What can we Learn from Large Music Databases?

Dan Ellis
Laboratory for Recognition and Organization of Speech and Audio
Dept. Electrical Engineering, Columbia University, NY USA
dpwe@ee.columbia.edu

1. Learning Music
2. Music Similarity
3. Melody, Drums, Event extraction
4. Conclusions
Learning from Music

• **A lot** of music data available
  - e.g. 60G of MP3
    ≈ 1000 hr of audio/15k tracks

• **What can we do with it?**
  - implicit **definition** of ‘music’

• **Quality vs. quantity**
  - Speech recognition lesson:
    10x data, 1/10th annotation, **twice** as useful

• **Motivating Applications**
  - music similarity / classification
  - computer (assisted) music **generation**
  - insight into music
Ground Truth Data

- A lot of **unlabeled** music data available
  - manual annotation is much rarer
- **Unsupervised structure discovery** possible
  - .. but labels help to indicate what you want
- **Weak annotation sources**
  - artist-level descriptions
  - symbol sequences without timing (MIDI)
  - errorful transcripts
- **Evaluation requires ground truth**
  - limiting factor in Music IR evaluations?
Talk Roadmap

1. Anchor models
2. Melody extraction
3. Fragment clustering
4. Eigen-rhythms
5. Drums extraction
6. Event extraction

Semantics bases

Music audio

Similarity/recommend'\'n

Synthesis/generation

?
1. Music Similarity Browsing

• **Musical information overload**
  - record companies filter/categorize music
  - an automatic system would be less odious

• **Connecting audio and preference**
  - map to a ‘semantic space’?

![Diagram of music similarity browsing](attachment:image.png)
Anchor Space

- Frame-by-frame high-level categorizations
  - compare to raw features?
  - properties in distributions? dynamics?

Cepstral Features

Anchor Space Features
### Playola Similarity Browser

#### Artist: The Woodbury Muffin Outbreak

<table>
<thead>
<tr>
<th>Song Title</th>
<th>Artist</th>
<th>Time</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Ballad of Tabitha</td>
<td>The Woodbury Muffin Outbreak</td>
<td>4:00</td>
<td></td>
</tr>
<tr>
<td>Monkey Dreams</td>
<td>The Woodbury Muffin Outbreak</td>
<td>2:57</td>
<td></td>
</tr>
<tr>
<td>A Cold Dark Night (Live)</td>
<td>The Woodbury Muffin Outbreak</td>
<td>3:13</td>
<td></td>
</tr>
<tr>
<td>Leo, The Ballad of Tabitha</td>
<td>The Woodbury Muffin Outbreak</td>
<td>1:48</td>
<td></td>
</tr>
<tr>
<td>Baby I Forgot To Tell You</td>
<td>The Woodbury Muffin Outbreak</td>
<td>4:04</td>
<td></td>
</tr>
</tbody>
</table>

#### Music-Space Browser

<table>
<thead>
<tr>
<th>Feature</th>
<th>Less</th>
<th>More</th>
</tr>
</thead>
<tbody>
<tr>
<td>AltGRunge</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CollegeRock</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Country</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DanceRock</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Electronica</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MetalNPunk</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NewWave</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rap</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RnBSoul</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SingerSongwriter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SoftRock</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TradRock</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Female</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HiFi</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Similar Songs:

<table>
<thead>
<tr>
<th>Song Title</th>
<th>Artist</th>
<th>Distance</th>
<th>Good Match?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baby I Forgot To Tell You</td>
<td>The Woodbury Muffin Outbreak</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>Number five</td>
<td>Bizzi Chyld</td>
<td>0.07</td>
<td></td>
</tr>
<tr>
<td>Waiting for Your Love</td>
<td>Toto</td>
<td>0.08</td>
<td></td>
</tr>
<tr>
<td>Excerpt from 'CD'</td>
<td>Weirdomusic</td>
<td>0.08</td>
<td></td>
</tr>
</tbody>
</table>
Semantic Bases

- What should the ‘anchor’ dimensions be?
  - hand-chosen genres? X
  - somehow choose automatically

- “Community metadata”:
  Use Web to get words/phrases.
  - .. that are informative about artists
  - .. and that can be predicted from audio

- Refine classifiers to below artist level
  - e.g. by EM?

<table>
<thead>
<tr>
<th>adj Term</th>
<th>K-L bits</th>
<th>np Term</th>
<th>K-L bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>aggressive</td>
<td>0.0034</td>
<td>reverb</td>
<td>0.0064</td>
</tr>
<tr>
<td>softer</td>
<td>0.0030</td>
<td>the noise</td>
<td>0.0051</td>
</tr>
<tr>
<td>synthetic</td>
<td>0.0029</td>
<td>new wave</td>
<td>0.0039</td>
</tr>
<tr>
<td>punk</td>
<td>0.0024</td>
<td>elvis costello</td>
<td>0.0036</td>
</tr>
<tr>
<td>sleepy</td>
<td>0.0022</td>
<td>the mud</td>
<td>0.0032</td>
</tr>
<tr>
<td>funky</td>
<td>0.0020</td>
<td>his guitar</td>
<td>0.0029</td>
</tr>
<tr>
<td>noisy</td>
<td>0.0020</td>
<td>guitar bass and drums</td>
<td>0.0027</td>
</tr>
<tr>
<td>angular</td>
<td>0.0016</td>
<td>instrumentals</td>
<td>0.0021</td>
</tr>
<tr>
<td>acoustic</td>
<td>0.0015</td>
<td>melancholy</td>
<td>0.0020</td>
</tr>
<tr>
<td>romantic</td>
<td>0.0014</td>
<td>three chords</td>
<td>0.0019</td>
</tr>
</tbody>
</table>
2. **Transcription as Classification**

- **Signal models** typically used for transcription
  - harmonic spectrum, superposition
- **But ... trade domain knowledge for data**
  - transcription as pure classification problem:

  ![Diagram]

  - single N-way discrimination for “melody”
  - per-note classifiers for polyphonic transcription
Classifier Transcription Results

- Trained on MIDI syntheses (32 songs)
  - SMO SVM (Weka)
- Tested on ISMIR MIREX 2003 set
  - foreground/background separation

Frame-level pitch concordance

<table>
<thead>
<tr>
<th>system</th>
<th>“jazz3”</th>
<th>overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>fg+bg</td>
<td>71.5%</td>
<td>44.3%</td>
</tr>
<tr>
<td>just fg</td>
<td>56.1%</td>
<td>45.4%</td>
</tr>
</tbody>
</table>
Forced-Alignment of MIDI

- **MIDI** is a handy description of music
  - notes, instruments, tracks
  - .. to drive synthesis

- **Align MIDI ‘replicas’** to get GTruth for audio
  - estimate time-warp relation

"Don't you want me" (Human League), verse1

![Graph showing frequency vs. time for MIDI notes and replicas](image_url)
3. Melody Clustering

- **Goal:** Find ‘fragments’ that recur in melodies
  - across large music database
  - trade data for model sophistication

- **Data sources**
  - pitch tracker, or MIDI training data

- **Melody fragment representation**
  - $\text{DCT}(1:20)$ - removes average, smoothes detail
Melody clustering results

- Clusters match underlying contour:

- Finds some similarities:
  - e.g. Pink + Nsync
4. Eigenrhythms: Drum Pattern Space

- Pop songs built on repeating “drum loop”
  - variations on a few bass, snare, hi-hat patterns

- Eigen-analysis (or ...) to capture variations?
  - by analyzing lots of (MIDI) data, or from audio

- Applications
  - music categorization
  - “beat box” synthesis
  - insight
Aligning the Data

• Need to **align** patterns prior to modeling...

**Tempo** (stretch): by inferring BPM & normalizing

**Downbeat** (shift): correlate against ‘mean’ template
Eigenrhythms (PCA)

- Need 20+ Eigenvectors for good coverage of 100 training patterns (1200 dims)
- Eigenrhythms both add and subtract
Posirhythms (NMF)

- Nonnegative: only adds beat-weight
- Capturing some structure
Eigenrhythms for Classification

- **Projections in Eigenspace / LDA space**

PCA(1,2) projection (16% corr)

LDA(1,2) projection (33% corr)

- **10-way Genre classification (nearest nbr):**
  - PCA3: 20% correct
  - LDA4: 36% correct
Eigenrhythm BeatBox

• Resynthesize rhythms from eigen-space
5. Event Extraction

- **Music often contains many repeated events**
  - notes, drum sounds
  - but: usually overlapped...

- **Vector Quantization** finds common patterns:
  - representation...
  - aligning/matching...
  - how much coverage required?
Drum Track Extraction

• Initialize dictionary with Bass Drum, Snare
• Match only on a few spectral peaks
  ○ narrowband energy most likely to avoid overlap
• Median filter to re-estimate template
  ○ .. after normalizing amplitudes
  ○ can pick up partials from common notes
Generalized Event Detection

• Based on ‘Shazam’ audio fingerprints (Wang’03)

  - relative timing of $F_1 - F_2 - \Delta T$ triples discriminates pieces
  - narrowband features to avoid collision (again)

• Fingerprint events, not recordings:
  choose top triples, look for repeats

  - rank reduction of triples x time matrix
Event detection results

- **Procedure**
  - find hash triples
  - cluster them
  - patterns in hash co-occurrence = events?
Conclusions

- Lots of data
  + noisy transcription
  + weak clustering
⇒ musical insights?
Approaches to Chord Transcription

- **Note transcription**, then note→chord rules
  - like labeling chords in MIDI transcripts
- **Spectrum→chord rules**
  - i.e. find harmonic peaks, use knowledge of likely notes in each chord
- **Trained classifier**
  - don’t use any “expert knowledge”
  - instead, learn patterns from labeled examples
- **Train ASR HMMs with chords ≈ words**
Chord Sequence Data Sources

• All we need are the **chord sequences** for our training examples
  - Hal Leonard “**Paperback Song Series**”
    - manually retyped for 20 songs:
      “Beatles for Sale”, “Help”, “Hard Day’s Night”

- hand-align chords for 2 test examples
## Chord Results

- Recognition weak, but forced-alignment OK

### Frame-level Accuracy

<table>
<thead>
<tr>
<th>Feature</th>
<th>Recognition</th>
<th>Alignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC</td>
<td>8.7%</td>
<td>22.0%</td>
</tr>
<tr>
<td>PCP_ROT</td>
<td>21.7%</td>
<td>76.0%</td>
</tr>
</tbody>
</table>

*MFCCs are poor (can overtrain)*

PCPs better *(ROT helps generalization)*

---

[Diagram of Beatles - Beatles For Sale - Eight Days a Week (4096pt)]

Learning from Music - Ellis

2004-12-18  p. 27/24
What did the models learn?

- Chord model centers (means) indicate chord ‘templates’: