Signal Processing For Speech Applications - Part 2

May 22, 2012
References

- Huang et al., Chapter on DSP
- Classical paper: Schafer/Rabiner in Waibel/Lee (on the web)
- Nahin: "Dr. Euler's Fabulous Formula" – excellent explanation of Fourier sums and the Fourier Transform, written for Engineering students

Note: many slides of this lecture are from Rich Stern
What we have seen so far …

Short-Term Spectral Analysis

- Multiplication with window function
- Discrete Time Fourier Transform (DTFT)
- Mel-scaled filterbank
**Facts:**

- The frequency distribution over an entire utterance does not help much for recognition.
- Most acoustic events (e.g. phonemes) have durations in the range of 10 to 100 ms.
- Many acoustic events are not static (diphtongs) and need more detailed analysis.

**Solution:**

- Partition the entire recording in a sequence of **short segments**
- The segments may **overlap** each other
Short-time Fourier Analysis

• **Problem:** Conventional Fourier analysis does not capture **time-varying nature** of speech signals

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

• **Solution:** Multiply signals by **finite-duration window** function, then compute DTFT:

\[
X[n, \omega] = \sum_{m=0}^{N-1} x[m]w[n - m]e^{-j\omega m}
\]

• **Side effect:** Windowing causes **spectral blurring**
Using Filterbanks

- All Fourier coefficients reflect too much of the signals microstructure
- The microstructure contains redundancies but also a lot of "misleading" information
- Solution – **Filterbanks**: The human **ear** also works with "filterbanks"
- Different approaches to computing filterbank coefficients:

  **Fixed width filters:**

  0  1000  2000  3000  4000  5000  6000  7000  8000

  **Variable width:**

  ...500 1000  2000  4000  8000

  **Overlapping filters:**

  0  1000  2000  3000  4000  5000  6000  7000  8000

- **Typical filterbanks**: **mel** or **bark** scales
Additionally, we need the following for a conventional preprocessing:

- Cepstrum
- Delta Coefficients
Overview (I)

- The Source-Filter Model For Speech
- The Cepstrum
- Features For Speech Recognition: Cepstral Coefficients
  - The Mel-Cepstrum
  - Computing Mel Frequency Cepstral Coefficients (MFCC)
    - Example:
      - Deriving MFCC coefficients
      - Weightening the Frequency Response
      - The Actual Mel Weighting Functions
      - Log Energies Of Mel Filter Outputs
      - The Cepstral Coefficients
      - Logspectra Recovered From Cepstra
      - Comparing Spectral Representations
  - Computing Delta Coefficients
Overview (II)

- Features For Speech Recognition: Cepstral Coefficients
  - Z-transform
    - Relationship DTFT and Z-transform
    - Filtering
      - Why Filtering?
      - Linear time-invariant (LTI) filter
      - Filters as difference equations
      - Poles And Zeros
  - Summary Of Z-transform Discussion
Overview (III)

• Features For Speech Recognition: Cepstral Coefficients
  – Linear Predictive Coding
  – Linear Prediction Of Speech
  – Two Ways Of Deriving Cepstral Coefficients
  – Computing LPC Cepstral Coefficients
  – The Time Function After Windowing
  – The Raw Spectrum
  – Pre-emphasizing The Signal
  – The Spectrum Of The Pre-emphasized Signal
  – The LPC Spectrum
  – The Transform Of The Cepstral Coefficients
  – The Original Spectrogram
  – Effects Of LPC Processing
  – Comparing Representations
  – Summary
The Source-Filter Model For Speech

- Sounds are produced either by
  - vibrating the vocal cords (voiced sounds) or
  - random noise resulting from friction of the airflow (unvoiced sounds)
  - voiced fricatives need a mixed excitation model
- Signal $u_n$ is modulated by the vocal tract, whose impulse response is $v_n$
- Resulting signal is modulated by the lips and nostrils' radiation response $r_n$
- Eventually the resulting signal $f_n$ is emitted.
Remember the **source-filter model** of speech production.

So if \( f = e \ast h \),

- \( \text{FT}\{f\} = \text{FT}\{e\} \cdot \text{FT}\{h\} \)
- \( \log \text{FT}\{f\} = \log \text{FT}\{e\} + \log \text{FT}\{h\} \)
- \( \text{FT}^{-1}\{\log \text{FT}\{f\}\} = \text{FT}^{-1}\{\log \text{FT}\{e\}\} + \text{FT}^{-1}\{\log \text{FT}\{h\}\} \)

- It can be clearly seen that the transformation \( \text{FT}^{-1}\{\log \text{FT}\{f\}\} \) deconvolves the excitation signal \( e \) and the channel \( h \).
- The coefficients of this transformation are called **cepstral coefficients** or simply **cepstrum**.
- If we assume the channel to be constant during an utterance, we can subtract the average cepstrum from every short-time cepstrum and eliminate the channel effect.
Features For Speech Recognition: Cepstral Coefficients (I)

- The **cepstrum** is the *inverse Fourier transform of the log of the magnitude of the spectrum*

- Sometimes also called the **spectrum of the spectrum**

- Useful for separating convolved signals (like the source and filter in the speech production model)

- I.e. the low-frequency periodic excitation from the vocal cords and the formant filtering of the vocal tract, which are
  - convolved in the time domain
  - multiplied in the frequency domain,
  - but additive and in different regions in the quefrency domain
Features For Speech Recognition: Cepstral Coefficients (II)

- **The cepstrum** can be seen as *information about rate of change in the different spectrum bands*
- **Cepstral Coefficients** provide efficient and robust coding of speech information
- Most common basic feature for speech recognition !!!
- Example of application:
  Pitch extraction - Effects of the vocal excitation (pitch) and vocal tract (formants) are additive and thus clearly separate
- Its name CEPSTRUM was derived by reversing the first four letters of "spectrum"
- Operations on cepstra are labelled *quefrency analysis*, *liftering*, or *cepstral analysis*
The Mel-Cepstrum

- For speech recognition, only the lower cepstral coefficients are used.
- When we set some of the coefficients to 0.0 then this process is called liftering (in analogy to corresponding operation on spectrum).
- The lower coefficients reflect the macrostructure of the spectrum.
- The higher coefficients reflect the microstructure of the spectrum.
- The 0th coefficients reflects the signal energy.
- The independent variable of a cepstral graph is called the quefrency.
- Quefrency is a measure of time, but not in sense of a signal in time domain.
- Example: If the sampling rate of an audio signal is 44100 Hz and the cepstrum has a peak at quefrency = 100 samples, the peak indicates the presence of a pitch that is 44100/100 = 441 Hz. This peak occurs because the harmonics in the spectrum are periodic, and the period corresponds to the pitch.
Computing Mel Frequency Cepstral Coefficients (MFCC)

1. Segment incoming waveform into **frames** (10 ms)
2. Compute frequency response for each frame using **DTFT**
3. **Group magnitude of frequency response** into 25-40 channels using triangular weighting functions
4. Compute **log** of weighted magnitudes for each channel
5. Take **inverse DTFT** of weighted magnitudes for each channel, producing ~13 cepstral coefficients for each frame
6. (Calculate delta and double-delta coefficients OR frame stacking)
Example: Deriving MFCC coefficients

1. Segment incoming waveform into frames
2. Compute frequency response for each frame using DTFT
3. Group magnitude of frequency response into 25-40 channels using triangular weighting functions
Example: The Actual Mel Weighting Functions
Example: Log Energies Of Mel Filter Outputs

4. Compute log of weighted magnitudes for each channel
Example: The Cepstral Coefficients

5. Take inverse DTFT of weighted magnitudes for each channel, producing ~13 cepstral coefficients for each frame
Example: Logspectra Recovered From Cepstra

- Only first 13 coefficients used but macrostructure is conserved.
Example: Comparing Spectral Representations

ORIGINAL SPEECH

MEL LOG MAGS

AFTER CEPSTRA

8000 coefficients

13 coefficients → 39 dimensional instead of 8000 dimensional
Computing Delta Coefficients

• Comments:
  – MFCC is currently the most popular representation.
  – Typical systems include a combination of
    • MFCC coefficients
    • “Delta” MFCC coefficients
    • “Delta delta” MFCC coefficients
    • Power and delta power coefficients

• Deltas are acceleration features that measure the change of a signal
  – e.g. Delta: -3 -2 -1 0 1 2 3

• Or use frame stacking
Computing Delta Coefficients

- **Frame stacking**

- **Delta / Delta delta**
Z-transform

- The **Z-transform** is a generalization of the discrete-time Fourier transform (DTFT).
- In particular we can use it to describe the effect of filters.
- Let’s take a look at the DTFT.

A signal $x[k]$ is transformed to

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

- The Z-transform of $x[k]$ is

$$X(z) = \mathcal{Z}(x[k]) = \sum_{k=-\infty}^{\infty} x[k] z^{-k}.$$ 

where $z$ is a complex number and

$$z^n = r^n e^{j\phi n} = r^n (\cos(\phi n) + j \sin(\phi n)).$$
Relationship DTFT and Z-transform

- What is the relationship?
- The Z-transform considers the complex plane, the DTFT only the unit circle.
- The DTFT is the Z-transform restricted to the unit circle!

Example:

![Z-transform (absolute value)](image1)

![DTFT (absolute value)](image2)
Filtering

- A filter transforms an input signal into an output signal

- Examples for filters:
  - Acoustic filters (e.g. exhaust of a car, concert hall, vocal tract)
  - Analog (electronic) filters (combination of resistors, capacitors, and inductors)
  - Digital filters (sequence of coefficients)
Why Filtering?

1. Filters influence the frequencies of an input signal. Therefore several important signal processing steps (e.g. modulation, noise reduction) can be applied with filters.

2. Filters occurring in the nature can be simulated and described with digital filters. In this way we can model certain steps of the development of a signal.

3. Human senses often work frequency-dependently. For example, the eyes perceive electromagnetic waves of different frequencies as different colors.

4. Filtering is a very “fundamental” operation.
Linear time-invariant (LTI) filter

- Let $H$ be a filter which transforms an input signal $x[n]$ into an output signal $y[n]$. 

- We take 2 assumptions about the property of this filter:
  - **Linearity**: $y[\cdot]$ is a linear function of $x[\cdot]$ 
  - **Time invariance**: The properties of $H$ do not change over time 

- Not that important for us but also helpful:
  - Causality: The output of the filter depends on the past 
  - A limited input signal should produce only a limited output signal (for now) 

- Now we excite the **linear time-invariant (LTI) filter** with a Dirac impulse and get a (finite) output signal $h[n]$ 

- $h[n]$ is called the **impulse response** of the filter. 

- What happens if we use a complex signal as input of the filter?
Linear time-invariant filter (2)

- Let $x[n]$ be an arbitrary signal.

- $x$ is a weighted sum of shifted impulses!

- As $H$ is linear (and time-invariant), the output $y$ is already defined by the impulse response $h[n]$:

$$y[n] = \sum_{\nu=-\infty}^{\infty} x[\nu] \cdot h[n-\nu]$$

- This operation is also called (discrete) convolution:

$$x \ast h := \sum_{\nu=-\infty}^{\infty} x[\nu] \cdot h[n-\nu]$$

- Then the output signal is $y=x \ast h$. 

\[\sum_{n} x[n] \ast h[n] = y[n]\]
Filters as difference equations

- Let $H$ be a recursive filter.
  \[ x[n] \xrightarrow{\text{Filter } H} y[n] \]

- As for non-recursive filters, we can characterize recursive filters with a convolution.

- In the time-domain, we get a difference equation.

- Example:
  \[ y[n] = -a_1 y[n-1] - a_2 y[n-2] - \ldots - a_m y[n-m] + b_0 x[n] + \ldots + b_l x[n-l] \]
  also:
  \[ y[n] + a_1 y[n-1] + a_2 y[n-2] + \ldots + a_m y[n-m] = b_0 x[n] + \ldots + b_l x[n-l] \]

- How does it look in the Z-domain (or frequency domain)?
Filters as difference equations (2)

- The transform into the Z-domain works as described, where
  - Left: \( y[n] + ... + a_m y[n-m] = (a * y)[n] : A(z) \cdot Y(z) \)
  - Right: \( b_0 x[n] + ... + b_l x[n-l] = (b * x)[n] : B(z) \cdot X(z) \)
  and
    - \( b = (b_0, ..., b_n) \)
    - \( a = (1, a_1, ..., a_n) \) (the coefficient \( a_0 \) is normalized to 1).

- Now we can define a **Z-transfer function**
  \[
  H(z) = \frac{Y(z)}{X(z)} = \frac{B(z)}{A(z)}
  \]

- ... it is given by the Z-transform of the sequence of coefficients.
- From the the filtering we get a **multiplication** in the Z-domain.
Filters as difference equations (3)

- **Example** (Difference equation characterizing system):
  \[ y[n] - 1.27y[n-1] + 0.81y[n-2] = x[n] - x[n-1] \]

- The sequence of coefficients is \( a=(1, -1.27, 0.81) \) and \( b=(1, -1) \).

- Transform into the Z-domain:
  \[
  A(z) = 1 - 1.27z^{-1} + 0.81z^{-2} \quad \text{and} \quad B(z) = 1 - z^{-1}
  \]

- The Z-transfer function is
  \[
  H(z) = \frac{Y(z)}{X(z)} = \frac{B(z)}{A(z)} = \frac{1 - z^{-1}}{1 - 1.27z^{-1} + 0.81z^{-2}}
  \]
Poles And Zeros

- **Poles** and **zeros** are the roots of the denominator and numerator of LSI systems:

\[
H(z) = \frac{Y(z)}{X(z)} = \frac{1 - z^{-1}}{1 - 1.27z^{-1} + .81z^{-2}}
\]

\[
= \frac{z(z - 1)}{(z - .9e^{j\pi/4})(z - .9e^{-j\pi/4})}
\]

- **Zeros** of system are at \( z = 0, \quad z = 1 \)
- **Poles** of system are at \( z = .9e^{j\pi/4}, \quad z = .9e^{-j\pi/4} \)
We can determine the DTFT from the Z-transform:

To find the absolute value of the DTFT, consider the Z-transform on the unit circle.

For each point on the unit circle: The absolute value of the DTFT results from the product of the distances to the zeros divided by the product of the distances to the poles of the Z-transform.

We can apply these rules to describe the impact of the filter on certain frequencies.

For this purpose, we have to consider the whole Z-plane, not only the unit circle.
Summary Of Z-transform – Discussion

- Z-transforms are a generalization of DTFTs

- **Difference equations** can be easily obtained from Z-transforms

- Locations of **poles** and **zeros** in z-plane provide insight about frequency response
Linear Predictive Coding

- Alternative method to represent the speech signal
- Idea:
  In speech signals, periodicity can be expressed by rules how samples can be approximated from past samples.

\[ s[n] \approx a_0 + a_1s[n-1] + a_2s[n-2] + \cdots + a_ps[n-p] \]

- The actual signal, of course, differs from the estimated signal, such that:

\[ s[n] = -\sum_{j=1}^{p} a_j s[n - j] + e[n] \Rightarrow e[n] = s[n] - \hat{s}[n] = \sum_{j=0}^{p} a_j s[n - j] \]

which after a Z-transform becomes:

\[ E(z) = S(z) \cdot A(z) \quad \text{or} \quad S(z) = E(z) \cdot 1 / A(z) \]
Linear Predictive Coding (2)

- When we want to find good LPC coefficients $a_j$, we have to minimize the squared error:

\[
\sum_{n=0}^{N} e_n^2 = \sum_{n=0}^{N} \left( \sum_{j=0}^{p} a_j f_{n-j} \right)^2
\]

- i.e. we have to find $a_j$, such that the error is minimized. Eventually, this leads to a system of linear equations.

- Interpretation of LPC coefficients:
  The values of the z-transform of the LPC-coefficients on the unit-circle approximate the spectrum of the signal.
Linear Prediction Of Speech

• Find the best “all-pole” approximation to the DTFT of a segment of speech:

\[
H(z) = \frac{G}{1 - \sum_{k=1}^{P} \alpha_k z^{-k}}
\]

• All-pole model is reasonable for most speech
• Very efficient in terms of data storage
• Coefficients \( \{a_k\} \) can be computed efficiently
• Phase information not preserved (not a problem for us)
Linear Prediction – Example

- Spectra from the /ih/ in “six”:

- Comment: LPC spectrum follows peaks well
Two Ways Of Deriving Cepstral Coefficients

- Mel-frequency cepstral coefficients (MFCC):
  - Compute log magnitude of windowed signal
  - Multiply by triangular Mel weighting functions
  - Compute inverse discrete cosine transform

- LPC-derived cepstral coefficients (LPCC):
  - Compute “traditional” LPC coefficients
  - Convert to cepstra using linear transformation
  - Warp cepstra using bilinear transform
Computing LPC Cepstral Coefficients

- Example Procedure:
  - A/D conversion at 16 kHz sampling rate
  - Apply Hamming window, duration 320 samples (20 msec) with 50% overlap (100-Hz frame rate)
  - Pre-emphasize to boost high-frequency components
  - Compute first 14 auto-correlation coefficients
  - Perform Levinson-Durbin recursion to obtain 14 LPC coefficients
  - Convert LPC coefficients to cepstral coefficients
  - Perform frequency warping to spread low frequencies
  - (Apply vector quantization to generate three codebooks)
An example: the vowel in “welcome”

- The original time function:
The Time Function After Windowing
The Raw Spectrum

![Raw DFT Coefficients]

- Spectral magnitude, dB
- f, Hz
The Spectrum Of The Pre-emphasized Signal
The LPC Spectrum

LPC Spectrum
Pre-emphasized DFT Coefficients
The Transform Of The Cepstral Coefficients

Spectrum from LPC Cepstrum
LPC Spectrum

Spectral magnitude, dB

Frequency, Hz
The Original Spectrogram
Effects Of LPC Processing
Comparing Representations

ORIGINAL SPEECH
(unwarped)

LPCC CEPSTRA
Summary

Accomplish **feature extraction** for speech recognition

Some specific topics:
- Quantization (A/D Conversion)
- Sampling
- Filter Bank Coefficients
- Linear predictive coding (LPC)
- LPC-derived cepstral coefficients (LPCC)
- Mel-frequency cepstral coefficients (MFCC)

Some of the underlying mathematics
- Continuous-time Fourier transform (CTFT)
- Discrete-time Fourier transform (DTFT)
- Z-transform
Thanks for your interest!