2010

An empirical evaluation of computational and perceptual multi-label genre classification on music / Christopher Sanden

Sanden, Christopher

Lethbridge, Alta. : University of Lethbridge, Dept. of Mathematics and Computer Science, c2010

AN EMPIRICAL EVALUATION OF COMPUTATIONAL AND PERCEPTUAL MULTI-LABEL GENRE CLASSIFICATION ON MUSIC

CHRISTOPHER SANDEN
Bachelor of Science, University of Lethbridge, 2007

A Thesis
Submitted to the School of Graduate Studies
of the University of Lethbridge
in Partial Fulfillment of the
Requirements for the Degree

MASTER OF SCIENCE

Department of Mathematics and Computer Science
University of Lethbridge
LETHBRIDGE, ALBERTA, CANADA

Abstract

Automatic music genre classification is a high-level task in the field of Music Information Retrieval (MIR). It refers to the process of automatically assigning genre labels to music for various tasks, including, but not limited to categorization, organization and browsing. This is a topic which has seen an increase in interest recently as one of the cornerstones of MIR. However, due to the subjective and ambiguous nature of music, traditional single-label classification is inadequate.

In this thesis, we study multi-label music genre classification from perceptual and computational perspectives. First, we design a set of perceptual experiments to investigate the genre-labelling behavior of individuals. The results from these experiments lead us to speculate that multi-label classification is more appropriate for classifying music genres. Second, we design a set of computational experiments to evaluate multi-label classification algorithms on music. These experiments not only support our speculation but also reveal which algorithms are more suitable for music genre classification. Finally, we propose and examine a group of ensemble approaches for combining multi-label classification algorithms to further improve classification performance.

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Dr. John Zhang, whose expertise, understanding, and patience, made this thesis possible. In addition, I would like to thank the other members of my committee, Dr. Howard Cheng, and Dr. Rolf Boon, for their continual support and constructive criticisms throughout my studies. A very special thanks goes out to my colleague Chad Befus for his motivation, collaboration, and committed friendship. I am further indebted to Dr. Matthew Tata for his indispensable advice and assistance towards conducting my perceptual experiments. I would also like to thank my other colleagues for providing a stimulating and fun environment in which to learn and grow. I am especially grateful to Mahmudul Hasan, Tarikul Sabbir, and Ben Burnett. Finally, I cannot find words to express my gratitude to my friends and family who helped contribute to my success. Specifically, I would like to thank Mecole Maddeaux-Young without whose support this study would not have been successful.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

For the past decade, digital music collections have been growing in volume due to advances in technologies such as storage capacity, network transmission, data compression, information retrieval, etc. Never before have listeners had access to such extensive collections of music. Furthermore, the steep rise in music downloading has created a major shift in the music industry away from physical media formats and towards electronic distribution. Large online music providers now offer consumers millions of songs from catalogs of material extending over many decades, genres, and styles. Application requirements from users, including music recommendation, recognition, and categorization are becoming more and more demanding [62].

At present, digital music collections are commonly represented and accessed through *textual meta-data*, such as *genre*, *style*, *mood*, *release year*, and *artist*. For instance, traditional methods of searching and indexing categorize audio pieces into music genres such as *Classical*, *Rock*, *Jazz*, *Pop*, and others. This method relies on human experts as well as amateurs to annotate the music [81]. Although this meta-data can be rich and descriptive, it is difficult to maintain consistency when music collections become large. Moreover, annotating music manually is tedious, time-consuming, and erroneous [23]. For example, Microsoft's MSN Music Search Engine required the assistance of 30 musicologists over a period of one year in order to manually label approximately 200,000 songs [1, 13].

All these call for novel approaches to organize, browse, and update music, which are collectively referred to as *Music Information Retrieval.*

1.1 Music Information Retrieval

Music Information Retrieval (*MIR*) is a growing interdisciplinary area, engaging in the design and implementation of algorithmic approaches to managing digital music collections for preservation, access, and other uses [28]. Practitioners come from backgrounds, including, but not limited to, *computer science*, *information retrieval*, *audio engineering*, *cognitive science* and *musicology.* [28] The primary goal of MIR research is to facilitate access to the world's vast music collections, both new and historical, on a level equal to that currently being afforded by text-based search engines, such as *Google* [1]. The growth of interest in MIR is evidenced from the number of publications in multimedia conferences, e.g., ISMIR (International Conference on Music Information Retrieval), ICMC (International Computer Music Conference), etc.

MIR approaches can be typically categorized into two groups: (1) those that are based on meta-data and (2) those that are based on the music content directly. Due to the limited amounts of meta-data and the uncertain quality associated with the data, meta-based approaches are often seen as less reliable than their alternative. Content-based approaches, on the other hand, describe a music piece by a set of features that are directly computed from its content. For this reason, we believe content-based approaches provide greater potential towards the development of novel approaches to MIR problems. Below is a list of some typical problems along this direction.

- *Automatic genre classification*: The task of automatically separating music into different groups such that each group uniformly represents a music genre, i.e. the classification of music based on genres [62].

- *Fingerprinting*: The process of creating an acoustic fingerprint, a reproducible hash

[1] http://www.google.com.

extracted from the audio content, that can identify an audio sample or locate similar pieces in a collection [77, 78].

- *Tag prediction*: Automatic textual annotation of music with tags that listeners would likely use or find helpful. The tags associated with a song can be used to search for new music (tag-based browsing) or to automatically generate music recommendations [32, 51].
- *Play-list generation*: The automatic formulation of playlists consisting of music pieces that satisfy some ordering criteria. This is typically done with respect to content descriptors previously collected by listeners [4].
- *Mood classification*: The task of automatically classifying music based on mood or emotion [66, 84].
- *Music recommendation*: The automatic recommendation of music to a listener. It consists of suggesting, providing guidance, or advising on interesting music. Music recommendation systems are generally structured to find songs which are similar to an input song based on a defined metric [4].

This thesis focuses on the automatic classification of music. We attempt to classify music into different genres using information derived from the content of the audio. This is a topic which has seen an increased interest recently as one of the cornerstones of music information retrieval.

1.2 Automatic Music Genre Classification

As a fundamental task of music information retrieval, automatic genre classification has attracted considerable attention. For instance, musical genres have been historically used as categorical descriptions to organize music collections. Although no universally accepted definition of genres exists, a genre can be characterized by the common characteristics

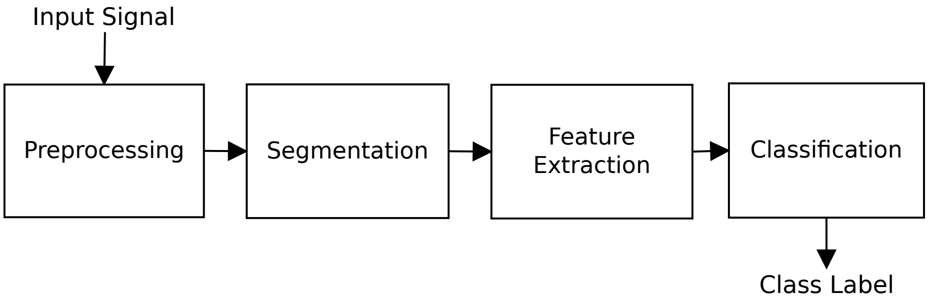

Figure 1.1: The generic process of automatic genre classification.

shared by its members. These characteristics are related to the instrumentation, harmonic content, rhythmic structures, scales, modes, etc. [75]

One of the easiest ways to annotate genre information from a music database is to ask human experts. However, human annotations are difficult to maintain and in cases are not comprehensive, due to the ambiguity and subjectivity that is introduced in this process [53]. Moreover, music genre annotation by human experts is an involved process, in terms of financial and labor costs. Therefore, manual annotation is not sufficient to handle a large volume of music titles.

In response to these, automatic genre classification techniques are being investigated and developed to assist in annotating large collections of digital music. In general, these techniques make use of perceptual, statistical, and spectral features derived from the audio content to automatically label a music piece. This is also in accordance with the statement from Tzanetakis *et al.* [76] that "there are perceptual criteria related to the texture, rhythmic structure and instrumentation of music that can be used to characterize a particular genre [algorithmically]". In addition, automatic genre classification facilitates structuring and organizing large archives of audio.

The basic process of automatic genre classification is composed of four steps, as shown in Figure 1.1, including: (1) Preprocessing, (2) Segmentation, (3) Feature extraction, and (4) Classification. The preprocessing step consists of performing operations, such as *normalization* and *downsampling*, to an audio signal. The purpose of this step is to provide a low-level signal representation on which the succeeding segmentation and feature extraction steps can be conducted [47]. The segmentation step partitions a signal into a series of sections that are of perceptually, statistically, or spectrally "meaningful", while the feature extraction step describes each section using set of characteristics [16, 58]. These two steps provide a low-dimensional representation as the basis for the audio classification step, in which a classification model is built to predict the genre of an unknown music piece. In Chapter 2 we examine these steps in detail.

1.2.1 Current State in Automatic Genre Classification

It is obvious that automatic genre classification has value in both research and applications. However, as McKay indicates [48], there is some controversy regarding the current state of automatic genre classification with some works even suggesting that it be abandoned in favor of a more general similarity search problem. For instance, genre-labeling is intrinsically difficult as the ground truth against which it is compared is based upon subjective responses. It has been suggested that only limited agreement can be achieved among human annotators when classifying music by genres. This imposes an unavoidable ceiling on the performance of automatic genre classification [48]. Moreover, the time and expertise needed to manually classify a corpus of music poses serious obstacles to generating quality ground truth. To further complicate matters, the understanding of existing genres can change with time and even merge, which can necessitate the re-annotation of ground truth [48].

Despite these controversies, genre categorization provides a common vocabulary which can be used to discuss musical categories. Additionally, users are already accustomed to browsing music collection by genres, and this approach is proven to be at least reasonably effective [48]. Although one may argue that in automatic genre classification generating "ground truth" is a serious issue that needs to be handled, manually labeling growing music collections is a much greater challenge. Therefore, continuing efforts in automatic genre classification have much to offer.

1.3 Contribution

In the current practice of automatic genre classification, an important issue that has rarely been addressed is the assignment of multiple genre labels to individual music pieces. It is often observed that a music piece could belong to multiple genres. Furthermore, different portions of a music piece might be recognized as different genres which stand in contrast to each other. Therefore, it is desirable that automatic genre classification be modeled as a multi-label problem.

The work presented in this thesis is motivated by the current state of automatic genre classification approaches. A survey of existing music information retrieval literature reveals little work focusing on multi-label genre classification. Furthermore, there has been limited work investigating genre-labeling behavior of individuals. Therefore, in this thesis, we investigate various issues pertaining to the task of multi-label genre classification.

First, we will explore the genre-labeling behavior of humans. The purpose of this is to investigate inter-song similarity and dissimilarity, with regard to music genres. We are interested in exploring how participants classify music excerpts extracted from the same piece of music. This stands in contrast to other human genre-labeling studies which compare the performance of each participant to some ground truth.

To support our speculation that a music piece might be a composition of multiple genres, we design a set of classification experiments. Insight is offered into what classification algorithms and parameters are best suited for the task of multi-label genre classification. While previous studies have used these techniques for detecting emotion in music, to the best of our knowledge, our attempt is the first to use them for music genre classification.

A wealth of knowledge exists in machine learning for improving classification accuracy. However, there has been little work concentrating on improvements to multi-label classification performance. In our work we propose a set of ensemble techniques for improving multi-label genre classification. Specifically, background information relating to genre structure is used in an attempt to improve classification performance.

1.4 Outline

The thesis is outlined as follows. Chapter 2 reviews previous works related to perceptual and algorithmic genre classification. Moreover, issues relating to the use of music genres are explored.

In Chapter 3 a series of perceptual experiments are performed to explore the multi-genre-labeling behavior of individuals. Given a set of excerpts from an audio recording, participants are asked to classify each excerpt and assign a genre class to it. While previous studies have explored the genre-labeling behavior of participants, to the best of our knowledge, there have been no studies investigating perceptual multi-genre-labeling.

In Chapter 4, a selection of multi-label classification algorithms are evaluated for the task of genre classification. Issues pertaining to the creation of a dataset are explored. Results are presented for various feature extraction parameters. Specifically, classification performance is presented for an assortment of frame sizes. Finally new techniques for improving multi-label genre classification, based on an ensemble of classifiers approach, are presented in Chapter 5. These techniques aim to improve classification performance and are evaluated for their effectiveness.

In Chapter 6, we present our conclusion and some discussion including limitations and recommendations for future research.

Chapter 2

Background

This chapter reviews previous works related to perceptual and algorithmic genre classification and is structured as follows. Section 2.1 discusses issues pertaining to music genres and human categorization of music. Section 2.2 presents relevant background on content-based audio analysis, including feature extraction and segmentation techniques. Section 2.3 proceeds with an overview of classification algorithms widely used in music retrieval and Section 2.4 focuses on the automatic classification of music. We dedicate some discussions to multi-label classification algorithms in Section 2.5 and introduce a set of evaluation measures for determining predictive performance in Section 2.6. Finally, previous works on the multi-label classification of music are presented in Section 2.7.

2.1 Music Genres

Music genres are labels that have been created to help describe the vast universe of music and have historically been used as categorical descriptions to organize music collections. However, no universally accepted definition of music genres exists as boundaries between genres become vague due to complex interactions among historical, cultural, and personal backgrounds [75].

It is commonly accepted that the definition of most music genres is subjective; the boundary of each music genre can be based on individual perspectives. As a result, genre boundaries can shift from individual to individual [3]. These observations have led people to suggest a new definition of genre classification for the purpose of *Music Information Retrieval* (MIR) [54]. In spite of this, musical genre is still the primary descriptor used to classify music [1, 53] and it is clear that the members of some genres share certain characteristics [75]. Although a genre may represent a simplification of an artist's musical discourse, it is of great interest as a summary of some shared characteristics [62].

Research into music genres attempts to address issues related to how they are created, defined, and perceived. Fabbri [24] suggests that music genres can be characterized using a set of rules including behavior, economical, social, ideological, and technical. Note that only the last rule deals with musical content. In addition, Firth [26] and Brackett [9, 10] offer insight into how music genres are formed, characterized and constructed. However, Aucouturier and Pachet [1] describe that "[musical genres are] not founded on any intrinsic property of the music, but rather depends on cultural extrinsic habits". Craft *et al.* [17] propose that there are three factors influencing how an individual assigns a genre label to a music piece: (1) The number of musical cues associated with different genres in the piece; (2) The social and cultural background of the individual; and (3) The individual's personal experience of the genres involved.

2.1.1 Genre Labeling by Humans

One of the easiest way to extract genre information from a collection of music is to ask human experts. However, in addition to the deficiencies outlined in Chapter 1 regarding the use of human experts, there has been few studies examining human genre-labeling behavior.

An interesting experiment conducted by Chase [11] investigates the discrimination of musical stimuli by fish (Cyprinus carpio). In the study, a fish learns to discriminate blues recordings from classical ones with a low error rate. In addition, Crump [18] reports that pigeons demonstrate the ability to discriminate between Bach and Stravinksy. These results suggest that music genres may depend not only on cultural extrinsic habits but also on intrinsic properties of the music content.

Gjerdingen and Perrott [29] find a close immediacy of genre identification as a response to a musical stimulus. A group of participants are asked to evaluate brief excerpts of commercially recorded music and assign to them one of ten genre labels. The participants only need approximately 300 milliseconds of audio to accurately predict a music genre.

Lippens *et al.* [42] conduct a human-labeling experiment using a collection of 160 pieces of music. A group of participants are asked to evaluate and assign one of six genre labels to a music piece. The experiment is designed to compare the performance between automatic and human genre classification. Their results show that there is a certain degree of subjectivity in genre annotations by humans and, consequently, automatic classification [31]. The results from this study are re-analyzed rigorously by Craft *et al.* [17] to include music pieces that are annotated as "other" or with multiple genres.

Additional studies in [33, 49, 50] deal with human-involved evaluations of music genres. It is generally concluded that intrinsic properties of music exist that can be used for genre classification. However, some music pieces from different genres do share similar characteristics, such as *timbre*, *tempo*, etc., which can result in poor genre classification. It is therefore important to emphasize that a degree of inconsistency exists in music classification [3].

2.2 Content-based Audio Analysis

In content-based audio analysis, a music piece is described by a set of features (to be discussed in the following) that are directly computed from its content, i.e., the audio signal is parameterized into suitable feature vectors, which should retain salient information while discarding unnecessary details [57]. More specifically, samples from an audio signal cannot be used directly for analysis due to the large volume of information in the signal. Therefore, the initial step in content-based audio analysis is to represent the audio signal in a low-dimension form which can be used to manipulate more meaningful information [62].

The basic process for representing an audio signal can be defined by four steps: (1) Data preprocessing, (2) Segmentation, (3) Feature extraction, and (4) Data aggregation, as discussed in Chapter 1. In the following we review each step in detail.

2.2.1 Preprocessing

In order to extract features from an audio signal, samples are often preprocessed using standard *digital signal processing* techniques. The purpose of this is to help improve performance when features are processed by classification algorithms.

One common technique is to *normalize* the amplitudes in an audio signal such that they are uniformly scaled for further manipulation [47]. This is achieved by uniformly increasing or deceasing the amplitude of an audio signal such that the resulting peak amplitude matches a desired target. Normalization is helpful when applying classification algorithms as it controls variability in recording levels. However, this might not be desired in cases where variability can be a useful indicator in an application under consideration.

Another common technique is *downsampling*, which reduces the sampling rate of an audio signal. It is useful for adjusting different audio sources to the same bandwidth. It is also useful to reduce data size, which helps speed up the feature extraction process and the related classification algorithms [47]. Note that in the downsampling process, the sampling quality of some audio source is reduced for the sake of representing multiple audio sources. Other preprocessing techniques, such as *rectification* [47] and *channel merging* [47, 81], can also be used to further reduce the data size.

2.2.2 Feature Extraction

The feature extraction step transforms the input audio signal into a low-dimensional representation which contains the information necessary for classification or content analysis.

Feature extraction methods draw inspiration from a variety of sources, including signal processing, physics of sound, psychoacoustics, speech perception, and music theory [8]. They use information such as *spectral* or *statistical* variations in order to determine *rhythm*, *pitch*, *tempo*, *melody*, and *timbre.* [60]

As discussed by Gouyon *et al.* [52], "a key assumption [in feature extraction] is that the signal can be regarded as being stationary over intervals of a few milliseconds". Therefore, an audio signal is typically divided into small, possibly overlapping, *frames* (also referred to as *analysis windows* [75]). This approach, commonly referred to as the bag-of-frames approach, models the audio signal as a statistical distribution of audio features computed from individual, short frames [55]. These frames can vary in size. For instance, Tzanetakis and Cook [75] use a frame size of 23-milliseconds while Jiang *et al.* [35] partition an audio signal into frames of 200-milliseconds. To minimize the discontinuities at the beginning and end of a frame, a *window function* (e.g. *Gaussian* or *Hanning window*) is applied. As an example, analysis of the spectral content of each frame can be performed and a vector of features calculated [74]. These features are a summary of the corresponding spectral characteristics of a signal.

To capture the long term nature of sound "texture", i.e., the *melodic*, *rhythmic*, and *harmonic* composition, features are typically computed as the running means and variance over a set of consecutive frames, which is collectively referred to as a *texture window* [75]. In practice, instead of using the feature values directly from the frames, the parameters of a running multi-dimensional Gaussian distribution are estimated for a texture window [75].

In our work we make use of low-level features for music classification. These are simple mathematical transformations designed to capture perceptually significant aspects of the sound. However, one would ideally like to be able to extract high-level features such as chords, rhythmic patters, and pitches. Despite this, it is not currently known as how to reliably extract such high-level features from audio signals [47]. For this reason, a comprehensive survey of high-level features and their extraction techniques is beyond the scope of our discussion. For more information regarding these features, the interested reader is referred to [1, 75]. In the following, a selection of acoustic features for audio classification and music content analysis are presented.

Temporal Features

Features can be calculated directly from the temporal representation of the audio signal, i.e., the sequence of samples. It has been shown that some of these low-level features correlate with perceptual qualities. For instance, *amplitude* is correlated with *loudness* while *frequency* is related to *pitch* [52]. The following is a list of some low-level features that pertain to our work, where $x[n]$ refers to the value of the signal x at sample n, for a frame consisting of N samples.

Zero Crossings Rate (ZCR) is defined as the rate of sign change along an audio signal, i.e., the number of times the signal changes its sign in a frame. This feature provides an indication of signal noisiness [75]. The zero crossing rate is computed as

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

where the *sign* function is 1 for a positive argument and 0 for a negative argument.

Root Mean Square (RMS): A measure of the average energy, or loudness, of an audio signal calculated over a frame. It is calculated by taking the mean of the square of all sample values in a frame and then taking the square root. RMS gives a good indication of loudness and may also serve for high-level tasks such as tempo/beat estimation [16].

$$RMS = \sqrt{\frac{1}{N}\sum_{n=1}^{N} x[n]^2}$$

Fraction of Low Energy Frames: The percentage of a set of consecutive frames that have a RMS below some threshold. This feature measures the amplitude distribution of an audio signal [47].

Relative Difference Function: A measure of relative change in an audio signal. This feature can be useful in detecting significant changes, such as *note onsets* [47].

For more information regarding temporal features, the reader is referred to [1, 75, 87].

Spectral Features

It is common for a feature to be calculated on a different representation, e.g., the spectral domain of an audio signal. A spectrum is obtained from each frame by applying the *Discrete Fourier Transform* (*DFT*) to it. These features, also known as *Spectral Distribution Descriptors*, describe the shape of the spectral representation for a given frame and are considered to be indicative of perceptual characteristics of the audio signal [62, 81]. The following is a list of spectral features used in our work, where $X_t(n)$ is the n^{th} frequency sample of frame t and N is the index of the highest frequency sample. More detailed discussions on other spectral features are found in [47, 75, 87].

Spectral Centroid (SC): An indicator of the balancing point of a spectrum [47, 75]. It is considered to correlate with the perceptual qualities *brightness* or *sharpness*. The spectral centroid of frame t is calculated as

$$SC_t = \frac{\sum_{n=1}^{N} |X_t(n)| \cdot n}{\sum_{n=1}^{N} |X_t(n)|}.$$

Spectral Flux (SF): A measure representing the change in shape of the magnitude spectrum by calculating the difference between magnitude spectra of successive frames [47, 75]. It is calculated as

$$SF_t = \sum_{n=1}^{N} |(X_t(n) - X_{t-1}(n))|.$$

Spectral Rolloff (SR): An indicator of frequency below which a certain amount, represented as a percentage, of spectral energy resides [47, 75]. It measures the “skewness” of the spectral shape for frame t. The spectral rolloff is defined as SR_t such that

$$\sum_{n=1}^{SR_t} X_t(n) = C \cdot \sum_{n=1}^{N} X_t(n),$$

where C is a threshold between 0 and 1.

Cepstral Features

Mel-frequency cepstral coefficients (*MFCC*) are a popular feature extraction method in audio analysis. MFCCs are designed to capture short-term spectral-based features and are perceptually motivated based on the DFT [75]. They are calculated by taking the log-amplitude of the magnitude spectrum and then grouping and smoothing the spectrum bins based on the Mel-frequency scale [1, 75]. The Mel-frequency scale is intended to simulate the distribution of the human ears' critical bandwidth. It is based on the mapping between actual frequency and perceived pitch. Since the human auditory system does not perceive pitch in a linear manner, the mapping is approximately linear below 1kHz and logarithmic above [43, 81]. Typically, there are 13 coefficients extracted for audio representation.

Chroma Features

While MFCCs are a popular feature extraction method for audio analysis, they mainly reflect the instruments and arrangement of the music rather than the melodic or harmonic content [22]. *Chroma features* attempt to represent the harmonic content (e.g., keys and chords) of a short-time window of audio while minimizing the influence of instrumentation [30]. This is achieved by computing the energy present at frequencies that correspond to each of the 12 notes in a standard chromatic scale (e.g., the black and white keys within one octave on a piano) [46]. That is, the energy is collapsed into a 12-bin octave-independent histogram representing the relative intensity of each of the 12 semitones of an equal-tempered chromatic scale. There are several advantages to using the chroma feature including the ability to effectively identify chord names and detect modulating repetition [30].

2.2.3 Segmentation

The features extracted from each frame of an audio signal represent a significant reduction in terms of data size and dimensionality. However, this yields a large number of feature vectors for each audio signal. To further reduce an audio signal to a lower dimensional form, instead of extracting features on a frame-by-frame basis, features can be calculated on a group of consecutive frames. To this end, we need to determine the grouping of consecutive frames. This process, known as *segmentation*, is accomplished by applying acoustic analysis to the frames from an audio signal and looking for "significant" transitions. For instance, Bello and Pickens [7] describe a beat-synchronous segmentation method in which an audio signal is divided into frames whose sizes are equal to the beat period, and whose starts and ends are synchronized with beats.

Tzanetakis and Cook [73] develop an automatic segmentation algorithm based on the detection of temporal change. The method uses the amount of change of a feature vector as a boundary detector. That is, when the amount of change in a feature vector is greater than a given threshold, a boundary decision is made [52].

A slightly different approach is to develop a similarity matrix between frames and their neighbors [25]. This method embeds the parameterized audio into a 2-dimensional matrix and then calculates a measure of (dis)similarity between the feature vectors of two audio frames. A *kernel function* is used to find a measure of similarity and cross-similarity within the matrix. The difference between these two values estimates the *novelty* of an audio signal. The data points corresponding to the extremes of these novelty scores are selected as segmentation locations [6, 25].

West and Cox [82] introduce an event-level segmentation method based on an *onset detection function*. As described, "event-level segmentation of the audio stream aims to produce more informative features than those produced for individual audio frames, whilst generating significantly less data". Moreover, they perform an evaluation of segmentation techniques, concluding that event-level segmentation methods based on an onset detection function outperform fixed-length techniques.

2.2.4 Aggregating Features

In order to efficiently perform content-based audio analysis, inducing genre classification, feature vectors must be aggregated to produce a smaller number of more informative vectors. A common approach is to summarize the distribution of feature vectors over the whole audio signal.

Previous works have made use of estimating a Gaussian distribution of feature values. For example, Lambrou *et al.* [37], Tzanetakis and Cook [75], and Li *et al.* [40] use a Gaussian distribution with diagonal covariance. More recently, some have demonstrated approaches based on Gaussian mixtures, which are fitted to the distribution of features [45]. The advantage of this approach is that it can provide a better fit to the empirical distribution. However, these distributions are over the frame-level *feature space*, which can have many dimensions. Furthermore, a single distribution can be efficiently represented as a single vector and can be used for distribution-based comparisons, vector distance measures, or classification algorithms [81].

2.3 Single-Label Classification Algorithms

Machine learning has developed a multitude of methods for deriving statistical classifiers from example instances. These classifiers are first constructed via learning algorithms based on a set of training instances. A prediction can then be made for an unseen test instance regarding what class it most likely belongs to. In the traditional task of single-label classification each instance is associated with a single class label and a classifier learns to associate each new test instance with one of these known class labels. In the context of this section, we define an instance to be a music piece characterized by a list of audio features and labeled with an appropriate music genre. In the following, we introduce an overview of classification algorithms that are widely used in music retrieval.

2.3.1 Gaussian Mixture Models

Gaussian Mixture Models (GMM) are used for estimating the distribution of features. The basis for using GMM is that the distribution of feature vectors extracted from a class can be modeled by a mixture of Gaussian densities. For each class we assume the existence of a probability density function expressed as the weighted sum of simpler Gaussian densities, called components of the mixture. An iterative algorithm is typically used to estimate the parameters for each Gaussian component and the mixture weights [41]. Classification can be performed by modeling each class as a GMM; an instance is then classified by calculating, for each class (GMM), the probability that the instance is produced by the respective GMM and predicting the class with the maximum likelihood [52].

2.3.2 k-Nearest Neighbor

The *k-Nearest Neighbors* (*k-NN*) classifier is a non-parametric, lazy classifier which has been commonly used for various tasks in MIR, including genre classification [41, 52]. Lazy classifiers do not form a model of the data prior to classification. Instead, the entire classification procedure is executed ad-hoc. Training is performed by storing the features of each training instance. A test instance can then be classified by calculating the distance between its features and those of the training instances. The class label of the testing instance can be determined by the class that occurs most frequently among the k nearest neighbors' [47].

There are a variety of distance metrics that can be used to calculate the distance between features of two instances. For example, *Euclidean distance* is the most commonly used metric. However, it is sometimes more appropriate to use alternatives, such as *Manhattan* *distance*, *Mahalanobis distance*, or *Tanimoto distance* [47].

2.3.3 Support Vector Machine

Support Vector Machines (*SVM*) are an efficient training algorithm that can represent complex, non-linear functions. They have been successfully applied in pattern recognition and are commonly used in genre classification [45, 50, 85].

In essence, SVMs search for a hyperplane that separates the positive data points and the negative data points with maximum margin. If the data are not linearly separable in the feature space, they can be projected into a higher dimensional space by means of a *kernel function*, which maps a lower-dimensional space to a higher-dimensional one where a hyperplane can be used to effectively discriminate between data points. The most popular kernel functions are polynomials of various degrees, radial-basis functions, and sigmoid functions.

2.3.4 Decision Trees

Decisions Trees are one of the most popular classification methods in statistical pattern classification. In general, a decision tree corresponds to a set of classification rules that predict the class of an instance based on specific characteristics of its features. Classification is performed using a divide-and-conquer strategy where complex decision functions are broken down into a series of simple decisions.

A decision tree for a classification task produces a hierarchy of if-then rules, which can be examined in order to gain insight into the classification process. These rules are a natural way of thinking, making decision tree classification much more human-meaningful than other classification algorithms.

2.4 Single-Label Classification on Music

In this section we briefly review related works in the area of automatic music classification and focus on single-label approaches.

Berstra *et al.* [8] explore the effects of segment size on classification accuracy. They use a variety of frame sizes and analyze the results for evidence that certain segment sizes work better than others. A supervised learning algorithm for classifying music, based on an ensemble classifier *ADABOOST*, is presented. The input music is initially partitioned into short frames. Features, such as MFCCs, ZCRs, Spectral Centroid, Spectral Rolloff, etc., are then extracted from each frame. After this, non-overlapping blocks of consecutive frames are grouped into segments. Each segment is summarized by fitting independent Gaussians to the features. The resulting means and variance are then input to the classifier. An "optimal" segment size is experimentally determined and the evaluation of the classification accuracy is performed using a data set consisting of 1000 30-second music pieces labeled with one of the ten genres.

Tzanetakis *et al.* [76] describe a Gaussian classifier for the automatic genre classification of music and propose a set of features for representing texture, instrumentation and rhythmic structure. To evaluate the performance on the proposed feature set, the classifier is trained and evaluated using audio datasets collected from radio broadcasts, compact disks and the Internet. The mean and standard deviation of the features are calculated over a texture window, consisting of 40 analysis windows. The dataset consists of 50 audio clips, each 30-seconds long. The classification accuracy is better for the proposed feature set than any randomly generated ones.

Pampalk *et al.* [56] demonstrate that the performance of genre classification can be improved by combining spectral similarity with complementary information, in particular, fluctuation patterns. This is evaluated using a nearest neighbor classifier and four music collections with a total of approximately 6000 pieces. Improvements in classification accuracy are reported for only one of the four collections. It is suggested that this confirms the claims by Aucouturier and Pachet [2] who infer the existence of a “glass ceiling” which cannot be surpassed without taking cognitive processing at a higher level into account. In addition, using an artist filter is recommended to ensure that artists in the test set are not present in the training set; otherwise, genre classification might transform into artist identification.

Xu *et al.* [85] present and demonstrate an automatic classification approach for musical genres using a multi-layer support vector machine. They consider features that are related to temporal, spectral and cepstral domains. The selected features include ZCR, MFCCs, etc. A dataset consisting of 100 music samples collected from compact disks and the Internet are used for experiments and divided into four major genres: classical, jazz, pop and rock. Different features and support vectors are employed for separate layers in the classifier. In the first layer, a music piece is classified into either pop/classical or rock/jazz. In the second layer, further classification is performed between pop and classical or between rock and jazz. Experiment results illustrate that the multi-layer classifier has good classification performance and is more advantageous than Euclidean distance-based methods, such as k-NN, and other statistical learning methods.

For more discussions on genre classification, the interested reader is referred to [62].

2.5 Multi-Label Classification Algorithms

Different from traditional single-label classification where each object belongs to only one class, multi-label classification deals with problems where an object may belong to more than one class. That is, multi-label classification algorithms assign one or multiple classes to an instance simultaneously. These algorithms can be grouped into two categories as proposed in [67]: (1) *problem transformation methods*, and (2) *algorithm adaptation methods*. Problem transformation methods transform a multi-label classification problem into one or more single-label classification problems while algorithm adaptation methods extend traditional single-label classifiers to handle multiple labels directly. Below, we review a set of multi-label classification algorithms.

2.5.1 Label Powerset

Multi-label classification can be decomposed to a conventional classification problem by considering each unique set of class labels as one single class [69]. This approach is referred to as *label powerset* and is a simple but effective problem transformation method. It has the advantage of taking label correlations into account. Given a new instance, a selected single-label classifier outputs the most probable class, which is a set of labels. However, it is challenged by domains with a large number of labels and training instances [70].

2.5.2 Binary Relevance

Binary relevance is a popular problem transformation approach where a separate binary classifier is trained for each label, i.e., it trains $|L|$ binary classifiers C_1, $\cdots$, $C_{|L|}$, where $L = \{l_1, l_2, \cdots, l_N\}$ is the finite set of labels in a multi-label classification task. Each classifier C_j is responsible for predicting the presence or absence of each corresponding label $l_j \in L$. When classifying a new instance, this approach outputs the union of the labels that are predicted by the $|L|$ binary classifiers [64, 69]. Binary relevance is simple to implement and has been used as a baseline throughout the multi-label literature. Although it is has a linear complexity with respect to the number of labels, an obvious disadvantage of this approach is that it ignores correlations and interdependencies between labels. [27].

2.5.3 Multi-label k-NN

A number of multi-label classification approaches have been based on the traditional k-NN (discussed in Section 2.3.2) algorithm. For example, *Multi-label k-NN* (ML-kNN, for short) [86] is an instance-based adaptation of the kNN algorithm for multi-label data. The algorithm identifies, for each unseen test instance, the k nearest neighbors in the training set. The maximum *a posteriori* probability principle is utilized to determine the label set of the test instance, based on prior and posterior probabilities for the frequency of each label from the k nearest neighbors [69]. This method is shown to perform well in practice and outperforms some state-of-the-art classification approaches [64, 86].

2.5.4 Instance Based Logistic Regression

Cheng and Hüllermeier [14] present another approach to instance-based classification that can be used for both general and multi-label cases. It considers the labels of neighboring instances as features of an unseen instance and reduces instance-based learning to logistic regression. The approach allows one to take into consideration interdependencies among different labels for multi-label classification. The approach is shown to improve upon some existing multi-label classification approaches, in particular, ML-kNN, which is considered state-of-the-art in multi-label classification.

2.5.5 Random k-Labelsets

To deal with the problems inherent to Label Powerset, a variety of algorithms have been proposed. One approach is Random k-Labelsets [70, 72]. This approach constructs an ensemble of label powerset classifiers, each of which is trained using a different small random subset of labels. This approach has the advantage of taking label correlations into account while avoiding the problems associated with label powerset when a large number of labels and training instances are used [69]. However, to get “near-optimal” performance, appropriate parameters in the approach must be optimized. When the number of training instances is insufficient, this can be difficult.

2.5.6 Calibrated Label Ranking

Fürnkranz *et al.* [27] propose an efficient pairwise method for multi-label classification referred to as *Calibrated Label Ranking*. This approach performs classification by introducing an artificial calibration label that, in each instance, separates the relevant labels from the irrelevant ones. An additional L binary classifiers approximating the standard binary relevance problem are required to calibrate this label at training time. The key idea is to add an additional label to the original label set which is interpreted as a "neutral element". This approach has been shown to perform well against other multi-label classification approaches and has become a heavily cited work. It is important to note that calibrated label ranking is an extension of the *ranking by pairwise comparison* scheme (RPC), which obtains a ranking by counting the votes received by each label [34].

2.5.7 Hierarchy of Multi-label Classifiers

High dimensionality of label space may pose challenges for a multi-label classification algorithm. For example, the computational cost of training a multi-label classifier may be strongly affected by the number of labels. Tsoumakas *et al.* [68] describe an algorithm, referred to as *Hierarchy of Multi-label Classifiers* (HOMER), for effective and computationally efficient multi-label classification in domains with many labels. The approach constructs a hierarchy of multi-label classifiers, each dealing with a much smaller set of labels. Its efficiency is due to splitting up the label set using a modified k-means clustering algorithm and handling each subset individually [59]. It is shown that this approach provides more accurate predictions than Binary Relevance in less time [68].

2.6 Multi-label Evaluation

In single-label classification, predictive performance can be calculated by the traditional accuracy measure, where each test instance is either correct or incorrect, and performance is given by the number of correctly classified test instances relative to the total number of test instances. However, multi-label classification requires different evaluation measures. In the following, we review a set of evaluation measures proposed in literature that will be used in our work. For consistency, we use the same notation as used in [86].

Let $\mathcal{X}$ denote the domain of instances (D) and let $\mathcal{Y} = \{l_1, l_2, \cdots, l_N\}$ be the finite set of labels (L). We assume that a multi-label classifier induces an ordering of the possible labels for a given instance. That is, the output of the classifier is a real-valued function of the form $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$. It is supposed that, given an instance $x_i \in \mathcal{X}$ and its associated label set $Y_i \subseteq \mathcal{Y}$, a successful classifier will output larger values for labels in Y_i than those not in Y_i. That is, $f(x_i, y_1) > f(x_i, y_2)$ for any $y_1 \in Y_i$ and $y_2 \notin Y_i$.

The function $f(\cdot, \cdot)$ can be transformed to a ranking function $rank_f(\cdot, \cdot)$ which maps the output of $f(x_i, y)$ for any $y \in \mathcal{Y}$ to $\{l_1, l_2, \cdots, l_N\}$ such that if $f(x_i, y_1) > f(x_i, y_2)$ then $rank_f(x_i, y_1) < rank_f(x_i, y_2)$. A corresponding multi-label classifier $h(\cdot)$ can be derived from the function $f(\cdot, \cdot)$ by setting $h(x_i) = \{y | f(x_i, y) > t(x_i), y \in \mathcal{Y}\}$, where $t(\cdot)$ is a threshold function.

2.6.1 Example-based

Several measures have been proposed that are calculated based on the average difference of the actual and the predicted set of labels over all test examples while others decompose the evaluation process into separate evaluations for each label, which they subsequently average over all labels. Tsoumakas *et al.* [69] refer to the former as *example-based* and the latter as *label-based* evaluation measures. Below we present a set of example-based measures including *Accuracy*, *Recall*, *Precision*, F_1, and *Hamming Loss.*

Accuracy, Recall, Precision, F-measure

The following measures are related to the predicted and original labels for each instance in the testing set. That is, they are discussed on a per instance basis and the aggregate value is an average over all instances. The bigger the value of each measure, the better the performance.

$$Accuracy(h) = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i \cap h(x_i)|}{|Y_i \cup h(x_i)|} \qquad Recall(h) = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i \cap h(x_i)|}{|Y_i|}$$

$$Precision(h) = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i \cap h(x_i)|}{|h(x_i)|} \qquad F - measure(h) = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{2|Y_i \cap h(x_i)|}{|h(x_i)| + |Y_i|}$$

Hamming Loss

Hamming Loss computes the percentage of labels that are misclassified, i.e. a label not belonging to the instance is predicted or a label belonging to the instance is not predicted [63, 86]. The smaller the value of $HammingLoss_s(h)$, the better the performance. Hamming Loss is calculated as

$$HammingLoss_s(h) = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{1}{|L|}|h(x_i)\Delta Y_i|,$$

where the Δ operator stands for the symmetric difference between two sets and corresponds to the XOR operation in Boolean logic [67].

2.6.2 Label-based

Any known measure for binary evaluation can be used here, such as accuracy, precision, and recall. The calculation of theses measures for all labels can be achieved using two averaging operations, called *macro-averaging* and *micro-averaging* [69]. These operations are typically considered for averaging precision, recall, and their harmonic mean (F-measure) in information retrieval tasks.

Consider a binary evaluation measure $M(tp, tn, fp, fn)$ that is calculated based on the number of true positives (tp), true negatives (tn), false positives (fp), and false negatives (fn). Let tp_λ, fp_λ, tn_λ, and fn_λ be the number of true positives, false positives, true negatives, and false negatives after binary evaluation for a label λ. The macro-averaged and micro-averaged versions of M are calculated as:

$$M_{macro} = \frac{1}{|L|}\sum_{\lambda=1}^{|L|} M(tp_\lambda, fp_\lambda, tn_\lambda, fn_\lambda),$$

$$M_{micro} = M\left(\sum_{\lambda=1}^{|L|} tp_\lambda, \sum_{\lambda=1}^{|L|} fp_\lambda, \sum_{\lambda=1}^{|L|} tn_\lambda, \sum_{\lambda=1}^{|L|} fn_\lambda\right).$$

Note that micro-averaging has the same result as macro-averaging for some measures, such as accuracy, while it differs for other measure, such as precision and recall. While the previous evaluation measures are based on the multi-label classifier $h(\cdot)$, the remaining measures are defined based on the real-valued function $f(\cdot, \cdot)$ which concerns the ranking quality of different labels for each instance.

2.6.3 Ranking

Average Precision

Average Precision evaluates the average fraction of labels ranked above a particular label $y \in Y$ which are actually in Y. This measure is frequently used for the evaluation of information retrieval tasks [63]. The bigger the value of $AvgPrecision_s(f)$, the better the classification performance.

$$AvgPrecision(f) = \frac{1}{|D|}\sum_{i=1}^{|D|} \frac{1}{|Y_i|}\sum_{y \in Y_i} \frac{|\{y' | rank_f(x_i, y') \leq rank_f(x_i, y), y' \in Y_i\}|}{rank_f(x_i, y)}$$

One-Error

One-Error calculates the number of times the top-ranked label is not in the label set of an instance [63, 86]. That is, it determines whether the top-ranked label is relevant and ignores the relevancy of all other labels [27]. The smaller the value of $OneError_s(f)$, the better the performance. One-Error is calculated as

$$OneError(f) = \frac{1}{|D|}\sum_{i=1}^{|D|}[\![[\arg\max_{y\in Y} f(x_i,y)] \notin Y_i]\!],$$

where for any predicate π, $[\![\pi]\!]$ equals 1 if π holds and 0 otherwise [86].

Coverage

Coverage measures how far we need, on average, to go down the list of labels in order to cover all the possible labels assigned to an instance. The goal of coverage is to assess the performance of a classifier for all the possible labels of instances [63, 86]. The smaller the value of $Coverage_s(f)$, the better the performance.

$$Coverage(f) = \frac{1}{|D|}\sum_{i=1}^{|D|}(\max_{y\in Y_i} rank_f(x_i,y) - 1)$$

Ranking loss

Ranking loss computes the average fraction of label pairs which are not correctly ordered, i.e., label pairs that are reversely ordered for an instance [86, 27]. The smaller the value of $RankingLoss_s(f)$, the better the performance. Ranking Loss is calculated as

$$RankingLoss(f) = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{1}{|Y_i||\overline{Y_i}|}|\{(y_1,y_2)|f(x_i,y_1) \le f(x_i,y_2), (y_1,y_2) \in Y_i \times \overline{Y_i}\}|,$$

where $\overline{Y}$ denotes the complementary set of Y in $\mathcal{Y}$.

2.7 Multi-Label Classification on Music

As of recently, multi-label classification methods have been used to categorize music into emotions. The classes representing emotions can be labeled in different ways. For instance, Dellaert *et al.* [20] classify emotions (in speech) into 4 classes: happy, sad, anger, and fear. In addition, Li and Ogihara [39] use 13 classes including labels such as frustrated, passionate, dark, depressing, dramatic, and etc.

Wieczorkowska *et al.* [84] address the problem of multi-label classification of emotions using a modified k-NN algorithm. A set of 875 audio samples, 30 seconds each, manually labeled into 13 classes of emotion are used for their experiment. The modified k-NN algorithm aims at taking multiple labels into account. Therefore, the algorithm returns a set of labels for each neighbor of a test instance. In addition, Trohidis *et al.* [66] evaluate four multi-label classification algorithms for the purpose of detecting emotions in music. They classify emotions into 6 main clusters: amazed-surprised, happy-pleased, relaxing-calm, quiet-still, sad-lonely, and angry-fearful. Experiments are conducted on a set of 593 samples, from which a 30-second excerpt is extracted and annotated with a set of emotions. It is claimed that the overall predictive performance is high and encourages further investigation of multi-labeling methods.

Multi-label classification methods have also been used for the purpose of automatic tag annotation of music. These tags can be any semantically meaningful words and can represent a variety of different concepts including genre, instrumentation, geographic origins, social conditions, and emotions. Automatic tag annotation of music learns a relationship between acoustic features and words from a dataset of labeled audio tracks [51]. The resulting trained model can retrieve audio samples based on lists of tags and annotate unlabeled ones. For example, Wang *et al.* [79] study the problem of combining user-generated tags and music content for artistic style classification. They investigate the effectiveness of using tags and audio content separately for clustering, and propose a novel language model that makes use of both of them. Results show that tag features are more effective than music content for style clustering, and the proposed model can marginally improve clustering performance by combining tags and music content. In addition, Wang *et al.* [80] propose a multi-label music style classification approach, called *Hypergraph integrated Support Vector Machine*, which can integrate both music content and music tags for automatic music style classification. The effectiveness of the approach is demonstrated using a set of experiments on real world data.

Pachet and Roy [55] address the problem of improving multi-label analysis of music using a correction scheme. In this scheme, an extra layer of classifiers is built to exploit redundancies between labels and correct some of the errors coming from the individual classifiers. Statistical redundancies in the training dataset are used to correct individual classifier errors. A series of experiments are conducted to validate the approach using a large-scale database of music and metadata. Each music piece is annotated with a set of labels from 16 categories representing a specific dimension of music description. Categories include style, genre, dynamics, tempo, metric, mood, character, etc. The experiments show that the approach brings statistically significant improvements and is worth considering for multi-label classification of music.

2.8 Summary

Music genre classification is a high-level task in Music Information Retrieval. It has wide applications in the management of music repositories, including music categorization, organization, and browsing. However, to the best of our knowledge, there has been limited work regarding multi-label genre classification on music. In the following chapters, we study this problem from perceptual and algorithmic perspectives.

Chapter 3

Perception Based Multi-Genre Labeling

In this chapter we explore the multi-genre-labeling behavior of individuals by conducting a series of perceptual experiments[1]. Given a set of excerpts from a music piece, individuals are asked to classify each excerpt and assign to it a single genre class. The purpose of this study is to investigate inter-song similarity and dissimilarity, with regard to music genre, i.e., how (dis)similar is a song to itself at different time intervals. This stands in contrast to other human genre-labeling studies which compare the performance of each individual to some ground-truth. This distinction is important as the evaluation of any genre-labeling process is intrinsically difficult. Recall that the ground-truth against which any genre-labeling is compared is subjective and ambiguous [17].

This chapter is organized as follows. Section 3.1 explains the preparation of our experiments, including the description of the participants, the music used, the experiment environment, etc. In Sections 3.2 and 3.3 we analyze and discuss the results of our experiment. Section 3.4 summarizes the chapter along with some remarks.

[1] A preliminary version of this chapter is reported in [61].

3.1 Experiment Setup

Previous human classification experiments explore the genre-labeling behavior of individuals by presenting them with a short excerpt of audio and asking them to categorize it into one of predefined genre class labels.

In this section we describe the setup for our human classification experiments where participants are asked to classify multiple audio excerpts, from different sections of an audio piece, according to their perceived genre. We desire to gain insight into how participants label a full-length piece by breaking it into a series of smaller sections with a finer temporal resolution.

From this study, we expect to find a large diversity in the responses from our individuals as the relative knowledge, social, and cultural background of them will give rise to different answers. As described by Craft *et al.* [17], “any unity of response is [due to] the widespread agreed nature of the musical cues to genre of particular pieces, but the expected response from a group of individuals will be a diversity”.

3.1.1 Dataset

The Magnatune dataset, licensed as creative commons [56], is a free collection of full-length polyphonic tracks which are unequally distributed over eight musical genres, including *Classical*, *Jazz*, *World*, *Electronic*, *Metal*, *Pop*, *Rock*, and *Punk*. Each track in this dataset is annotated with one of eight musical genres according to the Magnatune website[2].

For our experiment we use a variant of the Magnatune dataset [21], denoted as D_v, which consists of approximately 700 tracks. The eight genres, presented in Table 3.1, are used to represent the diversity of musical styles in the dataset. This collection was used as a training set for the *International Society for Music Information Retrieval* genre classification contest in 2004 [16, 21] and has been used for a variety of experiments in content-based music analysis.

[2] http://www.magnatune.com.

Genre	**Number of Tracks**
Classical	320
World	122
Electronic	115
Rock	95
Metal	29
Jazz	26
Punk	16
Pop	6

Table 3.1: Distribution of songs in D_v.

Although genres can lose their prominence or change their characteristics over time [48], which can necessitate re-annotation of ground-truth [29], this collection is considered representative of mainstream music genres at the time of the study. The distribution of songs in D_v is shown in Table 3.1.

3.1.2 Stimuli

A total of 23 audio tracks, defined as D_t ($\subset D_v$), are selected for our study. For each of the eight genres included in D_v, we attempt to select three or four tracks that we subjectively feel have the characteristic of the respective genre. Due to the unbalanced distributions of songs in D_v [16, 56], some genres are represented using only two tracks in D_t. As discussed by Gjerdingen and Perrott [29], " when an artist is strongly associated with one genre, that association may trump musical distinctions that would otherwise indicate a different genre". For this reason our selection criteria is also based on choosing tracks and artists that are not common in mainstream music at the time of this study.

From each track in D_t, denoted as T, we manually extract a set of 30-second audio clips, C_T, based on what we subjectively feel are drastic changes in terms of the *sound texture* and *global timbre*. We choose a clip length of 30-seconds as it is common in MIR. For example, Barrington *et al.* [4] adopt a clip length of 30-seconds when performing human evaluation of music recommendation systems, while Lippens *et al.* [42] also use the same length clips to perform a comparison of human and automatic musical genre classification. Typically, 30 seconds is used as it is sufficiently short to make evaluation trials manageable.

For the purpose of this study we define $2 \leq |C_T| \leq 4$, i.e., we extract a minimum of two clips and a maximum of four from each track $T \in D_t$. In order to represent different sections and structures of T, we attempt to extract clips from the beginning, middle, and end of the track. A total of 63 30-second audio clips, denoted as D_c, are extracted from the tracks in D_t.

3.1.3 Participants

As discussed by Levitin [15], to compensate for large individual differences in human-involved experiments, it is desirable that a group of 30 to 100 individuals participate. Additionally, for an assigned labeling to be considered reliable, an individual must be cognizant of the stylistic characteristics of each genre in the dataset [17]. For these reasons, our goal is to include a large number of participants who are familiar with the eight genres in our dataset.

A total of 101 participants majoring in *psychology*, *music*, and *computer science* volunteered for this study. There are 29 males and 72 females, with a mean age of 21.1 years. 83 of them report listening to music on a frequent basis while 15 report occasionally and 3 rarely. The participants in this study come from a wide musical background: 49% have played an instrument (for a mean length of 9.13 years), 58% have taken formal music lessons, and 16% have taken music courses at the college or university level.

3.1.4 Experiment Procedure

To ensure accurate and consistent results, each participant is tested on an IBM-compatible PC using a custom audio player designed to minimize visual distractions. S/he is expected to listen to a playlist of 63 audio clips, which are randomly ordered with the criteria that any two clips from the same track are of maximum distance apart in the playlist. This is to ensure responses are not influenced by the order in which the stimuli is presented [15]. The participants are instructed to indicate which genre of music they consider best represent the clip by clicking one of eight genre buttons displayed on screen. After every five music clips, 30-seconds of silence is presented to prevent audio fatigue [15]. At any time during the testing individuals are given the opportunity to change their genre selection for a particular clip.

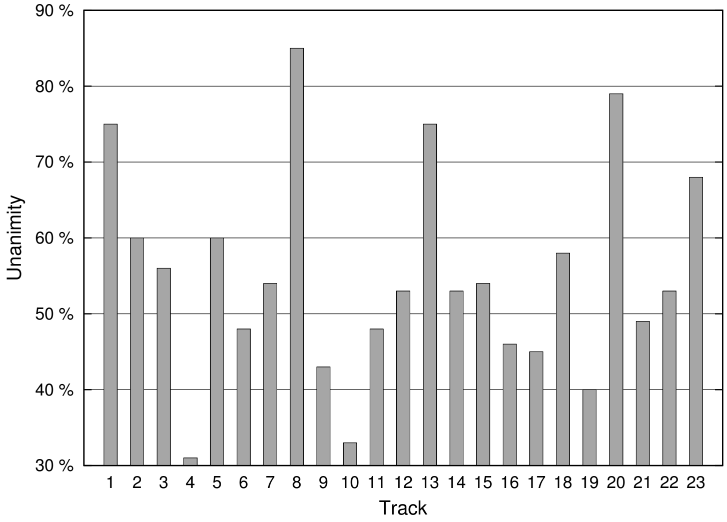

Figure 3.1: Unanimity of dataset D_t.

3.2 Experiment Results

In this section we report the results obtained in our human classification experiment. We follow an approach similar to other human genre-labeling studies and first report the classification accuracy.

For each individual we compare her/his selected genres with the labels defined by the Magnatune website. We observe that for approximately 65% of all the votes, the participants and Magnatune agree on the same genres. However, this does not imply that the other 35% are incorrect, since genre classification of music involves a degree of subjectivity [29]. We now examine the results from our study with regard to inter-song genre similarity and dissimilarity.

Using a notation similar to the one introduced by Lippens *et al.* [42], for each track $T \in D_t$, we denote $Q_{T,g}$ as the number of votes for a given genre g. To calculate $Q_{T,g}$, we sum up the total number of votes for genre g over all the individual clips from track T. We then define Q_T to be the total number of genre votes for a given track T. For each track T we consider the genre with the maximum number of votes to be the elected genre and set ${Q_T}^{max}$ to be this number. We also calculate Q_T^{max}/Q_T to represent the *unanimity*, or *consensus*, of the elected genre from track T. Figure 3.1 shows the unanimity of votes for each track in D_t.

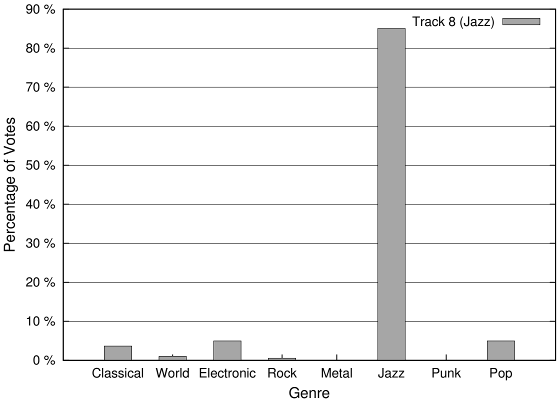

Figure 3.2: Distribution of votes for Track T_8 (High Unanimity).

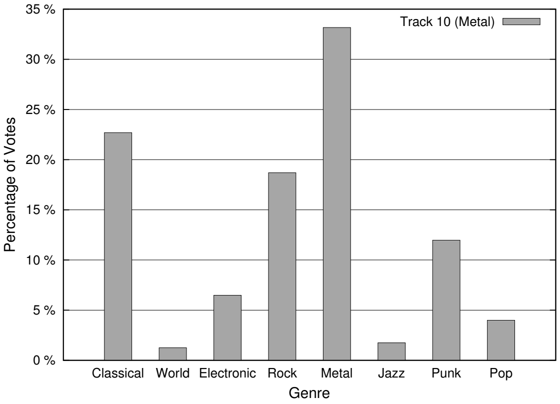

Figure 3.3: Distribution of votes for Track T_{10} (Low Unanimity).

From Figure 3.1 we observe that there is a degree of uncertainty involved in the classification of dataset D_t as the unanimity of votes range from 33% to 85%. However, this is not the case for some of the tracks. For example, Figure 3.2 shows the distribution of votes for track T_8. We observe that the votes are primarily centered around a small number of genres, e.g., Jazz.

When we analyze the distribution of votes for clips $D_c = \{1, 2, \cdots, 63\}$, as shown in Figure 3.4, and examine the clips for track T_8, where $C_{T_8} = \{19, 20, 21\}$, we find that the majority of them are voted for in a similar way. Figures 3.4 and 3.5 reveal the distribution

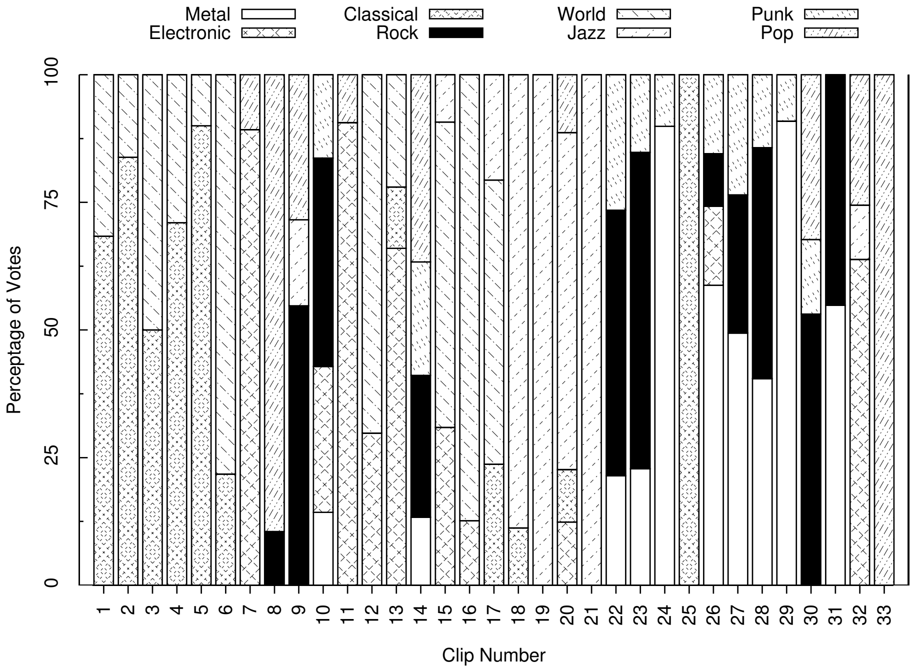

Figure 3.4: Distribution of votes for clips 1-33 in D_c.

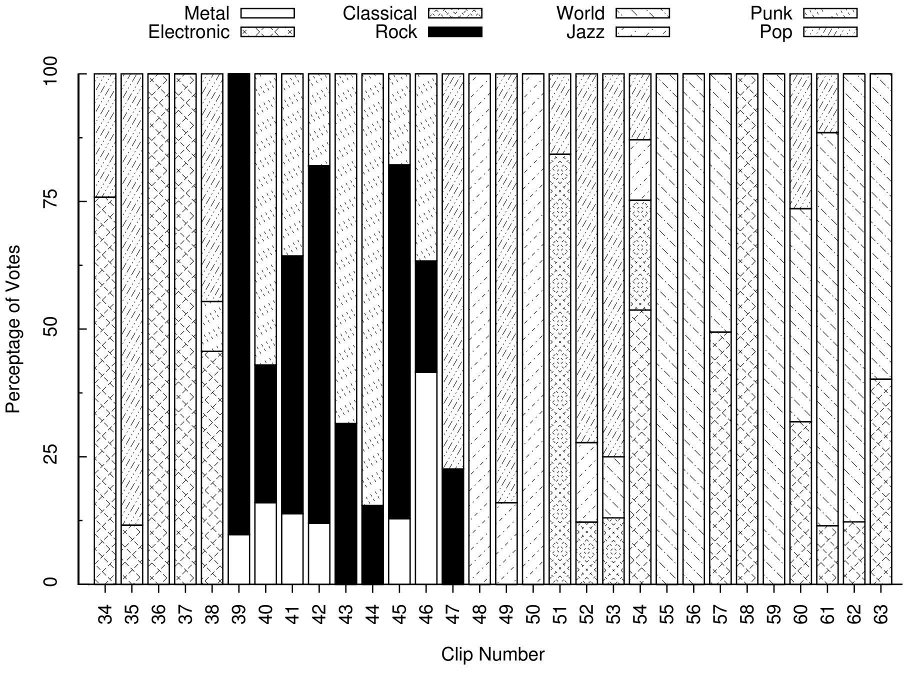

Figure 3.5: Distribution of votes for clips 34-63 in D_c.

Track	Clip Number(s)
1	1,2
2	3,4
3	5,6
4	7,8,9,10
5	11,12,13
6	14,15,16
7	17,18
8	19,20,21
9	22,23,24
10	25,26,27,28
11	29,30,31
12	32,33,34,35
13	36,37,38
14	39,40,41
15	42,43,44
16	45,46
17	47,48
18	49,50
19	51,52,53,54
20	55,56,57
21	58,59
22	60,61
23	62,63

Table 3.2: Set of clips extracted from each track in D_t.

of votes for each clip in D_c. Only those genres with a consensus of $\geq 10\%$ are reported in the figures for illustration purposes. Table 3.2 reports the set of clips, extracted form each track in D_t, used to construct Figures 3.4 and 3.5, for cross reference.

Although we find cases where individuals have a high degree of agreement on some genres, there are instances where they do not and the distribution of votes are spread over multiple genres. For instance, Figure 3.3 shows an example distribution of votes for a track with low unanimity, i.e., the votes are distributed among a number of different genres. This result supports our expectation of finding a large diversity in the responses of individuals.

Track	Elected Genre(s)
1 (Classical)	Classical
2 (Classical)	Classical, World
3 (Classical)	Classical, World
4 (Electronic)	Electronic, Pop, Rock
5 (Electronic)	Electronic, World
6 (Electronic)	Pop, World
7 (Jazz)	Jazz, World
8 (Jazz)	Jazz
9 (Metal)	Metal, Rock
10 (Metal)	Classical, Metal, Rock
11 (Metal)	Metal, Rock
12 (Pop)	Electronic, Pop
13 (Pop)	Electronic
14 (Punk)	Rock, Punk
15 (Punk)	Rock, Punk
16 (Punk)	Rock, Metal
17 (Rock)	Jazz, Pop
18 (Rock)	Jazz, Pop
19 (Rock)	Classical, Pop, Electronic
20 (World)	Electronic, World
21 (World)	Electronic, World
22 (World)	World
23 (World)	World

Table 3.3: Elected genres for each track in D_t

Table 3.3 shows the elected genre(s), based on Figures 3.4 and 3.5, for each song in D_t. We can see that different sections of a music piece can be classified into different genres and stand in contrast to the rest of the piece [5]. This shows that assigning only a single genre label to an entire music piece can be an ambiguous process. It is important to note that the elected genre for each clip is calculated in the following manner. For each clip, we collect the number of votes for a genre assigned to it. Then, we consider the genre with the maximum number of votes to be elected.

3.3 Discussions

There are a number of interesting results from our genre classification experiment. However, we limit our discussions to those results and issues pertinent to perceptual and automatic genre classification.

Genre Consensus: We have observed that there is little ambiguity in the votes for some of the tracks. Consequently, those tracks are easier to classify using a single label. To help explain this, Gjerdingen and Perrott [29] describe that "it may be helpful to think of an abstract space of genres. In that space, the location of Classical might be off by itself, a considerable distance from any popular genre". From this description, certain genres are intrinsically close to others in the abstract genre space while some are not.

A genre that is derived or influenced by others will reside closer to its influencing genres. Therefore, a music piece which is a composition of multiple artistic styles will result in confused boundaries in the abstract genre space, as different sections of it may sound like other genres, leading to a lower unanimity.

Length of Audio Excerpt: It has been a common practice for automatic genre classification approaches to evaluate a piece of music based on a random 30-second excerpt [41, 75, 76, 82, 83]. While this is efficient it is not effective as it can be a source of inaccuracy and confusion when trying to classify a corpus of music. One critical problem is how representative a randomly selected excerpt is. Performance is highly dependent on the dataset and, in particular, on the individual audio clips used for training and classification.

Furthermore, using the genre label associated with the entire track as the ground-truth for a random 30-second excerpt will result in varying performance. For instance, consider T_{19} from Table 3.4. Although the entire track has been annotated by Magnatune as *Rock*, classifying any of the four clips will result in a predicted genre different from the ground-truth. This result is confirmed by the participants in our study who have shown the track to be a composition of multiple genres for different sections.

Track	**Clip**	**Clas.**	**World**	**Elec.**	**Rock**	**Metal**	**Jazz**	**Punk**	**Pop**
19 (Rock)	51	**76.53%**	4.08%	1.02%	1.02%	0.00%	2.04%	1.02%	14.29%
	52	11.00%	2.00%	0.00%	8.00%	0.00%	14.00%	0.00%	**65.00%**
	53	12.00%	11.00%	0.00%	2.00%	0.00%	5.00%	1.00%	**69.00%**
	54	19.80%	5.94%	**49.50%**	0.99%	0.00%	10.89%	0.99%	11.88%

Table 3.4: Distribution of votes for T_{19}.

To enhance performance, we suggest that genre classification approaches take the entirety of a music piece into consideration. Although this might bring new challenges, we consider that this is a direction which should be further explored.

Multi-genre Labeling: Currently, genre classification paradigms are usually designed for strict classification, i.e., one excerpt must belong to one genre. This type of classification works well when genre boundaries are known and have clear distinctions. However, artists mix and incorporate different musical styles to create music, which can result in fuzzy genre boundaries [62]. The problem then becomes more challenging since standard classification approaches may not be sufficient.

New directions are needed to help deal with the ambiguity of classifying audio using only a single genre descriptor; it may be hard to assign unambiguously one label to one music piece. To address this problem and offer a realistic classification approach which is close to

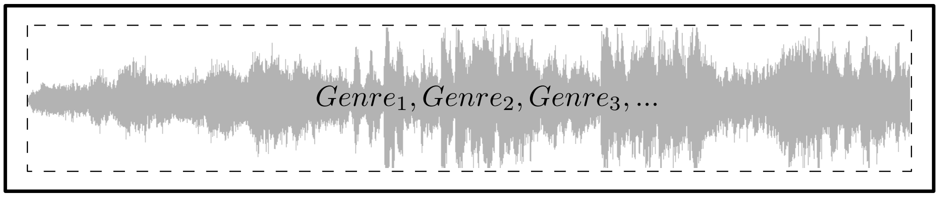

Figure 3.6: Multi-label Classification

the human experience, one needs to consider the assignment of multiple or compound genre labels to a music piece [17, 62]. This methodology is illustrated in Figure 3.6.

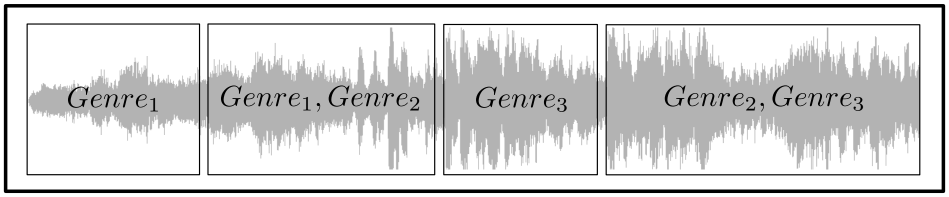

Figure 3.7: Segmented Multi-label Classification

However, as evident from our study, different sections of an audio recording can belong to a variety of genres. Therefore, classification approaches should ideally annotate each section of a music piece with a set of labels [44, 48]. Figure 3.7 depicts this scenario.

We conjecture that multi-label classification is a step in the right direction for tackling the issues of automatic genre classification. Although it suffers from difficulties similar to current classification approaches, such as the acquisition of accurate ground-truth, it has the potential of overcoming current performance limitations and offering a classification approach which is closer to the human experience.

3.4 Summary

The study in this chapter has demonstrated, using a series of perceptual experiments, that a music piece can be considered a composition of different genres. More specifically, the participants in our study consistently indicate that different portions of a music piece can be classified into different genres. As musical genres mix, there is an increasing need for genre classification to consider multi-label techniques. Furthermore, there is a strong need to consider the analysis of entire music pieces for the purpose of genre classification.

In the next chapter, we design a series of computational experiments to evaluate a set of multi-label classification approaches. Our goal is to support our speculation that music genre classification is a multi-label process.

Chapter 4

Multi-Label Genre Classification

Previous genre classification approaches are concerned with learning from a set of instances that are associated with a single genre label. However, a music piece may belong to an unrestricted set of musical genres, making single-label classification problematic, as evidenced from our study in Chapter 3.

In this chapter, we design a series of experiments to evaluate a set of multi-label classification algorithms. The goal of this work is to support our speculation that music genre classification is a multi-label process, in which multiple labels can be assigned to a music piece, and to show which algorithms are more suitable for multi-label genre classification.

The chapter is organized as follows. Section 4.1 introduces the area of multi-label classification. Section 4.2 presents our experimental setup for the evaluation of multi-label genre classification including, dataset preparation and feature extraction parameters. Experiment results are presented in Section 4.3. Finally, Sections 4.4 and 4.5 conclude the chapter with some discussions.

4.1 Introduction

Multi-label classification is the task of assigning one or multiple classes to an instance simultaneously, i.e., instances are associated with a set of labels $Y \subseteq L$, where L ($|L| > 1$)

is a set of disjoint class labels. This makes multi-label classification more difficult than traditional single-label classification. Despite this, multi-label classification has been increasingly used by current applications, including semantic classification of images, protein function classification, etc. [69] The initial work along this direction is motivated to handle the ambiguous situation in text categorization, where each document may belong to multiple topics simultaneously [86].

Despite the volume of previous results on multi-label classification, to the best of our knowledge, there has been limited work on the application of them to music genre classification. Lukashevich *et al.* [44] present a two-dimensional approach for genre classification. The multi-label classification problem is decomposed into multiple single-class problems. The novelty of the approach lies in the combination of segment-wise and domain-specific genre classification. The music collection used for testing is comprised of 430 full-length music pieces from 16 world genres. However, as described by Zhang and Zhou [86], decomposing multi-label problems into multiple binary classification problems does not consider the correlation between the labels of each instance. Therefore, the expressive power of such an approach can be limited.

In the following sections, we show our initial investigations on the application of multi-label classification algorithms on music genres.

4.2 Experiment Setup

4.2.1 Datasets

There are a variety of benchmark datasets available for multi-label classification in various domains. However, no dataset exists specifically for the task of multi-label genre classification on music. Current datasets used in the evaluation of genre classification are comprised of music pieces annotated with a single genre. Therefore, for the purpose of our experiment, we derive three multi-genre datasets from the *Magnatagatune* collection [38].

Magnatagatune is a collection of approximately 25,000 clips of music, each annotated with a combination of 188 different tags including tempo, mood, and style. The annotations are collected through an on-line game, referred to as “TagATune”, developed to collect tags for music and sound clips. Each clip, 29 seconds in length, is an excerpt of a music piece published by Magnatune. All of the tags in the dataset have been verified, i.e., a tag is associated with a clip only if it is generated independently by more than two players. Moreover, only those tags that are associated with more than 50 clips are included in the collection.

For our experiment, we are only interested in those clips annotated with musical genres. A set of 22 genre tags, including examples such as classical, dance, pop, rap, funk, opera, country, etc, are identified and used to create a subset of the Magnatagatune collection, referred to as D_s in our following discussions. D_s is created by filtering the Magnatagatune collection to contain only clips annotated with the set of selected genre tags. In addition, a clip is selected only if it is annotated with a minimum of two genres. A total of 3969 clips are included in D_s. The following three datasets are further derived from D_s.

The *Random* dataset, denoted D_{Ra}, consists of 1000 clips chosen at random from D_s. No other selection criteria are used in the creation of the dataset.

The *UniqueArtist* dataset, denoted D_{Ar}, consists of 198 clips derived from D_s. The dataset is created by randomly selecting a single clip form each artist in D_s. The use of an artist filter ensures that the artists in the test set are not present in the training set during classification.

In addition to using an artist filter, we also employ an album filter. The *UniqueAlbum* dataset, denoted as D_{Al}, consists of 375 clips derived from D_s. The dataset is created by randomly selecting a single clip form each album in D_s. Similar to the artist filter, the use of an album filter ensures that clips from the same album are not present in the training and testing sets simultaneously.

Table 4.1 displays the datasets and their associated statistics. The *label cardinality*

Dataset (D)	**Instances ($\|D\|$)**	**Labels ($\|L\|$)**	**Cardinality (LC)**	**Density (LD)**
Random	1000	22	2.278	0.1035
UniqueArtist	198	22	2.166	0.0984
UniqueAlbum	375	22	2.212	0.1005

Table 4.1: Multi-genre music datasets and their statistics

(LC) of a dataset D is the average number of labels each instance has in D and is used to indicate the number of alternative labels that characterize the instances in a multi-label training dataset [67]. Let D be a multi-label dataset consisting of $|D|$ multi-label examples (x_i, Y_i) where $i = 1, 2, \cdots, |D|$. Label cardinality is calculated as

$$LC(D) = \frac{1}{|D|} \sum_{i=1}^{|D|} |Y_i|.$$

The *label density* (LD) of a dataset D is the average number of labels of the instances in D divided by the total number of labels $|L|$ [67]. Label density is calculated as

$$LC(D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i|}{|L|}.$$

Label density takes into consideration the number of labels in the domain. Two datasets with the same label cardinality but with a difference in the number of labels (different label densities) might cause different behavior to multi-label classification.

4.2.2 Audio Parameters

Prior to classification, a music piece must be segmented and parameterized. The parameterization of music data can be based on audio features and their changes over time. However, there is no accepted criteria as which features are best for music classification [8]. Therefore, we use the following set of features which are commonly employed for genre classification: MFCC, ZCR, Spectral Centroid, Rolloff, Spectral Flux, and Chroma. See Chapter 2 for a discussion on them.

Following a general trend in MIR research, we consider a bag-of-frames approach. That is, in order to capture the fine-timescale structure of music, such as timbre, features are extracted from short audio frames. While promising results have been reported for a wide range of frame sizes for conventional single-label genre classification, we present a systematic exploration of the effects of frame size on multi-label classification. For each <dataset, classifier> pair, we examine the classification performance as we adjust the frame size, f_r, represented as the number of samples collected during a certain time period. Each pair is evaluated using $f_r \in \{256, 340, 512, 1024, 2048, 3200, 4096\}$, corresponding to approximately 16ms, 21ms, 32ms, 64ms, 128ms, 200ms, and 256ms, respectively. A total of seven experiments are performed for each <dataset, classifier> pair.

The process of extracting features from each frame yields a large number of feature vectors. To reduce them to a lower dimensional format, the feature vectors must be aggregated. Similar to the works by Tzanetaks and Cook [75] and Li *et al.* [41], frame-level features in our experiment are compressed into a single set of song-level features by fitting individual Gaussians to each feature (diagonal covariance among Gaussians). This results in a single feature vector for each music piece which can then be used for classification.

4.2.3 Multi-label Classifiers

The following multi-label classification algorithms, discussed in Chapter 2, are evaluated for multi-label genre classification: Binary Relevance (BR), Label Powerset (LP), Random k-Labelsets (RakEL), Calibrated Label Ranking (CLR), ML-kNN, HOMER, and Instance Based Logistic Regression (IBLR).

Furthermore, a Decision Tree (DT) and Support Vector Machine (SVM) are used as base classifiers in BR, LP, CLR, HOMER, and RakEL. A total of 12 classifiers, presented in Table 4.2, are derived for experimentation. The Mulan [71] open source library for multi-label learning is used to train each of the classification algorithms using default parameters,

Classifier	Base Classifier	Classifier	Base Classifier
BR	DT	HOMER	DT
BR	SVM	HOMER	SVM
LP	DT	RakEL	DT
LP	SVM	RakEL	SVM
CLR	DT	ML-kNN	N/A
CLR	SVM	IBLR	N/A

Table 4.2: Multi-label classifiers along with the associated base classifier, where applicable.

e.g., the number of neighbors is set to 10 for ML-kNN and IBLR; the SVM is trained with a linear kernel and complexity constant equal to one.

A total of 84 classification experiments are performed using 10-fold cross validation for each dataset presented in Table 4.1, i.e., the dataset is divided into two groups: one used for training and the other for testing. More specifically, each dataset is divided into 10 subsets of (approximately) equal size. Classification is then performed 10 times, each time leaving out one of the subsets from training and using it for testing. For each of the 12 classifiers, seven experiments are conducted using different frame sizes. A total of 252 experiments are conducted on the Random, UniqueArtist, and UniqueAlbum datasets, as introduced above.

In order to evaluate the performance of multi-label classification algorithms, different evaluation measures, as presented below, are employed.

4.2.4 Evaluation Measures

Performance evaluation of multi-label classification requires different measures than those used in traditional single-label classification. In our experiments, the evaluation measures we select are as follows: *Hamming Loss*, *One-Error*, *Coverage*, *Ranking Loss*, and *Average Precision*. For discussions on them, see Chapter 2.

4.3 Experiment Results

In this section, we present the results from our experiments on the three datasets Random (D_{Ra}), UniqueArtist (D_{Ar}), and UniqueAlbum (D_{Al}). To reduce the number of figures, we omit reporting results for the evaluation measure Coverage as it exhibits patterns similar to Ranking Loss.

4.3.1 Random Dataset

In the first set of experiments, we evaluate the multi-label classifiers using D_{Ra}. Table 4.3 shows the comparison of performance over all frame sizes f_r; the best result for each measure is shown in bold face. We note that in the Table 4.3, (↓) indicates better performance when the result is smaller while, (↑) indicates better performance when the result is bigger. The frame-level features, calculated for each f_r, are used directly for genre classification. The classifier CLR(SVM) outperforms the other classifiers for all evaluation measures. To our surprise, LP performs poorly for all evaluation measures, with the exception of Hamming Loss. Moreover, LP(SVM) performs poorly for One-Error, which might be due to the large number of labels in the dataset [70].

	HamLoss ↓	**OneError ↓**	**Coverage ↓**	**RankLoss ↓**	**AvgPrec ↑**
BR (DT)	0.079	0.355	8.884	0.218	0.642
BR (SVM)	**0.063**	0.304	10.497	0.273	0.629
CLR (DT)	0.071	0.243	3.857	0.076	0.767
CLR (SVM)	**0.063**	**0.229**	**3.588**	**0.070**	**0.780**
HOMER (DT)	0.095	0.378	7.658	0.187	0.636
HOMER (SVM)	0.070	0.296	7.573	0.176	0.690
IBLR	0.071	0.285	4.093	0.087	0.748
LP (DT)	0.092	0.493	10.953	0.324	0.542
LP (SVM)	0.070	0.973	19.496	0.822	0.135
MLkNN	0.067	0.255	3.990	0.082	0.761
RAkEL (DT)	0.070	0.255	6.523	0.145	0.729
RAkEL (SVM)	**0.063**	0.259	9.949	0.246	0.665

Table 4.3: Average classification performance of D_{Ra} for all frame sizes.

In the following, we present the results for each classifier over individual frame sizes.

For illustration purpose, we only present the six top classifiers that have the best classification performance. Figure 4.1 shows the classification performance of CLR, HOMER(SVM), IBLR, ML-kNN, and RAkEL(DT) using the evaluation measures Average Precision, Hamming Loss, Coverage, and One-Error. Classification performance is reported for frame sizes f_r chosen from $\{256, 340, 512, 1024, 2048, 3200, 4096\}$.

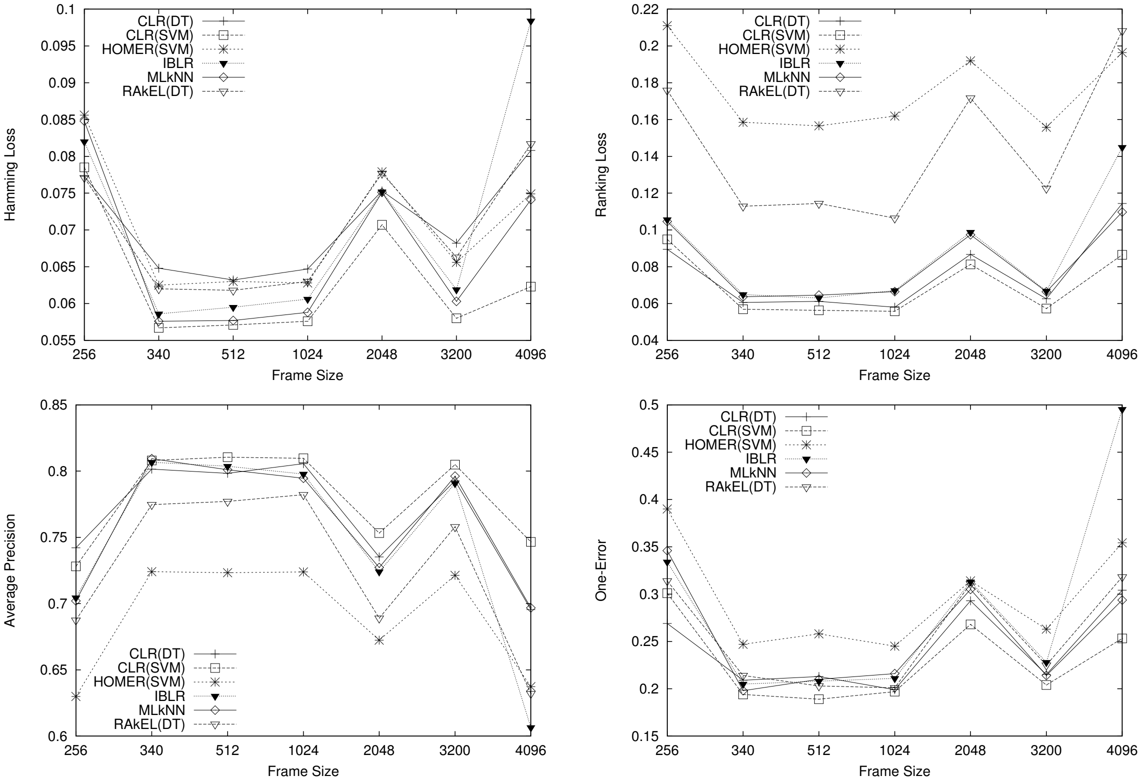

Figure 4.1: Classification performance using the D_{Ra} dataset for different frame sizes

We first observe that all classifiers tend to follow the same pattern. That is, a decrease in performance is observed across all measure when $f_r = 2048$ and $f_r = 4096$. Moreover, IBLR's performance decreases dramatically for Hamming Loss and One-Error when $f_r = 4096$. It is important to note the performance achieved when $f_r = 3200$. More specifically, for this frame size, we observe a similar classification performance but with a decreased experiment running time. Therefore, it might be more advantageous to model the audio signal using larger frames while at the same time maintaining classification performance.

Interestingly, all of the classifiers tend to increase in performance and then plateau when $f_r = \{340, 512, 1024\}$. Although the variance in folds from the cross validation makes it difficult to speak of trends, it appears that classification performance is optimal at a frame size between 340 and 1024 for different classifiers. More specifically, CLR(SVM) outperforms the others when $f_r = 512$ for a majority of evaluation measures.

We observe that IBLR and ML-kNN perform analogous to CLR(SVM) for a variety of measures and follow similar patterns. Moreover, we find situations where IBLR outperforms ML-kNN. This is expected as IBLR has been shown to outperform ML-kNN in previous works [14].

HOMER has been shown to handle domains with large number of labels more accurately than Binary Relevance [68]. Although this result is confirmed in our experiments, we find that it performs poorly as evaluated by Coverage, Ranking Loss, and Average Precision, in comparison to CLR, IBLR, ML-kNN, and RAkEL. This is surprising as HOMER is designed to deal with high dimensionality of label space by splitting up the label set. Whether this is true in general needs further investigation.

	f_r	**HamLoss** ↓	**OneError** ↓	**Coverage** ↓	**RankLoss** ↓	**AvgPrec** ↑
CLR(SVM)	512	0.057	0.189	3.206	0.056	0.811
ML-kNN	340	0.058	0.198	3.442	0.064	0.809
IBLR	340	0.059	0.205	3.451	0.065	0.808

Table 4.4: Comparison of CLR(SVM), ML-kNN, and IBLR for D_{Ra}

Overall, we observe that CLR(SVM), ML-kNN, and IBLR report the best performance on D_{Ra}. Table 4.4 presents a comparison of these results.

4.3.2 UniqueArtist Dataset

Table 4.5 shows the comparison of multi-label classifiers, averaged over all frame sizes, for the *UniqueArtist* dataset, D_{Ar}. We find it interesting that for this dataset the average classification performance of each classifier is lower than the performance on D_{Ra}. For example, while CLR(SVM) and CLR(DT) perform well on D_{Ra}, we note a decrease in their classification performance on D_{Ar}. Additionally, we notice a performance difference between IBLR and ML-kNN on this dataset. These results are expected as classification accuracy can be lower if an artist filter is used [56], as discussed in Chapter 2.

	HamLoss ↓	**OneError** ↓	**Coverage** ↓	**RankLoss** ↓	**AvgPrec** ↑
BR (DT)	0.094	0.466	10.221	0.271	0.545
BR (SVM)	**0.070**	0.400	12.051	0.341	0.518
CLR (DT)	0.082	0.331	5.074	0.116	0.679
CLR (SVM)	0.071	**0.303**	**4.615**	**0.104**	**0.703**
HOMER (DT)	0.121	0.520	9.708	0.266	0.508
HOMER (SVM)	0.086	0.399	8.751	0.223	0.601
IBLR	0.104	0.512	6.225	0.160	0.580
LP (DT)	0.107	0.619	12.577	0.384	0.444
LP (SVM)	0.085	0.962	19.024	0.771	0.146
MLkNN	0.080	0.338	5.082	0.118	0.671
RAkEL (DT)	0.085	0.348	8.163	0.200	0.629
RAkEL (SVM)	0.071	0.356	11.441	0.312	0.559

Table 4.5: Average classification performance of D_{Ar} for all f_r

On average, we observe that CLR(SVM), CLR(DT), and ML-kNN demonstrate good performance on D_{Ar}. As before, CLR(SVM) ranks first in the majority of evaluation measures while the performance of CLR(DT), IBLR, and ML-kNN varies. We observe that LP performs poorly, once again, for all evaluation measures. However, for One-Error and Ranking Loss, LP(SVM) is evaluated better on D_{Ar}.

Figure 4.2 shows the performance comparison of the chosen classifiers for each frame size on D_{Ar}. Once again we observe that the classifiers follow a similar trend with regard to frame size. The classification performance initially increases when $f_r = 340$ and decreases until $f_r = 2048$. However, the performance peaks for various classifiers, as evaluated for some measures, when $f_r = 4096$. For instance, CLR(SVM) peaks in performance for Coverage at a frame size of 4096. We find that the classifiers have an inferior performance when $f_r = 2048$; a similar result is observed for D_{Ra}.

The performance tends to be good for a variety of evaluation measures and classifiers when $f_r \in \{340, 512, 4096\}$. Specifically, CLR(SVM) outperforms the other classifiers and

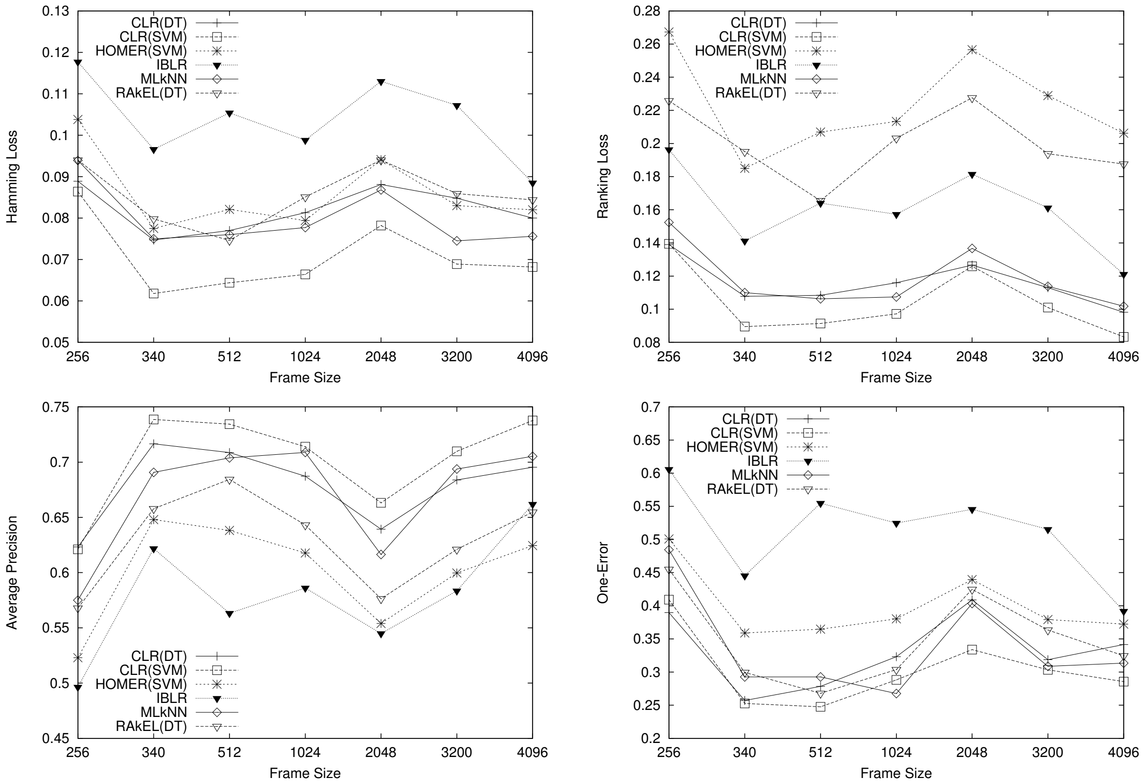

Figure 4.2: Classification performance using the D_{Ar} dataset for different frame sizes

achieves the best performance on D_{Ar} when $f_r = 340$. It is interesting to note that the performance of the classifiers on D_{Ar} increases for $f_r = 4096$, as compared to D_{Ra}, where an improvement is not reported by all the evaluation measures.

Furthermore, we have a similar observation when $f_r = 4096$ as we do on D_{Ra} with respect to the relation between classification performance and experiment running time. To our surprise, we find that IBLR does not perform well on D_{Ar}. Inadequate results are reported when compared to CLR, HOMER(SVM), ML-kNN, and RAkEL(DT). This is in contrast to its performance on D_{Ra}, where it produces comparable results to CLR(SVM). We note that ML-kNN outperforms IBLR for measures of Average Precision, Hamming Loss and One-Error. This is unexpected as IBLR has been shown to improve upon ML-kNN [14].

In summary, CLR(SVM), CLR(DT), and ML-kNN report the best performance on D_{Ar}. Table 4.6 presents a comparison of the results.

	f_r	**HamLoss** $\downarrow$	**OneError** $\downarrow$	**Coverage** $\downarrow$	**RankLoss** $\downarrow$	**AvgPrec** $\uparrow$
CLR(SVM)	340	0.067	0.253	4.134	0.090	0.739
CLR(DT)	340	0.075	0.257	4.906	0.108	0.717
ML-kNN	4096	0.076	0.314	4.565	0.102	0.705

Table 4.6: Comparison of CLR(SVM), CLR(DT), and ML-kNN for D_{Ar}

4.3.3 UniqueAlbum Dataset

For the final set of experiments we evaluate the multi-label classifiers using the *UniqueAlbum* dataset, D_{Al}. Table 4.7 shows the comparison of classifiers, averaged over all f_r for D_{Al}. We observe that for this dataset the average performance of each classifier is lower than its counterpart on D_{Ra} but higher than on D_{Ar}. For example, CLR(SVM) achieves an Average Precision of 0.780 on D_{Ra}, 0.703 on D_{Ar}, and 0.723 on D_{Al}. On average, CLR(SVM), CLR(DT), and ML-kNN outperform the other classifiers on D_{Al} while CLR(SVM) achieves the highest performance.

	HamLoss $\downarrow$	**OneError** $\downarrow$	**Coverage** $\downarrow$	**RankLoss** $\downarrow$	**AvgPrec** $\uparrow$
BR (DT)	0.092	0.437	10.360	0.276	0.564
BR (SVM)	**0.073**	0.445	12.127	0.347	0.511
CLR (DT)	0.084	0.334	4.694	0.104	0.694
CLR (SVM)	**0.073**	**0.301**	**4.261**	**0.091**	**0.723**
HOMER (DT)	0.114	0.485	9.170	0.243	0.541
HOMER (SVM)	0.086	0.391	8.612	0.215	0.613
IBLR	0.089	0.424	5.093	0.123	0.651
LP (DT)	0.107	0.587	12.289	0.377	0.462
LP (SVM)	0.083	0.975	19.103	0.780	0.143
MLkNN	0.079	0.333	4.725	0.105	0.693
RAkEL (DT)	0.085	0.345	7.635	0.183	0.646
RAkEL (SVM)	**0.073**	0.375	11.363	0.309	0.563

Table 4.7: Average classification performance of D_{Al} for all f_r

Figure 4.3 shows the performance comparison of CLR, HOMER(SVM), IBLR, ML-kNN, and RAkEL(DT) for each frame size $f_r \in \{256, 340, 512, 1024, 2048, 3200, 4096\}$ on D_{Al}. We first notice that all classifiers follow similar patterns as in D_{Ra} and D_{Ar}. More specifically, the classifiers closely emulate those patterns found in D_{Ar} with an exception that the performance increases for $f_r = 4096$. For instance, CLR(SVM) achieves good performance when $f_r = 4096$, behind which ML-kNN and CLR(DT) follow shortly and offer comparable performance across the evaluation measures. This is in contrast to D_{Ra}, where the performance peaks when $f_r \in \{340, 512, 1024\}$.

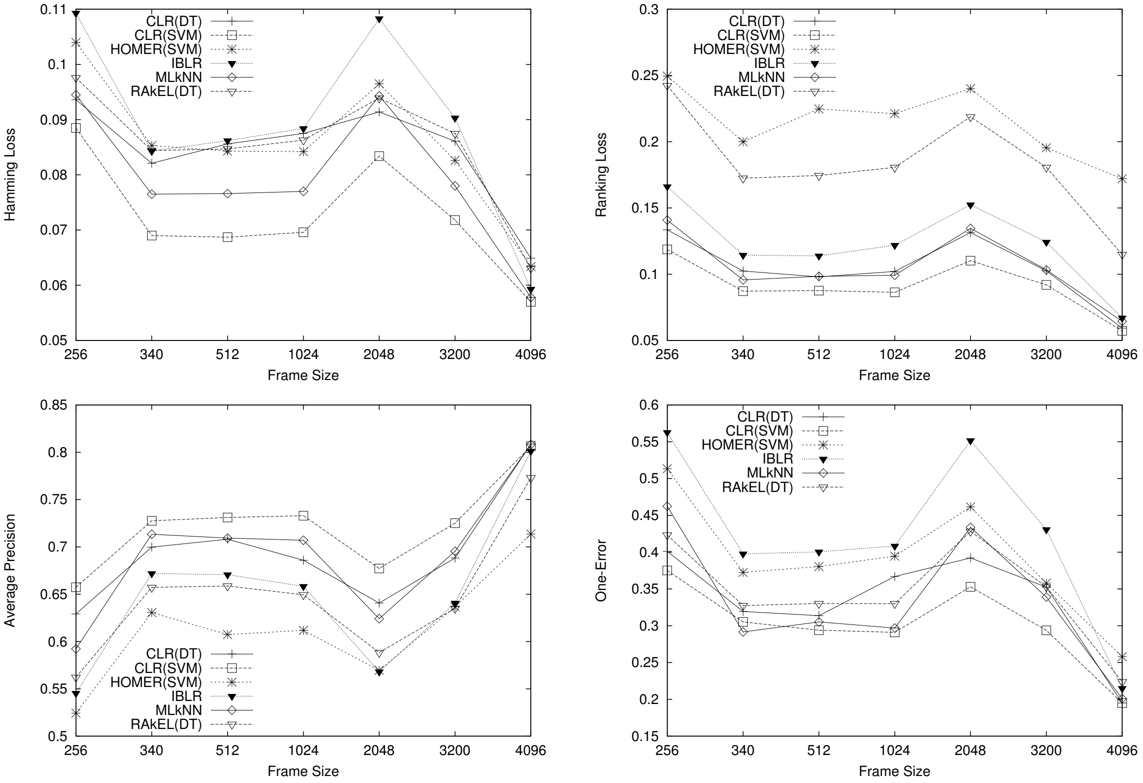

Figure 4.3: Classification performance using the D_{Al} dataset for different frame sizes

	f_r	**HamLoss** ↓	**OneError** ↓	**Coverage** ↓	**RankLoss** ↓	**AvgPrec** ↑
CLR(SVM)	4096	0.057	0.195	3.23	0.057	0.806
CLR(DT)	4096	0.065	0.195	3.33	0.059	0.809
ML-kNN	4096	0.058	0.201	3.52	0.064	0.808

Table 4.8: Comparison of CLR(SVM), CLR(DT), and ML-kNN for D_{Al}

As before, we observe that IBLR performs poorly on D_{Al} for a selection of evaluation measures. However, we note its performance is moderate for Ranking Loss in comparison to HOMER(SVM) and RAkEL(DT). Overall, CLR(SVM), CLR(DT), and ML-kNN demonstrate good achievement on D_{Al}. Table 4.8 shows a comparison of results for D_{Al}.

4.4 Discussions

Realizing that multi-label genre classification is well motivated and little previous work has been done toward this, we show our initial attempts in this chapter along this direction. We perform a series of empirical experiments using a set of multi-label classification algorithms. The results in our experiments show the merits of multi-label genre classification. In all three datasets, we observe a set of classifiers consistently perform well, namely, CLR(SVM), CLR(DT), and ML-kNN. In the following, we also discuss some other related issues.

Frame Size: It is difficult to select an optimal frame size for all classifiers. For instance, CLR(SVM) performs good on D_{Ra} when $f_r = 512$ and it does the same on D_{Al} when $f_r = 4096$. In spite of this, we also observe patterns that can be exploited to achieve adequate performance. For instance, a variety of classifiers tend to perform well when $f_r \in \{340, 512, 1024\}$. Although good performance may not be achieved, a trade off is made between an exhaustive search for the best frame size and a decrease in performance.

Data Selection: We observe, on average, D_{Ra} achieves the best dataset performance as shown in Table 4.3. Moreover, classification performance on D_{Ar} is lower than D_{Ra} and D_{Al}. The artist filter limits performance by restricting artists from appearing in both training and testing sets. Consequently, there is little overlap in artistic styles among the music pieces in these sets, resulting in a “unique” collection used for training and testing. In addition, D_{Al} employs an album filter to exclude music pieces from the same album appearing in both training and testing sets. However, this does not exclude artists from appearing in them. For this reason, we observe classification performance on D_{Al} is better than D_{Ar}.

Texture Window: As previously discussed in Chapter 2, classification performance can be increased by using a *texture window*. This captures the long term nature of sound “texture”

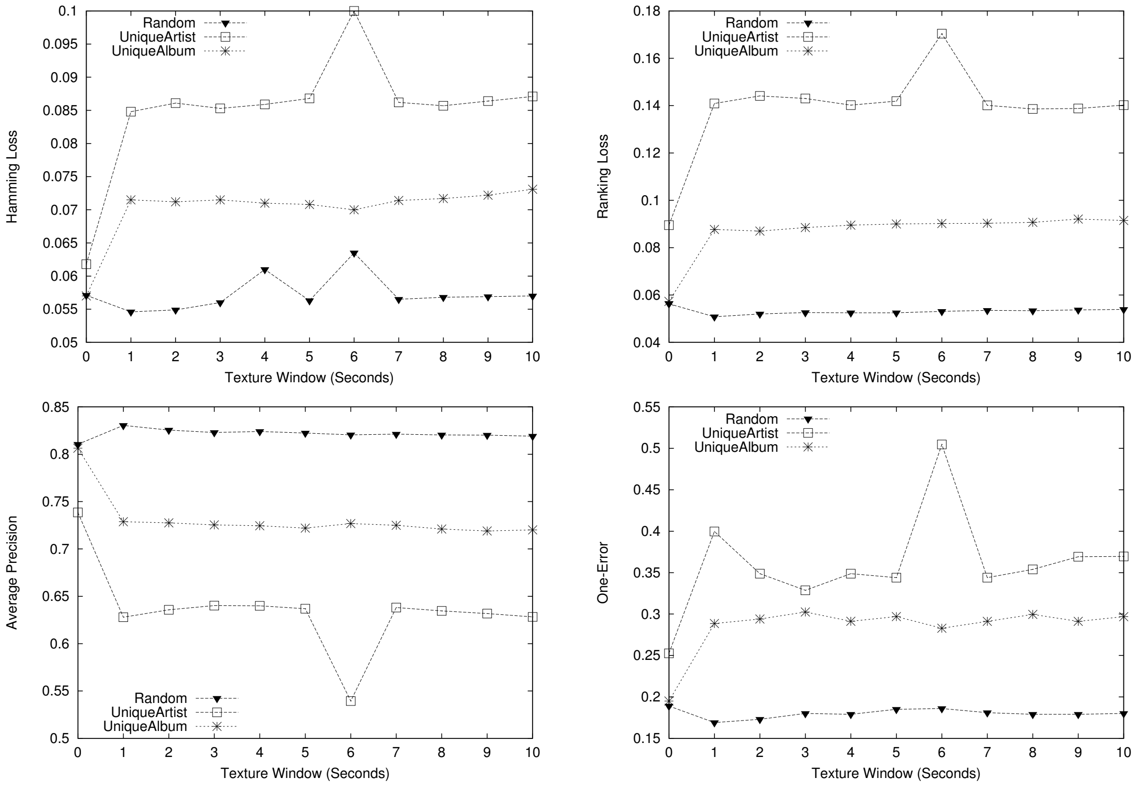

Figure 4.4: Classification performance of CLR(SVM) on each dataset using a texture window.

by estimating the parameters of a running multi-dimensional Gaussian distribution. It has been shown by Tzanetakis and Cook [75] that a texture window of approximately one (1) second can increase classification performance. Therefore, we apply a texture window, with sizes varying from one (1) second to ten (10) seconds, to the feature extraction process and plot the classification performance on each of the three datasets. Figure 4.4 shows the classification results of CLR(SVM) on each dataset using a texture window.

We observe that the classification performance on D_{Ra} is increased when a texture window of one (1) second is used. However, the performance dramatically decreases for the other datasets when a texture window of any size is applied. These results are consistently observed for other classifiers. This is contrary to what is commonly reported in other studies [75, 83]. It would be interesting to further investigate along this direction.

4.5 Summary

In this chapter we report our recent study on multi-label music genre classification. Although we only create three datasets in our experiments, it is our belief that our results should be applicable to other music collections. The results presented not only show the necessity of multi-label genre classification on music but also offer insight into which algorithms are more suitable for the task. Further investigation is needed into alternative segmentation and feature extraction algorithms in the hope of further increasing classification performance.

In the next chapter, we propose a set of ensemble techniques to combine together the predictive power of multi-label classifiers, with the aim to further improve the performance of music genre classification.

Chapter 5

Ensemble of Multi-Label Classifiers

In this chapter, we propose an ensemble of classifiers for improving multi-label music genre classification. Our approach is to combine the predictive power of multiple classifiers and show performance gains over the individual multi-label classification algorithms introduced in Chapter 2 and Chapter 4.

This chapter is organized as follows. Section 5.1 introduces ensemble techniques for combining multiple classifiers. Section 5.2 discusses our proposed ensemble technique for combining multi-label classifiers. In Section 5.3 we design a series of experiments to evaluate our techniques. Experiment results are presented in Section 5.4. Finally, Sections 5.5 and 5.6 summarize the chapter with discussions and future work.

5.1 Introduction

Despite extensive work in multi-label classification, there exist two major challenges, among others, especially in music genre classification. The first challenge is that we can have highly imbalanced data sets, due to the availability of instances for some labels, while the second is related to our limited knowledge regarding the correlation among class labels for a given dataset.

Most multi-label classification approaches are designed to focus mainly on the second problem and limited work has been devoted to handling imbalanced datasets [65]. One approach is an ensemble of multi-label classifiers, which consists of a set of individually trained classifiers, $H_1, H_2, ..., H_q$, whose predictions are combined when classifying instances. Figure 5.1 illustrates the basic approach for building an ensemble of diverse classifiers. This approach is generally more accurate and achieves greater predictive performance than any of the individual classifier making up the ensemble [12].

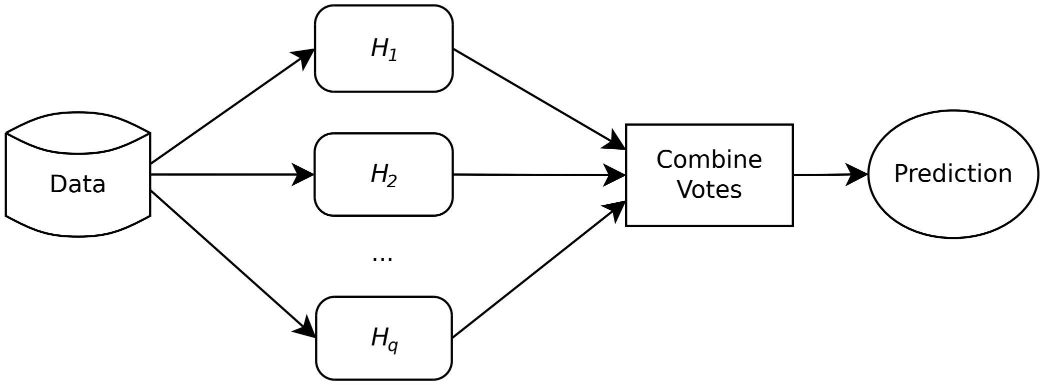

Figure 5.1: Building an ensemble of classifiers: generate a set of diverse classifiers, $H_1, H_2, ..., H_q$, and combine the predictions for an unknown instance using voting strategies.

Ensembles can be homogeneous, where every individual classifier is constructed using the same algorithm, or heterogeneous, where each classifier is constructed from a different algorithm [12, 65]. Some multi-label classification algorithms directly use homogeneous or heterogeneous ensemble techniques internally to improve overall performance. For example, Instance Based Logistic Regression [14], introduced in Chapter 2 and Chapter 4, uses a combination of Logistic Regression and Nearest Neighbor classifiers to improve the overall performance.

Tahir *et al.* [65] present a first study, as claimed, on combining the outputs of multi-label classification algorithms. They propose two heterogeneous ensemble techniques and apply them to publicly available datasets using a selection of evaluation measures. The results show that these approaches provide significant performance improvements when compared with individual multi-label classifiers.

The goal of our work in this chapter is to use a heterogeneous ensemble of existing multi-label classification algorithms to improve music genre classification. The advantage of using an ensemble approach is that both aforementioned problems can be handled simultaneously. That is, training set imbalance can be addressed by using multiple multi-label classifiers. Moreover, the correlation problem can be handled using classifiers that consider label correlations. In the following, we introduce a set of ensemble techniques that combine multiple classifiers for the purpose of music genre classification.

5.2 Ensemble of Multi-label Classifiers (EML)

Let D denote a set of music pieces and let $L = \{l_1, l_2, ...l_N\}$ be the finite set of labels. Given a training set $D_s = \{(x_1, Y_1), (x_2, Y_2), \cdots, (x_m, Y_m)\}$, where $x_i \in D$ is a single music piece and $Y_i \subseteq L$ is the label set associated with x_i. We attempt to design a multi-label classifier that predicts a set of labels for an unseen music piece from a test set $D_t = \{(x_1, Y_1), (x_2, Y_2), \cdots, (x_n, Y_n)\}$.

An ensemble of multi-label (EML) classifiers trains q multi-label classifiers $H_1, H_2, ..., H_q$. In our work, all classifiers are likely to be unique and able to generate different multi-label predictions. For an unseen instance x_j, classifier H_k produces two N-dimensional vectors: a score vector $P_k^j = [p_{1,k}^j, p_{2,k}^j, \cdots, p_{N,k}^j]$, where the value $p_{b,k}^j$ is the confidence of the class label l_b assigned by classifier H_k being correct, and a bipartition vector $V_k^j = [v_{1,k}^j, v_{2,k}^j, \cdots, v_{N,k}^j]$ where $v_{b,k}^j$ is 1 if the class label l_b is predicted by classifier H_k and 0 otherwise.

We denote by H_{eml} the classifier obtained after applying an ensemble technique to the q classifiers. We use $P_{eml}^j = [p_{1,eml}^j, p_{2,eml}^j, \cdots, p_{N,eml}^j]$ to represent the resulting score vector for the unseen instance y_j and use $V_{eml}^j = [v_{1,eml}^j, v_{2,eml}^j, \cdots, v_{N,eml}^j]$ to represent the corresponding binary vector.

There are a variety techniques to combine the outputs of these q classifiers. While some combiners are based on the bipartition vector, others are based on the score vector. In this work, we propose two novel ensemble techniques – one that considers both the bipartition and score vectors and the other that utilizes a taxonomy of musical genres. These techniques are discussed below.

5.2.1 Bipartition-based Ensemble

Intersection Rule

The *Intersection rule* (denoted as EML_I) calculates the intersection of the bipartition vectors from the q classifiers using $v^j_{i,eml} = \sum_{s=1}^{q} \cap v^j_{i,s}$,for $i = 1, 2, \cdots, N$, i.e., the binary values for each label in all vectors are combined using the logical AND operator. The set of output labels common to all classifiers result in an ensemble decision. We propose this naive method to emphasize the common labels output by all individual classifiers.

Union Rule

The *Union rule* (denoted as EML_U) calculates the union of the binary vectors from the q classifiers using $v^j_{i,eml} = \sum_{s=1}^{q} \cup v^j_{i,s}$, for $i = 1, 2, \cdots, N$, i.e., the binary values for each label in all vectors are combined using the logical OR operator. An ensemble decision is constructed by computing, for each label, the union of outputs from the q classifiers. We propose this method to optimistically select a label for the ensemble if it is selected by any of the individual classifiers.

Majority Vote Rule

The *Majority Vote rule* (denoted as EML_{MV}) counts the number of times a label appears in the q classifiers. It is one of the most frequently used methods for combining label outputs from classifiers [36],

$$v^j_{i,eml} = \begin{cases} 1 & \text{if } \sum_{s=1}^{q} v^j_{i,s}/q \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

where $i = 1, 2, \cdots, N$.

5.2.2 Score-based Ensemble

Minimum Rule

The *Minimum rule* [36, 65] (denoted as EML_{Min}) represents a pessimistic view of the scores from the q classifiers and only considers the smallest score for each class label. We calculate the minimum score as shown in the following equation,

$$p^j_{i,eml} = min_s p^j_{i,s},\ i = 1, 2, \cdots, N.$$

Maximum Rule

The *Maximum rule* [36, 65] (denoted as EML_{Max}) represents an optimistic view of the scores from the q classifiers and only considers the largest score for each class label. We calculate the maximum score as shown in the following equation,

$$p^j_{i,eml} = max_s p^j_{i,s},\ i = 1, 2, \cdots, N.$$

Mean Rule

The *Mean rule* [36, 65] (denoted as EML_{Mean}) considers the scores from the q classifiers for a class label. We calculate the mean score as shown in the following equation,

$$p^j_{i,eml} = \sum_{s=1}^{q} v^j_{i,s}/q,\ i = 1, 2, \cdots, N.$$

Top-k Rule

We propose the *Top-k rule* (denote as EML_{Topk}), which is a combination of the Maximum and Mean rules. It selects the top k largest scores and averages them. We use the following equation to calculate $p^j_{i,eml}$. Constant k is a user-selected parameter. This method represents our desire to have a degree of certainty when calculating the score for H_{eml}.

$$p^j_{i,eml} = avg(topk(v^j_{i,1}, v^j_{i,2}, \cdots, v^j_{i,q}))$$

where $i = 1, 2, \cdots, N$, function $topk(\cdot)$ picks the largest k elements from a set and function $avg(\cdot)$ averages them.

5.2.3 Score-based Label Selection

In addition, we also consider the score output from each classifier and select the top n scores and the corresponding labels. We then merge the results from all the classifiers, from which we select the top k labels that appear the most, where $n \geq k$. If there is a tie, we arbitrarily select one. For each of these k labels, we set the corresponding entry to be “1” in the bipartition vector for the final ensemble H_{eml}. For the other labels, we set the respective entries to be “0”. In this proposed technique, since we first select the top n scores and then select the top k class labels accordingly, we refer to it as $EML_{C_nL_k}$ in the following discussions.

5.2.4 Hierarchical Label Substitution

We propose a novel ensemble technique that utilizes taxonomy information of music genres. We call this technique *Hierarchical Label Substitution* (HLS). The technique reduces the number of labels using a substitution method. A set of multi-label classifiers are trained and the resulting output is combined using one of the bipartition-based techniques described above. A set of labels in the ensemble decision are then substituted based on a local genre hierarchy, represented as a *taxonomy.* For our experiments, we derive our local genre hierarchy based on the taxonomy of music genres developed by Allmusic[1] to help navigate music catalogs. Figure 5.2 depicts a small portion of the local genre hierarchy.

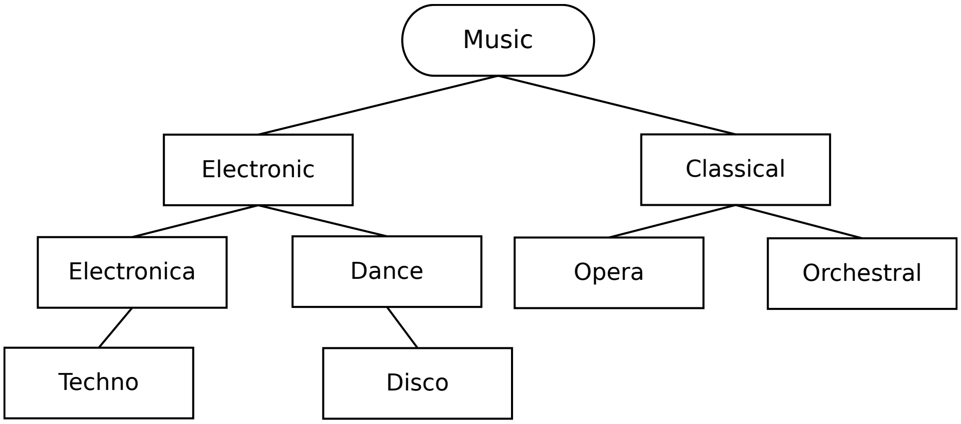

Figure 5.2: Example genre taxonomy used for Hierarchical Label Substitution.

Our hierarchy is constructed to only contain music genres in our datasets and consists of top level nodes referred to as "meta genres" [19], as shown in Figure 5.2. These should represent main musical genres, such as Classical, Rock, Jazz, etc. The hierarchy is further defined to consist of children nodes, i.e., *sub-genres*, which can be conceptualized as specific genres derived from a parent genre and become more concrete as the depth increases. For example, from Figure 5.2, *Techno* is considered as a sub-genre of *Electronica.* It is important to emphasize that our taxonomy is by no means authoritative but our ensemble technique is applicable to any other genre taxonomy.

We define a *depth* d for label substitution, where all labels below the level at d are substituted with the parent label at this level. For example, for genre *Electronic*, at $d = 1$, any occurrence of labels in the resulting classifier output below level d will be replaced with their parent label recursively, e.g., *Disco* would be replaced with *Dance* and *Techo* with *Electronica.* It is easy to see that, as d decreases, the resulting label set shrinks.

Although our ensemble technique produces a smaller label set, the usefulness lies in its

[1] http://www.allmusic.com.

ability to simplify the label space by reducing the number of possible "overlapping" genres. Moreover it has been shown [1] that there is little consensus between the genre labels used by individuals. Simplification to a common set of high-level genres may provide a remedy to this.

5.3 Experiment Setup

In this section we describe the preparation for our experiments. We apply the ensemble techniques discussed above to a set of multi-label classification algorithms and evaluate their performance. The evaluation is conducted on the three datasets, D_{Ra}, D_{Ar}, and D_{Al}, using 10-fold cross validation.

5.3.1 Multi-label Classification Algorithms

We select and combine the following multi-label classification algorithms using a set of ensemble techniques: Random k-Labelsets (RakEL), Calibrated Label Ranking (CLR), Multi-label k-Nearest Neighbor (ML-kNN), Hierarchy of Multi-label Classifiers (HOMER), and Instance Based Logistic Regression (IBLR). As shown in Chapter 4, these classification algorithms achieve good performance on a variety of music datasets, as compared to the other multi-label classifiers. Furthermore, a Decision Tree is used as a base classifier in RakEL while a Support Vector Machine is used in CLR and HOMER.

5.3.2 Audio Parameters

Similar to Chapter 4, we select the following set of features for classification: MFCC, Zero Crossing Rate, Spectral Centroid, Rolloff, Spectral Flux, and Chroma. Moreover, to capture the fine-timescale structures of music, we extract features from short audio frames. However, it is difficult to select an optimal frame size for all classifiers as discussed in Section 4.4.

For the purpose of our experiments, we select a frame size of 340 samples, corresponding to approximately 23 milliseconds. This size has been commonly used in MIR literature for a variety of tasks.

5.3.3 Evaluation Measures

To evaluate the performance of our proposed ensemble techniques, a variety of evaluation measures are employed. Recall from Chapter 2 that these measures can be categorized into three groups: example-based, label-based and rank-based. The first two groups are based on the bipartition vectors while the third group derives ranking information from the score vectors and conducts evaluations accordingly. We consider the following measures.

- Example-based: Hamming Loss (HamLoss), Accuracy (Accu.), Recall, F-Measure (F_1) and Precision (Prec.).
- Label-based: Micro Precision (MicroP), Micro F-Measure (MicroF_1), Micro Recall (MicroR), Macro Precision (MacroP), Macro F-Measure (MacroF_1) and Macro Recall (MacroR).
- Rank-based: Average Precision (AvgPrec), Coverage , Ranking Loss (RankLoss), and One-Error.

5.3.4 Ensemble Parameters

We use the following ensemble parameters in our experiments. For hierarchical label substitution (HLS), we set $d = 1$, i.e., sub-genres are replaced with their respective "meta genre". In addition, we set $k = 3$ for EML_{Topk}, that is, we select the top 3 largest scores and average them. For score-based label selection ($EML_{C_nL_k}$), we set $n = 3$ and $k = 2$. More specifically, we first select the top 3 scores from each classifier and then select the top 2 class labels accordingly. We provide more discussions on these parameters in Section 5.5.

5.4 Results

In this section, we present the results from our experiments on the three datasets Random (D_{Ra}), UniqueArtist (D_{Ar}), and UniqueAlbum (D_{Al}). Each ensemble technique is compared with the individual multi-label classifiers to determine its performance.

	HamLoss ↓	Accu. ↑	Recall ↑	F_1 ↑	Prec. ↑
CLR	0.0567	0.5697	0.5959	0.8098	0.8255
IBLR	0.0586	0.6045	0.6481	0.8040	0.7715
MLKNN	0.0576	0.6042	0.6354	0.8120	0.7764
RAkEL	0.0620	0.5918	0.6648	0.7739	0.7357
HOMER	0.0635	0.6255	0.6949	0.7867	0.7299
EML$_U$	0.0710	0.6239	0.7893	0.7568	0.6861
EML$_I$	0.0607	0.4828	0.4936	0.8008	0.8853
EML$_{MV}$	0.0544	0.6233	0.6559	0.8123	0.7958
HLS(EML$_U$)	0.0402	0.7001	**0.8435**	0.8177	0.7652
HLS(EML$_I$)	0.0337	0.6198	0.6198	**0.9180**	**0.9453**
HLS(EML$_{MV}$)	**0.0304**	**0.7322**	0.7491	0.8978	0.8741
EML$_{C_nL_k}$	0.0601	0.6277	0.6731	0.7843	0.7390

Table 5.1: Comparison of bipartition-based ensembles for D_{Ra} using example-based measures.

5.4.1 Random Dataset

Table 5.1 shows the comparison of the bipartition-based ensemble techniques with the individual multi-label classifiers for D_{Ra} using example-based evaluation measures, as discussed in Section 5.3.3; the best result on each measure is shown in bold face. Note that in the Table 5.1, (↓) indicates better performance when the value is smaller while (↑) indicates better performance when the value is bigger. When the individual multi-label classifiers are compared with each other, it is hard to pick between CLR and HOMER for this dataset since their performance is comparable for different measures. Moreover, when we compare the classifiers using label-based measures, presented in Table 5.2, we find that CLR performs well for MicroP, MacroP, and MacroF_1, while IBLR, HOMER, and RAkEL perform well for MicroF_1, MicroR, and MacroR respectively. However, by using ensemble techniques, significant performance gains have been observed in almost all example-based and label-based measures. Specifically, HLS outperforms the individual multi-label classification algorithms and the other EML techniques for example-based measures and also offers an improvement in classification performance for a selection of label-based measures.

	MicroP ↑	MicroF_1 ↑	MicroR ↑	MacroP ↑	MacroF_1 ↑	MacroR ↑
CLR	0.8273	0.6759	0.5721	0.8141	**0.8088**	0.2613
IBLR	0.7661	0.6886	0.6259	0.6190	0.6438	0.3277
MLKNN	0.7841	0.6874	0.6125	0.7910	0.6622	0.2857
RAkEL	0.7255	0.6829	0.6453	0.6490	0.6507	0.3317
HOMER	0.7039	0.6877	0.6737	0.6347	0.7110	0.3175
EML$_U$	0.6294	0.6918	**0.7688**	0.4677	0.6513	**0.4221**
EML$_I$	0.8904	0.6171	0.4728	0.9196	0.7637	0.2201
EML$_{MV}$	0.8011	0.7065	0.6322	0.8124	0.6816	0.2905
HLS(EML$_U$)	0.6623	0.7081	0.7626	0.4731	0.6206	0.3864
HLS(EML$_I$)	**0.9453**	0.6560	0.5031	**0.9549**	0.7956	0.1788
HLS(EML$_{MV}$)	0.8566	**0.7262**	0.6312	0.8363	0.6735	0.2333
EML$_{C_n L_k}$	0.7390	0.6911	0.6494	0.7493	0.6543	0.3258

Table 5.2: Comparison of bipartition-based ensembles for D_{Ra} using label-based measures.

Table 5.3 presents the comparison of score-based ensemble techniques with the individual classifiers for D_{Ra} using rank-based evaluation measures. First, when the individual classifiers are compared with each other, we find that CLR delivers the best performance for all of the measures. We find this interesting as CLR does not outperform the other multi-label classifiers for the example-based and label-based measures. Further investigation is needed to determine why this situation occurs. As before, the fusion of multi-label classifiers has improved the overall performance for rank-based measures. We observe that EML$_{Topk}$ makes an impact on the performance in comparison to the other EML techniques and multi-label classifiers. Moreover, it offers the best performance for all of the rank-based measures for D_{Ra}.

We find that the largest performance improvements are offered by HLS for bipartition-based ensembles and EML$_{Topk}$ for score-based ensembles, respectively. However, it is important to note that not all EML techniques offer improvements for D_{Ra}.

	AvgPrec ↑	Coverage ↓	RankLoss ↓	OneError ↓
CLR	0.8080	3.2160	0.0569	0.1940
IBLR	0.8067	3.4510	0.0649	0.2050
MLKNN	0.8094	3.4420	0.0637	0.1980
RAkEL	0.7747	5.5360	0.1129	0.2140
HOMER	0.7240	7.2170	0.1617	0.2440
EML$_{Max}$	0.7905	3.243	0.0592	0.2530
EML$_{Min}$	0.7584	7.181	0.1556	0.1860
EML$_{Mean}$	**0.8195**	3.206	0.0556	0.1840
EML$_{Topk}$	**0.8195**	**3.1690**	**0.0550**	**0.1820**

Table 5.3: Comparison of score-based ensembles for D_{Ra} using rank-based measures.

	HamLoss ↓	Accu. ↑	Recall ↑	F_1 ↑	Prec. ↑
CLR	0.0618	0.4641	0.4833	0.7390	0.8314
IBLR	0.0966	0.3811	0.4684	0.6723	0.5691
MLKNN	0.0750	0.3765	0.3965	0.6801	0.7418
RAkEL	0.0798	0.4500	0.5234	0.6932	0.6498
HOMER	0.0818	0.4900	0.5706	0.6984	0.6478
EML$_U$	0.1149	0.4647	0.7210	0.6297	0.5139
EML$_I$	0.0749	0.2671	0.2680	0.6865	0.9291
EML$_{MV}$	0.0646	0.4719	0.4934	0.7311	0.7843
HLS(EML$_U$)	0.0763	0.5146	**0.7908**	0.6753	0.5631
HLS(EML$_I$)	0.0485	0.4134	0.4134	**0.8517**	**0.9401**
HLS(EML$_{MV}$)	**0.0432**	**0.6064**	0.6273	0.8465	0.8201
EML$_{C_nL_k}$	0.0779	0.5053	0.5735	0.7078	0.6133

Table 5.4: Comparison of bipartition-based ensembles for D_{Ar} using example-based measures.

5.4.2 UniqueArtist Dataset

Table 5.4 shows the comparison of the bipartition-based ensembles with the individual multi-label classifiers for D_{Ar} using example-based evaluation measures. It is interesting to observe that for this data set, when the individual multi-label classifiers are compared with each other, we find a similar trend to the results reported in Table 5.1. That is, CLR and HOMER perform well for different evaluation measures. When we compare the individual classifiers using label-based measures, presented in Table 5.5, we observe that CLR performs well for MicroP, MicroF_1, MacroP, and MacroF_1, while HOMER and RAkEL perform well for MicroR, and MacroR, respectively. This is similar to the results reported in Table 5.2. As before, using the proposed ensemble of multi-label classifiers has improved the overall performance for example-based and label-based measures. That is, HLS(EML$_I$), HLS(EML$_U$), HLS(EML$_{MV}$), and EML$_U$ deliver improvements for example-based and label-based measures.

	MicroP ↑	MicroF_1 ↑	MicroR ↑	MacroP ↑	MacroF_1 ↑	MacroR ↑
CLR	0.8164	0.5995	0.4775	0.7903	0.7590	0.2987
IBLR	0.5159	0.4847	0.4603	0.3592	0.6399	0.3440
MLKNN	0.7229	0.5025	0.3892	0.6971	0.6920	0.2476
RAkEL	0.6127	0.5595	0.5182	0.5298	0.6665	0.3448
HOMER	0.5951	0.5759	0.5637	0.5184	0.6760	0.3723
EML$_U$	0.4514	0.5528	0.7194	0.3221	0.6463	**0.5386**
EML$_I$	0.9227	0.4034	0.2610	0.9137	0.6463	0.1638
EML$_{MV}$	0.7741	0.5946	0.4847	0.7402	0.7530	0.3058
HLS(EML$_U$)	0.4786	0.5744	**0.7241**	0.3342	0.6415	0.5217
HLS(EML$_I$)	**0.9401**	0.4962	0.3409	**0.9402**	0.6588	0.2014
HLS(EML$_{MV}$)	0.7979	**0.6338**	0.5276	0.7781	**0.8005**	0.3062
EML$_{C_nL_k}$	0.6133	0.5890	0.5671	0.5815	0.7141	0.3658

Table 5.5: Comparison of bipartition-based ensembles for D_{Ar} using label-based measures.

Table 5.6 presents the comparison of score-based ensembles with the individual classifiers for D_{Ar} using rank-based evaluation measures. When the individual multi-label classifiers are compared with each other, we find that CLR performs the best for all of the measures. However, the combination of multi-label classifiers offers improvements for different evaluation measures. For example, performance gains are observed for AvgPrec and OneError with an ensemble using the Top-k rule (EML$_{Topk}$). We also observe that CLR outperforms all of the proposed ensemble techniques for Coverage and RankLoss.

	AvgPrec ↑	Coverage ↓	RankLoss ↓	OneError ↓
CLR	0.7386	**4.1342**	**0.0895**	0.2526
IBLR	0.6218	5.7395	0.1412	0.4453
MLKNN	0.6907	4.8618	0.1100	0.2926
RAkEL	0.6579	8.0492	0.1951	0.2995
HOMER	0.6591	7.4418	0.1762	0.3089
EML$_{Max}$	0.6910	4.3029	0.0978	0.3695
EML$_{Min}$	0.6574	8.7550	0.2165	0.2534
EML$_{Mean}$	0.7372	4.2997	0.0923	0.2482
EML$_{Topk}$	**0.7430**	4.3258	0.0929	**0.2179**

Table 5.6: Comparison of score-based ensembles for D_{Ar} using rank-based measures.

It is interesting to observe that for this dataset the average classification performance of each multi-label classifier and ensemble is lower than the performance on D_{Ra}. This is analogous to the results presented in Chapter 4.

	HamLoss ↓	Accu. ↑	Recall ↑	F_1 ↑	Prec. ↑
CLR	0.0690	0.4240	0.4438	0.7726	0.7915
IBLR	0.0843	0.4327	0.5075	0.7008	0.6239
MLKNN	0.0765	0.3893	0.4105	0.7160	0.7189
RAkEL	0.0844	0.4364	0.5193	0.6830	0.6097
HOMER	0.0837	0.4964	0.5896	0.7077	0.6208
EML$_U$	0.1071	0.4795	0.7246	0.6465	0.5289
EML$_I$	0.0788	0.2661	0.2705	0.7156	0.8560
EML$_{MV}$	0.0687	0.4685	0.4932	0.7551	0.7376
HLS(EML$_U$)	0.0722	0.5309	**0.7950**	0.6898	0.5778
HLS(EML$_I$)	0.0505	0.4106	0.4106	**0.9001**	**0.8925**
HLS(EML$_{MV}$)	**0.0444**	**0.5891**	0.6084	0.8479	0.8034
EML$_{C_nL_k}$	0.0774	0.5115	0.5803	0.7078	0.6272

Table 5.7: Comparison of bipartition-based ensembles for D_{Al} using example-based measures.

5.4.3 UniqueAlbum Dataset

Finally, Table 5.7 shows the comparison of the bipartition-based ensembles with the individual multi-label classifiers for D_{Al} using example-based evaluation measures. It is easy to see that the results for the individual classifiers are similar to those observed for D_{Ra} and D_{Ar}. Specifically, we observe that CLR performs well for HamLoss, F_1 and Prec. while HOMER delivers the best performance for Accu. and Recall. Moreover, when we compare the individual classifiers using label-based measures, presented in Table 5.8, we find that CLR and HOMER perform well. However, using the proposed ensemble techniques improves classification performance. Once again, we observe that HLS outperforms the other ensembles and classifiers for the majority of the evaluation measures and delivers an increase in performance. For example, there is a 17.63% increase in MacroP for HLS(EML$_I$) when compared to CLR.

	MicroP ↑	MicroF_1 ↑	MicroR ↑	MacroP ↑	MacroF_1 ↑	MacroR ↑
CLR	0.7905	0.5522	0.4276	0.7645	0.6965	0.2408
IBLR	0.5977	0.5425	0.4976	0.3931	0.6262	0.3296
MLKNN	0.7206	0.5090	0.3960	0.6664	0.6498	0.2220
RAkEL	0.5942	0.5479	0.5094	0.4929	0.6156	0.3079
HOMER	0.5884	0.5801	0.5753	0.5282	0.6702	0.3394
EML$_U$	0.4788	0.5739	0.7180	0.3176	0.6098	**0.4924**
EML$_I$	0.8626	0.3923	0.2558	0.8269	0.5679	0.1400
EML$_{MV}$	0.7468	0.5822	0.4786	0.7075	0.7246	0.2630
HLS(EML$_U$)	0.4887	0.5821	**0.7227**	0.2983	0.5937	0.4760
HLS(EML$_I$)	**0.8925**	0.4613	0.3135	**0.8993**	0.6222	0.1615
HLS(EML$_{MV}$)	0.7852	**0.6102**	0.5004	0.7204	**0.7495**	0.2458
EML$_{C_nL_k}$	0.6272	0.5956	0.5672	0.6308	0.6484	0.3301

Table 5.8: Comparison of bipartition-based ensembles for D_{Al} using label-based measures.

Table 5.9 shows the comparison of score-based ensembles with the individual classifiers for D_{Al} using rank-based evaluation measures. With regard to the individual multi-label classifiers, we observe that CLR performs the best for all of the measures. This is analogous to the results presented for D_{Ra} and D_{Ar}. As before, improvements are observed by using the proposed ensemble of multi-label classifiers. Specifically, EML$_{Mean}$ offers the best performance of the proposed ensembles while EML$_{Topk}$ follows shortly behind. Furthermore, we observe that CLR outperforms the other proposed ensemble techniques for Coverage. However, we note that the difference is marginal.

	AvgPrec ↑	Coverage ↓	RankLoss ↓	OneError ↓
CLR	0.7276	**4.1210**	**0.0872**	0.3052
IBLR	0.6719	4.9301	0.1144	0.3977
MLKNN	0.7134	4.4022	0.0957	0.2916
RAkEL	0.6572	7.2234	0.1725	0.3275
HOMER	0.6341	8.1062	0.1988	0.3527
EML$_{Max}$	0.6892	4.2461	0.0939	0.3834
EML$_{Min}$	0.6506	8.7422	0.2130	0.2994
EML$_{Mean}$	**0.7388**	4.1885	**0.0872**	0.2658
EML$_{Topk}$	0.7369	4.1936	0.0882	**0.2576**

Table 5.9: Comparison of score-based ensembles for D_{Al} using rank-based measures.

5.5 Discussions

In summary, the ensemble techniques proposed not only offer additional performance improvements for multi-label genre classification, but also overcome limitations of individual multi-label classification algorithms, as aforementioned.

The results in our experiments show the merits of combining multi-label classification algorithms for music genre classification. For all three datasets, we observe improvements to classification performance when using heterogeneous ensemble techniques. However, this performance gain is at the expense of inherent computational cost as individual classifiers need to be trained and combined. Additionally, some of the proposed ensemble techniques provide no improvement and it would be interesting to further explore them. It is important to note that all of the processing and experiments performed in this work are conducted off-line. Therefore, we are not primarily concerned with the inherent computational cost incurred. In the following, we discuss some other related issues.

k	AvgPrec ↑	Coverage ↓	RankLoss ↓	OneError ↓
2	0.8118	3.2420	0.0575	0.1950
3	**0.8195**	**3.1690**	**0.0550**	0.1820
4	0.8194	3.2000	0.0559	**0.1760**
5	0.8190	**3.1690**	0.0553	0.1900

Table 5.10: Classification performance of EML_{Topk} for different k.

Top-k rule: The Top-k rule (EML_{Topk}) selects the top k largest scores and averages them, where k is a user-selected parameter. To determine this value, we perform a series of experiments on D_{Ra} and examine the performance as we adjust k. Table 5.10 shows the classification performance of EML_{Topk} where $k \in \{2, 3, 4, 5\}$ using the evaluation measures AvpPrec, Coverage, RankLoss, and OneError. We observe that EML_{Topk} demonstrates good achievement on D_{Ra} when $k = 3$ for the majority of evaluation measures. We note that when k is set to the number of classifiers in the ensemble, the result will be the same as the one obtained using the Mean rule.

	HamLoss ↓	Accu. ↑	Recall ↑	F_1 ↑	Prec. ↑
n=3, k=2	**0.0603**	**0.6255**	0.6725	**0.7799**	**0.7380**
n=4, k=2	0.0674	0.5799	0.6366	0.7422	0.6990
n=5, k=2	0.0681	0.5786	0.6328	0.7484	0.6950
n=5, k=3	0.0820	0.5339	0.7825	0.7034	0.5790
n=5, k=5	0.1483	0.3902	**0.8934**	0.5611	0.4016
n=10, k=3	0.1398	0.3006	0.4876	0.5230	0.3670

Table 5.11: Classification performance of $EML_{C_nL_k}$ for different n and k.

Score-based Label Selection: Similar to the Top-k rule, we perform a series of experiments on D_{Ra} to determine the values of n and k for $EML_{C_nL_k}$. Recall that in this ensemble, we first select the top n scores and then select the top k class labels accordingly. Table 5.11 shows the classification performance of $EML_{C_nL_k}$ using the evaluation measures HamLoss, Accu., Recall, F_1, and Prec. We observe that as both values of n and k increase, classification performance tends to decrease for the evaluation measures. We can see that $EML_{C_nL_k}$ performs best when $n = 3$ and $k = 2$. This could be plausibly explained by the label cardinality which is between 2 and 3 for each instance in our datasets, as shown in Table 4.1. It is important to note that we examine a wide range of values for n and k. However, due to space limitations, we only report a subset of the results.

Hierarchical Label Substitution: From the results presented above, we observe that Hierarchical Label Substitution (HLS) outperforms the other bipartition-based ensemble techniques for a majority of the evaluation measures. Moreover, significant performance gains are observed for all of the dataset when HLS is used with one of the proposed bipartition-based ensemble techniques. For example, there is a 14.51% increase in Prec. for D_{Ra} when compared to CLR. These results are attributed to the simplification of the label space by reducing the number of possible “overlapping” genres. We believe this ensemble produces a common set of high-level genres which is closer to the human experience.

5.6 Summary

To the best of our knowledge, this is the first study that aims to combine heterogeneous multi-label classification algorithms for music genre classification. Toward this end, we propose a set of ensemble techniques that not only improve upon individual multi-label classification algorithms, but also overcomes their limitations as aforementioned. In all of the datasets, we observe significant improvements to classification performance when using ensemble techniques. Specifically, we observe that HLS(EML_I) and HLS(EML_{MV}) performs the best out of the proposed bipartition-based ensemble techniques for a selection of evaluation measures. Furthermore, EML_{Mean} and EML_{Topk} demonstrate good performance from the proposed score-based ensemble techniques. However, we note that CLR consistently outperforms the other score-based ensembles for measures of Coverage and Ranking Loss.

In our future work, we plan to consider more multi-label classification algorithms and further investigate other ensemble techniques. From our study, we observe that some of the proposed ensemble techniques provide no improvements and it would be interesting to further explore them. A wealth of work exists surrounding the area of ensemble methods and may offer some insight. Furthermore, alternative segmentation and feature extraction algorithms need to be explored in the hope of further increasing multi-label classification performance.

Chapter 6

Conclusion

6.1 Summary

Music genre classification is a high-level task in Musical Information Retrieval (MIR). It has wide applications in managing music repositories, including music categorization, organization, and browsing. However, music genre classification is a subjective and ambiguous process, due to its inherent nature, historical background, cultural diversity, and personal experience. This not only shows that traditional single-label genre classification is inadequate but also asserts that multi-label music classification is needed, since a music piece can be assigned different genres, depending on the stand of individuals. In this thesis, we study multi-label music genre classification from perceptual and algorithmic perspectives.

In Chapter 2 we review previous related works. First, we introduce some discussions pertaining to music genres and human categorization. Relevant information is then presented on content-based audio analysis, such as feature extraction and segmentation techniques. After this, we dedicate our discussions to single-label classification algorithms common in MIR and previous works on the automatic classification of music. Additionally, multi-label classification algorithms are reviewed along with a set of evaluation measures for determining their predictive performance. Finally, previous work on the multi-label categorization of music, including tagging, emotion, etc., is presented.

In Chapter 3 we describe a series of perceptual experiments to explore the multi-genre-labeling behavior of individuals. Given a set of excerpts from a music piece, participants are asked to classify each excerpt and assign it to a single genre class. From this experiment we have found that participants consistently indicate that different portions of a music piece can be categorized into different genres. For this reason, we assert that genre classification approaches should consider multiple and compound genre labels. While previous studies explore the genre-labeling behavior of individuals, to the best of our knowledge, there have been no studies investigating perceptual multi-genre labeling.

Previous genre classification approaches are concerned with learning from a set of instances that are associated with a single label. However, as seen in Chapter 3, a music piece may belong to an unrestricted set of musical genres, making single-label classification problematic. In Chapter 4 we design a series of experiments to evaluate a set of multi-label classification algorithms. Moreover, issues pertaining to the creation of a multi-label dataset are explored. Experiment results are presented for a selection of feature extraction parameters. They not only support our speculation of employing multi-label learning algorithms for music genre classification but also demonstrate which algorithms are more suitable for this task.

In Chapter 5 we propose an ensemble of classifiers to further improve multi-label music genre classification. Our approach is to combine the predictive power of multiple classification algorithms to demonstrate performance gains. The results show the merits of combining multi-label classification algorithms. We believe that this is the first study that aims to combine multi-label approaches for music genre classification.

In summary, we address issues pertaining to the task of multi-label genre classification. We show through a series of perceptual experiments that there is a strong need for genre classification approaches to consider multiple compound genre labels. More efforts are needed to help deal with the ambiguity associated with classifying music using a single genre descriptor. Toward this end, we design a set of computation experiments investigating multi-label genre classification. In doing so, we identify a set of multi-label classification algorithms suitable for this task. Moreover, we propose a set of ensemble techniques for further improving multi-label genre classification.

6.2 Limitations

There are several limitations to the work presented in this thesis. The first is the limited size of the dataset used in the perceptual experiment in Chapter 3; some genres are represented with few examples. Although this is within the same range of other similar works', it is by no means statistically representative of all music.

A second limitation is that, to the best of our knowledge, no datasets exist specifically for multi-label genre classification. This makes it difficult to compare classification results. In addition, classification in Chapters 4 and 5 is performed on short excerpts of music. Although this is a standard approach in the MIR community, there are limitations to using short excepts of audio for classification as described in Chapter 3.

A third identified limitation is the set of measures used to evaluate classification performance in Chapters 4 and 5. No standardized set of measures exist in the literature. It is important to note that we use the most common set of evaluation measures for comparing multi-label genre classification performance.

Perhaps the most obvious limitation of this thesis is the assumption that music genre is an intrinsic attribute of a particular music piece. Although debate surrounds this issue, we believe that the members of a particular genre share certain characteristics, such as timbre, tempo, rhythm, etc., that can be used for categorization. However, this does not imply that genre is solely an intrinsic attribute of music. That is, they may also be founded on cultural extrinsic habits.

A final limitation is the set of features used to parameterize the audio. Several studies have shown a relationship between features and classification performance. That is, classification performance can be improved by using a set of "optimal" features. However, no standardized set of features exist, which makes it difficult to select a set which are best suited for classification.

6.3 Future Work

In the future we plan to further explore and analyze our results from the perceptual experiment. We hope to determine if an individual's predisposition to a given genre influences her/his classification result. In addition, we plan to experimentally analyze the influence of music expertise on the genre-labeling behavior of individuals.

Further investigation is needed into multi-label classification algorithms and parameters for music genre classification. Recall from Chapter 4 that each algorithm was trained using default parameters, e.g., the number of neighbors was set to 10 for ML-kNN and IBLR. It would be interesting to explore the influence of these parameters on classification performance. In addition, alternative segmentation and feature extraction algorithms need to be explored in the hope of further increasing multi-label classification performance.

We observe that classification performance can be influenced by the use of a texture window. Specifically, performance dramatically decreases for two of the datasets when a texture window of any size is applied. This is contrary to what is commonly reported in other studies. We plan to further investigate along this direction.

A great deal of additional research needs to be conducted with respect to combining multi-label classification algorithms. Alternative ensemble techniques for combining classifiers are needed. From our study, we observe that some of the proposed ensemble techniques provide no improvements. A wealth of work exists surrounding the area of ensemble methods and may offer some insight into this. Furthermore, we plan to conduct a series of experiments comparing the performance of ensemble techniques, presented in Chapter 5, on larger music datasets.

Bibliography

[1] J. J Aucouturier and F. Pachet. Representing musical genre: A state of the art. *Journal of New Music Research*, 32(1):83–93, 2003.

[2] J. J. Aucouturier and F. Pachet. Improving timbre similarity: How high is the sky? *Journal of Negative Results in Speech and Audio Sciences*, 1(1), 2004.

[3] J. G. Barbedo and A. Lopes. Automatic genre classification of musical signals. *EURASIP Journal on Applied Signal Processing*, 2007(1):157–157, 2007.

[4] L. Barrington, R. Oda, and G. Lanckriet. Smarter than genius? Human evaluation of music recommender systems. In *Proceedings of the International Conference on Music Information Retrieval*, pages 357–362, 2009.

[5] D. Baur, T. Langer, and A. Butz. Shades of music: Letting users discover sub-song similarities. In *Proceedings of the International Conference on Music Information Retrieval*, pages 111–115, 2009.

[6] C. R. Befus. Design and evaluation of dynamic feature-based segmentation on music. M.Sc thesis, University of Lethbridge, Canada, 2010.

[7] J. Bello and J. Pickens. A robust mid-level representation for harmonic content in music signals. In *Proceedings of the International Conference on Music Information Retrieval*, pages 304–311, 2005.

[8] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kgl. Aggregate features and adaboost for music classification. In *Machine Learning Journal: Special Issue on Machine Learning in Music*, volume 65, pages 2–3, 2006.

[9] D. Brackett. *Interpreting Popular Music.* Cambridge: Cambridge University Press, 1995.

[10] D. Brackett. *(In Search Of) Musical Meaning: Genres. Categories and Crossover.* London: Arnold, 2002.

[11] A. Chase. Music discriminations by carp. *Animal Learning & Behavior*, 29(4):336–353, 2001.

[12] N. V. Chawla and J. Sylvester. Exploiting diversity in ensembles: improving the performance on unbalanced datasets. In *Proceedings of the International Conference on Multiple Classifier Systems*, pages 397–406, 2007.

[13] L. Chen, P. Wright, and W. Nejdl. Improving music genre classification using collaborative tagging data. In *Proceedings of the International Conference on Web Search and Data Mining*, pages 84–93, 2009.

[14] W. Cheng and E. Hullermeier. Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*, 76(2-3):211–225, 2009.

[15] P. Cook, editor. *Music, cognition, and computerized sound: an introduction to psychoacoustics.* MIT Press, Cambridge, MA, USA, 1999.

[16] M. Cord and P. Cunningham. *Machine Learning Techniques for Multimedia.* Springer-Verlag, 2008.

[17] A. Craft, G. Wiggins, and T. Crawford. How many beans make five? the consensus problem in music-genre classification and a new evaluation method for single-genre categorisation systems. In *Proceedings of the International Conference on Music Information Retrieval*, pages 73–76, 2007.

[18] M. Crump. A principal components approach to the perception of musical style. Honours thesis, University of Lethbridge, Canada, 2002.

[19] F. Pachet Daniel and D. Cazaly. A taxonomy of musical genres. In *Proceedings RIAO of Content-Based Multimedia Information Access*, pages 1238–1245, 2000.

[20] F. Dellaert, T. Polzin, and A. Waibel. Recognizing emotion in speech. In *Proceedings of the International Conference on Spoken Language Processing*, pages 1970–1973, 1996.

[21] J. S. Downie. Music information retrieval evaluation exchange (MIREX). 2005.

[22] D. P. W. Ellis. Classifying music audio with timbral and chroma features. In *Proceedings of the International Conference on Music Information Retrieval*, pages 339–340, 2007.

[23] A. Eronen. *Signal Processing Methods for Audio Classification and Music Content Analysis.* PhD thesis, Tampere University of Technology, Tampere, Finland, June 2009.

[24] F. Fabbri. A theory of musical genres: Two applications. In *Popular Music Perspectives*, pages 52–81. Goteborg and Exeter: IASPM, 1981.

[25] J. Foote. Automatic audio segmentation using a measure of audio novelty. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 452–455, 2000.

[26] S. Frith. *Performing Rites: On the Value of Popular Music.* Cambridge, MA: Harvard University Press, 1996.

[27] J. Fürnkranz, E. Hüllermeier, E. L. Mencía, and K. Brinker. Multilabel classification via calibrated label ranking. *Machine Learning*, 73(2):133–153, 2008.

[28] J. Futrelle and J. S. Downie. Interdisciplinary communities and research issues in music information retrieval. In *Proceedings of the International Conference on Music Information Retrieval*, pages 121–131, 2002.

[29] R. O. Gjerdingen and D. Perrott. Scanning the dial: The rapid recognition of music genres. In *Journal of New Music Research*, volume 37, pages 93–100, 2008.

[30] M. Goto. A chorus section detection method for musical audio signals and its application to a music listening station. *IEEE Transactions on Audio, Speech & Language Processing*, 14(5):1783–94, 2006.

[31] E. Guaus. *Audio content processing for automatic music genre classification: descriptors, databases, and classifiers.* PhD thesis, Universitat Pompeu Fabra, 2009.

[32] P. Heymann, D. Ramage, and H. G. Molina. In *Proceedings of the International ACM Conference on Research and Development in Information Retrieval*, pages 531–538, 2008.

[33] L. Hofmann-Engl. Towards a cognitive model of melodic similarity. In *Proceedings of the International Conference on Music Information Retrieval*, pages 143–151, 2001.

[34] E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172(16-17):1897–1916, 2008.

[35] D. N. Jiang, L. Lu, H. J. Zhang, J. H. Tao, and L. H. Cai. Music type classification by spectral contrast feature. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 113–116, 2002.

[36] L.I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms.* Wiley-Interscience, 2004.

[37] T. Lambrou, P. Kudumakis, R. Speller, M. Sandler, and A. Linney. Classification of audio signals using statistical features on time and wavelet transform domains. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3621–3624, 1998.

[38] E. Law and L. von Ahn. Input-agreement: a new mechanism for collecting data using human computation games. In *Proceedings of the International Conference on Human Factors in Computing Systems*, pages 1197–1206, 2009.

[39] T. Li and M. Ogihara. Detecting emotion in music. In *Proceedings of the International Conference on Music Information Retrieval*, pages 239–240, 2003.

[40] T. Li, M. Ogihara, and Q. Li. A comparative study on content-based music genre classification. In *Proceedings of the International Conference on Research and development in information retrieval*, pages 282–289, 2003.

[41] T. Li and G. Tzanetakis. Factors in automatic musical genre classification of audio signals. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 143–146, 2003.

[42] S. Lippens, J. P. Martens, T. De Mulder, and G. Tzanetakis. A comparison of human and automatic musical genre classification. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 233–236, 2004.

[43] B. Logan. Mel frequency cepstral coefficients for music modeling. In *Proceedings of the International Conference on Music Information Retrieval*, 2000. n.p.

[44] H. Lukashevich, J. Abeer, C. Dittmar, and H. Grossmann. From multi-labeling to multi-domain-labeling: A novel two-dimensional approach to music genre classification. In *Proceedings of the International Conference on Music Information Retrieval*, pages 459–464, 2009.

[45] M. Mandel and D. Ellis. Song-level features and support vector machines for music classification. In *Proceedings of the International Conference on Music Information Retrieval*, pages 594–599, 2005.

[46] R. Mayer, J. Frank, and A. Rauber. Analytic comparison of audio feature sets using self-organising maps. In *Proceedings of ECDL Workshop on Exploring Musical Information Spaces*, pages 62–67, 2009.

[47] C. Mckay. *Automatic Music Classification with jMIR.* PhD thesis, McGill University, Montreal, Canada, 2010.

[48] C. McKay and I. Fujinaga. Musical genre classification: Is it worth pursuing and how can it be improved? In *Proceedings of the International Conference on Music Information Retrieval*, pages 101–106, 2006.

[49] A. Meng, P. Ahrendt, and J. Larsen. Improving music genre classification by short time feature integration. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 497–500, 2005.

[50] A. Meng and J. Shawe-Taylor. An investigation of feature models for music genre classification using the support vector classifier. In *Proceedings of the International Conference on Music Information Retrieval*, pages 604–609, 2005.

[51] S. R. Ness, A. Theocharis, G. Tzanetakis, and L. G. Martins. Improving automatic music tag annotation using stacked generalization of probabilistic svm outputs. In *Proceedings of the ACM International Conference on Multimedia*, pages 705–708, 2009.

[52] D. Rocchesso P. Polotti, editor. *Sound to Sense, Sense to Sound: A state of the art in Sound and Music Computing.* Logos Verlag, 2008.

[53] F. Pachet. Content management for electronic music distribution: The real issues. *Communications of the ACM*, 46(4):71–75, 2003.

[54] F. Pachet and D. Cazaly. A classification of musical genre. In *Proceedings RIAO Content-Based Multimedia Information Access Conference*, pages 1238–1245, 2000.

[55] F. Pachet and P. Roy. Improving multi-label analysis of music titles: A large-scale validation of the correction hypothesis. *IEEE Transactions on Audio, Speech & Language Processing*, 17(2):335–343, 2009.

[56] E. Pampalk, A. Flexer, and G. Widmer. Improvements of audio-based music similarity and genre classification. In *Proceedings of the International Conference on Music Information Retrieval*, pages 628–633, 2005.

[57] D. Pye. Content-based methods for the management of digital music. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 2437–2440, 2000.

[58] L. Rabiner and B. Juang. *Fundamentals of speech recognition.* Prentice-Hall, Inc., 1993.

[59] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. In *Machine Learning and Knowledge Discovery in Databases*, Lecture Notes in Computer Science, pages 254–269. Springer Berlin / Heidelberg, 2009.

[60] G. Reynolds, D. Barry, T. Burke, and E. Coyle. Towards a personal automatic music playlist generation algorithm: The need for contextual information. In *Proceedings of Audio Mostly Conference: Interaction with Sound*, pages 84–89, 2007.

[61] C. Sanden, C. R. Befus, and J. Zhang. Perception based multi-label genre classification on music data. In *Proceedings of the International Computer Music Conference*, pages 9–15, 2010.

[62] N. Scaringella, G. Zoia, and D. Mlynek. Automatic genre classification of music content: a survey. *IEEE Signal Processing Magazine*, 23(2):133–141, 2006.

[63] R. E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.

[64] E. Spyromitros, G. Tsoumakas, and I. Vlahavas. An empirical study of lazy multilabel classification algorithms. In *Proceedings of the Hellenic Conference on Artificial Intelligence*, pages 401–406. Springer-Verlag, 2008.

[65] M. Tahir, J. Kittler, K. Mikolajczyk, and F. Yan. Improving multilabel classification performance by using ensemble of multi-label classifiers. In *Multiple Classifier Systems*, volume 5997, chapter 2, pages 11–21. Springer Berlin Heidelberg, 2010.

[66] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas. Multilabel classification of music into emotions. In *Proceedings of the International Conference on Music Information Retrieval*, pages 325–330, 2008.

[67] G. Tsoumakas and I. Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2007.

[68] G. Tsoumakas, I. Katakis, and I. Vlahavas. Effective and efficient multilabel classification in domains with large number of labels. In *Proceedings of ECML/PKDD Workshop on Mining Multidimensional Data*, 2008. n.p.

[69] G. Tsoumakas, I. Katakis, and I. Vlahavas. Mining multi-label data. *Data Mining and Knowledge Discovery Handbook*, 2010.

[70] G. Tsoumakas, I. Katakis, and I. Vlahavas. Random k-labelsets for multi-label classification. *IEEE Transactions on Knowledge and Data Engineering*, 99(PrePrints), 2010.

[71] G. Tsoumakas, J. Vilcek, E. Spyromitros, and I. Vlahavas. Mulan: A Java library for multi-label learning. *Journal of Machine Learning Research*, 2010. Accepted for publication.

[72] G. Tsoumakas and I. Vlahavas. Random k-labelsets: An ensemble method for multilabel classification. In *Proceedings of the European Conference on Machine Learning*, pages 406–417, 2007.

[73] G. Tzanetakis and P. Cook. Multifeature audio segmentation for browsing and annotation. In *Proceedings Workshop Applications of Signal Processing to Audio and Acoustics*, pages 103–106, 1999.

[74] G. Tzanetakis and P. Cook. Audio information retrieval (AIR) tools. In *Proceedings of the International Conference on Music Information Retrieval*, 2000. n.p.

[75] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

[76] G. Tzanetakis, G. Essl, and P. Cook. Automatic musical genre classification of audio signals. In *Proceedings of the International Conference on Music Information Retrieval*, pages 293–302, 2001.

[77] A. Wang. An industrial-strength audio search algorithm. In *Proceedings of the International Conference on Music Information Retrieval*, pages 7–13, 2003.

[78] A. Wang. The shazam music recognition service. *Communications of the ACM*, 49(8):44–48, 2006.

[79] D. Wang, T. Li, and M. Ogihara. Are tags better than audio? the effect of joint use of tags and audio content features for artistic style clustering. In *Proceedings of the International Conference on Music Information Retrieval*, pages 57–62, 2010.

[80] F. Wang, X. Wang, B. Shao, T. Li, and M. Ogihara. Tag integrated multi-label music style classification with hypergraphs. In *Proceedings of the International Conference on Music Information Retrieval*, pages 363–368, 2010.

[81] K. West. *Novel Techniques for Audio Music Classification and Search.* PhD thesis, University of East Anglia, UK, September 2008.

[82] K. West and S. Cox. Finding an optimal segmentation for audio genre classification. In *Proceedings of the International Conference on Music Information Retrieval*, pages 680–685, 2005.

[83] K. West and Stephen Cox. Features and classifiers for the automatic classification of musical audio signals. In *Proceedings of the International Conference on Music Information Retrieval*, pages 531–537, 2004.

[84] A. Wieczorkowska, P. Synak, and R. Zbigniew. Multi-label classification of emotions in music. In *Intelligent Information Processing and Web Mining*, volume 35 of *Advances in Intelligent and Soft Computing*, pages 307–315. Springer Berlin / Heidelberg, 2006.

[85] C. Xu, N.C. Maddage, X. Shao, F. Cao, and Q. Tian. Musical genre classification using support vector machines. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 429–432, 2003.

[86] M-L. Zhang and Z-H. Zhou. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, July 2007.

[87] X. Zhang and Z. W. Ras. Analysis of sound features for music timbre recognition. In *Proceedings of the International Conference on Multimedia and Ubiquitous Engineering*, pages 3–8, 2007.