Audio data analysis

CES Data Scientist

Slim ESSID
Audio Data Analysis and Signal Processing team
slim.essid@telecom-paristech.fr
http://www.telecom-paristech.fr/~essid

Credits
O. GILLET, C. JODER, N. MOREAU, G. RICHARD, F. VALLET, …
About “audio”…

Audio frequency:
the range of audible frequencies (20 to 20,000 Hz)

Threshold of pain

Audible sound

Minimal audition threshold

CC Attribution 2.5 Generic
About “audio”…

Audio content categories

Speech

Music

Environmental
An important distinction: speech vs non-speech

**Speech signals**

“Simple” production model: the source-filter model

**Music & non-speech** (environmental)

No generic production model: “timbre”, “pitch”, “loudness”, ...

---

Image: Edward Flemming, course materials for 24.910 Topics in Linguistic Theory: Laboratory Phonology, Spring 2007. MIT OpenCourseWare (http://ocw.mit.edu/), Massachusetts Institute of Technology. Downloaded on 05 May 2012
About “audio”…

► Different research communities

Music Information Research

- Music classification (genre, mood, ...)
- Transcription
- Rhythm analysis

Speech

- Signal representations
- Audio coding
- Source separation
- Sound synthesis

- Speech recognition
- Speaker recognition
- Speech enhancement

Machine Listening / Computer audition
About “audio”…

Research fields

- Acoustics
- Psychoacoustics
- Signal processing
- Linguistics
- Machine learning
- Statistics
- Psychology
- Musicology
- Knowledge engineering
- Databases

Audio content analysis
About “audio”…

Research fields

- Acoustics
- Psychoacoustics
- Signal processing
- Audio content analysis
- Machine learning
- Statistics
- Linguistics
- Psychology
- Musicology
- Knowledge engineering
- Databases
- Statistics
- Psychology
Why analyse audio data?

- **For archive management, indexing**
  - Broadcast content segmentation and classification: speech/music/jingles…, speakers
  - Music **autotagging**: genre (classical, jazz, rock,…), mood, usage…
  - Search engines

- **For broadcasters**
  - Music/effects/speech excerpt search
  - Playlist generation, Djing
Why analyse audio data?

- For designers and producers
  - Audio sample search
  - Music transcription (beat, rhythm, chords, notes)
  - Broadcast content monitoring, plagiarism detection, *hit prediction*

- For end-users
  - Content-based search (shazam++)
  - Non-linear and interactive content consuming (“skip intro”, “replay the chorus”, Karaoke: “remove the vocals”…)
  - Recommendation
  - Personalised playlist generation
Audio-driven multimedia analysis

- **Motivation** for audio-driven content analysis
  - critical information is conveyed by the audio content
  - audio and visual information play complementary roles for the detection of key concepts/events

- Video examples
Audio-driven multimedia analysis

► Video examples

→ Use audio-based laughter detection

• Applause detection
• Cheering detection

• Keyword spotting: “Goal!”
• Sound loudness
• Applause/cheering detection
Audio-driven multimedia analysis

Key audio-based components

- Speech activity detection
- Jingle detection
- Speech/Music/Applause/Laughter/… detection
- Speaker identification
- Speaker diarization: “Who spoke when?”
- Emotion recognition
- Social signal processing: roles, interactions,…
- Speech-to-text transcription
- Music classification genre, mood, …
- Natural language processing

→ At the heart of all components: a **classification task** (supervised or unsupervised)
General classification architecture

Overview

**Development database**

Audio segment

Feature extraction

Classifier training

Feature vectors

\[ x_i = \begin{bmatrix} x_{i,1} \\ \vdots \\ x_{i,d} \end{bmatrix} \]

**TRAINING/DEVELOPMENT PHASE**

(At the lab.)

**Testing database**

**CLASSIFICATION PHASE**

(Exploitation)

Feature extraction

Classification

Decision functions

Segment identified

Audio segment

Feature extraction

\[ x_t \]
Classification architecture

Feature extraction process

Motivation:
- signal denoising/enhancement
- information rate reduction, e.g. subsampling
- normalisation, e.g.:

\[
\tilde{s}(n) = s(n) - \bar{s}, \quad \bar{s} = \frac{1}{L} \sum_{n=0}^{L-1} s(n)
\]

\[
\tilde{s}(n) = \frac{\tilde{s}(n)}{\max_n |\tilde{s}(n)|}
\]

Exercise
In Python:
- load an audio file;
- normalise it;
- visualise it.

Use librosa
Classification architecture

Feature extraction process

- Relies on audio signal processing techniques
Audio signal analysis

► Short-Term analysis windows

Drawing by J. Laroche, modified
Signal framing

» Static temporal segmentation

» Dynamic temporal segmentation
Feature types

- **Temporal features**: extracted directly from the waveform samples

- **Spectral features**: extracted from a frequential representation of the signal

- **Perceptual features**: extracted using a perceptual representation based on *psychoacoustic* considerations
Temporal features - ZCR

Zero Crossing Rates

\[ \frac{1}{2} \sum_{n=2}^{N} |\text{sign}(x_n) - \text{sign}(x_{n-1})| \]

Characterises noisy and transient sections
Spectral analysis

Discrete Fourier Transform

\[ X_k = \sum_{n=0}^{N-1} x_n \exp(-j2\pi \frac{k}{N}), \]

\[ x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \exp\left(j2\pi \frac{k}{N}\right) \]

In practice: computed using the Fast Fourier Transform (FFT)
Discrete Fourier Transform (DFT)

**Important properties**

- Being a **discrete time** Fourier Transform, the DFT is **periodic**, with period 1 (in reduced frequency $f = \frac{f}{f_s}$; $f_s$ : sampling frequency)
- For signals $x(n)$ and $y(n)$; $n \in \{0, ..., N - 1\}$

<table>
<thead>
<tr>
<th>Property</th>
<th>Numerical series</th>
<th>DFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linearity</td>
<td>${ax(n) + by(n)}$</td>
<td>${aX(k) + bY(k)}$</td>
</tr>
<tr>
<td>Hermitian symmetry</td>
<td>$x(n)$ real</td>
<td>$X(k) = X^*(-k)$</td>
</tr>
<tr>
<td>Time translation</td>
<td>$x(n - n_0)$</td>
<td>$X(k)e^{\frac{2j\pi k}{N}n_0}$</td>
</tr>
<tr>
<td>Convolution</td>
<td>$x(n) \ast y(n)$</td>
<td>$X(k)Y(k)$</td>
</tr>
<tr>
<td></td>
<td>$\triangle \sum_k x(k)y(n - k)$</td>
<td></td>
</tr>
<tr>
<td>Conjugation</td>
<td>${x^*(n)}$</td>
<td>${X^*(-k)}$</td>
</tr>
</tbody>
</table>
Spectral analysis

Spectral analysis by Short-Term Fourier Transform (STFT)

Drawing by J. Laroche
Spectral analysis

► Violin excerpt: 20-ms overlapping windows \( (s_r = 44.1\text{kHz} ; N = 882\text{ samples}) \)
Spectral analysis

► Spectrogram

C note (262 Hz) produced by a piano and a violin

Spectral analysis

Exercise

In Python:

- Compute short-term spectra of an audio signal using FTT
- Compute and display spectrogram

- Use
  - `scipy.fftpack`
  - `librosa`
Spectral analysis

- Limitations of the spectrogram representation
  - Large representation
    - Typically 512 coefs every 10 ms
    - High dimensionality
  - Much detail
    - Redundant representation
    - High-level features (pitch, vibrato, timbre) are not highlighted

→ Still a low-level representation, not yet a model
The source-filter model

- Distinction between:
  - **source**: excitation $\rightarrow$ fine spectral structure
  - **filter**: resonator $\rightarrow$ coarse structure

![Diagram]

- Source signal $(Vocal folds)$
- Resonator $(Vocal tract)$
- Speech

$X(f)$ $\times$ $H(f)$ $=$ $Y(f)$
Cepstrum

**Principle**

- **Source-filter model:**
  \[ y(n) = x(n) \ast h(n) \]

- **In the frequency domain:**
  \[ Y(f) = X(f)H(f) \]
  \[ \log |Y(f)| = \log |X(f)| + \log |H(f)| \]

- **By inverse DFT:**
  \[ c_y(q) = c_x(q) + c_h(q) \]

where \( c_y(q) = \text{iDFT}[\log |Y(f)|] \): real cepstrum definition

→ **deconvolution** is thus achieved: filter is separated from excitation

- **First few cepstral coefficients**
  - low quefrency: “slow iDFT waves”
  - represent the filter → **spectral envelope**

- **Next coefficients represent the source** → **fine spectral structure**
Cepstral representations

- **MFCC: Mel Frequency Cepstral Coefficients**

  - Audio frame
  - DFT
  - Magnitude spectrum
  - Triangular filter bank in Mel scale

  ![Mel scale graph](image-url)
Cepstral representations

► MFCC: Mel Frequency Cepstral Coefficients

Audio frame → DFT → Magnitude spectrum → Triangular filter banc in Mel scale
Cepstral representations

► MFCC: Mel Frequency Cepstral Coefficients

Audio frame

DFT

Magnitude spectrum

Triangular filter banc in Mel scale

Log

DCT

13 first coefs (in general)

Discrete Cosine Transform:
- nice decorrelation properties (like PCA)
- yields diagonal covariance matrices
Cepstral representations

**MFCC: Mel Frequency Cepstral Coefficients**

- Audio frame
- DFT
- Magnitude spectrum
- Triangular filter bank in Mel scale
- Log
- DCT
- 39-coefficient feature vector

First and second derivatives: speed and acceleration

→ Coarse temporal modelling

13 first coeffs (in general)

dt
dt²
About MFCCs

► ... very popular!

- In speech applications:
  » Well justified: source-filter model makes sense
  » Nice properties from a statistical modelling viewpoint: decorrelation
  » Effective: state-of-the-art features for speaker and speech tasks

- In general audio classification:
  » “Source-filter” model does not always hold
  » Still, MFCCs work well in practice! they are the default choice
Exercise

- Use librosa to extract MFCCs from an audio file and visualise them
Other spectral features: spectral moments

- **Ordre 1**: Centre de Gravité Spectral (centroïde spectral)

\[ CGS = \frac{\sum_{k=1}^{N} k \cdot |X_k|}{\sum_{k=1}^{N} |X_k|} \]
- CGS élevé: son brillant
- CGS faible: son chaud, rond

- **Ordre 2**: Rayon de Giration Spectral

\[ RGS = \sqrt{\frac{\sum_{k=1}^{N} (k - CGS)^2 \cdot |X_k|}{\sum_{k=1}^{N} |X_k|}} \]
- RGS faible, le timbre est "compact"

- Ordres 3,4 également utilisés…
Other spectral features

Fréquence de coupure

définition de la fréquence $F_c$ au dessous de laquelle 85% de la distribution spectrale est concentrée.

\[ F_c \sum_{k=1}^{N} |X_k| = 0.85 \times \sum_{k=1}^{N} |X_k| \]

Platitude spectrale

définition par sous-bandes $sb$ (MPEG7 ASF)

\[ ASF(sb) = \frac{\left( \prod_{k \in sb} X_k \right)^{1/K_{sb}}}{\frac{1}{K_{sb}} \sum_{k \in sb} X_k} \]

Spectre plat : $ASF \uparrow$, $0 < ASF < 1$

Flux spectral (variation temporelle du contenu spectral)

\[ Flux = \sum_{k=1}^{N} \left( |X_k(m)| - |X_k(m-1)| \right)^2 \]
Classification architecture

Feature extraction process

- Preprocessing
- Feature extraction
- Temporal integration

- Cepstral: MFCC, ...
- Spectral
- Temporal
- Perceptual

Which features to use?
Which features to use for a given task?

- Use intuition/expert knowledge
- Use automatic **feature selection** algorithms
- Alternatively, use **feature learning**
Classification architecture

Feature extraction process

- Preprocessing
- Feature computation
- Temporal integration

Feature vectors:

\[ x[k_T] \ldots x[k_T + L - 1] \]

Integration:

\[ x_T = g\{x[k_T], \ldots, x[k_T + L - 1]\} \]

eg.

\[ x_T = \text{mean}\{x[k_T], \ldots, x[k_T + L - 1]\} \]
Temporal integration

► At the feature level

- smoothing to improve robustness
- synchronise features extracted from different temporal horizons
- capture temporal evolution of features
Classification architecture

Classifier training

Training database

Audio segment

Feature extraction

Classifiers

Training data: assembled from all available audio instances

\[
X = \begin{pmatrix}
    x_1^T \\
    \vdots \\
    x_i^T \\
    \vdots \\
    x_l^T
\end{pmatrix}
= \begin{pmatrix}
    x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,d} \\
    \vdots & \vdots & \vdots & \vdots & \vdots \\
    x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,d} \\
    \vdots & \vdots & \vdots & \vdots & \vdots \\
    x_{l,1} & \cdots & x_{l,j} & \cdots & x_{l,d}
\end{pmatrix},
\]

\[
y = \begin{pmatrix}
    y_1 \\
    \vdots \\
    y_i \\
    \vdots \\
    y_l
\end{pmatrix}
\]

Training examples

Class labels

Unknown in non-supervised problems
References

**Books**

- (Kompatsiaris et al., 2012) *TV Content Analysis: Techniques and Applications (Multimedia Computing, Communication and Intelligence)*, Yiannis Kompatsiaris (Editor), Bernard Merialdo (Editor), Shiguo Lian (Editor). Taylor & Francis, 2012.

**Articles and others**


Software: HTK, Torch, YAAFE, MARSYAS, Sonic Annotator, MIR toolbox, .openSMILE, ...