Content- and Context-based Music Similarity and Retrieval

Markus Schedl
Peter Knees

{markus.schedl, peter.knees}@jku.at

Department of Computational Perception
Johannes Kepler University (JKU)
Linz, Austria
Overview

Goals of this class

– Introduction to the field of music similarity estimation
– Approaches to music retrieval

Parts:

I. About Music Similarity

II. Music Content Analysis and Similarity

III. Music Context-Based Similarity and Indexing

IV. Personalization and User Adaptation
Schedule

Monday (today!)
  Introduction to MIR, About music similarity, Evaluation of MIR systems, Basics in audio signal processing

Tuesday
  Music content based methods, MFCCs, FPs, PCPs, Similarity calculation

Wednesday
  Music context based methods, Text based methods, Co-occurrences, Collaborative filtering

Thursday
  User context, Personalization, Hybrid Methods

Friday
  Practical Exercise: Hybrid Music Recommender
Who we are

Markus Schedl
Assistant Professor of the Department of Computational Perception, JKU Linz
M.Sc. in Computer Science from Vienna University of Technology
Ph.D. in Computational Perception from Johannes Kepler University Linz
M.Sc. in Int’l Business Administration from Vienna University of Economics and Business Administration
Research interests: social media mining, music and multimedia information retrieval, recommender systems, information visualization, and intelligent/personalized user interfaces

Peter Knees
Assistant Professor of the Department of Computational Perception, JKU Linz
M.Sc. in Computer Science from Vienna University of Technology
Ph.D. in Computer Science from Johannes Kepler University Linz
Research interests: music and web information retrieval, multimedia, user interfaces, recommender systems, digital media arts
What is MIR? An Information Retrieval view

flowchart
  startbox{music entities (e.g. songs)} -> fetching -> resulting music entities
  fetching -> feature extraction
  feature extraction -> features
  features -> matching, similarities
  matching, similarities -> index
  index

UI
  query formulation
  visualization, auralization

browsing
  direct query
  query by example
Some Definitions of Music IR

“MIR is a multidisciplinary research endeavor that strives to develop innovative content-based searching schemes, novel interfaces, and evolving networked delivery mechanisms in an effort to make the world’s vast store of music accessible to all.”

[Downie, 2004]

“...actions, methods and procedures for recovering stored data to provide information on music.”

[Fingerhut, 2004]

“MIR is concerned with the extraction, analysis, and usage of information about any kind of music entity (for example, a song or a music artist) on any representation level (for example, audio signal, symbolic MIDI representation of a piece of music, or name of a music artist).”

[Schedl, 2008]
Typical MIR Tasks

- Feature extraction (audio-based vs. context-based approaches)
- Similarity measurement, recommendation, automated playlist generation (last.fm, Pandora, Echo Nest, ...)
- User interfaces, visualization, and interaction
- Audio fingerprinting (copyright infringement detection, music identification services like shazam.com or musicbrainz.org)
- Voice and instrument recognition, speech/music discrimination
- Structural analysis, alignment, and transcription (segmentation, self-similarities, music summarization, audio synthesis, audio and lyrics alignment, audio to score alignment (aka score following), and audio to score transcription)
- Classification and evaluation (ground truth definitions, quality measurement, e.g. for feature extraction algorithms, genre classification)
- Optical music recognition (OMR)
“Personalized Radio Stations”

- Pandora
- Last.fm
- Spotify Radio
- iTunes Radio
- Google Play Access All Areas
- Xbox Music

Continuously plays similar music

Based on content or collaborative filtering data

Optionally, songs can be rated for improved personalization

Applications: Automatic Playlist Generation
Applications: Browsing Music Collections

Intelligent organization for “one-touch access”

- music collections become larger and larger (on PCs as well as on mobile players)
- most UIs of music players still only allow organization and searching by textual properties according to scheme (genre-)artist-album-track

→ novel and innovative strategies to access music are sought in MIR

“intelligent iPod“ by CP@JKU [Schnitzer et al., MUM 2007]
Applications: Audio Identification

Query-by-example/audio fingerprinting:
excerpt of a song (potentially recorded in low quality) used to identify the piece

Query-by-humming:
input is not excerpt of a song, but melody hummed by the user

Examples:
www.shazam.com
www.soundhound.com
www.musicline.de/de/melodiesuche
Applications: Music Tweet Map
Applications: Music Tweet Map
Applications: Automatic Accompaniment
Part I

ABOUT MUSIC SIMILARITY
Music Retrieval and Similarity

To retrieve music (query-by-example), we need to calculate how similar two music pieces are.

What does similar mean?

- Sounding similar
- What does sounding similar mean?
  Genre (what is genre?), instruments, mood, melody, tempo, rhythm, singer/voice, … all of them? a combination?
- Any of that can contribute to two songs being perceived as similar, but describing sound alone falls short of grasping that phenomenon

Music similarity is a multi-faceted task
Music Similarity Examples

Which are similar?

Which go together?

Which are more similar?
The term “music similarity” is ill-defined

Experiments show that humans only agree to about 80% when asked to assign music pieces to genres (Lippens et al.; 2004) (Seyerlehner et al.; 2010)

Music similarity is highly subjective

Contextual factors are also important (but not in the signal!)

– artist/band context, band members, city/country, time/era, lyrics, language, genre, …
– political views of artists, marketing strategies, …
– also listening context, mood, peers (= user context)

Optimally, similarity is calculated taking into account all influencing factors:

audio content, music context, user context (difficult!), user properties (also difficult!)
Computational Factors Influencing Music Perception and Similarity

**Music Content**
- rhythm
- timbre
- melody
- harmony
- loudness

**Music Context**
- semantic labels
- song lyrics
- album cover artwork
- artist's background
- music video clips

**User Perception and Similarity**
- mood
- activities
- social context
- spatio-temporal context
- physiological aspects

**User Properties**
- music preferences
- musical training
- musical experience
- demographics

(Schedl et al.; JIIS 2013)
Implications for Evaluation

If similarity is such a subjective concept, how can we evaluate algorithms that claim to find similar pieces?

What is the Ground Truth?

- Class labels (genres)? Often used, often criticized
- Multi-class labels (tags)?

How to obtain (ranked) relevance?

Best strategies so far:

- Use listening data as retrieval ground truth (playlists)
- Ask users directly about similarity (listening tests)
Evaluation Campaign: MIREX

Music Information Retrieval Evaluation eXchange
  – Annual MIR benchmarking effort
  – Organized by UIUC since 2005 (Prof. J.S. Downie + team)

~ 20 tasks in 2013
  – Melody extraction, onset/key/tempo detection
  – Score following
  – Cover song detection
  – Query-by-singing/humming/tapping
  – etc.

Audio/signal-based tasks only so far
Evaluates query-by-example algorithms

Results evaluated by humans

“Evaluator question: Given a search based on track A, the following set of results was returned by all systems. Please place each returned track into one of three classes (not similar, somewhat similar, very similar) and provide an indication on a continuous scale of 0 - 100 of how similar the track is to the query.”

Each year: ~100 randomly selected queries, 5 results per query per algorithm (joined), “1 set of ears” per query

Friedman’s test to compare algorithms

No “winners,” but algorithm ranking
Other Evaluation Campaigns

**Million Song Dataset Challenge** (McFee et al.; 2012)
- Task: predicting songs a user will listen to
- Data: user listening history playcounts (48M)
- Evaluation: recall on ranking, MAP

**KDD Cup 2011** (Dror et al.; 2012)
- Task: predicting song ratings
- Data: Yahoo! Music data set (260M ratings)
- Evaluation: RMSE

**MusiClef** (e.g. @ MediaEval 2012)
- Task: multi-modal tagging of songs
- Data: audio, web, tag features, expert labels; 1355 songs
- Evaluation measures: precision, recall, F1-measure
The MusiClef 2012 Data Set

RAW CORPUS
- 1355 Tracks
- 218 Artists
  - "Rolling Stone 500 Greatest Songs of All Time"

Track
- Editorial Metadata
- Artist, Title, MusicBrainzID
  - FB-Mel
  - MFCCs
  - BLF
  - PS09

Track
- Audio Features

ANNOTATIONS
- Manual Annotation by Experts
  - 167 genres, 288 moods
- MusicBrainzID
- Last.fm [track.getTopTags]
- User tags
- MSD
- Web search
  - [Google]
  - [ArtistBased]
  - [English]
- Web pages
  - URL
  - Lucene
  - TF.IDF
Part II
MUSIC CONTENT ANALYSIS AND SIMILARITY
Categorization of Content-Based Features

Domain:

- **Time domain**
  consider signal in time/amplitude representation ("waveform")

- **Frequency domain**
  consider signal in frequency/magnitude representation

Transformation from time to frequency domain using, e.g., Fast Fourier Transform (FFT)
Categorization of Content-Based Features

Temporal scope:

- **Instantaneous**
  feature is valid for a “point in time” (NB: time resolution of ear is several msec!)

- **Segment**
  feature is valid for a segment, e.g., phrase, chorus (on a high level), or a chunk of \( n \) consecutive seconds in the audio signal

- **Global**
  feature is valid for whole audio excerpt or piece of music
Categorization of Content-Based Features

Level of abstraction:

- **Low-level**
  properties of audio signal (e.g., energy, zero-crossing-rate)

- **Mid-level**
  aggregation of low-level descriptors,
  applies psycho-acoustic models (cf. MFCC, FP);
  *typically the level used when estimating similarity*

- **High-level**
  musically meaningful to listener, e.g., melody, themes, motifs;
  “semantic” categories, e.g., genre, time period, mood, …
  (cf. semantic tags learned from audio features)
How to Describe Audio Content?

Possible idea: get features that describe music the way humans do and compute similar songs based on this information.

Unfortunately we are not able to extract most of these features reliably (or at all…)

– even “simple” human concepts are difficult to model ("semantic gap")
– even tempo estimation is very hard…
– NB: a human annotation approach is done in the Music Genome Project (cf. Pandora’s automatic radio station service)

Furthermore some of these features are quite subjective (e.g., mood)

Need to find computable descriptors that capture these dimensions somehow (…and work acceptably)
Acoustic property to describe:

- **Loudness**: perceived strength of sound; *e.g.*, *energy*
- **Pitch**: frequency, psychoacoustic ordering of tones (on scale; from low to high); *e.g.*, *chroma-features*
- **Timbre**: “tone color”, what distinguishes two sounds with same pitch and loudness; *e.g.*, *MFCCs*
- **Chords and harmony**: simultaneous pitches
- **Rhythm**: pattern in time; *e.g.*, *FPs*
- **Melody**: sequence of tones; combination of pitch and rhythm

cf. (Casey et al.; 2008)
Scheme of Content-Based Feature Extraction

- Analog signal
- Sampling
- Quantization
- Pulse Code Modulation (PCM)
- Framing
- Time domain feature calculation
  - Windowing
  - FFT
- Frequency domain feature calculation
- Aggregation, model building (mean, median, sum, GMM, HMM)
- Feature value, vector, or matrix
PCM: analog signal is sampled at equidistant intervals and quantized in order to store it in digital form (here with 4 bits)

Problems that may occur in ADC:

- **Quantization error**: difference between the actual analog value and quantized digital value
- **Solution**: finer resolution (use more bits for encoding), common choice in music encoding: 16 bits per channel

- Due to **Nyquist–Shannon Sampling Theorem**, frequencies above $\frac{1}{2}$ of sampling frequency (Nyquist frequency) are discarded or heavily distorted
- **Solution**: choose a sampling frequency that is high enough (e.g. 44,100 Hz for Audio CDs)
In short-time signal processing, pieces of music are cut into segments of fixed length, called frames, which are processed one at a time; typically, a frame comprises 256 - 4096 samples.
Scheme of Content-Based Feature Extraction

1. Analog signal
2. Pulse Code Modulation (PCM)
   - Sampling
   - Quantization
   - Framing
   - Frame 1: e.g. sample 1...256
   - Frame 2: e.g. sample 129...384
   - Frame 3: e.g. sample 257...512
   - ... Frame n
3. Time domain feature calculation
   - Windowing
4. Frequency domain feature calculation
   - FFT
5. Aggregation, model building (mean, median, sum, GMM, HMM)
6. Feature value, vector, or matrix
Low-Level Feature: Zero Crossing Rate

**Scope:** time domain

**Calculation:**

\[
ZCR_t = \frac{1}{2} \cdot \sum_{k=t\cdot K}^{(t+1)\cdot K-1} |\text{sgn}(s(k)) - \text{sgn}(s(k + 1))|
\]

**Description:**
number of times the amplitude value changes its sign within frame \(t\)

**Remarks:**
commonly used as part of a low-level descriptor set
+ might be used as an indicator of pitch
+ sometimes stated to be an approximate measure of the signal’s noisiness
– in general, low discriminative power
Zero Crossing Rate: Illustration

\( K=20 \)

\( \text{hop size} = 10 \)
Zero Crossing Rate: Examples
Low-Level Feature: Amplitude Envelope

Scope: time domain

Calculation:

\[
AE_t = \max_{k=t\cdot K} s(k)
\]

Description:
maximum amplitude value within frame \( t \)

Remarks:

similar to RMS energy (see next), but less stable
+ important for beat-related feature calculation, e.g. for beat detection
– discriminative power not clear
– sensitive to amplitude outliers

\( s(k) \) ... amplitude of \( k^{th} \) sample in time domain
\( K \) ... frame size (number of samples in each frame)
Amplitude Envelope: Illustration

\( K = 20 \)
\( \text{hop size} = 10 \)
Amplitude Envelope: Examples

Amplitude Envelope

Baroque

Blues

Choir

Electronic

Indian

Metal

Piano

Zen Flute

0 200 400 600 800 1000 1200 1400 1600 1800 2000
Framas
Low-Level Feature: RMS Energy

Root-Mean-Square Energy (aka RMS power, RMS level, RMS amplitude)

Scope: time domain

Calculation:

$$RMS_t = \sqrt{\frac{1}{K} \cdot \sum_{k=t \cdot K}^{(t+1) \cdot K-1} s(k)^2}$$

Remarks:

+ beat-related feature, can be used for beat detection
+ related to perceived intensity
+ good loudness estimation
– discriminative power not clear
RMS Energy: Illustration

\( K = 20 \)

\( \text{hop size} = 10 \)
Scheme of Content-Based Feature Extraction

- **analog signal**
  - **sampling**
  - **quantization**
  - **Pulse Code Modulation (PCM)**
  - **framing**

  - frame 1: e.g. sample 1...256
  - frame 2: e.g. sample 129...384
  - frame 3: e.g. sample 257...512
  - ... frame n

- **time domain feature calculation**
  - **windowing**
  - **FFT**

- **frequency domain feature calculation**
  - **aggregation, model building (mean, median, sum, GMM, HMM)**
  - **feature value, vector, or matrix**
Fourier Transform

Transformation of the signal
  from **time domain** (time vs. amplitude)
  to **frequency domain** (frequency vs. magnitude)

- Theorem: any continuous periodic function with a period of $2\pi$ can be represented as the sum of sine and/or cosine waves (of different frequencies)
- Implication: any audio signal can be decomposed into an infinite number of overlapping waves when periodic

- Periodicity is achieved by multiplying the PCM magnitude values of each frame with a suited function, e.g., a Hanning window (**windowing**)
- In our case: **Discrete Fourier Transform (DFT)**
- In practice efficiently calculated via **Fast Fourier Transform (FFT)** (Cooley, Tukey; 1965)
Spectrogram

(aka Sonogram)

Fourier Transform actually results in complex values (representing amplitude and phase)
Transformation for display and better interpretation of frequency magnitudes:

\[ \text{spectrogram}(t, \omega) = |\text{STFT}(t, \omega)|^2 \]

Activation strength is coded with color (or grey value) rather than plotted as a curve
Allows for two-dimensional representation of activations over whole piece
Spectrogram
Representation as STFT Spectrogram

STFT

Barsque

Choir

Indian

Piano

Blues

Electronic

Metal

Zen Flute
Low-Level Feature: Spectral Centroid

**Scope**: frequency domain

**Calculation**:
\[ C_t = \frac{\sum_{n=1}^{N} M_t(n) \cdot n}{\sum_{n=1}^{N} M_t(n)} \]

**Description**: center of gravity of the magnitude spectrum of the DFT, i.e. the frequency (band) region where most of the energy is concentrated

**Remarks**:
- used as measure of sound sharpness (strength of high frequency energy)
- sensitive to low pass filtering (downsampling) as the high frequency bands are given more weight
- sensitive to white noise (for the same reason)
Spectral Centroid: Illustration

Spectral Centroid

Baroque

Blues

Choir

Electronic

Indian

Metal

Piano

Zen Flute
Low-Level Feature: Bandwidth

Scope: frequency domain

Calculation:

\[ BW_t^2 = \frac{\sum_{n=1}^{N} (n - C_t)^2 \cdot M_t(n)}{\sum_{n=1}^{N} M_t(n)} \]

Description: describes the spectral range of the interesting parts of the signal

Remarks:
+ average bandwidth of a piece of music may serve as indicator of aggressiveness
– no information about perceived rhythmic structure
– not suited to distinguish different parts of a piece of music (cf. vocal part in metal piece not visible)
Bandwidth: Illustration
Low-Level Feature: Spectral Flux

(aka Delta Spectrum Magnitude)

Scope: frequency domain

Calculation:

\[ F_t = \sum_{n=1}^{N} \left( N_t(n) - N_{t-1}(n) \right)^2 \]

Description:
measures the rate of local spectral change, big spectral change from frame \( t-1 \) to \( t \) \( \rightarrow \) high \( F_t \) value

Remarks:

- commonly used as part of a low-level descriptor set
- may be used to distinguish between aggressive and calm music
- may serve as speech detector

\( N_t \)...frame-by-frame normalized frequency distribution in frame \( t \)
\( N \)...number of highest frequency band
Spectral Flux: Illustration

Spectral Flux

Metal

Choir