Music Understanding

Roger B. Dannenberg
School of Computer Science

Music Understanding

- *Music Understanding*: Recognition of Pattern and Structure in Music
- Surface structure:
  - Pitch – Loudness
  - Harmony – Notes
- Deep structure:
  - Phrase relationships
  - Score following
  - Emotion
  - Expressive performance
Accompaniment Video

Computer Accompaniment

Performance

Score for Performer

Score for Accompaniment

Input Processing

Matching

Accompaniment Performance

Music Synthesis

Accompaniment
Vocal Accompaniment

- Lorin Grubb’s Ph.D. (CMU CSD)
- Machine learning used to:
  - Learns what kinds of tempo variation are likely
  - Characterize sensors
    - When is a notated G sensed as a G#?
- Machine learning necessary for good performance
How It Works

Score Position

Listening to Jazz Styles

Pointilistic

Lyrical

Frantic

Syncopated

© 2015 Roger B. Dannenberg
Oct 2015
Jazz Style Recognition

Onset Detection
Why?

- Beat Detection
- Tempo Detection
- Computer Accompaniment
- Music Transcription
  - Query-By-Humming
- Automatic Intelligent Audio Editor

Intelligent Audio Editor

- This excerpt is included in the audio examples:

Before: After:
Some Approaches

- Features and Thresholds
  - High Frequency
  - Phase Change
- Neural Networks
- Hierarchical Models
- HMM

A Bootstrap Method for Training an Accurate Audio Segmenter

Ning Hu and
Roger B. Dannenberg
Carnegie Mellon University
Introduction

- Audio segmentation is one of the major topics in MIR research:
  - HMM approach (Raphael, 1999)
  - Neural Network approach (Marolt, et al., 2002)
  - Support Vector Machine (Lu, et al. 2001)
  - Hierarchical Model (Kapanci and Pfeffer, 2004)

- In many cases, collecting training data is time-consuming and expensive.

Detour - Audio Alignment
Audio Alignment Concepts

- "Score"
  - Midi File, Note List, not necessarily "real" notation
- Similarity Matrix
- Chroma Vectors
- Distance/Similarity Function
- Research on accurate alignment

Chromagram Representation

- Spectrum
- Linear frequency to log frequency: "Semi vector": one bin per semitone
- Projection to pitch classes: "Chroma vector"
  - C₁+C₂+C₃+C₄+C₅+C₆+C₇,
  - C#₁+C#₂+C#₃+C#₄+C#₅+C#₆+C#₇, etc.
- "Distance Function": Euclidean, Cosine, etc.
Segmentation and Alignment

- Segmentation, audio alignment, and score-following are related
  - Rely on acoustic features
  - Precise alignment to symbolic score provides segmentation data
- We use alignment data to train a segmenter
  - Alignment avoids gross errors in segmentation
  - Segmenter learns fine-grain features that improve precision beyond initial alignment
- → high quality segmentation and alignment

Motivation

- We need very accurate segmentation to extract trumpet envelopes (attacks ~30ms)
  - (for research on capturing synthesis models)
- Alignment is based on chroma (100 – 250ms)
- Orio & Schwarz (2001) also use DTW and short-term features (5.8 ms windows), but alignment (an O(N^2) algorithm) is slow.
  - Our system performs alignment 25x faster.
- Our small non-DTW analysis windows can use different features.
Audio-to-(MIDI)-Score Alignment

- Chromagram features from Audio
- Synthetic chromagram features for MIDI

Acoustic Features for Segmentation – 5.8 ms window

- Log energy (dB)
- F0 with SNDAN’s (Beauchamp) MQ analysis
- Relative strengths of first 3 harmonics:
  - \( \text{Amplitude}_i / \text{Amplitude}_{\text{overall}} \)
- Relative frequency deviations, first 3 harmonics:
  - \( (f_i - i \times F0) / f_i \)
- Zero-crossing rate
- Derivatives of all of the above
Neural Network

Segment boundary PDF

- Gaussians
- On alignment boundaries
- Width based on alignment window size
- $P=0.04$ between boundaries
Bootstrap learning process

- Multiply neural net output by PDF
- For each neighborhood around a segment boundary, find the peak → “adjusted onset”
- Retrain the neural network:
  - adjusted onsets are 1, other points are 0

Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Miss Rate</th>
<th>Spurious Rate</th>
<th>Av. Error</th>
<th>STD</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNTHETIC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline Segmenter</td>
<td>8.8%</td>
<td>10.3%</td>
<td>21 ms</td>
<td>29 ms</td>
</tr>
<tr>
<td>Segmenter w/ Bootstrap</td>
<td>0.0%</td>
<td>0.3%</td>
<td>10 ms</td>
<td>14 ms</td>
</tr>
<tr>
<td>REAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline Segmenter</td>
<td>15.0%</td>
<td>25.0%</td>
<td>35 ms</td>
<td>48 ms</td>
</tr>
<tr>
<td>Segmenter w/ Bootstrap</td>
<td>2.0%</td>
<td>4.0%</td>
<td>8 ms</td>
<td>12 ms</td>
</tr>
</tbody>
</table>
Sound Examples

- Input

- Output – segmenter was trained on similar data using the bootstrap method. This input was segmented without using any score information.

Conclusions

- Supervised learning often wins over hand-crafted systems
- Segmentation training data is expensive, so supervised training is difficult
- Alignment provides strong hints, but not accurate enough for training
- Bootstrapping allows segmenter to generate its own training data
- Dramatic improvements in accuracy, even when tested without alignment “hints”
Summary

- Computer Accompaniment
- Offline Score Alignment
- Onset Detection