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

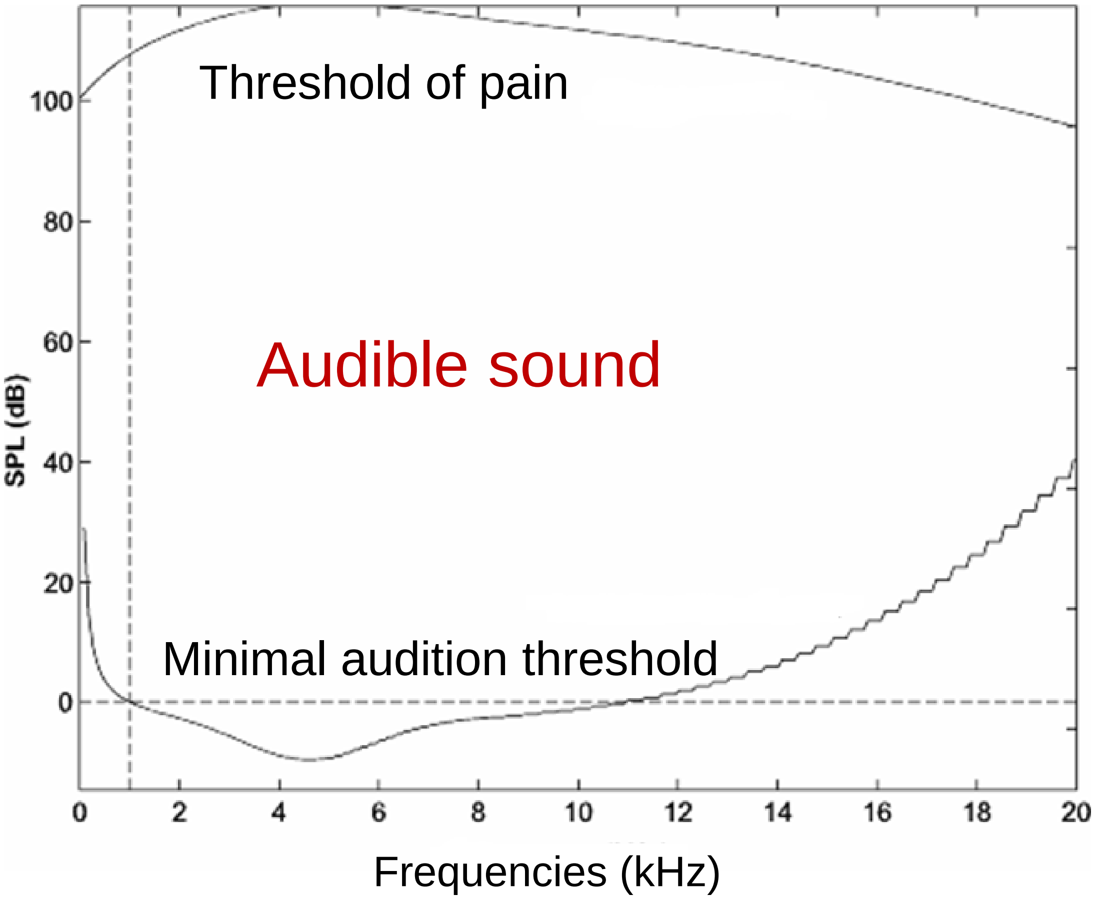
Credits

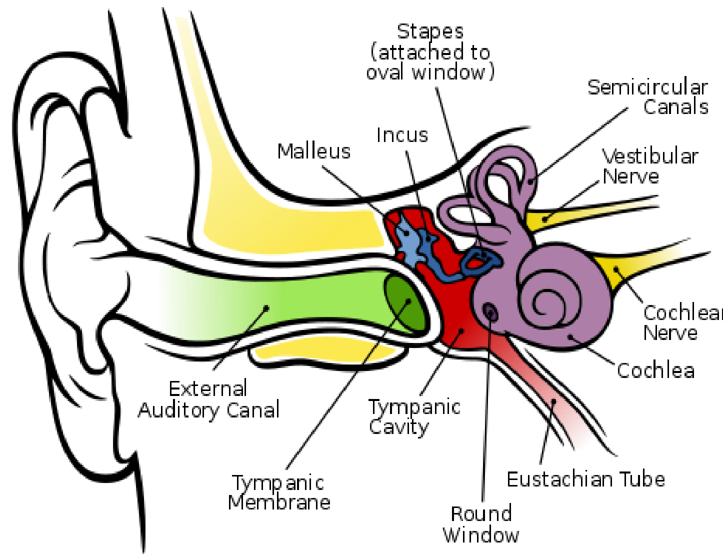
O. GILLET, C. JODER, N. MOREAU, G. RICHARD, F. VALLET, …

About “audio”…

▶ Audio frequency:

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

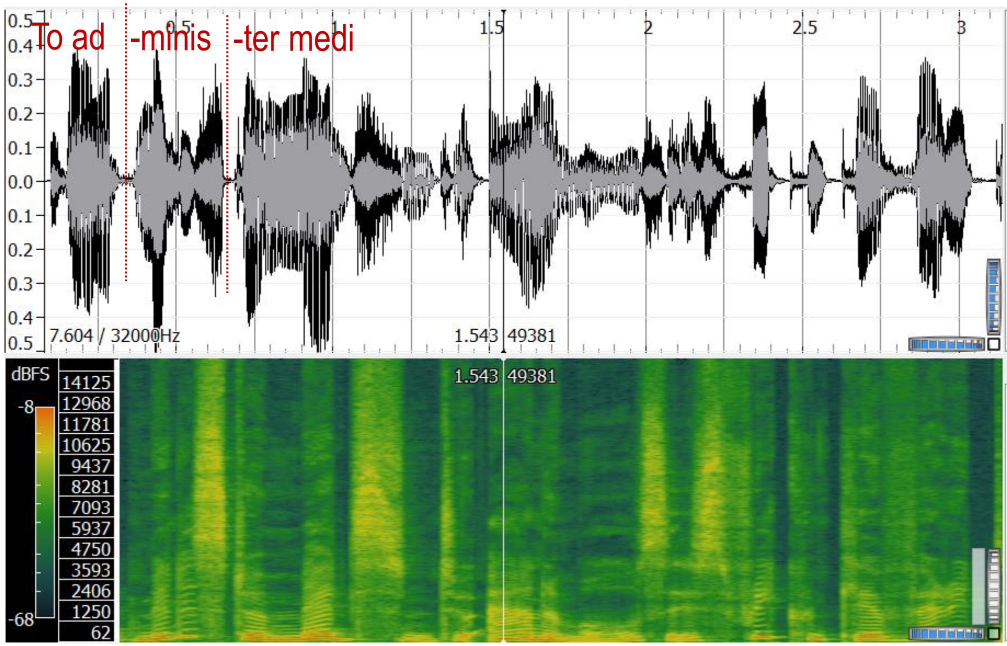




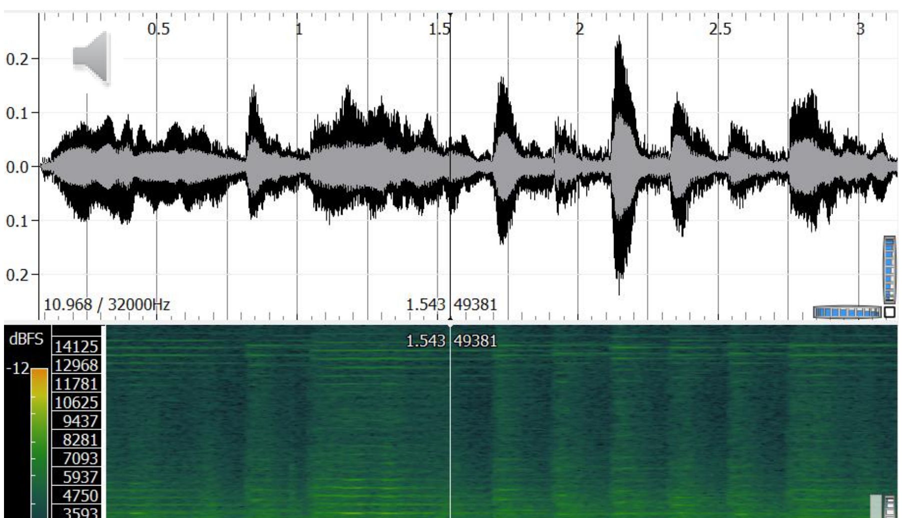
About “audio”…

▶ Audio content categories

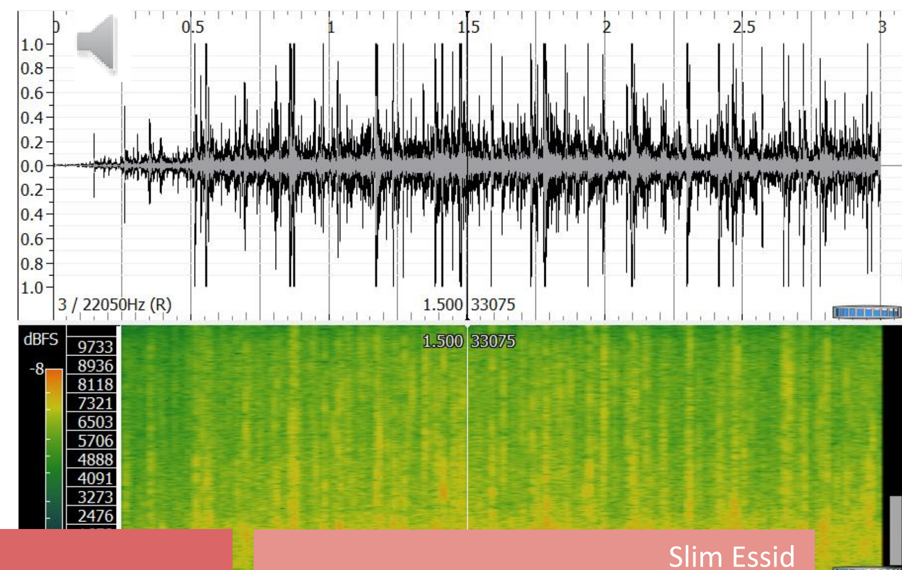
Speech



Music



Environmental

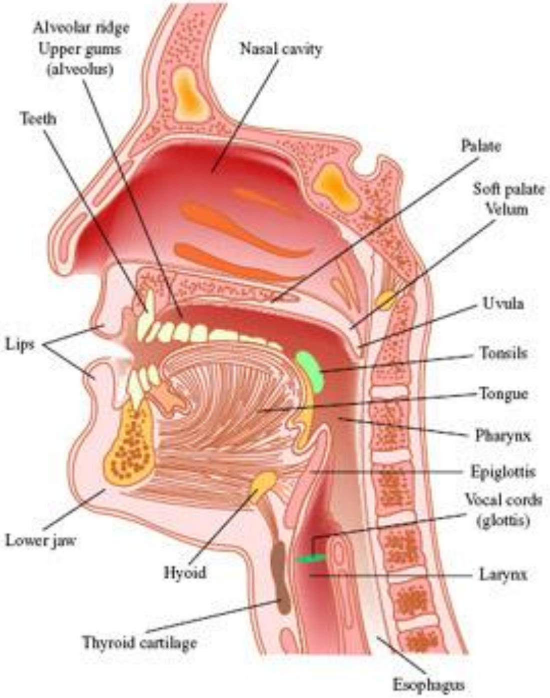


About “audio”…

▶ 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
- …

Signal representations

Audio coding

Source separation

Sound synthesis

…

Speech

- Speech recognition
- Speaker recognition
- Speech enhancement
- …

Machine Listening / Computer audition

About "audio"…

► Research fields

- Acoustics
- Linguistics
- Psychology
- Psychoacoustics
- **Audio content analysis**
- Musicology
- Signal processing
- Knowledge engineering
- Machine learning
- Databases
- Statistics

About "audio"…

► Research fields

Acoustics

Linguistics

Psychology

Psychoacoustics

Audio content analysis

Musicology

Signal processing

Knowledge engineering

Machine learning

Databases

Statistics

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

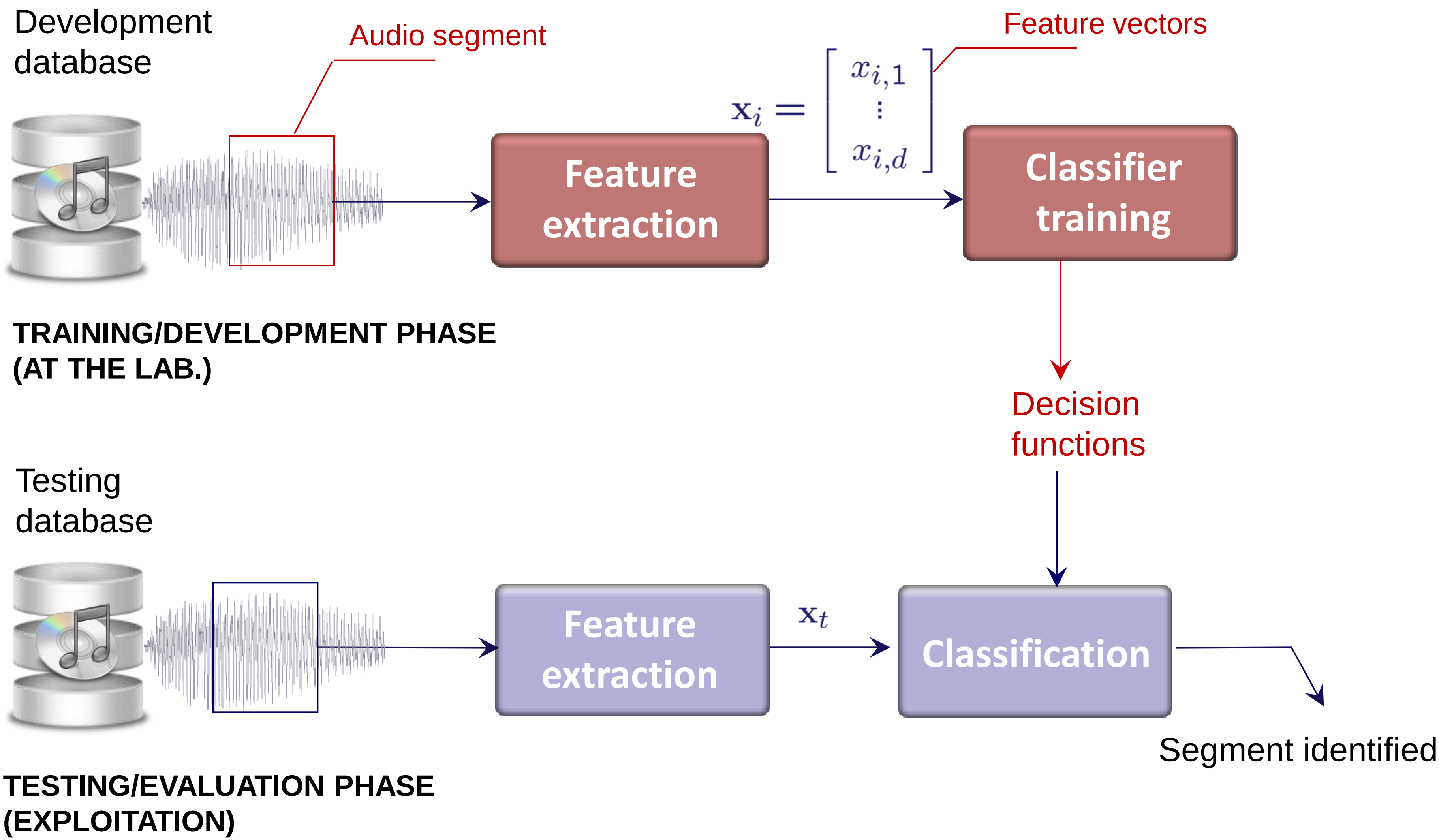
► Key audio-based components

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

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

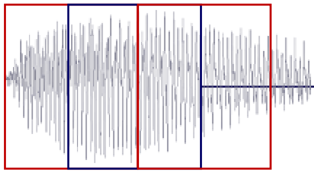
General classification architecture

► Overview



Classification architecture

▶ Feature extraction process



Preprocessing → Feature computation → Temporal integration →

Motivation:

- signal denoising/enhancement
- information rate reduction, eg. subsampling
- normalisation, eg.:

$$\tilde{s}(n) = s(n) - \bar{s}, \; \bar{s} = \frac{1}{L}\sum_{n=0}^{L-1} s(n)$$

$$\hat{s}(n) = \frac{\tilde{s}(n)}{\max_n |\tilde{s}(n)|}$$

Exercise

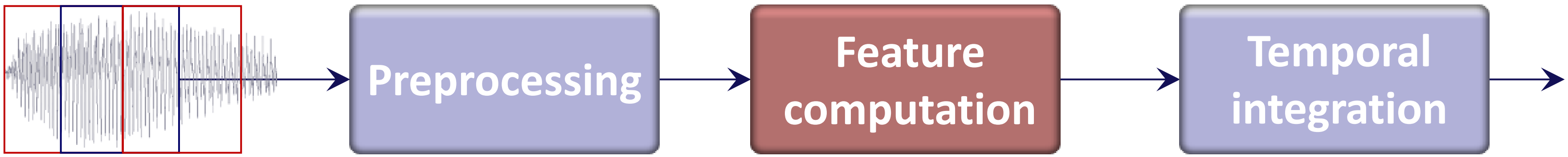
In Python:

- load an audio file;
- normalise it;
- visualise it.

Use librosa

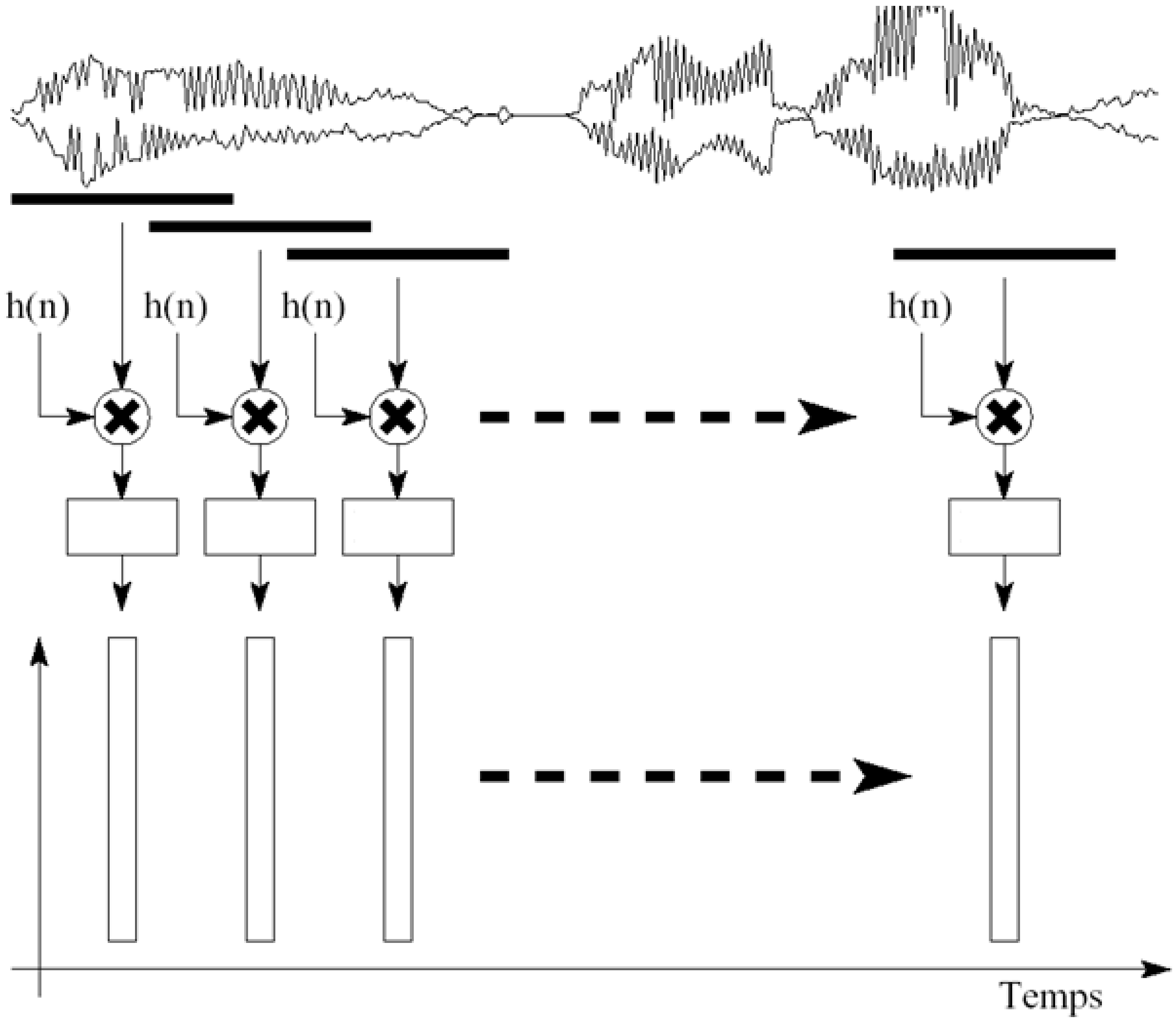
Classification architecture

▶ Feature extraction process



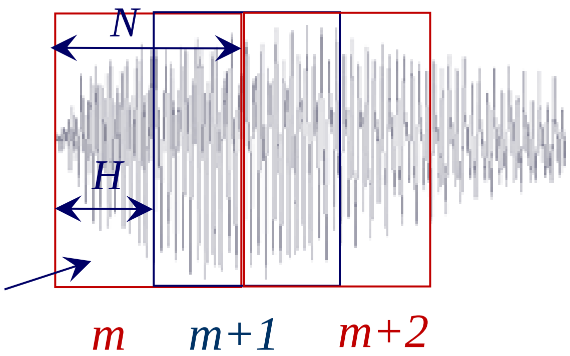
→ Relies on audio signal processing techniques

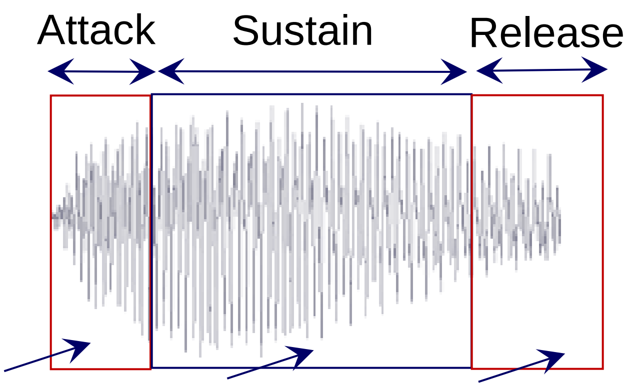
► Short-Term analysis windows



Drawing by J. Laroche, modified

Signal framing





» Static temporal segmentation

» Dynamic temporal segmentation

Feature types

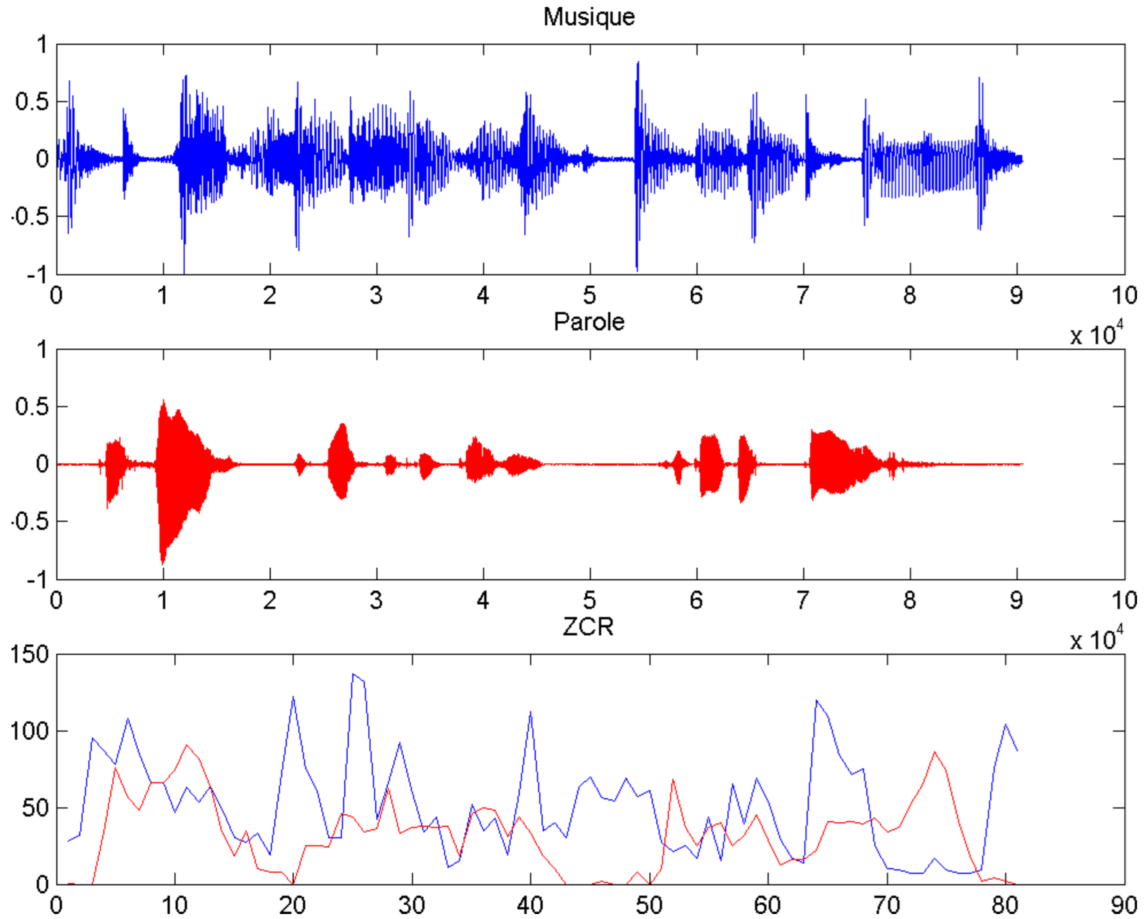
- **Temporal features**: extracted directly from the waveform samples
- **Spectral features:** extracted from a frequential representation of the signal
- **Perceptual features**: extracted using a perceptual representation based on **psychoacoustic** considerations

Temporal features - ZCR

Zero Crossing Rates

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

Characterises noisy and transient sections

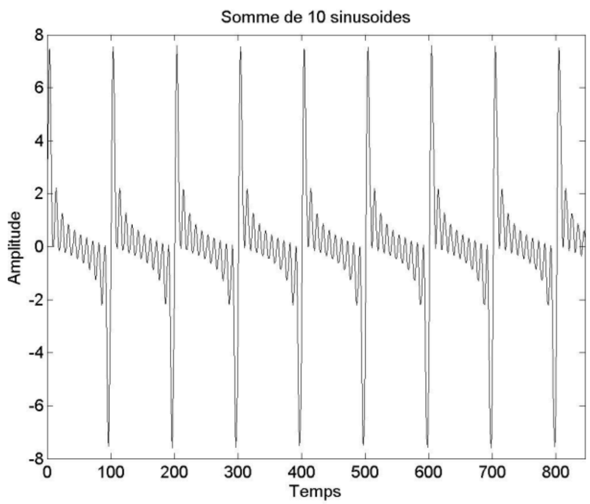


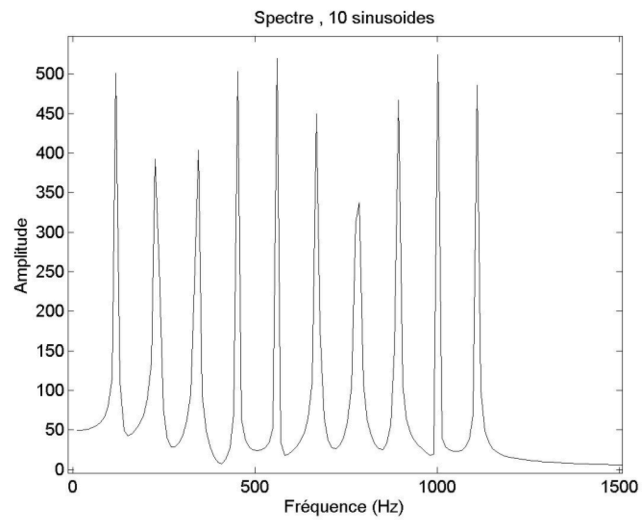
▶ Discrete Fourier Transform

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

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

$$|X_k|$$





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

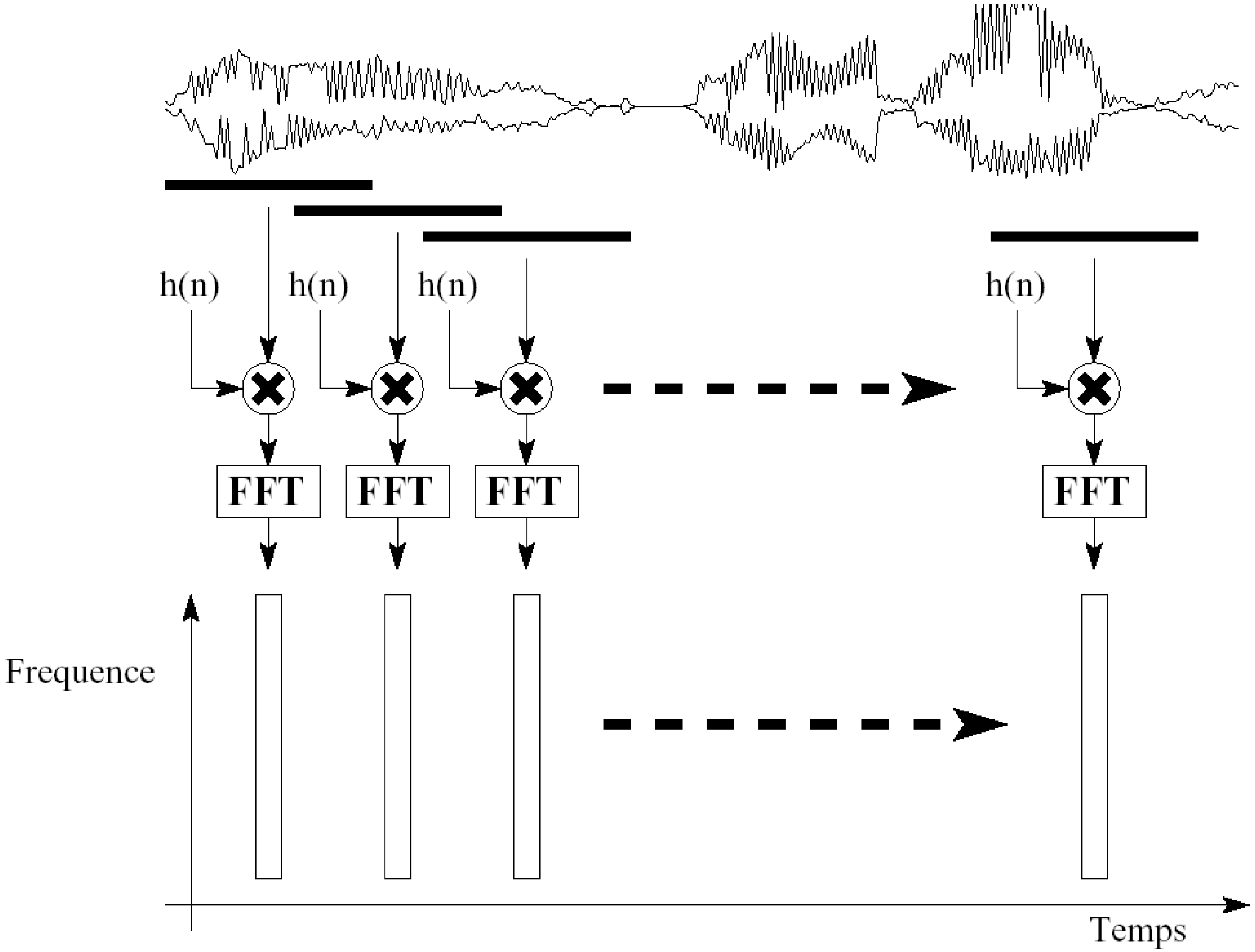
Discrete Fourier Transform (DFT)

▶ 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 $y(n)$; $n \in \{0, \dots, N-1\}$

Property	Numerical series	DFT
Linearity	$\{ax(n) + by(n)\}$	$\{aX(k) + bY(k)\}$
Hermitian symmetry	$x(n)$ **real**	$X(k) = X^*(-k)$
Time translation	$x(n - n_0)$	$X(k)e^{-\frac{2j\pi k}{N}n_0}$
Convolution	$x(n) \star y(n)$ $\triangleq \sum_k x(k)y(n-k)$	$X(k)Y(k)$
Conjugation	$\{x^*(n)\}$	$\{X^*(-k)\}$

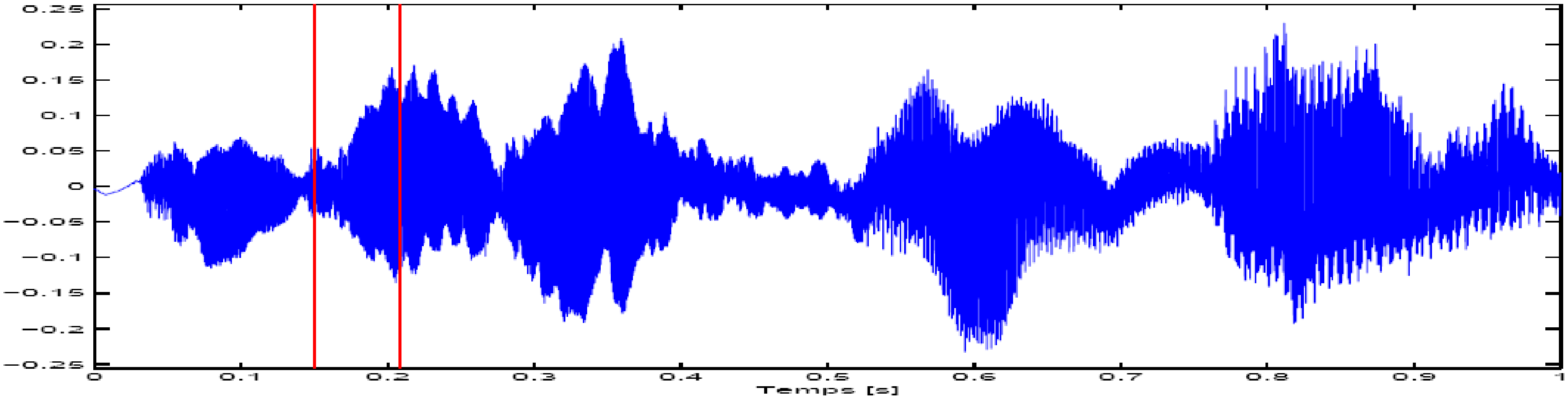
▶ Spectral analysis by Short-Term Fourier Transform (STFT)

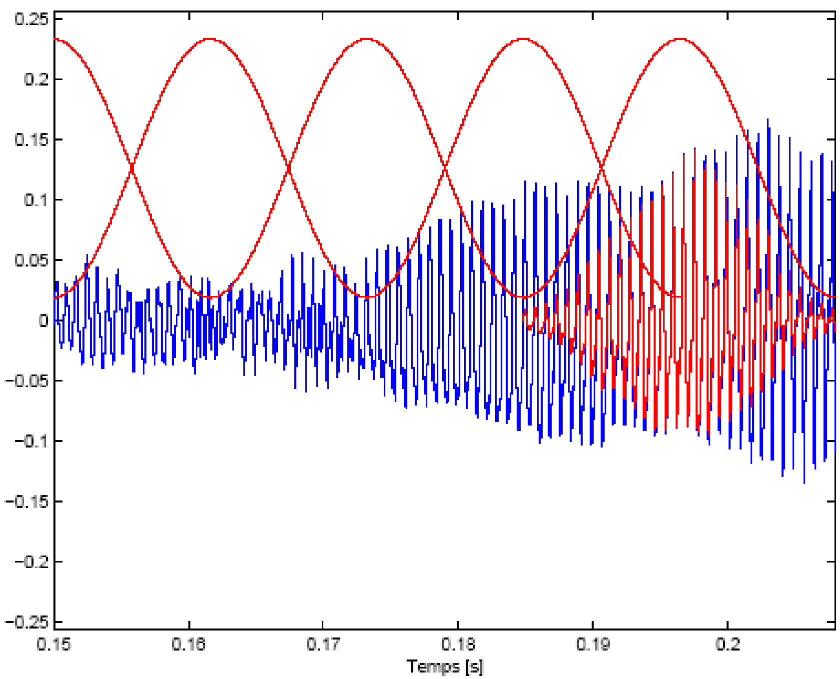


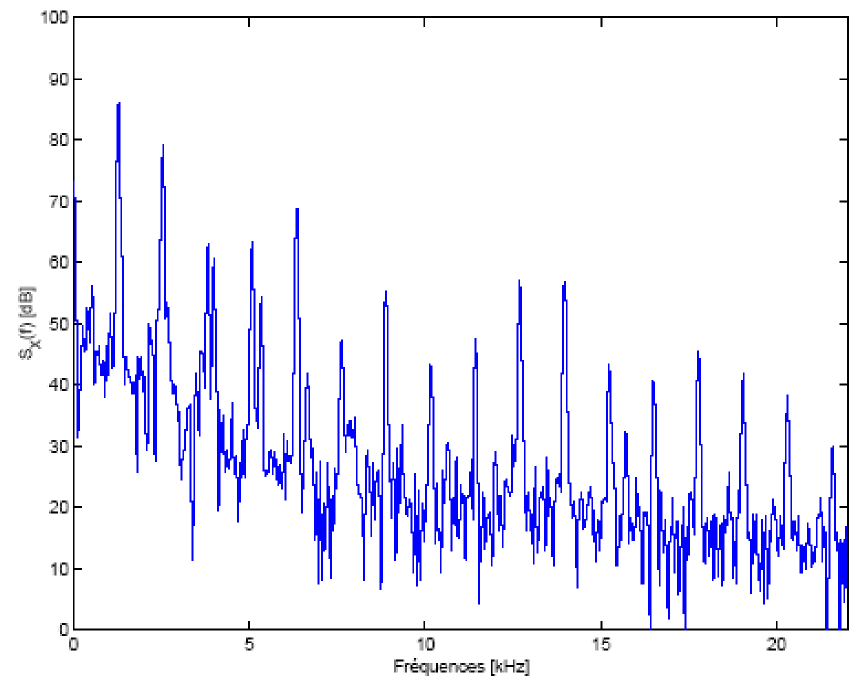
Drawing by J. Laroche

Spectral analysis

▶ Violin excerpt: 20-ms overlapping windows (s_r = 44.1kHz ; N = 882 samples)





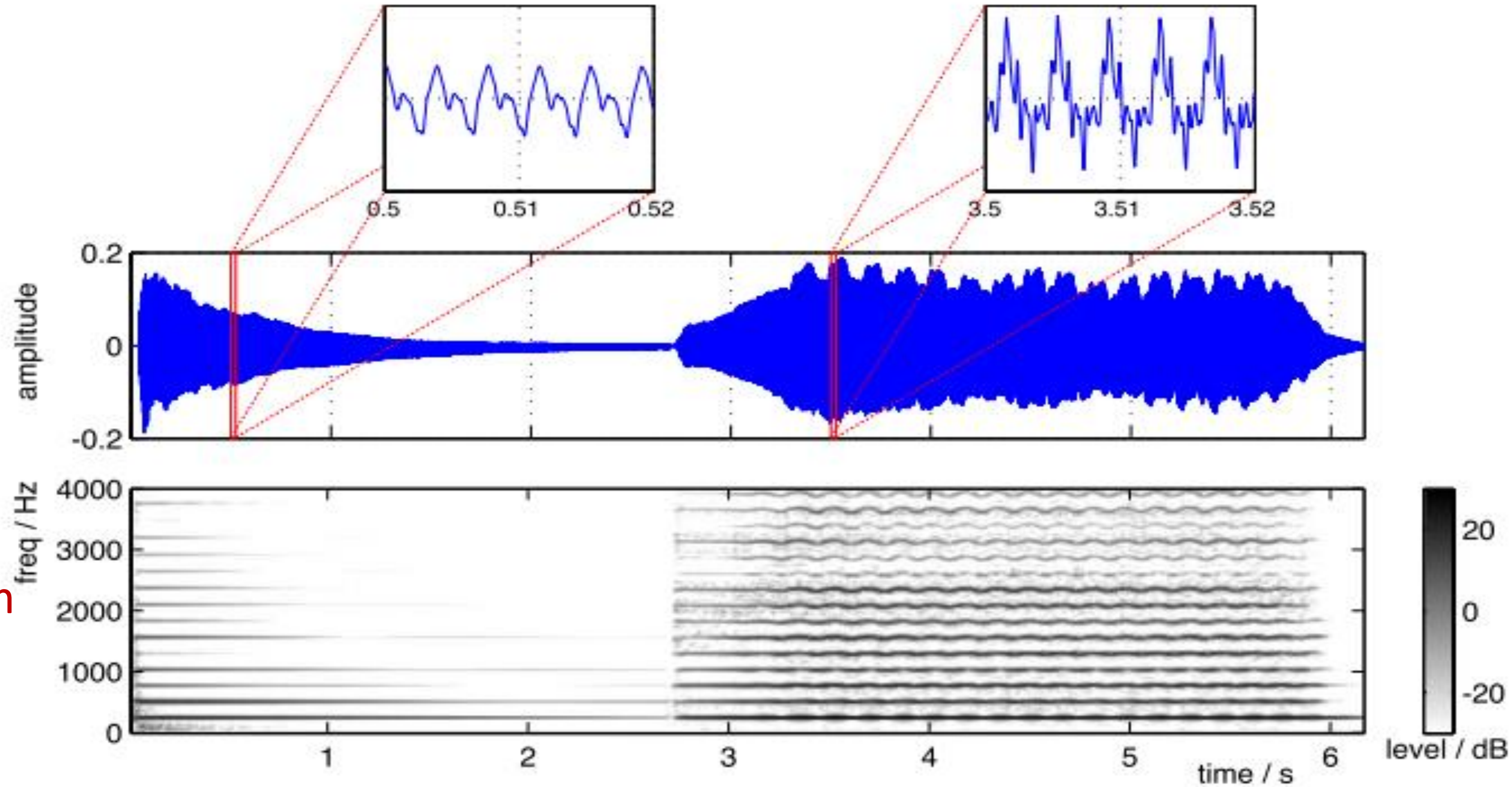


▶ Spectrogram

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

Temporal signal

Spectrogram



From M. Mueller & al. « Signal Processing for Music Analysis, IEEE Trans. On Selected topics in Signal Processing », October 2011.

► Exercise

In Python:

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

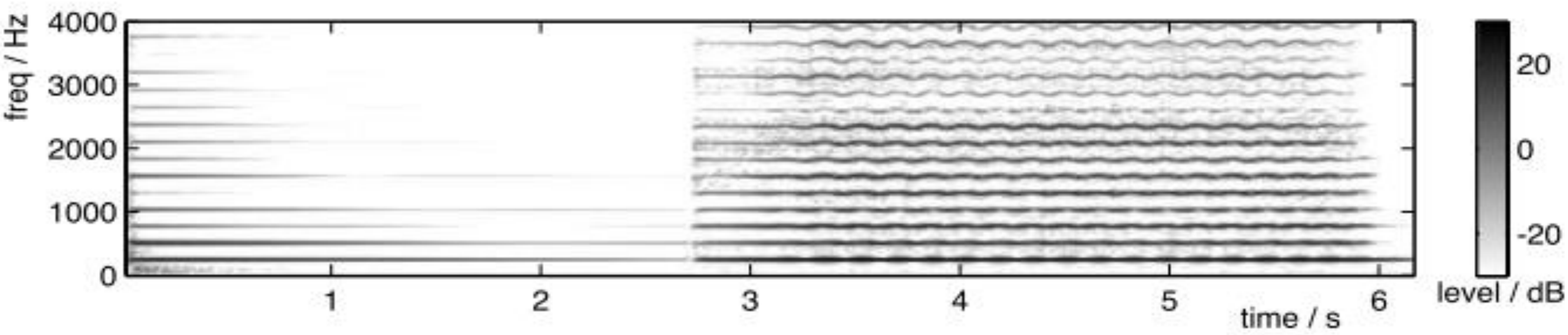
- Use
  - **scipy.fftpack**
  - **librosa**

Spectral analysis

► Limitations of the spectrogram representation

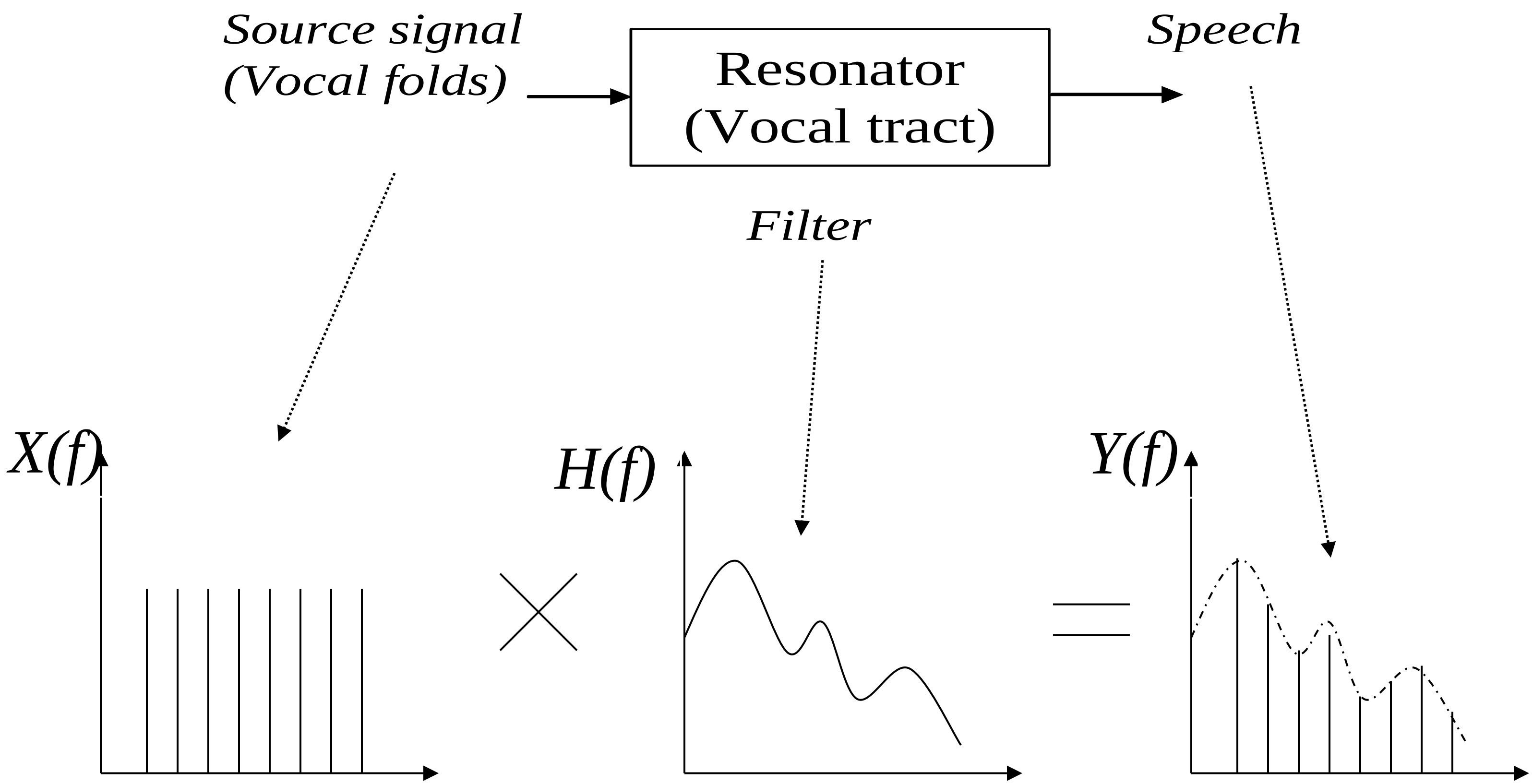
- Large representation
  - » Typically 512 coefs every 10 ms
  - » High dimensionality
- Much detail
  - » Redundant representation
  - » High-level features (pitch, vibrato, timbre) are not highlighted

→ Still a low-level representation, not yet a model



The source-filter model

- Distinction between:
  - » source: excitation → fine spectral structure
  - » filter: resonator → coarse structure



▶ Principle

- Source-filter model: $$y(n) = x(n) * h(n)$$
- In the frequency domain: $$Y(f) = X(f)H(f)$$

$$\log|Y(f)| = \log|X(f)| + \log|H(f)|$$

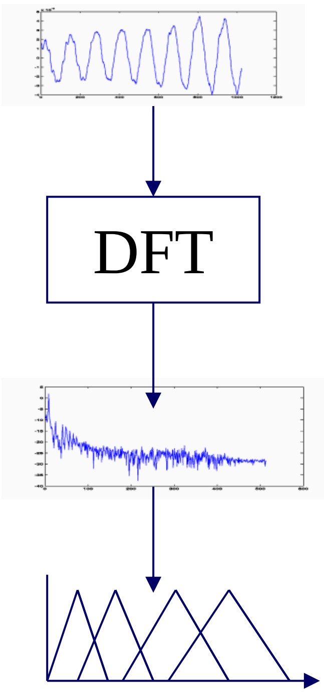
- By inverse DFT: $$c_y(q) = c_x(q) + c_h(q)$$

where $c_y(q) = \mathrm{iDFT}[\log|Y(f)|]$: 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

▶ MFCC: Mel Frequency Cepstral Coefficients

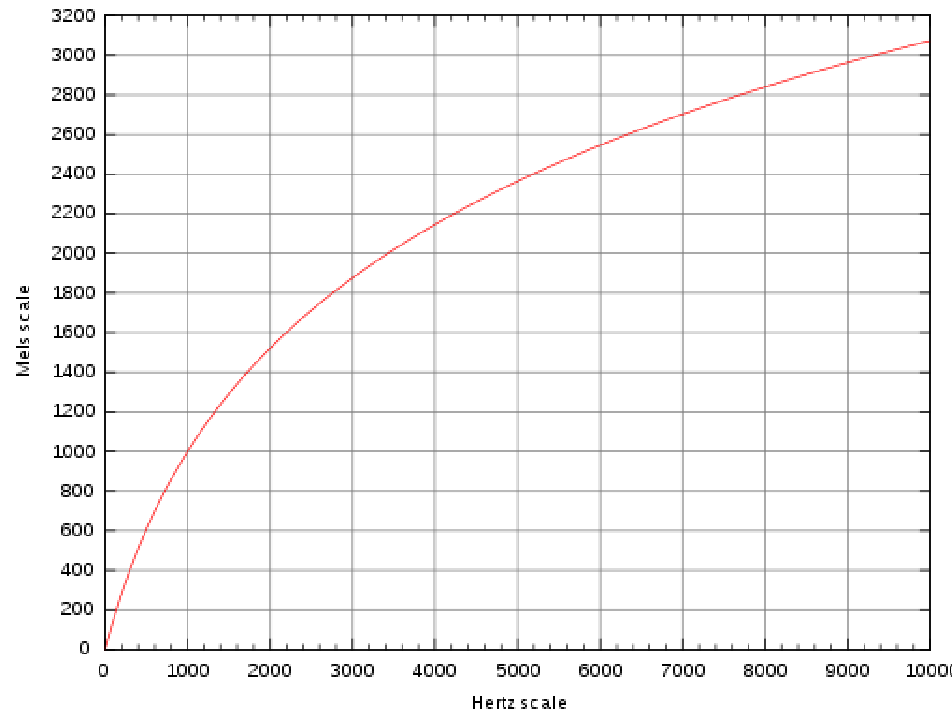


Audio frame

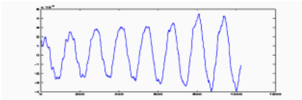
Magnitude spectrum

Triangular filter banc in Mel scale

Mel scale

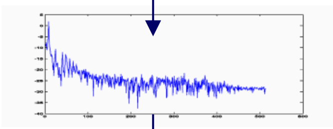


► MFCC: Mel Frequency Cepstral Coefficients

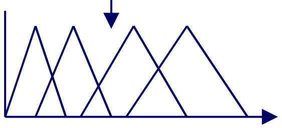


Audio frame

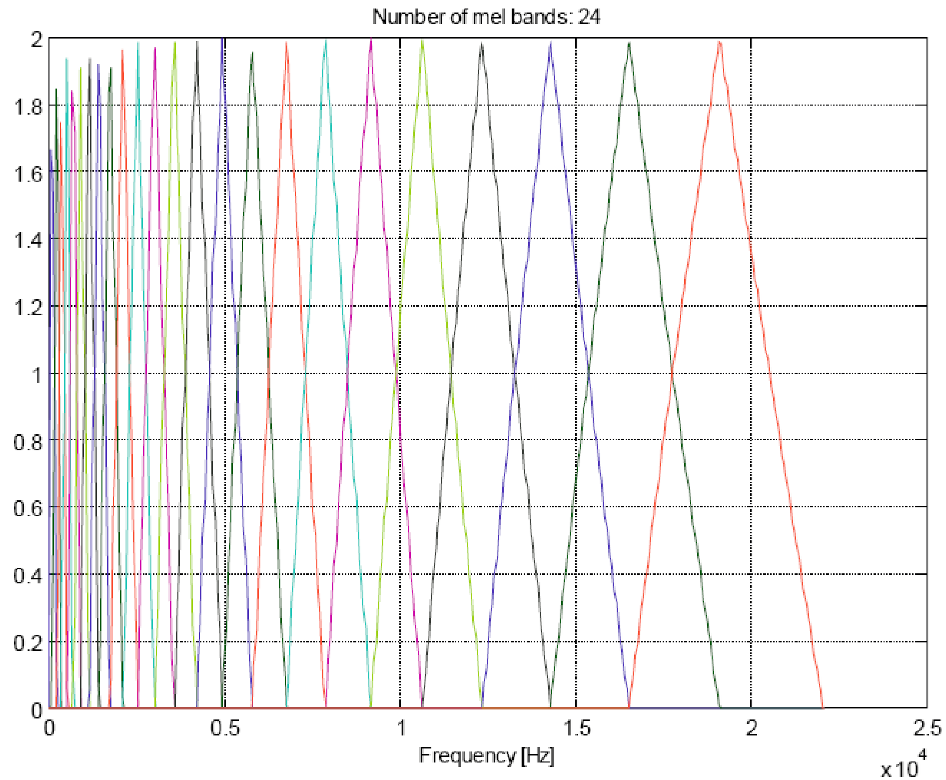
DFT



Magnitude spectrum

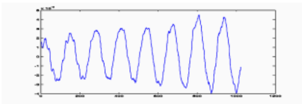


Triangular filter banc
in Mel scale



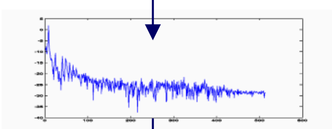
Cepstral representations

► MFCC: Mel Frequency Cepstral Coefficients

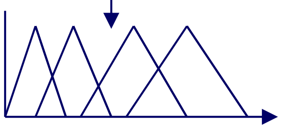


Audio frame

DFT



Magnitude spectrum



Triangular filter banc
in Mel scale

Log → DCT

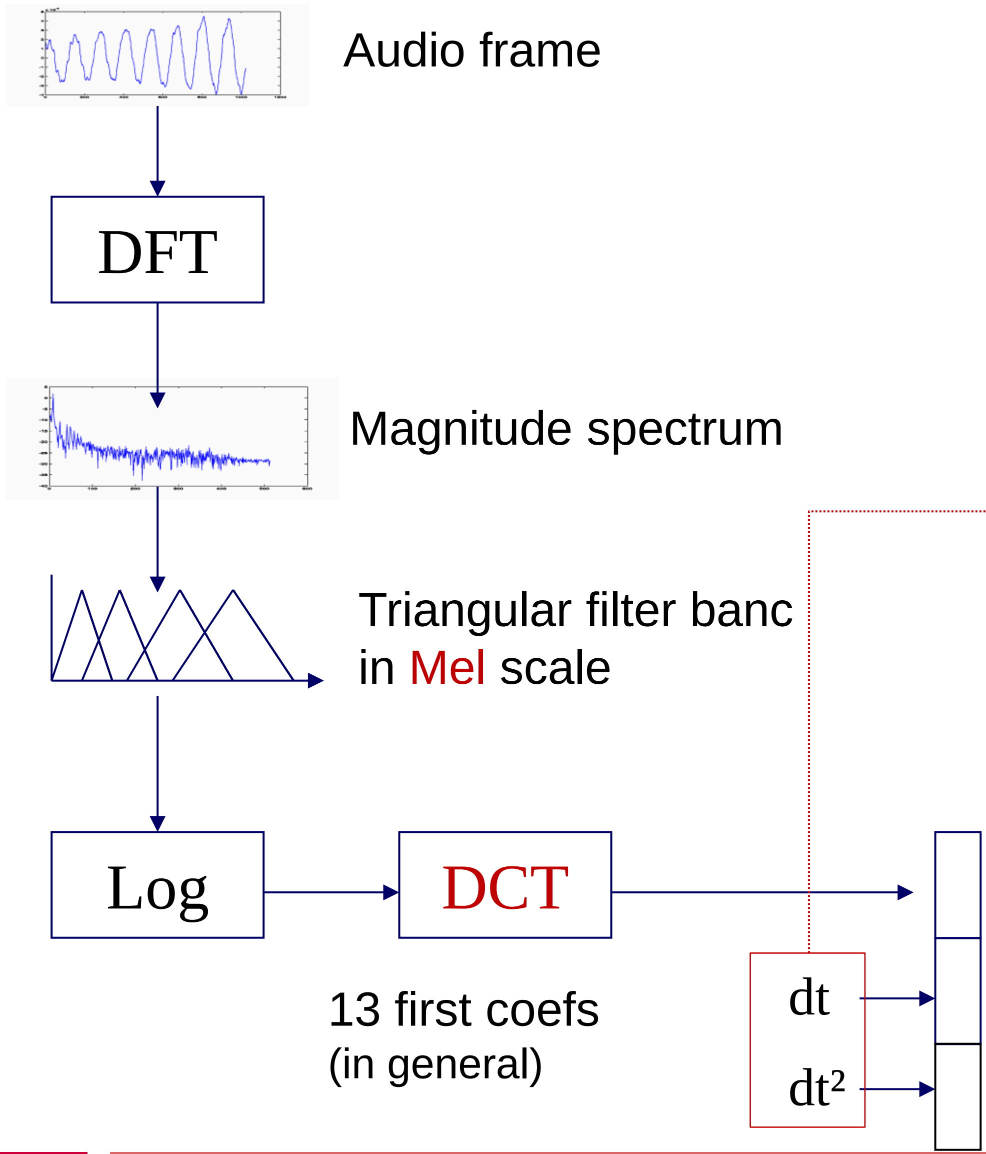
13 first coefs
(in general)

Discrete Cosine Transform:

- nice decorrelation properties (like PCA)
- yields diagonal covariance matrices

Cepstral representations

► MFCC: Mel Frequency Cepstral Coefficients



First and second derivatives:
speed and acceleration

→ Coarse temporal modelling

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

MFCC

▶ 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)

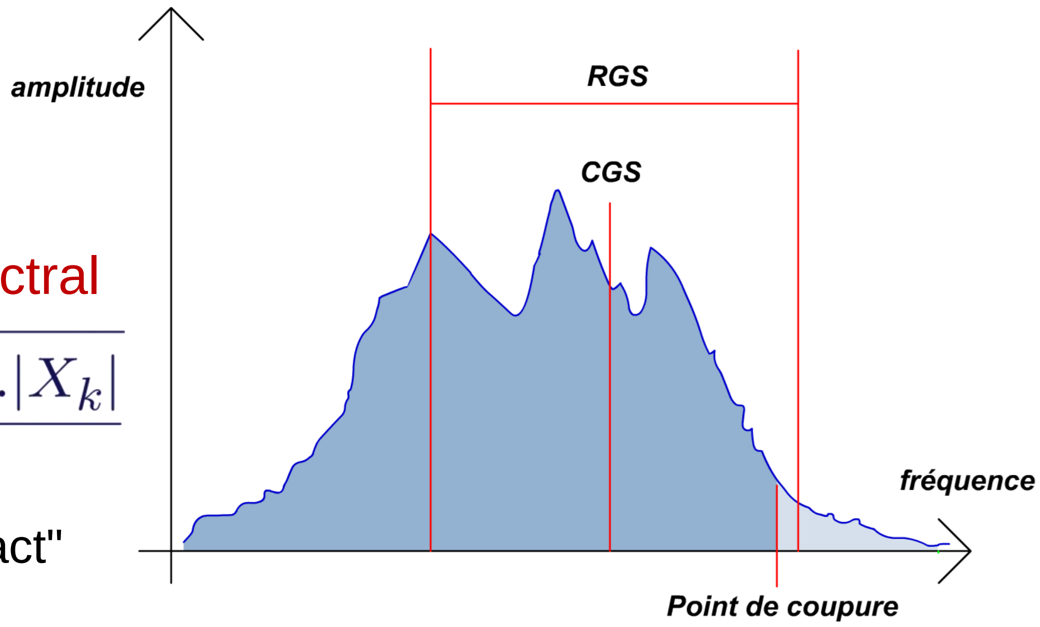
$$CGS = \frac{\sum_{k=1}^{N} k.|X_k|}{\sum_{k=1}^{N} |X_k|}$$

  - CGS élevé: son brillant
  - CGS faible: son chaud, rond

- Ordre 2 : Rayon de Giration Spectral

$$RGS = \sqrt{\frac{\sum_{k=1}^{N} (k - CGS)^2.|X_k|}{\sum_{k=1}^{N} |X_k|}}$$

  - RGS faible, le timbre est "compact"

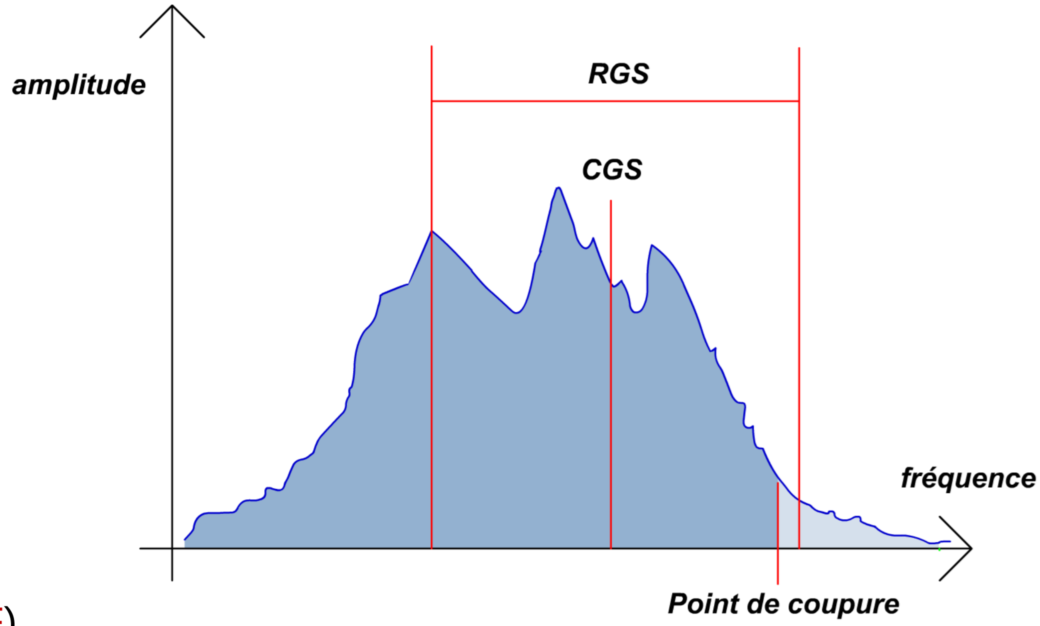


- Ordres 3,4 également utilisés…

Fréquence de coupure

fréquence Fc au dessous de laquelle 85% de la distribution spectrale est concentrée

$$\sum_{k=1}^{Fc} |X_k| = 0.85 \times \sum_{k=1}^{N} |X_k|$$



Platitude spectrale

mesurée par sous-bandes sb (MPEG7 ASF)

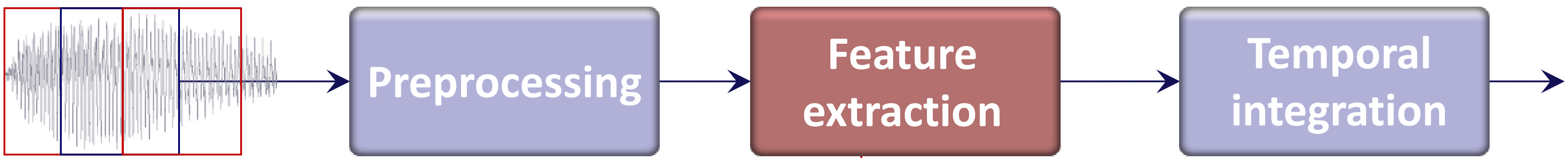
$$ASF(sb) = \frac{\left(\prod_{k \in sb} X_k\right)^{\frac{1}{K_{sb}}}}{\frac{1}{K_{sb}} \sum_{k \in sb} X_k}$$

Spectre plat : ASF ↗ , 0 < ASF < 1

Flux spectral (variation temporelle du contenu spectral)

$$Flux = \sum_{k=1}^{N} \left(|X_k(m)| - |X_k(m-1)|\right)^2$$

▶ Feature extraction process

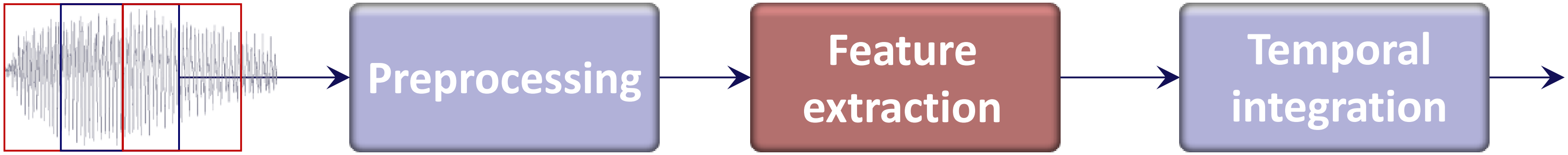


- Cepstral: MFCC, …
- Spectral
- Temporal
- Perceptual

Which features to use?

Classification architecture

► Feature extraction process

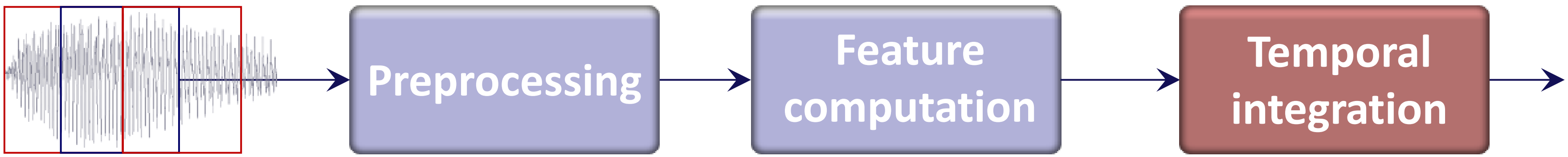


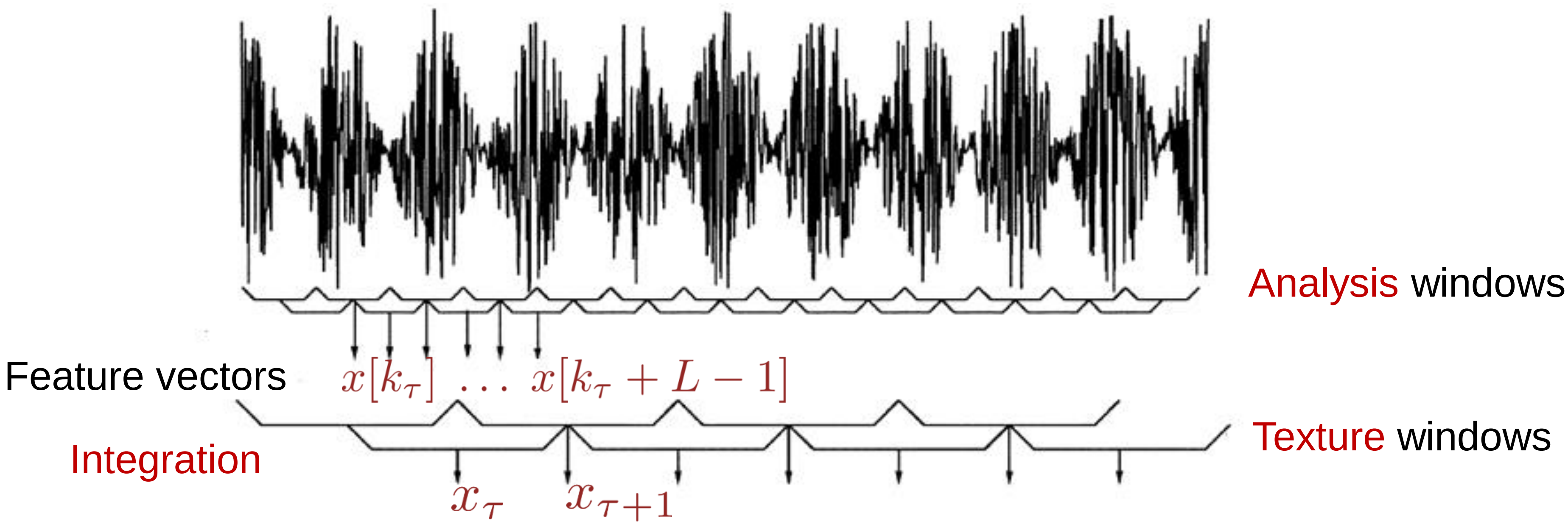
Which features to use for a given task?

- Use intuition/expert knowledge
- Use automatic **feature selection** algorithms

- Alternatively, use **feature learning**

► Feature extraction process



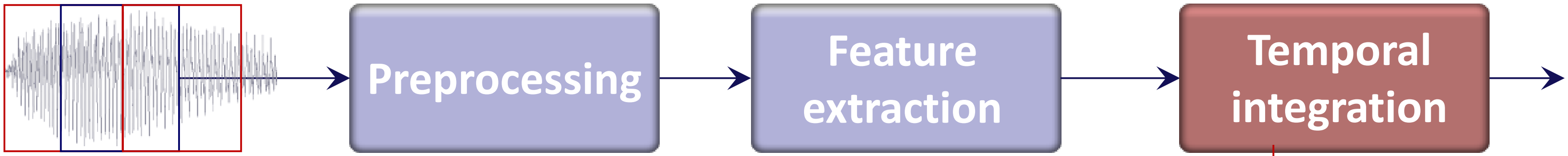


$$x_\tau = g\{x[k_\tau], \ldots, x[k_\tau + L - 1]\}$$

eg. $$x_\tau = \mathrm{mean}\{x[k_\tau], \ldots, x[k_\tau + L - 1]\}$$

Temporal integration

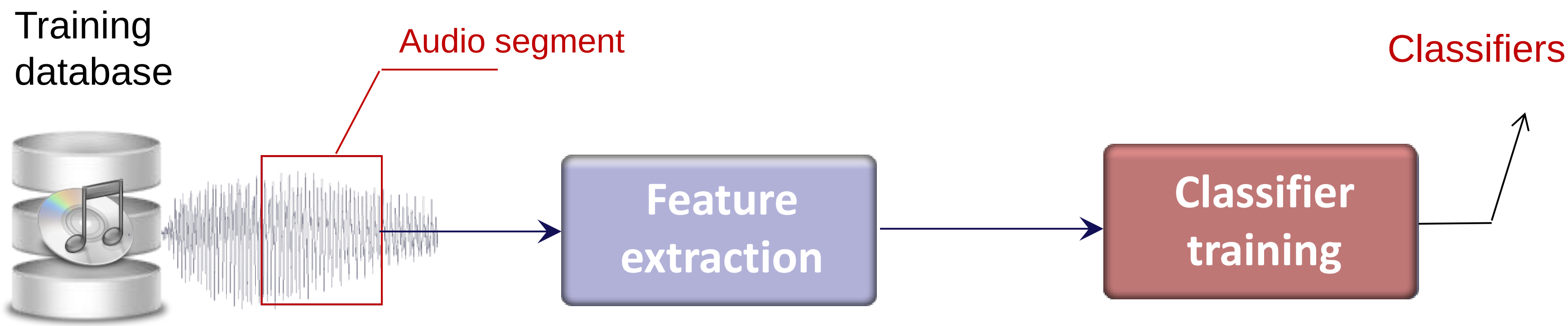
▶ At the feature level



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

Classification architecture

► Classifier training



Training data: assembled from all available audio instances

$$\boldsymbol{X} = \begin{pmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_i^T \\ \vdots \\ \mathbf{x}_l^T \end{pmatrix} = \begin{pmatrix} x_{1,1} & \dots & x_{1,j} & \dots & x_{1,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{i,1} & \dots & x_{i,j} & \dots & x_{i,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{l,1} & \dots & x_{l,j} & \dots & x_{l,d} \end{pmatrix}, \quad \boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_l \end{pmatrix}$$

Training examples

Class labels

Unknown in non-supervised problems

References

Books

- (Rabiner, 93) L. R. Rabiner, *Fundamentals of Speech Processing*. PTR Prentice-Hall, Inc., 1993.
- (Ben Gold et al., 2011) B. Gold, N. Morgan, and D. Ellis, *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*. John Wiley & Sons, 2011.
- (Klapuri & Davy, 2010) A. Klapuri and M. Davy, *Signal Processing Methods for Music Transcription*, vol. 1. Springer, 2006.
- (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.
- (Troncy et al. 2011) Raphael Troncy (Editor), Benoit Huet (Co-Editor), Simon Schenk (Co-Editor). *Multimedia Semantics: Metadata, Analysis and Interaction*. John Wiley & Sons, 2011.
- (Bishop, 2006) C. M. Bishop, *Pattern Recognition And Machine Learning*. Springer, 2006.
- (Hastie et al., 2009) T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

Articles and others

- (Anguera et al., 2010) X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, “Speaker diarization: A review of recent research,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 2011.
- (Dempster et al., 1977) A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society*, vol. 39, pp. 1-38, 1977.
- (Peeters, 2003) G. Peeters, “A large set of audio features for sound description (similarity and classification) in the CUIDADO project,” IRCAM, 2004.
- (Reynolds & Rose, 1995) D. A. Reynolds and R. C. Rose, “Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models,” *IEEE Transactions on Speech and Audio Processing*, vol. 3, pp. 72-83, 1995.
- (Tranter et al., 2006) S. E. Tranter and D. A. Reynolds, “An overview of automatic speaker diarization systems,” *IEEE Transactions on Audio Speech and Language Processing*, vol. 14, no. 5, pp. 1557-1565, 2006.
- Software: HTK, Torch, YAAFE, MARSYAS, Sonic Annotator, MIR toolbox, .openSMILE, ...