Wavelet Transform

Machine Learning

Amir Mohammad Kharazi

Tarbiat Modares University

GitHub

2022

Table of Contents

Signals and Time Series

- In a time-series data set, the to-be-predicted value (y) is a function of time ($y = y(t)$).
- A signal is a more general version of this where the dependent variable y does not have to be a function of time ($y = y(t)$); it can be a function of spatial coordinates ($y = y(x, y)$), distance from the source ($y = y(r)$), etc.

Signals

- Signals can come in many different forms and shapes: you can think of audio signals, pictures, video signals, geophysical signals (seismic data), sonar and radar data and medical signals (EEG, ECG, EMG).

Time Signals

- Continuous time-domain (Most of real Signals) - Analog
- Discrete-time domain (Digitized version)

Digitizing an analog signal

- Is done by sampling it with a specific sampling rate
- Under-Sampled (aliasing and the wagon-wheel effect)
- Nyquist rate : twice the highest frequency present in the signal
- To prevent under-sampling, usually a frequency much higher than the Nyquist rate is chosen as the sampling frequency.

Digitizing an analog signal

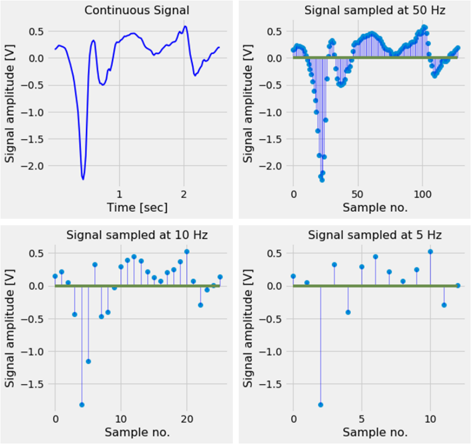

Figure: A continuous signal sampled at different frequencies.

Periodic Signals

- If the signal contains a pattern, which repeats itself after a specific period of time.
- The time it takes for an periodic signal to repeat itself is called the period P.
- The distance it travels in this period is called the wavelength λ
- The frequency f is the inverse of the Period.
- $f = \frac{1}{P} = \frac{V}{\lambda}$

Fourier Transform

- Every signal, no matter how complex it looks, can be decomposed into a sum of its simpler signals.
- These simpler signals are trigonometric functions (sine and cosine waves).
- Transform a signal from its **time-domain** to its **frequency domain**.
- Inverse Fourier Transform.
- DFT : FFT in scipy and numpy
  (fft , ifft , fft2 , ifft2 , fftn , ifftn).

Fourier Transform

$$y[k] = \sum_{n=0}^{N-1} x[n]. \exp^{-2\pi ik\frac{n}{N}} \tag{1}$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} y[k]. \exp^{2\pi ik\frac{n}{N}} \tag{2}$$

Examples

Full Explanation at GitHub .

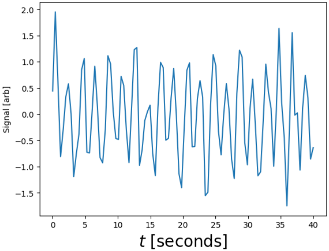

Figure: Composite Signal with 0.24Hz, 0.49Hz and some random noise

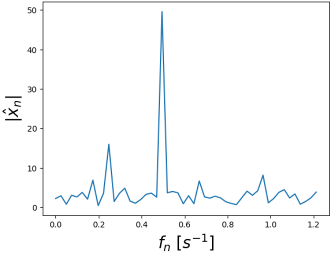

Figure: Fourier Transform of the Signal. Two spikes at 0.24 and 0.49

Examples

Full Explanation at GitHub .

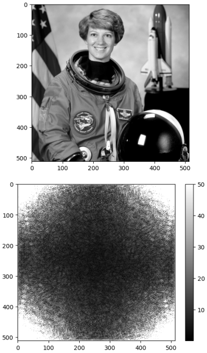

Figure: Gray scale image and its FT

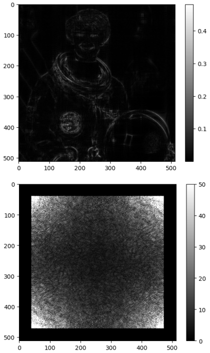

Figure: removed low frequency and inverse FT

Statistical parameter estimation and feature extraction

- After we have transformed a signal to the frequency-domain, we can extract features from each of these transformed signals.
- Use these features as input in standard classifiers like Random Forest, Logistic Regression, Gradient Boosting or Support Vector Machines.

Statistical parameter estimation and feature extraction

Which features can we extract from these transformations?

- A good first step is the value of the frequencies at which oscillations occur and the corresponding amplitudes.
- In other words; the x and y-position of the peaks in the frequency spectrum.

Summary of Fourier Transform

- Transform a signal from its time-domain to its frequency domain
- The peaks in the frequency spectrum indicate the most occurring frequencies in the signal
- The location (frequency-value) and height of the peaks in the frequency spectrum then can be used as input for Classifiers like Random Forest or Gradient Boosting

Problem ?

- The general rule is that this approach of using the Fourier Transform will work very well when the frequency spectrum is stationary.
- That is, the frequencies present in the signal are not time-dependent; if a signal contains a frequency of $x\ Hz$ this frequency should be present equally anywhere in the signal.

Problem ?

- The more non-stationary/dynamic a signal is, the worse the results will be.
- That's too bad, since most of the signals we see in real life are non-stationary in nature.

Solution

A much better approach for analyzing dynamic signals is to use the Wavelet Transform instead of the Fourier Transform.

Example

- The thing about the Fourier Transform is that it has a high resolution in the frequency-domain but zero resolution in the time-domain.
- This means that it can tell us exactly which frequencies are present in a signal, but not at which location in time these frequencies have occurred.
- Code is available at GitHub .

Example

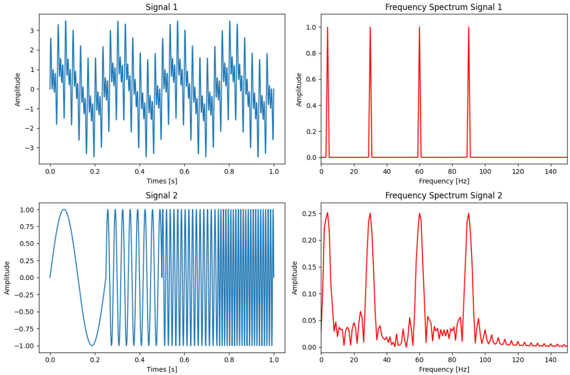

Figure: The signals and frequency spectrum of a signal which contains four frequencies at all times (top), four different frequencies at four different times (bottom).

Short-Time Fourier Transform.

- Original signal is splitted into several parts of equal length (which may or may not have an overlap)
- How ? by using a sliding window before applying the Fourier Transform.
- if we split our signal into 10 parts, and the Fourier Transform detects a specific frequency in the second part,
- then we know for sure that this frequency has occurred between $\frac{2}{10}$th and $\frac{3}{10}$th of our original signal.

Short-Time Fourier Transform.

The main problem with this approach is that you run into the theoretical limits of the Fourier Transform known as the uncertainty principle.

That is :

The smaller we make the size of the window the more we will know about where a frequency has occurred in the signal, but less about the frequency value itself. The larger we make the size of the window the more we will know about the frequency value and less about the time.

Wavelet Transform

- A better approach for analyzing signals with a dynamical frequency spectrum is the Wavelet Transform.
- The Wavelet Transform has a high resolution in both the frequency- and the time-domain. It does not only tell us which frequencies are present in a signal, but also at which time these frequencies have occurred.
- This is accomplished by working with different scales. First we look at the signal with a large scale/window and analyze 'large' features and then we look at the signal with smaller scales in order to analyze smaller features.

TS vs FT vs Short-Time FT vs Wavelet

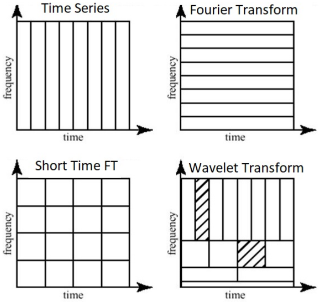

Figure: A schematic overview of the time and frequency resolutions of the different transformations in comparison with the original time-series dataset. The size and orientations of the block gives an indication of the resolution size.

Wavelet Transform

The Wavelet Transform has:

- For small frequency values a high resolution in the frequency domain, low resolution in the time- domain,
- For large frequency values a low resolution in the frequency domain, high resolution in the time domain.

Wavelet Transform

In other words, the Wavelet Transforms makes a trade-off; at scales in which time-dependent features are interesting it has a high resolution in the time-domain and at scales in which frequency-dependent features are interesting it has a high resolution in the frequency domain.

How does the Wavelet Transform work?

- The Fourier Transform uses a series of sine-waves with different frequencies to analyze a signal. That is, a signal is represented through a linear combination of sine-waves.
- The Wavelet Transform uses a series of functions called wavelets, each with a different scale.

How does the Wavelet Transform work?

The word wavelet means a small wave, and this is exactly what a wavelet is.

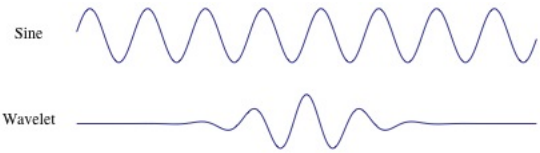

Figure: The difference between a sine-wave and a Wavelet. The sine-wave is infinitely long and the Wavelet is localized in time.

How does the Wavelet Transform work?

- The main difference is that the sine-wave is not localized in time
- A wavelet is localized in time, which allows the wavelet transform to obtain time-information in addition to frequency information

The different types of Wavelet families

- Another difference between the Fourier Transform and the Wavelet Transform is that there are many different families (types) of wavelets.
- This means that we can choose a specific wavelet family which fits best with the features we are looking for in our signal.
- Check the code at GitHub .

Wavelet families

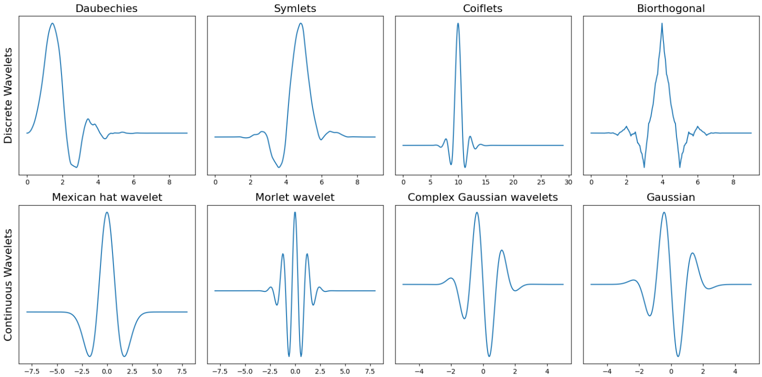

Figure: Several families of Wavelets. In the first row we see discrete wavelets and in the second row we see several continuous wavelets.

A wavelet must have

1. finite energy : Means that it is localized in time and frequency; it is integrable and the inner product between the wavelet and the signal always exists.
2. zero mean :
   Wavelet has zero mean in the time-domain, a zero at zero frequency in the time-domain. This is necessary to ensure that it is integrable and the inverse of the wavelet transform can also be calculated.

Continuous Wavelet Transform

$$X_w(a, b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{+\infty} x(t)\bar{\psi}\left(\frac{t-b}{a}\right) dt$$

- where $\psi(t)$ is the continuous mother wavelet which gets scaled by a factor of a and translated by a factor of b.
- The values of the scaling and translation factors are continuous,
- which means that there can be an infinite amount of wavelets. You can scale the mother wavelet with a factor of 1.3, or 1.31, and 1.311, and 1.3111, etc.

Discrete Wavelet Transform

- the main difference is that the DWT uses discrete values for the scale and translation factor.
- The scale factor increases in powers of two, so ($a = 1, 2, 4, \ldots$) and the translation factor increases integer values ($b = 1, 2, 3, \ldots$).
- **PS**: The DWT is only discrete in the scale and translation domain, not in the time-domain.

Apply the DWT on a signal

- We start with the smallest scale. As we have seen before, small scales correspond with high frequencies. This means that we first analyze high frequency behavior.
- At the second stage, the scale increases with a factor of two (the frequency decreases with a factor of two), and we are analyzing behavior around half of the maximum frequency.
- At the third stage, the scale factor is four and we are analyzing frequency behavior around a quarter of the maximum frequency.
- And this goes on and on, until we have reached the maximum decomposition level.

maximum decomposition level?

- we should know that at each subsequent stage the number of samples in the signal is reduced with a factor of two.
- At lower frequency values, you will need less samples to satisfy the Nyquist rate so there is no need to keep the higher number of samples in the signal
- Due to this downsampling, at some stage in the process the number of samples in our signal will become smaller than the length of the wavelet filter and we will have reached the maximum decomposition level.

Example

Suppose we have a signal with frequencies up to 1000 Hz

- In the first stage we split our signal into a low-frequency part and a high-frequency part. i.e. $0 - 500$ Hz and $500 - 1000$ Hz.
- At the second stage we take the low-frequency part and again split it into two parts: $0 - 250$ Hz and $250 - 500$ Hz.
- At the third stage we split the $0 - 250$ Hz part into a $0 - 125$ Hz part and a $125 - 250$ Hz part. This goes on until we have reached the level of refinement we need or until we run out of samples.

Example

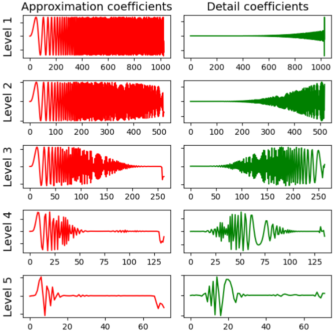

Figure: The approximation and detail coefficients of the sym5 wavelet (level 1 to 5) applied on a chirp signal, from level 1 to 5. On the left we can see a schematic representation of the high pass and low pass filters applied on the signal at each level. available at GitHub .

Practical Example

Using the Continuous Wavelet Transform and a Convolutional Neural Network for classification of signals. Full detail at GitHub .