Monash University

Enhanced Polyphonic Music Genre Classification Using High Level Features

This thesis is presented in partial fulfillment of the requirements for the degree of Master of Information Technology (Minor Thesis) at Monash University

By:
Arash Foroughmand Arabi

Supervisor:
Guojun Lu

Year:
2009

Copyright Notices

Notice 1

Abstract

The task of classifying the genre of polyphonic music signals is traditionally done using only low level features of the signal. It has been suggested in the literature that high level musical features are also good source of information on music genre. In this thesis high level features have been applied to improve and enhance the task of music genre classification. Features that capture high level conceptual harmonic, pitch, and rhythmic contents of the music are proposed to be used in conjunction with low level features to increase the classification accuracy of polyphonic music signal genre classification techniques.

In this thesis chord, chord progression, and beat features are used in conjunction with timbral features to improve the music genre classification methods. Since chord and chord progressions are perceptual concepts and differ from timbral and beat features in nature, they cannot be directly integrated with each other. Therefore specific techniques have been developed in this thesis to capture the high level information of chord and chord progressions into feature vectors so they can be integrated with beat and timbral features.

To capture chord information, a statistical chord feature vector is proposed. And to capture chord progression information, a technique called Chord Mining is developed. In this technique, chord progression content of the songs are manifested in genre probability descriptors calculated using a pattern matching algorithm. Our proposed method provides an improvement of 12.4% in the classification results over a commonly compared technique.

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the work of others has been acknowledged.

Signed: …….....

Name

Date

Acknowledgements

I am delighted to acknowledge and thank all those who have helped me during the course of my research in any way. I am especially thankful to Professor Guojun Lu, my supervisor, for all his help and support. I am honoured to have the privilege of experiencing his guiding and caring supervision. I express my most grateful thanks for his indefatigable support, guidance, inspiration, and belief in my work.

I would also like to thank George Tzanetakis (University of Victoria, Canada) and Peter Mcilwain (Monash University School of Music – Conservatorium, Australia) for their useful comments and helps.

Finally I would like to express my gratitude to my patient and kind wife, Monireh Taban for her unfailing and unconditional support and encouragement.

Publications

ARABI, A. F. & LU, G. (2009) “Enhanced Polyphonic Music Genre Classification Using High Level Features”. *International Conference on Signal and Image Processing Applications (ICSIPA).* in press.

Table of Contents

Chapter 1: Introduction

In the past few years the amount of digital music has increased significantly. With the rapid increase of storage capacity and processing power of digital devices and with the boom of World Wide Web and high speed broadband internet, digital music has become increasingly popular. Particularly the amount of digital music available online has increased significantly and the amount of purchases from online music stores have become incredibly popular.

According to (Bergstra et al., 2006) by 2006 Apple had sold more than 42 million iPods and more than a billion songs from their online store. According to (Turnbull et al., 2008) by 2008 last.fm[1] had a rapidly growing database of 150 million songs by 16 million artists. These numbers illustrate the need for automatic content based music annotation, indexing and retrieval systems. Manual annotation and tagging of such huge datasets are extremely costly, time consuming, and erroneous.

Genre is one of the most important descriptors of music. According to (Bainbridge et al., 2003) genre is the second most popular metadata searched by the users of music information retrieval systems after bibliographic information (such as artist, title, etc.).

[1] http://www.last.fm/

In this thesis a system is proposed which can automatically identify the music genre based on the music signal. Such system has application in automatic content based music annotation, indexing, and retrieval.

There are many systems available for automatic polyphonic music genre classification. Some of these systems classify symbolic music (e.g. MIDI[2]) while the others classify digital music (e.g. MP3) based on its signal. Here we propose enhancements on one of the prominent digital music genre classification techniques (Tzanetakis, 2007) and propose a method to improve its classification accuracy.

The task of music genre classification consists of two steps. The first step is feature extraction where specific features are extracted from the music signal. These features capture the properties of music signal. The second step is classification; where data mining algorithms are applied on the extracted features to classify the songs based on genre.

Almost all of the current techniques for music genre classification extract low level features of the signal and use them in the classification. Low level features are those features that capture low level physical signal properties. These features are usually calculated over very short timeframes and usually correspond to timbre and texture of the sound. We propose that high level conceptual music features are also important and they must be used in conjunction of the low level features to improve the classification accuracy. High level features capture perceptual concepts that are conveyed by music such as harmony, tempo, music key, etc.

In this thesis we propose rhythmic, chord, and chord progression features to capture high level musical concepts. In particular we pay more attention to chord and chord progression features. Combining the information captured from chord and chord progressions with low level features necessitates them to be in the same format and organisation as the low level features. Therefore a feature vector is extracted from the chords and chord progressions of the songs. An algorithm called ChordMiner has been developed to extract information from the chord progressions of the songs into a feature vector which can be then combined with the other features and used in the classification.

The main contributions of this thesis are:

1. Reviewing of the existing works and identifying their limitations.
2. Proposing to combine high level and low level features to enhance the state-of-the-art methods.
3. Proposing methods and developing an algorithm to capture high level chord and chord progression information into feature vectors.

[2] Music Instrument Digital Interface

4. Providing experimental studies on different combinations of features for music genre classification.

The rest of this thesis is organised in the following way:

Chapter 2 provides a comprehensive literature review. The strengths and weaknesses of existing works are also highlighted in this chapter. In Chapter 2 The details of our baseline system (Tzanetakis, 2007) is also described and our motivations for using high level features are identified.

In Chapter 3 our proposed technique is presented and discussed in details. The reasons for selecting chord based features are also highlighted in this chapter. Then the architecture of the proposed system is illustrated and each of the modules of the system is discussed.

Chapter 4 provides the details of the experimental studies on our proposed system. In this chapter our proposed techniques are evaluated and the results are analysed and discussed. Finally Chapter 5 concludes this thesis.

Chapter 2:
Literature Review

In the past 10 years a lot of attention has been paid to automatic music indexing and retrieval. Particularly there has been a lot of research in the task of automatic music genre classification. Many researchers have worked on this problem and a lot of novel systems have been developed. The annual MIREX[3] tasks have played an important role in attracting researchers to this area and other domains of music information retrieval.

In this chapter a review of the most significant works on music genre classification will be presented. For works that have directly influenced our method, a more detailed review will be given. The strengths and weaknesses of the current research trends shall also be analysed.

Perhaps the most significant and influential literature available on automatic music genre classification of polyphonic signals, is the work of Tzanetakis and Cook (2002). A search on Google Scholar reveals that this paper has been cited over 720 times in various academic publications. Tzanetakis and Cook also laid down the foundations of Marsyas[4], a continually evolving open source software program. The techniques that they proposed in their 2002 paper was used as a base or a benchmark by a majority of other works in this subject area including the work of (Lampropoulos et al., 2005, Shen et al., 2006, Lidy et al., 2007).

[3] Music Information Retrieval Evaluation eXchange (http://www.music-ir.org/)

[4] http://marsyas.sness.net/

Many researchers in this area compare their results with the results obtained using the method proposed by Tzanetakis and Cook (2002). In other words this technique has become a benchmark in the area of automatic polyphonic music genre classification.

Marsyas was originally developed by Tzanetakis and Cook in year 2000. Since 2000 (Tzanetakis and Cook, 2000), many improvements have been done to Marsyas, however the basics of the feature extraction algorithms are the ones described in (Tzanetakis and Cook, 2002). We base our work on their latest work (Tzanetakis, 2007) which is their submission to 2007 and 2008 MIREX genre classification task. Our proposed improvement to this technique is based on incorporating high level features. Particularly we focus on chord and chord progression features.

Although there have been newer techniques than those of (Tzanetakis, 2007) because of the following reasons we have selected (Tzanetakis, 2007) as the baseline of our system. Firstly this method is still one of the best performing methods. This method achieved the highest average classification accuracy in the 2008 MIREX genre classification contest. Secondly as mentioned before an open source software has been implemented by the authors giving us the chance of examining the exact setup that was described in their paper. And finally since this method is used as a benchmark by many researchers in the community it will be a more reliable comparison point both for us and other researchers who want to compare their results with ours.

In this chapter we first provide an overview of the previous psychoacoustic research on how humans classify music. Then the works of Tzanetakis and Cook are described in detailed. And finally other significant works are described and their strengths and limitations are analysed.

2.1. Psychoacoustic and music psychology research

There has been very limited psychology and psychoacoustic research on how humans perceive and classify music genre. One research work known to us in this discipline is the controversial paper of (Gjerdingen and Perrott, 1999).

This paper was presented at the annual meeting of the Society for Music Perception and Cognition (SMPC) in Northwestern University (Evanston, IL) and it never existed in print. Therefore anyone who has cited this paper must have been either present in the meeting, personally communicated with the authors, or cited the paper without reading it. According to a study by Aucouturier and Pampalk (2008) most of the researchers who have cited this paper have never even seen the paper. In this study they tracked the citations through the typographical and spelling errors in the names of the authors as appeared in the citations (Aucouturier and Pampalk, 2008).

In 2008 Gjerdingen and Perrott published a revised version of their paper in the Journal of New Music Research. However prior to this publication we had acquired a draft copy of their original paper through personal communication.

In their research they showed that humans are able to detect the genre of the music by listening to very short excerpts of music (Gjerdingen and Perrott, 2008). This means that low level features of the music which define the sound texture are enough for determining the genre of the music, because high level features cannot be realised in such short timeframes.

Beside the above mentioned study there is no other psychoacoustics research known to us examining the human ability to classify genre. Unlike music emotion classification the discipline of music genre classification seriously lacks the psychological and musicological basis. More input from the musicology and psychology can certainly help us create better and more accurate methods for music genre classification.

2.2. Works of Tzanetakis and Cook

The work of Tzanetakis (2007) is the latest implementation of (Tzanetakis and Cook, 2002). Tzanetakis and Cook (2002) concentrated mostly on the use of timbral features. These features define the texture of the music. The idea behind this comes from the aforementioned study of (Gjerdingen and Perrott, 1999). They concluded that since humans are able to identify music genre in very short time periods accurately, surface information such as MFCC (Mel-Frequency Cepstral Coefficients) which define the sound timbre and texture are enough for classifying music genre.

For genre classification Tzanetakis (2007) used five features which define the timbral texture of music. These 5 features are: MFCC, Spectral Centroid, Spectral Rolloff, Spectral Flux, and Time Domain Zero Crossing. These features were originally used for speech recognition and are mostly based on short time Fourier transform over very small frames of the sound. These features are described here in more details. The definitions presented here are from (Tzanetakis and Cook, 2002) and were used in Marsyas feature extraction algorithms.

1. Spectral Centroid

This feature represents the centre of gravity in the magnitude spectrum of Short Time Fourier Transform, it is defined as:

$$C_t = \frac{\sum_{n=1}^{N} M_t[n] \times n}{\sum_{n=1}^{N} M_t[n]} \quad (1)$$

Where $M_t[n]$ is the magnitude of Fourier transform at frame t for frequency bin n. Frames which have higher frequency have higher spectral centroid.

2. Spectral Rolloff

The frequency threshold under which 85 percent of the magnitude distribution is concentrated is the spectral rolloff. R_t is defined as:

$$\sum_{n=1}^{R_t} M_t[n] = 0.85 \times \sum_{n=1}^{N} M_t[n] \quad (2)$$

3. Spectral Flux

This feature measures the local change in the spectrum and it is defined as:

$$F_t = \sum_{n=1}^{N} (N_t[n] - N_{t-1}[n])^2 \quad (3)$$

Where $N_t[n]$ is the normalised magnitude of the Fourier transform of frame t.

4. Time domain zero crossing

This feature defines the noisiness of the signal and it is defined as:

$$Z_t = \frac{1}{2} \sum_{n=1}^{N} |sign(x[n]) - sign(x[n-1])| \quad (4)$$

Where x[n] is the time domain signal for that frame.

5. MFCC

Briefly, Mel-Frequency Cepstral Coefficients are perceptually motivated features based on short time Fourier transform. MFCC are result of performing discrete cosine transform over Mel-Frequency cleaned fast Fourier transform bins, which are the result of taking log-amplitude of the magnitude spectrum. More information on MFCC can be found at (Rabiner and Juang, 1993).

Tzanetakis and Cook in their initial paper in 2002 suggested although usually 13 coefficients are used in speech processing they found that using only the first 5 coefficients are enough. However in their latest experiment they used all the 13 coefficients (Tzanetakis, 2007). Therefore we will also use all the 13 coefficients. In our initial experiments we have also tested the system using 9 and 11 MFCC, and the classification results using 9 and 11 MFCC were significantly lower than the classification results of using 13 MFCC.

All the above features are calculated over very short time frames. To calculate these features the signal is broken down to analysis windows which are small and possibly overlapping segments of time. The time of these segments are so short that the frequency characteristics of the magnitude spectrum remain relatively stable. A few analysis windows following each other produce the sensation of the sound texture. The texture window is a couple of analysis windows following each other and refers to the minimum time required to identify the texture of a sound. The texture window includes the mean and standard deviation of features over a number of analysis windows. In their system the size of the analysis window is 23 ms (512 samples at 22 050 Hz sampling rate) and the texture window is 43 analysis windows which add up to be 1 s.

To summarize the feature values for each song the mean and standard deviation of a running multidimensional Gaussian distribution is estimated based on the feature vector of the current texture window in addition to a number of previous feature vectors. So the texture window can be described as a memory of the past. To implement this efficiently a circular buffer holding the previous feature vectors was implemented (Tzanetakis and Cook, 2002). In other words a running mean and standard deviation is calculated for each texture window to summarise the feature values of the analysis windows. Then again the mean and standard deviations of the texture windows are calculated to summarise the values for the whole song. This means that for each feature we have mean of means, mean of standard deviations, standard deviation of means, and standard deviation of standard deviations (Tzanetakis, 2007).

Since we have 17 features (zero crossing, spectral centroid, spectral rolloff, spectral flux, and 13 MFCC) the feature vector will be 68 dimensional (4×17) because for each feature we have mean of means, mean of standard deviations, standard deviation of means, and standard deviation of standard deviations. This 68 dimensional feature vector represents the entire song.

Throughout the rest of this thesis we will call this feature vector, timbral feature vector, because these features represent the timbre of the sound. In this thesis we propose improvements over this technique by combining high level features including beat and chord features with the abovementioned 68 dimensional timbral feature vector. The details of our proposed high level features are presented in *Chapter 3*.

Tzanetakis (2007) states that some of the limitations of their system are the lack of rhythm based and pitch/chroma based features. Although they initially proposed techniques for extracting beat and pitch features in 2002 they did not implement it in their 2007 method. In our proposed method we use features that contain information on both rhythm and Chroma.

To evaluate their proposed feature set Tzanetakis and Cook (2002) originally used a GMM (*Gaussian* Mixture Models) and a KNN (K-Nearest Neighbour) classifier. However in (Tzanetakis, 2007) an SVM (Support Vector Machine) classifier is used. This classifier is developed using LIBSVM[5] library and is available in Marsyas. In our experimental studies we have also used this SVM classifier along with a couple of other classifiers. More details on the classification method that we used are available in *Chapter 4*.

2.3. Other existing techniques

2.3.1. Techniques based on low level features

Many other novel methods have been proposed for the task of music genre classification. Many of these methods provide improvements on the works of Tzanetakis and Cook (2002). What most of the previous works have in common is that they mainly utilize low level features. Here we describe the most significant researches in this area and briefly discuss their strengths and limitations.

Tsuchihashi et al. (2008) used pitch based bass-line features to classify music into 6 different genres. They stated that the average accuracy has been improved from 54.3% to 62.7% by incorporating features from bass-line (Tsuchihashi et al., 2008).

A better method to incorporate the way humans perceive music is proposed by Chua (2007). Instead of concentrating on bass-line features and other features separately they converted the music signal to Phons and then to Sone before the feature extraction. This technique addresses the fact that the perceived intensity of the signal by humans depend on both the frequency and

[5] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

loudness of the signal. This means that bass-line features are only considered when they are loud enough (Chua, 2007).

(Panagakis et al., 2008) used a multilinear approach in genre classification. They used tensors in representing spectro-temporal modulation features extracted from the music and used these tensors to classify the songs. Their results were comparable to state-of-the-art algorithms. On the GTZAN data set of 1000 songs in 10 genres (100 songs each) they achieved an accuracy of 75.01% to 78.20% and on the dataset of ISMIR2004 of 1458 songs in 6 different genres (Classical (640), Electronic (229), JazzBlues (52), MetalPunk (90), RockPop (203), World (244)) they achieved an accuracy of 87.53% to 80.95% (Panagakis et al., 2008).

Sundaram and Narayanan (2007) recognised the fact that due to artistic style, different sections of a song may sound like different genres while the song as a whole belongs to a particular genre. To address this issue they used activity rate and Dynamic Time Wrapping to determine the overall genre of a full length song. The activity rate consists of 3 time vectors each representing the value of one of the 3 attributes over time. The 3 attributes are speech-like attributes, harmonic attributes, and noise-like attributes. These attributes are determined using a 39 dimensional MFCC related feature vector. However in their experiment of classifying music into 6 genres the results were significantly lower when using activity rate compared to when using just the MFCC features. (Sundaram and Narayanan, 2007)

Holzapfel and Stylianou (2008) Used Nonnegative matrix factorization (NMF) to describe the timbre of the music and they achieved a maximum improvement of 23% compared to MFCC based models using GTZAN and ISMIR2004 datasets. They suggest NMF must also be used in future for describing rhythm and modulation characteristics (Holzapfel and Stylianou, 2008).

Barbedo and Lopes (2007) used a very well defined set of hierarchical genre taxonomies. These taxonomies are very wide and deep so a very meticulous comparison can be done between the genres. Their feature set included spectral roll-off, loudness, frequency bandwidth, and spectral flux. They had 4 hierarchies of genres and they achieved an average accuracy of 87 % for level 1, 80 % for level 2, 72 % for level 3, and 61 % for level 4 (Barbedo and Lopes, 2007).

To overcome the limitations of mean and standard deviation in summarizing the feature data, Meng et al. (2007) used a multivariate autoregressive feature integration Scheme to integrate temporal features (short-time features, medium-time features, and ling-time features). As a result two new feature sets of DAR (Diagonal Autoregressive) and MAR (Multivariate Autoregressive) were created. Their experiment resulted in significant improvement of classification results compared to the traditional mean/variance method.

Meng et al. (2007) defined short time features as features which are calculated on very short time frames of usually 20 to 40 milliseconds (Analysis Window). MFCC are an example of short time features. Medium time features were defined as features that are calculated on timeframes of 1 to

2 seconds (Texture Window) such as Zero Crossing. And finally they defined long time features as the features that incorporate statistical information of the whole song such as the beat histogram (Meng et al., 2007).

Kotov et al. (2007) proposed wavelet and pseudo-wavelet based features and used Support Vector Machine to classify Genres (Kotov et al., 2007).

Barrington et al. (2008) used features to derive kernels which were then used to classify music into 174 tags (including emotions, genres, etc.) they used features based on MFCC, Chroma, social tags(text) and web mined documents (text). Interestingly they have stated that they had investigated high level features such as song's key and tempo and found that these features perform poorly in combination with social tags and web mined documents (Barrington et al., 2008).

Lidy et al., (2007) combine audio and symbolic descriptions to classify the songs into genres. In their method they first extracted the following audio features from the songs music signal.

- Rhythm patterns: extracted using short term FFT (Fast Fourier Transform) and sonogram and then applying DFT (Discrete Fourier Transform) to the sonogram.
- Rhythm histogram: based on the rhythm patterns.
- Statistical spectrum descriptors: to capture additional timbral information.
- Onset Features: determines whether an audio frame is an onset frame or a non-onset frame. Also the onset detection algorithm can be used determined the attack which provides information on music instrument.

Then they converted the audio files to MIDI files only considering pitches and duration of the notes. And then they extract the features proposed by (Leon and Inesta, 2007) and (Rizo et al., 2006) from the MIDI files and combine them with the respective audio features.

These MIDI features include overall descriptors (such as number of notes), pitch descriptors, note duration descriptors, silence duration descriptors, inter-onset interval descriptors, pitch interval descriptors, non diatonic note (harmonic) descriptors, syncopation (rhythmic) descriptors, and their normalities. For classification, they use the SMO (Sequential Minimal Optimization) implementation of SVM (Support Vector Machine) in WEKA[6] on three datasets of GTZAN, ISMIRrythm, and ISMIRgenre. However they did not observe substantial improvements (Lidy et al., 2007).

[6] www.cs.waikato.ac.nz/ml/weka

Although Lidy et al. (2007) made use of symbolic features in their experiment, they were mostly low level features and their feature set lacked high level information of the songs.

Lampropoulos et al. (2005) Proposed to separate the musical instrument sources in the signal first and then extract the features from each individual separated source. The used the original feature set of Tzanetakis and Cook (2002) and Marsyas 0.1 for feature extraction.

The source separation algorithm that they proposed was based on the works of (Virtanen, 2004). They used Convolutive Sparse Coding (CSC) algorithm to separate the instrument team sources such as strings, wind, and percussion. The CSC takes into account the fact that each source may not be active for the whole duration of the song and it may only be active for a certain period of time in a song (Lampropoulos et al., 2005).

As a result of their experiment they observed 1 to 2 percent increase in the accuracy when the sources are separated. They argue that this is because the source separation revealed more information about the timbral texture, rhythm, and pitch (Lampropoulos et al., 2005). So although they have separated the instruments in a music signal, they did not add any high level features to the feature set rather they extracted same features but from instruments instead of the whole signal.

2.3.2 Motivations and preliminary works on high level features

All the studies outlined above only make use of low level features as deduced from the Gjerdingen and Perrott (1999). The only research that briefly examined high level features was the work of Barrington et al. (2008). But as stated above they did not make use of high level features in their system. However we have sufficient motivation to experiment with high level features. There have been a few works in the literature that used high level features and observed improvements in the results. This does not mean that Gjerdingen and Perrott (1999) were wrong, it means that beside timbral features there are many other factors that affect the music genre and they should not be ignored.

McKay (2004) suggested that the study of Gjerdingen and Perrott (1999) does not imply that high level musical features should be ignored. The musical form and structure also play an important role in music classification but they are not the basic essential feature used for classifying music genres. McKay (2004) also concluded that based on the study of (Tekman and Hortacsu, 2002) music genres are strongly related to music emotion (McKay, 2004). There have been numerous studies in psychoacoustics and automatic music emotion classification and they mostly suggest that the high level music features play the most important role in music emotion classification (Chua, 2007, Gabrielsson and Juslin, 1996, Kamenetsky and Hill, 1997). So perhaps in the case of genre classification, similarly, high level features play an important role.

McKay and Fujinaga (2006) proposed ways to improve the automatic genre classification techniques. They suggested that information from low-level, high-level and cultural features must be combined. They state that since currently most genre classification techniques only incorporate low-level timbre based features their performance have been limited. Then they argue that timbre only represents a small portion of what humans use for classifying music genre and high level features are central to music performers and composers in performing and composing in different music genres (McKay and Fujinaga, 2006).

McKay and Fujinaga (2008) showed that combining audio features with symbolic and cultural features improves the classification results significantly. They combined features extracted from audio recordings, MIDI recordings, and cultural metadata which resulted in about 13% increase in classification results compared to average classification accuracy using these features individually (McKay and Fujinaga, 2008).

The high level features used by (McKay, 2004, McKay and Fujinaga, 2006, McKay and Fujinaga, 2008) are all extracted from symbolic music (MIDI). In their experiments they extracted numerous features from symbolic music pieces from which most are low level and some are high level. Their huge symbolic feature set included features based on texture, beat, melody, pitch, and instrumentation.

Another study that provides motivation for studying high level features is the work of (Chase, 2001). Chase (2001) performed experiments to investigate the ability of Koi (*Cyprinus carpio*) fish to discriminate musical genre. And the results indicate that Koi can discriminate music by their genres. The fish learned to classify blues and classical genres. They were trained using one blues and one classical song and they were able to generalize it to discriminate blues and classical genres (Chase, 2001).

As a result of their study compared to other studies they concluded that animals such as Koi use both local features as well as pattern features to classify music. However they prefer local features instinctively because features that are instantly recognisable grant a survival advantage (Chase, 2001). This means that although short time features (low level features) play the most important role in music genre classification in animals and therefore humans, pattern features (high level features) also provide some form of extra clue that helps the classification.

To combine low level features with human perception of music Shen et al. (2006) suggested using PCA (Principal Component Analysis) to reduce dimensionality (Shen et al., 2006). However their hypothesis is not based on any psychological and psychoacoustic argument.

2.3.3. Techniques based on high level features

Perhaps Zhu et al. (2004) are one of the few who have truly used high level features for music genre classification. They used features such as the instrument distribution and instrument based notes features to do genre classification. The distribution of instrument groups show how long each instrument group is played. And the mean and standard deviation of the notes played within each instrument group characterises the melody and captures some statistical features. (Zhu et al., 2004).

They used a dictionary of instruments' spectrum to represent information on different instruments. These spectrums were constructed by performing FTT on sample audios. The dictionary is defined for j different instruments and each instrument has k notes. The dictionary D is represented by a 2 dimensional M-by-N matrix where M is the number of frequency bins and N is j multiplied by k.

The distribution of instruments and the instrument based notes are calculated using matrix X. Matrix X is calculated by solving the following equation:

$$DX = A \tag{5}$$

Where A is the matrix of spectrum of each frame in the target music.

The authors state that there are many possible solutions to matrix X when $N>>M$ which usually is the case. They argue the optimal solution is the one which contains the greater number of zeros. This is based on the assumption that "few musicians play a lot of notes at the same time. And the number of instruments involved in a song rarely exceeds 3 or 4." The authors state that by obtaining the right information on dominant notes and instruments, the ration of errors and inaccuracies become acceptable in the optimal solution (Zhu et al., 2004).

Their dictionary of instruments consists of 30 instruments selected form 127 MIDI instruments forming 8 instrument groups. For each instrument 79 notes are sampled. For their experiment they used a dataset of 1699 songs within 4 genres of rock, classical, jazz , and pop. And for the classification they used a GMM with 16 components. The probability for the whole music is calculated by multiplying the probability of each clip (Zhu et al., 2004).

On average Zhu et al. (2004) achieved 4.5% improvement in the classification results compared to using MFCC and energy features. And when combining their features with the MFCC features they achieved an improvement of 11.3% compared to using only MFCC and energy features.

In this thesis we derive high level features from musical chords and combine them with low level features for genre classification. During the final stages of preparing this thesis, we noted a paper published in the Connection Science which also uses chord based high level features (Pérez-Sancho et al., 2009). They extracted chord progression features from MIDI files and used them

for genre classification. To classify music genre, based on chord progression features they proposed n-grams approach which is similar to our proposed approach.

In our system we propose using a combination of low level and high level features including chord progressions whereas Pérez-Sancho et al. (2009) only concentrates on using chord progression features. Moreover, we extract all features directly from the music signal (i.e. mp3 and au formats) rather than MIDI music.

Chapter 3:
Proposed Genre Classification Technique

To overcome the limitations of the current methods for genre classification of polyphonic music signal we propose the use of high level features in combination with low level features. In particular we concentrate on chord and chord progression features. In our proposed system we also examine beat (rhythm) features.

In this chapter we first provide an overview of our proposed genre classification technique outlining the motivations and reasons for selecting chord and chord progression features. Then the architecture of our proposed method is provided following by detailed discussion on each of the steps in the system.

3.1. Overview of the Proposed Genre Classification Technique

In our proposed method we use a combination of low level features which represent the signal physical properties and information extracted from high level features which represent perceptual musical concepts to do genre classification. To be more specific, low level timbral features of (Tzanetakis, 2007) are used in conjunction with Beat features and information from chords and chord progressions to construct a unified feature vector. Then machine learning and data mining algorithms are used to predict the genre of the songs based on this feature vector.

In our proposed method we introduce ways to extract useful information from chord and chord progressions and methods to organise these information in a system that can be used in conjunction with other features to create a unified feature vector which can be used as an input to machine learning algorithms. We also experiment with rhythmic features, although rhythmic features have been used before (Tzanetakis and Cook, 2002, Lidy et al., 2007, Meng et al., 2007, Lampropoulos et al., 2005).

Chords and chord progressions are very high level musical concepts. Since studying chords of music require dealing with an extremely wide and unstable problem space, and since the number of chords played in different songs is extremely variable, we cannot directly use chords in a feature vector. Therefore we extract statistical chord information from chords and genre likelihood information from chord progressions to capture chord properties of the songs. The details of these feature vectors and how they capture the chord and chord progression information are discussed in the following chapters.

3.1.1. Why Chords

According to Cambridge dictionary chord is defined as "Three or more musical notes played at the same time". A chord progression is a number of chords played consecutively. Chord progression structures represent high level harmony of the songs (Paiement et al., 2005).

Chords and chord progressions are good candidates for extracting high level information of the songs. They are both conceptual notions that are close to human cognition of music theory. Chord progressions contain pattern information that requires longer time to build conceptually in contrast with timbral features that are comprehended in very short timeframes by human beings.

Our rational and motivation behind using chords and chord progressions come from (Paiement et al., 2005). They demonstrated that chord progressions capture high level and sophisticated harmony organizations, which we believe is a good source of information for genre classification. In addition, Lee (2007) used genre information to increase the accuracy of chord extraction algorithms. This shows a relationship between chords and genres and indicates that chord information can be a candidate for increasing the accuracy of genre classification.

Additionally a number of researchers (Tzanetakis and Cook, 2002, Lampropoulos et al., 2005, Barrington et al., 2008) have proposed that chroma and pitch based features be used to increase the classification accuracy. Chords capture the chroma and pitch based information of the songs indirectly.

3.2. System Architecture

Figure 1 illustrates the architecture of the proposed system. The proposed technique consists of the following steps. These modules are described in more details on the following sections of this chapter.

1. Timbral Feature Extraction

In this stage timbral features are extracted from the audio signal. These are the 5 timbral features used in (Tzanetakis, 2007). These features are extracted using Marsyas 0.2. Timbral features serve the baseline of our proposed system.

2. Beat Feature Extraction

In this stage, 8 beat histogram features are extracted from the audio signal using Marsyas 0.2.

3. Chord Extraction

In this stage the chords of the songs are extracted from the audio signal using Clam Annotator's ChordExtractor[7].

4. Statistical Chord Feature Extraction

In this stage a feature vector is constructed by counting the number of roots and modes of the chords of the songs.

5. Chord Pre-Processing

In this stage the chords extracted by Clam Annotator are converted to a format suitable for chord mining.

6. Chord Mining

In this stage a pattern matching algorithm estimates the likelihoods of each song belonging to each genre based on the chord progressions of the songs.

7. Feature Integration

In this stage different feature vectors are constructed by combining the outputs of the timbral feature extraction, beat feature extraction, statistical chord feature extraction, and chord mining stages.

8. Classification

This is the final stage of the system where data mining and machine learning algorithms predict the genre of the music based on the features in the feature vector.

[7] http://clam-project.org/

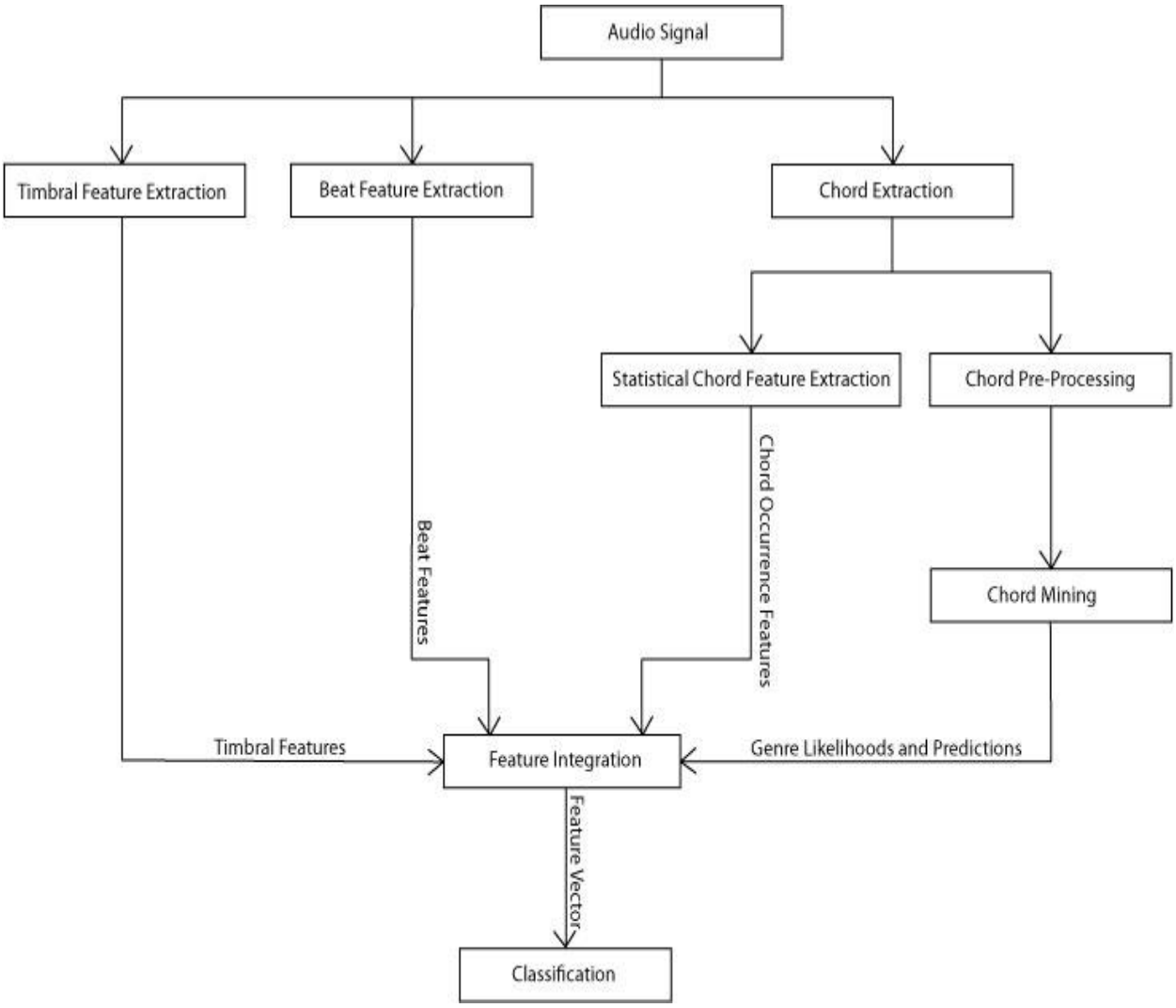

Figure 1. System Architecture

3.3. Timbral Feature Extraction

These features were originally proposed in (Tzanetakis and Cook, 2002) and were used by (Tzanetakis, 2007). As mentioned before these features capture the timbral texture of the sound and are calculated over very short time periods. Using Marsyas 0.2 Spectral Centroid, Spectral Rolloff, Spectral Flux, Time Domain Zero Crossing, and 13 Mel-Frequency Cepstral Coefficients are extracted into a 68 dimensional feature vector. To summarise the feature values over the whole song a running mean and standard deviation is calculated resulting in 4 values per feature. So each of the 17 features mentioned above are represented by 4 dimensions in the feature vector. The definition of the timbral features, technique for summarising them, and timbral feature vector construction were discussed in details in S*ection 2.2*.

This feature vector serves the baseline of our system. We will compare the classification accuracy when using our proposed feature vectors with the classification accuracy when using this feature vector.

3.4. Beat Feature Extraction

Beat (rhythm) features capture the rhythmic content of the signal. We use eight beat features extracted from beat histogram. The details of these features and the beat histogram extraction algorithm are discussed here.

The beat histogram demonstrates the BPM (Beat-Per-Minute) value on the abscissa and the signal amplitude or beat strength on the ordinate. The BPMs of the significant peaks in the histogram are more intensely perceived to be the beat and rhythm of the song. We extract low peak amplitude, low peak BPM, high peak amplitude, high peak BPM, and high to low ratio from the beat histogram using Marsyas. This histogram is also used to extract 3 values representing the sum of energies in BPM bands. The details of beat histogram extraction and these 8 features are explained later in this section.

Tzanetakis and Cook (2002) originally proposed techniques for extracting the timbral and beat features. In (Tzanetakis, 2007), however, the author decided not to make use of beat features. Here we extract the beat features using Marsyas 0.2. The beat histogram extraction algorithm implemented in Marsyas is based on (Tzanetakis and Cook, 2002). Here we describe the algorithm they used for extracting beat histogram.

The algorithm for extracting beat histogram is based on Discrete Wavelet Transform (DWT) rather than Short Time Fourier Transform (STFT). Tzanetakis and Cook (2002) argue that this is to overcome the resolution problem of STFT. They believe that while STFT provides uniform

time resolution over all frequencies, Wavelet Transform provides high time resolution and low frequency resolution for higher frequencies, and low time resolution and high frequency resolution for lower frequencies.

In their technique, Tzanetakis and Cook (2002) first decompose the signal to a number of sub bands, one for each octave frequency band, using Discrete Wavelet Transform. Then for each octave frequency band they perform full wave rectification, low pass filtering, downsampling, and mean removal to extract the time domain amplitude envelope. Then these envelopes are summed up and an autocorrelation function is applied. Finally multiple peak picking is performed to construct the beat histogram.

The different periodicities of the signal are demonstrated in dominant peaks of the autocorrelation function. The signal amplitude of the peaks selected by the peak picking algorithm are then added up to the histogram. This means that the peaks of the beat histogram are higher where the signal pattern is more similar to itself, i.e. where there is a stronger beat (Tzanetakis and Cook, 2002).

The following are the definitions of the components used in the beat histogram extraction algorithm of (Tzanetakis and Cook, 2002) where x[n] is the time domain signal of sample n.

1. Full wave rectification:
Converts the time domain signal to the temporal envelope of the signal. It is defined as:

$$y[n] = |x[n]| \tag{6}$$

2. Low pass filtering:

$$y[n] = (1-\propto)x[n]+\propto y[n-1] \tag{7}$$

To smooth the envelope a one-pole filter with an $\propto$ of 0.99 is used. A low pass filter after a full wave rectification is the typical procedure for extracting envelope.

3. Downsampling:

$$y[n] = x[kn] \tag{8}$$

Downsampling reduces the computation time of the autocorrelation function without affecting the performance of the algorithm. The authors used k=16.

4. Mean removal:
This is used to make the signal zero-cantered for the autocorrelation function:

$$y[n] = x[n] - E[x[n]] \tag{9}$$

5. Autocorrelation:
They used the following definition for their autocorrelation function:

$$y[k] = \frac{1}{N}\sum_{n} x[n]x[n-k] \tag{10}$$

Where k is the lag time. So when the signal is largely similar to itself there will be a peak in the autocorrelation function. In other words the time lags that result in a peak are most likely to be the periodicities of rhythm and beat.

6. Peak selection algorithm and beat histogram calculation:
To construct the beat histogram, the first three peaks of the autocorrelation function which are within a particular range are selected. These peaks are added to the beat histogram with their corresponding peak amplitude. The bins in their peak histogram correspond to BPM and cover a range of 40 to 200 BPM.

After the beat histogram is extracted, Marsyas calculates the following eight features from the histogram. These features capture detailed information on the number and position of the peaks and the strength of the beats. The high and low peaks are selected within a specific range.

1. Low peak amplitude
This feature represents the amplitude of the lower peak.

2. Low peak BPM
This feature contains the BPM of the lower peak.

3. High peak amplitude
This feature represents the amplitude of the higher peak.

4. High peak BPM
This feature represents the BPM of the higher peak.

5. High to low ratio
This feature is the ratio of the highest selected peak to the lowest selected peak.

6. Sum values
Three sum values are calculated each for one BPM band. This signifies where the beat

energy has higher concentration. For example if the sum value of the lower BPM band is higher it means that there is more energy in slower tempos.

Marsyas captures the beat features of the song into an eight dimensional beat feature vector consisting the above eight features.

Marsyas is an open source software and is evolving constantly. The experimental results using this software are not published as fast as its evolution. As a result, although the algorithm for extracting these eight features has been implemented in Marsyas, we do not know of any experimental work in the literature applying these features in music genre classification. However similar beat features have been applied to genre classification successfully so far (Tzanetakis and Cook, 2002, Meng et al., 2007, Lidy et al., 2007, Lampropoulos et al., 2005).

3.5. Chord Extraction

Because of the aforementioned reasons in *Section 3.1.1* we have proposed using features based on chord and chord progressions to capture high level information of the songs. To capture these high level information the chords of the songs must first be extracted.

To extract the chords of the song ChordExtractor which is a part of Clam Annotator software is used. ChordExtractor has been implemented based on (Harte and Sandler, 2005).

The algorithm of Harte and Sandler (2005) extracts the chords using a Chromagram. They exploited the psychological theory; that it is possible to model the human perception of pitch using a helix (Shepard, 1964). Using the helical representation they assigned 2 attributes to the pitch; pitch height which increases with the octave and represented on the vertical axis, and pitch class or Chroma which is represented by the rotation of the helix (Harte and Sandler, 2005). In their initial experiments they achieved an average of 62.4% accuracy in chord extraction.

In the algorithm, first a 36-bin HPCP (Harmonic Pitch Class Profile) is derived from constant-Q spectral transform. HPCP or Chromagram is a vector representation of the Chroma circle. Then a peak picking algorithm is used to find the exact location of the amplitude of the peaks. Then a tunning algorithm followed by a pitch class allocation is applied to produced a 12-bin semitone-quantised Chromagram to represent the 12 chords roots (A, A#, B, C, C#, D, D#, E, F, F#, G, G#). finally a pattern matching algorithm is used to compare each frame of the Chromagram with a set of predefined chord templates and then the closest match is recorded as the chord of that frame (Harte and Sandler, 2005).

Their original chord identification pattern matching algorithm could only identify 4 modes of major, minor, augmented, and diminished. However the ChordExtractor implementation of the

Clam Annotator software can identify 9 modes of Augmented, Diminished, Diminished7, Dominant7, Major, Major7, Minor, Minor7, and MinorMajor7.

ChordExtractor extracts the root and mode of the chords. The root of a chord is one of the twelve notes in a chromatic scale and mode (type) of the chord is one of the above mentioned nine modes. So in total 12 × 9 (108) different chords can be represented.

The number of chords extracted from each song depends on the content of the song (i.e. how many chords are played in that song). Some songs may have very few chords while some other may have many chords. An XML file containing the chords of the song is produced by ChordExtractor as a result of extracting the chords.

3.6. Statistical Chord Feature Extraction

As mentioned before the number of chords extracted varies for different songs. This is because different songs naturally have different number of chords played in them. The aim in this section is to capture chord information in a way that can be combined with other features to be used as an input for machine learning and data mining algorithms. The do this the information must be represented as a feature vector.

For the purpose of classification we construct a 21-dimensional feature vector which contains statistical information on the chords. This feature vector consists of 12 dimensions corresponding to the 12 roots and 9 dimensions corresponding to the 9 modes. This vector identifies the number of occurrences of each root and each mode in a song. This is done by simply counting the number of occurrences of roots and modes in a song.

In other words we count how many chords of each root are present in a song and store the number for the corresponding roots. Then count how many chords of each mode are present in each song and store the number for the corresponding modes. The 21 dimensions of the feature vector are A, A#, B, C, C#, D, D#, E, F, F#, G, G#, Augmented, Diminished, Diminished7, Dominant7, Major, Major7, Minor, Minor7, and MinorMajor7.

In our initial experiments we have also tested other statistical information such as the statistical mode of the roots, modes, and chords but they did not show any improvements in the results.

3.7. Chord Pre-Processing

To capture the chord progression information into a feature vector we propose a pattern matching approach inspired by data mining algorithms. We have called this approach chord mining and developed an algorithm based on this approach called ChordMiner. Details of this approach and algorithm are discussed in *Section 3.8*.

The format in which the extracted chords are represented is not suitable for chord mining. Therefore we have to pre-process the chord data outputted by ChordExtractor to make it suitable to be used as input to ChordMiner.

Clam Annotator's ChordExtractor outputs the chords of the songs in XML files. These files contain a lot of extra information which is not required and is stored in a format which is not efficient for chord mining. The roots and modes of the chords are stored in separate XML tags. ChordExtractor creates one separate XML file for each music (mp3) file.

We developed an algorithm to read, transform and consolidate these chord data. The output of the algorithm is one CSV (comma separated values) file (dataset) that contains all the chords for all the song. Each record (row) of the file corresponds to a music (mp3) file. Each row starts with the file name which is followed by the genre name and then the chords of the song. Note that the number of columns of the dataset is not fixed because there is no fixed number of chords and each song have different number of chords.

Instead of roots and modes being stored separately, in this dataset the chords are stored as a whole entity (i.e. C Major or G# Minor).

3.8. Chord Mining

A chord progression is a sequence of chords played in a row. Different songs have different number of chord progressions and different chord progressions are made up of different number of chords. There is no fixed rule for the number of chords in a chord progression, just any pattern of chords played consecutively makes up a chord progressions. Well known chord progressions are sometimes broken up into multiple sections with irrelevant chords being played in between them. Musicians sometimes perform such acts to adjust the consonance and dissonance of the harmony in the songs.

The abovementioned properties of chord progressions demonstrate that we are dealing with an extremely wide problem space. As mentioned before our aim here is to capture high level information in a format that can be combined with other low level features to be used in data mining.

Chord progressions can be comprised of different number of chords. We cannot always rely on fixed chord progression patterns because musicians often break or modify the musical clichés to produce original musical pieces. A composer may use a combination of progressions, each made up of different number of chords, in the same music piece. Incorporating musical knowledge of chord progressions into genre classification thus requires dealing with an extremely wide problem space which makes the task very complicated.

To capture the high level harmonic information embedded in chord progressions into a feature vector that can be used in conjunction with other features; a pattern matching algorithm is proposed. This algorithm is inspired by the way data mining algorithms train and test a model therefore we name this algorithm ChordMiner and call this technique Chord Mining. Similar to data mining where meaningful information is extracted from abundance of data, in chord mining meaningful information is extracted from the chords of the songs.

To simplify the problem space we assume that the number of chords in all the chord progressions is fixed. For example we assume that all the chord progressions of the songs are made of three chords. We have compiled our algorithm 4 times with the assumption of having 2, 3, 4, and 5 chords in progressions.

ChorMiner estimates the likelihood of songs belonging to each genre from the chord patterns within the songs. It produces two different feature vectors. The first one is an n-dimensional feature vector where n is the number of genres available in the training dataset. Each of the dimensions of this vector corresponds to one of the music genres. For each song the likelihood of it belonging to all the genres are estimated and recorded in the corresponding dimension. The second feature vector is a 1-dimentional vector comprising of the predicted genre of the song. Chord miner predicts the genre of the songs by selecting the genre with the highest likelihood.

So the number of dimensions of the likelihoods feature vector will be the same as the number of genres present in the training set. In other words each dimension of the vector represents the likelihood of one genre.

Like data mining algorithms ChordMiner has the two stages of training and testing. In the training phase ChordMiner builds a model for each genre in the dataset. These models consist of the chord patterns that appear in each genre more than once and the number of times that they appear divided by the number of unique chord patterns within the genre. ChordMiner trains and tests the model based on a 10 fold cross validation approach.

In each of the iterations 90% of the songs in each genre are selected. Then for all the songs in each genre all the patterns of 3 chords and their occurrence frequency is recorded. Then for all the chords where their frequency is larger than 1 a probability descriptor P_g is calculated:

$$P_g(c) = \frac{f_g(c)}{n_g} \tag{11}$$

Where $P_g(c)$ is the probability descriptor of chord pattern c appearing in genre g. And $f_g(c)$ is the number of times that chord pattern c appears in the songs of the genre g. And n_g is the number of unique chord patterns within the songs of the genre g. This process is repeated for all the genres within the dataset.

In the testing stage ChordMiner selects the remaining songs and estimates their likelihood for belonging to different genres.

$$L_g(s) = \sum P_g(c) \tag{12}$$

Where $L_g(s)$ is the likelihood of song s belonging to genre g. And $P_g(c)$ is the probability descriptor of a given chord pattern c for genre g. In other words $L_g(s)$ is the sum of the probability descriptors of the chord progressions present in each song and is calculated for each genre separately using the probability descriptors of the corresponding genre. Note that $P_g(c)$ will be zero if the given chord pattern does not exist in the model of the given genre.

In other words, for each song we will have one likelihood per genres and each of these likelihoods are calculated by adding up the genre probability descriptors of the chord patterns present within the song. And the genre probability descriptors of the chord progressions are calculated before by counting the number of times that the chord patterns appear in each genre and dividing it by the number of unique chord patterns within that genre.

This process is then repeated 9 more times using different chunks of the dataset to complete the 10 fold cross validation. ChordMiner also predicts the genre of the songs by selecting the genre which has the highest likelihood. The estimated likelihood values and the predicted genre are then stored for all the songs so they can later be used in conjunction with other features for classification.

ChordMiner has been implemented with C++. The following are the steps taken in ChordMiner:

- In each fold
    - [Training] For each genre in dataset
        - Select 90% of the songs
        - For each song read all the combinations of n consecutive chords present in it
        - For each chord pattern store them in an array or if they already exist in the array increase their count
        - Go through the array of chord patterns and for each pattern if the count is greater than one
            - Calculate the probability descriptor by dividing the count by the number of unique chord patterns
            - Store the chord pattern, its probability descriptor, and respective genre in an array called Model
    - [Testing] For each genre in dataset
        - Select the remaining 10% of the songs
        - For each song read all the combinations of n consecutive chords present in it
        - For each chord pattern if it exists in the model
            - Add its probability descriptor to the likelihood of the respective genre in the likelihood feature vector
        - For each song select the genre with the highest likelihood and store it in the prediction feature vector

We have experimented with ChordMiner for patterns of 2, 3, 4, and 5 chords and found that the best results are achieved when using permutations of 3 and 4 chords.

3.9. Feature Integration

During the previous steps 7 feature vectors have been extracted. *Table 1* lists the details of the aforementioned feature vectors.

Table 1. Different feature vectors extracted

Feature Vector	Abbreviation	Dimensionality	Description
Timbral Features	T	68	Extracted during Timbral Feature Extraction. This feature vector corresponds to those of (Tzanetakis, 2007). These features capture the low level information on timbral texture of the songs. This feature vector is the baseline of our experiment.
Beat Features	B	8	Extracted during Beat Feature Extraction. These features capture the rhythmic content of the songs.
Statistical Chord Features	C	21	Extracted during Statistical Chord Features Extraction. These features capture high level information on the chord contents of the songs. So indirectly this feature vector also captures the pitch and harmony content of the songs.
Genre Likelihoods Extracted from Progressions of 3 Chords	L3	10	Extracted during Chord Mining. These features capture high level information on the chord progressions of the songs where it is assumed that all the progressions consist of three chords. This vector consists of the likelihoods of each of the 10 genres in our ground truth dataset. This vector therefore captures high level sophisticated harmony structures of the song indirectly.
Genre Predictions Extracted from Progressions of 3 Chords	P3	1	Extracted during Chord Mining. This feature vector captures high level information on the chord progressions of the songs where it is assumed that all the progressions consist of three chords. This vector consists of the predicted genre by ChordMiner. This vector therefore captures high level sophisticated harmony structures of the song indirectly.
Genre Likelihoods Extracted from Progressions of 4 Chords	L4	10	Extracted during Chord Mining. These features capture high level information on the chord progressions of the songs where it is assumed that all the progressions consist of four chords. This vector consists of the likelihoods of each of the 10 genres in our ground truth dataset. This vector therefore captures high level sophisticated harmony structures of the song indirectly.
Genre Predictions Extracted from Progressions of 4 Chords	P4	1	Extracted during Chord Mining. This feature vector captures high level information on the chord progressions of the songs where it is assumed that all the progressions consist of four chords. This vector consists of the predicted genre by ChordMiner. This vector therefore captures high level sophisticated harmony structures of the song indirectly.

During the feature integration step different combinations of the above feature vectors are created and their respective data is consolidated. We have experimented with 18 different feature vectors made up of a combination of the above feature vectors. These 18 feature vectors are identified in *Table 2*.

Table 2. Combined feature vectors

Combined Feature Vector	Dimensionality
T	68
B	8
TB	76
C	21
TC	89
TBC	97
L3	10
TL3	78
TP3	69
TL3P3	79
TBL3	86
TBCL3	107
L4	10
TL4	78
TP4	69
TL4P4	79
TBL4	86
TBCL4	107

Here we have experimented with these 18 feature vectors. These feature sets are some of the possible combinations that we have experimented on. The best performing feature vectors are identified in *Section 4.4* based on the classification performance.

3.10. Classification

During this stage any machine learning algorithm can be utilised to classify the music pieces. The feature vector created in the feature integration step is used for the classification. Because of the high dimensionality of the feature vectors algorithms such as Support Vector Machine (SVM) seem to be more suitable for this task.

In our experiments we have performed the classification using 3 different algorithms. Two of these classifiers are SVM based and one is Neural Networks based. These algorithms are selected by considering the high dimensionality of the feature vectors and also performing initial greedy based search to find the most suitable algorithm. More details on the classification methods and the algorithms used are presented in the next chapter and the experimental results are analysed.

Chapter 4:
Experimental Studies

We have performed experimental studies on our proposed system to examine the performance of the proposed technique. Experiments using different feature vectors and classifiers are performed to evaluate our proposed method and the results are discussed and analysed.

In this Chapter, we study effects of high level features discussed in the previous section on genre classification. The best performing feature vector is identified here and the different effects of the different features on the classification results are analysed. Also experimental results of the performance of ChordMiner are presented here and compared with those of Pérez-Sancho et al., (2009)

In this chapter our experiment set up is demonstrated first. Then the baseline performance is evaluated by experimenting classification on Timbral features. Then experimental studies on ChordMiner are presented and its performance is analysed in details. And finally the effects of different features on the classification results are analysed and discussed after performing classification using different combination of feature vectors.

4.1. Experiment Setup

In the experiments GTZAN dataset is used as the ground truth. GTZAN is a well known and widely used dataset in the community. This dataset consists of 10 genres with 100 songs per genre, which adds up to be 1000 songs in total. Each song is 30 seconds. The 10 genres in this dataset are blues, classical, country, disco, hip hop, jazz, metal, pop, reggae, and rock.

Various softwares have been used in these experiments. Timbral and Beat Features are extracted using Marsyas. The song chords are extracted using Clam Annotator's chordExtractor. Statistical chord features are extracted from the chords using Microsoft Excel and a custom built algorithm written in C++. Chord Pre-Processing is done using Altova Mapforce, Altova XMLSpy, Microsoft Excel, and a custom built algorithm written in C++. Chord mining is performed using ChordMiner. Feature Integration is performed using Microsoft Excel and WEKA. And finally classification is performed using WEKA and Marsyas's KEA[8].

The Music files in the GTZAN dataset are in SUN Audio (.au) format. Since Clam Annotator does not support .au files, all the files have been converted to MP3 using FairStars Audio Converter before chord extraction.

The base system which is used as a benchmark in our experiment is based on (Tzanetakis, 2007). MFCC, Spectral Centroid, Spectral Rolloff, Spectral Flux, and Zero Crossing features are extracted using Marsyas. Then Marsyas KEA Support vector Machine (SVM) classifier and two algorithms (SMO and Multilayer Perceptron) in WEKA are used to classify the music pieces using the above features. The classification results achieved here are set as the baseline and all the experimental results are compared with these. This experiment corresponds to applying the proposed system on Timbral feature set (T).

To experiment our proposed technique three different classifiers are applied to the proposed feature sets. Because of high dimensionality of the proposed feature sets SVM based classifiers seem to be more suitable. However we apply a greedy search based systematic approach to select the most suitable classifier.

In the first step of our experiments, classification on timbral features (T) extracted from GTZAN is performed using a number of data mining algorithms and the baseline performance is evaluated. Based on the classification results of the baseline the best three classification methods (algorithms) are selected. These results correspond to those of (Tzanetakis, 2007) and provide the baseline for comparison. Then different combinations of beat features (B), chord occurrence features (C), genre likelihoods (L3 and L4), and genre predictions (P3 and P4) are added to the feature set. Note that the genre likelihoods and genre predictions are extracted by ChordMiner

[8] http://marsyas.sness.net/docs/manual/marsyas-user/kea.html

twice: once using patterns of 3 chords (L3 and P3) and once using patterns of 4 chords (L4 and P4).

4.2. Baseline Performance

As mentioned in the previous chapters, timbral features are the most important features in determining the music genre and high level features just provide enhancements and improve the classification results. So we select the algorithms which perform more accurately in classifying music genre using the timbral features and use them throughout the rest of experiments.

In the initial step of the experiments eight different data mining algorithms are applied on the Timbral feature set (T). Out of these eight algorithms one is the KEA's SVM classifier which is developed specifically for music processing problems and the rest are WEKA classifiers. KEA's SVM is the same algorithm used by Tzanetakis (2007) for classification. From these eight classifiers, the 3 algorithms with the highest classification accuracy are selected and used throughout the rest of the experiment. The classification accuracies of these eight algorithms are illustrated in *Table 3*.

Table 3. Performing classification on timbral features

Tool	Classifier	Accuracy %
KEA	SVM Classifier	79.7798
WEKA	Multilayer Perceptron (Neural Networks based)	71.6
WEKA	SMO (SVM based)	70.5
WEKA	RBF Network (Gaussian based)	62.5
WEKA	IBK (K Nearest Neighbour based)	62.3
WEKA	Logistics (multinomial logistic regression based)	61.1
WEKA	Bayes Net (Bayesian Network based)	59.9
WEKA	Bagging (on REP Tree which is a decision/regression tree based classifier)	59.3

All the above algorithms have been applied on a 10 fold cross validation. From these algorithms KEA's SVM Classifier, WEKA's Multilayer Perceptron, and WEKA's SMO are selected because they had resulted in significantly greater classification accuracy. These results also endorse the fact that such algorithms are more suitable for classifying high dimensional data.

4.3. ChordMiner Performance

ChordMiner as described in S*ection 3.8* is an algorithm written in C++ to perform Chord Mining. ChordMiner estimates the likelihood of music pieces belonging to different genres based on the chord Progressions of the songs. It also predicts the music genre by selecting the genre with the highest likelihood. In this section the prediction accuracy of ChordMiner is analysed and compared to similar techniques. This analysis provides good overview on capability of ChordMiner in processing and extracting useful information from chord progressions of the songs.

Table 4 and 5 illustrate the confusion matrix of genre prediction by ChordMiner. There are 100 songs per genre so the numbers in each row and each column add up to be 100. These confusion matrices demonstrate the ability of ChordMiner to classify music genre just by using chord progression features and without the aid of sophisticated machine learning algorithms. ChordMiner performs this classification simply by selecting the genre which has the highest likelihood.

The average accuracy when using sequence of 3 chords is 46.80% and when using sequence of 4 chords is 54.20%. In both cases the performance is very good for metal but poor in pop and reggae. However a lot of songs have been wrongly classified to metal. The opposite of this phenomenon can be observed in hip hop and jazz were there are only very few false positives classifications.

Table 4. Confusion matrix of ChordMiner for sequence of 3 chords (bl: Blues, cl: Classical, co: Country, di: Disco, hi: Hip Hop, ja: Jazz, me: Metal, po: Pop, re: Reggae, ro: Rock)

	bl	**cl**	**co**	**di**	**hi**	**ja**	**me**	**po**	**re**	**ro**
bl	41	3	1	14	0	0	29	0	4	8
cl	2	52	14	7	1	1	11	5	1	6
co	1	5	62	5	0	0	18	2	1	6
di	1	3	9	52	0	0	24	0	3	8
hi	7	1	3	14	34	0	27	1	6	7
ja	3	1	4	18	0	39	20	3	4	8
me	1	1	1	2	0	2	89	1	1	2
po	1	4	11	16	0	0	22	17	6	23
re	4	3	8	11	0	0	20	2	17	35
ro	1	1	14	6	0	1	28	1	1	47

Table 5. Confusion matrix of ChordMiner for sequence of 4 chords (bl: Blues, cl: Classical, co: Country, di: Disco, hi: Hip Hop, ja: Jazz, me: Metal, po: Pop, re: Reggae, ro: Rock)

	bl	cl	co	di	hi	ja	me	po	re	ro
bl	45	4	2	9	0	0	31	2	2	5
cl	12	64	7	5	0	1	7	1	0	3
co	1	2	74	4	0	2	11	1	1	4
di	3	1	6	65	0	0	15	1	2	7
hi	8	0	2	10	38	0	23	3	5	11
ja	2	1	2	11	0	58	14	3	6	3
me	1	1	3	2	0	2	81	1	2	7
po	1	2	8	9	0	0	16	27	5	32
re	2	2	4	8	0	0	17	2	32	33
ro	1	1	9	3	0	0	21	1	6	58

A technique similar to those of ChordMiner has been recently proposed by Pérez-Sancho et al. (2009). Pérez-Sancho et al. (2009) propose a genre classification method solely based on chord Progression. Our propopsed techniques and those of Pérez-Sancho et al. (2009) are both based on the same idea of using a probabilistic model to extract useful information from the chord patterns by assessing how often chord progressions are present in different genres. Essentially these two techniques both implement the same idea using two different methods. It is very intresting that such a similar idea was used by two separate researches at the same time without knowing each other.

However the proposed technique of Pérez-Sancho et al. (2009) is for classifying MIDI music files and not the music signal whereas our proposed method applies to the music signal (e.g. MP3 files). Another difference is that the aim of Chord Mining as described earlier in this thesis is to construct a feature vector that can be used in cunjunction with other features in the classification but the techniques of Pérez-Sancho et al. (2009) aim on classifying music only using the chord progressions. Here we compare the classification accuracy of ChordMiner with those claimed by Pérez-Sancho et al. (2009).

Pérez-Sancho et al. (2009) used a hierarchical ground truth. The dataset that they used for their experiment has two levels. On the first level there are three genres and on the second level each of these genres are devided into three sub genres. Therefore the second level consists of nine genres.

Pérez-Sancho et al. (2009) achieved a classificaiton accuracy of 72% to 87% on the 3 class problem (first level hierarchy) and 40% to 64% on the 9 class problem (second level hierarchy).

The classification of ChordMiner on a 10 class problem is 46.80% (3 chords) and 54.20% (4 chords).

However, it should be stated that the method proposed by Pérez-Sancho et al. (2009) is more sophisticated than those of ChordMiner. They apply some musical knowledge into their technique and their method is integrated with sophisticated and advanced statistical and data mining techniques. ChordMiner can achieve 46.80% classification accuracy by building a simple probabilistic model and selecting the genre with the highest likelihood.

Running sophisticated data mining algorithms such as neural networks on genre likelihoods produced by ChordMiner results in a significant increase of the classification accuracy. As illustrated in *Section 4.4.3* performing classification on genre likelihoods (L4) using Multilayer Perceptron classifier results in 61.7% classification accuracy which is very close to the maximum accuracy of 64% reported by Pérez-Sancho et al. (2009). And this 61.7% accuracy is achieved when classifying 10 genres while the 64% accuracy reported by Pérez-Sancho et al. (2009) is achieved when classifying 9 genres.

Appendix A provides statistical details on the 10 most popular chord progressions of different genres extracted by ChordMiner. This provides an overview of how much different genres are close to each other in terms of chord progressions. This data has been captured by ChordMiner as intermediate output data during the model building process to allow further analysis of the algorithm.

4.4. Effects of High Level Features on Genre Classification

In this section the experimental results of performing classification on different feature sets are provided and analysed. The three classifiers used are KEA's SVM Classifier, WEKA's Multilayer Perceptron, and WEKA's SMO. The feature sets that are used in the classification experiments are those described in *Section 3.9. Table 6* indicates the classification accuracies achieved by running these three algorithms on each of the feature vectors.

Table 6. Classification Results (The first row, T, indicates the baseline. The features are abbreviated to T: Timbral features, B: Beat features, C: Chord occurrence features, L3: Genre likelihood features of 3 chords, L4: Genre likelihood features of 4 chords, P3: prediction features of 3 chords, P4: prediction features of 4 chords)

	KEA	Multilayer Perceptron	SMO
T	79.78%	71.60%	70.50%
B	24.52%	25.90%	23.80%
TB	80.68%	74.20%	72.30%
C	42.34%	32.80%	36.30%
TC	85.39%	72.50%	71.60%
TBC	85.99%	73%	73%
L3	48.55%	59.80%	46.10%
TL3	86.59%	80.60%	78.30%
TP3	78.68%	75.90%	73.70%
TL3P3	82.18%	79.40%	77.60%
TBL3	90.79%	84%	82.80%
TBCL3	90.29%	79.80%	77.10%
L4	43.24%	61.70%	41.40%
TL4	86.89%	80.20%	78.30%
TP4	79.78%	76.90%	75.80%
TL4P4	86.89%	81.90%	79.50%
TBL4	88.99%	82.40%	80.20%
TBCL4	90.39%	81.60%	79.70%

Based on the data illustrated in *Table 6* the effects of different features on the classification accuracy are discussed here.

4.4.1. Effects of Beat Feature on Classification Results

As illustrated in *Table 6* beat features (B) when used alone perform very poorly in classifying music genre. A maximum accuracy of 25.90% is achieved when classifying music genre using

only the beat feature vector. However combining beat features and timbral features (TB) do increase the classification accuracy. Using beat features in conjunction with timbral features result in a maximum of 2.6% increase in classification accuracy when compared to the baseline (T).

4.4.2. Effects of Statistical Chord Features on Classification Results

Statistical chord features (C) when used alone result in a maximum classification accuracy of 42.34%. This classification accuracy is still significantly lower than the baseline (T). However as illustrated in *Table 6* a combination of statistical chord features and timbral features (TC) provide a maximum of 5.61% improvement over the baseline (T).

Even more increase in the classification accuracy is observed when statistical chord features are combined with both beat and timbral features (TBC). A maximum improvement of 6.21% is observed when comparing the feature set containing the timbral, beat, and statistical chord features with the baseline (T).

4.4.3. Effects of Chord Progression Features on Classification Results

As demonstrated in *Table 6*, chord progression features had the most significant effect on the classification. The high level information of chord progressions is manifested in four different feature vectors (L3, P3, L4, and P4). The L3 and L4 feature vectors include the genre likelihoods estimated by ChordMiner, while P3 and P4 are just the predictions of ChordMiner.

The genre likelihood features extracted from chord progressions (L3 and L4) perform reasonable when used on their own but they still perform poor compared to timbral features. A maximum accuracy of 61.70% is achieved only using the feature vector of genre likelihoods extracted from progressions of 4 chords (L4). For classification accuracy of P3 and P4 please refer to S*ection 4.3*.

Combining the ChordMiner predictions with timbral features (TP3 and TP4) improves the classification accuracy by a maximum 5.30%. The combination of genre likelihoods and timbral features (TL3 and TL4) increase the classification accuracy by a maximum of 9%. Combining the timbral feature vector with both the likelihoods and prediction feature vectors improves the classification results even further. These combinations (TL3P3 and TL4P4) result in a maximum increase of 10.30% in the classification accuracy.

Consolidating likelihoods, timbral, beat, and statistical chord features (TBCL3 and TBCL4) also increase the classification accuracy with a maximum of 10.61%. But the best performance belongs to the combination of likelihoods, timbral and beat features (TBL3 and TBL4). A

maximum improvement of 12.4% can be observed by comparing TBL3 (Combination of Timbral features, Beat features, and genre likelihoods extracted from progressions of 3 chords) and the baseline using the Multilayer Perceptron algorithm.

Another important observation is that although the genre likelihoods extracted from sequences of 4 chords perform better on their own, the likelihoods extracted from sequence of 3 chords perform better when combined with timbral and beat features

4.5. Discussion and Summary

Table 6 indicates the classification results of our experiment using all the features that we have extracted. The classification has been performed using a Multilayer Perceptron implementation of Neural Network and a Sequential Minimal Optimization (SMO) implementation of SVM in WEKA and an SVM implementation in KEA. The SVM implementation of KEA performs significantly better than the WEKA algorithms. But in WEKA the Neural Network algorithm outperforms the SVM.

Overall the KEA's SVM classifier performs the best among the two other algorithms. The highest classification result in our experiments (90.79%) is achieved by performing classification using this algorithm on the TBL3 feature vector. This accuracy of 90.79% indicates and improvement of 11.01% over the baseline. Nevertheless the maximum improvement of 12.40% mentioned in the previous section is achieved by classifying the same feature vector using the Multilayer Perceptron which is a result of 84% classification accuracy.

Figure 2 illustrates and compares the best results achieved using the important feature sets when using Kea's SVM classifier. This classifier has been designed for music classification as a part of Marsyas and is the same classifier that has been used by Tzanetakis (2007). As indicated in Figure 2 each of the proposed features increase the classification accuracy. And the maximum improvement is achieved with a combination of timbral features, beat features, and genre likelihoods extracted from progressions of 3 chords. This combination improves the results by 11.01% when compared to timbral features and by 10.11% when compared to timbral and beat features.

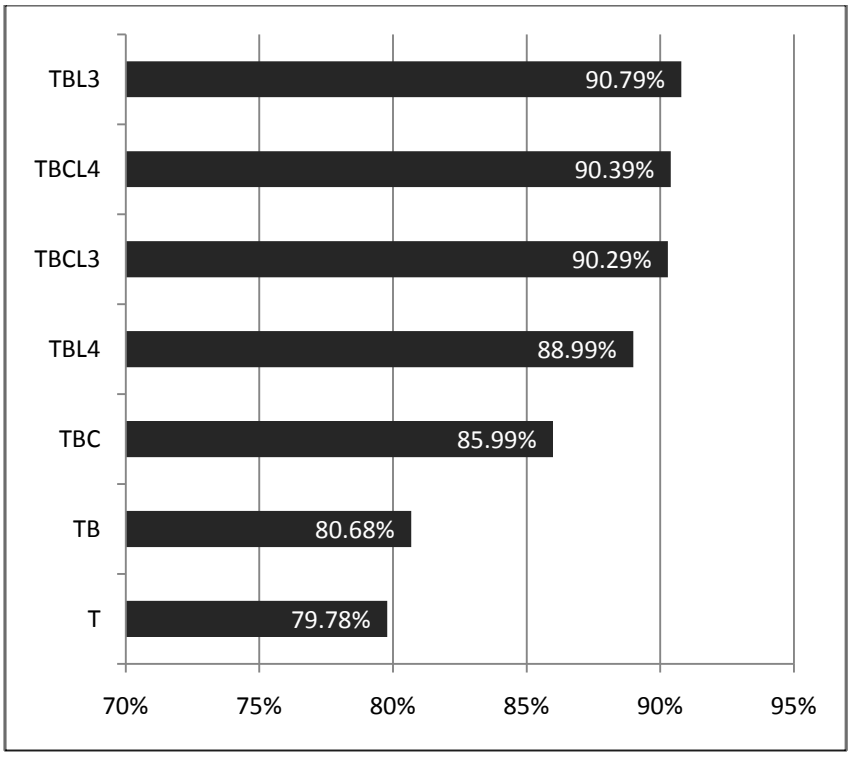

Figure 2. The best classification results using Kea's SVM

The experimental studies performed here show that the best combination of features to be used is timbral, beat, and genre likelihoods extracted from progressions of 3 chords (TBL3). However TBL4, TBCL4, TBCL3, and TL4P4 feature vectors also perform well in the classification and provide significant improvements over the baseline.

Chapter 5: Conclusion

In this thesis we have proposed the use of high level features particularly chords and chord progressions and used them in conjunction with low level features to enhance and improve music genre classification accuracy. Our results agree with the proposition in (McKay, 2004) that a combination of high and low level features provide the highest classification accuracy.

To capture the chord information we have counted the number of roots and modes of the chords present in songs. And for chord progressions we had developed a pattern matching algorithm called ChordMiner that predicts the likelihood of songs belonging to different genres. This algorithm captures and summarizes the chord progression information to a format that can be used in conjunction with other features as an input to data mining algorithms. In other words ChordMiner translates the chord progressions into genre likelihoods and by doing so the extremely wide, unorganized, and unpredictable problem space is converted to an organized, unchanging, and narrow problem space so the chord progression information can be coupled with other features for more accurate classification. the proposed system achieved a maximum of 61.70% accuracy on a dataset of 10 genres only using chord progressions which is comparable to those of (Pérez-Sancho et al., 2009) with a maximum accuracy of 64% on a dataset of 9 genres.

The results achieved in our experiments confirmed all the previous works in the sense that low level features are essential in genre classification, but high level features are effective too. We found that high level features do contain extra information which increases the classification accuracy significantly therefore they should not be ignored.

To conclude, the following are the main contributions of this thesis:

- A review of the existing works has been performed.
- A system is proposed to address the limitations in the literature which is lack of high level features in genre classification of music signal.
- In this system high level features such as chord and chord progression features are proposed to be used in conjunction with timbral features to enhance the classification accuracy.
- The statistical chord feature vector is proposed to capture chord information.
- A technique called chord mining is proposed to capture high level chord progression information into a feature vector.
- Experimental studies are performed to evaluate the proposed system, identify the best combination of features, and identify the best classifier for this task. These experimental studies show that KEA's SVM classifier produces the highest improvement when used on a combination of timbral, beat, and chord progression features (TBL3). A maximum improvement of 12.4% shows that our proposed system does improve the genre classification accuracy.

The achievements of this thesis signify the need for more musicology and psychoacoustic research in this area. There is also the need for more research experimenting with other high level features such as conceptual tempo and incorporating detailed musical knowledge of chord progressions and instruments to make the algorithms smarter.

Appendix A

During the model building process ChordMiner outputs the 10 most frequent chord progressions for each genre. However we cannot say for certain that these progressions have the most significant impact on the model. The total number of possible chord progressions when the sequences are made of three chords is 108^3 (1,259,712) and the total number of possible progressions when they are made of four chords is 108^4 (136,048,896). Nevertheless out of these possible progressions three to five thousand unique chord progressions are used on average to build the model for each genre when there are three chords in each progression. And on average four to eight thousand unique chord progressions are used to build the model for each genre when there are four chords in each chord progression. So the top 10 progressions may not have a significant impact on determining the genres. These progressions are just presented here to provide an outline of the intermediate steps of the ChordMiner and may be helpful in understanding how ChordMiner determines the genre likelihoods.

Table 7 and *Table 8* show how often these chord progressions appear in each genre. The genre names are abbreviated to cl: Classical, co: Country, bl: Blues, di: Disco, hi: Hip Hop, ja: Jazz, me: Metal, po: Pop, re: Reggae, and ro: Rock.

Table 7. Frequency of most Common chord progressions of 3 chords appearing in different genres

Chord Progression	cl	co	bl	di	hi	ja	me	po	re	ro
A Major - A Minor - A Major		35	37		20		74		28	48
A Major - D Major - A Major		40								
A Major - E Major - A Major		27								
A Major - F# Minor - A Major									22	
A Minor - A Major - A Minor				25			55		45	40
A# Major - A# Minor - A# Major			33		18					
A# Major - D Minor - A# Major	13									
A# Major - D# Major - A# Major	13									
A# Major - F Major - A# Major	15									
A# Major - G Minor - A# Major	12									
A# Minor - A# Major - A# Minor			33		14					
B Major - B Minor - B Major				28			33			
B Minor - B Major - B Minor				35			34			
C Major - C Minor - C Major				37	18	16		25		23
C Major - E Minor - C Major		25								
C Major - F Major - C Major	19	24								

Chord Progression	cl	co	bl	di	hi	ja	me	po	re	ro
C Major - G Major - C Major	17	29								28
C Minor - C Major - C Minor				59	15	18		42		
C Minor - D# Major - C Minor				30		20		17		
C# Major - C# Minor - C# Major							42			
C# Minor - C# Major - C# Minor					17		31			
D Major - A Major - D Major		37								
D Major - D Minor - D Major							40			23
D Major - G Major - D Major										23
D Minor - D Major - D Minor						15				
D# Major - C Minor - D# Major						19				
D# Major - D# Minor - D# Major			36							
E Major - E Minor - E Major			40				87			22
E Major - G# Minor - E Major		22								
E Minor - E Major - E Minor			46				78			
E Minor - G Major - E Minor									30	
F Major - A Minor - F Major	15								24	
F Major - A# Major - F Major	15									
F Major - C Major - F Major	13	28								
F Major - F Minor - F Major			46							
F Minor - F Major - F Minor			30							
F Minor - G# Major - F Minor						15				
F# Major - F# Minor - F# Major								17		
F# Minor - A Major - F# Minor									32	
G Major - C Major - G Major	14	26							29	
G Major - D Major - G Major										23
G Major - E Minor - G Major				25				23	32	
G Major - G Minor - G Major			34	28	21	21		49	37	47
G Minor - A# Major - G Minor				24						
G Minor - G Major - G Minor			30	43	23	26		46	37	38
G# Major - D# Major - G# Major								20		
G# Major - F Minor - G# Major						16				
G# Major - G# Minor - G# Major					22		30	32		
G# Minor - G# Major - G# Minor					19	14		27		

Table 8. Frequency of most Common chord progressions of 4 chords appearing in different genres

Chord Progression	cl	co	bl	di	hi	ja	me	po	re	ro
A Major - A Minor - A Major - A Minor		10	14	9	8		24		17	21
A Major - D Major - A Major - D Major		15								
A Major - F# Minor - A Major - F# Minor									11	
A Minor - A Major - A Minor - A Major			16		7		28		16	25
A# Major - A# Minor - A# Major - A# Minor			13		6					
A# Major - D Minor - A# Major - D Minor	5									
A# Major - F Major - A# Major - F Major	5	14								
A# Major - G Minor - A# Major - G Minor						5				
A# Major - G Minor - G Major - G Minor						6				
A# Minor - A# Major - A# Minor - A# Major			19							
B Major - B Minor - B Major - B Minor				21			21			
B Minor - B Major - B Minor - B Major			11	21					12	
C Major - A Minor - C Major - A Minor	6									
C Major - C Minor - C Major - C Minor				15		5		9		
C Major - F Major - C Major - F Major	8	13								
C Major - G Major - C Major - G Major	7	12								
C Minor - C Major - C Minor - C Major				23	6			13		
C# Major - C# Minor - C# Major - C# Minor					7		22			12
C# Minor - C# Major - C# Minor - C# Major					6		20			12
D Major - A Major - A Minor - A Major		11								
D Major - A Major - D Major - A Major		17								
D Major - D Minor - D Major - D Minor							26			
D Major - G Major - D Major - G Major										9
D Minor - A# Major - D Minor - F Major	5									
D Minor - D Major - D Minor - D Major							20	8		
D# Major - C Minor - D# Major - C Minor						7				
D# Major - G# Major - D# Major - G# Major								9		
E Major - E Minor - E Major - E Minor			17				48			
E Minor - E Major - E Minor - E Major			23				43			9
E Minor - G Major - E Minor - G Major									12	
F Major - A# Major - D Minor - F Major	5									
F Major - A# Major - F Major - A# Major		11								
F Major - C Major - F Major - C Major		13								
F Major - F Minor - F Major - F Minor			16	9						

Chord Progression	cl	co	bl	di	hi	ja	me	po	re	ro
F Minor - F Major - F Minor - F Major			21							
F Minor - G# Major - F Minor - G# Major						8				
F# Major - B Major - F# Major - B Major										10
F# Major - F# Minor - F# Major - F# Minor								9		
F# Minor - A Major - F# Minor - A Major									15	
G Major - C Major - G Major - C Major	6	13							12	9
G Major - D Major - G Major - D Major	7									
G Major - E Minor - G Major - E Minor									10	
G Major - G Minor - G Major - G Minor				14	6	7		18	14	24
G Minor - G Major - G Minor - G Major			12	13	10	7		24	16	20
G# Major - F Minor - G# Major - F Minor						7				
G# Major - G# Minor - G# Major - G# Minor	6			11	11	7		13		
G# Minor - E Major - G# Minor - E Major								10		
G# Minor - G# Major - G# Minor - G# Major				9	7	10	17	14		

Table 9 and *Table 10* compare different genres based on the top 10 chord progressions. The number in each cell indicates the number of chord progressions that the two genres have in common. The higher the number the more similar are the genres in terms of their top 10 chord progressions.

Table 9. Genre similarity in terms of top 10 progressions of 3 chords

	cl	co	bl	di	hi	ja	me	po	re	ro
cl	10	4	0	0	0	0	0	0	1	1
co	4	10	1	0	1	0	1	0	2	2
bl	0	1	10	2	5	2	3	2	3	4
di	0	0	2	10	4	5	3	6	4	4
hi	0	1	5	4	10	5	3	6	3	4
ja	0	0	2	5	5	10	1	6	2	3
me	0	1	3	3	3	1	10	1	2	4
po	0	0	2	6	6	6	1	10	3	3
re	1	2	3	4	3	2	2	3	10	4
ro	1	2	4	4	4	3	4	3	4	10

Table 10. Genre similarity in terms of top 10 progressions of 4 chords

	cl	co	bl	di	hi	ja	Me	po	re	ro
cl	10	4	0	1	1	1	0	1	1	1
co	4	10	1	1	1	0	1	0	2	2
bl	0	1	10	4	4	1	4	1	4	4
di	1	1	4	10	6	5	3	6	4	3
hi	1	1	4	6	10	4	5	5	4	6
ja	1	0	1	5	4	10	1	5	2	2
me	0	1	4	3	5	1	10	2	2	5
po	1	0	1	6	5	5	2	10	2	2
re	1	2	4	4	4	2	2	2	10	5
ro	1	2	4	3	6	2	5	2	5	10

References

AUCOUTURIER, J. J. & PAMPALK, E. (2008) Introduction-From Genres to Tags: A Little Epistemology of Music Information Retrieval Research. *Journal of New Music Research,* 37**,** 87-92.

BAINBRIDGE, D., CUNNINGHAM, S. J. & DOWNIE, J. S. (2003) How people describe their music information needs: A grounded theory analysis of music queries. *fourth International Conference on Music Information Retrieval (ISMIR 2003).* Maryland, USA.

BARBEDO, J. G. A. & LOPES, A. (2007) Automatic Genre Classification of Musical Signals. *EURASIP Journal on Advances in Signal Processing 2007,* 2007.

BARRINGTON, L., YAZDANI, M., TURNBULL, D. & LANCKRIET, G. (2008) Combining feature kernels for semantic music retrieval. *Ninth International Conference on Music Information Retrieval (ISMIR 2008).* Philadelphia, USA.

BERGSTRA, J., CASAGRANDE, N., ERHAN, D., ECK, D. & KÉGL, B. (2006) Aggregate features and ADABOOST for music classification. *Machine Learning,* 65**,** 473 - 484

CHASE, A. R. (2001) Music discrimination by carp (Cyprinus carpio). *Animal Learning & Behavior,* 29**,** 336 - 353.

CHUA, B. Y. (2007) Automatic extraction of perceptual features and categorization of music emotional expression from polyphonic music audio signals. Monash University.

GABRIELSSON, A. & JUSLIN, P. N. (1996) Emotional Expression in Music Performance: Between the Performer's Intention and the Listener's Experience. *Psychology of Music,* 24**,** 68-91.

GJERDINGEN, R. O. & PERROTT, D. (1999) Scanning the dial: An exploration of factors in identification of musical style. *The annual meeting of the Society for Music Perception and Cognition (SMPC).* Evanston, USA.

GJERDINGEN, R. O. & PERROTT, D. (2008) Scanning the Dial: The Rapid Recognition of Music Genres. *Journal of New Music Research,* 37**,** 93.

HARTE, C. & SANDLER, M. (2005) Automatic Chord Identification using a Quantised Chromagram. *118th convention of the Audio Engineering Society (AES).* Barcelona, Spain.

HOLZAPFEL, A. & STYLIANOU, Y. (2008) Musical Genre Classification Using Nonnegative Matrix Factorization-Based Features. *IEEE Transactions On Audio, Speech, And Language Processing,* 16**,** 424-434.

KAMENETSKY, S. B. & HILL, D. S. (1997) Effect of Tempo and Dynamics on the Perception of Emotion in Music. *Psychology of Music,* 25**,** 149-160.

KOTOV, O., PARADZINETS, A. & BOVBEL, E. (2007) Musical genre classification using modified wavelet-like features and support vector machines. *IASTED European Conference: internet and multimedia systems and applications.* Chamonix, France.

LAMPROPOULOS, A. S., LAMPROPOULOU, P. S. & TSIHRINTZIS, G. A. (2005) Genre classification enhanced by improved source separation technique. *International Conference on Music Information Retrieval (ISMIR 2005).* London, UK.

LEE, K. (2007) A System for Automatic Chord Transcription Using Genre-Specific Hidden Markov Models. *International Workshop on Adaptive Multimedia Retrieval.* Paris, France.

LEON, P. J. P. D. & INESTA, J. M. (2007) A pattern recognition approach for music style identification using shallow statistical descriptors. *IEEE Trans. on Systems, Man, and Cybernetics,* 37**,** 248 - 257.

LIDY, T., RAUBER, A., PERTUSA, A. & IÑESTA, J. E. M. (2007) Improving Genre Classification by Combination of Audio and Symbolic Descriptors Using a Transcription System. *nternational Conference on Music Information Retrieval (ISMIR 2007).*

MCKAY, C. (2004) Automatic Genre Classification of MIDI Recordings. McGill University.

MCKAY, C. & FUJINAGA, I. (2006) Musical genre classification: Is it worth pursuing and how can it be improved? *7th International Conference on Music Information Retrieval (ISMIR 2006).* Victoria, Canada

MCKAY, C. & FUJINAGA, I. (2008) Combining features extracted from audio, symbolic and cultural sources. *Ninth International Conference on Music Information Retrieval (ISMIR 2008).* Philadelphia, USA.

MENG, A., AHRENDT, P., LARSEN, J. & HANSEN, L. K. (2007) Temporal feature integration for music genre classification. *IEEE Transactions on Audio, Speech and Language Processing,* 15**,** 1654 - 1664.

PAIEMENT, J. F., ECK, D. & BENGIO, S. (2005) A Probabilistic Model for Chord Progressions'. *Sixth International Conference on Music Information Retrieval (ISMIR 2005).* London, UK

PANAGAKIS, I., BENETOS, E. & KOTROPOULOS, C. (2008) Music genre classification: a multilinear approach. *Ninth International Conference on Music Information Retrieval (ISMIR 2008).* Philadelphia, USA.

PÉREZ-SANCHO, C., RIZO, D. & INESTA, J. M. (2009) Genre classification using chords and stochastic language models. *Connection Science,* 21**,** 145 - 159.

RABINER, L. & JUANG, B. H. (1993) *Fundamentals of Speech Recognition*, Englewood Cliffs, N.J. : PTR Prentice Hall.

RIZO, D., LEON, P. J. P. D., PEREZ-SANCHO, C., PERTUSA, A. & INESTA, J. M. (2006) A Pattern Recognition Approach for Melody Track Selection in MIDI Files. *7th International Conference on Music Information Retrieval (ISMIR 2006).* Victoria, Canada

SHEN, J., SHEPHERD, J. & NGU, A. (2006) InMAF: indexing music databases via multiple acoustic features. *The 2006 ACM SIGMOD international conference on Management of data.* Chicago, USA.

SHEPARD, R. N. (1964) Circularity in judgment of relative pitch. *The journal of the acoustical society of America,* 35**,** 2346 - 2353.

SUNDARAM, S. & NARAYANAN, S. (2007) Experiments in Automatic Genre Classification of Full-length Music Tracks using Audio Activity Rate. *IEEE 9th Workshop on Multimedia Signal Processing (MMSP 2007).* Crete, Greece.

TEKMAN, H. G. & HORTACSU, N. (2002) Aspects of Stylistic Knowledge: What Are Different Styles Like and Why Do We Listen to Them? . *Psychology of Music,* 30**,** 28 - 47.

TSUCHIHASHI, Y., KITAHARA, T. & KATAYOSE, H. (2008) Using bass-line features for content-based mir. *Ninth International Conference on Music Information Retrieval (ISMIR 2008).* Philadelphia, USA.

TURNBULL, D., BARRINGTON, L. & LANCKRIET, G. (2008) Five approaches to collecting tags for music. *Ninth International Conference on Music Information Retrieval (ISMIR 2008).* Philadelphia, USA.

TZANETAKIS, G. (2007) MARSYAS Submissions to MIREX 2007. *MIREX 2007.*

TZANETAKIS, G. & COOK, P. (2000) Marsyas: A framework for audio analysis. *Organized Sound,* 4.

TZANETAKIS, G. & COOK, P. (2002) Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing,* 10**,** 293–302.

VIRTANEN, T. (2004) Separation of Sound Sources by Convolutive Sparse Coding. *ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing (SAPA).* Jeju, Korea.

ZHU, J., XUE, X. & LU, H. (2004) Musical genre classification by instrumental features. *Computer Music Conference (ICMC2004).*