3. Feature Extraction
3.1 Feature Extraction from Speech

... or other types of audio like music
Goal of Feature Extraction

- Capture essential information about speech
- Be robust against background noise
- Steps:
  - Sampling and quantization
  - Short time analysis
  - Transform to frequency space
  - Filtering
  - Optimize class separability
Overview Feature Extraction

Convert the continuous speech signal into a sequence of vectors

Each window gives one vector

The following slides will give the details of this procedure

From: HTK-manual
Sampling and Quantization

what happens when you store a signal in a computer?

Measure signal periodically and store in variable
Sampling rate: T
Quantization: use B bits to represent signal \( \rightarrow 2^B \) possible values
\( f_n \): sampled values of the signal numbered using index n
Sampling Theorem

• Reconstruction of original signal is only possible if the signals highest frequency is limited

• Let $f_G$ the frequency limit

$$f_G \leq \frac{1}{2T}$$

• Else: spectral aliasing that is frequencies will be confused
Pre-emphasis

- Correct for filtering of the lips
- Boosts higher frequencies
- Iterative scheme:
  \[ f_n' = f_n - \alpha f_{n-1} \]
- Typical values: \( \alpha = 0.95 \)

What does it do for \( \alpha = 1 \)?
From Signal to Spectrum: Fourier Transform

- Definition

\[ F^{(m)}(e^{i\omega}) = \sum_{n=-\infty}^{\infty} f_n w_{m-n} e^{-i\omega n} \]

- \( w_n \): window function
- \( \omega \): frequency times \( 2\pi \)
- \( i \): imaginary unit

The window cut’s the sum to a number of finite values
Complex exponentials are easier than cos or sin functions
Example: putting a rectangular on a speech signal

Frame shift typ.: 10ms

Frame width typ.: 25ms
Fourier Transform in Practice

• Use “Fast Fourier Transform” (FFT)
• Requires number of samples $N$ to be power of 2 (e.g. $N=256$)
• Code available
• Complexity $N \log(N)$
Established Window Functions

- Use to get sharper peaks $w_n^R = 1$
- Rectangular window:
- Generalized Hamming Window:
  $$w_n^H = (1 - \alpha) - \alpha \cos\left(\frac{2\pi n}{N - 1}\right)$$  
  ($\alpha = 0.54$ : standard Hamming window)
- Gauss window: $w_n^G = e^{-0.5\left(\frac{n-N/2}{3N/2}\right)^2}$
- Parabola window: $w_n^P = 4\frac{n}{N}(1 - \frac{n}{N})$

$n = 0 \ldots N - 1$

- Window functions vanish outside this interval
Rewrite of Fourier Transform

• Definition: 

\[ F^{(m)}(e^{i\omega}) = \sum_{n=-\infty}^{\infty} f_n w_{m-n} e^{-i\omega n} \]

• Window functions vanish outside the interval \( n=0 \ldots N-1 \)

• Define \( \omega = 2\pi v \frac{1}{N} \)

\[ F_v^{(m)} = \sum_{n=0}^{N-1} f_{m-n} w_n e^{-i2\pi v \frac{n}{N}} \]
Example for ö

How can you best look at multiple spectra at the same time
Spectrogram

- Calculate a spectrum for any point in time
- Code the local intensity: color/grey scale
"To return to the main menu, press the star key".
Use Praat to generate a Spectrogram

• Praat: software for doing phonetics by computer
• Written by: Paul Boersma and David Weenink
• Quite powerful: spectrograms, formants, pitch, …
• Download: http://www.fon.hum.uva.nl/praat/

Praat: doing phonetics by computer

Download Praat:

* Macintosh
* Windows
* Linux, FreeBSD
* SGI, Solaris, HP UX
* the source code

Information on Praat:

* Introductory tutorial: choose Intro from Praat's Help menu
* Extensive manuals and tutorials in Praat's Help menus
* Beginner's manuals by Sidney Wood and Pascal van Lieshout
* Paul Boersma's publications on algorithms and tutorials.

The authors

Paul Boersma and David Weenink
Institute of Phonetic Sciences
University of Amsterdam
Herengracht 338
1016 CG Amsterdam
The Netherlands

Questions, problems, solutions:

1. Many problems can be solved by upgrading to version 4.5.01 of Praat.
2. Make sure you have read the Intro from Praat's Help menu.
3. If that does not help, use the Search button in Praat's manual window.
4. Or consult the Frequently Asked Questions directly.
5. There is a user group on the Internet: the Praat User List.
6. If none of the above helps, you may send mail to paul.boersma@uva.nl
Functionality

The following gives you an idea of the features of the Praat program. The links take you into the web copy of the manual. The same manual is also available from Praat's Help menus, in which case you can see the pictures and do searches.

Speech analysis:
- spectral analysis (spectrograms)
- pitch analysis
- formant analysis
- intensity analysis
- jitter, shimmer, voice breaks
- cochleagram
- excitation pattern

Speech synthesis:
- from pitch, formant, and intensity
- articulatory synthesis

Listening experiments:
- identification and discrimination tests

Speech manipulation:
- change pitch and duration contours
- filtering

Labelling and segmentation:
- label intervals and time points on multiple tiers
- use phonetic alphabet
- use sound files up to 2 gigabytes (3 hours)

Learning algorithms:
- feedforward neural networks
- discrete and stochastic Optimality Theory

Statistics:
- multidimensional scaling
- principal component analysis
- discriminant analysis
Use praat to generate a Spectrogram
Smoothing the Spectrum: filter bank

• Idea: imitate ear
  • Do an average over neighboring frequencies
  • Scale the frequencies according to the Mel or the Bark scale
  ↦ Reduction from 256 Fourier coefficients to 24 outputs of a filter bank
Example of a Filterbank

Energy in Each Band

MELSPECF
Filterbank

- Spacing of center frequency:
  - According to mel scale:
    \[ Mel(f) = 2595 \log_{10}(1 + \frac{f}{700}) \]

- Low frequency cut off:
  - E.g. 300 Hz (for telephone speech)

- High frequency cut off:
  - E.g. 3400 Hz (for telephone speech)

- Different settings for e.g. head set connected PC

How can you adjust to different vocal tracts?
Vocal Tract Length Normalization

• Idea:
  • Average position of formants depends on length of vocal tract
  • \( \rightarrow \) varying position of frequencies of filter bank
  • A kind of speaker adaptation
Vocal Tract Length Normalization: Frequency Warping

- Translation table for frequencies

- Keep minimum and maximum frequency unchanged

$$\alpha_{\text{min}} = 0.8 \text{ to } \alpha_{\text{max}} = 1.2$$
Training the Warping Factor

• Issue: how to scale for a specific speaker

• Slow version:
  • Use 11 different warping factors
  • Do speech recognition with all of them
  • Pick the best one

• Oldest approach

• Not very efficient

• Improvement: 10% less recognition errors
From Spectrum to Cepstrum

- Name: swapping of letters (spectrum/cepstrum)
- Useful as a preparation to remove channel distortions

What are examples of channel distortions?

- Cepstral mean subtraction (CMS) method to remove channel distortions
Definition “Cepstrum”

1. Signal
2. Fourier Transform
3. log
4. Discrete Cosine Transform
5. Cepstrum
Math for Cepstrum

- $e_n$: original signal (e.g. excitation from glotis)
- $f_n$: measured signal
- $h_n$: impulse response of channel (e.g. vocal tract, telephone, room acoustics)

$$f_m = \sum_{n=-\infty}^{\infty} h_{m-n} e_n$$
• Apply Fourier transform $F$

\[
F\{f_n\} = F\{\sum_{n=-\infty}^{\infty} h_{m-n}e_n\}
\]

• Use convolution theorem

\[
F\{f_n\} = F\{h_n\}F\{e_n\}
\]
Math for Cepstrum

- Apply logarithm

\[
\log(\mathcal{F}\{f_n\}) = \log(\mathcal{F}\{h_n\}) + \log(\mathcal{F}\{e_n\})
\]

- Impulse response and excitation now separated
- If stationary part of impulse response \(h_n\) can now be removed
Cepstrum: do discrete cosine transform after log

- Discrete cosine transform:

\[
c_n^{(m)} = \sqrt{\frac{2}{N}} \sum_{l=1}^{N} \log(F_l^{(m)}) \cos\left(\frac{\pi n (l + 1/2)}{N}\right) \quad n = 1, 2, \ldots
\]

You do not need to remember this formula
Dynamic Features

- Spectrum captures local aspects of speech
- Window size 25 ms
- Capture slow changes in spectrum
- Other name: delta features
Dynamic Features
• Capture slow changes in spectrum
Dynamic Features

- Calculate first and second derivatives
- Naïve approach to first derivative
  - Continuous function
    \[
    \frac{df(t)}{dt} \approx \frac{f(t + \Delta t) - f(t - \Delta t)}{2\Delta t}
    \]
  - Time discrete sampling
    \[
    \frac{df(t_m)}{dt} \approx \frac{f(t_{m+\Delta}) - f(t_{m-\Delta})}{2\Delta}
    \]

\( t_m \): m-th sample of the signal
Difference/Regression

i-th component of feature vector

Line through extremes

Regression curve

Sample

m-3  m-2  m-1  m  m+1  m+2  m+3

34
Regression Formula

\[
\frac{df(t)}{dt} = \frac{\sum_{i=1}^{M} i(f(t_{m+i}) - f(t_{m-i}))}{2\sum_{i=1}^{M} i^2}
\]

Can you make it agree with

\[
\frac{df(t)}{dt} \approx \frac{f(t + \Delta t) - f(t - \Delta t)}{2\Delta t}
\]
Dynamic Features

- Invented by Furui 1981
- Standard in any modern ASR system

- Alternative:
  - Linear mapping of neighboring feature vectors

- Issue:
  - Dimension of feature vectors
Linear Discriminant Analysis

- Method to decrease size of feature vector
- Maximize severability of class regions
- Linear transform of feature vectors
- More: later in the lecture
Complete Pipeline for Mel-Frequency Cepstral Coefficients (MFCC)

- **Sampling**
- **Pre-emphasis**
- **Windowing**
- **Fast Fourier Transform**
- **Absolute Value**
- **Mel-scaled Filterbank**
- **log**
- **Discrete Cosine Transform**
- **Dynamic Features (1. and 2. derivative)**
- **Linear Discriminant Analysis**

**Typical values:**
- 16 kHz; 16 Bit quantization
- Window size: 25 ms
- 512 Fourier Coefficients
- 24 filterbank values
- keep only 20 lowest cepstra
- 60 dimensional vector
Alternative Feature Extraction Methods

- **LP-Cepstrum (LP=linear prediction)**
  - Derived from speech coding
  - No longer much in use
- **PLP (=Perceptual linear prediction)**
  - For certain applications popular
  - Claim: mode noise robust than MFCCs
  - Main change: use $|.|^{1/3}$ instead of log in MFCC
Summary

• Classical “plain vanilla” feature extraction: Mel-Frequency Cepstral Coefficients
• Main deficiency: not very noise robust
• Used in
  • Speech Recognition
  • Speaker Recognition
  • Music genre classification
3.2 Feature Extraction from Image Processing
Overview

- Feature types:
  - Color
  - Texture
  - Edge
Physics

- It’s all electromagnetic (EM) radiation
  - Different colors correspond to radiation of different wavelengths
  - Intensity of each wavelength specified by amplitude
- We perceive EM radiation within the 400-700 nm range, a tiny piece of spectrum between infra-red and ultraviolet
Visible Light
Color and Wavelength

Most light we see is not just a single wavelength, but a combination of many wavelengths (see below). This profile is often referred to as a spectrum, or spectral power distribution.
Image Representation (RGB)
Image Representation (Channels)
Image Representation

<table>
<thead>
<tr>
<th>(r,g,b)</th>
<th>C pixels wide</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Color Histogram

Calculate percentage of color present in image

Deficiency: loss of regional information
Localized Features

Do color histogram for any region of the image
Edge Detection: Sobel Operator

\[ G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \]

\[ G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \]

\[ |G| = \sqrt{G_x^2 + G_y^2} \]

\[ \phi = \arctan(G_y / G_x) \]

Apply matrices \( G_x \) and \( G_y \) to any image region.
Texture Image Examples

- From the VisTex Texture Database
Gabor filters

\[ G(\rho, \theta) = K \exp(-j\omega(\theta - \theta_0))\exp(- (\rho - \rho_0)^2 / 2\sigma^2_\rho)\exp(- (\theta - \theta_0)^2 / 2\sigma^2_\theta) \]

Gaussian window modulated with a complex sinusoid

Gabor filters at different scales and spatial frequencies

Top row shows anti-symmetric (or odd) filters, bottom row the symmetric (or even) filters

Visual Cortical cells have band-pass responses very similar to Gabor filters

You do not need to remember this formula
Summary

- Main features for image recognition
  - Color
  - Edges
  - Texture