Lecture 2:
Signal Processing Reminder and Feature Extraction
DT2118 Speech and Speaker Recognition

Giampiero Salvi

KTH/CSC/TMH giampi@kth.se

VT 2016
Outline

**Signal Processing Reminder**
- Linear Time-Invariant Systems
- Sampling Theorem

**Speech Signal Representations**
- Linear Prediction Analysis (LPA)
- Mel Frequency Cepstral Coefficients (MFCC)
- Features and Time Evolution
Outline

Signal Processing Reminder
- Linear Time-Invariant Systems
- Sampling Theorem

Speech Signal Representations
- Linear Prediction Analysis (LPA)
- Mel Frequency Cepstral Coefficients (MFCC)
- Features and Time Evolution
Continuous vs Digital Signals

**Sampling:** Discretisation in time

**Quantisation:** Discretisation in amplitude

(Figures from Wikipedia)
Linear Time-Invariant (LTI) Systems

In general:

\[ y[n] = T(x[n]) \]

Time invariance:

\[ y[n - n_0] = T(x[n - n_0]) \]

Linearity:

\[ T(a_1 x_1[n] + a_2 x_2[n]) = a_1 T(x_1[n]) + a_2 T(x_2[n]) \]
**LTI: Impulse Response**

In general we can always write:

$$x[n] = \sum_{k=-\infty}^{\infty} x[k] \delta[n - k]$$

For the linearity:

$$y[n] = T(x[n]) = \sum_{k=-\infty}^{\infty} x[k] T(\delta[n - k])$$

Where $h[n] \equiv T(\delta[n])$ is the system’s response to an impulse $\delta[n]$.

For the time invariance:

$$T(\delta[n - k]) = h[n - k]$$

$h[n]$ is a complete description of the system!
Convolution

\[ y[n] = T(x[n]) = \sum_{k=-\infty}^{\infty} x[k]h[n - k] = x[n] * h[n] \]

Properties:

\[ x[n] * h[n] = h[n] * x[n] \]

Kind of complicated to interpret.
Sinusoidal signals are eigensignals for LTI systems: if \( x[n] = e^{j\omega_0 n} \) then

\[
y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = \\
= \sum_{k=-\infty}^{\infty} h[k]x[n - k] = \sum_{k=-\infty}^{\infty} h[k]e^{j\omega_0(n-k)} \\
= \sum_{k=-\infty}^{\infty} h[k]e^{-j\omega_0 k}e^{j\omega_0 n} = e^{j\omega_0 n} \sum_{k=-\infty}^{\infty} h[k]e^{-j\omega_0 k} \\
= H(\omega_0)e^{j\omega_0 n}
\]
Transfer Function

\[ H(\omega) = \sum_{k=-\infty}^{\infty} h[k] e^{j\omega k} \]

Sinusoidal signals:

\[ x[n] = e^{j\omega_0 n} \rightarrow y[n] = H(\omega_0) e^{j\omega_0 n} \]

\( \omega = 2\pi f \), where \( f \) is the frequency
Fourier Transforms

Fourier transform of continuous signals

\[ X(\omega) = \int_{-\infty}^{\infty} x(t)e^{j\omega t} \, dt \]

Fourier transform of discrete signals

\[ X(\omega) = \sum_{k=-\infty}^{\infty} x[k]e^{j\omega k} \]

Discrete Fourier Transform

\[ X[n] = \sum_{k=-\infty}^{\infty} x[k]e^{j2\pi \frac{k}{K}n} \]
Transfer Function for Generic Signals

Sinusoidal signals:
\[ x[n] = e^{j\omega_0 n} \rightarrow y[n] = H(\omega_0) e^{j\omega_0 n} \]

Generic signals (can be decomposed in sinusoids):
\[ Y(\omega) = H(\omega) X(\omega) \]

\[ \omega = 2\pi f, \text{ where } f \text{ is the frequency} \]
Examples of Linear Systems

Pre-emphasis

\[ y[n] = x[n] - \alpha x[n-1], \quad \text{with} \quad \alpha = 0.97 \]
Pre-emphasis in frequency domain

The graph illustrates the pre-emphasis in the frequency domain. The x-axis represents the frequency (radians), ranging from 0 to 3.0, and the y-axis represents the FFT (dB), ranging from -80 to 20. The graph compares the original spectrum (blue line) with the pre-emphasis (green line).
Pre-emphasis applied to vowel
Examples of Linear Systems

Moving average

\[ y[n] = x[n] + x[n - 1] + \cdots + x[n - P] \]
Finite Impulse Response (FIR) Systems

$y$ only depends on (delayed) samples of the input (no feedback)

\[ y[n] = b_0 x[n] + b_1 x[n - 1] + \cdots + b_P x[n - P] \]

\[ = \sum_{i=0}^{P} b_i x[n - i] \]
Infinite Impulse Response (IIR) Systems

Auto regressive (AR): $y$ depends on (delayed) samples of the input, as well as the output at previous times (feedback)

$$
y[n] = \frac{1}{a_0} \left( b_0 x[n] + b_1 x[n - 1] + \cdots + b_P x[n - P] + \right.

- a_1 y[n - 1] - a_2 y[n - 2] - \cdots + a_Q y[n - Q])

= \frac{1}{a_0} \left( \sum_{i=0}^{P} b_i x[n - i] - \sum_{j=1}^{Q} a_j y[n - j] \right)
$$
IIR Example

\[ y[n] = x[n] - ay[n - 1] \]

stable only if \(|a| < 1\), here \(a = -0.8\)
Sampling Theorem (Nyquist-Shannon)

If $x(t)$ contains energy up to $B_x$, in order to reconstruct the signal we need to sample with

$$f_s > 2B_x$$
Aliasing

\[ X(\Omega) \]

\[ X_p(\Omega) \]

\[ \cdots -\Omega_s -\Omega_s/2 \Omega_s/2 \Omega_s \cdots \]

\[ \cdots -\Omega_s -\Omega_s/2 \Omega_s/2 \Omega_s \cdots \]

Figure from Huang, Acero and Hon (2001)
Outline

Signal Processing Reminder
  Linear Time-Invariant Systems
  Sampling Theorem

Speech Signal Representations
  Linear Prediction Analysis (LPA)
  Mel Frequency Cepstral Coefficients (MFCC)
  Features and Time Evolution
Components of ASR System

- Speech Signal
- Spectral Analysis
- Feature Extraction
- Search and Match
- Recognised Words

Representation

Constraints - Knowledge

- Acoustic Models
- Lexical Models
- Language Models

Decoder
Speech Signal Representations

Goals:
- disregard irrelevant information
- optimise relevant information for modelling
Speech Signal Representations

Means:

- try to model essential aspects of speech production
- imitate auditory processes
- consider properties of statistical modelling
First step: represent speech signal

- Pressure wave converted into electric current (microphone)
- Sampling
  - Nyquist-Shannon Theorem: sample at twice the band
  - 8kHz (4kHz band, telephone), 16kHz (8 kHz band, high quality)
  - TIDIGITS sampled at 20kHz
  - TIMIT sampled at 16kHz
- Quantisation
  - Type of quantisation: linear, a-law, μ-law
  - 8, 16 bits (more rare 32, floating point)
  - TIDIGITS and TIMIT are quantised with 16 bits linear
A time varying signal

- speech is time varying
- short segment are quasi-stationary
- use short time analysis
Short-Time Fourier Analysis
Short-Time Fourier Analysis

![FFT, window length: 512, window kind: hamming](image-url)
Short-Time Fourier Analysis

FFT, window length: 512, window kind: hamming
Short-Time Fourier Analysis

FFT, window length: 512, window kind: hamming
Short-Time Fourier Analysis

Effect of different window functions

Male voice /ah/  F0 = 110 Hz

Rectangular 30 ms

Hamming 30 ms

Rectangular 15 ms

Hamming 15 ms

Window should be long enough to cover 2 pitch pulses
Short enough to capture short events and transitions
Windowing, typical values

- signal sampling frequency: 8–20kHz
- analysis window: 10–50ms
- frame step: 10–25ms (100–40Hz)
Pre-emphasis

Compensate for the 6db/octave drop (radiation at the lips)

\[ y[n] = x[n] - \alpha x[n-1] \]

Corresponds to a linear filter with \( A = 1 \) and \( B = [1 \quad -\alpha] \)

\( \alpha \) is usually 0.95–0.97
$F_0$ and Formants

- **Varying $F_0$ (vocal fold oscillation rate)**

- **Varying Formants (vocal tract shape)**
Linear Prediction Coefficients (LPC)

- assume all-pole model:

\[ H(z) = \frac{S(z)}{U_g(z)} = AG(z)V(z)R(z) \triangleq \frac{A}{1 - \sum_{k=1}^{p} a_k z^{-k}} \]
Linear Prediction Coefficients (LPC)

- Assume all-pole model:

\[ H(z) = \frac{S(z)}{U_g(z)} = AG(z)V(z)R(z) \triangleq \frac{A}{1 - \sum_{k=1}^{p} a_k z^{-k}} \]

- The output signal \( s[n] \) can be expressed as the sum of the input \( u_g[n] \) and a number of previous samples \( a_k s[n - k] \):

\[ s[n] = \sum_{k=1}^{p} a_k s[n - k] + A u_g[n] \]
LPC Example

\[ s[n] = \sum_{k=1}^{p} a_k s[n - k] + A u_g[n] \]
Perceptual Linear Prediction

- Transform to the Bark frequency scale before computing the LPC coefficients
- Cubic root of energy instead of logarithm
LPC Limitations

- better match at spectral peaks than at valleys
- not accurate if transfer function contain zeros (nasals, fricatives...)
Mel Frequency Cepstrum Coefficients

- *de facto* standard in ASR (before Deep Learning)
- imitate aspects of auditory processing
- does not assume all-pole model of the spectrum
- uncorrelated: easier to model statistically
MFCCs Calculation

- pre-emph
- windowing
- FFT
- $|.|^2$

Spectrum of /a:/

Cepstrum of /a:/
Mel Frequency Cepstral Coefficients
Mel Frequency Cepstral Coefficients

Linear to Mel frequency
Mel Frequency Cepstral Coefficients

Linear to Mel frequency

Filterbank (∼ 20-25 filters) + log()
Mel Frequency Cepstral Coefficients

Linear to Mel frequency

Filterbank (≈ 20-25 filters) + log()

Discrete Cosine Transform
MFCC: Cosine Transform

$$C_j = \sqrt{\frac{2}{N}} \sum_{i=1}^{N} A_i \cos\left(\frac{j\pi(i-0.5)}{N}\right)$$

![Spectra and cepstra of /a:/ and /s:/ sounds with filterbank channels and their respective cepstra.](image)
MFCC Rationale

- signals combined in a convolutive way: $a[n] \ast b[n] \ast c[n]$
- in the spectral domain: $A(z)B(z)C(z)$
- taking the log: $\log(A(z)) + \log(B(z)) + \log(C(z))$
- to analyse the different contribution perform Fourier transform (DCT if not interested in phase information).
MFCC Rationale

- signals combined in a convolutive way: $a[n] * b[n] * c[n]$
- in the spectral domain: $A(z)B(z)C(z)$
- taking the log: $\log(A(z)) + \log(B(z)) + \log(C(z))$
- to analyse the different contribution perform Fourier transform (DCT if not interested in phase information).
- Terminology:
  - frequency vs quefrency
  - spectrum vs cepstrum
  - filter vs lifter
  - ...
MFCC Advantages [1]

- fairly uncorrelated coefficients (simpler statistical models)
- high phonetic discrimination (empirically shown)
- do not assume all-pole model
- low number of coeff. enough to capture coarse structure of spectrum
- Cepstral Mean Subtraction corresponds to channel removal

MFCCs: typical values

- 12 Coefficients C1–C12
- Energy (could be C0)
- Delta coefficients (derivatives in time)
- Delta-delta (second order derivatives)
- total: 39 coefficients per frame (analysis window)
Segment-Based Processing
Landmark-Based Processing
Frame-Based Processing

File: sx352.WAV   Page: 1 of 1   Printed: Mon Dec 05 09:01:39

kHz

0.05 0.10 0.15 0.20 0.25 0.30 0.35 0.40 0.45 0.50 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95
time
ttclnixaymaxttclniydxraokkclixh#
my
to
according

h# ix kcl k ao r dx iy n tcl t ax m ay ix n tcl t

according to my