Instrument Recognition

Maddy Cappelloni, Autumn Coe
Overview

● Applications and Context

● Methods
  ○ HMM
  ○ Source Filter Model
  ○ Cepstral Coefficients
  ○ Neural Networks

● Future Research Directions
Applications and human ability

- Label music for search by instrument
- Aid in genre recognition
- Inform music transcription

---

**TABLE I.** Summary of percent correct for previous human perception experiments on wind instruments. Results for the oboe, sax, clarinet, and flute are given when possible. The final column is the total number of instruments included in the experiment.

<table>
<thead>
<tr>
<th></th>
<th>Date</th>
<th>Oboe</th>
<th>Sax</th>
<th>Clar</th>
<th>Flute</th>
<th>Overall</th>
<th>Number of instruments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eagleson/Eagleson</td>
<td>1947</td>
<td>59</td>
<td>45</td>
<td>20</td>
<td></td>
<td>56</td>
<td>9</td>
</tr>
<tr>
<td>Saldanha/Corso</td>
<td>1964</td>
<td>75</td>
<td>84</td>
<td>61</td>
<td></td>
<td>41</td>
<td>10</td>
</tr>
<tr>
<td>Berger</td>
<td>1964</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>59</td>
<td>10</td>
</tr>
<tr>
<td>Clark/Milner</td>
<td>1964</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>90</td>
<td>3 (flute, clar, oboe)</td>
</tr>
<tr>
<td>Strong/Clark</td>
<td>1967a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>85</td>
<td>8</td>
</tr>
<tr>
<td>Campbell/Heller</td>
<td>1978</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>72</td>
<td>6 (2-note legato)</td>
</tr>
<tr>
<td>Kendall</td>
<td>1986</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>84</td>
<td>3 (trumpet, clar, violin)</td>
</tr>
<tr>
<td>Brown</td>
<td>1999</td>
<td>85</td>
<td>92</td>
<td></td>
<td></td>
<td>89</td>
<td>2 (oboe, sax)</td>
</tr>
<tr>
<td>Martin</td>
<td>1999</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>46</td>
<td>27 (isolated tone)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>67</td>
<td>27 (10-s excerpt)</td>
</tr>
<tr>
<td>Houix/McAdams/Brown</td>
<td></td>
<td>87</td>
<td>87</td>
<td>71</td>
<td>93</td>
<td>85</td>
<td>4 (oboe, sax, clar, flute)</td>
</tr>
</tbody>
</table>

Brown et al, 2001
Hidden Markov Models

- Test which instrument-specific HMM best explains observations

\[ \hat{i} = \arg\max_i P(Y | \lambda_i) \]

where \( i \) is the instrument, \( \lambda_i \) are the parameters for each instrument, and \( Y \) are the observations

- Choose model states
- Choose audio features
Model states

- **Attack, Decay, Sustain, Release**
  - Relative timing depends on the instrument
  - Lee et al, 2002

- **Ergodic model**
  - N: silence
  - A: attack
  - S1, S2, S3: vibrato
  - Lee et al, 2002

- **Arbitrary Topology**
  - Eichner et al, 2006
Audio features

Generally related to timbre of the instrument

- Amplitude of harmonics
  - Lee et al, 2002
- Interpolated spectral envelope
  - Lee et al, 2002
- Cepstral coefficients and their derivatives
Pros/Cons of HMM

Pros

- Performance approaches human capabilities

Cons

- Performance drops quickly with increasing number of instruments
- Observations may be very similar across instruments
  - Use ICA to maximize statistical independence
  - Discriminative training (maximum mutual information)
  - Eronen et al, 2006
Source-Filter Model

- Each instrument has its own spectral envelope shape and can be modeled as a note source (excitation spectrum) with a filter specific to the instrument.

\[ \hat{x}_t(k) = \sum_{n=1}^{N} \sum_{i=1}^{I} g_{n,i,t} e_{n,t}(k) h_i(k) \]

- Magnitude of STFT for frame \( t \)
- Notes
- Instruments
- Gain of each instrument playing each note in the frame
- Excitation spectra for notes
- Filters for instruments (linear combination of elementary responses)

Heittola et al, 2009
Fitting the model

- Do multipitch estimation

- Construct filter responses as a \textit{sum} of triangular bandpass filters with instrument specific weights

- Excitation spectra are harmonic combs with amplitude 1

Heittola et al, 2009
Fitting the model

- Fit filter weights and gains for each instrument/note combination by doing NMF with multiplicative update
- For each frame
  - Choose an excitation spectrum based on the multipitch estimation
  - Update filter weights
  - Update gains
  - Iterate until reconstruction error sufficiently small
Classification

- Test likelihood of signal for each instrument class
  - Likelihoods accumulate over the frames
  - Weight by energy (RMS of signal)
  - Classification by maximum likelihood
- Calculate F-score to evaluate
  - Weighted harmonic mean of precision and recall (more detailed than reporting just accuracy)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>no separation</td>
<td>62.0</td>
<td>18.7</td>
<td>12.1</td>
<td>13.7</td>
<td>24.4</td>
<td>29.5</td>
</tr>
<tr>
<td>no streaming</td>
<td>62.0</td>
<td>49.5</td>
<td>42.1</td>
<td>42.6</td>
<td>39.3</td>
<td>42.7</td>
</tr>
<tr>
<td>streaming (given F0s)</td>
<td>62.0</td>
<td>59.0</td>
<td>58.0</td>
<td>57.9</td>
<td>57.8</td>
<td>56.0</td>
</tr>
<tr>
<td>streaming (est. F0s)</td>
<td>61.0</td>
<td>60.2</td>
<td>53.5</td>
<td>56.7</td>
<td>55.2</td>
<td>53.8</td>
</tr>
<tr>
<td>streaming (timbre)</td>
<td>62.0</td>
<td>57.6</td>
<td>51.9</td>
<td>57.0</td>
<td>55.9</td>
<td>59.1</td>
</tr>
</tbody>
</table>

Heittola et al, 2009
Pros/Cons of Source-Filter Model

Pros

● Performance is not significantly reduced for polyphonic music

Cons

● Will not work well if multiple instruments are playing the same note
● Assumes all instruments are playing in every frame
● Assumes harmonicity
Cepstral Coefficients

- Cepstrum: Inverse Discrete Fourier Transform (IDFT) of the log magnitude of the Fourier transform of the signal

\[ C_{\log}(x) = \mathcal{F}^{-1}(\log |\mathcal{F}(x)|) \]

- Can use a power transformation instead of a log. (Yu et al, 2014)

\[ C(x) = \mathcal{F}^{-1}(g(\mathcal{F}(x))) \]

\[ g(x) = |x|^{1/2}, |x|^{1/3}, |x|^{1/4}, |x|^{1/5} \]

- Can use a Discrete Cosine Transform (DCT) instead of inverse Fourier transform
Sparse Cepstral Codes

- Identifies predominant instrument
- Input Cepstrum is calculated using a log function, power functions, or MFCC.
- Dictionary is trained using a joint optimization problem

\[
\hat{D} = \arg \min_{D} \frac{1}{N} \sum_{i=1}^{N} \left( \frac{1}{2} \| y_i - D \alpha_i \|_2^2 + \lambda \| \alpha_i \|_1 \right)
\]

- Sparse representation is calculated using trained dictionary and Least Angle Regression LASSO algorithm

\[
\hat{\alpha} = \arg \min_{\alpha} \| y - D \alpha \|_2^2 + \lambda \| \alpha \|_1
\]

Yu et al, 2014
Sparse Cepstral Codes Continued

- The more granular the dictionary, the better the result
- Power transforms perform better for polyphony than the traditional log transform

<table>
<thead>
<tr>
<th>$k$</th>
<th>$f(\mathcal{F})$</th>
<th>$f(C_{\log})$</th>
<th>$f(C_2)$</th>
<th>$f(C_3)$</th>
<th>$f(C_4)$</th>
<th>$f(C_5)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>0.818</td>
<td>0.833</td>
<td>0.910</td>
<td>0.892</td>
<td>0.919</td>
<td>0.916</td>
</tr>
<tr>
<td>128</td>
<td>0.873</td>
<td>0.875</td>
<td>0.926</td>
<td>0.924</td>
<td>0.918</td>
<td>0.921</td>
</tr>
<tr>
<td>256</td>
<td>0.902</td>
<td>0.920</td>
<td>0.936</td>
<td>0.935</td>
<td>0.936</td>
<td>0.938</td>
</tr>
<tr>
<td>512</td>
<td>0.907</td>
<td>0.926</td>
<td>0.947</td>
<td>0.942</td>
<td>0.944</td>
<td>0.947</td>
</tr>
<tr>
<td>1024</td>
<td>0.928</td>
<td>0.944</td>
<td>0.939</td>
<td>0.939</td>
<td>0.946</td>
<td>0.955</td>
</tr>
<tr>
<td>$\Delta$</td>
<td>0.110</td>
<td>0.111</td>
<td>0.029</td>
<td>0.047</td>
<td>0.027</td>
<td>0.039</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task</th>
<th>existing work</th>
<th>low-level feature</th>
<th>SC feature</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[17] [18]</td>
<td>MFCC $\mathcal{F}$</td>
<td>$C_{\log}$</td>
<td>$f(\mathcal{F})$</td>
</tr>
<tr>
<td>Uni-source (ParisTech)</td>
<td>0.820 – 0.541</td>
<td>0.684 0.705</td>
<td>0.934 0.945</td>
<td>0.941 0.939</td>
</tr>
<tr>
<td>Multi-source (MTG)</td>
<td>– 0.630</td>
<td>0.229 0.256</td>
<td>0.276</td>
<td>0.611 0.610</td>
</tr>
</tbody>
</table>
Uniform Discrete Cepstrum

\[ a = Mc \]

\[ c_{oc} = (M^T M)^{-1} M^T a = \frac{1}{N} M^T a \]

\[ \hat{a} = \hat{M} c \]

\[ c_{dc} = (\hat{M}^T \hat{M})^{-1} \hat{M}^T \hat{a} \]

\[ c_{udc} = \hat{M}^T \hat{a} = M^T \tilde{a} = N(M^T M)^{-1} M^T \tilde{a} \]

Duan et al, 2014
Unified Discrete Cepstrum Continued

- 13-Class SVM trained on 5 single frames of single notes from 13 instruments
- Tested on randomly generated chords from notes in a different data set

Duan et al, 2014
Pros/Cons of Cepstral Coefficients

Pros

- Can be used on polyphony without source separation

Cons

- Does not work well for unison or octave notes
- Requires isolated instrument note training data
Neural Networks

- Convolutional Neural Networks (CNNs) can be seen as a trainable feature extractor coupled with a learning model. (Li et al, 2015)
- Feature identification is done by the neural network instead of the designing engineer.
- No longer requires musical domain knowledge
Comparing Two CNNs

Li et al, 2014

- Input is raw audio
- 11 instrument classes
- Trained on audio clips with multiple instruments

Han et al, 2014

- Input is mel spectrogram
- 11 instruments
- Trained on audio clips chosen to contain a continuous presence of a single predominant instrument

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Input size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convolution Feature Maps 1</td>
<td>256</td>
<td>mel-spectrogram</td>
</tr>
<tr>
<td>Convolution Filter Size 1</td>
<td>3101</td>
<td>3 × 3 convolution, 32 filters</td>
</tr>
<tr>
<td>Maxpooling Stride Size 1</td>
<td>20</td>
<td>3 × 3 convolution, 32 filters 3 × 3 max-pooling</td>
</tr>
<tr>
<td>Maxpooling Size 1</td>
<td>40</td>
<td>dropout (0.25)</td>
</tr>
<tr>
<td>Convolution Feature Maps 2</td>
<td>384</td>
<td>3 × 3 convolution, 64 filters</td>
</tr>
<tr>
<td>Convolution Filter Size 2</td>
<td>300</td>
<td>3 × 3 convolution, 64 filters</td>
</tr>
<tr>
<td>Maxpooling Size 2</td>
<td>30</td>
<td>3 × 3 max-pooling dropout (0.25)</td>
</tr>
<tr>
<td>Maxpooling Stride Size 2</td>
<td>20</td>
<td>3 × 3 convolution, 128 filters</td>
</tr>
<tr>
<td>Convolution Feature Maps 3</td>
<td>384</td>
<td>3 × 3 convolution, 128 filters</td>
</tr>
<tr>
<td>Convolution Filter Size 3</td>
<td>20</td>
<td>3 × 3 convolution, 128 filters</td>
</tr>
<tr>
<td>Maxpooling Size 3</td>
<td>8</td>
<td>3 × 3 max-pooling dropout (0.25)</td>
</tr>
<tr>
<td>Maxpooling Stride Size 3</td>
<td>4</td>
<td>3 × 3 convolution, 256 filters</td>
</tr>
<tr>
<td>Layer 4 Output Size</td>
<td>400</td>
<td>global max-pooling flattened and fully connected</td>
</tr>
<tr>
<td>Final Output Size</td>
<td>11</td>
<td>dropout (0.50)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sigmoid</td>
</tr>
</tbody>
</table>
### Convolutional Neural Network Results

<table>
<thead>
<tr>
<th>Models</th>
<th>Accuracy</th>
<th>Exact Match</th>
<th>Precision</th>
<th>Recall</th>
<th>F-micro</th>
<th>F-macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio + CNN</td>
<td>82.74%</td>
<td>25.78%</td>
<td>0.7560</td>
<td>0.6888</td>
<td>0.7208</td>
<td>0.6433</td>
</tr>
</tbody>
</table>

- **Li et al (above)**
- **Han et al (right)**
- **Han et al single instrument breakdown (below)**
Pros/Cons of Neural Networks

Pros

- Training data can be less structured
- Does not require extensive domain knowledge

Cons

- Time consuming to train
- Requires extensive computing power
- Requires large amounts of data
Future Directions

Continue expanding research on neural network applications

Crowd source data labeling to increase and improve training/testing data sets

Begin applying musical domain knowledge to neural networks
Questions?
References


