

Introduction to Audio Content Analysis

Module 3.6: Instantaneous Features

alexander lerch

introduction

overview

corresponding textbook section

Section 3.6

■ lecture content

- introduction to the concept of features
- timbre
- spectral shape instantaneous features

■ learning objectives

- describe the process of feature extraction
- list possible pre-processing option and explain potential use cases
- describe the general impact of spectral shape on timbre perception
- summarize features, describe their computation, and discuss their meaning



introduction

overview

corresponding textbook section

Section 3.6

■ lecture content

- introduction to the concept of features
- timbre
- spectral shape instantaneous features

■ learning objectives

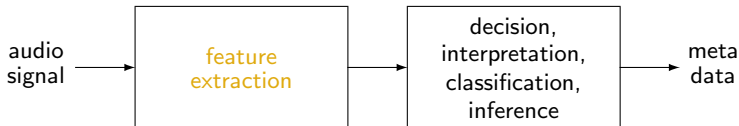
- describe the process of feature extraction
- list possible pre-processing option and explain potential use cases
- describe the general impact of spectral shape on timbre perception
- summarize features, describe their computation, and discuss their meaning



instantaneous features

introduction

remember the flow chart of a general ACA system:



feature:

■ terminology:

- audio descriptor
- instantaneous/short-term/low-level feature

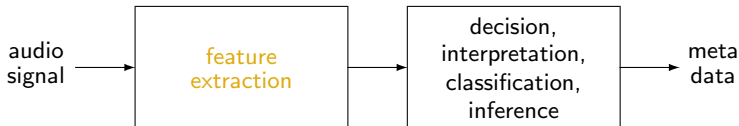
■ characteristics:

- not necessarily musically, perceptually, or semantically meaningful
- low-level: usually one value per block

instantaneous features

introduction

remember the flow chart of a general ACA system:



feature:

■ terminology:

- audio descriptor
- instantaneous/short-term/low-level feature

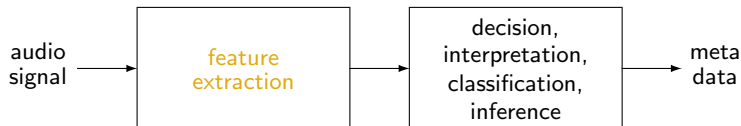
■ characteristics:

- not necessarily musically, perceptually, or semantically meaningful
- low-level: usually one value per block

instantaneous features

introduction

remember the flow chart of a general ACA system:



feature:

■ terminology:

- audio descriptor
- instantaneous/short-term/**low-level feature**

■ characteristics:

- not necessarily musically, perceptually, or semantically meaningful
- low-level: usually one value per block

instantaneous features

feature

a feature ...

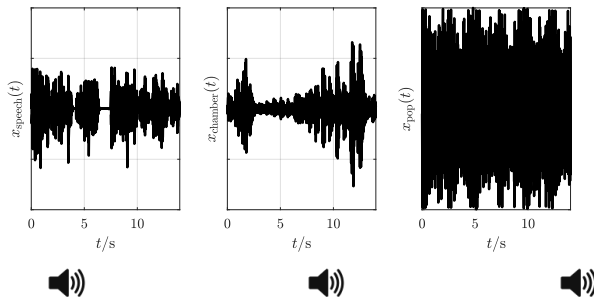
- is task-specific, i.e. holds descriptive power relevant to the task,
- may be custom-designed, chosen from a set of established features, or learned from data,
- can be a representation of any data (audio, meta data, other features, ...),
- is not necessarily musically, perceptually, or semantically meaningful or interpretable
- also: non-redundant, invariant to irrelevancies



instantaneous features

feature example

waveform envelope of three different signals



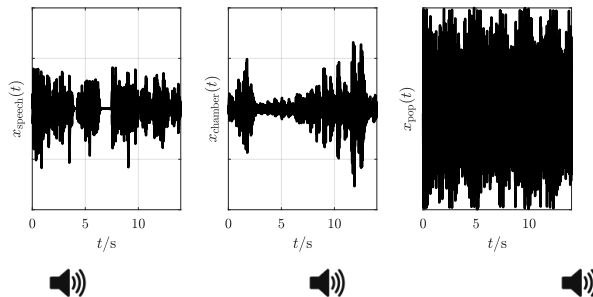
- envelopes of waveforms can have distinct shape
- ⇒ a feature describing envelope shape could help to distinguish these signal types



instantaneous features

feature example

waveform envelope of three different signals

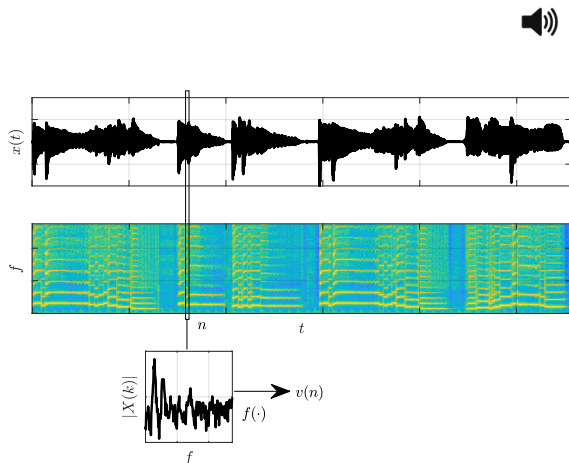


- envelopes of waveforms can have distinct shape
- ⇒ a feature describing envelope shape could help to distinguish these signal types



instantaneous features

feature extraction



- repeat for every block
- repeat for every feature:
Spectral Centroid, RMS, MFCCs, ...

⇒ feature matrix per audio input



definition (American Standards Association)

...that attribute of sensation in terms of which a listener can judge that two sounds having the same loudness and pitch are dissimilar

What is the problem with this definition?



¹ A. S. Bregman, *Auditory Scene Analysis*. MIT Press, 1994.

² S. McAdams and A. Bregman, "Hearing Musical Streams," *Computer Music Journal*, vol. 3, no. 4, pp. 26–60, Dec. 1979, ISSN: 0148-9267.

definition (American Standards Association)

...that attribute of sensation in terms of which a listener can judge that two sounds having the same loudness and pitch are dissimilar

What is the problem with this definition?

Bregman:¹

- 1 implies that timbre *only* exists for sounds with pitch!
- 2 only says that timbre *is not* loudness and pitch



¹ A. S. Bregman, *Auditory Scene Analysis*. MIT Press, 1994.

² S. McAdams and A. Bregman, "Hearing Musical Streams," *Computer Music Journal*, vol. 3, no. 4, pp. 26–60, Dec. 1979, ISSN: 0148-9267.

timbre

introduction 1/2

definition (American Standards Association)

...that attribute of sensation in terms of which a listener can judge that two sounds having the same loudness and pitch are dissimilar

What is the problem with this definition?

Bregman:¹

- 1 implies that timbre *only* exists for sounds with pitch!
- 2 only says that timbre *is not* loudness and pitch

→ [timbre is] " *...the psychoacoustician's multidimensional waste-basket category for everything that cannot be labeled pitch or loudness.*"²



¹ A. S. Bregman, *Auditory Scene Analysis*. MIT Press, 1994.

² S. McAdams and A. Bregman, "Hearing Musical Streams," *Computer Music Journal*, vol. 3, no. 4, pp. 26–60, Dec. 1979, ISSN: 0148-9267.

timbre

introduction 2/2

timbre is

- a function of **temporal envelope**

- attack time characteristics
- amplitude modulations
- ...

- a function of **spectral distribution**

- spectral envelope
- number of partials
- energy distribution of partials
- ...

when dealing with complex mixtures of sound, it is very difficult (maybe impossible?) to extract detailed temporal information for individual tones

⇒ timbre features typically focus on the **spectral shape**

timbre

introduction 2/2

timbre is

- a function of **temporal envelope**
 - attack time characteristics
 - amplitude modulations
 - ...
- a function of **spectral distribution**
 - spectral envelope
 - number of partials
 - energy distribution of partials
 - ...

when dealing with complex mixtures of sound, it is very difficult (maybe impossible?) to extract detailed temporal information for individual tones

⇒ timbre features typically focus on the **spectral shape**

timbre

introduction 2/2

timbre is

- a function of **temporal envelope**
 - attack time characteristics
 - amplitude modulations
 - ...
- a function of **spectral distribution**
 - spectral envelope
 - number of partials
 - energy distribution of partials
 - ...

when dealing with complex mixtures of sound, it is very difficult (maybe impossible?) to extract detailed temporal information for individual tones

⇒ timbre features typically focus on the **spectral shape**

timbre is

- a function of **temporal envelope**
 - attack time characteristics
 - amplitude modulations
 - ...
- a function of **spectral distribution**
 - spectral envelope
 - number of partials
 - energy distribution of partials
 - ...

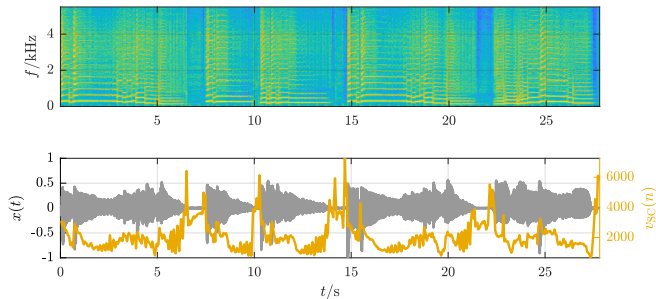
when dealing with complex mixtures of sound, it is very difficult (maybe impossible?) to extract detailed temporal information for individual tones

⇒ timbre features typically focus on the **spectral shape**

spectral shape features

spectral centroid

$$v_{SC}(n) = \frac{\sum_{k=0}^{K/2-1} k \cdot |X(k, n)|}{\sum_{k=0}^{K/2-1} |X(k, n)|}$$



spectral shape features

spectral centroid

$$v_{\text{SC}}(n) = \frac{\sum_{k=0}^{\mathcal{K}/2-1} k \cdot |X(k, n)|}{\sum_{k=0}^{\mathcal{K}/2-1} |X(k, n)|}$$

common variants:

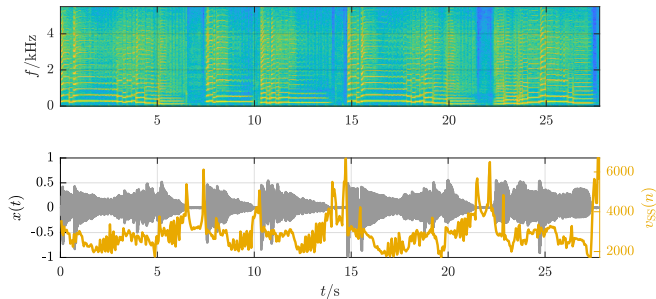
- power spectrum
- logarithmic frequency scale

$$v_{\text{SC},\log}(n) = \frac{\sum_{k=k(f_{\min})}^{\mathcal{K}/2-1} \log_2 \left(\frac{f(k)}{f_{\text{ref}}} \right) \cdot |X(k, n)|^2}{\sum_{k=k(f_{\min})}^{N/2-1} |X(k, n)|^2}$$

spectral shape features

spectral spread

$$v_{SS}(n) = \sqrt{\frac{\sum_{k=0}^{\mathcal{K}/2-1} (k - v_{SC}(n))^2 \cdot |X(k, n)|^2}{\sum_{k=0}^{\mathcal{K}/2-1} |X(k, n)|^2}}$$



spectral shape features

spectral spread

$$v_{SS}(n) = \sqrt{\frac{\sum_{k=0}^{\mathcal{K}/2-1} (k - v_{SC}(n))^2 \cdot |X(k, n)|^2}{\sum_{k=0}^{\mathcal{K}/2-1} |X(k, n)|^2}}$$

common variants:

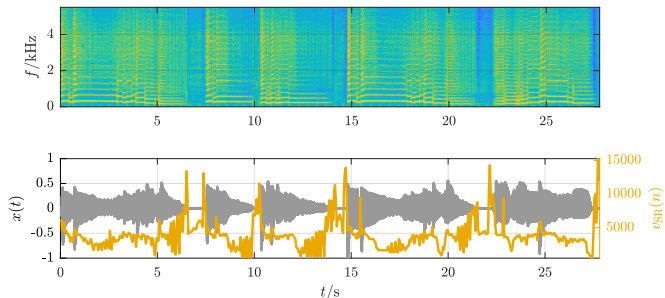
- same variants as with *Spectral Centroid*, e.g. logarithmic:

$$v_{SS, \log}(n) = \sqrt{\frac{\sum_{k=k(f_{\min})}^{\mathcal{K}/2-1} \left(\log_2 \left(\frac{f(k)}{1000 \text{ Hz}} \right) - v_{SC}(n) \right)^2 \cdot |X(k, n)|^2}{\sum_{k=k(f_{\min})}^{\mathcal{K}/2-1} |X(k, n)|^2}}$$

spectral shape features

spectral rolloff

$$v_{\text{SR}}(n) = i \quad \text{at} \quad \sum_{k=0}^i |X(k, n)| = \kappa \cdot \sum_{k=0}^{\kappa/2-1} |X(k, n)|$$



spectral shape features

spectral rolloff

$$v_{\text{SR}}(n) = i \quad \text{at} \quad \sum_{k=0}^i |X(k, n)| = \kappa \cdot \sum_{k=0}^{\mathcal{K}/2-1} |X(k, n)|$$

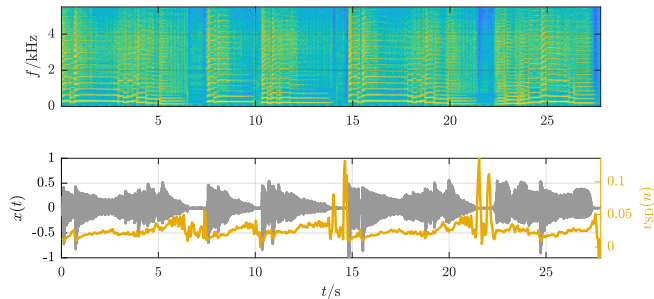
common variants:

- scaled to frequency
- power spectrum

spectral shape features

spectral decrease

$$v_{SD}(n) = \frac{\sum_{k=1}^{\kappa/2-1} \frac{1}{k} \cdot (|X(k, n)| - |X(0, n)|)}{\sum_{k=1}^{\kappa/2-1} |X(k, n)|}$$



spectral shape features

spectral decrease

$$v_{SD}(n) = \frac{\sum_{k=1}^{\kappa/2-1} \frac{1}{k} \cdot (|X(k, n)| - |X(0, n)|)}{\sum_{k=1}^{\kappa/2-1} |X(k, n)|}$$

common variants:

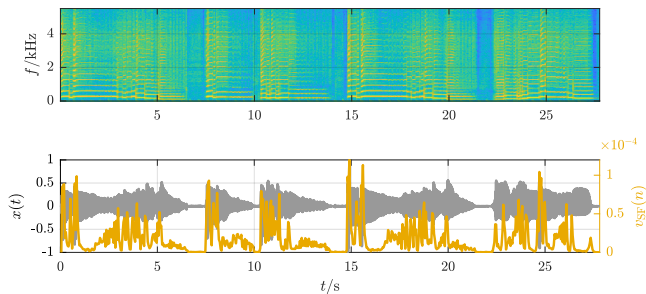
- restricted frequency range:

$$v_{SD}(n) = \frac{\sum_{k=k_l}^{k_u} \frac{1}{k} \cdot (|X(k, n)| - |X(k_l - 1, n)|)}{\sum_{k=k_l}^{k_u} |X(k, n)|}$$

spectral shape features

spectral flux

$$v_{\text{SF}}(n) = \frac{\sqrt{\sum_{k=0}^{\mathcal{K}/2-1} (|X(k, n)| - |X(k, n-1)|)^2}}{\mathcal{K}/2}$$



spectral shape features

spectral flux

$$v_{\text{SF}}(n) = \frac{\sqrt{\sum_{k=0}^{\mathcal{K}/2-1} (|X(k, n)| - |X(k, n-1)|)^2}}{\mathcal{K}/2}$$

common variants:

$$v_{\text{SF}}(n, \beta) = \frac{\sqrt[\beta]{\sum_{k=0}^{\mathcal{K}/2-1} (|X(k, n)| - |X(k, n-1)|)^\beta}}{\mathcal{K}/2}$$

$$v_{\text{SF}, \sigma}(n) = \sqrt{\frac{2}{\mathcal{K}} \sum_{k=0}^{\mathcal{K}/2-1} (\Delta X(k, n) - \mu_{\Delta X})^2}$$

$$v_{\text{SF}, \log}(n) = \frac{2}{\mathcal{K}} \sum_{k=0}^{\mathcal{K}/2-1} \log_2 \left(\frac{|X(k, n)|}{|X(k, n-1)|} \right)$$

fundamentals

cepstrum 1/3

signal model:

convolution of *excitation signal* and *transfer function*

$$x(i) = e(i) * h(i)$$

$$X(j\omega) = E(j\omega) \cdot H(j\omega)$$

$$\begin{aligned}\log(X(j\omega)) &= \log(E(j\omega) \cdot H(j\omega)) \\ &= \log(E(j\omega)) + \log(H(j\omega))\end{aligned}$$

fundamentals

cepstrum 1/3

signal model:

convolution of *excitation signal* and *transfer function*

$$x(i) = e(i) * h(i)$$

$$X(j\omega) = E(j\omega) \cdot H(j\omega)$$

$$\begin{aligned}\log(X(j\omega)) &= \log(E(j\omega) \cdot H(j\omega)) \\ &= \log(E(j\omega)) + \log(H(j\omega))\end{aligned}$$

fundamentals

cepstrum 1/3

signal model:

convolution of *excitation signal* and *transfer function*

$$x(i) = e(i) * h(i)$$

$$X(j\omega) = E(j\omega) \cdot H(j\omega)$$

$$\begin{aligned}\log(X(j\omega)) &= \log(E(j\omega) \cdot H(j\omega)) \\ &= \log(E(j\omega)) + \log(H(j\omega))\end{aligned}$$

fundamentals

cepstrum 2/3

$$\begin{aligned}
 c_x(i) &= \mathfrak{F}^{-1} \{ \log (X(j\omega)) \} \\
 &= \mathfrak{F}^{-1} \{ \log (E(j\omega)) + \log (H(j\omega)) \} \\
 &= \mathfrak{F}^{-1} \{ \log (E(j\omega)) \} + \mathfrak{F}^{-1} \{ \log (H(j\omega)) \}
 \end{aligned}$$

$$\hat{c}_x(i_s(n) \dots i_e(n)) = \sum_{k=0}^{K/2-1} \log (|X(k, n)|) e^{jki\Delta\Omega}$$

fundamentals

cepstrum 2/3

$$\begin{aligned}
 c_x(i) &= \mathfrak{F}^{-1} \{ \log (X(j\omega)) \} \\
 &= \mathfrak{F}^{-1} \{ \log (E(j\omega)) + \log (H(j\omega)) \} \\
 &= \mathfrak{F}^{-1} \{ \log (E(j\omega)) \} + \mathfrak{F}^{-1} \{ \log (H(j\omega)) \}
 \end{aligned}$$

$$\hat{c}_x(i_s(n) \dots i_e(n)) = \sum_{k=0}^{K/2-1} \log (|X(k, n)|) e^{jki\Delta\Omega}$$

fundamentals

cepstrum 2/3

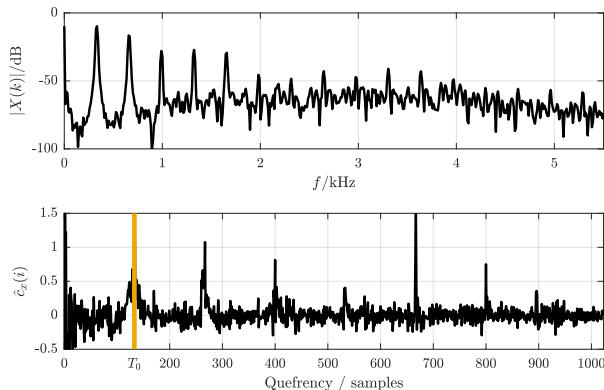
$$\begin{aligned}
 c_x(i) &= \mathfrak{F}^{-1} \{ \log (X(j\omega)) \} \\
 &= \mathfrak{F}^{-1} \{ \log (E(j\omega)) + \log (H(j\omega)) \} \\
 &= \mathfrak{F}^{-1} \{ \log (E(j\omega)) \} + \mathfrak{F}^{-1} \{ \log (H(j\omega)) \}
 \end{aligned}$$

$$\hat{c}_x(i_s(n) \dots i_e(n)) = \sum_{k=0}^{K/2-1} \log (|X(k, n)|) e^{jki\Delta\Omega}$$

fundamentals

cepstrum 2/3

$$\hat{c}_x(i_s(n) \dots i_e(n)) = \sum_{k=0}^{K/2-1} \log(|X(k, n)|) e^{jki\Delta\Omega}$$



fundamentals

cepstrum 3/3

■ summary:

- cepstrum 'replaces' time domain convolution operation with addition
- result is the *unfiltered* excitation signal *plus* the filter IR (both logarithmic)
- can be used for, e.g., *spectral envelope extraction* or *pitch detection*
- more naming silliness:
cepstrum, quefrency, liftering, ...

fundamentals

cepstrum 3/3

■ summary:

- cepstrum 'replaces' time domain convolution operation with addition
- result is the *unfiltered* excitation signal *plus* the filter IR (both logarithmic)
- can be used for, e.g., *spectral envelope extraction* or *pitch detection*
- more naming silliness:
cepstrum, quefrency, liftering, ...

spectral shape features

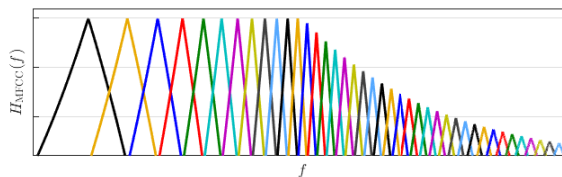
mel frequency cepstral coefficients 1/4

- typical processing steps for the mel frequency cepstral coefficients (MFCCs):
 - 1 compute magnitude spectrum
 - 2 convert linear frequency scale to logarithmic
 - 3 group bins into bands
 - 4 apply logarithm to all bands
 - 5 compute (inverse) cosine transform (DCT)

$$v_{\text{MFCC}}^j(n) = \sum_{k'=1}^{\mathcal{K}'} \log(|X'(k', n)|) \cdot \cos\left(j \cdot \left(k' - \frac{1}{2}\right) \frac{\pi}{\mathcal{K}'}\right)$$

spectral shape features

mel frequency cepstral coefficients 2/4

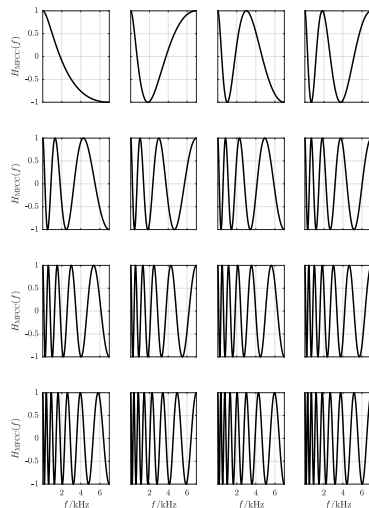


- constant Q filter spacing for higher frequencies (mel scale)
- FFT values are weighted and summed over bins for each band

spectral shape features

mel frequency cepstral coefficients 3/4

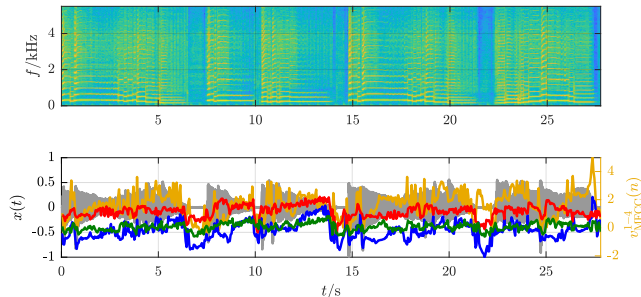
mel-warped cosine bases for DCT



spectral shape features

mel frequency cepstral coefficients 4/4

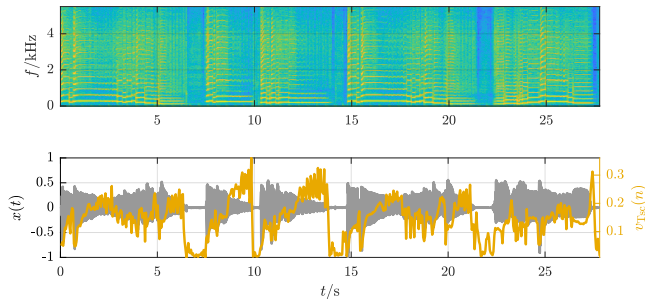
Property	DM	HTK	SAT
Num. filters	20	24	40
Mel scale	lin/log	log	lin/log
Freq. range	[100; 4000]	[100; 4000]	[200; 6400]
Normalization	Equal height	Equal height	Equal area



tonalness features

spectral crest factor

$$v_{\text{Tsc}}(n) = \frac{\max_{0 \leq k \leq \kappa/2-1} |X(k, n)|}{\sum_{k=0}^{\kappa/2-1} |X(k, n)|}$$



tonalness features

spectral crest factor

$$v_{\text{Tsc}}(n) = \frac{\max_{0 \leq k \leq \kappa/2-1} |X(k, n)|}{\sum_{k=0}^{\kappa/2-1} |X(k, n)|}$$

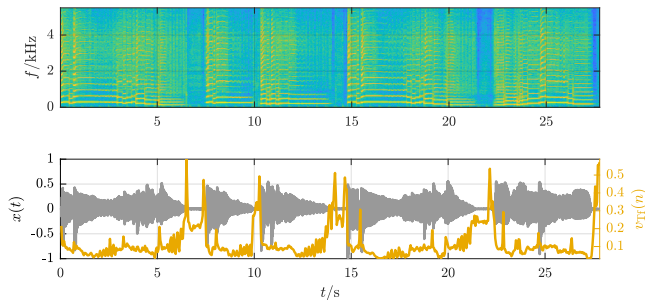
common variants:

- normalization
- power spectrum
- measure *per band* instead of whole spectrum

tonalness features

spectral flatness

$$v_{\text{Tf}}(n) = \frac{\sqrt{\prod_{k=0}^{\kappa/2-1} |X(k, n)|}}{2/\kappa \cdot \sum_{k=0}^{\kappa/2-1} |X(k, n)|}$$



tonalness features

spectral flatness

$$v_{\text{Tf}}(n) = \frac{\sqrt{\kappa/2} \prod_{k=0}^{\kappa/2-1} |X(k, n)|}{2/\kappa \cdot \sum_{k=0}^{\kappa/2-1} |X(k, n)|}$$

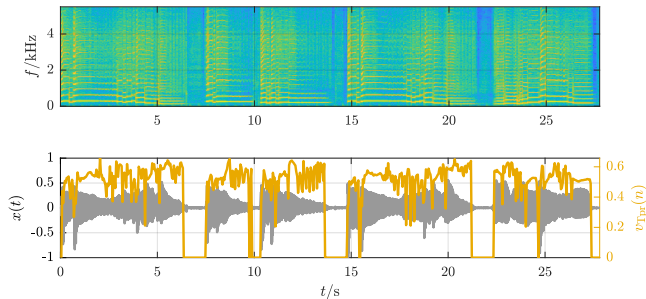
common variants:

- power vs. magnitude spectrum
- smoothed spectrum (avoid spurious 0-bins)
- measure *per band* instead of whole spectrum

tonalness features

spectral tonal power ratio

$$v_{\text{Tpr}} = \frac{E_T(n)}{\sum_{i=0}^{K/2-1} |X(k, n)|^2}$$



tonalness features

spectral tonal power ratio

$$v_{\text{TPr}} = \frac{E_{\text{T}}(n)}{\sum_{i=0}^{\kappa/2-1} |X(k, n)|^2}$$

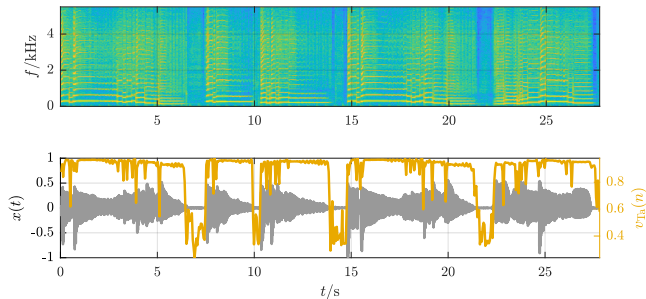
common variants:

- definition of tonal/non-tonal components
 - local maxima
 - peak salience
 - in periodic (harmonic) pattern
 - ...

tonalness features

maximum of ACF

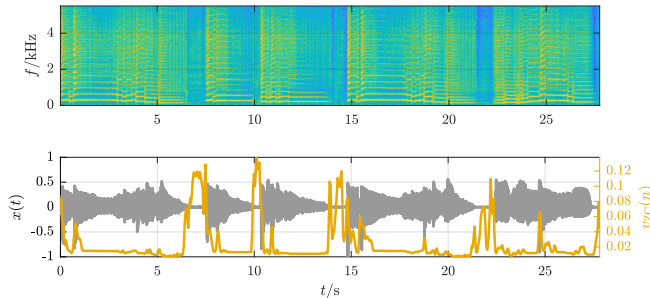
$$v_{Ta}(n) = \max_{\eta_1 \leq \eta \leq K-1} |r_{xx}(\eta, n)|$$



technical features

zero crossing rate

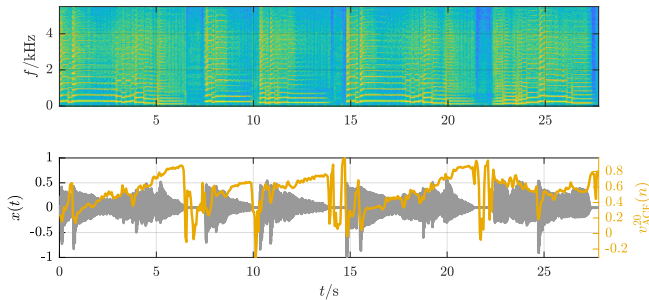
$$v_{ZC}(n) = \frac{1}{2 \cdot \mathcal{K}} \sum_{i=i_s(n)}^{i_e(n)} |\text{sign}[x(i)] - \text{sign}[x(i-1)]|$$



technical features

ACF coefficients

$$v_{\text{ACF}}^{\eta}(n) = r_{xx}(\eta, n) \quad \text{with } \eta = 1, 2, 3, \dots$$



feature learning

introduction

■ hand-crafted features:

- arbitrary definitions
- simple to compute
- mostly focus on one technical property
- provide limited information

■ feature learning:

- *automatically* learn features from data-set
- meaning not obvious, can combine multiple properties

feature learning

introduction

■ hand-crafted features:

- arbitrary definitions
- simple to compute
- mostly focus on one technical property
- provide limited information

■ feature learning:

- *automatically* learn features from data-set
- meaning not obvious, can combine multiple properties

feature learning

overview

■ principle

- 1 put (a lot of) raw data at input
- 2 learn a way of reducing dimensionality while keeping as much information as possible

■ advantages

- features might contain more useful information than provided by hand-crafted features
- no expert knowledge required

■ disadvantages

- usually time consuming
- limited ways of controlling the type of information learned

feature learning

overview

■ principle

- 1 put (a lot of) raw data at input
- 2 learn a way of reducing dimensionality while keeping as much information as possible

■ advantages

- features might contain more useful information than provided by hand-crafted features
- no expert knowledge required

■ disadvantages

- usually time consuming
- limited ways of controlling the type of information learned

feature learning

overview

■ principle

- 1 put (a lot of) raw data at input
- 2 learn a way of reducing dimensionality while keeping as much information as possible

■ advantages

- features might contain more useful information than provided by hand-crafted features
- no expert knowledge required

■ disadvantages

- usually time consuming
- limited ways of controlling the type of information learned

feature learning

approaches 1/2

■ dictionary learning (sparse coding, non-negative matrix factorization)

$$X = B \cdot A$$

X : input signal to be modeled (often spectrogram)

B : dictionary/template matrix (often set of single spectra that comprise the basic building blocks of X)

A : activation matrix indicating the weight and superposition of templates

- derive B, A , by minimizing a cost function, e.g. $\|X - BA\|_2$

→ templates are trained, activations are used as feature vector (length: number of templates)

feature learning

approaches 1/2

■ dictionary learning (sparse coding, non-negative matrix factorization)

$$X = B \cdot A$$

X : input signal to be modeled (often spectrogram)

B : dictionary/template matrix (often set of single spectra that comprise the basic building blocks of X)

A : activation matrix indicating the weight and superposition of templates

- derive B, A , by minimizing a cost function, e.g. $\|X - BA\|_2$

→ templates are trained, activations are used as feature vector (length: number of templates)

feature learning

approaches 2/2

■ clustering

- find clusters in data set (e.g., from magnitude spectra or simple features)
- store median of clusters (compare: template matrix)

→ features:

- ▶ binary vector (length: number of clusters, zero except for closest cluster)
- ▶ distance vector (distance to each cluster)

■ neural networks and **deep architectures**

- stack multiple layers of simple learning blocks
- each layer uses the output of the previous layer as input

→ feature: output of the highest layer

- task-dependency:
 - ▶ independent: use auto-encoder to maximize encoded information
 - ▶ dependent: train for specific task(s) (cf. multi-task learning) and use representation for other tasks (cf. transfer learning)

feature learning

approaches 2/2

■ clustering

- find clusters in data set (e.g., from magnitude spectra or simple features)
- store median of clusters (compare: template matrix)

→ features:

- ▶ binary vector (length: number of clusters, zero except for closest cluster)
- ▶ distance vector (distance to each cluster)

■ neural networks and **deep architectures**

- stack multiple layers of simple learning blocks
- each layer uses the output of the previous layer as input

→ feature: output of the highest layer

- task-dependency:
 - ▶ independent: use auto-encoder to maximize encoded information
 - ▶ dependent: train for specific task(s) (cf. multi-task learning) and use representation for other tasks (cf. transfer learning)

feature learning

approaches 2/2

■ clustering

- find clusters in data set (e.g., from magnitude spectra or simple features)
- store median of clusters (compare: template matrix)

→ features:

- ▶ binary vector (length: number of clusters, zero except for closest cluster)
- ▶ distance vector (distance to each cluster)

■ neural networks and **deep architectures**

- stack multiple layers of simple learning blocks
- each layer uses the output of the previous layer as input

→ feature: output of the highest layer

- task-dependency:
 - ▶ independent: use auto-encoder to maximize encoded information
 - ▶ dependent: train for specific task(s) (cf. multi-task learning) and use representation for other tasks (cf. transfer learning)

feature learning

approaches 2/2

■ clustering

- find clusters in data set (e.g., from magnitude spectra or simple features)
- store median of clusters (compare: template matrix)

→ features:

- ▶ binary vector (length: number of clusters, zero except for closest cluster)
- ▶ distance vector (distance to each cluster)

■ neural networks and **deep architectures**

- stack multiple layers of simple learning blocks
- each layer uses the output of the previous layer as input

→ feature: output of the highest layer

- task-dependency:
 - ▶ independent: use auto-encoder to maximize encoded information
 - ▶ dependent: train for specific task(s) (cf. multi-task learning) and use representation for other tasks (cf. transfer learning)

feature learning

approaches 2/2

■ clustering

- find clusters in data set (e.g., from magnitude spectra or simple features)
- store median of clusters (compare: template matrix)

→ features:

- ▶ binary vector (length: number of clusters, zero except for closest cluster)
- ▶ distance vector (distance to each cluster)

■ neural networks and **deep architectures**

- stack multiple layers of simple learning blocks
- each layer uses the output of the previous layer as input

→ feature: output of the highest layer

- task-dependency:
 - ▶ independent: use auto-encoder to maximize encoded information
 - ▶ dependent: train for specific task(s) (cf. multi-task learning) and use representation for other tasks (cf. transfer learning)

summary

lecture content

■ feature

- descriptor with condensed relevant information
- not necessarily interpretable by humans

■ low-level feature extraction

- usually extracted per short block of samples
- many features can be extracted from audio data, resulting in feature matrix

■ timbre

- mostly dependent on both spectral shape and time domain envelope characteristics
- multi-dimensional perceptual property not as clearly defined as pitch or loudness

■ instantaneous spectral shape features

- established set of baseline features
- usually extracted from the magnitude spectrum
- condensing various properties of the spectral shape into single values
- there exist multiple variants of “the same” feature

