Signal Processing
For Speech Applications
- Part 2

May 14, 2013
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
Short-Term Spectral Analysis

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 (diphongs) 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

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

- **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 and "misleading" information
- Solution – **Filterbanks**: The human **ear** also works with "filterbanks"
- Filterbanks cause a *reduction of resolution* in the frequency domain
- Different approaches to computing filterbank coefficients:

  **Fixed width filters:**

  ![Fixed width filters diagram]

  0  1000  2000  3000  4000  5000  6000  7000  8000

  **Variable width:**

  ![Variable width filters diagram]

  ...500  1000  2000  4000  8000

  **Overlapping filters:**

  ![Overlapping filters diagram]

  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

We will also look into

- Filtering
- Linear Predictive Coding
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)
  – 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 plus lips/nostrils, signal $f_n$ is emitted
- We will show later that this modulation (which we call $h$) is a convolution in the time domain (and consequently a multiplication in the frequency domain)

Diagram:
- Glottis (vowels) $s_n$
- White noise (consonants)
- Excitation function $e$
- Vocal tract $v_n$
- Lips/nostrils $r_n$
- Channel/Filter $h$
- Output signal $f_n$
The Cepstrum

Remember the **source-filter model** of speech production.

\[ f = e \ast h, \]

\[
\begin{align*}
\text{if} & \quad \mathcal{F}(f) = \mathcal{F}(e) \cdot \mathcal{F}(h), \\
\text{then} & \quad \log \mathcal{F}(f) = \log \mathcal{F}(e) + \log \mathcal{F}(h), \\
\text{and} & \quad \mathcal{F}^{-1}(\log \mathcal{F}(f)) = \mathcal{F}^{-1}(\log \mathcal{F}(e)) + \mathcal{F}^{-1}(\log \mathcal{F}(h))
\end{align*}
\]

• It can be seen that the transformation \( \mathcal{F}^{-1}(\log \mathcal{F}(f)) \) 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 excitation to be constant during an utterance, we can subtract the average cepstrum from every short-time cepstrum and eliminate the excitation.
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 cepstrum
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 quefreny alanysis, liftering, or cepstral analysis.
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: filtering). 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**. Example: The pitch and harmonics in the spectrum (left) appear as a peak in the cepstrum at 200Hz.
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 filterbanks

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
Example: Weightening the Frequency Response

3. Group magnitude of frequency response into 25-40 channels using triangular weighting functions (filterbanks)
Example: Log Energies Of Mel Filter Outputs

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
Example: Logspectra Recovered From Cepstra

- Recover spectrum with the first 13 cepstral coefficients
- Macrostructure is conserved.
Example: Comparing Spectral Representations

ORIGINAL SPEECH

MEL LOG MAGS

CEPSTRA
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**
The Z-transform is a generalization of the discrete-time Fourier transform (DTFT).

In particular, we will 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)  DTFT (absolute value)
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, but also criteria:
  - **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?

\[ \delta[n] = \begin{cases} +1 \text{ for } n = 0 \\ 0 \text{ else} \end{cases} \]

Wikipedia, „Dirac Delta Function“
Linear time-invariant filter (2)

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

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

\[
x[n] = \sum_{\nu} x[\nu] \cdot \delta[n-\nu]
\]

- 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 the discrete convolution:

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

- Then the output signal is $y = x \ast h$. 
How is a filter described in the frequency (or z) domain?

The convolution $y = x * h$ becomes a **multiplication** in the z-domain:

$$Y(z) = H(z) \cdot X(z), \text{ or } Y(e^{j\omega}) = H(e^{j\omega}) \cdot X(e^{j\omega}).$$

This means that filters boost or attenuate **frequencies**.

$H$ is called **transfer function**.

This also applies to all filters in nature which follow the generic rules we defined in the beginning (linearity, time-invariance)

Figure: example transfer function of a lowpass filter
Linear time-invariant filter (4)

• So far we have assumed that a filter can be described by a simple convolution:
  \[ y[n] = b_0 x[n] + ... + b_l x[n-l] = b \ast x \]

• Additionally one considers filters where the output has a (time-delayed) effect on the input (think of an echo!)
  \[ y[n] = -a_1 y[n-1] - a_2 y[n-2] - ... - a_m y[n-m] + b_0 x[n] + ... + b_l x[n-l] \]

• These filters have the property that the impulse response can be infinite!
  • In practice, it converges to zero

• Definition: If a filter output is affected by previous output, the filter is **recursive** or IIR (infinite impulse response)

• Otherwise, the filter is **FIR (finite impulse response)** or **non-recursive**
Filters as difference equations

• Let $H$ be a **recursive filter**.

  \[ x[n] \xrightarrow{\text{Filter } H} y[n] \]

• We can characterize recursive filters with a similar idea as before.

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

• Example:

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

  **thus:**

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

• These are two convolutions: The second equation reads

  \[ y * a = x * b \]

  where we set $a_0 = 1$. 
Filters as difference equations (2)

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

\[ H(z) = \frac{Y(z)}{X(z)} = \frac{B(z)}{A(z)} = \frac{1 - z^{-1}}{1 - 1.27z^{-1} + 0.81z^{-2}} \]
We can rewrite the transfer function using the roots of the numerator and denominator polynomials:

\[ H(z) = \frac{Y(z)}{X(z)} = \frac{1-z^{-1}}{1-1.27z^{-1}+0.81z^{-2}} = \frac{z(z-1)}{(z-0.9e^{j\pi/4})(z-0.9e^{-j\pi/4})} \]

- **Zeros** of system are at \( z = 0, z = 1 \): The roots of the numerator
- **Poles** of system are at \( z = 0.9e^{j\pi/4}, z = 0.9e^{-j\pi/4} \): The roots of the denominator

Remember that \( H(z) \) is the effect of a filter:

\[ Y(z) = H(z) \cdot X(z) \]

We just look at the amplitude spectrum:

\[ |Y(z)| = |H(z)| \cdot |X(z)| = \frac{|z||z-1|}{|z-0.9e^{j\pi/4}||z-0.9e^{-j\pi/4}|} \cdot |X(z)| \]
Poles And Zeros

• For each frequency, i.e. each point on the unit circle, the absolute value of the transfer function results from the product of the distances to the zeros divided by the product of the distances to the poles of the Z-transform.

• This means that we can determine the behavior of the filter from the location of the poles and zeros in the z-plane, and that we can use this to design filters with specific properties!

• Typical filters:
  – lowpass, highpass: Allow certain frequencies to pass
  – differentiator (not important for us), …

• Figure: a complicated example of a lowpass filter, with visualization of z transformation of transfer function
Another filter which is frequently used in speech processing: \textit{pre-emphasis}

Idea: In speech, low frequencies are too dominant: make that more balanced

See the example picture. Can we achieve that with a filter?
Pre-emphasis (2)

• A typical pre-emphasis filter:

\[ y[n] = x[n] - 0.96x[n - 1] \]

• The figure shows the magnitude response (the absolute value of the transfer function). We see that low frequencies are indeed attenuated.
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.

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

• The order \( p \) is fixed. The "minus" sign makes our next formula easier to read.
• The actual signal, of course, differs from the estimated signal, such that:

\[ s[n] = \sum_{k=1}^{p} -a_k s[n-k] + e[n] \quad \text{or} \quad e[n] = \sum_{k=0}^{p} a_k s[n-k] \]

Error function

with \( a_0 = 1 \), which after a Z-transform becomes:

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

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

$$
\sum_{n=0}^{N} e[n]^2 = \sum_{n=0}^{N} \left( s[n] + \sum_{k=1}^{p} a_k s[n-k] \right)^2
$$

- i.e. we have to find $a_j$ such that the error is minimized.
- Eventually, this leads to a system of linear equations which can be easily solved with an arbitrary method.
- 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

- Why does $A(z)$ approximate the speech spectrum?
- From the source-filter model we have $S(z) = E(z) \cdot H(z)$. $E(z)$ is the excitation, $H(z)$ is the vocal tract filter.
- Now we have $S(z) = E(z) \cdot 1 / A(z)$, i.e. $H(z)$ is estimated by an "all-pole" approximation $A(z)$.
- The information about the excitation (and the phase) is lost -> nice, we don't need that anyway.
- One can show that for speech understanding, the poles are most important -> the all-pole model is reasonable for most speech.
- Very efficient in terms of data storage
- Coefficients $\{a_k\}$ can be computed efficiently
Linear Prediction – Example

- Spectra 1

- LPC spectrum follows peaks well
- Useless microstructure is lost
Two Ways Of Deriving Cepstral Coefficients

- Now one can apply a cepstral transformation to the LPC coefficients, yielding LPCCs (LPC-derived cepstral coefficients). Compare:

- 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
An example: the vowel in “welcome”

- The original time function:
The Time Function After Windowing

The vowel “UH”

(time, sec)
The Raw Spectrum

![Graph showing raw DFT coefficients with spectral magnitude in dB vs frequency in Hz.](image)
The Spectrum Of The Pre-emphasized Signal

Pre-emphasized DFT Coefficients

Raw DFT Coefficients
The LPC Spectrum

LPC Spectrum
Pre-emphasized DFT Coefficients
The Original Spectrogram
Effects Of LPC Processing

![Spectrogram showing 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
- Mel-frequency cepstral coefficients (MFCC)
- Linear predictive coding (LPC)
- LPC-derived cepstral coefficients (LPCC)

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