

## AUTOMATIC COMPUTER MUSIC CLASSIFICATION AND SEGMENTATION

Adrian SIMION<sup>1</sup>, Stefan TRAUSAN-MATU<sup>2</sup>

**Rezumat.** *Lucrarea de față descrie și aplică diferite metode pentru segmentarea automată a muzicii realizată cu ajutorul unui calculator. Pe baza rezultatelor și a tehnicilor de extragere a caracteristicilor folosite, se încearcă de asemenea o clasificare/recunoaștere a fragmentelor folosite. Algoritmii au fost testați pe seturile de date Magnatune și MARSYAS, dar instrumentele software implementate pot fi folosite pe o gamă variată de surse. Instrumentele descrise vor fi integrate într-un „framework” / sistem software numit ADAMS (Advanced Dynamic Analysis of Music Software - Software pentru Analiza Dinamică Avansată a Muzicii) cu ajutorul căruia se vor putea evalua și îmbunătăți diferitele sarcini de analiză și compoziție a muzicii. Acest sistem are la bază biblioteca de programe MARSYAS și conține un modul similar cu WEKA pentru sarcini de procesare a datelor și învățare automată.*

**Abstract.** *This paper describes and applies various methods for automatic computer music segmentation. Based on these results and on the feature extraction techniques used, is tried also a genre classification/recognition of the excerpts used. The algorithms were tested on the Magnatune and MARSYAS datasets, but the implemented software tools can also be used on a variety of sources. The tools described here will be subject to a framework/software system called ADAMS (Advanced Dynamic Analysis of Music Software) that will help evaluate and enhance the various music analysis/composition tasks. This system is based on the MARSYAS open source software framework and contains a module similar to WEKA for data-mining and machine learning tasks.*

**Keywords:** automatic segmentation, audio classification, music information retrieval, music content analysis, chord detection, vocal and instrumental regions

### 1. Music Information Retrieval

The number of digital music recordings has a continuous growth, promoted by the users' interest as well as the advances of the new technologies that support the pleasure of listening to music. There are a few reasons that explain this trend, first of all, the existential characteristic of the musical language. Music is a form of art which can be shared by people that belong to different cultures because it surpasses the borders of the national language and of the cultural background. As an example the West American music has many enthusiasts in Japan, and many persons in Europe appreciate the classical Indian music. These forms of

---

<sup>1</sup>Eng., Ph.D. student, Faculty of Automatic Control and Computers, University Politehnica of Bucharest, Bucharest, Romania, (simion.adrian@gmail.com).

<sup>2</sup>Corresponding member of AOSR. Prof., Ph.D., Faculty of Automatic Control and Computers, University Politehnica of Bucharest, Bucharest, Romania, (stefan.trausan@cs.pub.ro).

expression can be appreciated without the need of a translation that is in most of the cases necessary for accessing foreign textual papers.

Another reason is the fact that technology for recording music, digital transformation and playback allows the users access to information that is almost comparable to live performances, at least at audio quality level.

Last, music is an art form that is cult and popular at the same time and sometimes is impossible to draw a line between the two, like jazz and traditional music.

The high availability and demand for music content induced new requirements about its management, advertisement and distribution. This required a more in-depth and direct analysis of the content than that provided by simple human driven meta-data cataloguing.

The new techniques allowed approaches that were only encountered in theoretical musical analysis. One of these problems was stated by Frank Howes [1]: There is thus a vast corpus of music material available for comparative study. It would be fascinating to discover and work out a correlation between music and social phenomena. With the current processing power and advancements we can answer questions such as: What is the ethnic background of a particular piece of music or what cultures it spawns.

In light of these possibilities and technological advances we needed a new discipline that would try to cover and answer the various problems. Music Information Retrieval (MIR) is an interdisciplinary science that retrieves its information from music. The origins of MIR are domains like: musicology, cognitive psychology, linguistic and computer science.

An active research area is composed of new methods and tools for pattern finding as well as the comparison of musical content. The International Society for Music Information Retrieval [2] is coupled with the annual Music Information Retrieval Evaluation eXchange (MIREX) [3]. The evaluated tasks include Automatic Genre Identification, Chord Detection, Segmentation, Melody Extraction, Query by Humming, to name a few. This paper will focus mostly on Automatic Segmentation and Genre Identification.

## **2. Former studies and related work on Automatic Music Segmentation**

The topic of speech/music classification was studied by many researchers. While the applications can be very different, many studies use similar sets of acoustic features, such as short time energy, zero-crossing rate, cepstrum coefficients, spectral roll off, spectrum centroid and “loudness,” alongside some unique features, such as “dynamism.” However, the exact combinations of features used can vary greatly, as well as the size of the feature set.

---

Typically some long term statistics, such as the mean or the variance, and not the features themselves, are used for the discrimination.

The major differences between the different studies lie in the exact classification algorithm, even though some popular classifiers (K-nearest neighbor, Gaussian multivariate, neural network) are often used as a basis.

For the studies, mostly, different databases are used for training and testing the algorithm. It is worth noting that in these studies, especially the early ones, these databases are fairly small. The following table describes some of the former studies:

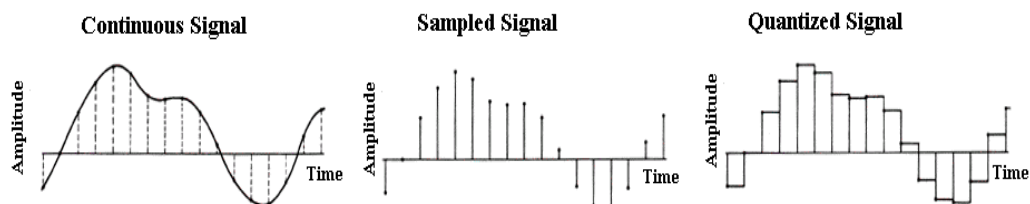
**Table 1.** Some of the former studies

<i>Author</i>	<i>Application</i>	<i>Features</i>	<i>Classification method</i>
Saunders, 1996 [4]	Automatic real-time FM radio monitoring	Short-time energy, statistical parameters of the ZCR	Multivariate Gaussian classifier
Scheirer and Slaney, 1997 [5]	Speech/music discrimination for automatic speech recognition	13 temporal, spectral and cepstral features (e.g., 4Hz modulation energy, % of low energy frames, spectral roll off, spectral centroid, spectral flux, ZCR, cepstrum-based feature, "rhythmicness"), variance of features across 1 sec.	Gaussian mixture model (GMM), K nearest neighbour (KNN), K-D trees, multidimensional Gaussian MAP estimator
Foote, 1997 [6]	Retrieving audio documents by acoustic similarity	12 MFCC, Short-time energy	Template matching of histograms, a tree-based vector quantizer, trained to maximize mutual information
Liu et al., 1997 [7]	Analysis of audio for scene classification of TV programs	Silence ratio, volume std, volume dynamic range, 4Hz freq. mean and std of pitch difference, speech, noise ratios, freq. centroid, bandwidth, energy in 4 sub-bands	A neural network using the one-class-in-one network (OCON) structure
Zhang and Kuo, 1999 [8]	Audio segmentation/retrieval for video scene classification, indexing of raw audio visual recordings, database browsing	Features based on short-time energy, average ZCR, short-time fundamental frequency	A rule-based heuristic procedure for the coarse stage, HMM for the second stage
Williams and Ellis, 1999 [9]	Segmentation of speech versus non speech in automatic speech recognition tasks	Mean per-frame entropy and average probability "dynamism", background-label energy ratio, phone distribution match—all derived from posterior probabilities of phones in hybrid connectionist-HMM framework	Gaussian likelihood ratio test
El-Malehet al., 2000 [10]	Automatic coding and content based audio/video retrieval	LSF, differential LSF, measures based on the ZCR of high-pass filtered signal	KNN classifier and quadratic Gaussian classifier (QCG)
Buggati et al., 2002 [11]	"Table of Content description" of a multimedia document	ZCR-based features, spectral flux, shorttime energy, cepstrum coefficients, spectral centroids, ratio of the high-frequency power spectrum, a measure based on syllabic frequency	Multivariate Gaussian classifier, neural network (MLP)
Lu, Zhang, and Jiang,	Audio content analysis in video parsing	High zero-crossing rate ratio (HZCRR), low short-time energy ratio (LSTER),	3-step classification: 1. KNN and linear spectral

2002 [12]		linear spectral pairs, band periodicity, noise-frame ratio (NFR)	pairs-vector quantization (LSP-VQ) for speech/nonspeech discrimination. 2. Heuristic rules for nonspeech classification into music/background noise/silence. 3. Speaker segmentation
Ajmera et al., 2003 [13]	Automatic transcription of broadcast news	Averaged entropy measure and “dynamism” estimated at the output of a multilayer perceptron (MLP) trained to emit posterior probabilities of phones. MLP input: 13 first cepstra of a 12th-order perceptual linear prediction filter.	2-state HMM with minimum duration constraints (threshold free, unsupervised, no training).
Burred and Lerch, 2004 [14]	Audio classification (speech/music/background noise), music classification into genres	Statistical measures of short-time frame features: ZCR, spectral centroid/roll off/flux, first 5 MFCCs, audio spectrum centroid/flatness, harmonic ratio, beat strength, rhythmic regularity, RMS energy, time envelope, low energy rate, loudness	KNN classifier, 3-component GMM classifier
Barbedo and Lopes, 2006 [15]	Automatic segmentation for real-time applications	Features based on ZCR, spectral roll off, loudness and fundamental frequencies	KNN, self-organizing maps, MLP neural networks, linear combinations
Muñoz-Expósito et al., 2006 [16]	Intelligent audio coding system	Warped LPC-based spectral centroid	3-component GMM, with or without fuzzy rules-based system
Alexandre et al, 2006 [17]	Speech/music classification for musical genre classification	Spectral centroid/roll off, ZCR, short-time energy, low short time energy ratio (LSTER), MFCC, voice to-white	Fisher linear discriminant, K nearest neighbor

## 2.1. Digital Audio Signals

When music is recorded, the continuous pressure from the sound wave is measured using a microphone. These measurements are taken at a regular time and each measurement is quantized.



**Fig. 1.** Digital sound representation (time domain):

- a.** Music is a continuous signal;...      **b.** that is sampled...      **c.** and Quantized

Sound can be represented as a sum of sinusoids. A signal of  $N$  samples can be written as:

$$x = \sum_{k=0}^{N/2} a_k^{(r)} \cos(2\pi(\frac{k}{N})) + a_k^{(i)} \sin(2\pi(\frac{k}{N})). \quad (1)$$

The signal can be represented in the *frequency* domain using the coefficients  $\{(a_1^{(y)}, a_1^{(i)}), \dots, (a_{N/2}^{(y)}, a_{N/2}^{(i)})\}$ .

The magnitude and phase of the  $k^{\text{th}}$  frequency component are given by:

$$X_M[k] = \sqrt{(a_k^{(r)})^2 + (a_k^{(i)})^2} \quad (2)$$

$$X_p[k] = \arctan(\frac{a_k^{(i)}}{a_k^{(r)}}) \quad (3)$$

Perceptual studies on human hearing show that the phase information is relatively unimportant when compared to magnitude information, thus the phase component during feature extraction is usually ignored. [19]

The *Spectral Centroid* is another spectral-shape feature that is useful in the extraction and analysis process. We can see from Table 1 its various uses. The *Spectral Centroid* is the center of gravity of the spectrum and is given by:

$$C = \frac{\sum_{k=1}^{N/2} X_M[k] * k}{\sum_{k=1}^{N/2} X_M[k]} \quad (4)$$

The Spectral Centroid can be thought of as a measure of ‘brightness’ since songs are considered brighter when they have more high frequency components.

## 2.2. Time-Frequency Domain Transforms

In MIR and sound analysis in general it is common to do transformation between the time and frequency domains. For this the mathematical apparatus gives us the real discrete Fourier transform (DFT), the real short-time Fourier transform (STFT), discrete cosine transform (DCT), discrete wavelet transform (DWT) and also the gammatone transform (GT).

Music analysis is not concerned with complex transforms, since music is always a real-valued time series and has only positive frequencies.

Given a signal  $x$  with  $N$  samples, the basis functions for the DFT will be  $N/2$  sine waves and  $N/2$  cosine waves that correspond to the previous coefficients.

The projection operator is correlation, which is a measure of how similar two time series are to one another. The coefficients are found by:

$$a_k^{(r)} = \frac{2}{N} \sum_{i=0}^{N-1} x[i] \cos(2\pi \frac{k}{N} i) \quad (5)$$

$$a_k^{(i)} = \frac{2}{N} \sum_{i=0}^{N-1} x[i] \cos(-2\pi \frac{k}{N} i) \quad (6)$$

The DFT is computed in an efficient manner by the fast Fourier transform FFT. One drawback of both the time series representation and the spectrum representation is that neither simultaneously represents both time and frequency information. A time-frequency representation is found using the short-time Fourier transform (STFT): First, the audio signal is broken up into a series of (overlapping) segments. Each segment is multiplied by a *window function*. The length of the window is called the *window size*.

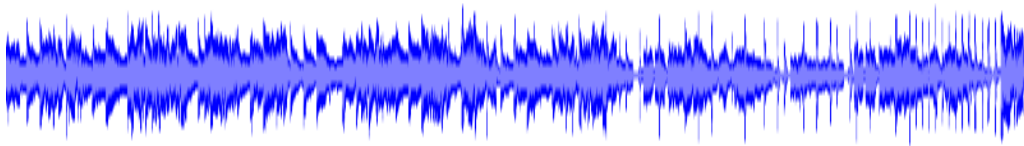


Fig. 2. Magnatune apa\_ya-apa\_ya-14-maani-59-88.wav (time domain).

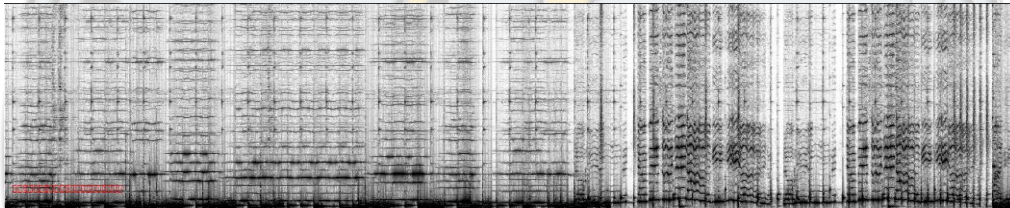


Fig. 3. Magnatune apa\_ya-apa\_ya-14-maani-59-88.wav (spectrogram).

Fig 2 and 3 were obtained with a tweaked version of the MARSYAS's tool sound2png with the following commands:

```
./sound2png -m waveform ../audio/magnatune/0/apa_ya-apa_ya-14-maani-59-88.wav
../saveres/magnatunewav.png -ff Adventure.ttf

./sound2png -m spectrogram ../audio/magnatune/0/apa_ya-apa_ya-14-maani-59-88.wav
../saveres/magnatunespec.png -ff Adventure.ttf
```

Another useful transformation is the wavelet transform.

### 2.3. Mel-Frequency Cepstral Coefficients (MFCC)

The most common set of features used in speech recognition and music annotation systems are the Mel-Frequency Cepstral Coefficients (MFCC). MFCC are short-time features that characterize the magnitude spectrum of an audio signal. For each short-time (25 ms) segment, the feature vector is found using the five step algorithm given in Algorithm 1. The first step is to obtain the magnitude of each frequency component in the frequency domain using the DCT. We then take the log of the magnitude since perceptual loudness has been shown to be

approximately logarithmic. The frequency components are then merged into 40 bins that have been spaced according to the Mel-scale.

The Mel-scale is a mapping between true frequency and a model of perceived frequency that is approximately logarithmic.

Since a time-series of these 40-dimensional Mel-frequency vectors will have highly redundant information, we could reduce dimensionality using PCA.

Instead, the speech community has adopted the discrete cosine transform (DCT), which approximates PCA but does not require training data, to reduce the dimensionality to a vector of 13 MFCCs. [20]

**Algorithm 1.** Calculating MFCC Feature Vector

- 1: Calculate the spectrum using the DFT
- 2: Take the log of the spectrum
- 3: Apply Mel-scaling and smoothing
- 4: Decorrelate using the DCT.

### 3. Problem description

A common feature that aids record producers to meet the demands of the target audiences, musicologists to study musical influences and music enthusiasts to summarize their collections is the musical genre identification.

The genre concept is inherently subjective because the influences, hierarchy or the intersection of a song to a specific genre isn't universally agreed upon.

This point is backed up by a comparison of three Internet music providers that found very big differences in the number of genres, the words that describe that genre, and the structure of the genre hierarchies. [18]

Although there are some inconsistencies caused by its subjective nature, the genre concept has shown interest from the MIR community.

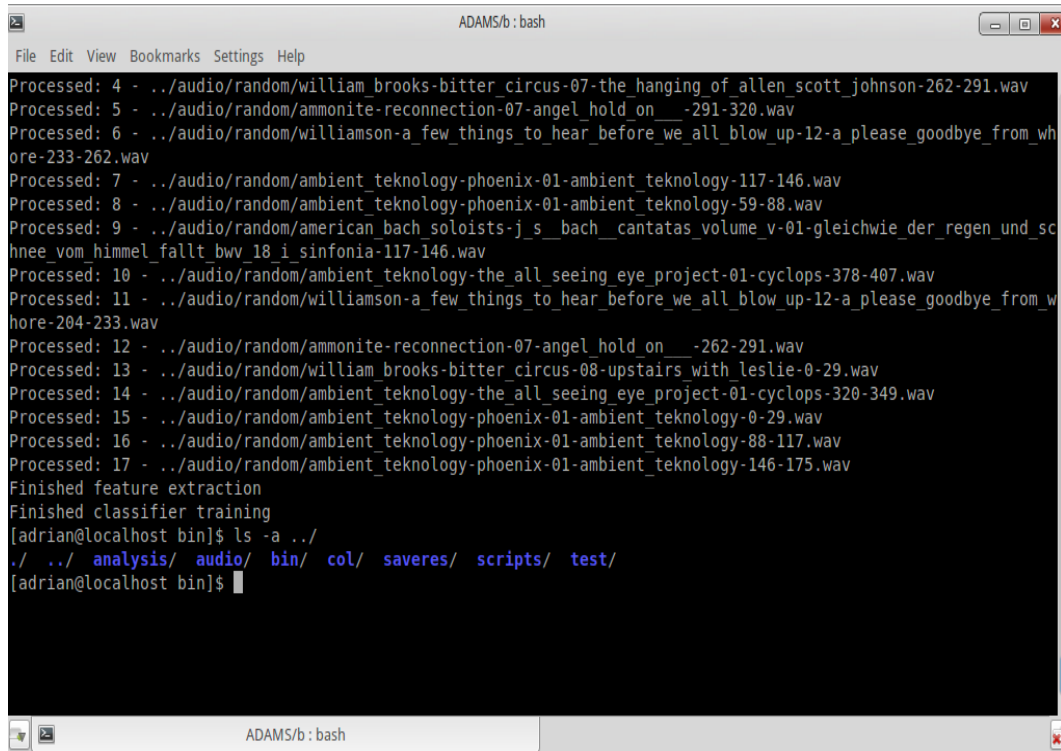
The various papers and works on this subject reflect the authors' assumptions about the genres. Copyright laws prevented authors from establishing a common database of songs, making it difficult to directly compare the results.

### 4. Experiments description

The datasets used for training and testing were MAGNATUNE [21] and two collections that were built in the early stages of the MARSYAS [22] framework.

As the ADAMS system is built in a modular form the various tasks (described below) can be automatized and the sound can "flow" through these modules until the complete analysis is made.

The ADAMS main directory structure can be seen in the following picture:



```

ADAMS/b : bash
File Edit View Bookmarks Settings Help
Processed: 4 - ../audio/random/william_brooks-bitter_circus-07-the_hanging_of_allen_scott_johnson-262-291.wav
Processed: 5 - ../audio/random/ammonite-reconnection-07-angel_hold_on___-291-320.wav
Processed: 6 - ../audio/random/williamson-a_few_things_to_hear_before_we_all_blow_up-12-a_please_goodbye_from_w
ore-233-262.wav
Processed: 7 - ../audio/random/ambient_teknology-phoenix-01-ambient_teknology-117-146.wav
Processed: 8 - ../audio/random/ambient_teknology-phoenix-01-ambient_teknology-59-88.wav
Processed: 9 - ../audio/random/american_bach_soloists-j_s_bach_cantatas_volume_v-01-gleichwie_der_regen_und_sc
hnee_vom_himmel_fallt_bwv_18_i_sinfonia-117-146.wav
Processed: 10 - ../audio/random/ambient_teknology-the_all_seeing_eye_project-01-cyclops-378-407.wav
Processed: 11 - ../audio/random/williamson-a_few_things_to_hear_before_we_all_blow_up-12-a_please_goodbye_from_w
hore-204-233.wav
Processed: 12 - ../audio/random/ammonite-reconnection-07-angel_hold_on___-262-291.wav
Processed: 13 - ../audio/random/william_brooks-bitter_circus-08-upstairs_with_leslie-0-29.wav
Processed: 14 - ../audio/random/ambient_teknology-the_all_seeing_eye_project-01-cyclops-320-349.wav
Processed: 15 - ../audio/random/ambient_teknology-phoenix-01-ambient_teknology-0-29.wav
Processed: 16 - ../audio/random/ambient_teknology-phoenix-01-ambient_teknology-88-117.wav
Processed: 17 - ../audio/random/ambient_teknology-phoenix-01-ambient_teknology-146-175.wav
Finished feature extraction
Finished classifier training
[adrian@localhost bin]$ ls -a ../
./ ../ analysis/ audio/ bin/ col/ saveres/ scripts/ test/
[adrian@localhost bin]$

```

Fig. 4. ADAMS Main Directory Structure.

The machine learning tasks are done with the WEKA [23] tool, loading the compatible arff files produced with the aid of MARSYAS.

The chosen OS for these experiments was Mandriva Linux 2011, the compiler version being “gcc (GCC) 4.6.1 20110627 (Mandriva)”.

Extractors that were used:

- BEAT: Beat histogram features
- LPCC: LPC derived Cepstral coefficients
- LSP: Linear Spectral Pairs
- MFCC: Mel-Frequency Cepstral Coefficients
- SCF: Spectral Crest Factor (MPEG-7)
- SFM: Spectral Flatness Measure (MPEG-7)
- SFMSCF: SCF and SFM features
- STFT: Centroid, Rolloff, Flux, ZeroCrossings
- STFTMFCC: Centroid, Rolloff Flux, ZeroCrossings, Mel-Frequency Cepstral Coefficients

On every experiment for the specified extractors are also presented the confusion matrices [24] in order to have an idea about the actual and the predicted classifications done by the classification system.



#### 4.1. Experiment 1: Classification using “Timbral Features”

This experiment uses the following extractors: Time ZeroCrossings, Spectral Centroid, Flux and Rolloff, and Mel-Frequency Cepstral Coefficients (MFCC).

We extract these features with the option – timbral and we also create the file that will be loaded with the WEKA environment for analysis with the following command:

```
./adamsfeature -sv -timbral ../col/all.mf -w ../analysis/alltimbral.arff
```

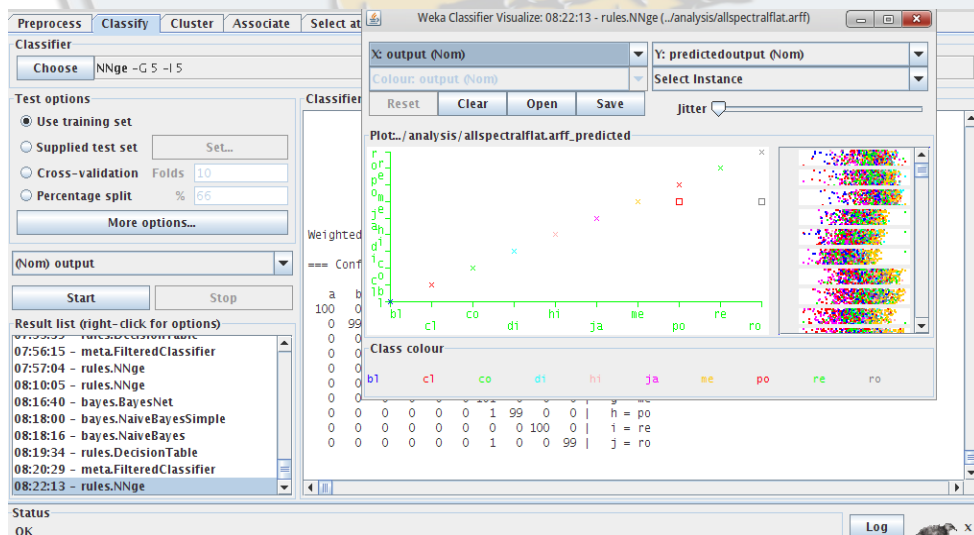
Based on experiment the following classifiers were chosen: Bayes Network, Naive Bayes, Decision Table, Filtered Classifier and NNGE.

The results are shown in the following table:

**Table 2.** Timbral Features - Classifier Results

Classifier	Model Build Time(s)	Coorectly Classified	Incorrectly Classified	Mean absolut error	Root mean squared error	Relative absolute error	Root relative squared error
Bayes Network	1.78	62.5%	37.5%	0.0753	0.2648	41.82%	88.28%
Naive Bayes	0.04	55%	45%	0.0902	0.2925	50.09%	97.51%
Decision Table	15.49	51.6%	48.4%	0.1467	0.2599	81.53%	86.64%
Filtered Classifier	4.55	87.8%	12.2%	0.0348	0.1318	19.31%	43.94%
NNGE	10.69	100%	0%	0	0	0	0

Table 2 was build loading the file alltimbral.arff in WEKA and training the built-in classifiers



**Fig. 5.** WEKA Prediction Errors Graph.

```

=== Confusion Matrix === Bayes Network
  a b c d e f g h i j <-- classified as
63 0 5 7 2 1 6 8 2 6 | a = bl
4 82 0 0 0 10 0 0 0 3 | b = cl
6 2 66 7 0 7 4 1 1 6 | c = co
1 1 6 64 4 1 2 6 9 6 | d = di
0 0 0 17 45 1 2 16 19 0 | e = hi
19 14 2 1 0 58 2 1 1 2 | f = ja
2 2 2 2 2 4 77 2 1 7 | g = me
7 0 1 10 7 2 1 66 3 3 | h = po
1 1 3 11 9 4 0 5 61 5 | i = re
5 1 8 14 0 9 14 1 5 43 | j = ro

=== Confusion Matrix === Naive Bayes
  a b c d e f g h i j <-- classified as
40 0 14 0 3 13 6 4 0 12 | a = bl
0 90 0 0 0 7 0 0 0 2 | b = cl
9 3 59 3 0 2 5 1 0 18 | c = co
1 0 5 45 5 2 3 2 6 31 | d = di
3 0 1 14 51 0 2 7 12 10 | e = hi
9 33 2 0 0 46 2 1 1 6 | f = ja
0 0 1 1 3 5 67 2 0 22 | g = me
2 0 4 10 11 2 2 51 9 9 | h = po
2 0 19 8 8 3 0 2 44 14 | i = re
4 3 11 3 0 4 12 1 5 57 | j = ro

=== Confusion Matrix === Decision table
  a b c d e f g h i j <-- classified as
26 0 14 12 1 5 5 9 23 5 | a = bl
2 73 0 0 0 15 4 0 0 5 | b = cl
2 1 66 11 0 4 3 2 1 10 | c = co
9 1 6 44 4 3 2 7 23 1 | d = di
3 0 4 15 45 0 0 9 24 0 | e = hi
9 10 4 5 0 57 1 0 6 8 | f = ja
5 1 9 11 0 1 60 4 0 10 | g = me
6 0 2 15 17 0 2 47 7 4 | h = po
7 0 4 7 10 2 0 2 67 1 | i = re
2 2 11 19 1 5 16 3 10 31 | j = ro

=== Confusion Matrix === Filtered classifier
  a b c d e f g h i j <-- classified as
95 0 3 0 0 0 0 1 1 0 | a = bl
3 91 0 0 0 4 0 0 0 1 | b = cl
7 0 87 0 0 1 2 1 0 2 | c = co
2 0 6 91 0 0 0 0 1 0 | d = di
2 0 1 4 86 0 1 2 4 0 | e = hi
2 3 0 1 1 93 0 0 0 0 | f = ja
2 1 0 3 1 1 90 1 0 2 | g = me
2 1 3 2 3 1 1 86 0 1 | h = po
1 0 2 4 4 1 2 0 84 2 | i = re
1 0 7 2 2 6 3 2 2 75 | j = ro

=== Confusion Matrix === NNGE
  a b c d e f g h i j <-- classified as
100 0 0 0 0 0 0 0 0 0 | a = bl
0 99 0 0 0 0 0 0 0 0 | b = cl
0 0 100 0 0 0 0 0 0 0 | c = co
0 0 0 100 0 0 0 0 0 0 | d = di
0 0 0 0 100 0 0 0 0 0 | e = hi
0 0 0 0 0 100 0 0 0 0 | f = ja
0 0 0 0 0 0 101 0 0 0 | g = me
0 0 0 0 0 0 0 100 0 0 | h = po
0 0 0 0 0 0 0 0 100 0 | i = re
0 0 0 0 0 0 0 0 0 100 | j = ro

```

Fig. 6. Confusion Matrices for Timbral Features Classification

#### 4.2. Experiment 2: Classification using “Spectral Features”

This experiment uses the following extractors: Spectral Centroid, Flux and Roll off. The feature extraction was done with the following command:

```
./adamsfeature -sv -spfe ./col/all.mf -w ./analysis/allspectral.arff
```

Using the same classifiers the results are:

**Table 3.** Spectral Features - Classifier Results

Classifier	Model Build Time(s)	Correctly Classified	Incorrectly Classified	Mean absolute error	Root mean squared error	Relative absolute error	Root relative squared error
Bayes Network	1.78	46.5%	53.5%	0.1192	0.2742	66.21%	91.41%
Naive Bayes	0.23	42.5%	57.5%	0.1205	0.2924	66.92%	97.47%
Decision Table	0.72	46.1%	53.9%	0.1491	0.2655	82.82%	88.49%
Filtered Classifier	0.41	63.6%	36.4%	0.099	0.2225	54.98%	74.15%
NNGE	2.02	100%	0%	0	0	0	0

```

=== Confusion Matrix === Bayes Network
  a b c d e f g h i j <-- classified as
41 2 11 6 0 9 16 2 7 6 | a = bl
 1 76 2 0 0 11 3 0 0 6 | b = cl
16 6 36 3 0 13 12 2 3 9 | c = co
10 0 4 41 11 0 10 18 5 1 | d = di
 4 0 2 15 40 0 1 19 18 1 | e = hi
10 23 12 3 1 34 5 6 0 6 | f = ja
 4 1 4 4 1 3 80 2 1 1 | g = me
 5 0 3 3 21 1 2 53 7 5 | h = po
 9 0 4 5 21 0 2 7 51 1 | i = re
14 8 8 10 1 5 23 9 9 13 | j = ro

=== Confusion Matrix === Naive Bayes
  a b c d e f g h i j <-- classified as
45 2 16 3 1 0 25 1 5 2 | a = bl
 2 81 2 0 2 5 5 0 0 2 | b = cl
19 13 37 2 1 3 15 2 0 8 | c = co
11 0 6 41 2 0 21 12 6 1 | d = di
 5 0 1 20 16 0 4 24 26 4 | e = hi
16 34 20 0 2 12 6 5 1 4 | f = ja
 5 0 7 8 0 0 77 2 0 2 | g = me
 3 2 6 14 3 1 2 57 5 7 | h = po
11 0 5 10 2 1 3 15 51 2 | i = re
11 10 13 11 4 2 33 4 4 8 | j = ro

=== Confusion Matrix === Decision table
  a b c d e f g h i j <-- classified as
31 4 21 10 1 12 16 2 2 1 | a = bl
 1 79 2 0 0 12 4 0 0 1 | b = cl
11 9 44 9 1 11 9 2 1 3 | c = co
 5 0 10 47 4 1 11 15 3 4 | d = di
 2 0 4 10 39 0 3 22 20 0 | e = hi
14 23 8 5 2 35 5 5 0 3 | f = ja
 0 2 6 11 1 5 62 3 0 11 | g = me
 7 4 1 8 18 1 0 57 3 1 | h = po
 7 0 6 9 16 2 2 9 49 0 | i = re
11 4 13 14 3 9 18 5 5 18 | j = ro

=== Confusion Matrix === Filtered classifier
  a b c d e f g h i j <-- classified as
74 1 6 2 0 8 4 0 1 4 | a = bl
 1 81 2 0 0 13 1 0 0 1 | b = cl
10 8 52 5 0 13 6 1 1 4 | c = co
 3 0 4 66 3 1 8 8 4 3 | d = di
 5 0 0 11 52 0 1 13 16 2 | e = hi
 9 15 4 0 1 63 4 1 0 3 | f = ja
 5 1 3 5 0 3 77 1 1 5 | g = me
 5 1 5 5 12 0 0 66 3 3 | h = po
 7 0 5 6 12 0 1 7 61 1 | i = re
14 4 8 4 2 3 14 4 3 44 | j = ro

=== Confusion Matrix === NNGE
  a b c d e f g h i j <-- classified as
100 0 0 0 0 0 0 0 0 0 | a = bl
 0 99 0 0 0 0 0 0 0 0 | b = cl
 0 0 100 0 0 0 0 0 0 0 | c = co
 0 0 0 100 0 0 0 0 0 0 | d = di
 0 0 0 0 100 0 0 0 0 0 | e = hi
 0 0 0 0 0 100 0 0 0 0 | f = ja
 0 0 0 0 0 0 101 0 0 0 | g = me
 0 0 0 0 0 0 0 100 0 0 | h = po
 0 0 0 0 0 0 0 0 100 0 | i = re
 0 0 0 0 0 0 0 0 0 100 | j = ro

```

Fig. 7. Confusion Matrices for Spectral Features Classification

### 4.3 Experiment 2: Classification using “MFCC”

This experiment uses the Mel-Frequency Cepstral Coefficients extractors. The feature extraction was done with the following command:

```
./adamsfeature -sv -mfcc ../col/all.mf -w ../analysis/allmfcc.arff
```

Table 4. MFCC Features - Classifier Results

Classifier	Model Build Time(s)	Correctly Classified	Incorrectly Classified	Mean absolute error	Root mean squared error	Relative absolute error	Root relative squared error
Bayes Network	1.23	63.3%	36.7%	0.0764	0.2475	42.42%	82.50%
Naive Bayes	0.22	58.5%	41.5%	0.0847	0.2694	47.07%	89.80%
Decision Table	6.4	49.1%	50.9%	0.1481	0.2638	82.27%	87.94%
Filtered Classifier	0.81	87.1%	12.9%	0.0363	0.1348	20.18%	44.92%
NNGE	3.74	99.8%	0.2%	0.0004	0.02	0.22%	6.66%

```

=== Confusion Matrix === Bayes Network
  a b c d e f g h i j <-- classified as
47 1 6 13 0 1 6 11 8 7 | a = bl
0 92 0 0 0 3 1 0 0 3 | b = cl
5 3 69 3 1 0 4 5 1 9 | c = co
3 0 5 48 1 1 9 10 10 13 | d = di
2 0 0 11 58 0 5 1 22 1 | e = hi
3 10 1 1 0 77 2 0 1 5 | f = ja
3 0 1 0 0 0 84 6 0 7 | g = me
1 0 9 13 9 3 3 45 12 5 | h = po
6 0 2 9 6 0 0 2 68 7 | i = re
4 0 13 8 3 3 18 5 1 45 | j = ro

=== Confusion Matrix === Naive Bayes
  a b c d e f g h i j <-- classified as
45 0 8 12 0 4 8 4 7 12 | a = bl
0 93 0 0 0 2 1 0 0 3 | b = cl
6 3 55 14 1 0 3 2 2 14 | c = co
2 0 5 47 1 1 16 6 10 12 | d = di
1 0 0 13 64 0 5 5 9 3 | e = hi
6 9 0 3 1 57 6 0 0 18 | f = ja
5 0 0 2 0 0 87 3 1 3 | g = me
2 0 6 22 10 5 3 31 9 12 | h = po
6 0 3 12 7 1 0 2 61 8 | i = re
2 0 9 11 4 2 23 2 2 45 | j = ro

=== Confusion Matrix === Decision table
  a b c d e f g h i j <-- classified as
36 1 14 8 2 1 7 22 9 0 | a = bl
0 74 0 0 0 21 1 0 0 3 | b = cl
18 2 29 8 1 4 3 19 14 2 | c = co
4 0 9 47 7 1 5 17 7 3 | d = di
5 0 3 13 45 1 1 19 13 0 | e = hi
2 25 2 0 0 60 4 1 3 3 | f = ja
4 0 1 16 0 1 70 7 0 2 | g = me
6 0 4 19 6 1 3 58 3 0 | h = po
13 0 6 2 11 2 1 8 57 0 | i = re
11 0 11 17 2 4 18 15 7 15 | j = ro

=== Confusion Matrix === Filtered classifier
  a b c d e f g h i j <-- classified as
91 0 1 2 0 1 2 1 1 1 | a = bl
0 95 1 0 0 1 2 0 0 0 | b = cl
5 0 86 1 0 1 0 4 0 3 | c = co
3 0 1 85 2 0 2 2 1 4 | d = di
3 0 0 2 88 1 3 2 1 0 | e = hi
0 3 1 2 1 92 0 0 0 1 | f = ja
1 0 0 2 0 0 93 1 1 3 | g = me
3 0 2 3 2 0 1 87 1 1 | h = po
6 0 1 1 7 0 0 3 79 3 | i = re
4 0 5 4 1 1 4 5 1 75 | j = ro

=== Confusion Matrix === NNGE
  a b c d e f g h i j <-- classified as
100 0 0 0 0 0 0 0 0 0 | a = bl
0 99 0 0 0 0 0 0 0 0 | b = cl
0 0 100 0 0 0 0 0 0 0 | c = co
0 0 0 100 0 0 0 0 0 0 | d = di
0 0 0 0 100 0 0 0 0 0 | e = hi
0 0 0 0 0 100 0 0 0 0 | f = ja
0 0 0 0 0 0 101 0 0 0 | g = me
0 0 0 0 0 0 0 100 0 0 | h = po
0 0 0 0 0 0 0 0 100 0 | i = re
0 0 0 0 0 0 0 0 0 100 | j = ro

```

Fig. 8. Confusion Matrices for MFCC Features Classification

#### 4.4 Experiment 4: Classification using “Zero Crossings”

The feature extraction was done with the following command:

```
./adamsfeature -sv -zcrs ../col/all.mf -w ../analysis/allzcrs.arff
```

Table 5. Zero Crossings Features - Classifier Results

Classifier	Model Build Time(s)	Correctly Classified	Incorrectly Classified	Mean absolute error	Root mean squared error	Relative absolute error	Root relative squared error
Bayes Network	0.09	34.7%	65.3%	0.1437	0.2789	79.83%	92.97%
Naive Bayes	0.01	34.5%	65.5%	0.1441	0.2869	80.06%	95.63%
Decision Table	0.22	42.4%	57.6%	0.1511	0.2691	83.95%	89.71%
Filtered Classifier	0.15	44%	56%	0.1403	0.2649	77.94%	88.24%
NNGE	0.52	99.8%	0.2%	0.0004	0.02	0.22%	6.66%

```

=== Confusion Matrix === Bayes Network
  a b c d e f g h i j <-- classified as
43 10 13 9 3 1 13 0 8 0 | a = bl
 6 76 4 0 1 6 6 0 0 0 | b = cl
28 14 12 6 5 8 10 1 16 0 | c = co
 9 0 4 32 10 1 16 19 9 0 | d = di
11 0 7 27 17 3 2 29 4 0 | e = hi
22 38 9 8 2 12 5 3 1 0 | f = ja
 3 3 1 10 3 1 72 7 1 0 | g = me
 4 0 2 14 9 0 1 62 8 0 | h = po
19 0 14 10 11 4 3 18 21 0 | i = re
17 7 10 15 4 4 23 10 10 0 | j = ro

=== Confusion Matrix === Naive Bayes
  a b c d e f g h i j <-- classified as
42 17 13 6 0 3 16 1 2 0 | a = bl
 2 86 0 0 0 6 5 0 0 0 | b = cl
25 28 16 5 2 3 12 2 1 6 | c = co
 5 3 7 35 5 1 23 15 0 6 | d = di
10 0 7 25 13 2 8 28 2 5 | e = hi
18 58 3 2 0 5 6 5 0 3 | f = ja
 8 5 0 15 0 0 72 1 0 0 | g = me
 3 1 8 24 1 0 1 57 1 4 | h = po
20 2 17 8 10 1 2 17 10 13 | i = re
12 15 9 18 3 3 24 6 1 9 | j = ro

=== Confusion Matrix === Decision table
  a b c d e f g h i j <-- classified as
52 5 5 1 1 7 9 0 12 8 | a = bl
 9 73 0 0 0 12 4 0 0 1 | b = cl
35 9 13 4 2 15 6 2 9 5 | c = co
10 0 10 24 14 2 9 19 2 10 | d = di
 6 0 3 9 41 0 2 18 20 1 | e = hi
22 28 5 3 3 26 5 1 6 1 | f = ja
 7 2 2 6 1 1 72 1 0 9 | g = me
 4 0 6 7 10 0 1 62 6 4 | h = po
 4 0 8 1 21 2 0 14 47 3 | i = re
15 4 7 9 5 8 20 5 13 14 | j = ro

=== Confusion Matrix === Filtered classifier
  a b c d e f g h i j <-- classified as
46 8 13 2 0 8 9 0 12 2 | a = bl
 4 87 0 0 0 3 4 0 0 1 | b = cl
23 13 24 4 2 13 6 2 9 4 | c = co
10 3 8 29 8 3 9 19 3 8 | d = di
 6 0 0 12 38 3 2 19 20 0 | e = hi
19 35 4 3 2 23 5 1 7 1 | f = ja
 7 5 1 7 0 2 71 1 0 7 | g = me
 6 1 4 7 10 2 1 62 6 1 | h = po
 4 0 7 2 19 3 0 14 48 3 | i = re
14 9 7 9 5 7 19 5 13 12 | j = ro

=== Confusion Matrix === NNGE
  a b c d e f g h i j <-- classified as
100 0 0 0 0 0 0 0 0 0 | a = bl
 0 99 0 0 0 0 0 0 0 0 | b = cl
 0 0 100 0 0 0 0 0 0 0 | c = co
 0 0 0 100 0 0 0 0 0 0 | d = di
 0 0 0 0 100 0 0 0 0 0 | e = hi
 0 0 0 0 0 100 0 0 0 0 | f = ja
 0 0 0 0 0 0 101 0 0 0 | g = me
 0 0 0 0 0 0 1 99 0 0 | h = po
 0 0 0 0 0 0 0 0 100 0 | i = re
 0 0 0 0 0 0 0 1 0 99 | j = ro

```

Fig. 9. Confusion Matrices for Zero Crossings Features Classification.

#### 4.5 Experiment 5: Classification using “Spectral Flatness Measure”

The feature extraction was done with the following command:

```
./adamsfeature -sv -sfm ./col/all.mf -w ./analysis/allsfm.arff
```

Table 6. SFM Features - Classifier Results

Classifier	Model Build Time(s)	Correctly Classified	Incorrectly Classified	Mean absolute error	Root mean squared error	Relative absolute error	Root relative squared error
Bayes Network	1.78	58.4%	41.6%	0.0838	0.2738	46.53%	91.28%
Naive Bayes	0.15	53.2%	46.8%	0.0935	0.294	51.96%	97.99%
Decision Table	12.35	50.4%	49.6%	0.1472	0.2621	81.78%	87.37%
Filtered Classifier	2.1	83.8%	16.2%	0.045	0.15	25.01%	50.12%
NNGE	9.24	99.8%	0.2%	0.0004	0.02	0.22%	6.66%

```

=== Confusion Matrix === Bayes Network
  a b c d e f g h i j <-- classified as
39 0 9 13 4 12 0 1 13 9 | a = bl
 0 78 12 0 0 8 1 0 0 0 | b = cl
 2 18 50 13 0 5 0 5 3 12 | c = co
 2 1 5 63 9 0 2 5 4 9 | d = di
 3 0 0 7 65 6 10 6 1 2 | e = hi
 5 13 4 1 2 61 2 6 0 6 | f = ja
 0 0 0 8 1 0 82 1 1 8 | g = me
 3 1 5 10 4 4 3 55 5 10 | h = po
 2 0 3 10 6 2 6 10 52 9 | i = re
 2 1 12 18 3 4 14 5 2 39 | j = ro

=== Confusion Matrix === Naive Bayes
  a b c d e f g h i j <-- classified as
34 0 4 23 3 11 0 0 16 9 | a = bl
 1 70 17 0 0 10 1 0 0 0 | b = cl
 2 8 39 20 0 6 0 7 1 17 | c = co
 0 1 1 62 5 1 9 3 6 12 | d = di
 1 0 0 9 61 7 10 3 5 4 | e = hi
17 9 4 5 2 49 3 4 1 6 | f = ja
 0 0 0 7 0 0 83 1 1 9 | g = me
 6 1 5 15 5 2 4 52 1 9 | h = po
 6 0 2 15 6 4 7 15 39 6 | i = re
 3 0 13 16 4 2 15 1 3 43 | j = ro

=== Confusion Matrix === Decision table
  a b c d e f g h i j <-- classified as
43 0 9 14 1 3 1 4 23 2 | a = bl
10 78 3 0 0 6 1 0 1 0 | b = cl
10 13 45 14 0 2 1 2 1 12 | c = co
 8 1 8 43 1 3 1 10 14 11 | d = di
 3 0 0 10 44 4 8 15 14 2 | e = hi
13 9 7 6 2 51 1 4 4 3 | f = ja
 2 0 0 9 0 0 81 1 3 5 | g = me
 9 3 3 12 11 8 4 32 13 5 | h = po
11 1 1 12 8 4 1 8 52 2 | i = re
 8 2 8 19 0 5 13 6 4 35 | j = ro

=== Confusion Matrix === Filtered classifier
  a b c d e f g h i j <-- classified as
88 1 1 1 1 4 1 2 1 0 | a = bl
 2 92 3 0 0 2 0 0 0 0 | b = cl
 4 2 93 0 0 0 0 1 0 0 | c = co
 2 0 9 82 1 1 1 1 2 1 | d = di
 3 1 1 5 86 0 0 2 1 1 | e = hi
 3 3 1 1 2 88 0 1 1 0 | f = ja
 0 0 2 3 1 0 94 1 0 0 | g = me
 2 2 3 3 3 2 1 82 2 0 | h = po
 4 1 3 2 4 3 0 9 73 1 | i = re
 3 2 12 3 4 2 8 2 4 60 | j = ro

=== Confusion Matrix === NNGE
  a b c d e f g h i j <-- classified as
100 0 0 0 0 0 0 0 0 0 | a = bl
 0 99 0 0 0 0 0 0 0 0 | b = cl
 0 0 100 0 0 0 0 0 0 0 | c = co
 0 0 0 100 0 0 0 0 0 0 | d = di
 0 0 0 0 100 0 0 0 0 0 | e = hi
 0 0 0 0 0 100 0 0 0 0 | f = ja
 0 0 0 0 0 0 101 0 0 0 | g = me
 0 0 0 0 0 0 0 1 99 0 | h = po
 0 0 0 0 0 0 0 0 0 100 | i = re
 0 0 0 0 0 0 0 1 0 99 | j = ro

```

Fig. 10. Confusion Matrices for Spectral Flatness Measure Features Classification.

## Conclusions

Five experiments were conducted for determining the music genre of a specific audio file. The extracted features varied in each experiment in order to determine which one was more suited to the dataset used. The five classifiers provided different results based on the extracted features and these were put to test with well known machine learning tools and music analysis frameworks like WEKA and MARSYAS, and also with an analysis system developed on top of the MARSYAS framework.

The results show that satisfactory results can be obtained even from the simplistic approaches as Naïve Bayes classification, but better results were obtained using more advanced techniques. The fact that the nearest neighbor produced very good results doesn't mean that it will have the same behavior on another dataset.

Improvements on the presented methods can be obtained by testing these methods on a broader dataset and determining the intrinsic influences of each genre on another.

The conclusions of these influences can have a more meaningful sense from the social point of view like blues and its derivatives and we can find very unlikely results like death metal having roots in jazz music.

## REFERENCES

- [1] Howes, F. *Man Mind and Music*. Marin Secker & Warbug LTD., 1948.
- [2] Ismir. <http://www.ismir.net/> (Visited on 2012/01/23)
- [3] Mirex. [http://www.music-ir.org/mirex/wiki/MIREX\\_HOME](http://www.music-ir.org/mirex/wiki/MIREX_HOME) (Visited on 2012/01/23)
- [4] J. Saunders, *Real-time discrimination of broadcast speech/music*, in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '96), vol. 2, pp. 993–996, Atlanta, Ga, USA, May 1996.
- [5] E. Scheirer and M. Slaney, *Construction and evaluation of a robust multifeature speech/music discriminator*, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97), vol. 2, pp. 1331–1334, Munich, Germany, April 1997.
- [6] J. T. Foote, *A similarity measure for automatic audio classification*, in Proceedings of the AAAI Spring Symposium on Intelligent Integration and Use of Text, Image, Video, and Audio Corpora, Stanford, Calif, USA, March 1997.
- [7] Z. Liu, J. Huang, Y. Wang, and I. T. Chen, *Audio feature extraction and analysis for scene classification*, in Proceedings of the 1st IEEE Workshop on Multimedia Signal Processing (MMSP '97), pp. 343–348, Princeton, NJ, USA, June 1997.
- [8] T. Zhang and C.-C. J. Kuo, *Hierarchical classification of audio data for archiving and retrieving*, in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '99), vol. 6, pp. 3001–3004, Phoenix, Ariz, USA, March 1999.
- [9] G. Williams and D. P. W. Ellis, *Speech/music discrimination based on posterior probability features*, in Proceedings of the 6th European Conference on Speech Communication and Technology (EUROSPEECH '99), pp. 687–690, Budapest, Hungary, September 1999.
- [10] K. El-Maleh, M. Klein, G. Petrucci, and P. Kabal, *Speech/music discrimination for multimedia applications*, in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '00), vol. 6, pp. 2445–2448, Istanbul, Turkey, June 2000.
- [11] A. Bugatti, A. Flammini, and P. Migliorati, *Audio classification in speech and music: a comparison between a statistical and a neural approach*, EURASIP Journal on Applied Signal Processing, vol. 2002, no. 4, pp. 372–378, 2002.
- [12] L. Lu, H.-J. Zhang, and H. Jiang, *Content analysis for audio classification and segmentation*, IEEE Transactions on Speech and Audio Processing, vol. 10, no. 7, pp. 504–516, 2002.

- [13] J. Ajmera, I. McCowan, and H. Bourlard, *Speech/music segmentation using entropy and dynamism features in a HMM classification framework*, *Speech Communication*, vol. 40, no. 3, pp. 351-363, 2003.
- [14] J. J. Burred and A. Lerch, *Hierarchical automatic audio signal classification*, *Journal of the Audio Engineering Society*, vol. 52, no. 7-8, pp. 724-739, 2004.
- [15] J. G. A. Barbedo and A. Lopes, *A robust and computationally efficient speech/music discriminator*, *Journal of the Audio Engineering Society*, vol. 54, no. 7-8, pp. 571-588, 2006.
- [16] J. E. Muñoz-Expósito, S. G. Galán, N. R. Reyes, P. V. Candeas, and F. R. Peñna, *A fuzzy rules-based speech/music discrimination approach for intelligent audio coding over the Internet*, in *Proceedings of the 120th Audio Engineering Society Convention (AES '06)*, Paris, France, May 2006, paper number 6676.
- [17] E. Alexandre, M. Rosa, L. Caudra, and R. Gil-Pita, *Application of Fisher linear discriminant analysis to speech/music classification*, in *Proceedings of the 120th Audio Engineering Society Convention (AES '06)*, Paris, France, May 2006, paper number 6678.
- [18] F. Pachet and D. Cazaly, *A taxonomy of musical genres*, *RIAO '00: Content-Based Multimedia Information Access*, 2000.
- [19] B. Logan, *Mel-Frequency Cepstral Coefficients for music modeling*, *ISMIR '00: International Symposium on Music Information Retrieval*, 2000.
- [20] D. Turnbull, *Automatic music annotation*, Department of Computer Science, UC San Diego, 2005.
- [21] Mangatune. <http://tagatune.org/Magnatagatune.html> (Visited on 2012/01/23).
- [22] MARSYAS. <http://marsyas.info/> (Visited on 2012/01/23).
- [23] WEKA. <http://www.cs.waikato.ac.nz/ml/weka/> (Visited on 2012/01/23).
- [24] [http://www2.cs.uregina.ca/~hamilton/courses/831/notes/confusion\\_matrix/confusion\\_matrix.html](http://www2.cs.uregina.ca/~hamilton/courses/831/notes/confusion_matrix/confusion_matrix.html).