathbf{h}] = \mathbf{U}(\mathbf{h} \odot \mathbf{U}^\top \mathbf{x}) \tag{38}$$

- GFT can be costly on large graph, filters can be approximated using Chebyshev polynomials.

Low and high pass filter : $H_1(\lambda) = \frac{1}{1+\tau\lambda}$, $H_2(\lambda) = \frac{\tau\lambda}{1+\tau\lambda}$



Graph Neural Networks (GNN)



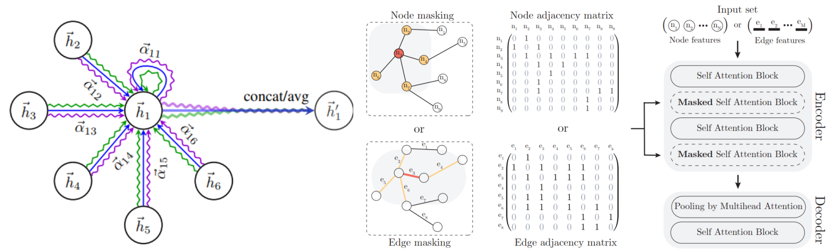
Graph Neural Network (Review : [Wu et al., 2020])

- GNN are a way to perform deep learning on graph structured data.
- Multiple layers alternate between filtering (message passing) and non-linear transformation [Scarselli et al., 2008].
- Spectral GNN are based on the graph Fourier transform and learn the filter $H(\lambda)$.
- Graph Convolutional Networks (GCN) [Kipf and Welling, 2016] are a popular variant of GNN where the local propagation update is :

$$\mathbf{X}_{l+1} = \sigma\left(\tilde{\mathbf{A}}\mathbf{X}_l\mathbf{W}_l\right) \quad \text{with } \tilde{\mathbf{A}} = \mathbf{D}^{-1/2}(\mathbf{A}+\mathbf{I})\mathbf{D}^{-1/2},\ \mathbf{D} = \text{diag}(\mathbf{A}\mathbf{1})$$

- Can perform node or edge prediction, graph classification (after pooling), etc.

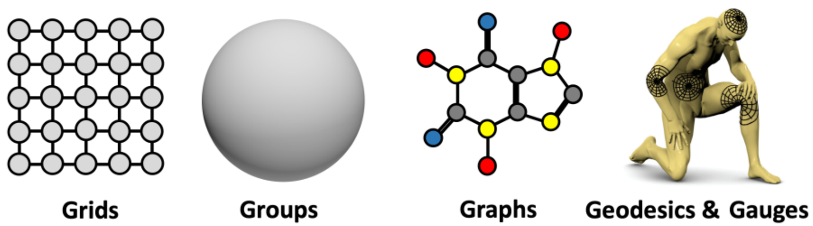
Attention mechanism on graphs



GAT: Graph Attention Networks [Velicković et al., 2017]

- Attention mechanism can be used to focus on specific nodes in the graph.
- Combination of a GNN and and attention layers where the message passing is weighted by the attention ($\tilde{\mathbf{A}}$ is attention matrix masked by $\mathbf{A}+\mathbf{I}$).
- The attention mechanism is learned from the data and allows to select the most relevant nodes in the neighborhood.
- Recent approach directly learn the attention mechanism between all nodes (and edges) in the graph [Buterez et al., 2024].

Geometric Deep Learning



Principle (Recent reference : [Bronstein et al., 2021])

- Objective : go beyond euclidean data (independent samples in $\mathbb{R}^d$)
- Importance of symmetry, invariance and equivariance on geometric data.
- Common framework for modeling
  - Convolutional Neural Networks (CNN)
  - Graph Neural Networks (GNN)
  - Recurrent Neural Networks (RNN)
  - Transformers can learn geometric structure from the data.

Image from [Bronstein et al., 2021, Figure 9]

Bibliography

- A Wavelet tour of signal processing [Mallat, 1999].
- Wavelets and sub-band coding [Vetterli and Kovacevic, 1995].
- Discrete-time signal processing [Oppenheim and Shafer, 1999].
- Signals and Systems [Haykin and Van Veen, 2007].
- Signals and Systems [Oppenheim et al., 1997].
- Signal Analysis [Papoulis, 1977].
- Fourier Analysis and its applications [Vretblad, 2003].

- Polycopiés from Stéphane Mallat and Éric Moulines [Mallat et al., 2015].
- Distributions et Transformation de Fourier [Roddier, 1985]

References I

[Adeli et al., 2003] Adeli, H., Zhou, Z., and Dadmehr, N. (2003).
Analysis of eeg records in an epileptic patient using wavelet transform.
Journal of neuroscience methods, 123(1):69–87.

[Ahmed et al., 1974] Ahmed, N., Natarajan, T., and Rao, K. R. (1974).
Discrete cosine transform.
IEEE transactions on Computers, 100(1):90–93.

[Boll, 1979] Boll, S. (1979).
Suppression of acoustic noise in speech using spectral subtraction.
IEEE Transactions on acoustics, speech, and signal processing, 27(2):113–120.

[Bronstein et al., 2021] Bronstein, M. M., Bruna, J., Cohen, T., and Velicković, P. (2021).
Geometric deep learning: Grids, groups, graphs, geodesics, and gauges.
arXiv preprint arXiv:2104.13478.

[Buterez et al., 2024] Buterez, D., Janet, J. P., Oglic, D., and Lio, P. (2024).
Masked attention is all you need for graphs.
arXiv preprint arXiv:2402.10793.

References II

[Carlotti et al., 2011] Carlotti, A., Vanderbei, R., and Kasdin, N. J. (2011).
Optimal pupil apodizations of arbitrary apertures for high-contrast imaging.
Optics Express, 19(27):26796–26809.

[Chen and Donoho, 1994] Chen, S. and Donoho, D. (1994).
Basis pursuit.
In *Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 41–44. IEEE.

[Chorowski et al., 2019] Chorowski, J., Weiss, R. J., Bengio, S., and van den Oord, A. (2019).
Unsupervised speech representation learning using wavenet autoencoders.
IEEE/ACM transactions on audio, speech, and language processing, 27(12):2041–2053.

[Combes et al., 2012] Combes, J.-M., Grossmann, A., and Tchamitchian, P. (2012).
Wavelets: Time-Frequency Methods and Phase Space Proceedings of the International Conference, Marseille, France, December 14–18, 1987.
Springer Science & Business Media.

[Defferrard et al., 2017] Defferrard, M., Martin, L., Pena, R., and Perraudin, N. (2017).
Pygsp: Graph signal processing in python.

References III

[Dosovitskiy et al., 2020] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2020).
An image is worth 16x16 words: Transformers for image recognition at scale.
CoRR, abs/2010.11929.

[Févotte et al., 2009] Févotte, C., Bertin, N., and Durrieu, J.-L. (2009).
Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis.
Neural computation, 21(3):793–830.

[Gong et al., 2021] Gong, Y., Chung, Y.-A., and Glass, J. (2021).
Ast: Audio spectrogram transformer.
arXiv preprint arXiv:2104.01778.

[Group et al., 2000] Group, J. P. E. et al. (2000).
Jpeg 2000 image coding system, iso/iec 15444-1: 2000.

[Gu and Dao, 2023] Gu, A. and Dao, T. (2023).
Mamba: Linear-time sequence modeling with selective state spaces.
arXiv preprint arXiv:2312.00752.

References IV

[Haykin and Van Veen, 2007] Haykin, S. and Van Veen, B. (2007).
Signals and systems.
John Wiley & Sons.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016).
Deep residual learning for image recognition.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[Herault and Jutten, 1986] Herault, J. and Jutten, C. (1986).
Space or time adaptive signal processing by neural network models.
In *AIP conference proceedings*, volume 151, pages 206–211. American Institute of Physics.

[Hyvärinen and Oja, 2000] Hyvärinen, A. and Oja, E. (2000).
Independent component analysis: algorithms and applications.
Neural networks, 13(4-5):411–430.

[Kipf and Welling, 2016] Kipf, T. N. and Welling, M. (2016).
Semi-supervised classification with graph convolutional networks.
arXiv preprint arXiv:1609.02907.

References V

[Lee and Seung, 2000] Lee, D. and Seung, H. S. (2000).
Algorithms for non-negative matrix factorization.
Advances in neural information processing systems, 13:556–562.

[MacQueen et al., 1967] MacQueen, J. et al. (1967).
Some methods for classification and analysis of multivariate observations.
In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

[Mairal et al., 2009] Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2009).
Online dictionary learning for sparse coding.
In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696.

[Mallat, 1999] Mallat, S. (1999).
A wavelet tour of signal processing.
Elsevier.

[Mallat et al., 2015] Mallat, S., Moulines, E., and Roueff, F. (2015).
Traitement du signal.
Polycopié MAP 555, École Polytechnique.

References VI

[Mallat, 1989] Mallat, S. G. (1989).
A theory for multiresolution signal decomposition: the wavelet representation.
IEEE transactions on pattern analysis and machine intelligence, 11(7):674–693.

[Oord et al., 2016a] Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016a).
Wavenet: A generative model for raw audio.
arXiv preprint arXiv:1609.03499.

[Oord et al., 2016b] Oord, A. v. d., Kalchbrenner, N., Vinyals, O., Espeholt, L., Graves, A., and Kavukcuoglu, K. (2016b).
Conditional image generation with pixelcnn decoders.
arXiv preprint arXiv:1606.05328.

[Oppenheim and Shafer, 1999] Oppenheim, A. V. and Shafer, R. W. (1999).
Discrete-time signal processing.
Prentice Hall Inc.

[Oppenheim et al., 1997] Oppenheim, A. V., Willsky, A. S., and Nawab, S. H. (1997).
Signals and systems prentice hall.
Inc., Upper Saddle River, New Jersey, 7458.

References VII

[Ortega et al., 2018] Ortega, A., Frossard, P., Kovacević, J., Moura, J. M., and Vandergheynst, P. (2018).
Graph signal processing: Overview, challenges, and applications.
Proceedings of the IEEE, 106(5):808–828.

[Ozerov and Févotte, 2009] Ozerov, A. and Févotte, C. (2009).
Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation.
IEEE Transactions on Audio, Speech, and Language Processing, 18(3):550–563.

[Papoulis, 1977] Papoulis, A. (1977).
Signal analysis, volume 191.
McGraw-Hill New York.

[Peebles and Xie, 2023] Peebles, W. and Xie, S. (2023).
Scalable diffusion models with transformers.
In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205.

[Ricaud and Torrésani, 2014] Ricaud, B. and Torrésani, B. (2014).
A survey of uncertainty principles and some signal processing applications.
Advances in Computational Mathematics, 40(3):629–650.

References VIII

[Roddier, 1985] Roddier, F. (1985).
Distributions et transformée de fourier.

[Scarselli et al., 2008] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2008).
The graph neural network model.
IEEE transactions on neural networks, 20(1):61–80.

[Sirovich and Kirby, 1987] Sirovich, L. and Kirby, M. (1987).
Low-dimensional procedure for the characterization of human faces.
Josa a, 4(3):519–524.

[Soummer et al., 2003] Soummer, R., Aime, C., and Falloon, P. (2003).
Stellar coronagraphy with prolate apodized circular apertures.
Astronomy & Astrophysics, 397(3):1161–1172.

[Turk and Pentland, 1991] Turk, M. and Pentland, A. (1991).
Eigenfaces for recognition.
Journal of cognitive neuroscience, 3(1):71–86.

References IX

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017).
Attention is all you need.
In *Advances in Neural Information Processing Systems*.

[Velicković et al., 2017] Velicković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2017).
Graph attention networks.
arXiv preprint arXiv:1710.10903.

[Vetterli and Kovacevic, 1995] Vetterli, M. and Kovacevic, J. (1995).
Wavelets and subband coding.
Number BOOK. Prentice-hall.

[Vretblad, 2003] Vretblad, A. (2003).
Fourier analysis and its applications, volume 223.
Springer Science & Business Media.

[Welch, 1967] Welch, P. (1967).
The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms.
IEEE Transactions on audio and electroacoustics, 15(2):70–73.

References X

[Wu et al., 2021] Wu, H., Xu, J., Wang, J., and Long, M. (2021).
Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting.
Advances in neural information processing systems, 34:22419–22430.

[Wu et al., 2020] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. (2020).
A comprehensive survey on graph neural networks.
IEEE transactions on neural networks and learning systems, 32(1):4–24.

[Xu et al., 2015] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A. C., Salakhutdinov, R., Zemel, R. S., and Bengio, Y. (2015).
Show, attend and tell: Neural image caption generation with visual attention.
arXiv preprint arXiv:1502.03044.

[Yu et al., 2008] Yu, G., Mallat, S., and Bacry, E. (2008).
Audio denoising by time-frequency block thresholding.
IEEE Transactions on Signal processing, 56(5):1830–1839.