Automatic Music Classification Problems

George Mitri, Alexandra L. Uitdenbogerd and Vic Ciesielski
Department of Computer Science, RMIT University
GPO Box 2476V, Melbourne 3001, Australia
+613 9925 4115
alu@cs.rmit.edu.au*

Abstract

Attempts to categorise music by extracting audio features from a sample have had mixed results. Some categories such as classical are easy to identify but attempts to distinguish between various types of popular music yield poor results. Part of the difficulty is that humans also disagree with each other when classifying music. We report on experiments that compare human classification of music samples to that based on audio feature extraction and machine learning techniques. We extracted a set of audio features and applied a range of machine learning techniques to a set of 128 pieces of music. Our work demonstrates that a single feature and a simple machine learning approach achieve results that are almost as consistent as humans for the same task. Further experiments revealed an even greater inconsistency amongst humans in selecting categories for music. Using a self-organising map on the same set of pieces and features produced some meaningful song clusters, that is, pieces by the same artist or composer, or of the same genre, were grouped together. It also showed some of the same cross-genre relationships shown by the human-based classifications.

1 INTRODUCTION

The ability to identify or extract meaningful information from musical audio data is of great benefit for a range of applications, including music retrieval and recommender systems. The last four years has seen an increase in interest in the field of music information retrieval on audio data. The state of the art consists of matching on features to classify audio into a small number of groups (for example Welsh et al. (Welsh, Borisov, Hill, von Behren & Woo 1999)), identifying a specific recording via digital signature matching (Haitsma & Kalker 2002), or identifying a work based on its entire structure (Foote 1999, Foote 2000). To match on melody against a large collection however, is still beyond the capabilities of practical systems.

Typical features extracted for audio matching are summaries of frequencies found in the audio sample, mean frequency values such as the centroid and approximations of rhythm. Matching music on these types of features provides us with the opportunity of locating works in a similar style or mood to those that a user likes (Uitdenbogerd & van Schyndel 2002). Thus the technique can be used to enhance music recommender systems.

Our goal in this research was to find out more about music classification by genre. Can it be successfully automated? What extracted audio features help when classifying pieces of music? How does an automated music classifier compare to human classification? Our yardstick for success is human classification of music, as there is no clear undisputed definition of genre for each piece of music, only human judgements. Our experiments reveal that humans disagreed with each other about 48% of the time for the data set and set of categories used in our research. Further, a single feature based on the amplitude of frequencies in a particular range was an excellent predictor of category. Using a self-organised map revealed some meaningful clusters, in particular, classical piano works were clustered together, as were ballads by the Beatles. Other works had less obvious groupings.

In this paper, we review related work, discuss the machine learning and feature extraction techniques used in our experiments, then report on the experiments themselves. This is followed by discussion of the implications of the experimental results.

2 RELATED WORK

On a computer, music information can be represented in one of two ways - by symbolic representation of notes, or by sampling and encoding an analogue signal that captures a recorded music performance, used in file formats such as the pule-coded modulation (PCM) format found in .wav files. The MP3 file format also uses this representation of music information to encode its files, but using psychoacoustic analysis, can reduce the space required to store the information.

Early work on extracting information from musical audio data concentrated on transcription. As this is a task that is quite difficult to achieve, the majority of research in this field worked on simple examples of data such as a single instrument, or working with a combination of audio and MIDI data to produce a MIDI transcription of a specific performance of a known work (Scheirer 2000). Other approaches have led to partial transcription, such as isolating the bass line (Hainsworth & Macleod 2001). Another related application is instrument detection, which currently concentrates on detecting instruments playing a single note in isolation (Fujinaga & MacMillan 2000).

A typical approach to extracting features from audio is to take Fourier Transforms from segments of an audio file, and then extract audio characteristics such as differences in pitch and average intensities of certain frequency bands. Further processing of this information can yield indexed measurements indicating characteristics such as the volume of an audio sample,

*Contact author for all enquiries

and how much noise compared to pure tone a sample has. Features can also be extracted out of the raw audio files without further processing, one example being the calculation of zero-crossings, another being the extraction of volume for a particular segment. Features can also be calculated from wavelet or other transforms in a similar fashion.

Welsh et al. applied feature extraction to a k-nearest neighbour algorithm with some success on a small collection (Welsh et al. 1999). Features indicative of the tonality of the song, the average volume level and the amount of noise present in the audio signal were used to index 7,000 MP3 songs. Then a search engine client allowed queries based on song samples. To test song classification according to genre, they manually divided 100 albums into seven categories: rock, pop, folk, electronic, indie, classical, and soul. All songs of each album were placed in the category associated with the album, giving 1225 songs in all. Tonal features were taken over the entire song length, whereas rhythm features were based on analysis of three samples of ten seconds each. Each song was represented by 1248 features. Matching had a 30–62% success rate depending on the category and features used.

In the work carried out by Tzanetakis et al.(2001) , features that were indicative of rhythm and music texture were extracted from 1,000 songs. Each of these feature sets was tagged with a genre descriptor. A Gaussian classifier was then used to partition the feature sets into categories. In their experiment on classification, the six categories selected were classical, country, disco, hip-hop, jazz and rock. Thirty-second samples were used from each song for feature extraction, 50 songs for training for each classifier. The resulting genre classifier had an accuracy of approximately 56%. In later work they applied pitch histograms to the task of classification, using 150 seconds of each of 100 pieces, and classifying into the five categories, electronica, classical, jazz, Irish folk, and rock. They achieved up to 70% accuracy for the set task, however, the music appears to have been selected to allow greater discrimination than achieved in their earlier experiments, so that it isn't certain whether the techniques or the data-set is responsible for the improvement.

The Self-Organizing Map (SOM) (Hollimin 1996, Kohonen, Hynninen, Kangas & Laaksonen 1996) is a neural network architecture loosely based on some aspects of biological brain function. The SOM has a very useful application in unsupervised clustering. Clustering is the term used to describe the segmentation of data when nothing is known about the 'class' of the data. In this example, there will not be tagged information defining a song as being a song in the 'rock' or 'pop' classification. Instead, the SOM will cluster similar sounding songs based on feature vectors extracted from MP3 audio files.

Work with clustering similar songs has been carried out by Pampalk et al . Their approach to feature extraction differed from (Welsh et al. 1999) in that they applied Fourier Transforms to the dataset and then applied critical band analysis to transform the Fourier Transform into a set of features that is more representative of human hearing, and then used Principal Component Analysis to reduce the dimensionality of the feature sets. Using a set of 3940 samples from 359 songs revealed clustering of samples from the same song.

In this section we discussed the systems used by different researchers and their experiments. In the next section we discuss the audio features that we used for our experiments.

3 AUDIO FEATURE EXTRACTION FOR MUSIC CATEGORISATION

To successfully categorise music using automatic means requires the extraction of salient features from the audio data. When humans categorise music they somehow use high-order information such as the tempo or speed, instrumentation, type of beat, tonality and vocal style as indicators. Determining these automatically from raw audio data is rather difficult. Currently, simpler features that approximate this high-order information are used. We divide these into two categories of features, rhythm and spectral. In this section we discuss the features of these two categories that we used in our experiments, including several novel features, one of which was shown to be useful for classification.

3.1 Rhythm-Based Features

Rhythm and tempo, are characteristics of music that are difficult to extract reliably. They have also been shown to be a one of the most important features in terms of humans identifying similar songs by style (Uitdenbogerd & van Schyndel 2002).

One approach to rhythm extraction was proposed by Jonathan Foote (Foote 1997). He segmented an audio sample into frames and parameterised the data using Fourier transforms, then used a cosine similarity measure to determine the similarity between two of these parameterised frames. He then constructed a two-dimensional similarity visualisation using these values and used an auto-correlation function along the diagonal of this visualisation to come up with a tempo estimate.

For this project, a simpler approach was taken to rhythm extraction. It was clear from looking at Fourier transforms of the samples that in pieces where there existed a clear, salient beat, there was a periodicity about the intensities of the audio signals in the low frequency range (0-100Hz) and in the high frequency range (12000-14000Hz).

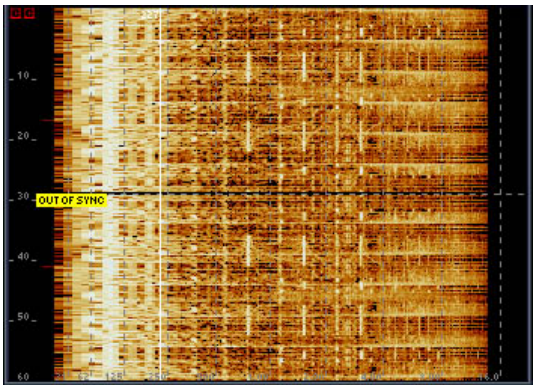

Figure 1: Fourier Transform of Chemical Brothers Song - Music: Response. The vertical axis represents frequency, the horizontal axis represents time slices and the brightness shows the intensity of the component frequencies within each time slice.

Figure 1, shows a spectrum analysis of a Chemical Brothers song, *Music: Response*. When looking at at the low frequency range of this song, a periodicity in the power is visible. This regularity in loudness can also be seen in the upper bands of the spectrum analysis. We hypothesized that it would be possible to get indexed measurements of how fast or slow the song is by looking at characteristics of these two bands.

In Figure 2, a similar spectrum analysis is carried out on a J.S. Bach piece. We can see that the peaks

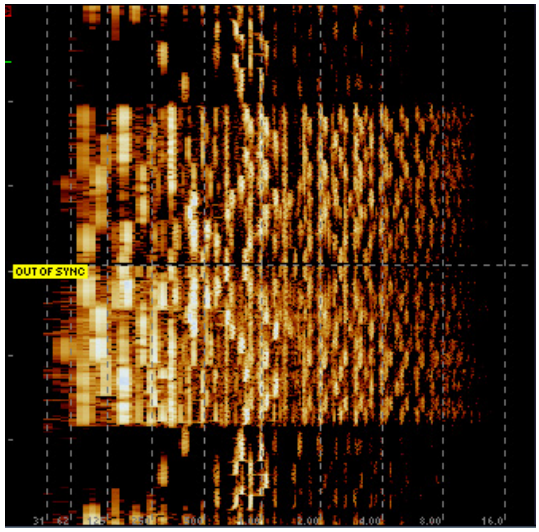

Figure 2: Fourier Transform of a Bach piece

in the upper and lower bands are nowhere near as tightly regulated as in Figure 1.

One of the first features extracted from this was a measure of the maximum and minimum intensities of average power of these two bands. Following on from that, there were measurements taken of how many times the average power of a band was within a 10% threshold of the maximum power of that band and a measure of how many peaks and troughs there were in those two bands and how widely they were spaced apart. A listing of all the rhythm features we extracted can be found in Table 6, along with descriptions of what each of these features represented.

3.2 Spectral Features

After rhythm features were extracted from the audio, the next step was to look at a set of features that were representative of the "feel" of the audio sample. Spectral features were considered (Tzanetakis, Essl & Cook 2001) as a way of parameterizing audio according to its recorded characteristics. These features indicate the colour and texture of tone of the song, particularly in regard to instrumentation.

Below is a list of the features we implemented that were based on the work presented in Tzanetakis et al. (Tzanetakis et al. 2001).

Note that $M[f]$ is the magnitude of the frequency f at bin Min the Fourier Transform of the data.

The **Centroid** is calculated as:

$$C = \frac{\sum_{i=1}^{N} fM[f]}{\sum_{i=1}^{N} M[f]}$$

The centroid is a measure of spectral brightness. That is, it is a measure of where most of the volume of the sample lies, in terms of frequency, on the Fourier transform.

Rolloff is the value R such that:

$$\sum_{i=1}^{R} M[f] = 0.85 \sum_{i=1}^{N} fM[f]$$

The rolloff is a good measure of spectral shape. That is, it is a measure of how the frequencies distribute themselves along the Fourier transform.

The **Flux** is calculated to be:

$$F = \|M[f] - M_p[f]\|$$

Flux is a measure of how much the Fourier spectrum changes.

We also implemented features that used **zero-crossings**. Zero-crossings are calculated by counting the number of times the untransformed waveform crosses from a positive to a negative value. Zero-crossings are useful for detecting the noise in a signal.

The spectral features we implemented are listed in Table 9.

4 EXPERIMENTS

We wished to determine which features are most effective for classifying music into several categories, and which machine learning techniques were most effective for this purpose.

4.1 The Collection

The 128 songs chosen for the collection were garnered from a wide variety of audio compact discs. The collection consists of songs of 32 different artists. A 10 second sample starting from exactly one minute into each song was extracted digitally from CD and compressed in the MP3 format. This sample length was chosen as appropriate given the resources available and yet still much longer than that required by humans for identification of style. The samples were kept at the volume level of the original CD recording.

4.2 Human Classification of Music

To see how consistent humans were at classifying a 10 second sample of a song, a small survey was conceived and distributed to four subjects. The survey was distributed with a CD of 64 of the 10-second samples, randomly drawn from the collection of 128 songs. Each participant was asked to classify each of the 10 second samples on one CD into one of seven genres: Rock, Pop, Classical, Hiphop, Electronic, Folk, and Dance. These categories were chosen by the researchers as representative of the collection and the same number of categories as that used for other work in the field (Welsh et al. 1999). Each CD was organised so that a set of 32 of the 64 samples was common to two discs. That is, the collection was divided into 4 sets of 32 samples, with each participant receiving two such sets, and sharing a set of 32 in common with two other participants.

The survey asked the subjects to specify their familiarity with music — that is, how much music tuition they have received — which artists or composers they were most familiar with, and a short paragraph about their motivation behind their classifications.

The human classifications were assessed for consistency in order to provide a rough confidence rating for computer based classification. Consistency of classification was measured using a disagreement calculation: a simple summing up of all the instances where two subjects disagreed on classification, divided by the total number of songs.

4.2.1 Results/Discussion

Table 1 shows the percentage of disagreement amongst participants for each set of 32 samples. The results for this were surprising. Overall, there was 48% disagreement between two subjects in relation to classification similarity.

From the survey comments, subjects found it very difficult to classify some of the samples. Two of the respondents replied that it was very difficult to classify songs by the "Magnetic Fields". This was borne out in the results - 5 of the 8 Magnetic Fields songs were

Table 1: Genre disagreement between participants

Set 1	Set 2	Set 3	Set 4	Overall
41%	38%	59%	53%	48%

Table 2: ZeroR Classifications and Classification Accuracy

	Classification		
	1	2	3
Accuracy:	23.43%	31.25%	32.81%
Classification:	Pop	Rock	Rock

classified inconsistently, with respondents not knowing whether to classify the pieces as "pop" or "rock".

In a further analysis of the genre disagreement we found that dance and electronic were frequently interchanged, and folk was very inconsistently classified. In Table 5 we show a confusion matrix of the two human-assigned genre labels for each song. Table 3 shows the percentage of agreement for each category as a proportion of the total number of songs that were placed in the category by at least one human classifier.

In one of several follow-up experiments, we asked two semi-professional musicians to assign music categories to the same pieces (half each), with no restriction on the nature of the categories. Both participants used many categories (28 and 11 respectively), with only 4 categories being identical: pop, rock, funk and classical.

In a second experiment on choosing categories for music without referring to specific pieces, there was no clear agreement amongst 37 participants, even when only two categories were requested. The task was perceived as quite difficult, with difficulty generally increasing with the number of categories requested (in the range 2–10). We intend to explore the issues related human classification of music further.

4.3 Automatic Classification with Training

In content-based audio categorisation, the goal is to extract features that emphasize the most important aspects of the audio sample and that are meaningful enough to allow classification and clustering algorithms to organise a music collection into pertinent genres and not be too expensive to compute. The following sections describe the feature extraction process employed here, which transforms 10 seconds worth of audio sample per song into 58 features which represent rhythmic and spectral characteristics of each sample.

Using seven samples of various styles for exploratory and development work, feature extraction algorithms were implemented, tested, and refined. Features were extracted using the Snack toolkit, which is a set of libraries that "plug-in" to the Tcl Scripting Language, and allows for easy audio manipulation in a wide variety of audio formats, including MP3.

The Weka Data Mining toolkit (Witten & Frank 2000) was used for classification on the collection of 128 songs, and as a tool to gain insight into the data. The ZeroR, OneR, and J48 Classification algorithms were used, as well as a greedy feature selection algorithm. These algorithms implement classifiers of increasing complexity. ZeroR simply classifies an unknown case according to the majority class of the training data. For example, if pop is the most common category, all items are classified as pop. OneR finds a rule based on the single attribute that most

Table 3: Percentage of human classifications that were consistent for each music category. Also shown is the total number of pieces that were given that category by at least one person.

Category	Percentage	Total
Classical	75	12
Dance	45	22
Electronic	15	26
Folk	14	7
Hip-hop	80	10
Pop	29	49
Rock	48	54

Table 4: OneR Classification Accuracy and Key Attribute Selected

Classification	Attribute	Accuracy
1	maxamplow	45.31%
2	maxamplow	53.13%
3	maxamplow	40.63%

accurately classifies the training data. J48 is an implementation of the C4.5 decision tree classifier, one of the most robust and widely used classification methods. Greedy feature selection is a method of finding the best subset of features for classification. All of the classifiers were trained and tested using 10-fold cross validation. In addition, a number of feature selection methods were used, with the aim of removing any irrelevant or redundant features from the original 58. New J48 decision trees were then obtained from the reduced feature sets. All the classifiers were trained and tested using 10-fold cross-validation of the data.

The classifications garnered from the survey responses were compiled into two complete sets of classifications, with which the feature sets were then tagged. We also devised an initial set of classifications relating artist to genre. This set is labelled Classification 1, and the remaining two classifications were garnered from the human survey (Classification 2 and Classification 3).

ZeroR was run as an initial classification algorithm to gain a baseline accuracy measurement for the three classifications, and to look at how skewed the classification was. Table 2 shows the classification selected by ZeroR, and the accuracy in the classification chosen.

Table 4 shows the accuracy in the classification when using the OneR classifier and which feature was chosen.

The J48 decision tree classifier was used with all settings kept at defaults, except that we tested the classifier both with and without error-reduction pruning of the decision tree. Table 7 shows the accuracy of classification that the J48 algorithm obtained.

For the feature selection we used greedy hill-climbing to select possible feature subsets and *Correlation-Based Feature Selection* (Hall 1999) to evaluate the subsets. Table 10 shows the results of this feature subset selection.

Using the recommendations for each of the three classifications (presented in Table 10), we limited the J48 Decision Tree algorithm to the selected feature subsets. Unpruned classification accuracy improved, but was still less than that achieved with the maxamplow feature in isolation.

Table 5: Confusion Matrix for Human Classifications 2 and 3

Category	Classical	Dance	Elec.	Folk	Hip-hop	Pop	Rock
Classical	9	0	1	1	0	1	0
Dance		10	10	0	0	1	1
Electronic			4	0	1	7	3
Folk				1	0	3	2
Hip-hop					8	1	1
Pop						14	22
Rock							26

Table 6: Rhythm Based Features Used

Feature Name	**Description**
maxamplow	max power of lower band frequencies
minamplow	min power of lower band frequencies
peaklow	number of times power of lower band is within 20% of max power
peakabslow	number of times power of lower band is within 10 levels of max power
maxamphi	max power of higher band frequencies
minamphi	min power of higher band frequencies
peakhi	number of times power of higher band is within 20% of max power
peakabshi	number of times power of higher band frequency is within 10 levels of max power
cons_peaklow	number of times power of lower band passes 15% of max power
meanlowdist	mean distance of peaks in lower band
stdlowdist	standard deviation distance of peaks in lower band
maxlowdist	maximum distance between peaks in lower band
minlowdist	minimum distance between peaks in lower band
meanlowdist_b	mean distance between peaks in lower band
stdlowdist_b	std dev of distance of peaks in lower band
maxlowdist_b	max distance between peaks in lower band
minlowdist_b	min distance between peaks in lower band
cons_peakhi	number of times power of higher bands passes 15% of max power
meanhidist	mean distance of peaks in higher band
stdhidist	std deviation of distance of peaks in higher band
maxhidist	maximum distance of peaks in higher band
minhidist	minimum distance of peaks in lower band
meanhidist_b	mean distance between peaks in higher band
stdhidist_b	standard deviation of distance between peaks in higher band
maxhidist_b	maximum dist between peaks in higher band
minhidist_b	minimum dist between peaks in higher band

Table 7: J48 Classification Accuracy

Classification	J48 (Pruned)	J48 (Unpruned)
1	42.18%	39.84%
2	37.5%	43.75%
3	35.94%	35.94%

Table 8: Results for J48 Classification with Selected Features

Classification	J48 (Pruned)	J48 (Unpruned)
1	40.63%	42.96%
2	41.41%	45.31%
3	35.93%	36.71%

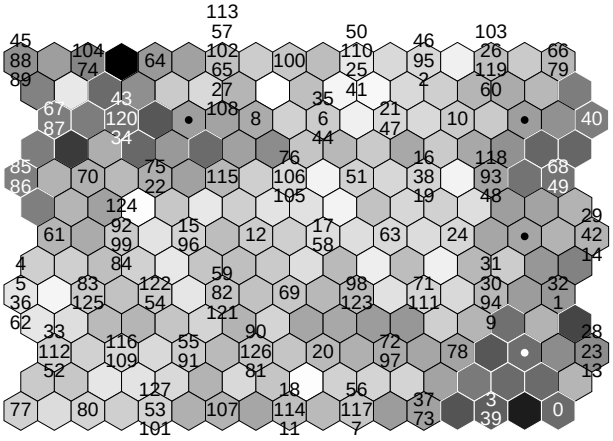

Figure 3: SOM 9x7 Dimension Hexagonal Topology Bubble Neighbourhood

4.4 Discussion

The best accuracy achieved was 53%. This is considerably better than guessing which would only achieve an accuracy of $(128 * 100/7) \approx 18\%$. Given the substantial disagreement between human classifications, this result is about as good as could be expected and is arguably consistent with the 56% obtained by Tzanetakis et al.

One interesting finding to come out of the classification experiments was that the accuracy of the OneR classifier, which chose one feature **maxamplow**, surpassed the J48 classifier either with that classifier working with all features, or a selected subset of all the features.

4.5 Self-Organising Maps

Clustering was done with the SOMPAK Self-Organising Map package. The training of the SOM was done on 11x11 and 9x7 grids, and neighbourhood functions were alternated between Gaussian and bub-

Table 9: Spectral Features Used

Feature Name	Description
cent1m	mean centroid for 5000-10000 samples into file
cent1std	stddev centroid for 5000-10000 samples into file
roll1m	mean rolloff for 5000-10000 samples into file
roll1std	stddev rolloff for 5000-10000 samples into file
flux1m	mean flux for 5000-10000 samples into file
flux1std	stddev flux for 5000-10000 samples into file
cent2m	mean centroid for 25000-30000 samples into file
cent2std	stddev centroid for 25000-30000 samples into file
roll2m	mean rolloff for 25000-30000 samples into file
roll2std	stddev rolloff for 25000-30000 samples into file
flux2m	mean flux for 25000-30000 samples into file
flux2std	stddev flux for 25000-30000 samples into file
cent3m	mean centroid for 50000-50000 samples into file
cent3std	stddev centroid for 50000-50000 samples into file
roll3m	mean rolloff for 50000-50000 samples into file
roll3std	stddev rolloff for 50000-50000 samples into file
flux3m	mean flux for 50000-50000 samples into file
flux3std	stddev flux for 50000-50000 samples into file
cent4m	mean centroid for 75000-80000 samples into file
cent4std	stddev centroid for 75000-80000 samples into file
roll4m	mean rolloff for 75000-80000 samples into file
roll4std	stddev rolloff for 75000-80000 samples into file
flux4m	mean flux for 75000-80000 samples into file
flux4std	stddev flux for 75000-80000 samples into file
zerocross1m	mean zerocrossings for 5000-10000 samples in
zerocross1std	stddev zerocrossings for 5000-10000 samples into the file
zerocross2m	mean zero crossings for 25000-30000 samples into file
zerocross2std	stddev zero crossings for 25000-30000 samples into file
zerocross3m	mean zero crossings for 50000-50000 samples into file
zerocross3std	stddev zero crossings for 50000-50000 samples into file
zerocross4m	mean zero crossings for 75000-80000 samples into file
zerocross4std	stddev zero crossings for 75000-80000 samples into file

Table 10: Attributes Automatically Selected by CFS

Features Selected		
Classification 1	**Classification 2**	**Classification 3**
maxamplow	maxamplow	maxamplow
minamplow	maxamphi	minamplow
cent1m	minamphi	maxamphi
cent4std	peakhi	minamphi
zerocross3std	cent1std	cent1m
	cent4std	cent2m
	zerocross4m	cent4std
		zerocross1std
		zerocross2std

ble. In the experiments, training was completed in two phases. In the first phase, the learning rate, $\alpha(0)$, was set to 0.5, and the initial kernel size, $\sigma(0)$, was set to 10. For the second pass, the learning rate was set to 0.2, and the kernel size was set to 3. Both a rectangular and a hexagonal topology was used. Due to space restrictions we only report on the hexagonal topology results here, but the clustering was similar for both.

Figures 4 and 3 show two-dimensional visualisations including song numbers. The visualisation contains two types of hexagons. The first type contain numbers or dots and have an associated code book vector produced by the SOM algorithm. The numbers in the hexagons represent songs whose feature vector is very close to the code book vector. These songs are thus considered to be very close to each other. For the hexagons that contain a dot there is no song associated with the codebook vector. The second type of hexagons contain no numbers or dots and their grey levels represent the size of the difference between code book vectors. Black indicates a large difference and increasingly lighter shades of grey indicate increasingly smaller differences. A very significant property of the SOM is that code book vectors which are physically close to each in the 2D map are also close to each other in the 58D feature space. Thus in analysing Figure 4 we can conclude that songs 45,88,89 are very similar to each other and very different to songs 113,57,102,100 and 28 and 66,60 which are a long way away on the map. Also, songs 45,88,89 are much more different to songs 67,86 than songs 103,26 are to 66,60. The characteristics of figure 3 that suggest further analysis are (1) songs that have been allocated to the same hexagon and (2) songs that have been allocated to the top left and bottom right corners as the map shows that they are significantly different to their neighbours.

In Figure 4, we see several clusters. Two of the Rachmaninov études (song numbers 88 and 89) are closely clustered in the top left cell, with the other two (86 and 87) in nearby cells. The Rachmaninov piano concertos (90 and 91) are near each other but not in the same cells. In the same zone as the études are the outliers "Is You Or Is You Ain't My Baby" (45), which combines a sample of an early recording

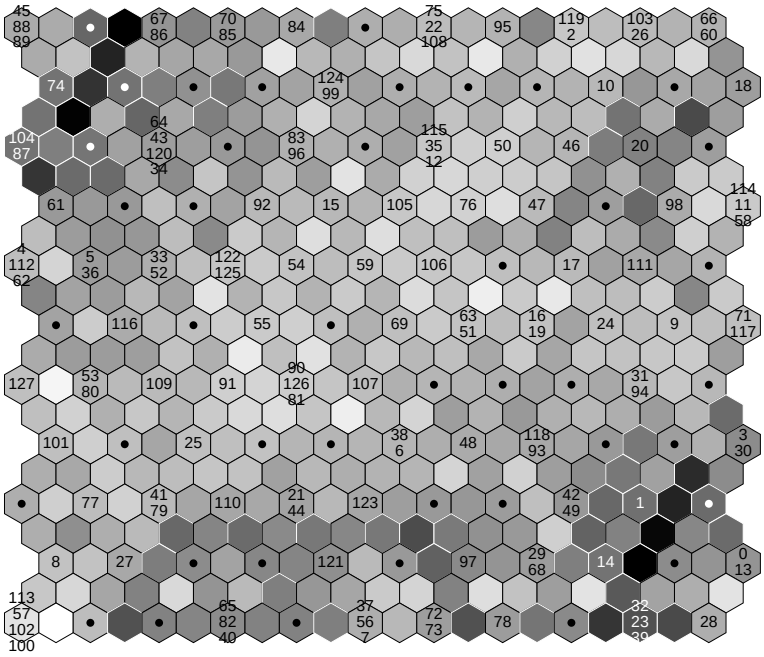

Figure 4: SOM 11x11 Dimension Hexagonal Topology Bubble Neighbourhood

with modern beats, "When my boy walks down the street" (67), in which the noise of the instruments and voice is louder than the percussion beats, and a Primal Scream track in which the beats are similarly muted compared to the other instruments (74). Also appearing is the Beatles' "Eleanor Rigby", which was labelled as "pop-classical" in a separate human classification study.

Seventy percent of the pieces that were labelled as *dance* by one person and *electronic* by another (numbers 0, 3, 13, 28, 30, 32, and 39) have been placed in the bottom right hand corner of the map. The only outlier in the area is a hip hop song (number 23). Slightly further out from this zone, but separated by a distinct boundary are pieces that were labelled as dance by at least one person (numbers 1, 14, 68).

Some of the pieces labelled *rock* are clustered to the right of the classical pieces at the top of the map. In this area are three pieces by the same artist (SOAD, songs 83–85). The Bach pieces are placed in the centre left region, and curiously, the one piece that was mislabelled classical by one human classifier was placed in close proximity to the Bach. Other close artist clusters that occur are Magnetic Fields at the top right (60 and 66), and the Pixies in the bottom middle (72 and 73).

In the bottom left hand corner, there are three *Beatles* songs stylistically similar to each other (the acoustic songs "Blackbird" (number 100) and "Dear Prudence" (number 102), and the gentle "Something" (number 113), grouped with the outlier Hip Hop piece "Quality Control" grouped with them (number 57).

In Figure 3 similar groupings occur, however in addition the smaller grid size has forced the creation of larger clusters. A *rock* cluster occurs in the centre left region, consisting of songs 84, 92, 99 and 124. All but the last of these was classed as *rock* by both judges. The fourth was classed as folk by one judge and rock by the other. Several of the pieces surrounding this cluster are also rock songs (numbers 4, 5, 15, 61, 70, 85, 83, 122). Another nearby cluster contains three songs, two of which were classed as rock by both assessors, and one that was classed as pop by one and rock by the other assessor. Once again the dance-electronic pieces are largely clustered in the bottom right corner and hip-hop is widely dispersed. Other clusters of note are songs 22 and 75, which both received a pop and an electronic human classification, songs 16 and 19 which were classed as folk by one assessor, and are quite close to song 48 which was also classed as folk by a single assessor. The Beatles songs "For No One" (105) and "Girl" (106), both rated as pop by both assessors, have been forced into the same cell in this smaller grid.

In other experiments not shown here we used the smaller feature sets identified in Table 10. These resulted in some similar clusterings to those produced by the full feature set, but no clearly defined zones were visible on the map.

4.6 Discussion

From the analysis above, it is clear that there is a lot more work to do in finding feature sets that are perceptually similar. In all of the SOM visualisations seen, there are visible clusters of songs that sound similar, such as the cluster of Rachmaninov piano concertos that is clustered around the top left hand corner in all of the SOM visualisations. However, for hip hop pieces, such as those performed by *Jurassic 5* and *Blackalicious*, there was scattering across the SOM, and into locations on the map typically occupied by softer songs.

One thing we have not tried is clustering songs based on a subset of the features extracted that can be determined by feature selection algorithms, such as those in Table 10. This might yield better results from the SOM clustering.

Unlike many automatic classification tasks, the classification of music is not precise. People disagree as to what category a piece of music belongs to, and even which categories should be used for the purpose. Despite this, the SOM showed some clear groupings on genre for some genres.

5 CONCLUSIONS

Unlike most other work on automatic classification of musical audio, we restricted ourselves to a single 10-second compressed sample, and simple features. Despite this, we achieved a 53% accuracy on a set of 128 such samples with a single feature, which was not

too different to the level of agreement between two sets of human judgements (52%) on the same task.

We found that applying machine learning to a single attribute (maxamplow) gave higher agreement with human classification than did a more complex set of features. From the experimental results, it was observed that some of the rhythm-based features that we extracted, in particular that of **maxamplow**, discriminated between genre categories more accurately than all or a selected subset of the features used. Features such as **minamphi**, **maxamphi**, and **minamplow**, which were relatively simple to compute were also good at discriminating between genre categories.

In an experiment with self-organising maps and the same sets of features, it was found that clear clusters were formed for dance-electronic music and music with muted or non-existent percussion. Several other artist and composer clusters, and general regions of music of a similar style were formed, however, hip-hop music was widely dispersed through the map.

From our experiments we conclude that it is indeed possible to classify audio using a short sample of 10 seconds. The features used here were fairly simple and still obtained results that are on a par with humans at the same task. However, some genres were more difficult to separate and may require more sophisticated techniques. Further, as genres are not so clearly defined by humans, more work is required in order to better model the way music is manually labelled and to determine how automatic classification systems should be evaluated.

Acknowledgements

We thank the volunteers that helped with our experiments, and Ron Van Schyndel for his expert advice.

References

Fingerhut, M., ed. (2002), *International Symposium on Music Information Retrieval*, Vol. 3, Paris, France.

Foote, J. (1997), A similarity measure for automatic audio classification, *in* 'Proc. AAAI Spring Symposium on Intelligent Integration and Use of Text, Image,Video, and Audio Corpora'. **URL:** *citeseer.nj.nec.com/foote97similarity.html*

Foote, J. (1999), Visualizing music and audio using self-similarity, *in* 'Proc. ACM International Multimedia Conference', Orlando Florida, USA, pp. 77–80.

Foote, J. (2000), ARTHUR: Retrieving orchestral music by long-term structure, *in* D. Byrd, J. S. Downie, T. Crawford, W. B. Croft & C. Nevill-Manning, eds, 'International Symposium on Music Information Retrieval', Vol. 1, Plymouth, Massachusetts.

Fujinaga, I. & MacMillan, K. (2000), Realtime recognition of orchestral instruments, *in* 'Proc. International Computer Music Conference', ICMA, Berlin, Germany, pp. 141–143.

Hainsworth, S. W. & Macleod, M. D. (2001), Automatic bass line transcription from polyphonic music, *in* 'Proc. International Computer Music Conference', ICMA, Havana, Cuba.

Haitsma, J. & Kalker, T. (2002), A highly robust audio fingerprinting system, *in* M. Fingerhut, ed., 'Third International Conference on Music Information Retrieval', Paris, France, pp. 107–115.

Hall, M. (1999), Correlation based Feature Selection for Machine Learning, PhD thesis, University of Waikato.

Hollimin, J. (1996), Process modelling using the self organising map, Master's thesis, Helsinki University of Technology.

Kohonen, T., Hynninen, J., Kangas, J. & Laaksonen, J. (1996), SOM PAK: The self-organizing map program package, Technical report, Helsinki University of Technology, Laboratory of Computer and Information Science. `http://www.cis.hut.fi/research/som_pak/`.

Pampalk, E. (2001), Islands of music: Analysis, organisation, and visualisation of music archives, Master's thesis, Austrian Research Institute for Artifical Intelligence.

Rauber, A., Pampalk, E. & Merkl, D. (2002), Using psycho-acoustic models and self-organising maps to create a hierarchical structuring of music by sound similarity, *in* Fingerhut (2002), pp. 71–80.

Scheirer, E. D. (2000), Music Listening Systems, PhD thesis, MIT, Massachusetts.

Tzanetakis, G., Ermolinskyi, A. & Cook, P. (2002), Pitch histograms in audio and symbolic music information retrieval, *in* Fingerhut (2002).

Tzanetakis, G., Essl, G. & Cook, P. (2001), Automatic musical genre classification of audio signals, *in* J. S. Downie & D. Bainbridge, eds, 'International Symposium on Music Information Retrieval', Vol. 2, Bloomington, Indiana, USA.

Uitdenbogerd, A. L. & van Schyndel, R. G. (2002), A review of factors affecting music recommender success, *in* M. Fingerhut, ed., 'Third International Conference on Music Information Retrieval', Paris, France, pp. 204–208.

Welsh, M., Borisov, N., Hill, J., von Behren, R. & Woo, A. (1999), 'Querying large collections of music for similarity'. Technical Report UCB/CSD00 -1096, U. C. Berkeley Computer Science Division. **URL:** *citeseer.nj.nec.com/welsh99querying.html*

Witten, I. H. & Frank, E. (2000), *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann, San Francisco. http://www.cs.waikato.ac.nz/ ml/.