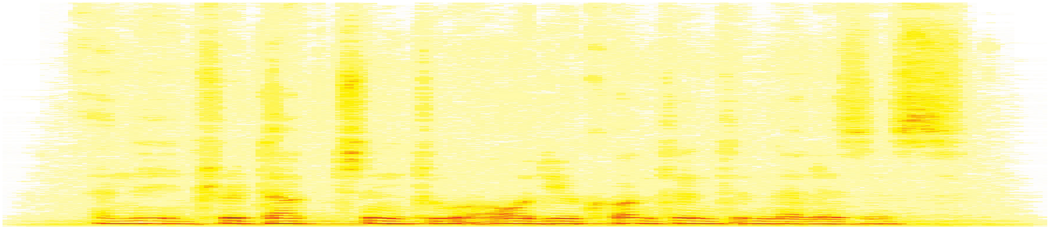
Introduction to Audio Content Analysis

Module 3.3: Feature Extraction — Additional (Instantaneous) Features

alexander lerch



Georgia Tech Center for Music Technology
College of Design

introduction
overview

corresponding textbook section

Chapter 3 — Instantaneous Features: pp. 54–63

- **lecture content**
  - tonalness features: describing ratio of tonal vs. noisy
  - technical features: describing basic signal properties
  - quick introduction to feature learning
- **learning objectives**
  - summarize features, describe their computation, and discuss their meaning
  - discuss advantages and disadvantages of feature learning as opposed to custom-designed features

Georgia Tech Center for Music Technology College of Design

corresponding textbook section

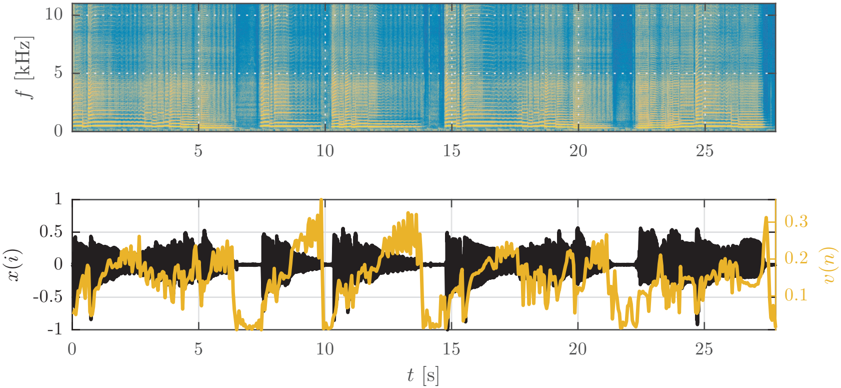
Chapter 3 — Instantaneous Features: pp. 54–63

- **lecture content**
  - tonalness features: describing ratio of tonal vs. noisy
  - technical features: describing basic signal properties
  - quick introduction to feature learning
- **learning objectives**
  - summarize features, describe their computation, and discuss their meaning
  - discuss advantages and disadvantages of feature learning as opposed to custom-designed features

tonalness features

spectral crest factor

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



matlab source: matlab/displayFeatures.m

tonalness features

spectral crest factor

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

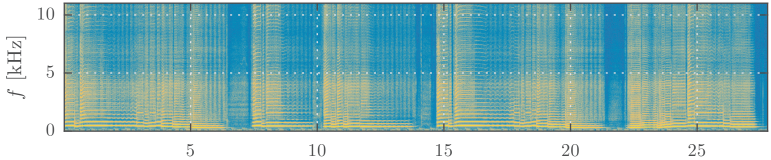
common variants:

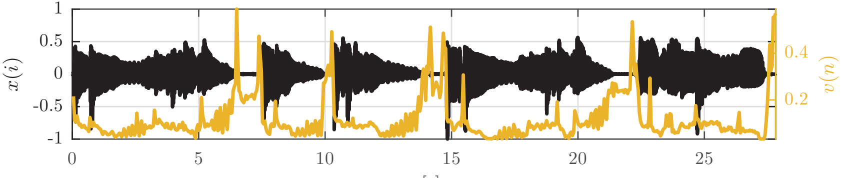
- normalization
- power spectrum
- measure *per band* instead of whole spectrum

tonalness features

spectral flatness

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





matlab source: matlab/displayFeatures.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:

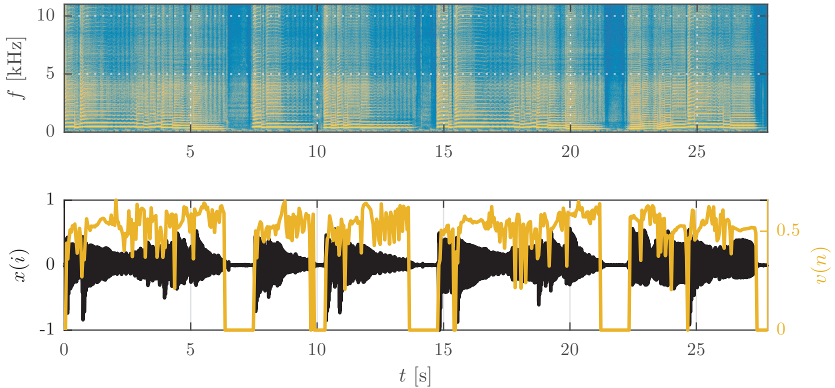
- power vs. magnitude spectrum
- smoothed spectrum (avoid spurious 0-bins)
- measure *per band* instead of whole spectrum

tonalness features

spectral tonal power ratio

Georgia Tech Center for Music Technology College of Design

$$v_{\mathrm{Tpr}} = \frac{E_{\mathrm{T}}(n)}{\sum\limits_{i=0}^{\mathcal{K}/2-1} |X(k,n)|^2}$$



matlab source: matlab/displayFeatures.m

tonalness features

spectral tonal power ratio

$$v_{\mathrm{Tpr}} = \frac{E_{\mathrm{T}}(n)}{\sum\limits_{i=0}^{\mathcal{K}/2-1} |X(k,n)|^2}$$

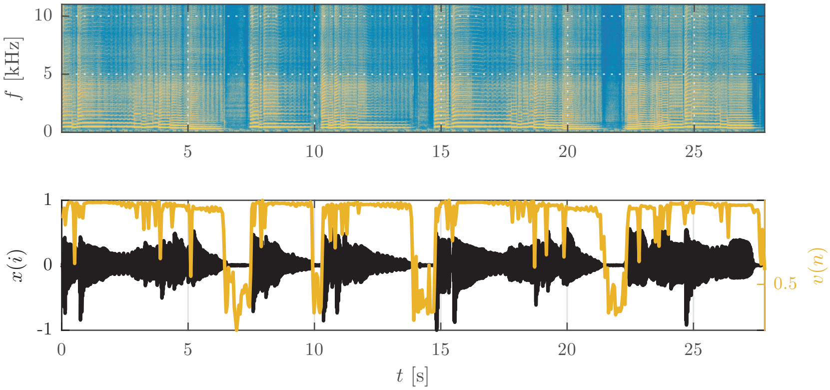
common variants:

- definition of tonal/non-tonal components

tonalness features

maximum of ACF

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

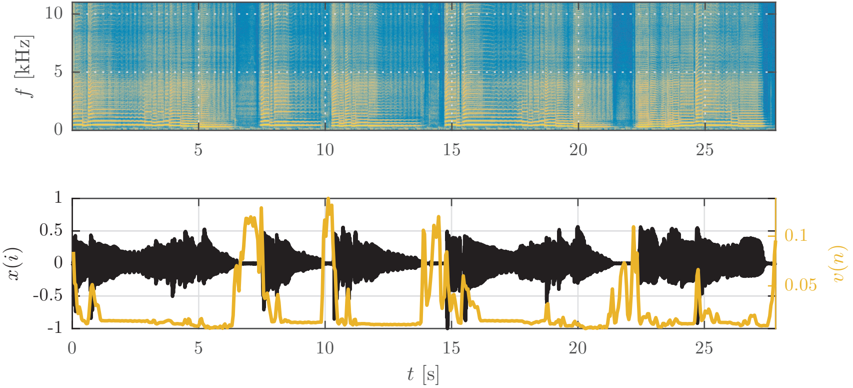


matlab source: matlab/displayFeatures.m

technical features

zero crossing rate

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

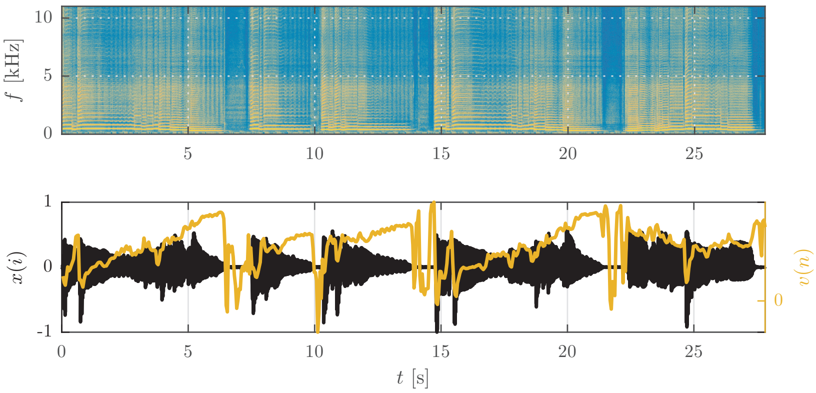


matlab source: matlab/displayFeatures.m

technical features
ACF coefficients

Georgia Tech Center for Music Technology College of Design

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



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

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

$\rightarrow$ 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

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

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

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

summary
lecture content

- **tonalness features**
  - in many cases closely related to spectral shape features
  - low level instantaneous features trying to estimate the ratio of tonal vs. noisy
- **technical features**
  - implementing a technical signal description that might or might not be interpretable
  - in many cases only of limited use
- **feature learning**
  - data-driven approach without expert knowledge
  - can be very powerful in the right context with enough representative data