DARK 2

Intelligent Audio Systems:
A review of the foundations and applications of
semantic audio analysis and music information retrieval
These lecture notes contain hyperlinks to the CCRMA Wiki.

On these pages, you can find supplemental material for lectures - providing extra tutorials, support, references for further reading, or demonstration code snippets for those interested in a given topic.

Click on the symbol on the lower-left corner of a slide to access additional resources.

WIKI REFERENCES...
Review from Day 1

- What are the 3 major components of a MIR system?
- Name 3 ways of segmenting audio into frames
- What problems did you experience in the lab?
- Follow-up questions?
- Did you try other audio files?
- Did you do the simple instrument recognition?
FEATURE DEMOS

• Simple re-ordering or slices:
  – Slice up loop into segments and sort via features
  – Play audio
  – Play whole song snippet
Basic system overview

Segmentation
(Frames, Onsets, Beats, Bars, Chord Changes, etc)

Feature Extraction
(Time-based, spectral energy, MFCC, etc)

Analysis / Decision Making
(Classification, Clustering, etc)
FEATURE EXTRACTION
Temporal Information

- Rise time or Attack time - time interval between the onset and instant of maximal amplitude
- Attack slope

Picture courtesy: Olivier Lartillot
Temporal Information

• Temporal Centroid
# Features – Frame 1

<table>
<thead>
<tr>
<th>Frame</th>
<th>ZCR</th>
<th>Centroid</th>
<th>BW</th>
<th>Skew</th>
<th>Kurtosis</th>
<th>E1</th>
<th>E2</th>
<th>E3</th>
<th>E4</th>
<th>E5</th>
<th>E6</th>
<th>E7</th>
<th>E8</th>
<th>E9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
<td>2.8kHz</td>
<td>5kHz</td>
<td>2.2</td>
<td>6.7</td>
<td>4000</td>
<td>10100</td>
<td>545</td>
<td>187</td>
<td>77</td>
<td>35</td>
<td>18</td>
<td>9</td>
<td>6</td>
</tr>
</tbody>
</table>
Frame 2

Energy

Octave

kick

24.8 32.8 5,308.1 1,366.4 360.4 180.2 194.5 68.6 5.3

1 2 3 4 5 6 7 8 9
Features : SimpleLoop.wav

<table>
<thead>
<tr>
<th>Frame</th>
<th>ZCR</th>
<th>Centroid</th>
<th>BW</th>
<th>Skew</th>
<th>Kurtosis</th>
<th>E1</th>
<th>E2</th>
<th>E3</th>
<th>E4</th>
<th>E5</th>
<th>E6</th>
<th>E7</th>
<th>E8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
<td>2.8kHz</td>
<td>5kHz</td>
<td>2.2</td>
<td>6.7</td>
<td>4000</td>
<td>10100</td>
<td>545</td>
<td>187</td>
<td>77</td>
<td>35</td>
<td>18</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>423</td>
<td>3.1kHz</td>
<td>4kHz</td>
<td>2</td>
<td>7.2</td>
<td>24</td>
<td>33</td>
<td>5300</td>
<td>1366</td>
<td>360</td>
<td>180</td>
<td>194</td>
<td>68</td>
</tr>
</tbody>
</table>
MFCCs

The idea of MFCCs is to capture spectrum in accordance with human perception.

1. STFT
2. log(STFT)
3. Perform mel-scaling to group and smooth coefficients. (perceptual weighting)
4. Decorrelate with DCT

[...continued...]
Weighting for FFT bins to Mel scale

MFCC

guitar.wav

FFT magnitudes

frame

ceps = DCT of mel filter output
MFCC of Music
(Petruncio, 2003)

Piano

Saxophone

Tenor Opera Singer

Drums
Features: Measuring changes

- Δ and Δ Δ
  - Change between frames
  - How quickly the change is occurring

- Spectral flux is the distance between the spectrum of successive frames
Spectral Features

- Spectral Flatness Measure
- Spectral Crest Factor
- Spectral Flux
Feature extraction

- Feature design and creation uses one’s domain knowledge.
- Choosing discriminating features is critical.
- Smaller feature space yields smaller, simpler models, faster training, often less training data needed.
Spectral Bands
Log Spectrogram
Chroma Bins
Example

Chromagram

chroma class
B    A#    A    G#    G    F#    F    E    D#    D    C#

0   0.5   1   1.5   2   2.5   3   3.5   4

time axis (in s.)

Picture courtesy: Olivier Lartillot
The resulting graph indicates the cross-correlation score for each different tonality candidate.
Decision stumps

- An example dataset:

This section contains slides adapted from Rob Schapire @ Princeton.
A decision threshold

- Single threshold: e.g., “output ‘+’ iff $x < .2$”

- Decision stump: 1 threshold decision
Many thresholds: Decision trees

• Consists of many decisions in succession (like a flowchart)
• General approach:
  – Recursively split training data into subsets based on simple thresholds
  – Optionally prune to avoid overfitting
• Common algorithms: CART, ID₃ => C₄.₅ (J₄₈)
Decision Trees

• Advantages:
  – Easy to interpret
  – Decision boundary is explicit and straightforward

• Disadvantages:
  – Can take a long time to learn
    • Finding optimal tree can be NP-complete
  – Prone to overfitting
  – Inherently heuristic
  – Slight perturbations of data can lead to very different trees
Boosting

- A “meta-algorithm” for creating a “strong” learner from many “weak” learners
- Iteratively train weak learners on variations of the dataset and combine in a principled way to produce classification outputs.
AdaBoost

- A popular boosting algorithm from Freund and Schapire
- Robust to overfitting: emphasis on **maximizing the margin**
Back to stumps

• Single threshold: e.g., “output ‘+’ iff $x < .2$”

• Makes a nice weak learner!
The AdaBoost algorithm

- Initialize $D_1$ to be the dataset with each example equally weighted.
- for round $t$ in 1 to $T$:
  - Train a weak learner, $h_t$, on the dataset $D_t$
  - If $h_t$ can’t achieve 50% accuracy, stop.
  - Choose $\alpha_t$ according to error rate of $h_t$ on $D_t$ (better $h_t$ => higher $\alpha_t$)
  - Update data weights $D_{t+1}$ to **increase** weight of examples $h_t$ got wrong, and **decrease** weight of examples $h_t$ got right.
- To classify new data, take a weighted majority vote of all weak learners, each $h_t$ weighted by its $\alpha_t$. 
AdaBoost illustrated

- Initial data:
Round 1

\[ h_1 \]

\[ D_2 \]

\[ \varepsilon_1 = 0.30 \]

\[ \alpha_1 = 0.42 \]
Round 2

\[ \varepsilon_2 = 0.21 \]
\[ \alpha_2 = 0.65 \]
Round 3

\[ \varepsilon_3 = 0.14 \]
\[ \alpha_3 = 0.92 \]
$H_{\text{final}} = \text{sign}(0.42 + 0.65 + 0.92)$
Final classifier: decision boundary
A typical AdaBoost run

- Test error does not increase, even after 1000 rounds
- Test error continues to drop, even after training error = 0.
The margin

- Narrow margin
- Wide margin

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>+</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+</td>
</tr>
</tbody>
</table>
Margin distribution after N rounds:

<table>
<thead>
<tr>
<th># rounds</th>
<th>5</th>
<th>100</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>train error</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>test error</td>
<td>8.4</td>
<td>3.3</td>
<td>3.1</td>
</tr>
<tr>
<td>% margins $\leq 0.5$</td>
<td>7.7</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>minimum margin</td>
<td>0.14</td>
<td>0.52</td>
<td>0.55</td>
</tr>
</tbody>
</table>
AdaBoost pro & con

• Advantages:
  – Robust to overfitting
  – Conceptually simple
  – Statistically very nice: maximizing the margin, game-theoretic understanding
  – Can work with any base learner
  – No parameters to tune

• Disadvantages:
  – Weak learner must achieve >50% or failure
  – Original formulation binary only
    • AdaBoost.M1 handles multi-class, but more required of weak learner
EVALUATION
Our classifier accuracy is 83.4%
Cross-validation

• Say, 10-fold cross validation
• Divide test set into 10 random subsets.
• 1 test set is tested using the classifier trained on the remaining 9.
• We then do test/train on all of the other sets and average the percentages. Helps prevent over fitting.
• Do not optimize too much on cross validation – you can severely overfit. Sanity check with a test set.
Cross-validation
Cross-validation

Fold 1: 70%
Cross-validation

Fold 1: 70%
Fold 2: 80%
Cross-validation

Fold 1: 76%
Fold 2: 80%
Fold 3: 77%
Fold 4: 83%
Fold 5: 72%
Fold 6: 82%
Fold 7: 81%
Fold 8: 71%
Fold 9: 90%
Fold 10: 82%

Mean = 79.4%
Stratified Cross-Validation

- Same as cross-validation, except that the folds are chosen so that they contain equal proportions of labels.
> End Day 2