Part 1: A Frame-Level Speech/Music Discrimination using AdaBoost

Norman Casagrande
Douglas Eck
Balázs Kégl
Part 2: Audio Genre Recognition – the winner of the MIREX 2005 competition

James Bergstra
Norman Casagrande
Douglas Eck
Balázs Kégl
Introduction

- Does voice have a **visible** distinguishable pattern?
The Features

- Using a robust image classifier by Viola & Jones (2001) we can quickly classify music against speech on real world data
The Features

- Using a robust image classifier by Viola & Jones (2001) we can quickly classify music against speech on real world data

- Find simple basic features that computes the difference of amount of energy

| Area B | Area W | Or | Area B | Area W | Area B |
The Features
The Features
The Features

\[ B_1 \cdot W + B_2 = x_i \]
The Features

- How do we find the position of the filter?
The Features

- How do we find the position of the filter?
- The filters have also different sizes!
The Features

- The output of this simple filter is already revealing a separation between speech and music.
The Features

- The output of this simple filter is already revealing a separation between speech and music.
- Accuracy \( \sim 68\% \)
- Still too weak alone
AdaBoost

- Successful general purpose learning method
- At each iteration $t$ a weak learner $h$ returns a binary prediction with error epsilon.

Strong learner

Weak learners

Input

$$D_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$$
AdaBoost

- Set a weight distribution $w^t = (w_1^t, ..., w_n^t)$ over the data points
AdaBoost

- Set a weight distribution $w^t = (w_1^t, \ldots, w_n^t)$ over the data points
- For $1 \ldots T$
  - Find $h^t$ by minimizing the weighted error
    $$\epsilon = \sum_{i=1}^{n} I_{\left\{ h^t(x_i) \neq y_i \right\}} w_i^t$$
    over the features parameters
AdaBoost

- Set a weight distribution \( \mathbf{w}^t = (w_1^t, \ldots, w_n^t) \) over the data points

- For 1 \( \ldots \) T
  - Find \( h^t \) by minimizing the weighted error
    \[
    \epsilon = \sum_{i=1}^{n} I_{\{h^t(x_i) \neq y_i\}} w_i^t
    \]
  - over the features parameters
AdaBoost

- Set a weight distribution $w^t = (w_1^t, \ldots, w_n^t)$ over the data points
- For $1 \ldots T$
  - Find $h^t$ by minimizing the weighted error
    \[ \epsilon = \sum_{i=1}^{n} I\{h^t(x_i) \neq y_i\} w_i^t \] 
    over the features parameters
AdaBoost

- Set a weight distribution \( \mathbf{w}^t = (w_1^t, \ldots, w_n^t) \) over the data points

- For 1 ... T
  - Find \( h^t \) by minimizing the weighted error
    \[
    \epsilon = \sum_{i=1}^{n} I_{\{h^t(x_i) \neq y_i\}} w_i^t \]
    over the features parameters
AdaBoost

- Set a weight distribution $w^t = (w_1^t, \ldots, w_n^t)$ over the data points

- For 1 … T
  - Find $h^t$ by minimizing the weighted error

$$
\epsilon = \sum_{i=1}^{n} I_{h^t(x_i) \neq y_i} w_i^t
$$
over the features parameters
AdaBoost

- Set a weight distribution \( w^t = (w_1^t, \ldots, w_n^t) \) over the data points
- For 1 … T
  - Find \( h^t \) by minimizing the weighted error
    \[
    \epsilon = \sum_{i=1}^{n} I_{\{h^t(x_i) \neq y_i\}} w_i^t
    \]
    over the features parameters
  - Compute the confidence
    \[
    \alpha^t = \frac{1}{2} \ln \left( \frac{1 - \epsilon^t}{\epsilon^t} \right)
    \]
AdaBoost

- Set a weight distribution \( w^t = (w_1^t, \ldots, w_n^t) \) over the data points
- For 1 \( \ldots \) T
  - Find \( h^t \) by minimizing the weighted error
    \[
    \epsilon = \sum_{i=1}^{n} I_{\{h^t(x_i) \neq y_i\}} w_i^t
    \]
    over the features parameters
  - Compute the confidence
    \[
    \alpha^t = \frac{1}{2} \ln \left( \frac{1-\epsilon^t}{\epsilon^t} \right)
    \]
  - Update weight vector \( w \)
    \[
    w_i^{t+1} = w_i^t \times \begin{cases} 
    \frac{1}{2(1-\epsilon^t)} & \text{if } h^t(x_i) = y_i \\
    \frac{1}{2 \epsilon^t} & \text{if } h^t(x_i) \neq y_i
    \end{cases}
    \]
AdaBoost

- Set a weight distribution \( w^t = (w_1^t, \ldots, w_n^t) \) over the data points
- For 1 \( \ldots \) T
  - Find \( h^t \) by minimizing the weighted error
    \[
    \epsilon = \sum_{i=1}^{n} I_{\{h^t(x_i) \neq y_i\}} w_i^t
    \]
    over the features parameters
  - Compute the confidence \( \alpha^t = \frac{1}{2} \ln \left( \frac{1 - \epsilon^t}{\epsilon^t} \right) \)
  - Update weight vector \( w \)
- Output the final strong learner

\[
 f(x) = \text{sign} \left( \sum_{t=1}^{T} \alpha^t h^t(x) \right)
\]
The amount of energy of the filter can be computed in constant time
Optimization

- The convolution of the filter can be computed in constant time

![Graph showing amplitude vs. frequency with integral image representation]
Optimization

- A comprehensive search in the parameter space can be avoided

- **Observation:**
  Slight change in the parameters do not change the error significantly
Optimization

- A set of random starting point is chosen, then a discrete gradient descent is performed.

- The overall performance is equivalent at the cost of few more iterations.
Optimization

- The performance can be increased by a **simple smoothing** on the output of the previous $k$ frames

**Observation:**
If in the last $k$ frames there has been speech (or music), it is **highly probable** that current frame $\tau$ will be speech (or music) too.

\[
g(x_\tau) = \frac{\sum_{i=\tau-k}^{\tau} a^{\tau-i} f(x_i)}{\sum_{j=\tau-k}^{\tau} a^{\tau-j}}
\]
Results

- Dataset of real world radio transmission with music, talks, jingles, etc. used by Scheirer and Slaney (1997)

- 11200 frames of normalized and processed with 20 ms STFT at a resolution of 256 points.
Results

- The error reaches a plateau after ~150 iterations/filters
- At frame level already the error get to ~12%
- With the smoothing the error drops to 6.7%
### Results

- Better than typical **frame level** features

<table>
<thead>
<tr>
<th>Feature</th>
<th>CPU</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rolloff</td>
<td>17%</td>
<td>~46 ±3 %</td>
</tr>
<tr>
<td>Spec. Cent.</td>
<td>17%</td>
<td>~39 ±8 %</td>
</tr>
<tr>
<td>Spec. Flux</td>
<td>17%</td>
<td>~39 ±1 %</td>
</tr>
<tr>
<td>ZCR</td>
<td>0%</td>
<td>~38 ±4 %</td>
</tr>
<tr>
<td>Ceps Resid</td>
<td>46%</td>
<td>~37 ±7 %</td>
</tr>
<tr>
<td><strong>Proposed</strong></td>
<td>&lt;1%</td>
<td>~12 ±2 %</td>
</tr>
</tbody>
</table>
Real-time implementation as Winamp plugin available at: www.iro.umontreal.ca/~casagran/winamp
Real-time implementation as **Winamp plugin** available at: www.iro.umontreal.ca/~casagran/winamp
Conclusions

- Best suited in an *ensemble* algorithm among other features
- Alone it is already capable of good performance
- Extremely fast during detection
Conclusions

- Best suited in an *ensemble* algorithm among other features
- Alone it is already capable of good performance
- Extremely fast during detection
- Can learn *any* type of pattern
  - The patterns do not need to be limited to one frame!
Conclusions
Conclusions
Conclusions

- Best suited in an *ensemble* algorithm among other features
- Alone it is already capable of good performance
- Extremely fast during detection
- Can learn *any* type of pattern
  - The patterns do not need to be limited to one frame!
- Simple and easy implementation
End Part 1

- Thank you!
  - Questions?
Part 2 – Audio Genre Recognition

- Blues
- Classical
- Country
- Disco
- Hiphop
- Jazz
- Metal
- Pop
- Reggae
- Rock

**DB** → audio features

.wav/.mp3
Audio Features: why?

- Signal audio is a real-valued vector
- Why not classify it directly?
  - Very high-dimensional vector
Audio Features: why?

- Signal audio is a real-valued vector
- Why not classify it directly?
  - Very high-dimensional vector
  - classification should have shift invariance
Audio Features: why?

- Signal audio is a real-valued vector
- Why not classify it directly?
  - Very high-dimensional vector
  - classification should have shift invariance
  - small differences in spectral magnitude are...
    - important for small magnitudes (quiet frequencies)
    - not important for large magnitudes (loud frequencies)
  - global frequency scaling is irrelevant
  - global frequency shifting is highly relevant
Audio Features: essence of music?

- **Timbre** – almost-instant sound quality

- **Rhythm** – repeated sound structure over a few seconds
Audio Features: essence of music?

- Timbre – almost-instant sound quality
  - cepstral coefficients: real(fft(log(T abs(fft(s)))) )
    - rceps: $T = I$
    - mfcc: $T$ implements a Mel-scale projection
    - other options for $T$ are Bark-scale and log-scale.

- Rhythm – repeated sound structure over a few seconds
Audio Features: essence of music?

- **Timbre** – almost-instant sound quality
  - cepstral coefficients: real(fft(log(T abs(fft(s)))))
    - rceps: T = I
    - mfcc: T implements a Mel-scale projection
    - other options for T are Bark-scale and log-scale.
  - zero-crossing rate

- **Rhythm** – repeated sound structure over a few seconds
Audio Features: essence of music?

- **Timbre** – almost-instant sound quality
  - cepstral coefficients: \( \text{real}(\text{fft}(\log( T \text{abs}(\text{fft}(s)))))) \)
    - rceps: \( T = 1 \)
    - mfcc: \( T \) implements a Mel-scale projection
    - other options for \( T \) are Bark-scale and log-scale.
  - zero-crossing rate
  - spectral centroid and flatness
    - \( E[] \) and \( \text{Var}[] \) of normalized FFT

- **Rhythm** – repeated sound structure over a few seconds
Audio Features: essence of music?

- Timbre – almost-instant sound quality
  - cepstral coefficients: real(fft (log( T abs(fft(s))) ) )
    - rceps: T = I
    - mfcc: T implements a Mel-scale projection
  - other options for T are Bark-scale and log-scale.
- zero-crossing rate
- spectral centroid and flatness
  - E[] and Var[] of normalized FFT
- linear predictive coefficients & reconstruction error

- Rhythm – repeated sound structure over a few seconds
Audio Features : essence of music?

- Timbre – almost-instant sound quality
  - cepstral coefficients: real(fft (log( T abs(fft(s)) ) ) )
    - rceps: T = 1
    - mfcc: T implements a Mel-scale projection
    - other options for T are Bark-scale and log-scale.
  - zero-crossing rate
  - spectral centroid and flatness
    - E[] and Var[] of normalized FFT
  - linear predictive coefficients & reconstruction error

- Rhythm – repeated sound structure over a few seconds
  - important, interesting, but not used at MIREX
Audio Features : for AdaBoost

- Segment Feature Extraction
Audio Features: for AdaBoost

- Segment Feature Extraction
  - input: a signal of (1024 * c) samples
  - define $s(j)$ to be the sub-signal of length 1024 starting at $j*1024$
  - define $r(j) = (\text{mfcc}(s(j)), \text{rceps}(s(j)), \text{lpc}(s(j)), \text{zcr}(s(j)), \text{ro}(s(j)), \text{fftc}(s(j)) )$

\[
\tilde{r}(j) = x_0, x_1, ..., x_{62}, x_{63}, ..., x_{369}, x_{370}, x_{371}, ..., x_{401}
\]
**Audio Features : for AdaBoost**

- **Segment Feature Extraction**
  - input: a signal of \((1024 \times c)\) samples
  - define \(s(j)\) to be the sub-signal of length 1024 starting at \(j\times1024\)
  - define \(r(j) = (\text{mfcc}(s(j)), \text{rceps}(s(j)), \text{lpc}(s(j)), \text{zcr}(s(j)), \text{ro}(s(j)), \text{fftc}(s(j)) )\)
  - return \((E[r(J)], \text{Var}[r(J)]\) for \(J\) uniform on \([0,c-1]\)
  - relative ordering of sub-signals is ignored

\[
\vec{r}(j) = x_0, x_1, \ldots, x_{62}, x_{63}, \ldots, x_{369}, x_{370}, x_{371}, \ldots, x_{401}
\]
Data

Song
Data
Data
Data

+ + + + ...

... =

winner
Algorithm

- We used AdaBoost.MH to classify each window
We used AdaBoost.MH (Schapire & Singer 1998) to classify each window.

- Weight distribution over examples and classes $W_{i,l}^t$
Algorithm

- We used AdaBoost.MH (Schapire & Singer 1998) to classify each window

  - Weight distribution over examples and classes $w_{i,l}^t$
  - Find the dimension and threshold that minimizes the weighted error on one-vs-all binary classifiers

$\begin{align*}
  d_k & \quad 2 \quad 2 \quad 1 \quad 1 \quad 3 \quad 1 \quad 2 \quad 3 \\
  h_1^{(t)} & \quad x \quad x \quad 1 \quad 1 \quad x \quad 1 \quad x \quad x \\
  h_2^{(t)} & \quad 2 \quad 2 \quad x \quad x \quad x \quad x \quad 2 \quad x \\
  h_3^{(t)} & \quad x \quad x \quad x \quad x \quad 3 \quad x \quad x \quad 3 \\
\end{align*}$

$\rightarrow h^{(t)}$
Evaluation

- Tzanetakis Database
  - 1100 files
  - 10 classes:
    - blues, classical, country, disco, hiphop, jazz, metal, pop, reggae, rock

<table>
<thead>
<tr>
<th></th>
<th>Correct Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>G. Tzanetakis (2002)</td>
<td>61%</td>
</tr>
<tr>
<td>T. Li (2003)</td>
<td>79%</td>
</tr>
<tr>
<td><strong>Our Approach (MIREX)</strong></td>
<td><strong>83%</strong></td>
</tr>
<tr>
<td>With Autocorrelation</td>
<td>86%</td>
</tr>
</tbody>
</table>
Evaluation

- The *weak* map

Data Dimensions:
Evaluation

- The weak map

Data Dimensions:
Evaluation

- The *weak* map

Data Dimensions:

```
1 %
-
%9%
-$
<
&<
]
```
Evaluation

- The *weak* map

Data Dimensions:

classes

Example 1: class 2
Evaluation

- The weak map

Data Dimensions:

classes

$+\alpha_1$ $-\alpha_2$ $+\alpha_3$ $-\alpha_4$ $+\alpha_5$

Example 1: class 2
Evaluation

- The weak map

Data Dimensions:

- $-\alpha_1$
- $-\alpha_2$
- $-\alpha_3$
- $+\alpha_4$
- $-\alpha_5$

Example 2: class 1
Evaluation

- The *weak* map
- RCEPS
Mirex Competition

- **Two Databases**
  - **Magnatune** – 10 classes
    - ambient, blues, classical, electronic, ethnic, folk, jazz, new age, punk, rock.
  - **USPOP** – 6 classes
    - country, electronica & dance, new age, rap & hiphop, reggae, rock.
# Mirex Competition Results - USPS

<table>
<thead>
<tr>
<th>Rank</th>
<th>Participant</th>
<th>USPOP Raw Classification Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bergstra, Casagrande, Eck &amp; Kégl (1)</td>
<td><strong>86.29%</strong></td>
</tr>
<tr>
<td>2</td>
<td>Mandel &amp; Ellis</td>
<td>85.65%</td>
</tr>
<tr>
<td>3</td>
<td>Pampalk, E.</td>
<td>80.38%</td>
</tr>
<tr>
<td>4</td>
<td>Lidy &amp; Rauber (SSD+RH)</td>
<td>79.75%</td>
</tr>
<tr>
<td>5</td>
<td>West, K.</td>
<td>78.90%</td>
</tr>
<tr>
<td>6</td>
<td>Lidy &amp; Rauber (RP+SSD)</td>
<td>78.48%</td>
</tr>
<tr>
<td>6</td>
<td>Ahrendt, P.</td>
<td>78.48%</td>
</tr>
<tr>
<td>8</td>
<td>Lidy &amp; Rauber (RP+SSD+RH)</td>
<td>78.27%</td>
</tr>
<tr>
<td>9</td>
<td>Scaringella, N.</td>
<td>75.74%</td>
</tr>
<tr>
<td>10</td>
<td>Soares, V.</td>
<td>66.67%</td>
</tr>
<tr>
<td>11</td>
<td>Tzanetakis, G.</td>
<td>63.29%</td>
</tr>
<tr>
<td>12</td>
<td>Burred, J.</td>
<td>47.68%</td>
</tr>
<tr>
<td>13</td>
<td>Chen &amp; Gao</td>
<td>22.93%</td>
</tr>
<tr>
<td>14</td>
<td>Li, M.</td>
<td>TO *</td>
</tr>
</tbody>
</table>
## Mirex Competition Results - Magna

<table>
<thead>
<tr>
<th>Rank</th>
<th>Participant</th>
<th>Magnatitude Hierarchical Classification Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bergstra, Casagrande, Eck &amp; Kégl (1)</td>
<td>77.25%</td>
</tr>
<tr>
<td>2</td>
<td>Mandel &amp; Ellis</td>
<td>71.96%</td>
</tr>
<tr>
<td>3</td>
<td>West, K.</td>
<td>71.67%</td>
</tr>
<tr>
<td>4</td>
<td>Lidy &amp; Rauber (RP+SSD)</td>
<td>71.08%</td>
</tr>
<tr>
<td>5</td>
<td>Lidy &amp; Rauber (RP+SSD+RH)</td>
<td>70.88%</td>
</tr>
<tr>
<td>6</td>
<td>Lidy &amp; Rauber (SSD+RH)</td>
<td>70.78%</td>
</tr>
<tr>
<td>7</td>
<td>Scaringella, N.</td>
<td>70.47%</td>
</tr>
<tr>
<td>8</td>
<td>Pampalk, E.</td>
<td>69.90%</td>
</tr>
<tr>
<td>9</td>
<td>Ahrendt, P.</td>
<td>64.61%</td>
</tr>
<tr>
<td>10</td>
<td>Burred, J.</td>
<td>59.22%</td>
</tr>
<tr>
<td>11</td>
<td>Tzanetakis, G.</td>
<td>58.14%</td>
</tr>
<tr>
<td>12</td>
<td>Soares, V.</td>
<td>55.29%</td>
</tr>
<tr>
<td>13</td>
<td>Li, M.</td>
<td>TO *</td>
</tr>
<tr>
<td>13</td>
<td>Chen &amp; Gao</td>
<td>DNC *</td>
</tr>
</tbody>
</table>
## Mirex Competition Results - Overall

<table>
<thead>
<tr>
<th>Rank</th>
<th>Participant</th>
<th>Mean of Magnature Hierarchical Classification Accuracy and USPOP Raw Classification Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bergstra, Casagrande, Eck &amp; Kégl (1)</td>
<td><strong>81.77%</strong></td>
</tr>
<tr>
<td>2</td>
<td>Mandel &amp; Ellis</td>
<td>78.81%</td>
</tr>
<tr>
<td>3</td>
<td>West, K.</td>
<td>75.29%</td>
</tr>
<tr>
<td>4</td>
<td>Lidy &amp; Rauber (SSD+RH)</td>
<td>75.27%</td>
</tr>
<tr>
<td>5</td>
<td>Pampalk, E.</td>
<td>75.14%</td>
</tr>
<tr>
<td>6</td>
<td>Lidy &amp; Rauber (RP+SSD)</td>
<td>74.78%</td>
</tr>
<tr>
<td>7</td>
<td>Lidy &amp; Rauber (RP+SSD+RH)</td>
<td>74.58%</td>
</tr>
<tr>
<td>8</td>
<td>Scaringella. N.</td>
<td>73.11%</td>
</tr>
<tr>
<td>9</td>
<td>Ahrendt, P.</td>
<td>71.55%</td>
</tr>
<tr>
<td>10</td>
<td>Soares, V.</td>
<td>60.98%</td>
</tr>
<tr>
<td>11</td>
<td>Tzanetakis, G.</td>
<td>60.72%</td>
</tr>
<tr>
<td>12</td>
<td>Burred, J.</td>
<td>53.45%</td>
</tr>
</tbody>
</table>
The End - Again

- Thank you!
  - Questions?