Automatic Mixing of Music Segments

Miguel Graça

Rheinisch-Westfälische Technische Hochschule Aachen

miguel.graca@rwth-aachen.de

Topics in Computer Music

June 17, 2016

Overview

Introduction

Problem

Given a collection of songs and an input song, find

- the most fitting follow-up song
- the most fitting transitioning

Introduction

Problem

Given a collection of songs and an input song, find

- the most fitting follow-up song
- the most fitting transitioning

Main issues:

- Subjective measure: What is the most fitting transition?
  - Humans require skill and experience to mix
- Machine interpretation of a song
- Different tempi

Related Work

- Automatic mixing (2003) [2]:
  - Supervised learning
  - Learn the preference of song transitions of a human

Related Work

- Automatic mixing (2003) [2]:
    - Supervised learning
    - Learn the preference of song transitions of a human
- Music mashups (2008) [8]:
    - Create a song by fusing multiple songs

Related Work

- Automatic mixing (2003) [2]:
  - Supervised learning
  - Learn the preference of song transitions of a human
- Music mashups (2008) [8]:
  - Create a song by fusing multiple songs
- Fully automatic mixing (2009) [5]:
  - Transition between any two songs
  - Use tempo adjustment techniques

Related Work

- Automatic mixing (2003) [2]:
    - Supervised learning
    - Learn the preference of song transitions of a human
- Music mashups (2008) [8]:
    - Create a song by fusing multiple songs
- Fully automatic mixing (2009) [5]:
    - Transition between any two songs
    - Use tempo adjustment techniques
- Vocal timbre analysis (2014) [6]:
    - Identify a singer based on patterns in audio signal
    - Representation of a song using words

Related Work

- Automatic mixing (2003) [2]:
  - Supervised learning
  - Learn the preference of song transitions of a human
- Music mashups (2008) [8]:
  - Create a song by fusing multiple songs
- Fully automatic mixing (2009) [5]:
  - Transition between any two songs
  - Use tempo adjustment techniques
- Vocal timbre analysis (2014) [6]:
  - Identify a singer based on patterns in audio signal
  - Representation of a song using words
- Topic-based mixing (2015) [3]:
  - Transition to the most similar songs in a dataset
  - Attempts to find a meaning in a song
  - Focus of this talk

Technical Approach

Idea

Consider similar segments of songs instead of songs for transitions

Technical Approach

Idea

Consider similar segments of songs instead of songs for transitions

Determine similarity of segments:

- Beat similarity: How similar are the beats?
- Topic similarity: Difference between the notes captured

Beat Similarity

Motivation

- Beat is given by percussion instruments
- Tempo is linked to beat
- Assumption: Similar songs have similar beats

Beat Similarity

Motivation

- Beat is given by percussion instruments
- Tempo is linked to beat
- Assumption: Similar songs have similar beats

Idea

Consider two segments i and j:

- Extract the low-frequency signal using a low-pass filter
- Calculate the distance between each peak
- Compare the distances of the peaks of each segment

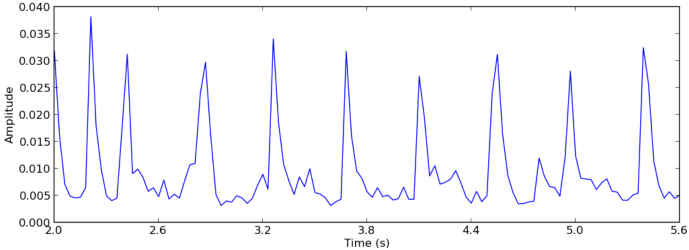

Figure: Audio signal after a low-pass filter of 500Hz.
Source: "Asche zu Asche - Rammstein"

Amplitude peak distances $D_{\text{peak}} \in \mathbb{R}^{N-1}$ are determined by:

- Highest amplitude within a time-frame
- N peaks are captured

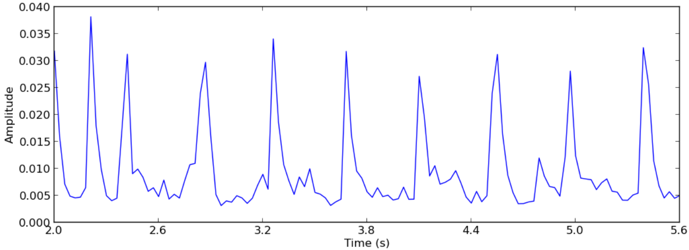

Figure: Audio signal after a low-pass filter of 500Hz.
Source: "Asche zu Asche - Rammstein"

Amplitude peak distances $D_{\text{peak}} \in \mathbb{R}^{N-1}$ are determined by:

- Highest amplitude within a time-frame
- N peaks are captured

Similarity measure S_{beat} of fragments i and j:

$$S_{\text{beat}}(i,j) = \frac{1}{\sum_{k=1}^{N-1} |D^i_{\text{peak},k} - D^j_{\text{peak},k}| + 1}$$

Topic Similarity

Motivation

Both music segments should have similar

(i) musical messages

(ii) notes played

Topic Similarity

Motivation

Both music segments should have similar

(i) musical messages

(ii) notes played

Idea

Interpret songs as word-documents:

- Words describe the topics of a song
- Determine similarity based on a topic distribution
- Possible to apply methods from natural language processing

Topic Similarity

Motivation

Both music segments should have similar

(i) musical messages

(ii) notes played

Idea

Interpret songs as word-documents:

- Words describe the topics of a song
- Determine similarity based on a topic distribution
- Possible to apply methods from natural language processing

Problem: How does one represent a song as a document?

Pre-processing of the audio signal:

- Capture note information within a time-frame
- Extract 12-element vectors (ChromaVector)
- Each entry is the intensity of a pitch in $\{C, C\#, \ldots, B\}$

Pre-processing of the audio signal:

- Capture note information within a time-frame
- Extract 12-element vectors (ChromaVector)
- Each entry is the intensity of a pitch in $\{C, C\#, \ldots, B\}$

ChromaWord [4] extraction:

- Ignore notes which are not part of 70% total power $\rightarrow$ noise
- The 4 strongest pitches represent a word
  - Words can have only $1, 2, 3$ pitches
  - 0 words corresponds to silence

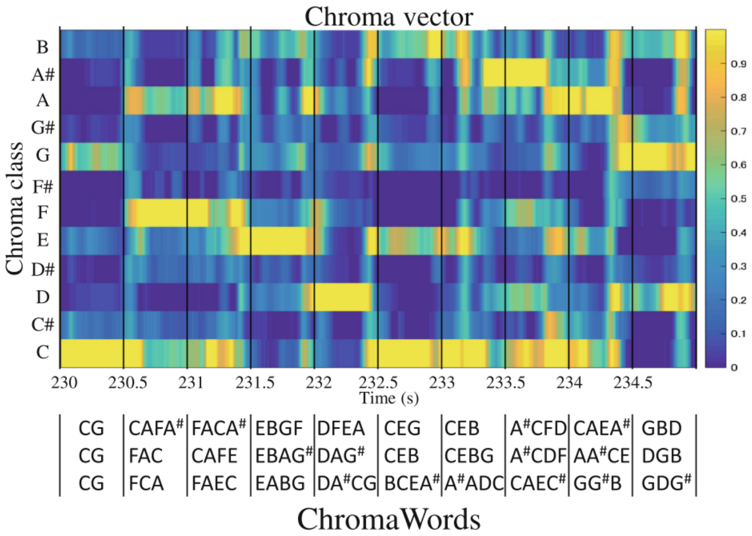

Figure: ChromaVector decomposition. Source [4]

Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) [1]:

- Latent: Assumption of hidden states (topics)
- Dirichlet: Usage of the Dirichlet distribution
- Allocation: Assignment of hidden states to observable events

Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) [1]:

- Latent: Assumption of hidden states (topics)
- Dirichlet: Usage of the Dirichlet distribution
- Allocation: Assignment of hidden states to observable events

Probabilistic modelling of topics:

- Each segment is assigned a probability to be of a certain topic
- Multiple topics are possible
- Similarity measure → compare topic distributions

Similarity measure $S_{\text{topic}}(i,j)$ for segments i, j:

$$S_{\text{topic}}(i,j) = \frac{1}{\sum_{k=1}^{K} |f_{i,k} - f_{j,k}| + 1}$$

- $f_{i,k}$ probability of k-th topic for segment i

Similarity measure $S_{\text{topic}}(i, j)$ for segments i, j:

$$S_{\text{topic}}(i, j) = \frac{1}{\sum_{k=1}^{K} |f_{i,k} - f_{j,k}| + 1}$$

- $f_{i,k}$ probability of k-th topic for segment i

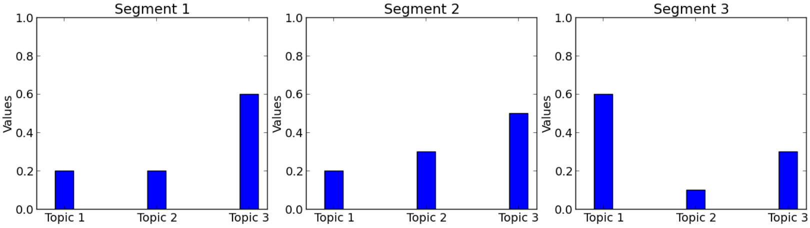

Figure: Fictional 3-topic distribution for three segments

First segment is more similar to the second than the third

Similarity Measure

Overall similarity S of segments i and j given by:

$$S(i,j) = \frac{S_{\text{topic}}(i,j) + S_{\text{beat}}(i,j)}{2}$$

Similarity Measure

Overall similarity S of segments i and j given by:

$$S(i,j) = \frac{S_{\text{topic}}(i,j) + S_{\text{beat}}(i,j)}{2}$$

Perform transitions using:

- The most similar song segment
- Volume cross-fading

Experimental Setup

Compare with state-of-the-art features that are applied with LDA:

- Mel Frequency Cepstral Coefficient (MFCC)
- ChromaVector
- ChromaWord

First two methods use k-means cluster means as words [6]

Experimental Setup

Compare with state-of-the-art features that are applied with LDA:

- Mel Frequency Cepstral Coefficient (MFCC)
- ChromaVector
- ChromaWord

First two methods use k-means cluster means as words [6]

Main question: Which representation better captures similarity?

Experimental Setup

Compare with state-of-the-art features that are applied with LDA:

- Mel Frequency Cepstral Coefficient (MFCC)
- ChromaVector
- ChromaWord

First two methods use k-means cluster means as words [6]

Main question: Which representation better captures similarity?

Setup:

- 50 rock, pop and dance songs as a dataset
- 2192 5s fragments in total
- 100 latent topics were assumed
- No beat similarity is taken into account

Results

Evaluation

- Pair-wise comparison of fragment similarity
- Three segment pairs were chosen per feature comparison
- Evaluation performed with 8 human subjects

Results

Evaluation

- Pair-wise comparison of fragment similarity
- Three segment pairs were chosen per feature comparison
- Evaluation performed with 8 human subjects

	MFCC	ChromaVector	ChromaWord
MFCC	-	Worse	Worse
ChromaVector	Better	-	Worse
ChromaWord	Better	Better	-

Table: Empirical results for feature performance. Row-wise comparison with each column.

Audio Examples

Carnival of Hono & Mori - Sekai No Owari
↓
Get Lucky - Daft Punk

Robot Rock - Daft Punk
↓
Y.M.C.A. - The Village People

Clips are credited to Tatsunori Hirai of Waseda University, Tokyo

Conclusion

A work was presented that

- automates song transitioning within a collection of songs
- applies beat similarity to ensure smooth transitions
- estimates similarity of song segments based on latent topics
- introduces a novel feature that represents topics effectively

Conclusion

A work was presented that

- automates song transitioning within a collection of songs
- applies beat similarity to ensure smooth transitions
- estimates similarity of song segments based on latent topics
- introduces a novel feature that represents topics effectively

Points of improvement:

- Non-trained songs cannot be evaluated with LDA
- ChromaWord information is limited to 12 pitches
- Take lyrics into consideration
- Tempo adjustment during transitions (see technique in [5])

D. M. Blei, A. Y. Ng, and M. I. Jordan.
Latent dirichlet allocation.
The Journal of Machine Learning Research, 3:993–1022, 2003.

T. Fujio and H. Shiizuka.
A system of mixing songs for automatic dj performance using genetic programming.
In *6th Asian Design International Conference*, 2003.

T. Hirai, H. Doi, and S. Morishima.
Musicmixer: Computer-aided dj system based on an automatic song mixing.
2015.

T. Hirai, H. Doi, and S. Morishima.
Musicmixer: Automatic dj system considering beat and latent topic similarity.
In *MultiMedia Modeling*, pages 698–709. Springer, 2016.

H. Ishizaki, K. Hoashi, and Y. Takishima.
Full-automatic dj mixing system with optimal tempo adjustment based on measurement function of user discomfort.
In *ISMIR*, pages 135–140, 2009.

T. Nakano, K. Yoshii, and M. Goto.
Vocal timbre analysis using latent dirichlet allocation and cross-gender vocal timbre similarity.
In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5202–5206. IEEE, 2014.

R. Schlüter and H. Ney.
Introduction to automatic speech recognition, 2015.

N. Tokui.
Massh!: a web-based collective music mashup system.
In *Proceedings of the 3rd International Conference on Digital Interactive Media in Entertainment and Arts*, pages 526–527. ACM, 2008.

Feature Extraction

ChromaVector extraction:

- Audio signal $\rightarrow$ 12-element vector
- Each entry is a musical note, i.e $\{C, C\#, \ldots, B\}$
- 200ms window moving each 10ms

ChromaWord extraction:

- The 4 strongest pitches represent a word
  - Words can have only $1, 2, 3$ pitches
  - 0 words corresponds to silence
- Ignore notes which are not part of 70% total power $\rightarrow$ noise
- 10ms window $\rightarrow$ 20 words per ChromaVector

Notation

- s_1^M: song segments with $M \in \mathbb{N}$
- $w_{m,1}^{m,N}$: words with $N \in \mathbb{N}$ of segment s_m
- t_1^K: topics with $K \in \mathbb{N}$
- $\theta_1^K \sim Dirichlet(\alpha_1^K)$ with $\alpha_k \in \mathbb{R}_{>0}$
  Dirichlet distribution parameters
- β_1^K with $\beta_k \in [0,1]^{|V|}$: Probabilities of each word being assigned the topic t_k

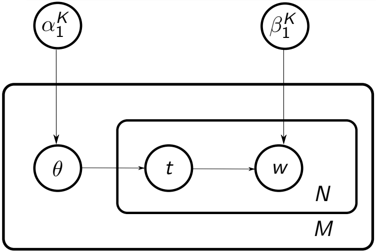

Figure: Variable hierarchy in latent dirichlet allocation. Source: [1]

Model

Joint model for segment s_m conditioned on parameters α_1^K, β_1^K:

$$p(\theta_m, z_1^K, w_{m,1}^{m,N} | \alpha_1^K, \beta_1^K) = p_{\text{Dir}}(\theta_m | \alpha_1^K) \cdot \prod_{n=1}^{N} p_{\text{Multinomial}}(z_n | \theta_m, 1) \cdot p(w_{m,n} | z_n, \beta_1^K) \quad (1)$$

Note that the multinomial distribution uses 1 trial

Training:

- α_1^K and β_1^K are the free parameters
- Variational expectation maximization [1]

The probability of a topic t_k of a song segment s_m is given by $\theta_{m,k}$

Generative process

Word generation is performed for each segment s_m as in Eq. 1:

(i) Choose topic weights $\theta_m \sim \text{Dirichlet}(\alpha)$

(ii) For each word $w_{m,n}$:

  (i) Assign a topic $t_{m,n,k} \sim \text{Multinomial}(\theta_m, 1)$

  (ii) Choose word $w_{m,n} \sim \text{Multinomial}(\beta_k, 1)$

Generative process:

- Samples can be generated by random processes
- Hidden variables are deduced by the following:

$$p(\theta_m, z_1^K | w_{m,1}^{m,N}, \alpha_1^K, \beta_1^K) = \frac{p(\theta_m, z_1^K, w_{m,1}^{m,N} | \alpha_1^K, \beta_1^K)}{p(w_{m,1}^{m,N} | \alpha_1^K, \beta_1^K)} \quad (2)$$

Mel Frequency Cepstral Coefficients (MFCCs)

Motivation

- Similar sounds should have similar features
- Noise suppression
- Emphasis of low-frequency differences

Feature vector $x \in \mathbb{R}^N$:

- $N \in [16, 50]$

Used in:

- Automatic speech recognition
- Music information retrieval

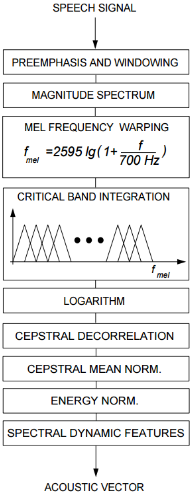

Figure: MFCC extraction process. Source: [7]

Approach of Nakano et al. [6]

Word representation

- Consider features in $\mathbb{R}^N$
- Perform K-means clustering and assign each feature to a cluster
- Words w_1^K are represented by one-hot encoded vectors
- A feature $x \in \mathbb{R}^N$ is assigned a word by $x_k \in \{0,1\}^K$ with:

$$x_{k,i} = \begin{cases} 1 & i = k \\ 0 & \text{otherwise,} \end{cases}$$

  with k being the index of the nearest cluster mean
- Assign words to features in a continuous space