Intro Lecture Notes

Zach Neveu

September 5, 2019

1 Course Intro

- Using DTU Learn instead of DTU Inside
- Goal: learn fundamental sp methods, and code in Matlab/Python
- Should be able to create a modern signal processing system with ML by the end
- 4 hours of hw/week
- each class 1 hour of lecture, 3 hours of exercises
- Exercises as live scripts/Jupyter notebooks

2 Course Plan

- 3 parts
- Conventional DSP
- Linear Methods
- Non-linear methods - hmms neural nets

3 Technical Lecture

- Traditional SP - doesn't care much about input content
- Traditional ML - not particularly friendly for time series of signals
- MLSP - combines the two
- Example Areas
  - sparsity-aware learning - compression
  - Information-theoretic learning
  - Adaptive filtering
  - Sound processing
  - Images/Videos
  - Telecommunications

- Sensors

4 Connection

- Simplest connection - $h(t)$ can be a classifier etc. ML is just a specific kind of of non-linear processing.
- Another idea: signal processing is how to get a small amount of features from a large amount of data

Day 2: Frequency Transforms, Filtering

Zach Neveu

September 12, 2019

1 Filtering

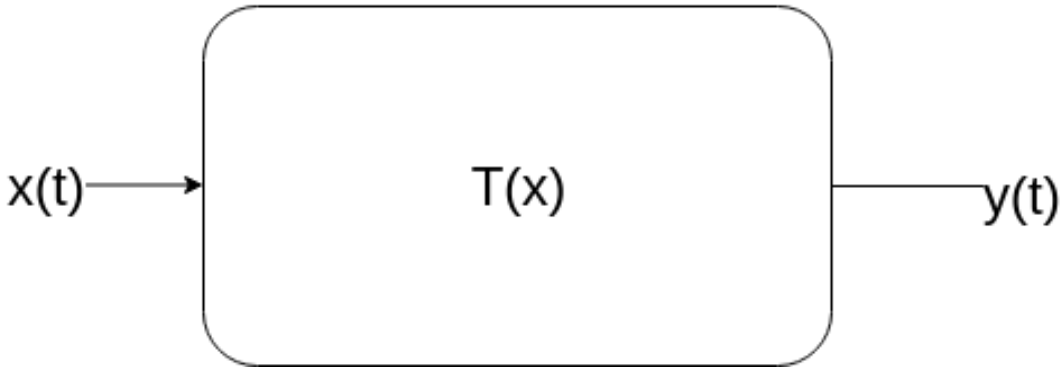

Figure 1: Generic System

- Traditional DSP System: $x(n) \rightarrow T[x(n)] \rightarrow y(n)$
- $T[x(n)] = x^2(n-1)$
- Bold number in sequence is where $n = 0$ by convention
- $x(n) = \{-1, \mathbf{0}, 1, 2, 3\}$
- $y(n) = \{\mathbf{1}, 0, 1, 4, 9\}$ delayed by 1
- Superposition (proves linearity):
- $x(n) = a_1 x_1(n) + a_2 x_2(n) \rightarrow T[x(n)] = a_1 T[x_1(n)] + a_2 T[x_2(n)]$
- $y(n) = a^2 x(n) \rightarrow$ linear
- $y(n) = a x^2(n) \rightarrow$ non-linear
- $y(n) = n x(n) \rightarrow$ time-varying
- $y(n) = x(-n) \rightarrow$ time-varying
- if $y(n+k) = T(x(n+k))$ then time invariant
- General linear system: $y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k)$
- This is just like a discrete convolution! No wonder this is so important to DSP...
- First term is recursive: goes from last output back to Nth previous
- Second term is non-recursive; depends only on past M inputs
- FIR filters have only second term

- IIR filters have first term
- General system equation can be distilled into $x(n) * h(n)$ using the convolution

2 Example Problems

- Linear system with coeffs $a_k = 0, b_0 = 1, b_1 = 2, b_2 = 1, b_i = 0, i \geq 3$
- $y(x) = x(n-1) + 2x(n-2) + x(n-3)$
- $x(n) = \{\ldots, 0, \mathbf{1}, 0, \ldots\}$
- $y(n) = \{\ldots, 0, 1, 2, 1, 0, \ldots\} \rightarrow$ Impulse Response!

3 More Notes

- Fourier vs. Z transform
- Z transform is generalized Fourier transform
- Can use Z transform instead of Fourier
- Equal when:
- $X(f) = X(z)_{z=e^{j2\pi f}}$
- Time domain: $y(n) = h(n)x(n)$
- z-domain: $Y(Z) = H(Z)X(Z)$
- $H(Z) = \sum_{n=-\infty}^{\infty} h(n)z^{-n} = \frac{Y(Z)}{X(Z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1+\sum_{k=0}^{N} a_k z^{-k}}$

4 Time-Frequency Analysis

- Spectrograms - STFT at each time window
- DFT - Discrete Fourier Transform
- DFT: $X(k) = \sum_{n=0}^{N-1} x(n) e^{j\frac{2\pi}{N}kn}$
- STFT: $x(n,k) = \sum_{m=-\infty}^{\infty} x(m)\, w(m-n) e^{-j\frac{2\pi}{N}kn}$
- New signal for STFT: $\hat{x}(n) = x(m) * w(m-n)$
- $w(m-n)$ is window function
- Multiplication in time domain is convolution in freq domain, so window function will be convolved with signal in freq. domain
- Ideal $W(F)$ is a dirac delta, so ideal $w(t)$ is a sinc function

- sinc has infinite time, so window is infinite, so not actually stft
- Must compromise and chop the sinc function
- Look for 3dB point of sinc, as well as main lobe width.
- Types of window functions: Rect, Bartlett, Hann, Hamming
- Rectangular tightest main lobe, but most leakage. High resolution, but lots of noise/ripples
- Hann most commonly used
- Important spectrogram params: block size (B) and hop size (S)
- Block size: how many samples in each calculation? How big is window?
- Hop size: how far to jump forward before calculating next frame?
- Large block size - low time resolution, small block size - low frequency resolution
- Hop size: small hop size - very smooth over time, large hop size - less smooth

5 Wavelet Transform

- $X(s,\tau) = \int_t x(t)\phi^*_{s,\tau}(t)\,dt$
- Similar to Fourier transform, $s,\tau \rightarrow F$
- $s \rightarrow$ scaling
- $\tau \rightarrow$ transformation
- Idea: start with a simple wavelet pulse function. Shift it to infinite positions in time domain, stretch it to shift in frequency domain.
- Essentially, STFT makes a grid in time/freq, wavelet creates a wavelet for each square and measures correlation
- Low frequencies have longer window length, high frequencies have shorter window lengths!