Introduction to Audio Content Analysis

Module 3.6: Instantaneous Features

alexander lerch

Georgia Tech Center for Music Technology
College of Design

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

corresponding textbook section

Section 3.6

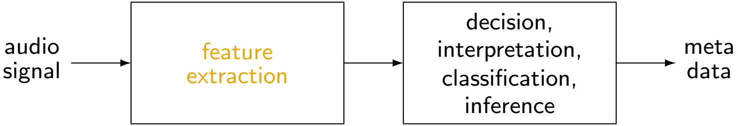
- **lecture content**
  - introduction to the concept of features
  - timbre
  - spectral shape instantaneous features

- **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:



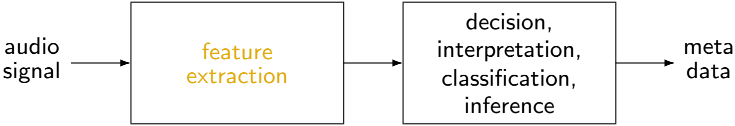
feature:

- *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:

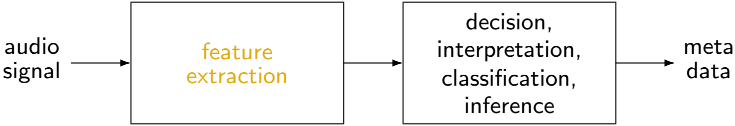


feature:

- *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

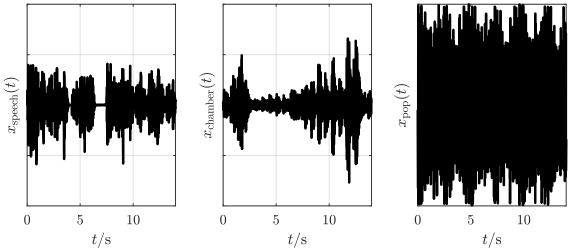
a feature . . .

- 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

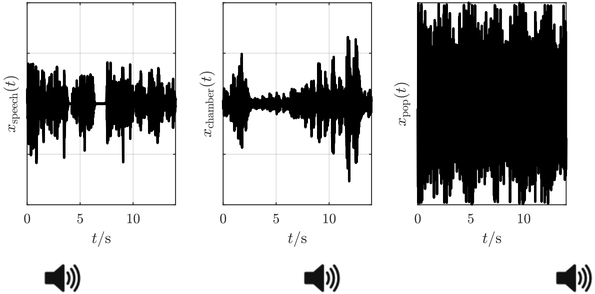
⇒ a feature describing envelope shape could help to distinguish these signal types

matlab source: plotWaveform.m

instantaneous features

feature example

waveform envelope of three different signals



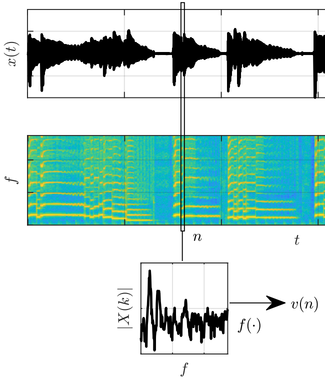
- envelopes of waveforms can have distinct shape

$\Rightarrow$ a feature describing envelope shape could help to distinguish these signal types

matlab source: plotWaveform.m

instantaneous features

feature extraction



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

$\Rightarrow$ feature matrix per audio input

matlab source: plotFeatureExtraction.m

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?

[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

[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

→ [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

$\Rightarrow$ timbre features typically focus on the **spectral shape** timbre is

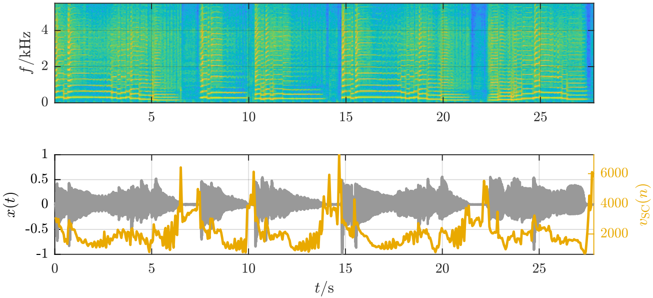
- a function of **temporal envelope**
    - attack time characteristics
    - amplitude modulations
    - ...
- a function of **spectral distribution**
    - spectral envelope
    - number of partials
    - energy distribution of partials
    - ...

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

$\Rightarrow$ timbre features typically focus on the **spectral shape**

spectral shape features
spectral centroid

$$v_{\mathrm{SC}}(n) = \frac{\sum\limits_{k=0}^{\mathcal{K}/2} k \cdot |X(k,n)|}{\sum\limits_{k=0}^{\mathcal{K}/2} |X(k,n)|}$$



matlab source: plotFeatures.m

spectral shape features
spectral centroid

$$v_{\mathrm{SC}}(n) = \frac{\sum\limits_{k=0}^{\mathcal{K}/2} k \cdot |X(k,n)|}{\sum\limits_{k=0}^{\mathcal{K}/2} |X(k,n)|}$$

common variants:

- power spectrum
- logarithmic frequency scale

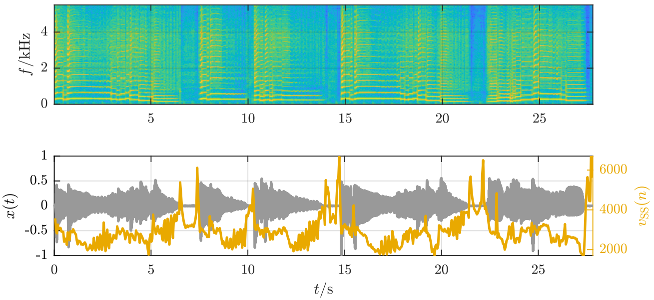
$$v_{\mathrm{SC,log}}(n) = \frac{\sum\limits_{k=k(f_{\mathrm{min}})}^{\mathcal{K}/2-1} \log_2\left(\frac{f(k)}{f_{\mathrm{ref}}}\right) \cdot |X(k,n)|^2}{\sum\limits_{k=k(f_{\mathrm{min}})}^{N/2-1} |X(k,n)|^2}$$

spectral shape features

spectral spread

Georgia Tech | Center for Music Technology
College of Design

$$v_{\mathrm{SS}}(n) = \sqrt{\frac{\sum\limits_{k=0}^{\mathcal{K}/2} \left(k - v_{\mathrm{SC}}(n)\right)^2 \cdot |X(k,n)|}{\sum\limits_{k=0}^{\mathcal{K}/2} |X(k,n)|}}$$



matlab source: plotFeatures.m

spectral shape features

spectral spread

$$v_{\mathrm{SS}}(n) = \sqrt{\frac{\sum\limits_{k=0}^{\mathcal{K}/2} \left(k - v_{\mathrm{SC}}(n)\right)^2 \cdot |X(k,n)|}{\sum\limits_{k=0}^{\mathcal{K}/2} |X(k,n)|}}$$

common variants:

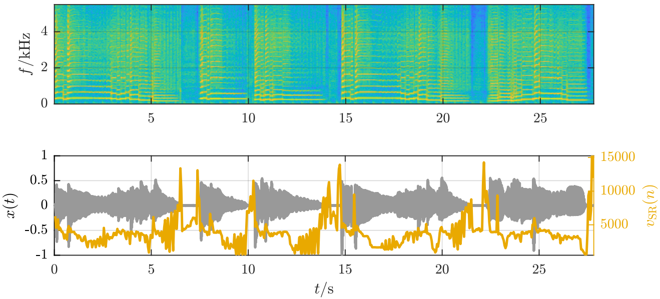
- same variants as with *Spectral Centroid*, e.g. logarithmic:

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

spectral shape features

spectral rolloff

$$v_{\mathrm{SR}}(n) = k_{\mathrm{r}} \Bigg|_{\sum\limits_{k=0}^{k_{\mathrm{r}}} |X(k,n)| = \kappa \cdot \sum\limits_{k=0}^{\mathcal{K}/2} |X(k,n)|}$$



matlab source: plotFeatures.m

spectral shape features

spectral rolloff

$$v_{\mathrm{SR}}(n) = k_{\mathrm{r}} \Bigg|_{\sum\limits_{k=0}^{k_{\mathrm{r}}} |X(k,n)| = \kappa \cdot \sum\limits_{k=0}^{\mathcal{K}/2} |X(k,n)|}$$

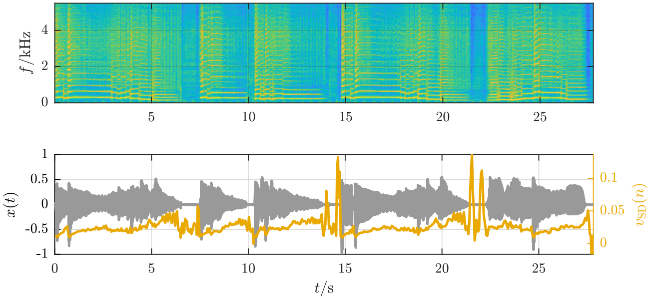
common variants:

- scaled to frequency
- power spectrum

spectral shape features

spectral decrease

$$v_{\mathrm{SD}}(n) = \frac{\sum\limits_{k=1}^{\mathcal{K}/2} \frac{1}{k} \cdot \big(|X(k,n)| - |X(0,n)|\big)}{\sum\limits_{k=1}^{\mathcal{K}/2} |X(k,n)|}$$



matlab source: plotFeatures.m

spectral shape features

spectral decrease

$$v_{\mathrm{SD}}(n) = \frac{\sum\limits_{k=1}^{\mathcal{K}/2} \frac{1}{k}\cdot\big(|X(k,n)| - |X(0,n)|\big)}{\sum\limits_{k=1}^{\mathcal{K}/2} |X(k,n)|}$$

common variants:

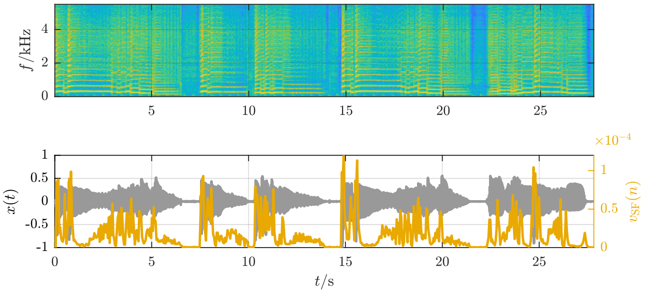
- restricted frequency range:

$$v_{\mathrm{SD}}(n) = \frac{\sum\limits_{k=k_{\mathrm{l}}}^{k_{\mathrm{u}}} \frac{1}{k}\cdot\big(|X(k,n)| - |X(k_{\mathrm{l}}-1,n)|\big)}{\sum\limits_{k=k_{\mathrm{l}}}^{k_{\mathrm{u}}} |X(k,n)|}$$

spectral shape features

spectral flux

$$v_{\mathrm{SF}}(n) = \frac{\sqrt{\sum\limits_{k=0}^{\mathcal{K}/2} \left(|X(k,n)| - |X(k,n-1)|\right)^2}}{\mathcal{K}/2 + 1}$$



matlab source: plotFeatures.m

spectral shape features

spectral flux

$$v_{\mathrm{SF}}(n) = \frac{\sqrt{\sum\limits_{k=0}^{\mathcal{K}/2}\left(|X(k,n)| - |X(k,n-1)|\right)^2}}{\mathcal{K}/2+1}$$

common variants:

$$\begin{aligned}
v_{\mathrm{SF}}(n,\beta) &= \frac{\sqrt[\beta]{\sum\limits_{k=0}^{\mathcal{K}/2-1}\left(|X(k,n)| - |X(k,n-1)|\right)^\beta}}{\mathcal{K}/2} \\
v_{\mathrm{SF},\sigma}(n) &= \sqrt{\frac{2}{\mathcal{K}}\sum\limits_{k=0}^{\mathcal{K}/2-1}\left(\Delta X(k,n) - \mu_{\Delta X}\right)^2} \\
v_{\mathrm{SF,log}}(n) &= \frac{2}{\mathcal{K}}\sum\limits_{k=0}^{\mathcal{K}/2-1}\log_2\left(\frac{|X(k,n)|}{|X(k,n-1)|}\right)
\end{aligned}$$

fundamentals
cepstrum 1/3

signal model:
convolution of *excitation signal* and *transfer function*

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

$$X(\mathrm{j}\omega) = E(\mathrm{j}\omega) \cdot H(\mathrm{j}\omega)$$

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

fundamentals

cepstrum 1/3

signal model:

convolution of *excitation signal* and *transfer function*

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

$$X(\mathrm{j}\omega) = E(\mathrm{j}\omega) \cdot H(\mathrm{j}\omega)$$

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

fundamentals
cepstrum 1/3

signal model:
convolution of *excitation signal* and *transfer function*

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

$$X(\mathrm{j}\omega) = E(\mathrm{j}\omega) \cdot H(\mathrm{j}\omega)$$

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

fundamentals

cepstrum 2/3

$$
\begin{aligned}
c_x(i) &= \mathfrak{F}^{-1}\left\{\log\left(X(\mathrm{j}\omega)\right)\right\} \\
&= \mathfrak{F}^{-1}\left\{\log\left(E(\mathrm{j}\omega)\right) + \log\left(H(\mathrm{j}\omega)\right)\right\} \\
&= \mathfrak{F}^{-1}\left\{\log\left(E(\mathrm{j}\omega)\right)\right\} + \mathfrak{F}^{-1}\left\{\log\left(H(\mathrm{j}\omega)\right)\right\} \\
\hat{c}_x(i_\mathrm{s}(n)\ldots i_\mathrm{e}(n)) &= \sum_{k=0}^{\mathcal{K}/2-1} \log\left(|X(k,n)|\right) e^{\mathrm{j}ki\Delta\Omega}
\end{aligned}
$$

fundamentals

cepstrum 2/3

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

$$
\hat{c}_x(i_\mathrm{s}(n)\dots i_\mathrm{e}(n)) = \sum_{k=0}^{\mathcal{K}/2-1} \log\left(|X(k,n)|\right) e^{\mathrm{j}ki\Delta\Omega}
$$

fundamentals

cepstrum 2/3

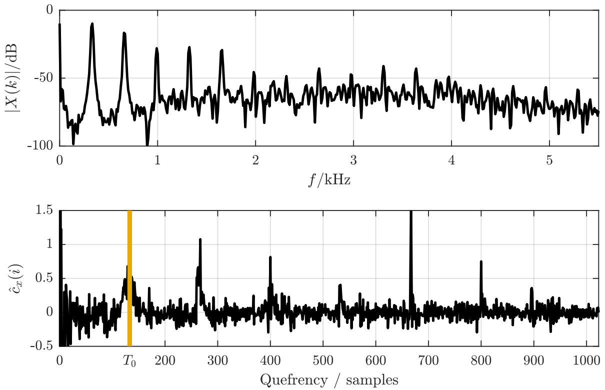
$$
\begin{aligned}
c_x(i) &= \mathfrak{F}^{-1}\left\{\log\left(X(\mathrm{j}\omega)\right)\right\} \\
&= \mathfrak{F}^{-1}\left\{\log\left(E(\mathrm{j}\omega)\right) + \log\left(H(\mathrm{j}\omega)\right)\right\} \\
&= \mathfrak{F}^{-1}\left\{\log\left(E(\mathrm{j}\omega)\right)\right\} + \mathfrak{F}^{-1}\left\{\log\left(H(\mathrm{j}\omega)\right)\right\}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

$$
\hat{c}_x(i_\mathrm{s}(n)\ldots i_\mathrm{e}(n)) = \sum_{k=0}^{\mathcal{K}/2-1} \log\left(|X(k,n)|\right) e^{\mathrm{j}ki\Delta\Omega}
$$

fundamentals

cepstrum 2/3

$$\hat{c}_x(i_\mathrm{s}(n)\ldots i_\mathrm{e}(n)) = \sum_{k=0}^{\mathcal{K}/2-1} \log\left(|X(k,n)|\right) e^{\mathrm{j}ki\Delta\Omega}$$



matlab source: plotF0Cepstrum.m

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

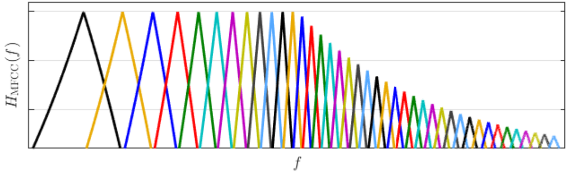
mel frequency cepstral coefficients 1/4

- 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)

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

spectral shape features

mel frequency cepstral coefficients 2/4



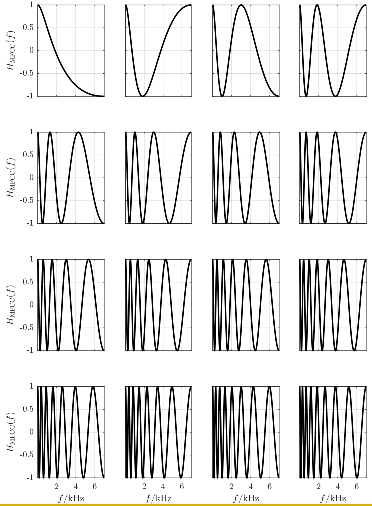
- constant Q filter spacing for higher frequencies (mel scale)
- FFT values are weighted and summed over bins for each band

matlab source: plotMfccFilterbank.m

spectral shape features

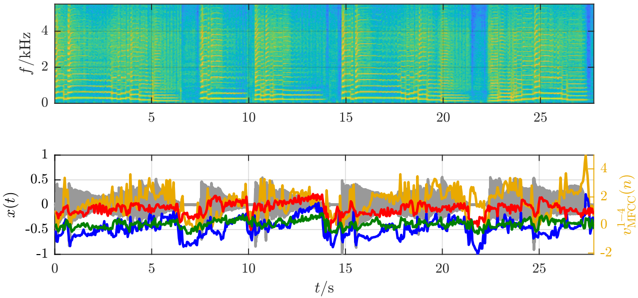
mel frequency cepstral coefficients 3/4

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



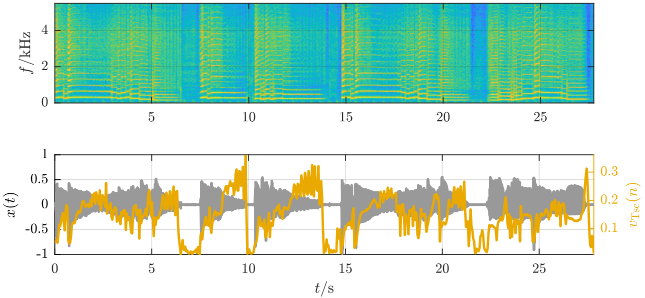
matlab source: plotFeatures.m

tonalness features

spectral crest factor

Georgia Tech Center for Music Technology College of Design

$$v_{\mathrm{Tsc}}(n) = \frac{\max\limits_{0\leq k\leq \mathcal{K}/2} |X(k,n)|}{\sum\limits_{k=0}^{\mathcal{K}/2} |X(k,n)|}$$



matlab source: plotFeatures.m

tonalness features

spectral crest factor

$$v_{\mathrm{Tsc}}(n) = \frac{\max\limits_{0 \leq k \leq \mathcal{K}/2} |X(k,n)|}{\sum\limits_{k=0}^{\mathcal{K}/2} |X(k,n)|}$$

common variants:

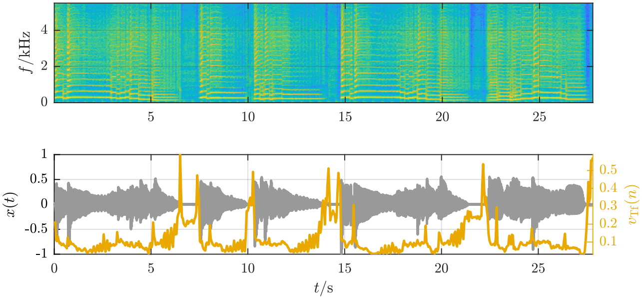
- normalization
- power spectrum
- measure *per band* instead of whole spectrum

tonalness features

spectral flatness

Georgia Tech Center for Music Technology College of Design

$$v_{\mathrm{Tf}}(n) = \frac{\sqrt[\mathcal{K}/2]{\prod\limits_{k=0}^{\mathcal{K}/2-1} |X(k,n)|}}{{}^{2}/_{\mathcal{K}} \cdot \sum\limits_{k=0}^{\mathcal{K}/2-1} |X(k,n)|}$$



matlab source: plotFeatures.m

tonalness features

spectral flatness

$$v_{\mathrm{Tf}}(n) = \frac{\sqrt[\mathcal{K}/2]{\prod\limits_{k=0}^{\mathcal{K}/2-1} |X(k,n)|}}{{}^{2}/_{\mathcal{K}} \cdot \sum\limits_{k=0}^{\mathcal{K}/2-1} |X(k,n)|}$$

common variants:

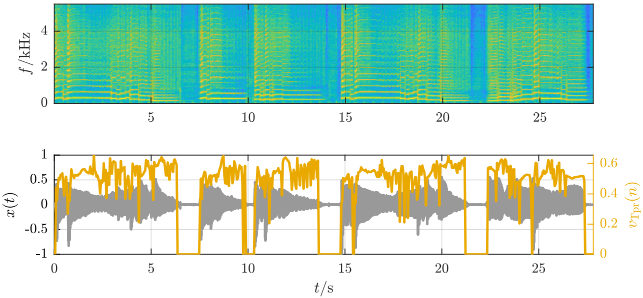
- power vs. magnitude spectrum
- smoothed spectrum (avoid spurious 0-bins)
- measure *per band* instead of whole spectrum

tonalness features

spectral tonal power ratio

Georgia Tech | Center for Music Technology
College of Design

$$v_{\mathrm{Tpr}} = \frac{E_{\mathrm{T}}(n)}{\sum\limits_{i=0}^{\mathcal{K}/2} |X(k,n)|^2}$$



matlab source: plotFeatures.m

tonalness features

spectral tonal power ratio

$$v_{\mathrm{Tpr}} = \frac{E_{\mathrm{T}}(n)}{\sum\limits_{i=0}^{\mathcal{K}/2} |X(k,n)|^2}$$

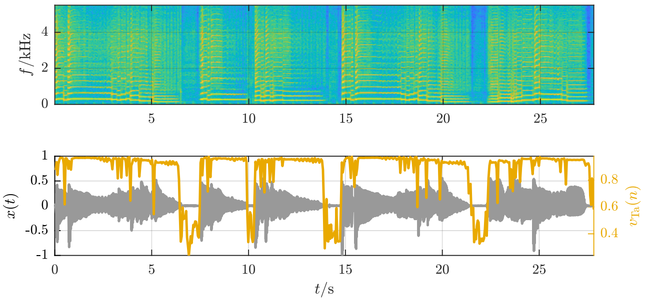
common variants:

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

tonalness features

maximum of ACF

$$v_{\mathrm{Ta}}(n) = \max_{0\leq\eta\leq\mathcal{K}-1} \left|r_{xx}(\eta, n)\right|$$

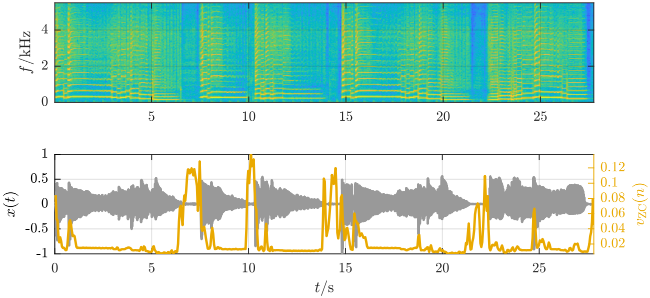


matlab source: matlab/displayFeatures.m

technical features

zero crossing rate

$$v_{\mathrm{ZC}}(n) = \frac{1}{2 \cdot \mathcal{K}} \sum_{i=i_\mathrm{s}(n)}^{i_\mathrm{e}(n)} \left| \mathrm{sign}\left[x(i)\right] - \mathrm{sign}\left[x(i-1)\right] \right|$$

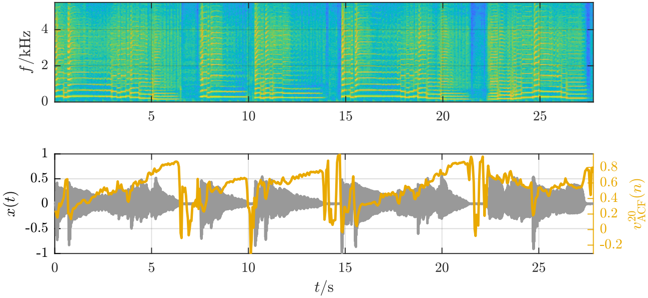


matlab source: matlab/displayFeatures.m

technical features

ACF coefficients

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



matlab source: matlab/displayFeatures.m

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

Georgia Tech Center for Music Technology College of Design

- **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. $||X - BA||_2$

$\rightarrow$ 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. $||X - BA||_2$

$\rightarrow$ 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)
    - $\rightarrow$ 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
    - $\rightarrow$ 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**
  - descriptor with condensed relevant information
  - not necessarily interpretable by humans
- **feature extraction**
  - usually extracted per short block of samples
  - many features can be extracted from audio data, resulting in feature matrix
- **instantaneous spectral shape features**
  - established set of baseline features
  - often extracted from the magnitude spectrum, describing timbre
  - condensing various properties of the spectral shape into single values
  - there exist multiple variants of “the same” feature
- **feature learning**
  - do not design but learn from data
  - not interpretable but should contain all relevant information