LEARNING SPECTRAL FILTERS FOR SINGLE- AND MULTI-LABEL CLASSIFICATION OF MUSICAL INSTRUMENTS

by

Patrick Joseph Donnelly

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

Doctor of Philosophy

in

Computer Science

MONTANA STATE UNIVERSITY
Bozeman, Montana

August, 2015

© COPYRIGHT

by

Patrick Joseph Donnelly

2015

All Rights Reserved

DEDICATION

To Ben, my wonderful and supportive husband.

ACKNOWLEDGEMENTS

First, I give a special thanks to my adviser and mentor Dr. John Sheppard. His constant support, guidance, and patience has guided me through this long arduous process beginning with my Masters coursework at Johns Hopkins University continuing through my doctoral defense at Montana State University. Dr. Sheppard has consistently, and at times, unrelentingly, challenged me to become a better writer, teacher, and researcher, and for that I am grateful.

I also would like to thank the members of my committee Dr. John Paxton, Dr. Brendan Mumey, Dr. Rob Maher, and Dr. Jason Bolte, for their guidance, patience, and their generous donation of their time. I thank Dr. Mark Greenwood for his help and guidance with our statistical experiments. I thank Ms. Kathryn Hollenback, Ms. Shelly Shroyer, and Ms. Jeannette Radcliffe for helping guide me through the maze of paperwork and administrivia during my time at Montana State University.

Lastly, I am most grateful for the support of my sisters, my friends, my colleagues at Montana State University, my lab-mates at the Numerical Intelligent Systems Laboratory, and most importantly, my ever patient husband Benjamin.

TABLE OF CONTENTS

TABLE OF CONTENTS – CONTINUED

TABLE OF CONTENTS – CONTINUED

TABLE OF CONTENTS – CONTINUED

TABLE OF CONTENTS – CONTINUED

LIST OF TABLES

LIST OF TABLES – CONTINUED

Table Page

LIST OF FIGURES

Figure Page

LIST OF ALGORITHMS

Algorithm Page

ABSTRACT

Musical instrument recognition is an important research task in music information retrieval. While many studies have explored the recognition of individual instruments, the field has only recently begun to explore the more difficult multi-label classification problem of identifying the musical instruments present in mixtures. This dissertation presents a novel method for feature extraction in multi-label instrument classification and makes important contributions to the domain of instrument classification and to the research area of multi-label classification.

In this work, we consider the largest collection of instrument samples in the literature. We examine 13 musical instruments common to four datasets. We consider multiple performers, multiple dynamic levels, and all possible musical pitches within the range of the instruments.

To the area of multi-label classification, we introduce a binary-relevance feature extraction scheme to couple with the common binary-relevance classification paradigm, allowing selection of features unique to each class label. We present a data-driven approach to learning areas of spectral prominence for each instrument and use these locations to guide our binary-relevance feature extraction. We use this approach to estimate source separation of our polyphonic mixtures.

We contribute the largest study of single- and multi-label classification in musical instrument literature and demonstrate that our results track with or improve upon the results of comparable approaches. In our solo instrument classification experiments, we provide the seminal use of Bayesian classifiers in the domain and demonstrate the utility of conditional dependencies between frequency- and time-based features for the instrument classification problem. For multi-label instrument classification, we explore the question of dataset bias in a cross-validation study controlled for dataset independence. Additionally, we present a comprehensive cross-dataset study and demonstrate the generalizability of our approach.

We consider the difficulty of the multi-label problem with regards to label density and cardinality and present experiments with a reduced label set, comparable to many studies in the literature, and demonstrate the efficacy of our system on this easier problem. Furthermore, we provide a comprehensive set of multi-label evaluation measures.

CHAPTER 1

INTRODUCTION

Music is diverse, culturally specific, and an inherently complex phenomenon. There are many dimensions and qualities to music including genre, style, mood, musical key, harmony, rhythm, loudness, spatial positioning, reverberation, instrumentation, and timbre [1]. In this age dominated by intelligent gadgets, such as smart phones, watches, and eye-wear, we increasingly rely on software to monitor, interpret, and react to our environment. Currently, music recommendation systems, such as Pandora or Spotify, are widely used to select the music to which we listen, and acoustic fingerprinting systems, such as Shazam, attempt to identify which music is playing. Some of the attributes of music sound correspond to physical aspects of the audio signal, such as rhythm, while others refer to more subjective interpretations of the sound, such as mood. Some of these qualities of music can be captured by meta-data extracted from album notes, reviews and commentary, or even by peer-recommendation systems. Many of these qualities of musical sound, however, are quite complex and do not easily map to physical attributes of sound.

The human brain is particularly adept at distinguishing between multiple musical instrument sounds. The perception of timbre allows humans to identify two different musical instruments, even if they play the same pitch, duration, and volume level. Humans can readily discern between different musical instruments under a variety of conditions including large and complex groupings of instruments, such as a large symphony orchestra, noisy environmental conditions, such as listening through headphones on the subway, or even following significant hearing loss. Researchers in the areas of psychoacoustic perception, neuroscience, psychophysics, statistical multidimensional scaling analysis, machine learning, and acoustic analysis of sound's physical characteristics, all suggest the perception of timbre is a complex multi-dimensional attribute that relies on both the spectra (frequency content) and the temporal features of the sound [2].

The identification of musical instruments in audio recordings is a frequently explored, yet unsolved, machine learning problem. Despite a number of experiments in the literature over the years, no single feature extraction scheme or learning approach has emerged as a definitive solution to this classification problem. The ability of a computer to learn to identify musical instruments is an important problem within the field of Music Information Retrieval (MIR), with high commercial value. For instance, companies such as Pandora or Amazon desire to index their music libraries automatically based on the musical instruments present in the recording, allowing search and retrieval by specific musical instrument. Timbre identification is also important to the ongoing research areas of musical genre categorization [3, 4, 5, 6], query by example [7, 8, 9], automatic score transcription [10, 11, 12, 13], score informed source separation [14, 15, 16, 17], musical audio annotation [18, 19, 20], score alignment [21, 22, 23, 24], musical content similarity and music recommendation systems [25, 26, 27, 28], and automatic musical accompaniment by a computer [21, 29, 30].

1.1 Motivation

Since the late 1970's [31], many researchers have attempted the automatic recognition of individual musical instruments in isolation. In these many years, numerous feature extraction schemes and classification algorithms have been proposed and tested. Because these studies differ in data sources, feature extraction schemes, feature content, feature dimensionality, experimental design, and classification algorithms, rarely are these studies directly comparable (see Section 3.1 for a review). Since these approaches are often sensitive to the feature input and classification algorithms; they most often do not generalize between different datasets. Furthermore, because the field lacks readily available or standardized datasets, most studies cannot be replicated.

Recent work in the field has shifted to the more complex case of identifying the instruments present in polyphonic mixtures. This is a more difficult problem because the spectral content of the constituent tones can overlap in time and frequency (see Section 2.1.4 for a discussion of timbre), often causing interference between spectral components. Most of the approaches developed to recognize of individual instruments are not scalable to the more complex case of polyphonic music signals [32].

In this dissertation, we present a feature extraction scheme designed for extensibility to multilabel classification of polyphonic mixtures. In the next sections, we identify and discuss the design criteria needed to extend an approach to multi-label classification of polyphonic mixtures.

1.1.1 Scalability

The goal of identifying instruments present in polyphonic mixtures is a multi-label classification problem. One possible approach is to train models on all possible mixtures of instruments [33]. This method, however, suffers from the combinatorial explosion of labels needed to classify, and it is not feasible to train models with every possible combination of instruments.

The task of polyphonic identification lends itself naturally to binary-relevance (BR) classification, a decomposition approach in which a single classifier is trained for each class label (see Section 2.2.2.3). In this work, we use a binary-relevance feature extraction scheme and a binary-relevance classification scheme, requiring a separate model for each instrument responsible for identifying the presence of that instrument in a signal, independent of any other instruments that may be present. The strength of BR classification for polyphonic mixture identification is that it only requires training models on single instrument data yet allows extensibility to unseen combinations of those instruments, requiring only an additional binary instrument for each new instrument trained.

1.1.2 Generalizability

Arguing that many approaches cannot generalize to new data, Livshin and Rodet [34] identified five different musical instrument datasets that shared a common subset of seven instruments and performed cross database evaluations. The authors received results ranging from 20% accuracy in the worst case up to 63% in the best, with an average accuracy of 42%. These results demonstrate the poor generalization abilities of common classification techniques across databases.

Their results indicate that many techniques overfit the training dataset and the features sets used do not sufficiently capture general qualities of instrumental timbre. Many of these approaches have significant limitations (see [35] for a discussion). These limitations include small datasets containing very few examples, the use of a small set of hand-picked instruments, often selected from different instrument families, datasets containing examples of only a single instrument or performer, differences in dynamic levels of the musical notes recorded, and differences in recording procedures, equipment, and levels. Additionally, many studies report low accuracy results, despite testing only a small number of instruments [35]. Furthermore, very few studies have addressed validation of approach against different datasets [34].

To demonstrate the generalizability of our binary-relevance feature extraction approach, in this work we compare cross dataset performance on four datasets and a large set of thirteen instruments. These datasets feature multiple performers, instrument manufacturers, dynamic levels, and cover the range of each musical instrument (see Chapter 5).

1.1.3 Practicality

Timbre perception and recognition rely on both the harmonic content of the musical partials and the fine timing of the envelope of each harmonic. The attack of an instrument sound and the differences in the fine-timing of the envelopes of individual partials are of particular importance in both perception and algorithmic recognition of timbre. Many classification approaches exploit this valuable information, as does the human auditory system [32].

The literature has most often focused on single instrument classification in which the datasets contain examples of the entire length of an instrument sample, including the attack and the decay, as does this work. However, any approach that relies on capturing the time differences of the instrument's temporal envelope may not be practical and properly capture situations in which signals contain only part of an instrument's note and to circumstances in which multiple notes overlap in time. A practical system cannot expect sterile notes in which the attack, sustain, and release portions of the signals are intact, but rather given an arbitrary time frame that contains some portion of a note in time. In this dissertation, we ignore any timing information and instead focus on identifying locations of harmonic content most useful in discriminating between musical instruments.

1.2 Contributions

This dissertation contributes a number of important advancements to the research area of musical instrument classification. They are listed as follows:

- In this dissertation, we provide a comprehensive review of the literature of both single-label and multi-label classification of musical instruments. The last thorough review of the monophonic classification literature appeared in 2008 [36] and this work is the first thorough literature review in the domain of classification of polyphonic instrument mixtures.
- In our study of single instrument classification, we present the seminal use of Bayesian networks for the classification of musical instruments. Additionally, we demonstrate the utility of conditional dependencies between features in the time and frequency domains, a novel contribution to the domain. Lastly, we present a novel topology, the grid-augmented naïve Bayes model, for modeling sequential conditional dependencies in two dimensions.
- We present an approach for binary-relevance feature extraction for use with binary-relevance classifiers for multi-label classification. Since binary-relevance classification requires training a separate classifier for each class, we argue that each classifier need not cover the same feature space, which is the common practice in multi-label classification. A binary-relevance feature extraction scheme does require an extra feature extraction for each instrument class, but provides the benefit of customizing each binary classifier to features that best represent the class it must learn. In this work, we use a data-driven clustering approach to learn locations to each instrument that best represent areas of spectral prominence in the instrument's signature. The combination of a binary-relevance

feature set and a binary-relevance classifier allows an estimation of source separation in the spectral domain in order to classify each contributing source separately.

- We demonstrate our approach to musical instrument classification on four large datasets with thirteen instruments in common. Studies in the musical instrument classification domain often overfit models to single datasets, often with very few instrument classes or few examples per instrument. In this work, we use several and diverse datasets to test the generalizability of our approach to capture features that represent qualities of the instruments' timbre. Three of the datasets are those most commonly reported in the literature and we contribute a fourth dataset, a novel use of this public collection for the research domain. We consider a large set of 13 instruments, the set of instruments common to the four datasets. We consider examples from multiple performers, covering multiple musical dynamic levels and notes covering the entire playable range of the instruments.
- We describe a novel data-driven approach to learn areas of spectral prominence for each instrument to guide feature-extraction. We use these instrument signatures to estimate source separation, attempting to minimize overlapping musical partials. These signatures guide our binary-feature extraction scheme in both single- and multi-label classification. This is the first study to consider a different feature space for each binary classifier in the domain of multi-label instrument classification.
- In this work, we demonstrate our our approach to single- and multi-label classification with a battery of experiments across four datasets mentioned above. We demonstrate that our approach achieves comparable efficacy across the datasets in both the single- and multi-label problems, indicating our approach captures

features that describe the timbre of the instrument, rather than cues from the recording procedure.

- To test their approaches to multi-label classification, researchers frequently generate test sets by mixing multiple solo examples into polyphonic mixtures, as we do in this work. Although the training set and the test set are not exactly the same, this relation raises the question of dataset bias. This study is the first to explore this question through a set of cross-validation experiments, permitting a comparison of these dataset independent results to the full dataset classification results.
- Many of the approaches to musical instrument classification, both of single and multiple instruments, do not generalize well beyond the dataset used for training. In this work, we demonstrate generalizability of the approach with cross-dataset experiments, training on one dataset and testing on a different dataset. In this work, we provide the most comprehensive cross-data study in both the single- and multi-label instrument classification literature.
- Although studies in general multi-label classification often report many of the different metrics for evaluation [37], studies in classification of polyphonic mixtures rarely report more than one single evaluation metric. We evaluate our multi-label experiments with a comprehensive set of evaluation measures for multi-label classification, including example-, label- and rank-based multi-label evaluation metrics. Although these metrics are widely used in other domains of multi-label classification [37], no musical instrument classification study has provided a systematic multi-label evaluation of their performance. We seek to align the domain of instrument classification with the standards of evaluation common elsewhere in multi-label classification. Additionally, we present an

extension to an existing multi-label evaluation measure, which is discussed in Section 2.2.3.

- In our multi-label classification experiments we consider the cardinality and density of the problem. Although these measures are often found in other multi-label domains, this dissertation presents the first discussion of difficulty of the multi-label problem, considering these measures in the instrument classification domain. Additionally, we demonstrate the efficacy of our system on a multi-label problem with a reduced cardinality, one commonly reported in the literature.

1.3 Organization

This section describes the organization of this dissertation, providing a brief overview of the focus of each remaining chapter.

In Chapter 2, we discuss many of the background topics and terminology necessary to understand the approaches given in this work. First, we review musical terminology and discuss the properties of sound. Next, we discuss the concept of timbre, which is integral to this work, and introduce methods for spectral analysis. Then we review supervised classification and the area of multi-label classification, including the binary-relevant approach for classification. This is followed by a discussion of the evaluation metrics for multi-label classification we use in the experiments. Finally, we review the algorithms for clustering and classification that are referenced in subsequent chapters.

In Chapter 3, we review the literature of the many varied approaches to musical instrument classification. We begin with a review of solo instrument classification and progress to the cases of multi-label classification of polyphonic mixtures.

In Chapter, 4, we present our study on classification of single instruments. This early work heavily influenced our subsequent approaches, including our choice of datasets, our cross-dataset experimental design, and our feature extraction technique.

In Chapter 5, we discuss the data used in this work, including the original sources, the list of musical instruments on which this work focuses, the pre-processing steps, the signal-processing steps, and the division of the data into datasets for classification experiments.

In Chapter 6, we present our approach to learn an instrument-specific feature extraction schemes for use as a binary-relevance feature extraction schemes in classification experiments. From datasets of recordings of solo instruments, we learn areas of spectral prominence for each instrument and use these areas as spectral filters to guide the feature extraction process. In this chapter, we validate this approach by using the feature extraction scheme learned from one dataset for extraction of features from a different dataset, demonstrating the generalizability of this approach.

In Chapter 7, we demonstrate the use of the feature extraction schemes learned in the previous chapter and discuss the extensibility of this approach, showing its use of this procedure on the multi-label polyphonic data.

In Chapter 8, we present experiments designed to test the generalizability of this approach by training models with data from one source but testing these models on an entirely different source.

In Chapter 9, we demonstrate the extensibility of this approach to multi-label data, with classification experiments on mixtures of two, three, and four instruments.

In Chapter 10, we conclude with a summary of results and contributions and discuss areas for future research.

In Appendix A, we provide a detailed walk-through example of the learning process described in Chapter 6.

In Appendix B, we provide results of classification result by instruments for the polyphonic mixture experiments given in Chapter 9.

CHAPTER 2

BACKGROUND

In this chapter, we provided background needed to understand the terminology, procedures and algorithms discussed in this dissertation. We begin by discussing relevant musical terminology and discuss the algorithm we use in processing our audio signals. In Section 2.2, we discuss the machine learning discipline of supervised classification, with special emphasis on the binary-relevance approach to multi-label classification. Section 2.2.3 presents the evaluation metrics we use to evaluate the multi-label classification experiments discussed in Chapter 9. Section 2.3 provides explanations of the algorithms explored in Chapters 6, 8, and 9. Lastly, we review Mel-Frequency Cepstral Coefficients, a feature space commonly used in single-label instrument classification, in Section 2.4

2.1 Musical Sound

This dissertation examines the automatic identification of musical instrument sounds from audio signals. In order to understand the approach taken in this work, one must first understand what constitutes a musical sound and which properties make a sound recognizable and distinguishable from other types of sounds.

The acoustic or physical properties of sound are those concerned with the production, transmission, and propagation of sonic waves. The psychoacoustic or perceptual properties of sound are those related to auditory perception and interpretation of sound. The aural recognition of sound is dependent on both the physics of sound –

how the ear receives the waveform, and psychoacoustic properties – how the brain interprets the sound.

A musical sound is characterized by four perceptual attributes: pitch, loudness, duration, timbre. The first three attributes have clear physical counterparts of frequency, amplitude, and time. The fourth attribute of sound, timbre, is the quality of sound that allows the human brain to distinguish between different musical instruments. Timbre is less well understood and does not have a direct mapping to a single physical characteristic.

In this section, we discuss the various properties of sound, both the acoustic properties of the sonic waveform and the psychoacoustic corollary of the perception of each property. This section introduces many important terms, which are referenced throughout this dissertation. The concept of timbre, which is integral to this work, is introduced and discussed along with an explanation of the Fast Fourier Transform used for spectral decomposition of the audio signals.

2.1.1 Pitch

Pitch is the perceptual quality of sound that allows us to distinguish between different music notes of the musical scale. For example, when one plays two consecutive notes on the piano, the human brain registers that there is a difference between the sounds and the relative direction of the change, either increasing or decreasing [38].

Pitch is a subjective measure that allows the ordering of sounds on a frequency-related scale. The differences between musical pitches are described by music intervals. The octave is a natural musical interval that occurs at the doubling of the frequency of a note. Western music divides the octave into 12 equal steps, the notes of a chromatic scale. Other cultures subdivide the octave in differing ways.

The simplest of sounds, a sine tone, contains energy of only a single frequency. When a musical instrument plays a note, however, we still perceive only a single tone, even though the sound contains energy at a number of different frequencies. The Fourier series describes a periodic waveform as a sum of a single fundamental frequency and the other harmonic components. For musical notes, the fundamental frequency, f_0, is the lowest frequency present in a periodic waveform. The remaining harmonic content are known as overtones and the set of the fundamental frequency together with overtones are known as partials.

2.1.2 Volume

The volume, or loudness, of a sound is the perceptual correlate of the physical strength, or amplitude, of the waveform. The American National Standards Institute (ANSI) defines loudness as "that intensive attribute of auditory sensation in terms of which sounds may be ordered on a scale extending from soft to loud" [39]. The perception of loudness is complex and depends on several factors, including the acoustic energy of the sound wave, the frequency content of the sound, and the duration of the sound.

In musical notation, notes are marked by a subjective and relativistic measure known as the dynamic level. The dynamic level indicates the volume level of which the note should be produced by a performer, relative to the dynamic levels of the other notes. Western music uses a system ranging from quiet to loud using the terms: pianissimo (*pp*), piano (*p*), mezzopiano (*mp*), mezzoforte (*mf*), forte (*f*), and fortissimo (*ff*). Several of the datasets we examine in this dissertation contain examples of instruments playing at three different dynamic levels (see Chapter 5).

2.1.3 Duration

The next principal parameter of sound is the duration, which correlates to the physical property of time, or length of the waveform. The duration of a sound affects the perception of the other properties of sound. The perception of loudness increases with duration of the sound. Additionally, musical notes often contain subtle fluctuations in pitch over time, and when the effect is intentionally exaggerated, it is technique known as vibrato. Lastly, the relative strengths of the individual harmonics vary over time and their relative intensities may differ between the attack, sustain, and decay portions of the musical note.

In this dissertation, we consider only the frequency content within one single time window, one second in length. In future work, we will discuss extending our approach to a temporal model in order to capture the fluctuation of amplitudes of frequency components over time.

2.1.4 Timbre

When a musical instrument plays a note, we perceive both a musical pitch and the instrument playing that note. Timbre, sometimes known as tone color, is the psychoacoustic property of sound that allows the human brain to distinguish readily between the same note, despite being played on two different instruments. The primary musical pitch we perceive is the first partial, known as the fundamental frequency. When an instrument produces an overtone series that matches the sequence of integer multiples of the fundamental frequency, the instrument is known as harmonic. With the exception of some drums and bells, such as chimes, most orchestral instruments are harmonic.

The perception of timbre depends on the harmonics (spectra) and the fine timing (envelope) of each harmonic constituent (partial) of the musical signal [40]. ANSI defines timbre as:

> *Timbre is that attribute of auditory sensation in terms of which a listener can judge that two sounds similarly presented and having the same loudness and pitch are dissimilar. Timbre depends primarily upon the spectrum of the stimulus, but it also depends upon the waveform, the sound pressure, the frequency location of the spectrum, and the temporal characteristics of the stimulus* [41].

Timbre is the least well understood property of sound. The other three psychoacoustic qualities of sound – pitch, volume, and duration – have direct correlates to the physical properties of the waveform. Timbre, however, cannot be so easily defined, and subsequently, it cannot be measured. Timbre is a multi-dimensional entity, dependent on the other attributes of sound.

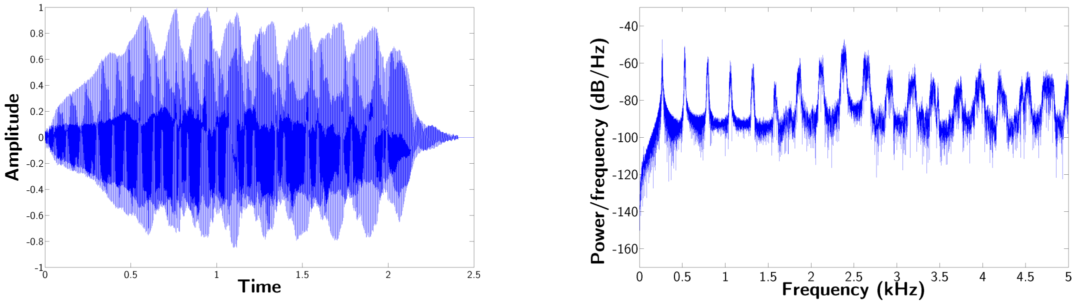
2.1.4.1 Consonance and Dissonance. The difference between any two pitches forms a musical interval. Although the interpretations vary across cultures, consonance and dissonance refer to the subjective perceived pleasantness or unpleasantness, respectively, of a musical interval. When two notes are play simultaneously, the harmonic partials of individual tones are interleaved in both the frequency and time domains. In some cases, partials from more than one instrument can overlap and cause destructive or constructive interference. Since the consonance and dissonance of an interval are tied to the ratio of the frequencies of the pitches, consonant intervals exhibit a greater coincidence of partials [42]. This is particularly the case in musically consonant intervals, such as the octave or the fifth, because two notes forming these intervals have the frequencies of many of their harmonic partials in common.

The more instruments playing at the same time increases the likelihood of source interference between contributing notes.

2.1.4.2 Perception of Timbre. Although the perception of timbre has been studied by the psychoacoustic community for many years, no consensus of definition or measure has been reached. In 1977, John Grey carried out a perception study to better understood how humans compare the similarities and differences between timbre. He presented subjects with pairs of tones that differed in timbre and asked the subjects to rate the dissimilarity between the sounds. Using the statistical technique of multidimensional perceptual scaling, Grey visualized the similarities and differences between musical instruments according to several dimensions. Grey reported that the three dimensions most important for the human perception of timbre is the distribution of energy among the harmonics, the presence of low-energy high frequency energy, and the fluctuation of spectral energy of the harmonics over time [43].

Although no single definition or measurement has emerged, the work of Grey and others make clear that the human perception of timbre is most reliant on the differences in the spectral content between sounds. Since the goal of this dissertation is the development of an approach for the machine recognition of timbre from audio waveforms, we must calculate and analyze the spectral content of a sound. This transformation from the waveform, or time domain, to the frequency domain is accomplished with a Fourier transformation.

2.1.4.3 Fourier Transform. The Fourier Theorem states that any complex signal can be described as the potentially infinite sum of a series of sine or cosine terms. This theorem allows a complex signal to be decomposed into a series of sine waves that differ in frequency, amplitude, and phase.



(a) Waveform of a Violin note (261.5 Hz)) (b) Spectra of a Violin note (261.5 Hz)

Figure 2.1: The FFT transforms a signal from the waveform in the time domain (left) to the spectra in the frequency domain (right).

The Fast Fourier Transform (FFT) is an efficient implementation of a Discrete Fourier Transform used to estimate the power spectra of discrete non-periodic signals [44]. The FFT algorithm is an important tool in science and engineering with applications in spectral analysis, image compression, partial differential equation solving methods, multiplication of polynomials, among many others. Fourier analysis yields a spectral decomposition, revealing periodicities in the input signal including the relative strengths of any periodic components, allowing a transformation of a signal as a function of time to a function of frequency. An example of a waveform of a musical note and the subsequent transformation to the spectra using an FFT is shown in Figure 2.1.

Although frequency is a linear concept, measured in Hertz, pitch is perceived logarithmically. In Western music notation, the semitone scale is a logarithmic mapping from frequency to pitch. FFT analysis, on the other hand, provides a uniform resolution across a linear Hertz scale. This has the implication that the high frequency partials are measured with a much higher resolution than lower frequency partials. The typical human ear can nominally hear sounds in the range 20 Hz to 20,000 Hz. The piccolo, the orchestral instrument with the highest range, has a top note with a pitch just under 4000 Hz. For contrast, the average male speaking voice ranges between 85 Hz and 155 Hz and a typical female ranges from 165 Hz to 255 Hz [45]. Therefore, if the analysis of the FFT spectra necessitates a high resolution, a time sample of sufficient length must be provided in order to provide adequate resolution in the lower frequency range.

The computational complexity of the FFT algorithm is loglinear in time: $\mathcal{O}(n \cdot \log n)$, where n is the number of samples in the audio file. Because of the linear resolution of the result, FFT analysis provides a constant trade-off between the length of input sample in time and the frequency resolution of the result. If the input sample is too short, the frequency resolution in the analysis of the lower partials will be too poor, and this is the frequency range that matters most for speech and music. For consistency in resolution, throughout this work, we consider a single time window of one second for all our sound examples, rather than considering multiple overlapping, but small, windows sliding over time.

2.2 Classification

Supervised learning is one of the most widely explored paradigms in the field of machine learning. Given a training set of examples with known class labels, supervised techniques attempt learn a model that outputs a class label for previously unseen examples. Classification is the task of assigning each example with one or more labels from a finite number of discrete categories. Regression, on the other hand, is the task of assigning one or more output variables with continuous values. This section discusses the various approaches to supervised classification, which we use in this dissertation.

2.2.1 Multi-class Classification

In classification tasks, $\mathcal{X}$ represents the input space, $\mathcal{L}$ represents the output space, and the task of learning is to derive a function $g : \mathcal{X} \rightarrow \mathcal{L}$. Each instance $\mathbf{x} \in \mathcal{X}$ is represented as a vector $\mathbf{x} = [x_1, \ldots, x_M]$ of length M, in which each x_i is a feature describing some property of the instance. Single-label classification describes the assignment of instance $\mathbf{x}$ to a label ℓ from a set of disjoint labels $\mathcal{L}$ in which $\ell \in \mathcal{L}$.

Tasks discerning only two label classes, $|\mathcal{L}| = 2$, are known as binary classification problems. In multi-class classification, the set of possible classification labels is larger, $|\mathcal{L}| > 2$. Traditional approaches limit the assignment of each example to only a single label, embedded with the assumption that each example can be associated with only a single concept or semantic meaning.

2.2.2 Multi-label Classification

In multi-label classification, on the other hand, an example to classify is relevant to more than one class label. Therefore each example $\mathbf{x} \in \mathcal{X}$ is assigned a set of labels $\mathcal{Y}$ by the classifier, where $\mathcal{Y} \subseteq \mathcal{L}$. Specifically, the classifier g, for a given instance $\mathbf{x} \in \mathcal{X}$, yields

$$g(\mathbf{x}) = [g_1(\mathbf{x}), g_2(\mathbf{x}), \cdots, g_k(\mathbf{x})]^T \tag{2.1}$$

where $g_j(\mathbf{x})(j = 1, \cdots, k)$ is either 0 or 1, indicating association of $\mathbf{x}$ with the jth label [46].

Multi-label classification is increasingly popular in many real-world domains, such as text categorization [47, 48], image annotation [49, 50], bioinformatics [51] medical diagnosis [52], classification of film genre [53], and the classification of emotions in music [54].

There are two broad approaches to multi-label classification: either transform the algorithm to fit the data or transform the data to fit the algorithm. Algorithm adaptation approaches adapt and extend algorithms designed for single-label classification tasks to handle multi-label classification tasks directly. In the problem transformation approach, on the other hand, a multi-label problem is transformed into a set of single-label classification problems so that existing algorithms for single-label classification can be applied without the need to change the algorithm.

2.2.2.1 Algorithm Adaptation. The first approach, known as algorithm adaptation, consists of adapting existing algorithms to return a set of labels instead of a single label. A number of algorithms have been adapted for use in multi-label classification including: decision trees [55], adaboost [56], k-nearest neighbor [57, 58], ranking support vector machines [59, 60], associative rule learners [61], neural networks [62, 51], as well as several probabilistic approaches [63, 64]. Many of these approaches, suffer limitations in scalability as the number of training examples, dimensionality of the feature space, or the number of class labels increases [65] and from correlations between labels, much like in multi-class counterparts.

2.2.2.2 Problem Transformation. The second approach, problem transformation methods, describes the transformation of a multi-label classification problem into a single label multi-class problem. One common approach is known as the Label Powerset (LP) method [66]. In this approach, all possible sets of labels are enumerated and used as if they were individual labels, which results in a combinatorial explosion in the number of labels [67]. For this reason, this approach is highly undesirable if $|\mathcal{L}|$ is a large number. The complexity of the LP approach is $\mathcal{O}(\min(n, 2^k))$, where n is the number of training examples, and k is the total number of class labels before the

Table 2.1: Illustration of multi-label data and problem transformation methods for multi-label classification.

Example	**Label Set**
$\mathbf{x}_1$	$\{\ell_3\}$
$\mathbf{x}_2$	$\{\ell_1, \ell_2\}$
$\mathbf{x}_3$	$\{\ell_2, \ell_4\}$
$\mathbf{x}_4$	$\{\ell_1, \ell_3, \ell_4\}$

(a) Example of multi-label data

Example	**Multiclass Label**
$\mathbf{x}_1$	$\{\ell_3\}$
$\mathbf{x}_2$	$\{\ell_{1,2}\}$
$\mathbf{x}_3$	$\{\ell_{2,4}\}$
$\mathbf{x}_4$	$\{\ell_{1,3,4}\}$

(b) Transformation of Fig. 2.1a using Label Powerset method

Ex.	**Label**
$\mathbf{x}_1$	$\neg\ell_1$
$\mathbf{x}_2$	ℓ_1
$\mathbf{x}_3$	$\neg\ell_1$
$\mathbf{x}_4$	ℓ_1

Ex.	**Label**
$\mathbf{x}_1$	$\neg\ell_2$
$\mathbf{x}_2$	ℓ_2
$\mathbf{x}_3$	ℓ_2
$\mathbf{x}_4$	$\neg\ell_2$

Ex.	**Label**
$\mathbf{x}_1$	ℓ_3
$\mathbf{x}_2$	$\neg\ell_3$
$\mathbf{x}_3$	$\neg\ell_3$
$\mathbf{x}_4$	ℓ_3

Ex.	**Label**
$\mathbf{x}_1$	$\neg\ell_4$
$\mathbf{x}_2$	$\neg\ell_4$
$\mathbf{x}_3$	ℓ_4
$\mathbf{x}_4$	ℓ_4

(c) Transformation of Fig. 2.1a using Binary-Relevance method transformation [65]. When the number of labels k becomes large, it adds significant complexity to the training process. Another problem is when the data contains a small n, and there may not be enough examples of particular label combinations to learn meaningful relationships. An example of problem transformation is shown in Table 2.1b.

Furthermore, since this approach requires training on all possible combinations of class labels, it is not readily extensible to handle new previously unseen class labels. Adding a new class label requires creating new examples that feature all combinations of the new label with all other labels, in addition to retraining the models.

2.2.2.3 Binary-Relevance Classification. The most common method to the problem transformation approach to multi-label classification is known as the Binary-Relevance (BR) method. The BR method learns $|\mathcal{L}|$ different binary classifiers, one for each possible label. Each binary classifier is trained to distinguish the examples in a single class from the examples in all remaining classes. When classifying a new example, all $|\mathcal{L}|$ classifiers are run and the labels associated with the classifiers that output the label *true* are added to $\mathcal{Y}$. This is known as the one-vs-all (OVA) scheme. More specifically, each binary classifier C_l is responsible for predicting the true/false association for each single label $\ell \in \mathcal{L}$. The final label set $\mathcal{Y}$ is the set of labels from all classifiers that returned true [68]. The complexity of the BR approach is linear in the number of models $\mathcal{O}(m)$ where $m = |\mathcal{L}|$ class labels. When adding a new class label to a BR approach, an additional model must be trained to handle to the new class label, and the existing models must be updated to be able to differentiate from the the new class label. An example of problem transformation is shown in Table 2.1c. This dissertation will use the BR approach to multi-label classification.

2.2.3 Evaluation Measures

In multi-class classification, performance of a classifier is most commonly evaluated by accuracy, error-rate, or information retrieval measures of precision, recall, and F_1-measure. Since multi-label classification yields a set of predicted labels, it presents new challenges in evaluating classifier performance. Subsequently, there are many ways to evaluate multi-label classifiers including strict measures that reward complete accuracy of the predicted label set to other measures that reward partial correctness.

The measures of cardinality and density, discussed below, are frequently used to discuss the difficulty of the problem in the multi-label classification literature, However, in the domain of multi-label classification, these measures are not discussed, nor do researchers comprehensively report the common multi-label evaluation measures described in this section. In this dissertation, we seek to align the musical instrument classification literature with the measures common to other domains of multi-label classification.

In this section, let there be n examples to classify. Assume there are $q = |\mathcal{L}|$ possible labels in $\mathcal{L}$. For each instance i, let Y_i be the set of true labels where $Y_i \subseteq \mathcal{L}$. The set Z_i is the set of labels predicted by the classifier for instance i.

2.2.3.1 Number of Labels. The number of labels $q = |\mathcal{L}|$ affects the difficulty of any multi-label classification problem. There are two measures that quantify the label space of the dataset: label cardinality and label density. Cardinality measures the mean number of labels of the instances in the dataset. Density, on the other hand, is the mean of the number of labels of the instances, normalized by the number of labels q [69, 70].

$$\text{Cardinality} = \frac{1}{n}\sum_{i=1}^{n}|Y_i| \tag{2.2}$$

$$\text{Density} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i|}{q} \tag{2.3}$$

The difficulty of a multi-label classification problem is tied to the label cardinality and density. Density is a value with the range $(0.0, 1.0)$ and the difficulty of the problem lessens as the density approaches 1.0. In a comprehensive empirical study, Bernardini *et al.* demonstrated a strong correlation between label density and multiple multi-label evaluation metrics and this correlation was stronger than the correlation between label cardinality and multiple multi-label evaluation metrics. The authors conclude "the lower the density and the higher the cardinality, the more difficult the multi-label learning process" [70].

2.2.3.2 Example-based Measures. Example-based measures examine the average difference between the actual and predicted sets of labels, averaged over all examples.

- Subset accuracy measures the fraction of correctly classified labels [63]. This metric is analogous to classification accuracy in multi-class classification.

  $$\text{Subset-Accuracy} = \frac{1}{n}\sum_{i=1}^{n}I(Z_i = Y_i) \tag{2.4}$$

  where $I(\cdot)$ is an indicator function. Subset accuracy, or exact match ratio, is a very strict measure, especially as the number of labels q is high, because it discounts a partially correct labeling as incorrect.

- Hamming loss measures the fraction of misclassified instance/label assignments, capturing both the cases of an incorrectly classified label as well as when a relevance label is missed [56].

$$\text{Hamming-Loss} = \frac{1}{n}\sum_{i=1}^{n}\frac{I(Z_i \,\Delta\, Y_i)}{q} \tag{2.5}$$

  where Δ indicates the symmetric difference between the two label sets. Hamming loss is normalized by the number of examples n and the number of labels q. The smaller the value of the Hamming loss, the better the performance of the classifier.

- Accuracy is the fraction of correctly predicted labels to the total number of predicted and actual labels, averaged over all classified instances. In other words, accuracy is the Jaccard similarity of the relevant and the true label sets.

$$\text{Accuracy} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \tag{2.6}$$

The common information retrieval measures have been extended for multi-label classification [60]. These measures account for partial correctness as compared to subset accuracy which does not.

- Precision is the ratio of correctly predicted labels to the total number of actual labels, averaged over all classified instances.

$$\text{Precision} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Z_i|} \tag{2.7}$$

- Recall is the ratio of correctly predicted labels to the total number of predicted labels, averaged over all classified instances.

$$\text{Recall} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i|} \tag{2.8}$$

- F_1 is a weighted measure of precision and recall, averaged over all classified instances.

$$F_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2 \cdot \text{Precision}(i) \cdot \text{Recall}(i)}{\text{Precision}(i) + \text{Recall}(i)} = \frac{1}{n}\sum_{i=1}^{n}\frac{2 \cdot |Y_i \cap Z_i|}{|Y_i| + |Z_i|} \tag{2.9}$$

2.2.3.3 Label-based Measures. Label based measures evaluate each label individually, and then aggregate over all labels. Any evaluation measure appropriate for a binary classifier can be used a label-based metric, such as the information retrieval measures given above. There are two approaches to calculating label-based measures: macro-averaged and micro-averaged approaches. Macro-averaging approaches compute the score on individual class labels and then average of the number of classes. Micro-averaging, on the other hand, approaches compute totals over all instances and all class labels, before calculating the measure.

- Macro-averaging approaches

$$\text{Precision}_{macro} = \frac{1}{q}\sum_{\ell=1}^{q}\frac{\sum_{i=1}^{n} Y_i^\ell Z_i^\ell}{\sum_{i=1}^{n} Z_i^\ell} \tag{2.10}$$

$$\text{Recall}_{macro} = \frac{1}{q}\sum_{\ell=1}^{q}\frac{\sum_{i=1}^{n} Y_i^\ell Z_i^\ell}{\sum_{i=1}^{n} Y_i^\ell} \tag{2.11}$$

$$\text{Macro-}F_1 = \frac{1}{q}\sum_{\ell=1}^{q}\frac{2 \cdot \sum_{i=1}^{n} Y_i^\ell Z_i^\ell}{\sum_{i=1}^{n} Y_i^\ell + \sum_{i=1}^{n} Z_i^\ell} \tag{2.12}$$

- Micro-averaging approaches

$$\text{Precision}_{micro} = \frac{\sum_{\ell=1}^{q}\sum_{i=1}^{n} Y_i^\ell Z_i^\ell}{\sum_{\ell=1}^{q}\sum_{i=1}^{n} Z_i^\ell} \tag{2.13}$$

$$\text{Recall}_{micro} = \frac{\sum_{\ell=1}^{q}\sum_{i=1}^{n} Y_i^\ell Z_i^\ell}{\sum_{\ell=1}^{q}\sum_{i=1}^{n} Y_i^\ell} \tag{2.14}$$

$$\text{Micro-}F_1 = \frac{2 \cdot \sum_{\ell=1}^{q}\sum_{i=1}^{n} Y_i^\ell Z_i^\ell}{\sum_{\ell=1}^{q}\sum_{i=1}^{n} Y_i^\ell + \sum_{\ell=1}^{q}\sum_{i=1}^{n} Z_i^\ell} \tag{2.15}$$

where

$$Y_i^\ell = \begin{cases} 1 & \text{if the } i\text{th example has true label } \ell \\ 0 & \text{otherwise} \end{cases}$$

and

$$Z_i^\ell = \begin{cases} 1 & \text{if the } i\text{th example is predicted with label } \ell \\ 0 & \text{otherwise.} \end{cases}$$

2.2.3.4 Rank-based Measures. Some multilabel classification approaches, such as the binary-relevance approach used in this dissertation, are able to learn rankings of predicted labels. The function $\text{rank}_i(\ell)$ returns the ranking of label ℓ for instance i, a value between $[1, q]$ in which 1 is the top ranked label and q is the last ranked label [37].

- One-error measures how often the top-most ranked label is not in the set of true labels of the examples.

$$\text{One-Error} = \frac{1}{n}\sum_{i=1}^{n} I(\operatorname*{argmin}_{\ell\in\mathcal{L}} \text{rank}_i(\ell) \notin Y_i) \tag{2.16}$$

- Coverage measures how far down the list of ranked labels, on average, is needed to go to find all true labels of the instance. This metric determines how many false positives must be considered in order to find all true positives.

$$\text{Coverage} = \frac{1}{n}\sum_{i=1}^{n} I(\max_{\ell\in Y_i} \text{rank}_i(\ell) - 1) \tag{2.17}$$

  In this dissertation, we consider an extension to the Coverage measure, Coverage_j, in which $j \in [1, q]$ represents the depth of coverage in the list of rankings. For example, Coverage_1 represents the average depth down the list of ranked labels to find the first label. Likewise, we consider Coverage_2, Coverage_3, and Coverage_4 for the depths of two, three, and four, respectively. These measures are useful to determine the partial coverage of j labels. The case in which j is equal to the label cardinality represents the traditional Coverage measure given by Equation 2.17.

- Ranking loss measures the number of times, on average, that an irrelevant label is ranked higher than a relevance label.

$$\text{RankingLoss} = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{|Y_i|}|\{(\ell_a, \ell_b) : \text{rank}_i(\ell_a) > \text{rank}_i(\ell_b), (\ell_a, \ell_b) \in Y_i \times \overline{Y_i}\}| \tag{2.18}$$

  where $\overline{Y_i}$ is the complement of the set of predicted labels Y_i.

- Average precision measures the fraction of relevant labels ranked above each relevant label, averaging over the number of relevant labels.

Table 2.2: List of multi-label classification measures with best and worst values.

Type	**Metric**	**Ideal Score**	**Worst Score**
Example-based	Subset Accuracy	1.0	0.0
	Hamming Loss	0.0	1.0
	Accuracy	1.0	0.0
	Precision	1.0	0.0
	Recall	1.0	0.0
	F_1 Measure	1.0	0.0
Label-based	Macro-Precision	1.0	0.0
	Macro-Recall	1.0	0.0
	Macro-F_1	1.0	0.0
	Micro-Precision	1.0	0.0
	Micro-Recall	1.0	0.0
	Micro-F_1	1.0	0.0
Rank-based	One-Error	0.0	1.0
	Coverage	0.0	$q-1$
	Ranking Loss	0.0	$\|Y_i\|$
	Average Precision	1.0	0.0

$$\text{AveragePrecision} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|Y_i|}\sum_{\ell \in Y_i}\frac{|\{\ell' \in Y_i : \text{rank}_i(\ell') \leq \text{rank}_i(\ell)\}|}{\text{rank}_i(\ell)} \tag{2.19}$$

Table 2.2 provides a list of the ideal score and worst scores for each of the above measures.

2.3 Algorithms

In this section, we discuss the algorithms we use in this work. We use the k-means algorithm (Section 2.3.1) to learn regions of prominent spectral energy for each instrument, presented in Chapter 6. In our classification experiments, given in

Chapter 8, We evaluate the efficacy of our proposed feature extraction approach with classification experiments using the k-nearest neighbor algorithm (Section 2.3.2).

2.3.1 k-Means Clustering

Formally, the k-means algorithm considers n data points in d-dimensional space, $\mathbf{X} = x_i$, i = 1, . . . , n, and clusters these points into a set of k clusters $c_j \in \mathbf{C}$ where $j = 1, \ldots, k$ [71]. At each iteration, the algorithm assigns an example to a cluster by minimizing the distortion of each data point to the nearest cluster center. For each cluster c_k with mean μ_k, the distortion is defined as:

$$\text{Distortion}_k = \sum_{x_i \in c_k} ||x_i - \mu_k||^2 \tag{2.20}$$

The k-means algorithm greedily minimizes the total distortion across the k clusters:

$$\text{Distortion} = \sum_{k=1}^{K} \sum_{x_i \in c_k} ||x_i - \mu_k||^2 \tag{2.21}$$

The k-means algorithm begins by randomly assigning all the data across the k clusters, where k is determined using a predefined criterion. For each cluster c_k, calculate the mean μ_k. The total distortion of the cluster assignment is calculated according to Equation 2.21. The data points are then reassigned to the cluster with the nearest mean as to minimize Equation 2.20. The process is repeated iteratively until the total error converges, which occurs when no data points are reassigned and the cluster means do not change in value.

Because k-means is a greedy algorithm, minimizing the total error of the cluster assignment, it converges to a local minimum. For well separated clusters, k-means has been shown to converge to the global optimum with high probability [72]. The choice of k is critical to the partitioning, and many approaches for determining k have been proposed, including techniques to vary k between iterations [73].

2.3.2 k-Nearest Neighbor

The k-nearest neighbor (k-NN) algorithm is a common lazy, or instance-based classification algorithm in which a previously unknown example is classified with the most common class among its k nearest neighbors, where k is a small positive integer. A neighbor is determined by the application of some distance metric $D(\cdot,\cdot)$, such as Euclidean distance, in d-dimensional feature space [74].

Formally, let $\mathbf{X}$ be a space of points where each feature vector $\mathbf{f} \in \mathbf{X}$ is defined as $\mathbf{f} = \langle \{f^1, \ldots, f^d\}; c \rangle$, where c is the class label, and $\mathbf{X}_{tr} \subset \mathbf{X}$ be a set of training examples. For a unknown query example $\mathbf{f_q} \in \mathbf{X} - \mathbf{X}_{tr}$. find an example $\mathbf{f_r} \in \mathbf{X}_{tr}$ such that $\forall\ \mathbf{f_x} \in \mathbf{X}_{tr}$, $\mathbf{f_x} \neq \mathbf{f_r}$, $D(\mathbf{f_q}, \mathbf{f_r}) < D(\mathbf{f_q}, \mathbf{f_x})$ and return the class label c_r associated with example $\mathbf{f_r}$ [75].

k-NN is popular because it is easy to implement, robust to noisy training data, and often effective when given sufficiently large training sets. However, as an instance-based learning algorithm, general k-NN requires retaining all training instances in memory in order to classify a new instance. This can lead to expensive computations, memory limitations, and slow running times. Additionally, k-NN suffers from bias to the value of k, is sensitive to irrelevant attributes, and does not scale well as the number of classes increases [76].

2.3.3 Support Vector Machine

The support vector machine (SVM) is a discriminant-based method for classification [77], regression [78], or ranking learning [79]. In recent years, SVMs have been successful employed in many domains, including: bioinformatics [80], computer vision [81], numerical optimization [82], text categorization [83], fault diagnosis [84], time-series forecasting [85], and event-detection [86].

The SVM algorithm constructs a hyperplane in high dimensional space that represents the largest margin separating two classes of data. The SVM is defined as the hyperplane $\mathbf{w}^\top \cdot \Phi(\mathbf{f}) - b = 0$ that solves the following quadratic programming problem:

$$\text{minimize} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + C \cdot \sum_i \xi_i \right\} \tag{2.22}$$

subject to:

$$y(\mathbf{w}^\top \cdot \Phi(\mathbf{f}) - b) \geq 1 - \xi_i,\ \xi_i \geq 0 \tag{2.23}$$

where $\mathbf{f}$ is a vector of features, $\mathbf{w}^\top$ is the discriminant vector, C is a regularizing coefficient, ξ_i is a slack variable, b is the bias offset, y is the class label such that $y \in \{-1, +1\}$, and the kernel function $K(\mathbf{f_i}, \mathbf{f_j}) = \Phi(\mathbf{f_i})^\top \cdot \Phi(\mathbf{f_j})$ is the inner product of the basis function [87].

The traditional form of the SVM is a binary classifier in with the output of the classifier is either -1 or 1. To support multiclass problems, the SVM is often implemented as a series of ‘one-versus-all’ binary classifiers in which multiple binary classifiers are coupled pairwise [88].

When the kernel function $K(\mathbf{f}) = \mathbf{f}$, the SVM is a linear classifier. When the kernel is a non-linear function, such as a polynomial, the features are projected into a higher order space. This allows the algorithm to fit the maximum margin hyperplane in the transformed feature space, which is no longer linear in the original space [89].

Support vector machines are popular because SVMs have relatively few parameters to adjust, good generalization across many domains and datasets [90], ability to map to non-linear feature space using a kernel, and robustness to large errors concerning

Table 2.3: Three common SVM kernel functions

Linear:	$K(\mathbf{f_i}, \mathbf{f_j}) = \mathbf{f_i} \cdot \mathbf{f_j}$, where $r \in \mathbb{R}$,
Polynomial:	$K(\mathbf{f_i}, \mathbf{f_j}) = (\mathbf{f_i} \cdot \mathbf{f_j} + b)^\delta$, where $b \in \mathbb{N}$,
Gaussian:	$K(\mathbf{f_i}, \mathbf{f_j}) = \exp(-\frac{\|\mathbf{f_i} - \mathbf{f_j})\|^2}{2\sigma^2})$, where $\sigma > 0$

only a small portion of the dataset as well as robustness to small errors affecting the whole dataset [91]. Furthermore since the optimality problem is convex, SVMs return a single solution as opposed to neural network which may return different solutions for different local minima [92]. Lastly, unlike k-NN and other online techniques, SVM techniques produce a model that can be used off-line.

On the other hand, SVMs do suffer a few disadvantages. The results of a learned model are often difficult to interpret without visualization tools [92]. As a binary method, SVM requires adaptation in order to be applied to multiclass problems, which can potentially add significant overhead [88], and the technique pairwise comparison of binary classifiers tends to obscure final class probabilities. Furthermore, selecting the kernel most appropriate to the data may require either expert knowledge of the data domain or extensively empirical testing [93].

2.3.4 Bayesian Networks

Bayesian networks are probabilistic graphical models that are comprised of random variables, represented as nodes, and their conditional independencies, represented as a directed edges. The joint probability of the variables represented in the directed, acyclic graph can be calculated as the product of the individual probabilities of each variable, conditioned on the node's parent variables. The Bayesian classifier with observed variables is defined as:

$$\text{classify}(\mathbf{f}) = \operatorname*{argmax}_{c \in C} P(c) \prod_{f \in \mathbf{f}} P(f \mid \text{parent}(f)) \tag{2.24}$$

where $P(c)$ is the prior probability of class c and $P(f \mid \text{parent}(f))$ is the conditional probability of feature f given the values of the variable's parents. The classifier finds the class label which has the highest probability of explaining the values of the feature vector [94].

2.4 Mel-Frequency Cepstral Coefficients

Mel-Frequency Cepstral Coefficients (MFCC) are features dominant in speech recognition and speaker identification tasks [95]. MFCCs are a nonlinear spectrum-of-a-spectrum that result from a cosine transform of the logarithm of a log power spectrum on a nonlinear mel scale of frequency.

The Mel scale was introduced by Stevens, Volkman and Newman in 1937 and is named after the word melody. The Mel scale is a subjective scale that relates perceived pitch to to a measured frequency, constructed by perceptional distances between musical pitches as determined by human subjects [96]. This scale reflects the fact that human perception of frequency is non-linear and that humans are less sensitive at higher frequencies, such as those above 1 kHz.

The formula to convert from a frequency f in Hertz to Mel m is [97]:

$$m = 2595 \cdot \log_{10}(1 + \frac{f}{700}) \tag{2.25}$$

To calculate the MFCCs, the first step is to divide the signal into short overlapping time frames, each multiplied by a smoothing windowing function. The discrete Fourier transform (DCT) is applied to each time frame to transform to the spectral domain. A Mel-spaced filterbank is multiplied by the spectra, which spaces the filters approximately linearly below 1 kHz and logarithmically above 1 kHz. A set of 26 triangular filters is most commonly used. Within each filter, the amplitude coefficients are summed, resulting in 26 values that indicate the energy in the filter bank. The log of the square magnitude is taken for each of the 26 values. Lastly, take the DCT of the 26 log-energies in the Mel-filtered spectrum to yield 26 cepstral coefficients. Speech recognition system frequently retain only the first 12 or 13 cepstral coefficients, but the instrument classification literature frequently retains all of them.

MFCCs are a commonly reported feature space for the classification of single instruments and are used by many of the studies reported in Section 3.1. We compare our single instrument approach with results using MFCCs as features, presented in Chapter 4. MFCCs, however, are summary statistics of the spectra audio signal and are not useful in polyphonic classification with some estimation of source separation, although some multi-label classification algorithm approaches have attempted to use MFCCs as features (see Section 3.2.1.3). For polyphonic signals, MFCCs often capture nearby spectral peaks under the same filter even when they originate from different sources. Our approach outlined in this work attempts to estimate source separation between instruments . We discuss extending our approach to combine our source separation techniques with MFCCs as future work.

CHAPTER 3

RELATED WORK

3.1 Monophonic Classification

Initial investigations in the task of musical instrument recognition focused on the identification of solo instruments. Although there have been a number of studies recognizing musical instruments playing isolated notes, no dominant learning strategy nor feature extraction technique has emerged.

These studies have explored various spectral, temporal, and cepstral features for instrument recognition (see [36] for a review). MFCCs are commonly used in both speech processing and music classification [98]. A variety of supervised classification techniques have been explored, including k-nearest neighbors, decision trees, support vector machines, linear discriminant analysis, Gaussian mixture models, Bayesian networks, and neural networks (see [99] for a review).

3.1.1 Nonparametric Methods

3.1.1.1 k-Nearest Neighbors. In 1995, Kaminskyj and Materka explored music instrument recognition using a nearest neighbor classifier. They extracted short-term RMS energy envelopes from their sound sources, transformed the features using Principle Component Analysis (PCA), and tested their approach with a single-neighbor classifier. On a set of four instruments (piano, marimba, guitar, accordion), restricted to a single octave but covering five dynamic levels, the authors achieved 98% accuracy [100]. In subsequent work, Kaminiskyj classified 19 instruments covering three octaves playing only a single dynamic level with 93% accuracy using k-NN and six spectral features [101].

In a series of experiments in the late 1990's, Fujinaga employed k-NN to classify a set of 23 musical instruments with 50% accuracy using spectral features. When he used timing information from the spectral envelope, classification accuracy increased to 64% [102, 103, 104]. Agostini reported 80% classification accuracy using k-NN on a dataset of 17 instruments and 66% accuracy on a dataset of 27 instruments (see Table 3.1) [105].

Using a k-NN classifier and 43 spectral and temporal features, Eronen and Klapuri achieved 80% recognition on a dataset of 1498 samples covering 30 instruments in 2000 [106]. However, in subsequent research, the authors used four combined datasets, totaling 5286 samples covering 16 instruments but achieved only 35% accuracy [107]. Another study demonstrated the ability of a k-NN classifier to outperform a decision tree classifier and a discriminant analysis on a set of nine drum sounds [108].

Martin and Kim compared a standard k-NN classifier against a hierarchical extension to the k-NN classifier. Using a set of 31 spectral and temporal features on a dataset of 14 musical instruments, the authors observed the hierarchical k-NN achieved 67% accuracy compared to 61% accuracy of the non-hierarchical k-NN version [109, 110]. In 2003, Peeters also compared k-NN and hierarchical k-NN classifier on a large dataset of 27 instruments. The flat k-NN (k=10) classifier achieved 54% accuracy and was outperformed by the hierarchical k-NN classifier, which achieved 64% classification accuracy [111].

Livshin and Rodet created an k-NN classifer to operate on a feature set reduced by Linear Discriminate Analysis (LDA). Working from a dataset of continuous recordings covering seven instruments, the authors achieved 88% accuracy for solo instruments. Furthermore, the authors report that their classification system, which was developed for single instrument recognition, is able to identify the dominant instrument in a two instrument mixture at rates greater that chance [112, 113]. In subsequent extensions of their work, the authors examined the importance of non-harmonic “noise” in musical instrument classification. Beginning with a dataset of 5000 recordings covering 10 instruments, the authors created synthetic copies of their dataset that contained only energy at the harmonic partials of the original signal. To accomplish this they performed Fourier analysis and extracted the harmonic partials while discarding the inharmonic components. Using only the harmonic information, the authors used additive synthesis to re-synthesize the audio signals. The authors used a feature selection algorithm to greedily select the ten best features from a set of 62 common spectral and temporal features. The authors used a k-NN classifier to compare the original signals against the harmonic-only synthetic copies, achieved 94% accuracy on the original dataset but only 90% accuracy on the synthesized dataset, demonstrating the utility of non-harmonic information when classified harmonic instruments [114, 115].

In a recent study, Jiang *et al.* explored the sensitivity of classifiers to selected feature sets. Using subsets of the MPEG7 audio descriptors, the authors demonstrated that k-NN is more sensitive to a selected feature set than a decision tree. On a dataset of 2762 examples covering 26 instruments playing middle C, the authors achieved results ranging from 47% up to 99% accuracy depending on the subset of MPEG7 features selected [116].

3.1.1.2 Decision Trees. One study claimed to show the ability of binary decision trees to classify musical instrument sounds from a dataset of seven instruments. However the study did not vary their features, compare their approach to other algorithms, nor give specific results [117]. Herrera *et al.* found a k-NN classifier to significantly outperform both C4.5 and Partial Decision Tree (PaRT) classifiers on a dataset of

Table 3.1: Instrument classification results of Agostini *et. al* [105].

Instruments	SVM	k-NN	QDA
17	80.2	73.5	77.2
20	78.5	74.5	75.0
27	69.7	65.7	68.5
Family	77.6	76.2	80.8

drum sounds [99]. Contemporaneously, Peeters demonstrated both a Gaussian classifier and k-NN classifier significantly outperforming a decision tree on a set of 27 instruments [111]. More recently, Jiang *et al.* observed that a decision tree (94% accuracy) was outperformed by a k-NN classifier (98% accuracy) on a set of 26 instruments [116]. For these reasons, decision trees have not become a popular approach for musical instrument classification.

3.1.2 Discrimination Methods

3.1.2.1 Support Vector Machines. In a seminal study using SVM, Marques and Moreno classified 200 milliseconds of recorded audio for eight musical instruments, using 16 Mel-frequency cepstral coefficients (MFCC) as features. The authors achieved 70% accuracy using a 'one versus all' multi-class SVM with a polynomial kernel [118].

In 2003, a study demonstrated the ability of SVMs to outperform k-NN on the task of musical instrument identification. Agostini *et al.* tested several algorithms, using a set of nine spectral features and classifying over three different sets of instruments (see Table 3.1) [105].

On a set of 10 instruments and selecting from among 150 features, Essid *et al.* achieved 87% accuracy using an SVM with a radial basis function kernel and a pairwise classification strategy, outperforming the 82% accuracy of a GMM [119, 120]. In subsequent work, the authors examined the impact of the integration of temporal information in the feature set. Examining features based on 20 frames of audio over a dataset of eight instruments, the authors achieved between 79 and 82% classification accuracy using an SVM [121].

Sturm *et al.* explored classification of musical instruments using various MFCC-derived features and a pairwise SVM classifiers. The authors achieved 80% accuracy on a dataset of seven instruments using standard MFCC and 84% accuracy when incorporating MFCCs features over consecutive time frames [122, 123].

Ligges and Krey investigated musical instrument classification with an SVM classifier considering four different kernel functions: linear, polynomial, Gaussian, and radial basis function (RBF). Over their large dataset of 38 instruments and using MFCC features, the authors achieved 81% accuracy with the polynomial kernel ($d = 2$) which outperformed the other kernels as well as several other baseline algorithms [124, 125]. Another study also demonstrated an SVM with a quadratic kernel outperforming linear, RBF, and other polynomial kernels with $d > 2$ [126].

A recent study explored the efficacy of the SVM on the family identification task for a dataset that included non-Western instruments, achieving 87% accuracy on a set of 8 instrument families covering both Western and Chinese instruments [127]. Other recent investigations of SVM classification of musical instruments have focused on feature extraction techniques [126, 128, 129], feature dimensionality reduction [130], SVM parameter optimization [131], or the classification of non-harmonic instruments [132, 133, 134] and non-Western instruments [127, 135].

3.1.2.2 Discriminant Analysis. On a dataset of 27 instrument and using a small set of spectral features, Agostini *et al.* achieved 77.2% accuracy using quadratic discriminant analysis (QDA). The authors also tested instrument family discrimination (e.g., strings, woodwinds) and achieved 80.8% accuracy using QDA (see Table 3.1), which outperformed SVM and k-NN [105]. Herrera *et al.* attempted classification of drum sounds using Canonical Discriminant Analysis (CDA), which achieved 83% accuracy but was outperformed by a k-NN classifier at 90% [99].

A recent study attempted to classify both Western and ethnomusicological instruments into broad taxonomic categories using Linear Discriminant Analysis (LDA), achieving 80% accuracy [136]. Other studies have attempted to develop decision boundaries between specific pairs of instruments, such as discrimination between flute and saxophone [137] or piano and guitar [138].

3.1.2.3 Higher-Order Statistical Methods . Dubnov *et al.* sought to use differing aperiodic, nonlinear fluctuations in harmonics as a method to to discriminate between sound types. In a series of investigations, the authors analyzed the waveforms of sound sources using higher-order statistical methods, such as kurtosis and skewness. They then applied a maximum-liklihood (ML) classifier to discriminate between musical instrument families (i.e., string, brass) [139, 140, 141, 142]. Unfortunately the authors did not attempt discrimination between individual musical instruments.

3.1.2.4 Non-Negative Matrix Factorization . Benetos *et al.* attempted musical instrument classification using non-negative matrix factorization (NMF) on a small dataset of six instruments. The authors achieved 95% accuracy with their NMF technique, which was outperformed by a GMM and HMM [143, 144, 145]. Another study created excitation-filter models for musical instruments using NMF to learn the excitation basis functions and weights. On a small dataset, the authors reported accuracy ranging from 60% to 80% over the five instruments [146].

3.1.3 Bayesian Methods

3.1.3.1 Naïve Bayes. In an early study, Brown extracted 18 cepstral coefficients following a constant Q-transform on each sound. Using the k-means algorithm to cluster the features, she extracted the cluster densities for use as probabilities in a Bayesian classifier. Over a small set of short oboe and saxophone sounds, she achieved 85% average accuracy [147]. In subsequent work, Brown *et al.* applied this technique using a variety of features, such as cepstral coefficients, bin-to-bin differences of the constant-Q coefficients, and autocorrelation coefficients. The authors achieved 74% to 84% accuracy on a set of four instruments: oboe, saxophone, clarinet, and flute [148].

Martin and Kim modified a tree augmented naïve Bayes (TAN) classifier, adding context-dependent feature selection and beam search, to estimate maximum likelihood of an instrument class for a given a feature set. On a dataset of 14 musical instruments, using a set of 31 common spectral and temporal features, this modified TAN classifier outperformed a k-NN classifier [109, 110].

Kitahara *et al.* examined the effectiveness of naïve Bayes on a feature set subject to dimensionality reduction. Using 129 features, normalized by the example's fundamental frequency f_0, the authors applied both PCA and LDA on the features extracted for each example from a dataset covering 19 musical instruments. The naïve Bayes classifier outperformed a k-NN classifier when the feature space was reduced to 18 dimensions. However, the k-NN classifer outperformed the naïve Bayes classifier when more dimensions were retained [149, 150].

3.1.3.2 Gaussian Mixture Models. Marques and Moreno classified 200 milliseconds of recorded audio for eight musical instruments, using 16 Mel-frequency cepstral coefficients (MFCC) as features. The authors achieved 63% accuracy using Gaussian mixture models (GMM) which were outperformed by a SVM [118]. Another study achieved 66% accuracy using GMM on a small set of five instruments [151]. Other studies have compared GMM to *k*-NN [106], decision trees [99], and Hidden Markov Models (HMM) [121], but in none of these studies was GMM the most successful classifier.

One study reported a GMM classifier outperforming an SVM classifier on a dataset of five instruments. However, these results were skewed by instrument, with high recognition accuracy for some instruments and extremely poor accuracy for others [119]. In another study, the authors reported a GMM outperforming both an HMM and Non-negative Matrix Factorization (NMF) classifier, although the study used only a very small dataset of six instruments, and the accuracy differences between the classifiers may not be statistical significant [143]. One study reported a hierarchical Gaussian classifier outperforming a flat Gaussian classifier on a set 27 instruments [152].

3.1.3.3 Hidden Markov Models. Eronen trained a continuous-density HMM with MFCC features transformed by independent component analysis (ICA). On a dataset of 27 musical instruments, he achieved between 50% and 68% accuracy as the number of states of the model was varied experimentally [153]. One study used HMMs to successfully discriminate between two acoustic guitar players, but such results likely will not scale as number of players or instruments increased. [154]. Through numerous experimental variations, Joder *et al.* demonstrated an SVM to consistently outperform both GMM and HMM classifiers on a dataset of 8 instruments [121]. However, one study demonstrated HMM to outperform GMM in the classification of seven musical instruments, but this result may not be statistically significant [155].

3.1.4 Soft Computing Methods

3.1.4.1 Artificial Neural Networks. In 1995, Kaminskyj and Materka explored instrument classification using a multi-layered perceptron (MLP). On a small set of four instruments (piano, marimba, guitar, accordion), restricted to a single octave but covering five dynamic levels, the authors achieved greater than 94% accuracy. The authors varied the number of hidden layers between two and five but did not observe performance increases above four hidden layers [100]. Cemgil and Gürgen compared several network architectures and found the MLP to outperform both a Time-Delay Neural Network (TDNN) and a Self-Organizing Map (SOM). However, this study used a very small dataset covering 10 instruments and limited only to one octave [156].

In a series of studies, Loughran and her team explored musical instrument classification using MLP combined with dimensionality reduction on the feature set. The authors reduced their feature set via PCA, and used three, four, and five principal components (PC) to classify instruments with an MLP with two hidden layers. The authors compared three features sets: MFCC, temporal envelopes, and spectral envelopes, and found the MFCC feature set yielded the highest performance. The authors also found that performance improved when a higher number of MFCCs were used. Lastly, the authors achieved the highest performance on all feature sets with four PCs across all three feature sets attempted. While these results are anecdotally interesting, one should be cautious generalizing from them, given the small number of instrument classes used and the limited number of PCs explored [157, 158, 159].

Kostek *et al.* have undertaken numerous and comprehensive studies on the efficacy of neural networks for instrument classification, experimentally exploring various network architectures, training algorithms, the number of hidden layers, and various feature sets. In the later experiments the authors achieved over 90% accuracy. However, although the authors tested several different sets of musical instruments, these sets always contained a maximum of four instrument classes, strongly implying their approach may not scale to a large number of instruments [160, 161, 162, 163].

Another study examined different feature sets for musical instrument classification using MLP trained with back propagation. On a dataset of 19 instruments, the authors achieved 88% using a mixture of spectral features compared to 41% using wavelet features [164]. Bai and Chen examined a Fuzzy Neural Network (FNN), a neural network combined with a fuzzy reasoning system, to classify the four closely related instruments of the Violin family. Although an ANN and the FNN outperformed *k*-NN and an HMM, the FNN achieved 95% accuracy compared to the 92% accuracy of a traditional neural network [165].

3.1.4.2 Self-Organizing Maps. Since self-organizing maps (SOM) are trained with unsupervised learning methods, they cannot be used for classification without an attempt to associate the output clusters to specific labels. Since this process can be difficult and introduce error, SOMs have not often been used for instrument classification. Several studies have attempted to use SOMs to cluster sounds using comparisons to human-based perception results [43, 166, 167, 168, 169].

However, one study attempted to train a SOM for classification of musical instruments. Using a set of just ten features, the authors associated the learned clusters with class labels for a small set of five instruments. Although the authors reported 83% accuracy, the test set was extremely small and these results may not scale to more instrument classes [170].

3.1.4.3 Rough Sets. Kostek developed a decision system based on rough set theory applied to the selection of features in musical instrument classification. On a set of 15 instruments, she achieved 80% accuracy, outperformed by both a neural network and k-NN [171]. Wieczorkowska compared feature sets using rough sets. On a dataset of 18 instrument classes, she achieved 68% accuracy selecting among spectral features compared to 51% accuracy selecting from wavelet-based features [172, 173].

3.1.4.4 Evolutionary Methods. Evolutionary techniques, genetic algorithms (GA) in particular, are often used in feature selection for pattern matching problems across many domains [174], and researchers have explored this approach in musical instrument identification problems. Loughran *et al.* used a GA for selection of ten most useful features from a set of 95 spectral features. These features were used by an MLP to classify 3006 samples covering only five instruments. Although the authors achieved only 64% average accuracy, with wide variance between instruments, they argued their GA selected the most optimal feature set [175]. Another study used Particle Swarm Optimization (PSO), a swarm inspired population-based search technique, to optimize the parameters of an SVM [131].

In subsequent work, Loughran's team evolved a genetic program (GP) for musical instrument classification. On the small set of five instruments, the GP achieved 75% classification accuracy. However, the authors compared their GP to an MLP, which achieved greater than 99% accuracy on the same dataset with the same feature set [176].

3.1.5 Summary

Although many of the above studies report high levels of accuracy in the recognition of musical instruments, these studies cannot be directly compared because they use differing datasets, varying instrument sets, unique feature sets, and different evaluation metrics. Furthermore, many of the feature extraction approaches attempted for single instrument classification are not extensible to the polyphonic mixture task.

3.2 Polyphonic Classification

Many of the techniques attempted in solo instrument classification, however, are not practical for classification of real music performances in which multiple instruments often play at the same time. The task of recognizing instruments present in polyphonic mixtures is a more complex task as the harmonics of the instruments are interleaved in both time and frequency. Furthermore, the sounds in the mixture can interfere with each other both constructively and destructively. These issues complicate the extraction of acoustic features from polyphonic mixtures and researchers have sought various approaches to overcome this problem.

There are two general approaches to instrument recognition in polyphonic mixtures. The first approach attempts to extract general or robust features directly from polyphonic mixtures without attempting source separation. Because of the overlapping harmonics and the potential interference between sources, these approaches face the difficulty that the extracted features are often very different than features extracted from monophonic sounds. Additionally these techniques often do not scale to combinations of instruments unseen in the training set.

The second approach attempts to separate notes from the mixture and applies techniques from monophonic instrument recognition. These methods often require estimation of the fundamental frequency f_0 of each note in the mixture, a difficult problem in itself, and any error made at this stage can potentially propagate to the feature extraction and classification stages of the approach.

3.2.1 Techniques with Polyphonic Features

There have been several different approaches to classifying musical instrument in polyphonic mixtures without attempting to estimate source separation of the mixture. One naïve approach is to train on all possible combinations of musical instruments, converting the multilabel problem to a multiclass problem (see Section 2.2.2.2). While this approach might be feasible for pairs or even trios of instruments, training on all possible combinations of instruments results in a combinatorial explosion of class labels and quickly becomes intractable.

3.2.1.1 Taxonomy. Essid *et al.* created a system that does not require source separation but instead uses hierarchical clustering to build a taxonomy of musical instruments playing simultaneously. They tested their system on 20 instrument groupings, ranging from solo instruments to mixtures of four instruments from a dataset of commercial jazz recordings, achieving 53% accuracy [33, 177]. Because these 20 instrument groupings represent the closed set of possible class labels from their training and testing sources, their approach is a multi-class problem but their results are poorer than monophonic instrument classification studies of a comparable class label size, such as those by [105] (see Section 3.1). Furthermore, these experiments were trained on fixed combinations of instruments and would not be extensible to unseen combinations of instruments.

Another study attempted a multi-timbral approach of eight predefined instrument mixtures, reporting that k-NN outperformed a Bayesian network and a decision tree [178]. Although the authors reported 80% accuracy in identifying the instrument mixture, these experiments considered only eight instrument mixtures drawn from only a few examples with a large skew in the distribution of the class labels.

3.2.1.2 Robust Features. Another strategy to recognize instruments within mixtures is to attempt to extract features robust enough to enable recognition of individual instruments despite source interference. One such approach used polyphonic mixtures as training data in which examples are labeled if the target instrument appears at some point in a mixture, but is not given the exact times when the instrument is present. Over a small set of four instruments, the authors demonstrated that training with weakly-labeled examples yielded an improvement over training with isolated examples. This approach trains on mixtures, treating other instruments as noise to create a training set more representative of the polyphonic test data [179].

Another strategy attempted to locate areas of minimal interference between sources and prioritize features extracted from these areas of the mixture signal. Kitahara *et al.* used linear discriminant analysis to minimize the weight of features most affected by overlapping partials in polyphonic mixtures of sounds. On a dataset of recordings of mixtures from a set of five different instruments, the authors achieved 84% accuracy for duets, 77% for trios, and 72% for quartets. Although these result may look promising at the surface, the authors were only able achieve to achieve 72% accuracy when the mixture contained four out of the five possible instruments. This problem has a label density of 0.8, arguably an easier multi-label classification problem. Also, since the authors tested only a small set of instruments taken from only three hand-picked recordings, it is not possible to compare these results to other multi-label musical instrument classification studies. Furthermore this method requires labeling the training and test data with the f_0, onset time, and the duration – effectively a musical score. As this information will not be available for real-world data, the practicality of such an approach is severely limited [180, 181].

3.2.1.3 Multilabel Algorithms. One approach to multilabel classification is to adapt algorithms to directly perform multi-label classification without any attempt at problem transformation. Several studies have attempted classification of polyphonic mixtures using multilabel classifiers and summary features that do not attempt source separation.

A neural network can be adapted to multilabel classification by creating an output node for each possible instrument class. An early study trained a multilabel MLP to identify the dominant instrument present in instrument duets with 80% accuracy [182]. Kostek, building off her substantial body of work in monophonic instrument classification (see Section 3.1.4.1), developed an MLP for the recognition of musical instruments in duet and trio mixtures. Although she demonstrated some promising results, these were limited to only a few groupings of instruments for a only two or three specific pitches and only musical intervals that contain minimal overlapping partials were tested [163, 183]. Consistent with expectation, Kostek found the greatest confusions between instruments of the same family, such as the violin and viola. Another study used 136 generic features without source separation to discriminate the family of the instrument present in polyphonic mixtures. Drawing from a small set of only six instruments on synthesized audio data, the authors reported a deep belief neural network outperformed an SVM classifier and an MLP classifier with only a single hidden layer [184, 185].

Paulus and Klapuri proposed a method for detecting drum sounds in polyphonic recordings. The authors subdivided the signal into temporal frames and extracted 26 MFCC features as observations for Hidden Markov Models (HMM) which are decoded with the Viterbi algorithm. On a dataset of recordings on three drummers playing three different drums, the authors identified the correct instrument as present in the recording 79% of the time [186].

Kitahara *et al.* modeled polyphonic signals as a temporal spectogram, training an HMM to predict an probability of an instruments presence in the mixture. The authors considered only a small set of four instruments and examined mixtures containing three out of the four instruments, reporting 63% recall and 69% precision for trios of instruments [187, 188, 189]. Although the authors report a multi-label precision and recall measure, they average the accuracy of individual 10 ms time windows over the length of a piece of music. Such an approach unfairly weighs more heavily the identification of long sustained notes compared to shorter notes.

In additional to neural networks and HMMs, researchers have attempted other multilabel classification techniques. Testing on mixtures of two instruments drawn from set of 26 single instruments, the authors achieved 48% recognition with a Multilabel Decision Tree and 87% with a Multilabel k-NN. Although these results might seem encouraging, the authors do not address the ability of the experiments to scale to beyond just pairs of instruments and only tested instruments playing a single note, middle C (261 Hz). Furthermore, the authors average over time frames in a piece of music rather than individual notes creating a dependency which creates a dependency on their system and the distribution of musical notes by instrument in their training dataset [190, 191].

On mixtures drawn from a set of nine instruments, Fuhrmann *et al.* achieved 63% recognition using an multi-class SVM classifer trained for solo recognition to detect the presence of an instrument in a mixture of three instruments [32]. The authors considered a large feature space, extracting features for many short time windows over the length of the sample. When the authors extended their work to attempt a source separation pre-processing step based on spacial information of stereo recordings, they improved recognition 19% [192]. A recent work explored a multi-label SVM with a dataset of 13 instruments testing polyphonic mixtures ranging from two to six instruments. Using a self-defined evaluation method, the authors report an accuracy of 37% for two instruments, 32% for three instruments, and 30% for four instruments [193]. This study involves the same instrument set as our experiments and the same label density. We refer to this study in Section 9.4 in comparison with our multi-label results.

Another study used 120 generic features without source separation to identify musical instruments present in pairs of instruments. The authors built a binary random forest model for each instrument class, and reported 72% subset accuracy on their test set of instrument duos, however, the authors considered only 12 possible pairings of instruments. Furthermore, the authors observed that training on both single instruments and mixtures of instrument pairs was more beneficial than training on single instruments alone [194]. A recent study demonstrated a single multi-label random ferns classifier outperforming the a set of binary random ferns classifier, however the authors only considered four instruments and the scope of test data was rather limited [195].

Recent studies have explored feature selection for instrument recognition techniques that do not require prior source separation. Vatolkin *et al.* used an evolutionary algorithm to optimize feature selection in order to minimize classification error

[196, 197]. In another study, Vatolkin *et al.* compared generic features to instrument-specific features extracted from polyphonic mixtures. The authors observe that a feature set optimized for a particular instrument yields degraded classification accuracy when applied to other instruments [198]. This is an important result which argues against the optimizing feature selection for specific instrument, given the limited number of data sources many studies use. Additionally, this result implies that features sets optimized for a specific instrument are more desirable than single feature set balanced across all instruments, which carries an important implication for binary-relevance approaches to multi-label classification.

3.2.2 Source Separation Techniques

The other general approach to multilabel classification of polyphonic mixtures is to attempt some form of source separation. Common approaches to source separation include matching single instrument templates in mixtures, feature selection to minimize interference, and modeling of signal mixtures inspired by computational auditory scene analysis.

3.2.2.1 Template Matching. Kashino *et al.* created a music transcription architecture named OPTIMA (Organized Processing toward Intelligent Music Scene Analysis) that attempted recognition of individual notes from signals using template matching that included mixtures of up to five instruments [199]. This scheme, however, required estimation of the f_0 and the onset time of each note. This work was continued in [200, 201], achieving 88% recognition of three different instruments, so long as the system was provided the true pitches of the notes. Kinoshita *et al.* further improved the performance of the system using a weighted template-matching scheme to achieve

75% accuracy in identifying musical notes, but not specific instruments, without requiring prior knowledge of the pitch [202].

Leveau *et al.* decomposed signals into a mid-level representation to train a dictionary of prototypical atoms based on solo instrument examples. The authors model signals as the composition of various pitch and instrument specific atoms using an optimization process. The authors achieved between 56% and 87% accuracy in a single instrument recognition task over a dataset of five instruments [203, 204]. Applying this technique to identification of musical instruments in polyphonic duets, the authors achieved an accuracy varying between 48% and 87%, a result that seems highly dependent on the specific combination of musical instruments [204]. Since the authors considered only four pairs of instruments, rather than all of the ten possible pairings of the five instruments, the results are more comparable to single instrument multi-class study than a multi-label classification approach. In extending their technique to three and four instrument mixtures, the authors achieved less than 45% and 15% accuracy, respectively [205].

3.2.2.2 Missing Feature Approach. Another approach to source separation is to attempt recognition of individual sources based on partial information. The missing feature approach has been used in speech recognition [206, 207, 208]. For the purposes of polyphonic classification, these approaches use knowledge or expectations of an instrument's timbre to guide feature extraction and selection.

Eggink and Brown proposed an approach that attempted to identify areas of interference between sound sources. In their missing feature approach, the authors assumed expert knowledge about the instrument's timbre to create a mask to exclude features from regions of hypothesized source interference. Excluding these unreliable features, the authors used a Gaussian Mixture Model (GMM) to classify the instrument [151]. From a set of five musical instruments, the authors achieved 63% accuracy in identifying two instruments playing concurrently [209]. The authors extended this approach to identify the solo instrument from the accompaniment in sonatas and concertos, achieving 47% accuracy from a set of five instruments [210]. Although the authors argue the success of their approach given a correctly estimated mask, this approach assumes a harmonic spectra and therefore would not be extensible to all musical instruments. Furthermore, determination of the mask relies on the correct identification of multiple fundamental frequencies in the signal, a difficult unsolved problem in itself [211, 212]. Although Eggink and Brown considered three different datasets, they did not include any cross-dataset experiments.

Barbedo *et al.* sought to identify areas of no interference and use only features extracted from these areas of the mixture's spectra, ignoring other areas of the spectra. In their study the authors attempted to identify isolated partials from which they extracted features for a pairwise linear discriminant classifier. The authors demonstrated their approach on a large set of 25 instruments drawn from multiple sources to achieve precision ranging from 0.84 on two instrument mixtures down to 0.5 on six instrument mixtures [35]. This approach partitions an example into many small time frames. Each time frame is classified and the chosen label votes toward the determination of the finals set of class labels for the example. Additionally, this approach, like many others mentioned previously, depends on the knowledge of the number of instruments in the mixture and their respective f_0s.

Another study modeled spectral envelopes as time-varying functions of log-frequency to estimate masks of an instrument's timbre. The authors calculated the probabilistic reliability of different features to an ideal mask and used bounded marginalization to marginalize the features considered unreliable. On a set of ten instruments, the study reported 64% average accuracy for single instruments and

61% average accuracy of recognition of individual instruments within mixtures of four instrument. Unfortunately the authors do not report a multi-label evaluation measure, but rather accuracy of recognition of each individual instrument, and this accuracy varied greatly between individual instruments. Since the dataset used contains a differing number of examples for each instrument, their approach indicates a bias towards the training data. Furthermore, this approach relies on accurate multi-pitch f_0 detection [213].

3.2.2.3 Source Factorization. Although instrument classification does not require precise signal separation, some approaches have borrowed techniques from the area of blind source separation and applied these to polyphonic instrument classification.

Virtanen *et al.* attempted an unsupervised approach to source separation by applying weighted non-negative matrix factorization on the power spectrogram on an input signal. The authors factorized the input power spectrogram into a sum of components that have a fixed magnitude spectrum with a time-varying gain and minimized reconstruction error between the input spectrogram and a set of linear models of the spectrogram. Using a database of six instruments and testing on a set of mixtures of two to five musical instruments, the authors achieved 73% accuracy for two instrument mixtures down to 66% accuracy for five instrument mixtures [146, 214, 215, 216]. However, this study considered a very small dataset including only 26 examples in the polyphonic test sets. This approach relies of temporal model which averages the classification over many small individual time frames.

Matrix factorization has also been attempted to separate two instruments [217], extract drum sounds from other harmonic sounds [218, 219, 220], separate vocals from musical accompaniment [221], and extract speech from background music [222].

Recently, studies have shown close relationships between NMF and both probabilistic latent semantic analysis (PLSA) and probabilistic latent component analysis (PLCA) [223]. Grindlay and Ellis attempted PLSA for polyphonic transcription including instrument identification. The authors modeled individual instruments as spectrograms containing a joint distribution of time and frequency. The authors considered mixtures to be weighted combinations of these instrument subspaces, estimating the unknown parameters using the EM algorithm. On a large set of 34 instruments, the authors achieved 45% recognition for two instruments, but only 26% recognition of four instruments [224, 225, 226]. This study was focused on transcription, averaging instrument recognition over many times frames and the data consisted of long musical examples in which presumably not all instruments were playing concurrently or perhaps contains moments of musical silence (rests). The frames containing only one instrument or none at all likely artificially inflate their results.

A more recent study using PLCA for instrument identification demonstrated improved recognition, but only for mixtures of two instruments drawn from a set of four instruments [227]. Bentos and Dixon created a shift-invariant PLCA system for polyphonic transcription achieving and a frame-wise F-measure of 45% instrument recognition in a sample four voice recording [228, 229].

Another study attempted a probabilistic mixture model decomposition in which the probability density function of an observed mixture note is estimated as a weighted sum approximation of time-frequency models for individual notes. The probability of the existence of an instrument playing a specific pitch is represented as a weight coefficient, which is estimated using the Expectation-Maximization (EM) algorithm. Training over a set of 14 musical instruments, the authors achieved 75% accuracy for two instruments down to 62% accuracy for four instrument mixtures [230]. This approach relies on onset detection of the note and is a temporal model.

3.2.2.4 Computational Scene Analysis. Inspired by computational auditory scene analysis, Vincent and Rodet represented the spectra of polyphonic mixtures as weighted non-linear combinations of typical note spectra plus background noise, learning the prototypical spectra from a dataset of solo instruments. The authors search for the combination of instruments with the highest probability of explaining the mixture [231]. Another approach uses sinusoidal modeling and dimensionality reduction to build prototypical spectro-temporal envelopes of different instruments. One study used a graph partitioning algorithm to cluster these envelopes and classify a set of six instruments, ranging from 83% accuracy in the single instrument case to 33% for four instrument mixtures [232].

Wu *et al.* modeled the spectral envelope as a Gaussian mixture of harmonic models and onset attack models for each potential note present in a mixture. On a dataset of six instruments and using a SVM to classify, the authors achieved 74.8% accuracy for two instrument mixtures down to 50.7% for four instrument mixtures using a customized metric for transcription accuracy over the times frames of the musical excerpt [233]. Another series of studies modeled temporal-spectral envelopes as Gaussian processes and used Euclidean distance to the prototypes as a classification metric, achieving 95% accuracy for single instruments, 73% accuracy for two instrument mixtures, and 54% for four instrument mixtures, however, drawing from a limited set of only five instruments [234, 235, 236].

3.2.3 Summary

Although there have been several approaches to multi-label classification of polyphonic mixtures, these studies vary in their choices of feature, classification algorithms, datasets, and evaluation metrics, and the reported results are not comparable. Furthermore, these approaches suffer many limitations in their datasets including the use of synthetic data [184, 185], consideration of a examples of single notes and only one musical interval [163, 183], evaluation of only a small instrument set of only four or five instruments [163, 179, 183, 189, 236].

Many of these approaches require temporal features [32, 35, 189, 191, 192, 231, 232] and report frame-based accuracy measures. Some approaches are not scalable as the number of labels increase [33, 177, 178]. Others hand-picked multi-label pairings, evading the multi-label problem [194]. Some studies require prior knowledge of the frequencies or timing information of the notes [181, 230], an expectation unrealistic for real-world data. Finally others of these studies reported in this chapter have goals that differ from multi-label classification including transcription, in which test examples contain passages of individual instruments [226, 229, 233], identification of instrument as present at some point in a musical except [186], or ability to identify only one instrument from a mix and ignoring the others [182].

CHAPTER 4

SINGLE INSTRUMENT CLASSIFICATION

In this chapter, we investigate classification of single, monophonic musical instruments using several different Bayesian network structures and a feature extraction scheme based on a psychoacoustic definition of timbre. This early work heavily influenced our subsequent approaches, including our choice of datasets, our cross-dataset experimental design, and our feature extraction technique. The results present a seminal use of graphical models in the task of musical instrument classification, and are compared to the baseline algorithms of support vector machines (SVM) and a k-nearest neighbor (k-NN) classifier.

4.1 Datasets and Feature Extraction

Feature extraction is a form of dimensionality reduction in which, for the purposes of this task, the audio files are transformed to a small vector of highly relevant numeric features. Recently, attention in the literature has centered on the task of feature identification (see [36] for a review) rather than on the choice of learning strategy. In addition to differing datasets, most of the studies in the literature have used varied sets of features, rendering many direct comparisons of studies in the literature impossible. In order to compare our Bayesian approach to timbre classification to the methods commonly used in the literature, we create a dataset, define a spectral-based feature extraction scheme – which is a preliminary version of our more complete feature extraction scheme described later in this dissertation, and empirically compare our

Table 4.1: EastWest dataset of 24 instruments sorted by instrument family.

Strings	Woodwinds	Brass	Percussion
Violin Viola Cello Contrabass Harp	Piccolo Flute Alto Flute Clarinet Bass Clarinet Oboe English Horn Bassoon Contrabassoon Organ	French Horn Trumpet Trombone Tuba	Chimes Glockenspiel Vibraphone Xylophone Timpani

Bayesian classifiers to a k-NN and two SVM classifiers. Additionally, we test our feature extraction scheme and our classifiers on the publicly available MIS dataset.

4.1.1 EastWest Dataset

For our experiments, in addition to the MIS dataset described in the next section, we create a dataset (EastWest) that contains 1000 audio examples for each musical instrument, covering the 24 different orchestral instruments shown in Table 4.1. Each audio file is two seconds in duration, consisting of the instrument sustaining a single note for one second, and time before and after to capture the attack and the resonant decay, respectively. The audio samples were created using the EastWest Symphonic Orchestra sample library [237] at the **MON**tana **ST**udio for **E**lectronics and **R**hythm (MONSTER) at Montana State University.

Figure 4.1 shows an overview of the data generation process. For each musical instrument, the Kontakt Virtual Studio Technology (VST) player [238] loads the respective samples from the EastWest sample library. For each musical example, a MIDI control sequence is sent from a Java program to the Kontakt sampler for

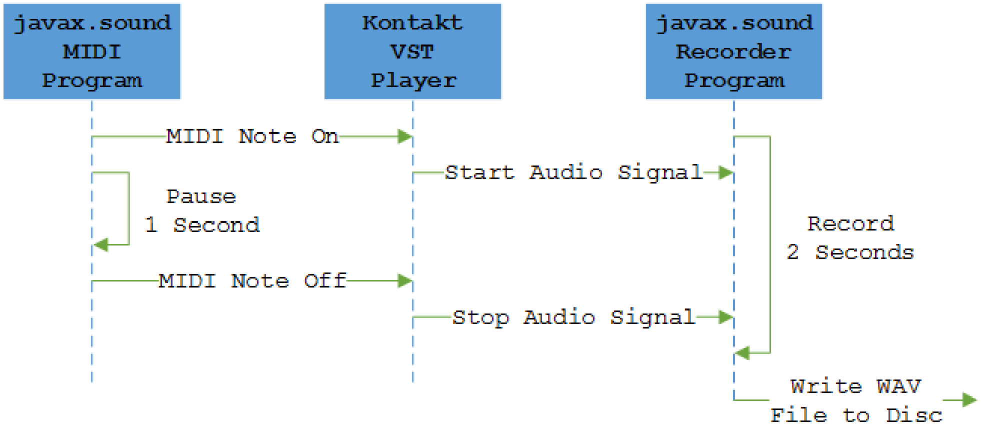


Figure 4.1: For a selected pitch and dynamic level, MIDI control signals are transmitted to the Kontakt VST player. The VST player renders an audio signal corresponding to the parameters of the MIDI messages. This signal is then recorded by another Java program and the resulting sample is saved to disc as a WAV file.

rendering to audio. The interaction between Java and the VST player is handled by the *jVSTwRapper* interface [239]. Using the EastWest Symphonic Library, the VST player produces an audio signal that corresponds to the parameters of the MIDI message. The resulting audio stream is recorded in another Java program using the javax.sound package. The samples are recorded at a 44.1 kHz sampling rate, 16-bits per sample, and stored as a single channel waveform audio file (WAV).

The pitch is randomly sampled uniformly with replacement covering the entire musical range of the instrument. The dynamic level is also sampled uniformly with replacement of the MIDI velocity parameter in the range $[40, 105]$, covering the dynamic range *pianissimo* to *fortissimo*. In total, there are 1000 audio samples for each of the 24 instruments, yielding 24,000 total examples.

The dataset is then normalized to the range $[0, 1]$ using the audio utility *normalize* [240]. The files are batch normalized to scale the loudest gain in any of the files to a value of one and adjusting all the other files by this offset. This method preserves the relative dynamic levels between example files.

Table 4.2: MIS dataset of 25 instruments sorted by instrument family.

Strings	Woodwinds	Brass
Piano	Alto Flute	
Guitar	Flute	
Violin	Bass Flute	French Horn
Viola	Soprano Saxophone	Trumpet
Cello	Alto Saxophone	Trombone
Bass	Bb Clarinet	Bass Trombone
Violin Pizzicato	Eb Clarinet	Tuba
Viola Pizzicato	Bass Clarinet	
Cello Pizzicato	Oboe	
Bass Pizzicato	Bassoon	

4.1.2 MIS Dataset

The MIS dataset (**M**usical **I**nstrument **S**amples), created by the Electronic Music Studios at the University of Iowa, contains scales of 21 different musical instruments each at three different dynamic levels: *pianissimo*, *mezzoforte*, and *fortissimo* [241]. We use a subset of the instruments from the MIS dataset in later experiments, described in Chapters $6 - 9$, and is discussed in more detail in Section 5.1.2.

We parsed these scales into individual files each containing a single note using the *Sound eXchange* (SoX) audio program [242]. For the purposes of these experiments, the bowed and *pizzicati* samples of the Violin, Viola, Cello, and Contrabass are considered to be eight separate classes. This dataset contains 4521 samples covering the 25 different instrument classes shown in Table 4.2. The number of samples for each instrument varies, ranging from 70 examples of the Bass Trombone up to 352 examples of the Cello. The samples are remixed in mono, 44.1 kHz, 16-bit, clipped to two seconds in duration, and batch normalized to the range $[0, 1]$ using the normalization strategy described in the previous section.

4.1.3 Feature Extraction

Each audio sample is transformed to the frequency domain using an FFT. The signal is first divided into equal width time windows. The number of time windows is selected to be twenty to yield 100-millisecond windows. Each of these 100-millisecond time windows is analyzed using a fast Fourier transform (FFT) to transform the data from the time domain into the frequency domain. This FFT transformation yields an amplitude value for each frequency point present in the analysis.

Frequency perception is a logarithmic concept but FFT analysis provides a resolution across a linear Hertz scale. Therefore, for example, the analysis provides a much lower resolution for the lowest note of the piano compared to the resolution of the highest note. In order to group nearby frequencies into a single window, the vector is divided into ten exponentially increasing windows, where each frequency window is twice the size of the previous window, covering the range $[0, 22050]$ Hertz. This scheme allows the system to generalize over musical pitch.

Ten frequency windows are selected and for each of the ten frequency windows, the peak amplitude is extracted as the feature. The feature set for a single musical instrument example consists of ten frequency windows j for each of twenty time windows i, yielding 200 features per audio example. The feature extraction scheme is outlined in Figure 4.2.

These 200 continuous features, ranging $[0, 1000]$, are discretized into a variable number of bins using a supervised entropy-based binning scheme [243]. Entropy provides a measure of purity of a certain interval. Let k correspond to the number of class labels and p_{ij} correspond to the conditional probability of class j occuring in

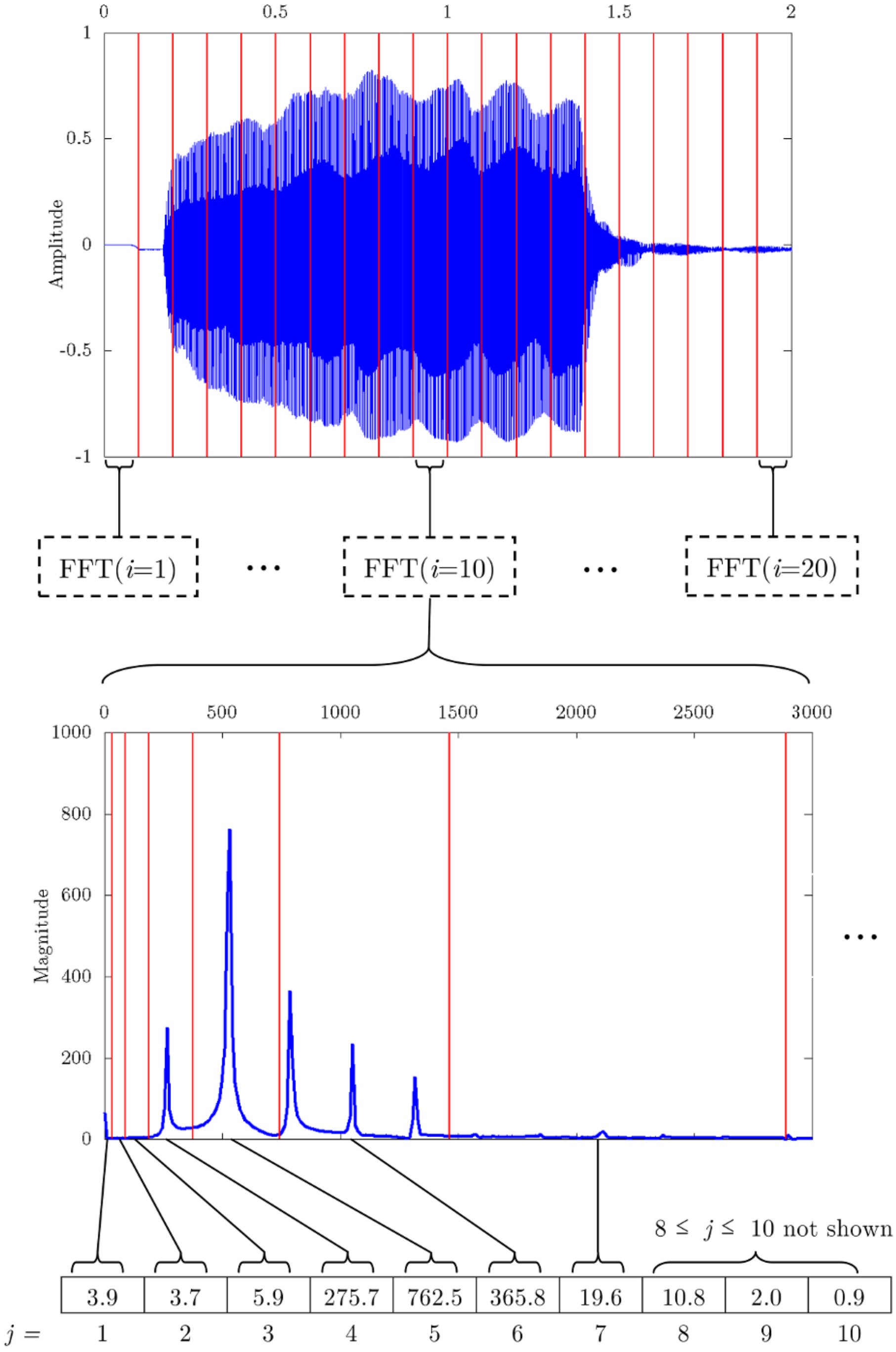


Figure 4.2: Each two second example is partitioned into twenty equal length windows. FFT analysis is performed on each 100 millisecond time window. The FFT analysis for i=10 is depicted. The FFT output is partitioned into ten exponentially increasing windows. For readability, only the first seven frequency windows are depicted above. The peak frequency from each window is extracted and used as a feature.

the ith interval. The entropy h_i of the interval i is given by the equation:

$$h_i = -\sum_{i=1}^{k} p_{ij} \log p_{ij} \tag{4.1}$$

The total entropy H of the discretization is the weighted average of the individual entropies:

$$H = \sum_{i=1}^{n} w_i h_i \tag{4.2}$$

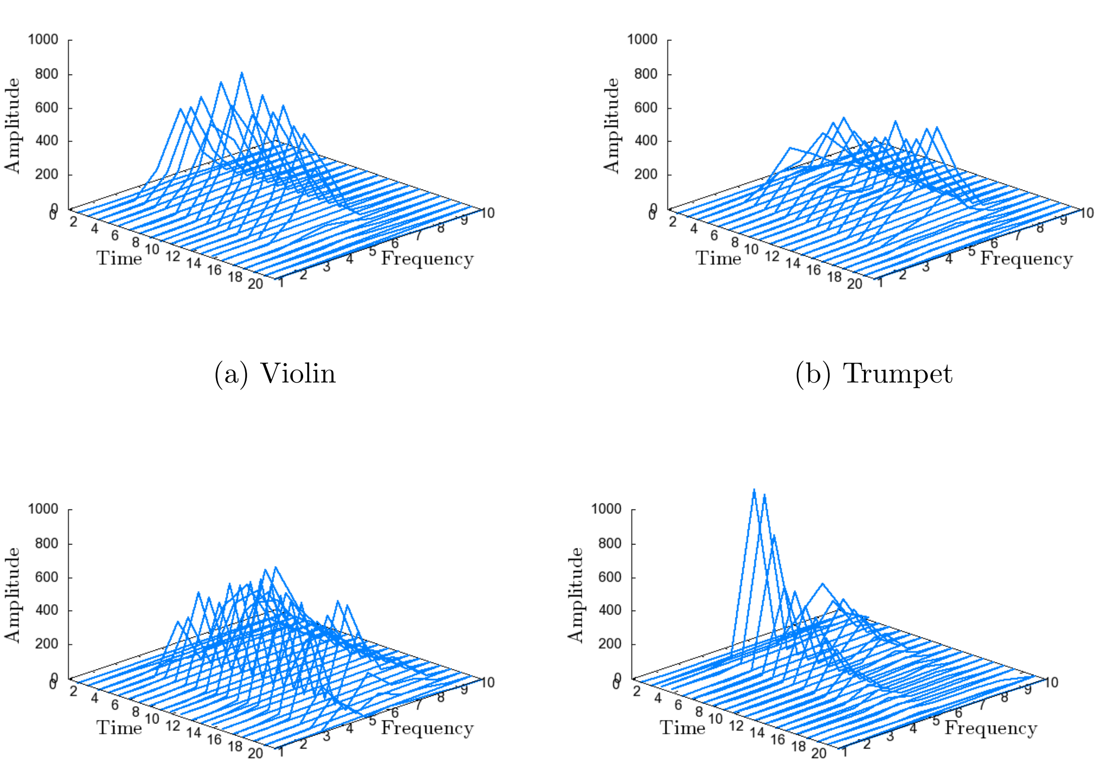
where m is the number of values in the dataset, $w_i = m_i/m$ is the fraction of the values in the ith interval, and n is the number of intervals.

Entropy based discretization considers all possible bisections of an interval, computes the associated entropies, and retains the bisection with the lowest entropy. The process continues by selecting the next interval with the highest entropy and repeating the process until the stopping criterion, given by [244], is reached.

This feature set attempts to capture the unique timbre of the each musical instrument by generalizing the changes in amplitude of groups of nearby partials over time for each instrument. Examples of the feature set for four musical instruments are visualized in Figures 4.3a - 4.3d.

4.2 Models and Experimental Design

On these datasets, these experiments compare the performance of several Bayesian model structures in the task of musical instrument classification (see Section 2.3.4). The first model described is the naïve Bayes classifier. The remaining three Bayesian networks consist of variations of a grid-augmented naïve Bayes model, each adding different conditional dependencies in the time and frequency domains. This novel



(a) Violin

(b) Trumpet

(c) Clarinet

(d) Xylophone

Figure 4.3: Visualization of the feature set for four different musical instruments each playing middle C at a mezzoforte dynamic level.

topology is a unique variation of a tree-augmented naïve Bayes structure and allows modeling dependencies in two dimensions [245].

For these descriptions, let f_j^i be the peak amplitude feature f at frequency window j for time window i, where $0 < i \leq 20$ and $0 < j \leq 10$.

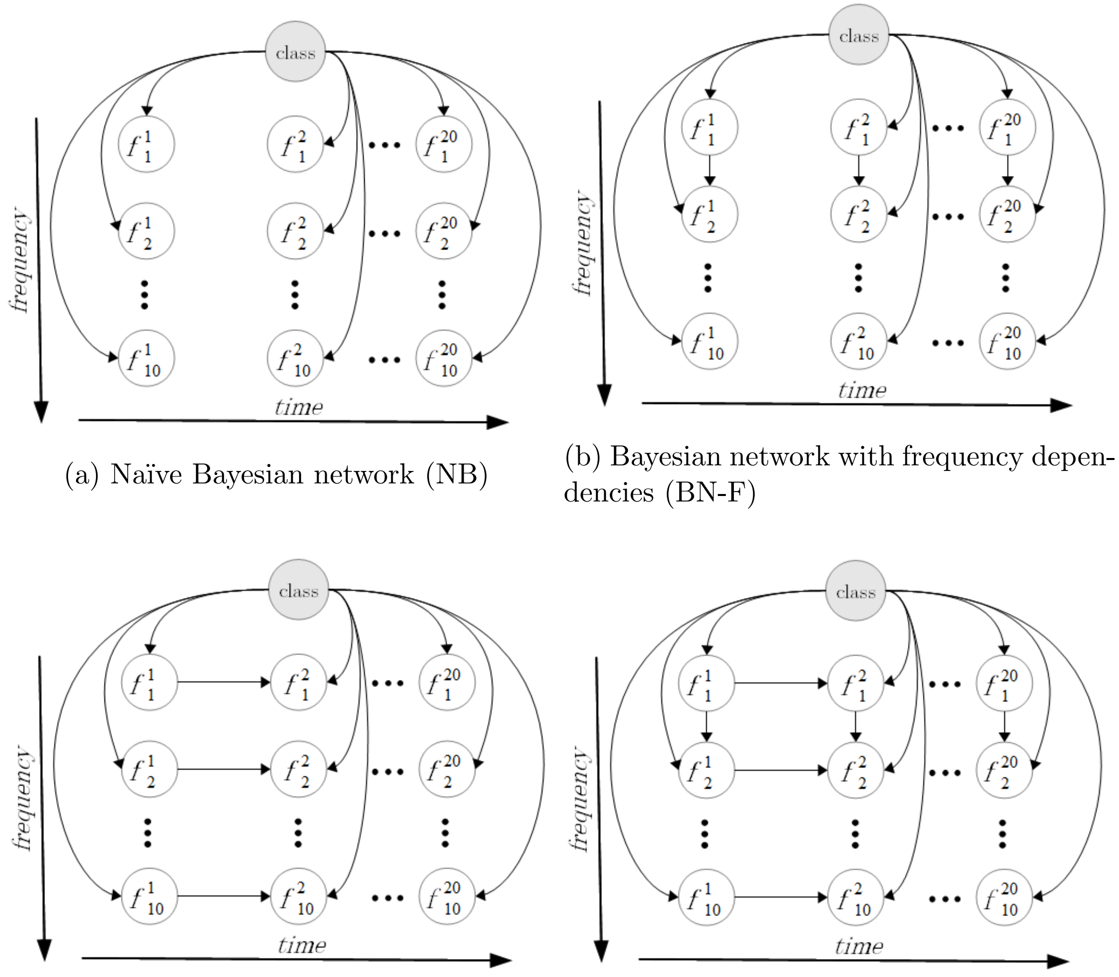
4.2.1 Naïve Bayes

For a baseline Bayesian model, we chose the common naïve Bayes classifier (NB). In the NB model, all evidence nodes are conditionally independent of each other, given the class. The formula for NB is shown as Equation 4.3 in which $P(c)$ is the class prior and $P(f \mid c)$ is the probability of a single feature within the feature set, given a particular class c. The NB network is shown graphically in Figure 4.4a.

$$P(c \mid \mathbf{f}) = P(c) \times \prod_{f \in \mathbf{f}} P(f \mid c) \tag{4.3}$$

4.2.2 Frequency Dependencies

The second model is a Bayesian network with frequency dependencies (BN-F), in which each feature f_j^i is conditionally dependent on the previous frequency feature f_{j-1}^i within a single time window as shown in Figure 4.4b, denoted as $f_{j-1}^i \rightarrow f_j^i$. Equation line 4.4a shows the class prior and the probability of the first row of the grid of features while line 4.4b defines the probability of the remaining features. There



(a) Naïve Bayesian network (NB)

(b) Bayesian network with frequency dependencies (BN-F)

(c) Bayesian network with time dependencies (BN-T)

(d) Bayesian network with frequency and time dependencies (BN-FT)

Figure 4.4: Structure of the different Bayesian networks.

are no dependencies between the different time windows.

$$P(c \mid \mathbf{f}) = P(c) \times \prod_{i=1}^{20} P(f_1^i \mid c) \tag{4.4a}$$

$$\times \left(\prod_{i=1}^{20} \prod_{j=2}^{10} P(f_j^i \mid f_{j-1}^i, c) \right) \tag{4.4b}$$

4.2.3 Time Dependencies

The third model, a Bayesian network with time dependencies (BN-T), contains conditional dependencies of the form $f_j^{i-1} \rightarrow f_j^i$ in the time domain, but contains no dependencies in the frequency domain (see Figure 4.4c). Equation line 4.5a shows the class prior and the probability of the first column of the grid of features while line 4.5b defines the probability of the remaining features.

$$P(c \mid \mathbf{f}) = P(c) \times \prod_{j=1}^{10} P(f_j^1 \mid c) \tag{4.5a}$$

$$\times \left(\prod_{i=2}^{20} \prod_{j=1}^{10} P(f_j^i \mid f_j^{i-1}, c) \right) \tag{4.5b}$$

4.2.4 Frequency and Time Dependencies

The final model, a Bayesian network with both time and frequency dependencies (BN-FT), is shown in Figure 4.4d. The BN-FT model is a combination of BN-F and BN-T and contains dependencies of the form $f_j^{i-1} \rightarrow f_j^i$ and $f_{j-1}^i \rightarrow f_j^i$. Equation line 4.6a shows the class prior and the probability of the upper-leftmost node (f_1^1) of the feature grid. Line 4.6b shows the probability of first column of the grid, line 4.6c, that of the first row of the grid, and line 4.6d, that of the remaining features.

$$\begin{aligned}
P(c \mid \mathbf{f}) &= P(c) \times P(f_1^1 \mid c) && \text{(4.6a)}\\
&\times \left(\prod_{i=2}^{20} P(f_1^i \mid f_1^{i-1}, c) \right) && \text{(4.6b)}\\
&\times \left(\prod_{j=2}^{10} P(f_j^1 \mid f_{j-1}^1, c) \right) && \text{(4.6c)}\\
&\times \left(\prod_{i=2}^{20} \prod_{j=2}^{10} P(f_j^i \mid f_j^{i-1}, f_{j-1}^i, c) \right) && \text{(4.6d)}
\end{aligned}$$

4.2.5 Baseline Algorithms

To explore the advantages of time and frequency dependencies between features, the accuracies of the grid-augmented Bayesian models are compared with two support vector machines, a k-nearest neighbor classifier, and naïve Bayes. SVM and k-NN are chosen as the baseline algorithms for comparison to the Bayesian networks given the prevalence of these algorithms in the literature. These algorithms are described in Section 2.3.

For the SVM, we selected both a linear (SVM-L) and polynomial kernel (see Equation 2.3) where $\delta = 2$ (SVM-Q). We also examined a radial basis function kernel and sigmoidal kernel; both scored at chance and were subsequently not included in the experiments. For k-NN, we empirically examined values of k from 1 to 10. k-NN with $k = 1$ achieved the highest accuracy and was selected for use in all experiments.

4.2.6 Experimental Design

All experiments were run using ten-fold stratified cross-validation for training and testing. For the Bayesian networks, the parameter learning stage consisted of constructing the conditional probability tables (CPT) using counts from the training data. For all the Bayesian networks, the worst case size complexity of any variable's CPT is $O(n \cdot a^p)$ where $n = 200$ is the number of features, $9 \leq a \leq 42$ is the number of discretized states for any variable, and p is the maximum number of parents. For the most complex model, the BN-FT model, $p \leq 3$ for all variables.

In the testing phase, any event unseen in the training data results yields a zero probability of the entire feature vector. To prevent this, we used the common technique of additive smoothing:

$$P(f_j^i) = \frac{x_i + \alpha}{N + \alpha \cdot d} \tag{4.7}$$

where $\frac{x_i}{N}$ is the probability of feature x_i, as indicated in the training data, and d is the total number of features [246]. The parameter α adds a small number of pseudo-examples to each possible feature value, eliminating a possible count of zero that might result in a zero probability. A value of $\alpha = 0.5$ was used in all experiments.

4.3 Experiments and Results

To test the utility of conditional dependencies between variables in the frequency and time realms, we conducted four experiments. In the first, we compare our Bayesian models against the baseline models on the EastWest data set in both the tasks of instrument identification and identification of musical instrument family. In the second experiment, we explore classification accuracy on instruments within the same musical family. In the third experiment we examine classification accuracy as a function of the number of instrument samples for each instrument. Lastly, in the

Table 4.3: Experiment 1 - Classification Accuracy (%) by instrument ($n = 24$) and by instrument family ($n = 4$) for the East West Dataset.

Algorithm	Instrument	Family
NB	81.57	80.94
BN-F	97.53	92.87
BN-T	96.36	94.39
BN-FT	**98.25**	97.09
SVM-L	81.46	85.57
SVM-Q	93.55	95.65
k-NN	92.99	**97.31**

the fourth experiment we examine the classification accuracy of all algorithms on the MIS data set.

4.3.1 Experiment 1: Instrument and Family Identification

The first experiment examines classification accuracy for both instrument identification ($n = 24$) and family identification ($n = 4$) on the EastWest dataset. The results are shown in Table 4.3. The statistical significances using a paired student t-test with $p \leq 0.01$ are shown in Table 4.4.

All of the Bayesian networks, with the exception of naïve Bayes, outperformed both SVMs and k-NN. The model with frequency dependencies (BN-F) outperforms the model with time dependencies (BN-T). The combination of both frequency and time dependencies outperforms BN-F and BN-T in both tasks, more significantly so in the family identification task.

In many previous experiments, the family identification problem was found to be an easier problem than the instrument identification problem. Conversely, in this experiment, the Bayesian networks perform less well on the family identification problem compared to the instrument identification problem. Both SVMs and k-

Table 4.4: Statistical significance of Experiment 1 using paired t-test with $p < 0.01$. Each cell indicates if the algorithm listed in the column performed significantly better (+), significantly worse (−), or not significantly different (0) when compared to the algorithm listed in the row. The first value is the significance of the instrument ($n = 24$) experiment and the second shows the family ($n = 4$) experiment.

Algorithm	NB	BN-F	BN-T	BN-FT	SVM-L	SVM-Q	k-NN
NB	—	+/+	+/+	+/+	0/+	+/+	+/+
BN-F	−/−	—	−/+	+/+	−/−	−/+	−/+
BN-T	−/−	+/−	—	+/+	−/−	−/+	−/+
BN-FT	−/−	−/−	−/−	—	−/−	−/−	−/0
SVM-L	0/−	+/+	+/+	+/+	—	+/+	+/+
SVM-Q	−/−	+/−	+/−	+/+	−/−	—	0/+
k-NN	−/−	+/−	+/−	+/0	−/−	0/−	—

NN, however, both yield improved classification accuracy on the family identification problem, consistent with the literature.

Confusion matrices for the family identification task are shown in Table 4.5. The Bayesian models show increased confusion between brass and woodwind instruments compared to string or percussion instruments. The SVMs, k-NN and naïve Bayes, on the other hand, more often confuses strings with either brass or woodwind compared to the Bayesian networks.

4.3.2 Experiment 2: Instrument Identification within Family

This experiment examines instrument classification by instrument family on the EastWest dataset. Unlike Experiment 1, this experiment trains and tests only on instruments within the same family (see Table 4.6). The dataset was divided into four separate datasets, one for each family, eliminating the possibility of confusion with instruments outside its own family. Ten-fold cross-validation is used on each of the family datasets.

Table 4.5: Confusion matrices for the family identification on the EastWest dataset, showing classification counts. Bold values indicate a correct classification.

Algorithm	S	B	W	P	← classified as
NB	**4470**	21	327	162	String
	24	**3021**	944	11	Brass
	277	1923	**7799**	1	Woodwind
	220	320	324	**4134**	Percussion
BN-F	**4865**	15	107	13	String
	3	**3756**	239	2	Brass
	97	883	**9009**	111	Woodwind
	123	86	133	**4658**	Percussion
BN-T	**4921**	0	34	45	String
	13	**3612**	364	11	Brass
	173	600	**9223**	4	Woodwind
	27	55	21	**4897**	Percussion
BN-FT	**4923**	3	67	7	String
	1	**3627**	372	0	Brass
	19	198	**9783**	0	Woodwind
	4	15	13	**4968**	Percussion
SVM-L	**4692**	11	254	43	String
	47	**1265**	2685	3	Brass
	140	226	**9626**	8	Woodwind
	25	3	19	**4953**	Percussion
SVM-Q	**4670**	69	188	73	String
	84	**3667**	245	4	Brass
	119	190	**9680**	11	Woodwind
	42	5	14	**4939**	Percussion
k-NN	**4792**	56	107	45	String
	40	**3795**	162	3	Brass
	43	145	**9802**	10	Woodwind
	22	6	6	**4966**	Percussion

Table 4.6: Experiment 2 - Classification accuracy (%) on the EastWest dataset by instrument family.

Algorithm	Strings	Woodwinds	Brass	Percussion
NB	89.76	84.58	92.43	99.64
BN-F	**99.86**	95.89	**99.70**	99.94
BN-T	99.12	95.56	99.36	99.92
BN-FT	99.60	**97.86**	99.58	**99.96**
SVM-L	98.66	92.01	98.65	98.18
SVM-Q	96.82	94.62	97.35	98.48
k-NN	98.72	92.67	98.63	99.72

Interestingly, the classification accuracy of strings, brass, and percussion exceeds 99% for all the Bayesian networks except naïve Bayes, whereas woodwinds, the largest set of instruments ($n = 10$), achieves 97.86% accuracy. For the strings, brass, and percussion, the BN-F and BN-FT achieves comparable accuracy, however, BN-FT outperforms BN-F on the more difficult woodwind set. The percussion set achieve the highest accuracy for all algorithms, including the SVMs and k-NN.

4.3.3 Experiment 3: Accuracy by Dataset Size

This experiment examines the classification accuracy by instrument ($n = 24$) on the EastWest dataset, similar to Experiment 1, but as the dataset size is varied from 100 to 1000 in increments of 100 for each instrument (see Figure 4.5). The Bayesian network models converge to their respective optimal accuracy between 500 and 800 data samples per instrument. However, both the SVMs and k-NN continue to improve as the number of examples increase. It is possible that both would continue to improve in accuracy if given more examples beyond 1000 examples per instrument. However, all the Bayesian models achieved much higher accuracy with far fewer examples than either SVMs or k-NN.

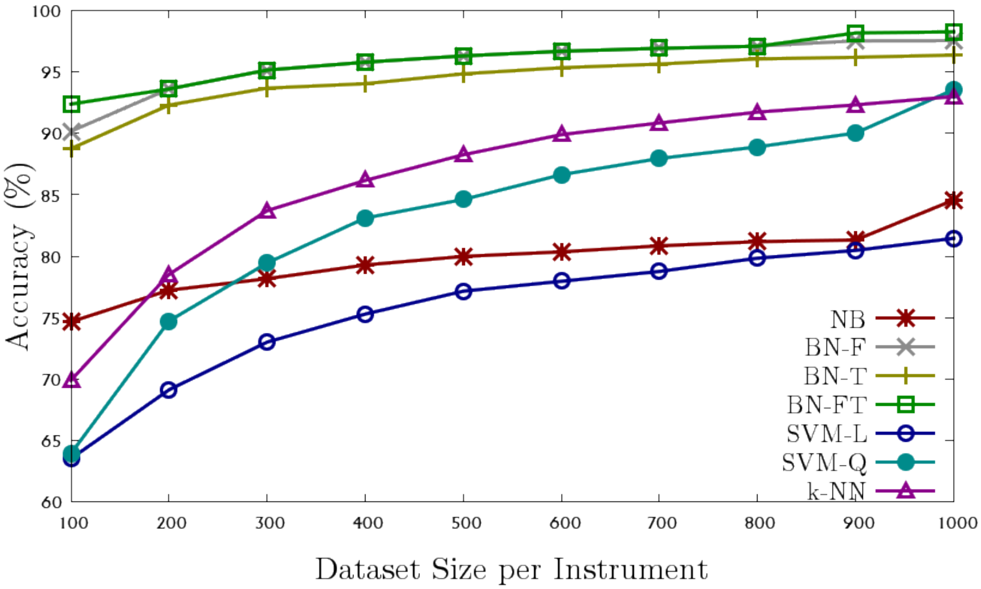


Figure 4.5: Experiment 3 - Accuracy (%) on the EastWest dataset by number of examples per instrument for each model.

4.3.4 Experiment 4: MIS Dataset

The final experiment examined classification accuracy for both instrument identification ($n = 25$) and family identification ($n = 3$) on the MIS dataset. The results are shown in Table 4.7.

As in Experiment 1, all of the Bayesian networks, again with the exception of naïve Bayes, outperform both SVMs and k-NN. The model with frequency dependencies outperforms the model with time dependencies. BN-FT and BN-F achieves comparable accuracies in the instrument task. The combination of both frequency and time dependencies outperforms BN-F and BN-T in the family identification task.

The MIS dataset contains fewer examples of each instrument compared to the EastWest dataset, and several instruments in the MIS dataset contain less than 100 examples each. Nevertheless, these results on the MIS dataset are consistent with

Table 4.7: Experiment 4 - Classification Accuracy (%) by instrument ($n = 25$) and by instrument family ($n = 3$) for the MIS Dataset.

Algorithm	Instrument	Family
NB	46.34	73.30
BN-F	**80.76**	81.82
BN-T	75.25	81.24
BN-FT	**80.31**	87.33
SVM-L	65.36	75.03
SVM-Q	65.89	83.19
k-NN	72.78	**89.67**

Table 4.8: Statistical significance of Experiment 4 using paired t-test with $p < 0.01$. Each cell indicates if the algorithm listed in the column performed significantly better (+), significantly worse (−), or not significantly different (0) when compared to the algorithm listed in the row. The first value is the significance of the instrument ($n = 25$) experiment and the second shows the family ($n = 3$) experiment.

Algorithm	NB	BN-F	BN-T	BN-FT	SVM-L	SVM-Q	*k*-NN
NB	—	+/+	+/+	+/+	+/0	+/+	+/+
BN-F	−/−	—	−/0	0/+	−/−	−/0	−/+
BN-T	−/−	+/0	—	+/+	−/−	−/0	0/+
BN-FT	−/−	0/−	−/−	—	−/−	−/−	−/+
SVM-L	−/0	+/+	+/+	+/+	—	0/+	+/+
SVM-Q	−/−	+/0	+/0	+/+	0/−	—	+/+
k-NN	−/−	+/−	0/−	+/−	−/−	−/−	—

Table 4.9: Confusion matrices for the family identification on the MIS dataset, showing classification counts. Bold values indicate a correct classification.

Algorithm	S	B	W	← classified as
NB	**1652**	425	450	String
	27	**403**	130	Brass
	99	76	**1259**	Woodwind
BN-F	**2013**	239	275	String
	12	**438**	110	Brass
	129	57	**1248**	Woodwind
BN-T	**1962**	157	408	String
	17	**413**	130	Brass
	110	26	**1298**	Woodwind
BN-FT	**2256**	41	230	String
	35	**413**	112	Brass
	144	11	**1279**	Woodwind
SVM-L	**2293**	78	156	String
	225	**183**	152	Brass
	486	32	**916**	Woodwind
SVM-Q	**2427**	41	59	String
	211	**286**	63	Brass
	338	48	**1048**	Woodwind
k-NN	**2303**	74	150	String
	18	**501**	41	Brass
	102	82	**1250**	Woodwind

the our results on the EastWest dataset when considering a smaller dataset (see Figure 4.5).

4.4 Discussion

Many previous approaches, such as [105], reported the greatest difficulty with classifying string instruments over any other type of instrument. In our experiments, the Bayesian network models, however, had the greatest difficulty with woodwind instruments, although the Bayesian model still outperformed both SVMs and k-NN on the woodwind dataset in Experiment 2. All algorithms tested perform extremely well on the percussion set, given the pronounced attack and immediate decay of these types of instruments, consistent with results from the literature.

The BN-FT model achieves comparable accuracy on both the instrument classification problem (n=24) and the family identification problem (n=4) on the EastWest dataset. However, the BN-F and BN-T models each achieves better accuracy on individual instrument classification than they achieve on family identification. This result suggests that neither the frequency nor time dependencies themselves are sufficient to generalize across musical instrument families, but the combination of both sets of dependencies are needed. For both datasets, k-NN achieves much higher accuracy on the family identification problem compared to the instrument identification problem, unsurprisingly, since k-NN is known not to scale well as the number of classes increases [247]. Although the results of k-NN and the BN-FT model are competitive on the EastWest dataset (n=4), k-NN outperforms the BN-FT model on the family identification task on the MIS dataset (n=3).

As shown in Tables 4.5 and 4.9, the Bayesian models more often confuse brass for woodwind instruments compared to either string or percussion. This is perhaps unsurprising as our feature extraction scheme sought to capture the conditional relationships of changes in amplitude of frequencies over time. Woodwind and brass instruments are both classified as aerophones, instruments that generate sound by vibrating air, under the [248] scientific classification of musical instruments, suggesting that our feature extraction scheme may better model the physical acoustics of the instruments.

As Deng *et al.* notes, the choice of feature extraction scheme is crucial to the success of any music instrument classification system [36]. Previous attempts to classify musical instruments have relied upon feature extraction schemes common in speech processing, most commonly the Mel-frequency cepstral coefficients (MFCC). Agostini *et al.* used a sparse set of nine spectral features to achieve 78.5% and 69.7% accuracy classifying 20 and 27 instruments, respectively, using an SVM [105]. Our feature extraction scheme, using 200 time and frequency varying features, achieved 93.6% accuracy classifying 24 instruments also using an SVM. Although not directly comparable, these results imply that our feature extraction scheme better handles more instrument classes. While our system employs a considerably larger feature set, both feature extraction schemes are bounded by the $O(n \log n)$ time complexity of the fast Fourier transform, where n is the number of samples in the audio file. Therefore we find no disadvantages in using a larger feature set.

The goal of the experiments presented in this chapter is to explore the utility of statistical dependencies of the features in both the time and frequency domains. In these experiments, the structure of the Bayesian models are tied to the feature extraction scheme employed. Therefore it is not possible to compare our feature extraction scheme to other schemes common in the literature using the Bayesian networks. Our experiments independently demonstrated the success of Bayesian classifiers on both the EastWest and MIS datasets. Livshin *et al.* noted the importance of cross-database comparison [34]. The examples in the MIS dataset are longer in duration than those in the EastWest dataset. Because our feature extraction scheme relies on temporal and frequency partitions, a cross-database comparison is not possible as the features do not align between the two datasets. In Chapters 6 and 7, we refine our feature extraction scheme to allow for alignment of musical partials in an approach designed to be extensible to feature extraction from polyphonic signals. Additionally, we focus on cross-dataset validation in our subsequent experiments in Chapters 8 and 9.

Anecdotally, our results, when compared to previously published results, indicate the value of our feature extraction scheme's ability to define statistical dependencies between features. Perhaps the feature extraction schemes that are optimized for speech recognition tasks may not be optimal in the musical instrument recognition task. Furthermore, these results also indicate that statistical dependencies modeling the changes in amplitude of partials over time, inspired by the human perception of timbre, are also useful in computational models.

4.5 Conclusion

In this chapter, we have presented a method for feature extraction, inspired by the psychoacoustic definition of timbre, that attempts to generalize the timbre of musical instruments probabilistically rather than rely on feature extraction schemes standard in speech recognition tasks. Furthermore, we demonstrate that modeling conditional dependencies between both time and frequency (BN-FT) improves classification accuracy over either dependency individually (BN-F, BN-T) or none at all (NB).

This chapter introduces the use of Bayesian networks for monophonic instrument classification as well as a novel topology, the grid-augmented naïve Bayes model. The experiments presented here demonstrate that Bayesian networks are a valid approach to the classification of individual musical instruments. Overall, the BN-F, BN-T, and BN-FT models outperformed naïve Bayes, both SVMs, and k-NN. In addition to outperforming the SVMs and k-NN, the Bayesian models achieved desirable accuracy with far fewer examples and with less execution time, albeit with a larger feature space than other approaches in the literature.

This early work in solo instrument classification heavily influenced the approaches presented in the reminder of this dissertation. In our subsequent work, we moved away from a self-generated dataset to instead focus on any database mentioned in the literature that is publicly obtainable. These datasets are discussed in detail in Chapter 5. Our feature extraction scheme presented in this Chapter, like most approaches for monophonic classification, is not scalable to the more complex polyphonic classification problem because we do not attempt to estimate any source separation. Guided by these preliminary experiments, we adapt our feature extraction approach to scale to the polyphonic mixture problem and learn a unique feature space for each musical instrument, capturing a feature space designed to minimize interference between spectral peaks of contributing sources. Our binary-relevance feature extraction scheme is discussed in Chapters 6 and 7. Lastly, we observed the need to validate any approach across multiple datasets in order to demonstrate that the feature extraction scheme indeed captures information about musical timbre rather than specifics of a single dataset. In this dissertation, we focus on demonstrating the generalizability of our approach, shown through our experiments in Chapters 8 and 9.

CHAPTER 5

DATASETS AND SIGNAL PROCESSING

Many of the approaches to music instrument classification are marred by significant limitations in the availability of data. Many studies consider only small datasets, containing only a limited number of instruments each with only few audio examples and rarely consider examples played by multiple performers, different instrument models, or differences in dynamic levels of the notes [35]. Rarely do studies consider different dataset sources which inevitably contain differences in recording procedures, equipment, and levels, although there are a few exceptions [249].

Livshin and Rodet demonstrated that many approaches to musical instrument classification do not generalize from one dataset to another [34]. The authors considered five datasets: IRCAM Studio Online (SOL), University of Iowa (IOWA), McGill Master Samples (McGill), and two small samples collections Pro and Vi collections. The IOWA and McGill datasets corresponds to our datasets MIS and MUMS, respectively, and the other three datasets were not available to us. Considering seven instruments, the authors performed cross dataset evaluations and discovered accuracies of 20% to 60% when training on one dataset and testing on another, despite classification results of over 90% for any single dataset using cross-validation (see Table 5.1).

If the goal of the Music Information Retrieval community is to develop means to automatically identify timbre from real-world recordings and commercial data collections, approaches to musical instrument identification must be designed with the goal of generalizability across multiple datasets recorded under differing conditions.

Table 5.1: Cross dataset validation results given by Livshin and Rodet [34]. The values along the diagonal, shown in parentheses, represent cross-validated self classification. The other results indicate cross-dataset validation, showing the dataset used to train in each row and the testing dataset listed in each column.

classified by	**SOL**	**IOWA**	**McGill**	**Pro**	**Vi**
SOL	(98.24)	39.93	20.14	21.51	58.17
IOWA	51.43	(97.75)	35.22	29.17	58.42
McGill	51.76	51.76	(60.78)	23.53	48.23
Pro	54.43	41.77	26.58	(48.04)	58.86
Vi	63.45	48.59	30.12	20.88	(64.42)

5.1 Dataset Sources

For this dissertation, our goal was to find multiple and varied datasets that contain a large set of musical instruments in common. The data used in the experiments were obtained from four different sources and contain 13 different musical instruments in common. Although a common practice in studies of human perception [250] and machine classification [107] of timbre, we chose not to use any artificially synthesized musical instrument samples, nor any examples generated by interpolating musical samples. This section discusses the original sources of the datasets used in this work.

5.1.1 McGill University Master Samples

The McGill University Master Samples (MUMS) is an 11 volume collection of compact discs, published 1987-1989 [251, 252]. MUMS is a library of isolated sample tones from a wide number of musical instruments, including most standard orchestral instruments as well as some popular musical instruments. Each instrument was recorded separately at a 24 bit, 44.1 kHz sampling rate. The collection covers the entire pitch range of each of the 34 instruments in the collection. This collection contains examples of musical scales, each played at a single dynamic level and is the smallest of our four datasets used in our experiments (see Section 5.2). Although once widely used in the musical instrument classification literature, the data sources were sold by McGill University to a commercial entity and the dataset is no longer available for purchase. For this work, we obtained copies of the published compact disc collection from an academic library archive. Portions of the MUMS have been used in a few polyphonic studies [151, 163, 198, 214].

5.1.2 Iowa Musical Instrument Samples

The University of Iowa Musical Instrument Samples (MIS) dataset was created by the Electronic Music Studios at the University of Iowa, beginning in 1997, and the collection was significantly expanded in 2011. These examples were recorded with a Neumann KM 84 cardioid condenser microphone in an anechoic chamber at the Wendell Johnson Speech and Hearing Center at the University of Iowa, and stored as 16 bit, 44.1 kHz sampling rate audio files [241]. Some of the examples recorded after 2011 were recorded in an ultra-high quality 24 bit, 96 kHz stereo format.

The samples are organized into chromatic scales played at *pp*, *mf*, and *ff* dynamic levels throughout the full range of each of the 23 instruments. Some instruments were performed with more than one technique, including arco, pizzicato, vibrato, and non-vibrato. The Iowa dataset is freely available to download from the University of Iowa Electronic Music Studios[1] and has been used in several polyphonic classification studies [151, 198, 193, 214].

[1] http://theremin.music.uiowa.edu/MIS.html

5.1.3 Real-World Computing Collection

The Real World Computing (RWC) Music Database is a database created by the RWC Music Database Sub-Working Group of the Real World Computing Partnership (RWCP) of Japan. The RWC Music Database is a large-scale music database compiled specifically for research purposes and includes six separate music datasets:

1. Popular Music Database (100 songs)
2. Royalty-Free Music Database (15 songs)
3. Classical Music Database (50 pieces)
4. Jazz Music Database (50 pieces)
5. Music Genre Database (100 pieces)
6. Musical Instrument Sound Database (50 instruments)

In this work, we use the Musical Instrument Sound Database which contains a total of 50 different instruments. In addition to the common western orchestral instruments, the collection also contains many traditional Japanese instruments. The samples were recorded as CD quality audio files, 16 bit, 44.1 kHz sampling rate, and contain instruments playing chromatic scales through the range possible for the instrument.

The collection features each instrument playing a variety of playing styles, dynamics, instrument manufacturers, and musicians. Each instrument set contains three different instrument manufacturers each played by a different musician. Each contains a variety of techniques or articulations, each played at three dynamic levels: *pp*, *mf*, and *ff*. The entire instrument dataset contains 29.1 Gigabytes containing 91.6 hours of audio [253, 254]; however, we use only a subset of the musical instruments in our experiments. Although the RWC dataset is a copyrighted collection, it is made available for research purposes for the costs of duplication of media and shipping and handling. However, because there is a nominal cost required to obtain the collection, the RWC dataset is less widely used in the literature than the aforementioned free collections, despite the fact that the RWC collection is the largest source of musical instrument samples currently available and therefore is increasing reported in the polyphonic classification literature [193, 198, 213, 215, 216, 232, 234, 235].

5.1.4 Philharmonia Orchestra Sound Sample Collection

The Philharmonia Orchestra Sound Sample Collection (PHO) is a collection of recordings of various musical instruments by the Philharmonia Orchestra in London [255]. The samples are licensed for use under the Creative Commons Attribution-ShareAlike License. The collection contains 18 common Western orchestral instruments as well as numerous percussion instruments. The collection features instruments playing notes of varying lengths, dynamics, and articulations. Unlike the other three data sources, the PHO dataset provides recordings of individual notes rather than musical scales, allowing omitting the file split step described in Section 5.3.1 for this dataset.

The files are released in an MP3 format at a 44.1 kHz sampling frequency with a bitrate varying between 64 and 96 kilobytes per second. Unlike the CD quality audio of the other three datasets, these samples are of much lower audio quality. For consistency of file format among the datasets, we convert these files to wav files. However, it is important to note that this conversion process does not change the underlying lower quality of the audio in this dataset. To our knowledge, this dataset has not previously been used in any instrument classification studies.

Table 5.2:
List of 13 instruments common the MUMS, MIS, RWC, and PHO datasets and the number of examples per instrument in each dataset.

Family	**Instrument**	**MUMS**	**MIS**	**RWC**	**PHO**
Brass	French Horn	74	55	1889	587
	Trumpet	97	211	1972	416
	Trombone	69	82	2743	796
	Tuba	38	65	480	954
Woodwind	Flute	66	227	1084	813
	Clarinet	37	139	1434	793
	Alto Saxophone	14	193	1098	678
	Oboe	32	91	773	557
	Bassoon	32	64	1405	666
String	Violin	152	832	2862	800
	Viola	144	583	2730	860
	Cello	150	658	2629	825
	Contrabass	138	671	3117	777
Total		**1043**	**3871**	**24,216**	**9522**

5.2 Common Instrument Set

Since a goal of this work is to validate the approach between different datasets, we chose the set of instruments common all four datasets. Although these datasets each contain additional instruments, only the 13 instruments shown in Table 5.2 are present in all four datasets.

5.3 Pre-Processing

If in a higher quality format than CD quality audio, each sound file is downsampled to a 44.1 kHz sampling rate with 16-bit per sample. Additionally, the examples are mixed down to a single channel waveform if provided in a stereo track, such as many from the MIS dataset. For the lower quality PHO dataset, the audio was upsampled from lower quality MP3 format to the aforementioned compact disc quality.

5.3.1 Splicing the Audio Files

Most of the original source files contain performers playing chromatic musical scales of individual notes separated by moments of silence, with the exception of the PHO dataset. The audio utility *SoX* [242] is used to detect the silence and split the recordings into individual files, in which each file represents a single, isolated musical note. All resulting files are checked to ensure they contain audio data and are not silent files. Any tracks that do not contain any signal above the amplitude threshold of -45 dB are flagged as silent and deleted. Among the resulting files, any silence detected before or after the musical note was trimmed. This procedure aligns the attack of each note to appear at the beginning of the sound file.

5.3.2 File Length

The various audio files differ in length of note played. Some notes, such as those played with a staccato articulation, are as short as 0.2 seconds. Other notes are much longer, up to several seconds in length. Since the resolution of the Fourier transform scales with the length in time of input file, all sound files were set to be the same length. This yields a consistent frequency resolution across all examples used in the experiments. All files were set to be 1.0 second in length. If the musical note is shorter than one second, silence was added to lengthen the file to be 1.0 seconds. If the musical note sample is longer than one second, the file was trimmed to be exactly 1.0 second. Next, a fade in of 10 milliseconds and a fade out of 10 milliseconds was imposed to eliminate any potential discontinuities in the waveform resulting from the trimming in the previous step.

In this work we examine spectral features in a single, static time window. In future work we will examine temporal models that capture changes in the features over time Numerous studies have emphasized the importance of the note's attack in timbre recognition [106, 249]. The process described here ensures each audio example contains the note's attack, even if the final decay and release of the note was not considered.

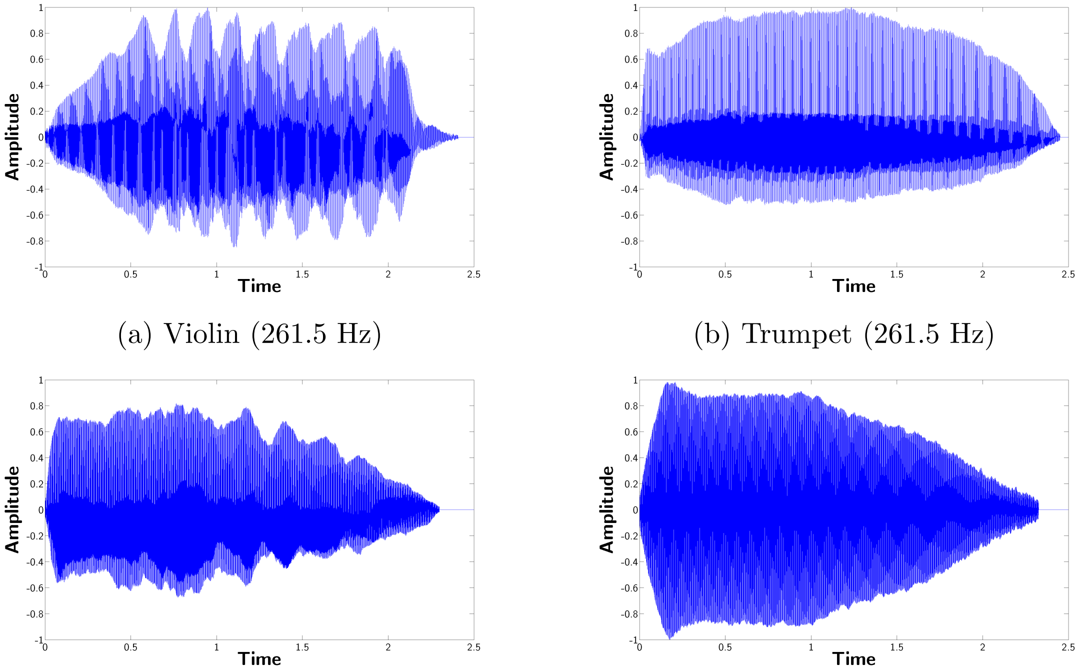
5.3.3 Volume Normalization

All sound files are then batch normalized to the range [0,1] using the audio utility *normalize*[2]. Within each dataset and for each instrument, the loudest gain in any of the files is scaled to a value of one and the volume of all other files are adjusted respectively. This batch normalization approach preserves the relative dynamic levels between all the examples for each instrument within each dataset.

5.4 Signal Processing

Following the pre-processing procedures in the previous section, each audio file consists of a single musical note with an immediate attack at the start of the audio file. Figure 5.1 shows examples of time domain waveforms for four different instruments. In this work, we are concerned with the extraction and analysis of spectral features for use in machine learning experiments. In order to extract meaningful spectral features from the audio file, each audio sample must be transformed from a time

[2] http://normalize.nongnu.org/



(a) Violin (261.5 Hz)

(b) Trumpet (261.5 Hz)

(c) Flute (440 Hz)

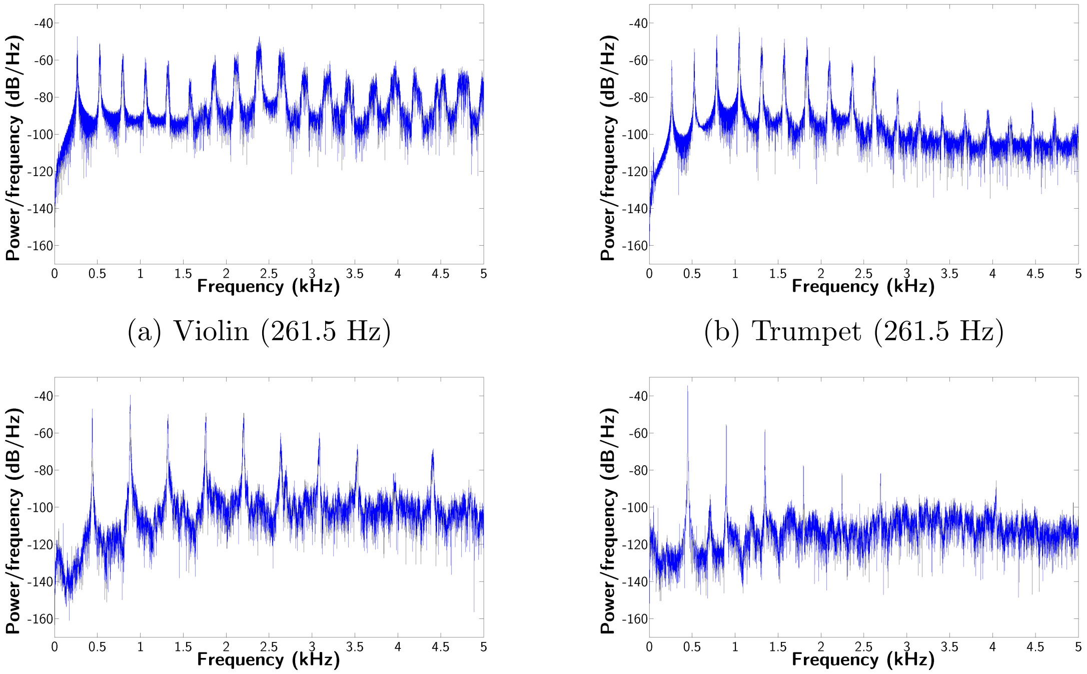
(d) Clarinet (440 Hz)

Figure 5.1: Time domain views of four different musical instruments

domain waveform into a frequency domain view using a Fourier transformation (see Section 2.1.4.3).

5.4.1 Fast Fourier Transform

On all audio examples, an FFT with a single time window the entire length of the recording transforms the waveforms to the frequency domain. This transformation estimates the energy levels of each of the frequency components of the signal, returning a set of the relative energy levels indexed by frequency. Figure 5.2 show the frequency-domain views of the spectra for the waveforms shown in Figure 5.1.



(a) Violin (261.5 Hz)

(b) Trumpet (261.5 Hz)

(c) Flute (440 Hz)

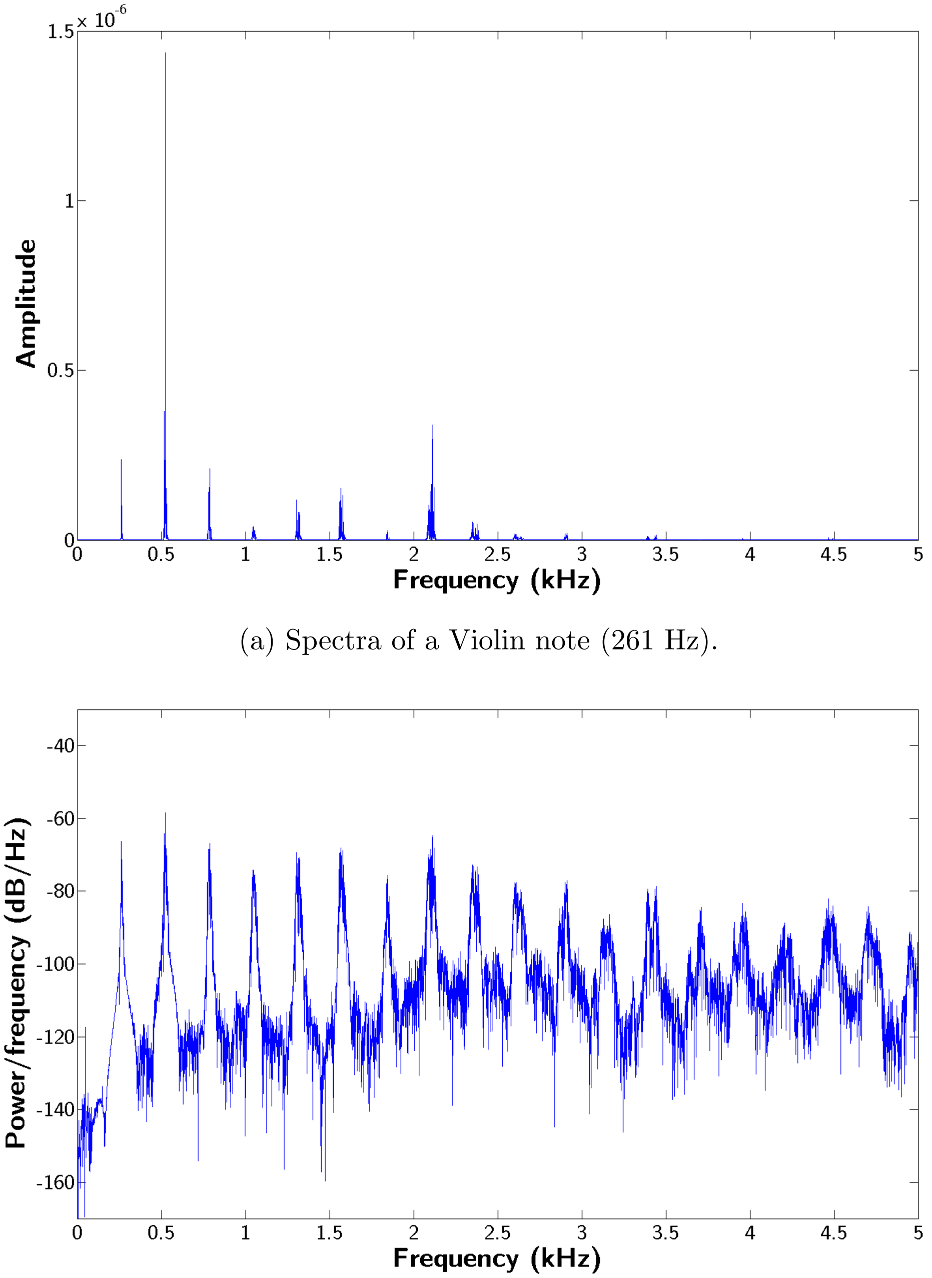
(d) Clarinet (440 Hz)

Figure 5.2: Frequency domain views of four different musical instruments

5.4.2 Amplitude Scaling

On a linear scale, the amplitude levels of the upper harmonics appear irrelevant compared to the dominating lower harmonics (Figure 5.3a). Because these peaks do have significant energy, relative to local frequency neighborhoods, throughout this work we consider a compression and use the logarithmic power spectral density of each amplitude, a common practice in the field [213]. This transformation scales the amplitudes by $10 \cdot \log 10$, as shown in Figure 5.3b.

Throughout this work, all sound examples are processed in the manner described in this section. We use this spectral data to learn instrument specific locations for feature extraction, discussed in Chapter 6, and for feature extraction from training and testing classifiers in Chapter 7.



(a) Spectra of a Violin note (261 Hz).

(b) Spectra of the same Violin note, showing amplitudes on a logarithmic scale.

Figure 5.3: Waveform and spectra of a Violin note.

5.5 Binary-Relevance Datasets

In this work, we examine a binary-relevance (BR) approach to multilabel classification discussed in Section 2.2.2.3. Therefore we train a classifier to identify the presence or absence of an individual instrument in an audio example, creating a set of k models for k instrument classes. For an audio signal containing an unknown instrument, each of the k models classify the signal and the signal is classified as the containing the instrument(s) corresponding to the instrument model that returned the highest confidence.

To train and test the instrument-specific binary classifier, we organize the datasets into BR datasets for each individual instrument. These are datasets containing only examples of single instruments, and a separate dataset is created for each instrument. Each instrument-specific dataset contains an equal number of examples of the target instrument and other instruments.

More precisely, for each instrument i, we create a dataset D_i in which 50% of the dataset are examples of instrument i, assigned the positive class (+) label. The other 50% of the examples in the dataset are examples of other instruments, that is, any instrument $\neg i$, which is assigned the negative class label ($-$). To select examples for the negative class label, we randomly select one of the other twelve instruments and then randomly select a sound example of the chosen instrument.

The files were chosen without regard to the dynamic level, and the binary-relevance datasets contain samplings of all the possible dynamic levels available from the data sources (see 5.1). Additionally an instrument-specific BR dataset D_i is created for each of the four data sources, and the BR datasets do not mix examples of the same instrument between datasets. Chapter 8 presents cross-dataset experiments in which a classifier is trained with one dataset but tests on another dataset.

Since the number of examples of instrument i available differs between instruments and datasets, the total size of each dataset D_i is twice the value given in Table 5.2 for each instrument and dataset. When training instrument-specific BR classifiers, features are extracted using the cluster signature for the positive instrument class (see Chapter 6).

5.6 Polyphonic Datasets

In Chapter 9, we apply our feature extraction approach to classify audio mixtures containing two to four different instruments playing simultaneously. Since the data sources used in this work consist of examples of solo instruments, we create polyphonic datasets containing mixtures of sets of solo instruments.

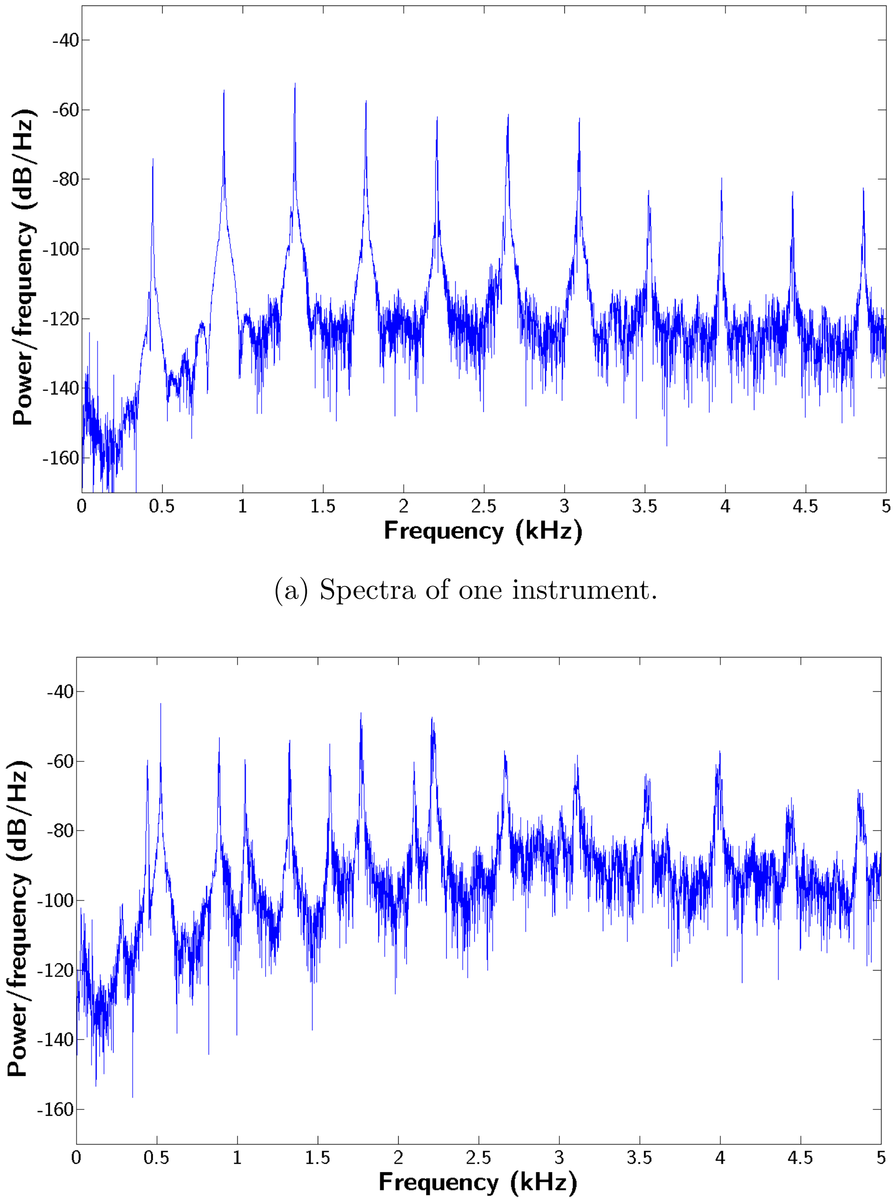
We begin by creating a dataset for duet mixtures of two instruments. For each instrument i, we create 1000 mixtures containing instrument i and another instrument $\neg i$, chosen randomly from the same data source. Each signal is scaled by 0.5 and added together. The resulting mixture is then normalized so that the single largest amplitude is 1.0 and the other amplitudes are scaled accordingly. The process prevents clipping of the mixed audio signal while preserving the relative difference in dynamic level between the two examples being mixed.

This process is repeated for each instrument, resulting in 13,000 mixtures for each of the four data sources. Instruments are not mixed between the four data sources. Each of the four polyphonic datasets contains at least 1000 examples of each instrument as well as variable number of additional examples, as the instrument was chosen at random during the aforementioned selection process.

We repeat this approach for mixtures of three and four instruments. Figure 5.4 shows the spectra of four examples, ranging from a solo note to a mixtures of two, three, and four instruments. The mixtures contain at most one instance of an individual instrument, that is to say, no mixture contains two examples of the same instrument. The examples mixed were selected without regard to the dynamic level, so the resulting set of mixtures contain examples of various combinations of dynamics. Preserving the relative dynamics of the source signals makes the polyphonic classification task more difficult, as the quieter instrument signal is more likely to be dominated by the louder instruments – especially in cases of constant musical intervals that result in interference of the harmonics. However, this better represents the challenges of real-world musical data.

5.7 Summary

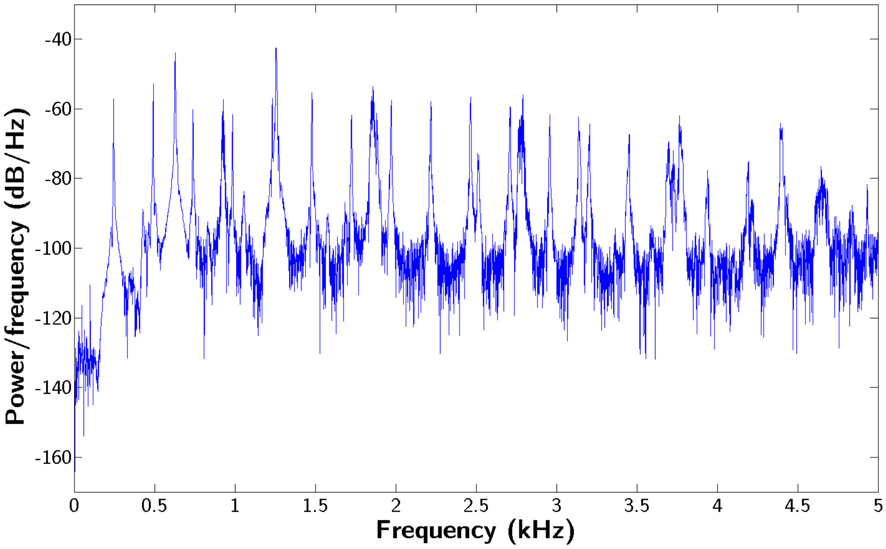
In this chapter we discuss our sources for data for our later experiments, including discussion of our pre-processing and signal processing procedures. We propose the largest collections of datasets ever considered in a single- or multi-label classification task. Lastly, we discuss our binary-relevant datasets, including monophonic datasets for training and testing the the experiments presented in Chapter 8 and the polyphonic datasets with mixtures of two, three, or four instruments, used in the experiments in Chapter 9.



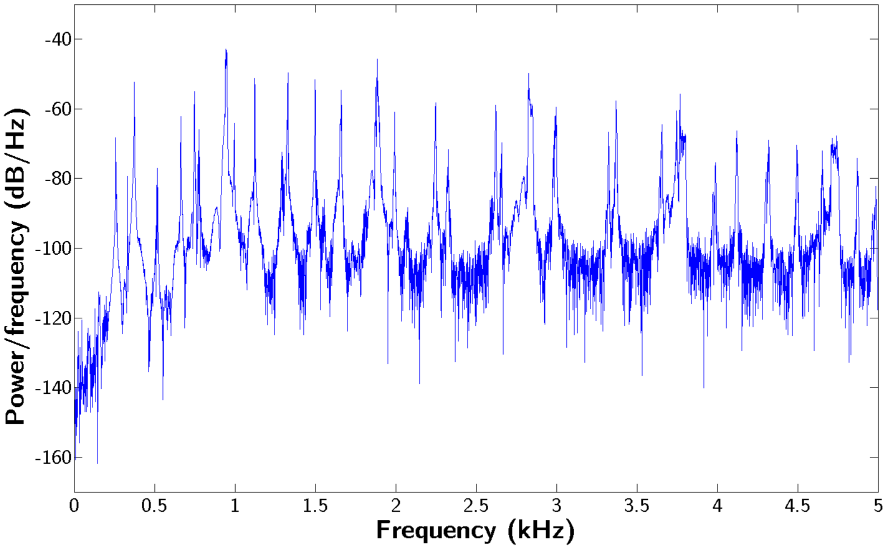
(a) Spectra of one instrument.

(b) Spectra of two instruments mixed together.

Figure 5.4: Frequency domain view of mixtures of one to four instruments.



(c) Spectra of three instruments mixed together.



(d) Spectra of four instruments mixed together.

CHAPTER 6

INSTRUMENT SIGNATURES

In this chapter, we outline an approach for binary-relevance feature extraction for the task of musical instrument classification. In Section 6.3, we propose a data-driven approach to training instrument-specific spectral filters for use in feature extraction in the classification of musical instruments. In Section 6.4, we present experimental results demonstrating that the feature spaces learned for instruments from any one dataset can be successfully used in feature extraction to classify the other datasets.

6.1 Motivation

In polyphonic mixtures of instruments, the harmonic partials of individual tones are interleaved, and many of the feature extraction approaches that are successful for classification of solo instruments are not extensible to polyphonic classification. These feature extraction techniques often take summary statistics of the spectral energy over large frequency ranges, unnecessarily grouping harmonics from different sources in cases in which the contributing harmonics are near in frequency but not necessarily overlapping.

Even among the binary-relevance approaches to multilabel classification of polyphonic mixtures (see Section 3.2), investigators use the same feature space for all the instrument-specific binary classifiers. However, because of the nature of BR classification, there is no requirement to use the same number of dimensions for each classifier, nor the same types of features for each binary classifier. Researchers usually use the same feature space for all binary classifiers because the features need only be extracted once for any unknown example to be classified, even though each binary classifier must be queried. However, some studies suggest that features optimized for a specific instrument increase recognition rates [198].

In this work, we propose a binary-relevance approach to feature extraction for the multilabel classification of polyphonic mixtures. We present an approach to learn instrument-specific locations of regions of spectral prominence, relative to each instrument studied. This procedure permits a feature extraction technique catered to each instrument and each BR classifier to use a different feature space. This approach does come with the burden of requiring a feature extraction step for each binary classifier when classifying an unknown example, a complexity avoided by the aforementioned approaches that use only a single feature space.

6.2 Binary-Relevance Feature Extraction

In the task of feature selection optimization, a relevance subset of features is selected from the set of all possible features. Such a feature subset may maximize some criteria, such as the independence between selected features or the efficacy of a classification algorithm using the subset of features. This chosen subset of features would then be used in subsequent learning experiments while the features not selected would be discarded.

Feature selection approaches are extended to multilabel data such that the selection algorithm searches for a subset of features that optimize a multilabel loss function [37]. In other words, the algorithm selects a subset of features that seem to work the best for the greatest number of labels. In the field of text categorization, a common approach uses a binary-relevance transformation in order to evaluate the discriminative ability of each feature with respect to each individual label. These scores are aggregated in order to provide rankings of the feature's utilities [256].

In feature selection, for both single label and multilabel classification, the algorithms select a single subset of features from a set of possible features and the feature extraction stage will use this same subset to extract features from all the examples. While this approach is necessary for multiclass classification and for the algorithm adaptation approach to multilabel classification, in binary-relevance multilabel classification each individual classifier need not share the same a feature space common to all classifiers.

In this work, we propose a data-driven feature selection approach for supervised binary-relevance multilabel classification in which each binary classifier g_i for some class label i is associated with its own feature vector $\mathbf{X}_i$. Feature vectors $\mathbf{X}_i$ and $\mathbf{X}_j$, for class labels i and j respectively, need not share the the same feature space nor contain the same number of features, i.e., $|\mathbf{X}_i| \neq |\mathbf{X}_j|$. In a supervised learning context, we propose using training examples for a specific class label i to learn a subset of features that best discriminate examples with label i from those with label $\neg i$. This approach will permit optimizing a unique subset of features for each binary classifier.

Compared to the traditional feature extraction approach for multilabel classification, this proposed approach does carry the additional overhead that features must be extracted multiple times from each example, once for each binary classifier. Another potential disadvantage of this approach is that it may exacerbate problems stemming from the label independence assumption of the binary-relevance approach.

6.3 Learning Instrument Signatures

Since we are using instrument-specific binary models for each different instrument, we are concerned with studying the feature spaces on individual instruments. Therefore, for the signature learning stage we examine audio examples containing only a single instrument per file.

In this section, we describe our approach to locate and extract the harmonics with significant spectral energy specific to each instrument and dataset. First, we transform the audio signals to the frequency domain using an FFT with a single time window (see Section 2.1.4.3). The resulting amplitudes are scaled by $10 \cdot \log 10$ dB to a power/frequency scale as described in Section 5.4.2.

6.3.1 Spectral Threshold

Spectral peak detection is necessary to detect and extract the harmonics in the spectra. In this stage, we identify the areas of the spectra that contain peaks of significant energy and must identify a baseline threshold of significance above the noise floor. Because the upper harmonics of a musical sound contribute less energy than their lower counterparts, we consider a frequency-dependent threshold that permits identifying peaks as significant to their local frequency neighborhoods, rather than compared to all peaks of the entire frequency range.

For each example analyzed, we calculate the threshold, identify all spectral peaks with amplitudes that exceed this threshold, and note the frequency locations of these spectral peaks. The first step is to establish a threshold above the noise floor. In order to capture the amplitudes of higher harmonics, despite the roll-off found at higher frequencies, we desire a threshold that considers amplitude values relative to the amplitudes at nearby frequencies.

We employ the thresholding strategy presented by Every and Szymanski [257]. First, a smoothed amplitude envelope E is calculated for each example by convolving the spectra F with a moving Hamming window h of length $256 + 1$ samples in which each value of E_j is set to be the average of the window with center point j. The moving Hamming window permits capturing the amplitude at each frequency relative to a small local frequency range and captures the contributions of the upper harmonics of the spectra.

The frequency-dependent threshold for each frequency bin j is calculated as

$$\hat{E}_j = e_{th} \cdot (E_j)^c \tag{6.1}$$

where c is a constant $[0.5, 1)$ that determines the flatness of the envelope shape and e_{th} is a frequency independent threshold height. The parameter e_{th} is defined as

$$e_{th} = b \cdot \mid \overline{F} \mid^{1-c} \tag{6.2}$$

where $\overline{F}$ is the average amplitude across all frequency bins and b is a positive scalar that raises the mean above the noise floor. We choose $c = 0.5$ to produce a flatter envelope and a value of $b = 2$ in all our experiments. An example spectrum with threshold $\hat{E}$ is shown in Figure 6.1.

This frequency-dependent variable threshold permits identifying peaks as significant to their local frequency neighborhood, allowing capture of significant peaks even in the higher frequency range.

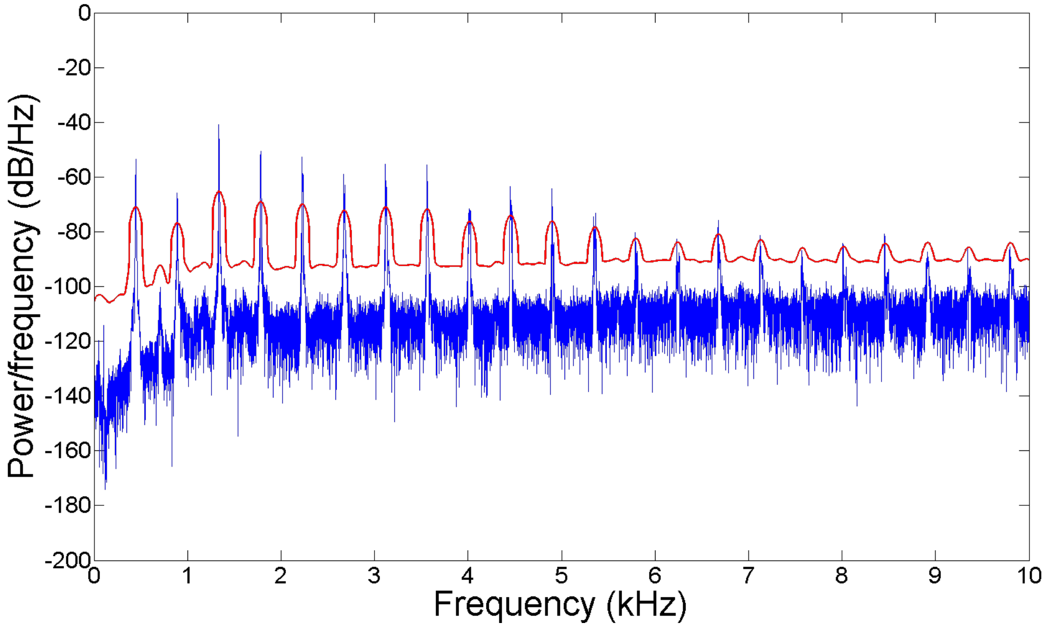


Figure 6.1: Amplitude spectrum of a Clarinet playing the note 440 Hz overlaid with threshold.

6.3.2 Fundamental Frequency Identification

Next, we identify the fundamental frequency f_0 in the signal. Since the signature learning stage requires learning from examples containing only a single instrument, we can assume the fundamental is the significant peak with the lowest frequency, within a localized frequency neighborhood. We employ a näive f_0 finding algorithm in which

$$f_0 = \underset{j}{\operatorname{argmin}}\{\operatorname{argmax}\ w_k \cdot F \mid w_k \cdot F_j > \hat{E}_j\} \tag{6.3}$$

where moving frequency window

$$w_k = \begin{cases} 1 \text{ if } j - 16 \leq k \leq j + 16 \\ \\ 0 \text{ if otherwise} \end{cases}$$

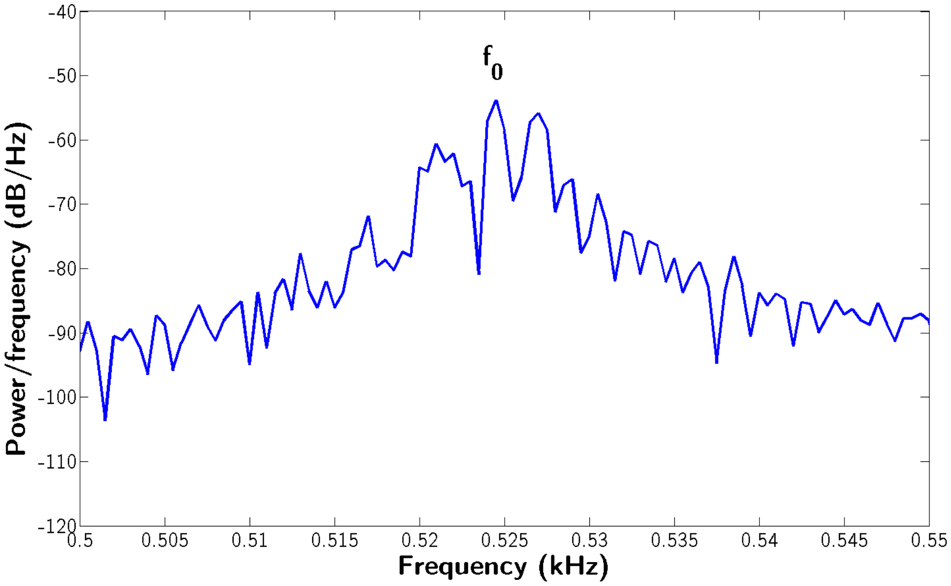


Figure 6.2: Highly zoomed view of the fundamental frequency of a Trumpet playing 527 Hz. The highest peak shown represents f_0 and the other local peaks are side-lobes resulting from the fast Fourier transformation.

First, we find the largest amplitude value within a small frequency window of 33 samples. The window is of an odd length for symmetry. Considering a small window allows capturing the maximum peak in the local frequency neighborhood, rather than a local maximum of a peak's side-lobe, such as those shown in Figure 6.2. The moving window w_k, centered on frequency j, is considered for all frequency values $0 < j \leq N$, and for each the peak with the maximum amplitude is considered if that peak also exceeds the threshold $\hat{E}_j$ as a potential f_0. We identify the fundamental frequency f_0 as corresponding to the frequency of the peak with the lowest frequency from the set of potential f_0s.

6.3.3 Spectral Peak Identification

The next step is to extract any amplitude peaks that exceed the threshold $\hat{E}_j$ and note the frequency location of each peak. In this stage, we are concerned with locating each significant peak relative to the fundamental frequency.

For each example, we locate each peak p that exceeds the threshold, $F_j > \hat{E}_j$, for all frequency bins j up to the Nyquist limit and save these values as vector $\mathbf{p}$. For each peak $p \in \mathbf{p}$, its ratio to f_0 is calculated as $r = p/f_0$. Any ratio $r > 64$ is discarded and the rest are saved in a vector of ratios $\mathbf{r}$. At this stage, the amplitude value in bin F_j is discarded because we are interested in identifying the ratios of these spectral peak locations to the fundamental frequency.

We repeat this process for all files for each musical instrument and save all ratios in a single dimension vector, with duplicate values allowed. By capturing the ratio of peak to fundamental, rather than absolute frequency values, we normalize away the pitch of the note, allowing direct comparisons between notes with different pitches.

6.3.4 Clustering Significant Peaks

In this stage, we cluster the vector of ratio data in order to learn the locations of various harmonic locations important to each instrument. k-means is a common clustering algorithm that partitions a set of n observations into k discrete clusters so that every observation is assigned to the cluster with the nearest mean [71]. For each instrument, we use k-means to partition the set of ratios into a set of Gaussian clusters, given as Algorithm 6.1.

We begin with an initial k=10 clusters. Since musical instruments contain a quasi-harmonic pattern of partials at near integer ratios, we expect that many clusters will contain means near integer ratios of the fundamental. We seed the initial k clusters with integer values $[2 \dots k+1]$, corresponding to the first ten overtones above the fundamental. We modify the traditional k-means to permit changing the number of clusters as the algorithm progresses. At each iteration, if the standard deviation exceeds a threshold, the cluster is split into two different clusters. We use a threshold of $\sigma = 0.5$, which represents the halfway point between two quasi-integer ratios and has precedence in the literature [35]. Likewise, if the means of two individual clusters overlap by less than $\sigma = 0.5$, the two clusters are combined into one. This method yields a variable number of clusters for each instrument and dataset.

For each cluster, we extract the mean and standard deviation. The mean μ of each Gaussian cluster indicates an important spectral location, and the standard deviation σ captures the variance in frequency of the harmonic over the set of examples for that instrument. Although the majority of the ratios learned are near-integer ratios, many clusters learned center around inharmonic ratios (such as $\mu = 11.50$).

Having learned a set of clusters, we return a spectral signature for each instrument, for each dataset. In the feature extraction stage of the experiments, the instrument signature is applied as a spectral mask. Only the spectral energy underneath the signature will be considered for feature extraction while the rest of the spectral signal will be disregarded as noise.

In the next section, we present classification experiments that show that these signatures learned for an instrument in one dataset can be successfully used in feature extraction for the same instrument but from a different dataset.

6.4 Signature Validation

In section 6.3, we present a data-driven approach to learning the locations, relative to the fundamental, of areas of significant spectral energy for each instrument. In

Algorithm 6.1 Adaptive k-Means Cluster

```
Given: set of values $X = x_1, ..., x_n$, initial $k = 10$
Initialize clusters $C$ with centroids $\mu_1, \mu_2, \ldots, \mu_k \in \{1, 2, 3, \ldots, 10\}$
while no convergence achieved do
  {Assign each example to the cluster with the nearest mean}
  for all $x_i \in X$ do
    Assign $x_i$ to cluster $c_j$ where $\text{argmin}_j \ ||x_i - u_j||^2$
  end for

  {If cluster's standard deviation exceeds threshold, split into two clusters}
  for all $c_j \in C$ do
    if $\sigma_j > 0.5$ then
      remove cluster $c_j$ from $C$
      create new cluster $c_r$ with mean $\mu_r \leftarrow (\mu_j - 0.25)$
      create new cluster $c_s$ with mean $\mu_s \leftarrow (\mu_j + 0.25)$
      add clusters $c_r, c_s$ to $C$
      $k \leftarrow k + 1$
    end if
  end for

  {Calculate new cluster means and standard deviations}
  for all $c_j \in C$ do
    $u_j \leftarrow \frac{\sum_{i=1}^{n} 1\{c_i = j\} x_i}{\sum_{i=1}^{n} 1\{c_i = j\}}$
    $\sigma_j \leftarrow \sqrt{\frac{1}{n} \sum_{i=1}^{n} 1\{c_i = j\}(x_i - u_j)^2}$
  end for
end while
```

this section, we test the generalizability of this feature extraction approach between datasets. We show that an instrument signature learned from one dataset can be used for feature extraction for the same instrument but in a different dataset. In other words, we use the locations of the features learned for one instrument in one dataset to extract the features for the same instrument on another dataset. These experiments both train and test on audio examples containing only instrument notes.

Table 6.1: List of number of clusters learned by instrument and dataset.

Instrument	**MUMS**	**MIS**	**RWC**	**PHO**
FrenchHorn	29	60	54	88
Trumpet	47	26	44	83
Trombone	64	86	95	110
Tuba	55	120	71	87
Flute	13	38	87	106
Clarinet	39	107	56	82
AltoSaxophone	52	75	84	94
Oboe	30	43	39	30
Bassoon	82	86	96	100
Violin	52	100	71	56
Viola	59	80	72	120
Cello	87	86	102	102
Contrabass	94	100	94	108

6.4.1 Experimental Design

Given the variable k-means clustering approach (Section 6.3.4), each instrumental signature varies in the number of clusters, as shown in Table 6.1. This means that each binary-relevance classifier operates on a different feature space. Using a particular instrument signature, for each cluster in the signature we extract a single amplitude as a feature, which is described in detail in the next chapter. Therefore the number of clusters in each learned instrument signature dictates the number of amplitude features used in the experiments. Since the number of clusters learned varies between instruments and datasets, the same instrument from two different datasets will vary in size of the feature space.

In the classification experiments in this and subsequent chapters, the same instrumental signature must be used for feature extraction on both the training and test sets. However, the data used in training and testing can come from different datasets. In subsequent experiments, such as those in Chapter 8, we explore cross-dataset classification, using one dataset for training and another for testing. In these signature validation experiments, however, we use a self-classification paradigm in which we train and test on the same dataset, using a 10-fold cross-validation approach, reporting the average of the results of the 10-folds. Instead we vary the instrumental signature used for feature extraction, using the important locations learned from one dataset to extract the features for the same instrument on another dataset.

For instance, consider the task of training the binary classifier c_i for instrument i, assuming four datasets containing examples of instrument i, labeled $\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i$ and $\mathbf{D}_i$. The signature learning approach described in 6.3 learns one signature for each instrument/dataset pair, S_i^A, S_i^B, S_i^C, and S_i^D. Next we apply the signature S_i^A to each of the four datasets, $\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i$, and $\mathbf{D}_i$. The signature S_i^A is learned from the data in dataset $\mathbf{A_i}$, but is learned independently of the examples in the other three datasets. We repeat this process for the remaining signature S_i^B, S_i^C, S_i^D.

6.4.2 Feature Extraction

For each dataset, we consider the cluster signature learned for each instrument, as described above. This signature informs the locations in the signal of the features to extract. We apply this signature to each of the other datasets and extract amplitude features at that location. The details of the feature extraction procedure using the instrumental signatures are discussed in the next chapter, and we use this approach in these experiments. For any instrumental signature containing k clusters, this procedure results in k amplitude features extracted.

6.4.3 Results

In Table 6.2, we report the F_1 measure result of each binary classifier. For most instruments and datasets, we show that a signature learned from one dataset can be applied for feature extraction on another dataset. In a numerous cases, we found a higher accuracy when applying a signature from one dataset to another dataset. For example, many of the instrument signatures learned from the large, high quality RWC dataset (Table 6.2c) produced a higher score than the self-classification result of the RWC dataset itself. This likely results from the diversity of performers and dynamic levels present in RWC, but absent from others such as MUMS.

In many cases, the lowest instrument scores result from applying the MUMS signature (Table 6.2a) to the other datasets. This is the smallest dataset and represents only a single player and a single dynamic level. The inability of the MUMS signature to generalize to the other larger, more diverse datasets is not surprising, but it does underscore the need for large and diverse datasets in instrument classification tasks. Another observation is that the PHO signature (Table 6.2d) was successful when applied to the other datasets. The PHO dataset is a large, but lower MP3 quality dataset. This implies that a large number of diverse examples of each instrument, even if of lower quality, is more useful when training models than a small number of high quality examples, such as the MUMS dataset provides. These results strongly imply that our binary-relevance feature extraction technique finds features that generalize an instrument's musical timbre, regardless of the dataset.

Table 6.2: Results of the signature validation experiments showing the F-measure for each binary classifier (instrument) for each dataset. Figures 6.2a − 6.2d report the results using instrument signatures learned from each of the four different datasets, respectively. The bold results indicate the signature was learned from the same dataset that is tested.

(a) Signature learned from the MUMS dataset.

Instrument	**MUMS**	**MIS**	**RWC**	**PHO**
French Horn	**0.64**	0.70	0.64	0.76
Trumpet	**0.75**	0.63	0.82	0.73
Trombone	**0.51**	0.58	0.67	0.64
Tuba	**0.65**	0.65	0.58	0.81
Flute	**0.77**	0.75	0.71	0.67
Clarinet	**0.73**	0.57	0.78	0.71
Alto Saxophone	**0.53**	0.61	0.61	0.93
Oboe	**0.54**	0.72	0.72	0.50
Bassoon	**0.73**	0.69	0.74	0.74
Violin	**0.72**	0.61	0.58	0.63
Viola	**0.71**	0.53	0.70	0.52
Cello	**0.79**	0.73	0.75	0.73
Contrabass	**0.80**	0.89	0.79	0.84

(b) Signature learned from the MIS dataset.

Instrument	**MUMS**	**MIS**	**RWC**	**PHO**
French Horn	0.74	**0.75**	0.81	0.74
Trumpet	0.88	**0.91**	0.83	0.80
Trombone	0.72	**0.74**	0.72	0.68
Tuba	0.77	**0.88**	0.87	0.90
Flute	0.69	**0.73**	0.68	0.72
Clarinet	0.85	**0.87**	0.87	0.89
Alto Saxophone	0.75	**0.79**	0.76	0.75
Oboe	0.78	**0.83**	0.74	0.78
Bassoon	0.70	**0.67**	0.76	0.68
Violin	0.86	**0.87**	0.88	0.86
Viola	0.74	**0.74**	0.73	0.70
Cello	0.78	**0.78**	0.77	0.80
Contrabass	0.89	**0.89**	0.90	0.87

(c) Signature learned from the RWC dataset.

Instrument	**MUMS**	**MIS**	**RWC**	**PHO**
French Horn	0.71	0.75	**0.77**	0.78
Trumpet	0.71	0.69	**0.73**	0.71
Trombone	0.75	0.76	**0.76**	0.74
Tuba	0.88	0.91	**0.86**	0.90
Flute	0.78	0.77	**0.77**	0.75
Clarinet	0.90	0.86	**0.88**	0.88
Alto Saxophone	0.75	0.78	**0.74**	0.77
Oboe	0.81	0.83	**0.80**	0.80
Bassoon	0.87	0.85	**0.87**	0.85
Violin	0.86	0.84	**0.84**	0.86
Viola	0.78	0.77	**0.78**	0.75
Cello	0.85	0.84	**0.83**	0.83
Contrabass	0.92	0.92	**0.91**	0.91

(d) Signature learned from the PHO dataset.

Instrument	**MUMS**	**MIS**	**RWC**	**PHO**
French Horn	0.81	0.79	0.80	**0.80**
Trumpet	0.76	0.78	0.76	**0.77**
Trombone	0.76	0.78	0.76	**0.77**
Tuba	0.91	0.92	0.89	**0.90**
Flute	0.80	0.87	0.84	**0.82**
Clarinet	0.91	0.87	0.89	**0.87**
Alto Saxophone	0.76	0.74	0.75	**0.75**
Oboe	0.89	0.88	0.88	**0.87**
Bassoon	0.87	0.86	0.86	**0.85**
Violin	0.82	0.85	0.85	**0.82**
Viola	0.82	0.81	0.83	**0.82**
Cello	0.83	0.82	0.80	**0.80**
Contrabass	0.90	0.90	0.92	**0.90**

6.5 Conclusion

In this chapter, we present a novel approach for binary-relevance feature extraction for use with binary-relevance classifiers. This approach allows us to use a different feature space for each binary classifier. We describe a data-driven clustering approach to learn locations from each instrument's spectra that best represent areas of spectral prominence in the instrument's signature. Furthermore, we present an approach that generalizes across examples of different pitches and volume levels. In the experiments, we demonstrate the ability to learn the locations of features in one dataset and use the signatures to classify another dataset, indicating the ability of this approach to generalize an instrument's timbre, independent of the dataset.

CHAPTER 7

FEATURE EXTRACTION

In Chapter 6, we present a data-driven approach to learn areas of spectral prominence for each instrument, known as instrument signature. In the feature extraction stage, presented in this chapter, the instrument signature is applied as a spectral mask or filter. Only the spectral energy underneath the signature will be considered eligible for feature extraction while the rest of the spectral signal is disregarded as noise. These features are used in the classification experiments, discussed in Chapters 8 and 9.

7.1 Applying Spectral Signatures

For each example to process, we first convert the sound file to the spectral domain using an FFT as described in Section 5.4. If the file to be processed is an example single instrument, such as the datasets used in training the models, the fundamental frequency is identified using the process in Section 6.3.2. If the file to be processed is a test case containing an unknown set of instruments, each significant peak must be considered as a possible fundamental frequency for each instrument hypothesis. This polyphonic case is discussed in detail in Section 7.2.

7.1.1 Signature as a Mask

Assuming a hypothesis of a particular instrument, the signature learned for that instrument (see Section 6.3) is applied to the amplitude spectra as a spectral mask.

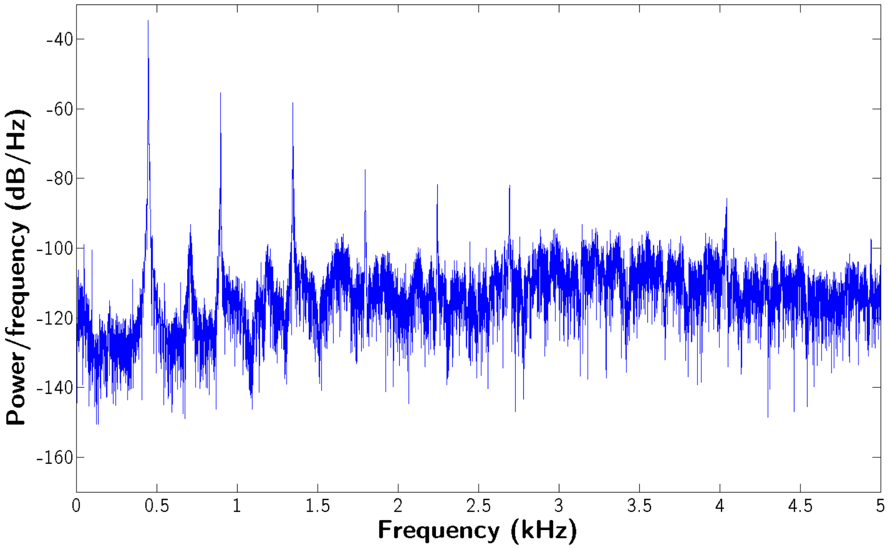
Each cluster c of the signature has a mean c_μ and a standard deviation c_σ. For each Gaussian cluster in the signature, we calculate a window centered on c_μ and ranging plus and minus one standard deviation. The ratio is calculated relative to the fundamental, and each window ranges $((c_\mu - c_\sigma) \cdot f_0)$ to $((c_\mu + c_\sigma) \cdot f_0)$. Figure 7.1b shows a Clarinet signature applied to the spectra of the Clarinet note shown in Figure 7.1a.

For each cluster i in the spectral signature, extract the maximum amplitude within the window of $(\mu_i \pm \sigma_i) \cdot f_0$. Consider this example. Given $f_0 = 446.0$ Hz, $\mu_1 = 2.003$, and $\sigma_1 = 0.026$, we calculate a window $[881.742, 904.934]$ and we extract the maximum amplitude in window: -65.81 at 890.0 Hz.

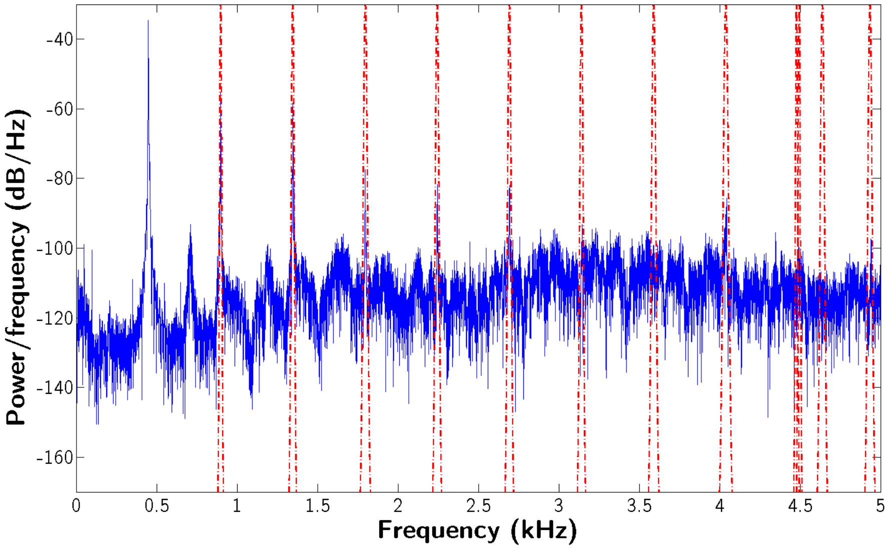
In the next section, we discuss extracting the full set of features from a signal for the entire instrumental signature.

7.1.2 Feature Extraction

Within each cluster window, the maximum amplitude is extracted to be used as a feature. This is repeated for each Gaussian cluster in the signature. In this dissertation we use the very simple feature of the maximum amplitude value within each window. In future work, we will explore using other more complex spectral features, such as those described in [36]. As our primary goal in this work centers on creating a flexible feature extraction scheme extensible to multilabel classification of polyphonic mixtures as well as providing generalizability between instruments of different datasets, we avoid potentially overfitting individual datasets by selecting a complex set of spectral features, and instead demonstrate our approach using a simple feature space.



(a) Spectra of the Clarinet note.



(b) Clarinet spectra with signature visualized (dotted)

Figure 7.1: Spectra of a Clarinet playing A (440 Hz).

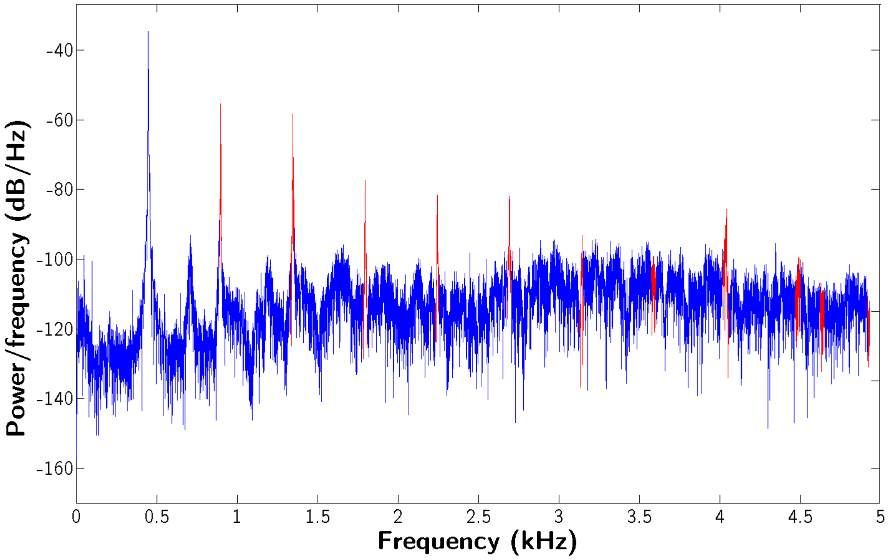
In Figure 7.2, we compare the entire spectra of a Clarinet note (Figure 7.2a) with the filtered view of the spectra of a Clarinet note that falls under the spectral mask (Figure 7.2b).

7.1.3 Amplitude Normalization

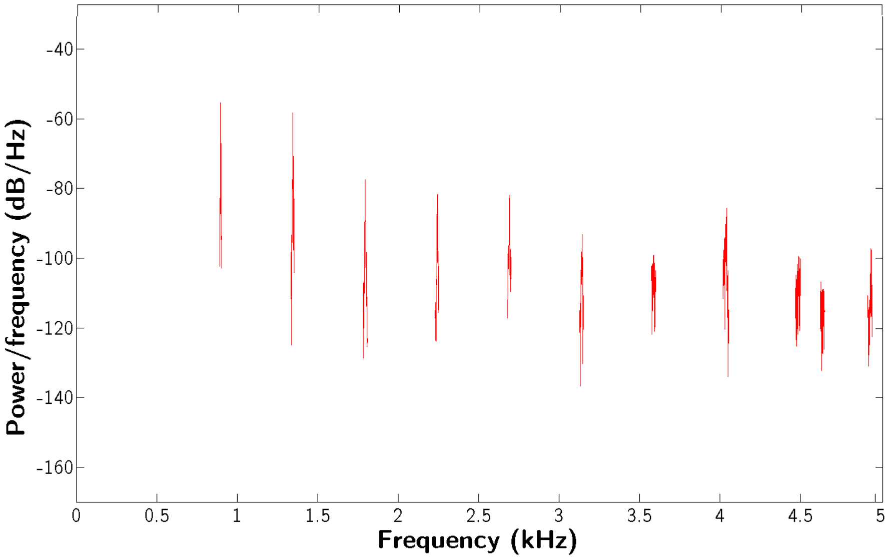
Since one of the goals of this work is to demonstrate generalization of this approach by validating across datasets, we desire our feature set to be comparable across different datasets that may have been recorded with different equipment, procedures, or volume levels. In order to accomplish this, we need a relative measure of amplitude rather than absolute measurements. Therefore, we normalize the amplitude values relative to the amplitude of the fundamental.

Given the amplitude of the fundamental, a_0, and the amplitude a_i of some harmonic i, where $i > 0$, the amplitude ratio is $r_i = a_0/a_i$. Since these are power spectral density measurements, the values are negative and a higher value corresponds to a frequency component with more energy. Therefore a ratio of the fundamental's amplitude to a partial's amplitude yields a value greater than one if the partial is stronger than the fundamental. The ratio is a value less than one if the partial is weaker than the fundamental's amplitude.

In Table 7.1, we show three Oboe notes from the RWC dataset, each played at A (440 Hz) at three different volume levels. This table shows the raw power density values in dB/Hz as well as the ratio values when normalized by the fundamental's amplitude. Observe that the amplitude value of the fundamental differs between the three examples. Even though the absolute power density values of the first overtone (f_1) differ among the three examples, the normalized amplitude ratios remain comparable because of the normalization procedure.



(a) Only the spectral mask of the Clarinet signature will be considered for feature extraction while the rest of the signal will be ignored as noise.



(b) Portion of the spectra from Fig. 7.2a that falls under the spectral signature.

Figure 7.2: Spectra of a Clarinet playing A (440 Hz)

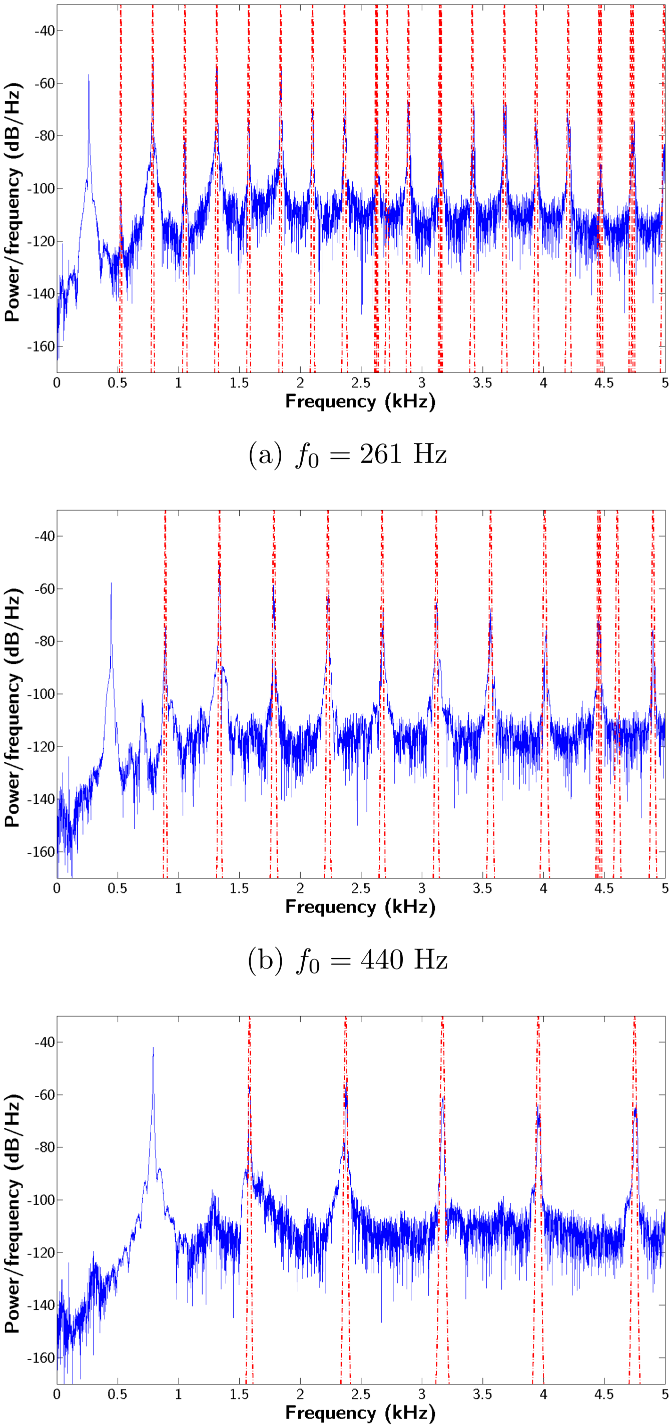
Table 7.1: Shown are three examples of an Oboe playing 440 Hz at three dynamic levels. The columns show the amplitudes of the fundamental frequency (f_0) and the first five overtones ($f_1 - f_5$) are shown. For each example, the top line shows the power spectral density (dB/Hz) of each partial and the bottom line shows the ratio of the amplitudes of the fundamental and the partial.

dynamic	f_0	f_1	f_2	f_3	f_4	f_5
forte	-69.53 –	-51.27 1.36	-51.42 1.35	-58.81 1.18	-61.14 1.14	-62.29 1.12
mezzo-forte	-73.94 –	-54.24 1.36	-52.26 1.40	-57.38 1.29	-62.01 1.19	-61.24 1.21
piano	-74.84 –	-53.72 1.39	-53.61 1.39	-58.56 1.28	-61.66 1.21	-67.35 1.11

Preserving feature values that are relative to the amplitude of the fundamental frequency allows us to compare notes played at different dynamic levels. Furthermore, this approach permits us to compare notes between datasets, helping to overcome differences between recording procedures. This allows us to compare features from examples of the same instrument extracted from different datasets.

Because our signatures learn ratios to f_0 rather than absolute frequency locations, we are able to compare all notes of the same instrument despite differences in pitch. Figure 7.3 shows the same Clarinet signature applied to three different notes and the clusters spacing is dependent on the example's f_0.

This process is repeated for all clusters in the signatures (see Table 6.1) and the resulting amplitude values are converted to the ratios to the fundamental's amplitude and stored as features. The final set of extracted amplitudes serves as the features for the example.



(a) $f_0 = 261$ Hz

(b) $f_0 = 440$ Hz

(c) $f_0 = 738$ Hz

Figure 7.3: Spectra of three Clarinet notes with signature (dotted) applied.

Algorithm 7.2 Feature Extraction for Single Instrument Example

```
Given: Instrument hypothesis $i$, spectral signature $S_i$, signal $X$
$Y = \text{FFT}(X)$
identify $f_0$ in $Y$
for all $c \in S_i$ do
    $W = [(c.\mu - c.\sigma) \cdot f_0, (c.\mu + c.\sigma) \cdot f_0]$
    $f_c \leftarrow$ extract maximum amplitude in $W \cdot Y$
end for
return $\boldsymbol{f}$, vector of amplitudes with one per cluster in $S_i$
```

7.2 Feature Extraction of a Polyphonic Source

When presented with a mixture of unknown instruments, our system must test for each possible instrument present in the mixture. For each instrument i, we must query each binary classifier m_i for a confidence that the instrument is present in the mixture.

Because the instrument signatures trained in Chapter 6 contain a different number of clusters for each instrument, each binary classifier model m_i requires a different feature space. Furthermore, for each instrument, we must extract the binary-relevance feature set and then query the binary classifier.

For each instrument and dataset, we first train a binary model on solo instrument data using the feature extraction scheme described in Section 7.1. The remainder of this section discusses the feature extraction and classification steps for testing unknown examples containing a mixture of more than one instruments.

In the case of solo instrument sound examples, we know there is only one instrument in the recording and could assume that the lowest significant peak corresponds to the fundamental frequency. In examples containing mixtures of instruments, this assumption is not valid as any significant peak could correspond to the fundamental frequency of one of the instruments. Some approaches rely on a multi-pitch finding algorithm for this stage of source separation, but that is an active field of research in itself. We instead consider all peaks as possible candidate fundamental frequencies, a naïve yet comprehensive approach which will in future work will be further optimized. Therefore, for a polyphonic mixture file, we must first extract all significant peaks, following the procedure given in Section 6.3.3.

In Algorithm 7.3, we give the pseudo-code of the feature extraction process for each instrument. For any unknown audio example X, we run the FFT algorithm to compute the spectrum Y. The FFT has the complexity $\mathcal{O}(n \cdot \log n)$, where n is the number of input samples. At the sampling rate of 44.1 kHz, $n = 44100$ for each one second example. The FFT must be computed only once per example to classify.

Next we iterate over all thirteen instruments. For each instrument i, we consider each significant peak p found in Y as a possible fundamental frequency for the instrument i. We use the frequency value of the peak as a hypothesis of an f_0 for instrument i.

In practice, we need not iterate over all the significant peaks but only those within the musical range possible for the current instrument. We restrict this step to only consider the frequencies within the range of the instrument shown in Table 7.2. To produce these ranges, we analyzed the four datasets and found the examples with the highest and lowest frequencies for each instrument. We expand the range to include one additional musical semitone on both the low and high end.

Continuing the process, for instrument i and hypothesized $f_0 = p$,we apply the spectral signature S_i to the spectra Y. This is visualized in Figure 7.4. We need only consider the portion of the spectrum under the signature as possible features and can ignore the rest of the spectra. For each cluster c in the signature S_i, we calculate the window W according to the process described in Section 7.1.1. Within W, we

Algorithm 7.3 Feature Extraction for Polyphonic Mixture

```
Given: signal $X$
$Y$ = FFT($X$)

{iterate over all instruments}
for all $i \in$ Instruments do
    $S_i \leftarrow$ load spectral signature for instrument $i$
    $sum_i,\ count_i \leftarrow 0$

    {iterate over all significant peaks}
    for all $p \in Y$ do
        $f_0 \leftarrow p$

        {iterate over all clusters}
        for all $c \in S_i$ do
            $W = [(c.\mu - c.\sigma) \cdot f_0, (c.\mu + c.\sigma) \cdot f_0]$
            $f_c \leftarrow$ extract maximum amplitude in $W \cdot Y$
        end for

        {classify the feature set $\boldsymbol{f}$}
        $conf \leftarrow$ classify$_i(\boldsymbol{f})$
        if $conf \geq 0.5$ then
            $sum_i \leftarrow$ classify$_i(\boldsymbol{f})$
            $count_i \leftarrow count_i + 1$
        end if
    end for

    {normalize sum by the number of peaks voting for instrument $i$}
    $confidence_i \leftarrow sum_i$ / $count_i$
    add $confidence_i$ to set of confidences
end for
return set of confidences by instrument
```

Algorithm 7.4 Ranking Algorithm for Binary-Relevance Classifiers

```
Given: set of confidences by instrument

sort the set of confidences from highest to lowest

{iterate over all instruments}
for all $i \in$ Instruments do

    assign $rank_i$ to the index of $confidence_i$

end for
return rankings of instrument labels from $1 \ldots 13$
```

extract the maximum amplitude. The amplitude ratio is calculated based on the amplitude of the hypothesized f_0. The set of amplitude ratios, one for each cluster in the signature, is used as the feature set for the hypothesis of instrument i with $f_0 = p$.

7.3 Classification from a Polyphonic Source

We use a adaptation of the k-NN classifier [258] to provide confidences for each instrument class. For each instance given to the binary classifier, we calculate the average mean-square error between the test exemplar and the k nearest training examples, and use this value of the confidence for the selected class. For each peak p, we query the binary classifier c_i for a confidence that the instrument i with $f_0 = p$ matches the trained model's expectation of instrument i. For all peaks for the hypothesis of instrument i that were classified as containing instrument i, we calculate an average confidence and use this average confidence, with a threshold of 0.5 to determine if instrument i should be a relevant label. After repeating this process for all

Table 7.2: Frequency ranges in Hertz of the musical instruments.

Instrument	**Low**	**High**
FrenchHorn	58.3	740.0
Trumpet	138.5	1975.5
Trombone	103.8	739.9
Tuba	38.8	415.3
Flute	246.9	2793.8
Clarinet	138.6	2093.0
AltoSaxophone	130.8	1479.9
Oboe	155.56	1760.0
Bassoon	46.2	698.5
Violin	185.0	3729.3
Viola	123.5	2793.8
Cello	46.25	2959.9
Contrabass	29.1	622.3

instruments, we use the set of confidences of each instrument as a ranking procedure for the multi-label classification. This ranking process is shown as Algorithm 7.4.

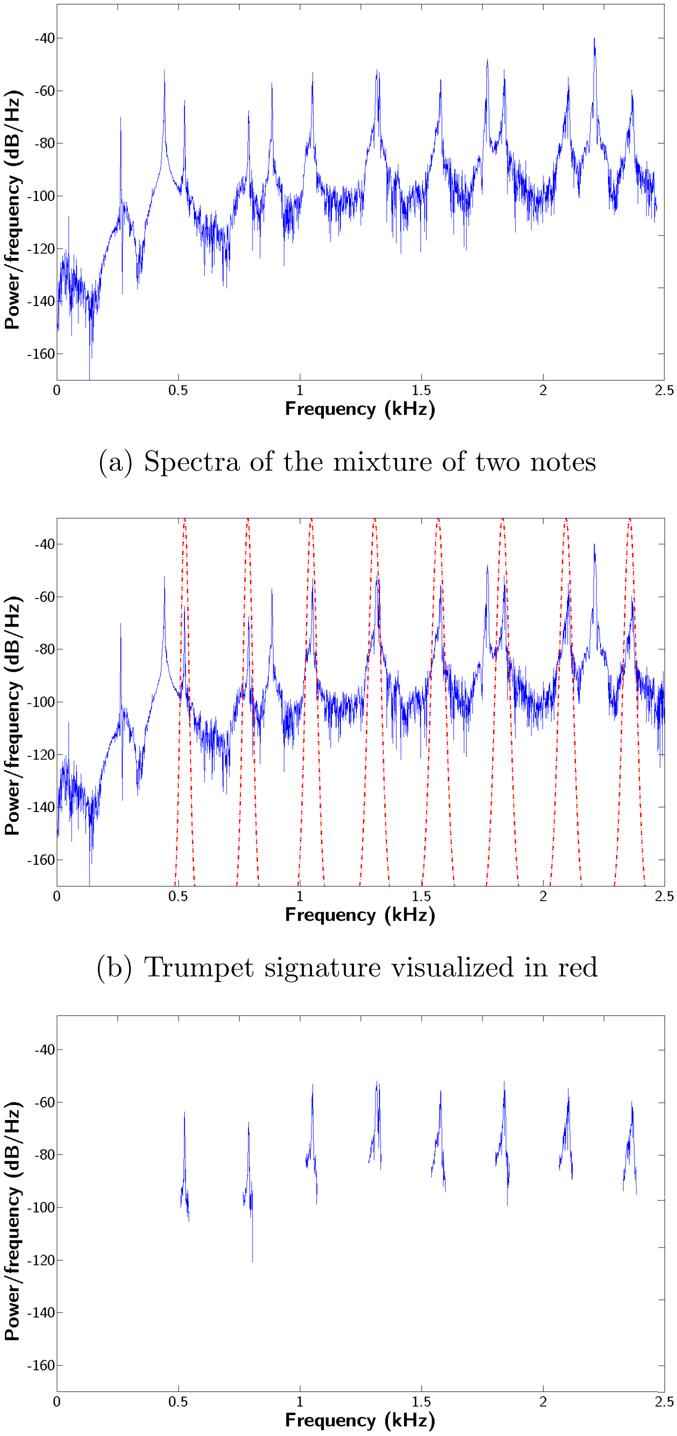
7.4 Complexity of Approach

The complexity of this feature extraction scheme is the sum of $\mathcal{O}(n \cdot \log n)$ for the FFT, which needs be computed only once, and the feature extraction and classification steps. The feature extraction repeats for all $i = 13$ instruments and significant peaks $P \subset Y$ in the range of the instrument. Additionally, for each peak, a windowing calculation is made for each cluster c in the signature S_i. Together this requires $\mathcal{O}(q \cdot |P| \cdot |C| + m)$, where q is the number of class labels (instruments), $|P|$ is a maximum number of peaks and $|C|$ is the maximum number of clusters. There are m models to query, and given the binary-relevance paradigm, $m = q$ classification steps. The number of clusters for each instrument does vary, but is bounded by the counts shown in Table 6.1. The maximum number of significant peaks $|P|$ is also a variable, but bounded value. In Section 6.3.3, we discuss the identification of spectral peaks within a local frequency neighborhood. Therefore the number of peaks each instrument must check is a small number in practice, often no more than a dozen, but variable depending on the frequency distribution of the sound example.

Combining with the FFT step, the complexity of the process is $\mathcal{O}(n \cdot \log n + 2 \cdot q \cdot |P| \cdot |C|)$. In practice, the process is dominated by the number of class labels q thus requiring a feature extraction and classification step for each instrument and peak. All binary-relevance classification approaches require querying a model for each class label q, but our approach requires an additional q feature extraction steps.

7.5 Conclusion

In this chapter, we present our feature extraction approach using a binary-relevance feature extraction approach, using the signatures learned in Chapter 6. First we discuss how to apply the instrument signatures to a monophonic examples, creating a spectral mask. Next we discuss our simple feature extraction scheme and novel amplitude normalization scheme. This amplitude normalization scheme allows us to compare examples of differing dynamic levels and from different datasets. Next we introduce our extension of our feature extraction approach to handle feature extraction and classification from polyphonic mixtures. Our multi-label experimental results for polyphonic mixtures are discussed in the next chapter. Lastly, we conclude with the computational complexity of our binary-relevance feature extraction and classification approach.



(a) Spectra of the mixture of two notes

(b) Trumpet signature visualized in red

(c) Spectra after filtering with Trumpet signature

Figure 7.4: Overlapping spectra of a Trumpet playing middle C (262 Hz) and a Violin playing A (440Hz).

CHAPTER 8

CROSS-DATASET VALIDATION

In this Chapter, we explore the ability of our approach to generalize between datasets. Specifically, we test the ability of the binary-relevance feature-extraction approach in experiments that train on one dataset and test on another dataset in the monophonic instrument classification task.

8.1 Experimental Design

For each dataset, we train a separate binary classifier for each instrument. For these experiments we use a k-NN classifier described in Section 2.3.2. To train each instrument classifier, we need only consider examples of that instrument in isolation, using the datasets described in 5.5. For the feature set, we extract features according to the procedure for solo instruments given in Chapter 7.

For the instrument classifier, we use the instrument signature learned on the training dataset for feature extraction of both training and test examples. The same signature must be used for feature extraction on both training and test examples, because the number of features extracted differs for each instrument and dataset. We use this trained instrument classifier to classify each of the other datasets. When the training set and test set are from the same dataset, we self-classify from the dataset using a 10-fold cross-validation paradigm as described in Section 6.4.2.

Table 8.1: Cross-dataset experiments showing the F-measure for each binary classifier (instrument) for each dataset. The column headers show the test dataset. The boldfaced values indicate self-classification. All others values represent cross-dataset classification

(a) Classifier trained on the MUMS dataset.

Instr.	**MUMS**	**MIS**	**RWC**	**PHO**
FH	**0.66**	0.65	0.59	0.59
TR	**0.79**	0.67	0.65	0.59
TB	**0.62**	0.66	0.65	0.61
TU	**0.69**	0.50	0.66	0.71
FL	**0.78**	0.73	0.72	0.63
CL	**0.81**	0.75	0.78	0.80
AS	**0.59**	0.38	0.47	0.44
OB	**0.68**	0.67	0.68	0.71
BS	**0.77**	0.72	0.70	0.68
VN	**0.73**	0.58	0.67	0.67
VA	**0.68**	0.63	0.66	0.65
VC	**0.78**	0.62	0.77	0.63
CB	**0.83**	0.74	0.84	0.77

(b) Classifier trained on the MIS dataset.

Instr.	**MUMS**	**MIS**	**RWC**	**PHO**
FH	0.65	**0.77**	0.65	0.62
TR	0.61	**0.91**	0.66	0.61
TB	0.65	**0.74**	0.69	0.68
TU	0.44	**0.88**	0.42	0.54
FL	0.66	**0.77**	0.74	0.66
CL	0.63	**0.88**	0.83	0.76
AS	0.66	**0.81**	0.64	0.55
OB	0.70	**0.85**	0.67	0.69
BS	0.81	**0.77**	0.73	0.68
VN	0.66	**0.87**	0.75	0.74
VA	0.69	**0.78**	0.58	0.61
VC	0.67	**0.80**	0.67	0.66
CB	0.91	**0.90**	0.88	0.81

Table 8.1: Cross-dataset experiments showing the F-measure for each binary classifier (instrument) for each dataset. The column headers show the test dataset. The boldfaced values indicate self-classification. All others values represent cross-dataset classification

(c) Classifier trained on the RWC dataset.

Instr.	**MUMS**	**MIS**	**RWC**	**PHO**
FH	0.78	0.75	**0.78**	0.67
TR	0.75	0.74	**0.72**	0.64
TB	0.78	0.74	**0.76**	0.72
TU	0.59	0.36	**0.87**	0.73
FL	0.77	0.73	**0.78**	0.67
CL	0.78	0.82	**0.89**	0.75
AS	0.78	0.75	**0.79**	0.53
OB	0.80	0.79	**0.82**	0.79
BS	0.83	0.81	**0.86**	0.77
VN	0.72	0.69	**0.87**	0.77
VA	0.83	0.60	**0.80**	0.61
VC	0.88	0.67	**0.84**	0.70
CB	0.93	0.85	**0.92**	0.81

(d) Classifier trained on the PHO dataset.

Instr.	**MUMS**	**MIS**	**RWC**	**PHO**
FH	0.70	0.65	0.66	**0.82**
TR	0.62	0.83	0.68	**0.79**
TB	0.78	0.73	0.66	**0.79**
TU	0.86	0.67	0.83	**0.91**
FL	0.49	0.74	0.68	**0.85**
CL	0.85	0.80	0.79	**0.88**
AS	0.57	0.55	0.62	**0.79**
OB	0.76	0.76	0.81	**0.88**
BS	0.70	0.73	0.77	**0.87**
VN	0.78	0.66	0.75	**0.84**
VA	0.68	0.63	0.65	**0.84**
VC	0.74	0.63	0.76	**0.83**
CB	0.90	0.87	0.87	**0.91**

8.2 Results

In Table 8.1 we report the F-measure results of each classifier. The dataset used for training is listed in the caption of each subtable. The row headers indicate the test set. The cases in which the training and test set are the same, and cross-validation is used, are shown in boldface. Table 8.2 shows the relative performance of the cross-dataset experiments to the self-classified dataset, noted with a dash $(-)$.

8.3 Discussion

In these experiments, we found that we are able to train on features from one dataset and test on features extracted from another dataset. As expected, we observe a reduced classification accuracy for the cross-dataset experiments compared to the self-classification experiments, represented as negative values in Table 8.2. However these results are far more promising than the cross-dataset results reported in [34]; although, given the differing data, features and classification algorithms, the results of the two approaches are not directly comparable.

Nevertheless, we are able to classify using the cross-dataset paradigm at rates well above chance for almost all datasets and instruments. In our preliminary experiments, we observe that setting a small value of k, such as $k = 1$ increased accuracy on the self-classification experiments dramatically but decreased accuracy on the cross-dataset experiments. This is an example of overfitting to a specific dataset, which is a common problem in the instrument classification literature. As we increased the value of k, the self-classification results decreased as the cross-dataset accuracy increased. In other words, comparing an unknown example to the single nearest instance is useful in the

Table 8.2: Shows the cross-dataset results shown in Table 8.1 as the relative difference to the results of the self-classified training dataset. A negative value indicates a lower F-measure score relative to the self-classification result, noted with a dash (−). A positive value indicates the cross-data set result outperformed the self-classification score.

(a) Classifier trained on the MUMS dataset.

Instr.	**MUMS**	**MIS**	**RWC**	**PHO**
FH	–	-0.01	-0.07	-0.07
TR	–	-0.12	-0.14	-0.20
TB	–	+0.04	+0.03	-0.01
TU	–	-0.19	-0.03	+0.02
FL	–	-0.05	-0.06	-0.15
CL	–	-0.06	-0.03	-0.01
AS	–	-0.21	-0.12	-0.15
OB	–	-0.01	0.00	+0.03
BS	–	-0.05	-0.07	-0.09
VN	–	-0.15	-0.06	-0.06
VA	–	-0.05	-0.02	-0.03
VC	–	-0.16	-0.01	-0.15
CB	–	-0.09	+0.01	-0.06

(b) Classifier trained on the MIS dataset.

Instr.	**MUMS**	**MIS**	**RWC**	**PHO**
FH	-0.12	–	-0.12	-0.15
TR	-0.30	–	-0.25	-0.30
TB	-0.09	–	-0.05	-0.06
TU	-0.44	–	-0.46	-0.34
FL	-0.11	–	-0.03	-0.11
CL	-0.25	–	-0.05	-0.12
AS	-0.15	–	-0.17	-0.26
OB	-0.15	–	-0.18	-0.16
BS	+0.04	–	-0.04	-0.09
VN	-0.21	–	-0.12	-0.13
VA	-0.09	–	-0.20	-0.17
VC	-0.13	–	-0.13	-0.14
CB	+0.01	–	-0.02	-0.09

(c) Classifier trained on the RWC dataset.

Instr.	**MUMS**	**MIS**	**RWC**	**PHO**
FH	0.00	-0.03	–	-0.11
TR	+0.03	+0.02	–	-0.08
TB	+0.02	-0.02	–	-0.04
TU	-0.28	-0.51	–	-0.14
FL	-0.01	-0.05	–	-0.11
CL	-0.11	-0.07	–	-0.14
AS	-0.01	-0.04	–	-0.26
OB	-0.02	-0.03	–	-0.03
BS	-0.03	-0.05	–	-0.09
VN	-0.15	-0.18	–	-0.10
VA	+0.03	-0.20	–	-0.19
VC	+0.04	-0.17	–	-0.14
CB	+0.01	-0.07	–	-0.11

(d) Classifier trained on the PHO dataset.

Instr.	**MUMS**	**MIS**	**RWC**	**PHO**
FH	-0.12	-0.17	-0.16	–
TR	-0.17	0.04	-0.11	–
TB	-0.01	-0.06	-0.13	–
TU	-0.05	-0.24	-0.08	
FL	-0.36	-0.11	-0.17	–
CL	-0.03	-0.08	-0.09	–
AS	-0.22	-0.24	-0.17	–
OB	-0.12	-0.12	-0.07	–
BS	-0.17	-0.14	-0.10	–
VN	-0.06	-0.18	-0.09	–
VA	-0.16	-0.21	-0.19	–
VC	-0.09	-0.20	-0.07	–
CB	-0.01	-0.04	-0.04	–

self-classification task, but more neighbors are required to better generalize between instruments across datasets.

In the musical instrument classification literature, most approaches are heavily biased by the training set and cannot be use to classify other datasets [34]. Cross-dataset validation needs to be a goal of any approach that hopes eventually to generalize to real-world musical data. Our cross-dataset experiments demonstrate an ability of our approach to provide such generalization.

8.4 Conclusion

In this chapter, we present experimental results evaluating the ability of our system to generalize between datasets and present the largest cross-dataset study in the domain of monophonic instrument classification. We show our ability to train on dataset but test on another dataset, indicating our binary-relevant feature extraction scheme is capturing information pertinent to the instrument's timbre, and not the specifics of the recording procedures of the dataset.

CHAPTER 9

POLYPHONIC INSTRUMENT CLASSIFICATION

In this chapter, we present experimental results for musical instrument classification for polyphonic mixtures of two to four instruments. In the classification stage, we use the instrumental signatures discussed in Chapter 6 and the feature extraction techniques described in Chapter 7.

9.1 Problem Difficulty and Significance

This difficulty of a multi-label problem can be discussed in terms of the label density and label cardinality measures discussed in Section 2.2.3.1. Recall that label cardinality measures the average number of true labels per example and label density measures the average numbers of true labels per example normalized by the total number of possible labels. In our experiments, the label cardinality corresponds to the number of instruments in the mixtures, either two, three, or four in our experiments. In our principal experiments, the number of labels considered is $q = 13$. For comparison, we also provide an experiment in Section 9.5 with a reduced label set, where $q = 5$, to demonstrate our approach on an easier multi-label problem. The label cardinality and density for the experiments in this chapter are listed in Table 9.1. Also listed are the probabilities of guessing the entire set of labels correctly at random, which is calculated as the inverse of the number of combinations with $n = q$ choose r equal to the number of labels in the example.

Table 9.1: Label density and cardinality of the multi-label experiments

q	Polyphony	Cardinality	Density	Chance
13	Two	2	0.154	0.01282
	Three	3	0.231	0.00349
	Four	4	0.307	0.00014
5	Two	2	0.400	0.1
	Three	3	0.600	0.1
	Four	4	0.800	0.2

9.2 Experimental Design

For each dataset, we train a separate binary classifier for each instrument. For these experiments, we use the k-NN classifier described in Section 2.3.2 with a value of $k = 5$, as discussed in Section 8.2. The classifiers are trained with examples of solo instruments using the datasets described in Section 5.5. However, to test the models, we use the polyphonic datasets described in Section 5.6. Similar to the approach given in Section 8.1, for each instrument classifier, we use the signature learned from the training set for the feature extraction process (see Algorithm 7.3). Our classification procedure for polyphonic mixtures is described in Section 7.3. In these experiments the training a set of solo instruments is different than the test set of polyphonic mixtures. However, the polyphonic mixtures were created from the solo examples of the same dataset, giving a dependency between the training and test sets. In Section 9.3.5, we discuss this potential dependency between datasets and present a set of cross-validation experiments that explore this question empirically.

Table 9.2: Results for mixtures of two instruments

Type	**Metric**	**MUMS**	**MIS**	**RWC**	**PHO**
Example-based	Subset Accuracy	0.022	0.028	0.030	0.028
	Hamming Loss	0.250	0.246	0.237	0.241
	Accuracy	0.132	0.143	0.164	0.155
	Precision	0.188	0.201	0.231	0.218
	Recall	0.188	0.201	0.231	0.218
	F_1 Measure	0.188	0.201	0.231	0.218
Label-based	Macro-Precision	0.189	0.204	0.229	0.208
	Macro-Recall	0.187	0.201	0.231	0.220
	Macro-F_1	0.151	0.174	0.213	0.185
	Micro-Precision	0.188	0.201	0.231	0.218
	Micro-Recall	0.188	0.201	0.231	0.218
	Micro-F_1	0.188	0.201	0.231	0.218
Rank-based	One-Error	0.816	0.797	0.764	0.789
	Coverage$_1$	2.993	2.948	2.563	2.693
	Coverage$_2$	6.921	7.065	6.622	6.711
	Ranking Loss	0.712	0.701	0.667	0.677
	Average Precision	0.363	0.372	0.400	0.388

9.3 Multi-label Classification of Polyphonic Mixtures

9.3.1 Multi-label Self-Classify Experiments

In these experiments, we test our system's ability to classify polyphonic mixtures of two, three, or four instruments. In this section, each monophonic training set and polyphonic test set originate from the same data source. These experiments use all 13,000 mixtures described in Section 5.6. The results of these experiments are shown in Tables 9.2 − 9.4 and these results show measures for each of the four datasets.

Example- and label-based measures assume the number of target labels is known and only that many labels are considered by the measure. In our experiments, all examples have the same number of labels and cardinality of the experiment, corre-

Table 9.3: Results for mixtures of three instruments

Type	**Metric**	**MUMS**	**MIS**	**RWC**	**PHO**
Example-based	Subset Accuracy	0.006	0.007	0.008	0.007
	Hamming Loss	0.342	0.340	0.327	0.331
	Accuracy	0.172	0.175	0.195	0.189
	Precision	0.259	0.263	0.292	0.283
	Recall	0.259	0.263	0.292	0.283
	F_1 Measure	0.259	0.263	0.292	0.283
Label-based	Macro-Precision	0.276	0.271	0.299	0.286
	Macro-Recall	0.259	0.263	0.291	0.284
	Macro-F_1	0.207	0.224	0.267	0.246
	Micro-Precision	0.259	0.263	0.292	0.283
	Micro-Recall	0.259	0.263	0.292	0.283
	Micro-F_1	0.259	0.263	0.292	0.283
Rank-based	One-Error	0.763	0.745	0.703	0.727
	Coverage$_1$	2.226	2.173	1.912	2.000
	Coverage$_2$	5.356	5.334	5.072	5.120
	Coverage$_3$	8.606	8.684	8.414	8.437
	Ranking Loss	1.018	1.01	0.953	0.984
	Average Precision	0.404	0.409	0.435	0.424

Table 9.4: Results for mixtures of four instruments

Type	**Metric**	**MUMS**	**MIS**	**RWC**	**PHO**
Example-based	Subset Accuracy	0.002	0.002	0.003	0.003
	Hamming Loss	0.413	0.406	0.398	0.400
	Accuracy	0.216	0.224	0.234	0.231
	Precision	0.329	0.340	0.353	0.350
	Recall	0.329	0.340	0.353	0.350
	F_1 Measure	0.329	0.340	0.353	0.350
Label-based	Macro-Precision	0.354	0.353	0.360	0.357
	Macro-Recall	0.328	0.339	0.351	0.348
	Macro-F_1	0.259	0.296	0.321	0.309
	Micro-Precision	0.329	0.340	0.353	0.350
	Micro-Recall	0.329	0.340	0.353	0.350
	Micro-F_1	0.329	0.340	0.353	0.350
Rank-based	One-Error	0.704	0.664	0.641	0.654
	Coverage$_1$	1.709	1.597	1.461	1.500
	Coverage$_2$	4.318	4.197	4.049	4.065
	Coverage$_3$	6.938	6.855	6.750	6.723
	Coverage$_4$	9.638	9.641	9.562	9.503
	Ranking Loss	1.268	1.247	1.173	1.219
	Average Precision	0.453	0.464	0.481	0.474

sponding to either two, three, or four labels, depending on the specific experiment. Although our system provides a ranking of labels, for these metrics, only the top two, three, or four labels are selected for polyphony of two, three, or four, respectively.

We begin by discussing the results of the example-based measures. Subset or exact accuracy is a very strict measure which does not reward partial accuracy. For this reason, it is rarely reported in the multi-label classification literature and has never previously been reported in the polyphonic instrument classification domain before this work. For mixtures of two instruments, our system achieves exact accuracy between 2% and 3%. This value falls to 0.2% for polyphony of four instruments. Although our exact match accuracy is quite low, as expected for a problem with a low label density, it does exceed the value of guessing the set at random (see Table 9.1). For polyphony of four, our exact match score is an order of magnitude higher than guessing at chance.

Hamming loss measures a fraction of misclassified instance/label assignments. As expected, the Hamming loss degrades as the cardinality of the problem increases. A Hamming loss of 1.0 implies a complete failure to find any correct labels. For polyphony of four, our system achieves around 0.4 for each of the four datasets. Our system finds similar Hamming loss for all the datasets, which is an encouraging result supporting our claim of the generalizability of our system. Accuracy measures partial accuracy between the true and predicted label sets. This partial accuracy is lowest for polyphony of two and increases as the label density increases to an accuracy over 20% for polyphony of four. This measure is low because our system often confuses instruments within the same family, a problem consistently encountered in the literature [163], and our relevant set may contain a similar but incorrect instrument label, such as mistaking the Violin for a Viola. In these cases many of our predicted labels that are also correct are ranked just below the cutoff of the cardinality.

The multiple-label information retrieval measures of precision, recall, and F_1 measure come in three variations, an example-based variation as well as label-based macro- and micro- variations (see Section 2.2.3). In our experiments, given the large number of examples and large value of q, we found the micro-based precision, recall, and F_1 values to converge to the values of the example-based precision, recall, and F_1 measures. This is not always the case for problems with a small label density and fewer examples in the dataset (for example, consider the results of Section 9.5). Like with other measures that permit partial accuracy, we found the F_1 to increase as the label density increased, ranging from around 0.2 for two-voice mixtures up to at least 3.2 for four-voice mixtures. We are encouraged by our highest F_1 scores, achieved by the RWC dataset for mixtures of four instruments, and recognize the need for improvement in the two and three instrument experiments.

Next, we discuss the ranking-based measures, which, unlike the example- and label-based measures, consider a ranked list of all q labels. These metrics are useful to consider how many false positives must be tolerated in order to achieve all true positives. As mentioned earlier, our system produces many confusions between similar instruments, resulting in additional false positives. The first ranking-based measure is One-Error, which considers how often the topmost ranked label is among the true label set. A value of 0.0 indicates perfect performance while a value of 1.0 indicated a complete failure. Our system ranked a true label first around 20%, 25%, and 30% for polyphony of two, three, and four respectively.

The Coverage$_j$ measure examines how far down the list, on average, we must go to cover j possible labels. For example, for polyphony of two, our system finds the first label, Coverage$_1$, within the first three labels on average. To cover both of the labels, it takes about seven labels to cover. This indicates that on average, our system finds both true labels within the first half of ranked label list. For polyphony of three, it takes about 2, 5, and 8 labels to cover the first, second, and third true labels, respectively. For polyphony of four, it takes around 2, 4, 7, and 9 labels to cover the first through fourth labels, respectively.

The last two rank-based measures explore the accuracy of the ranking order. For example, consider polyphony of two with an instance where true labels set $Y_i = \{\text{Trumpet, Viola}\}$ and a set of rankings $\{\text{Trumpet, Violin, Viola, } \ldots\}$. The above measures, such as F_1 would not consider the Violin in the calculation, but a rank-based measure would. Ranking loss, measures the fraction of times an irrelevant label outranks a relevant label. This measure helps capture relevant labels that might be ranked slightly below the cardinality cut-off used in label- and example-based measures. Like other of the measures, save subset accuracy, the values improve as polyphony increases and the label density of the problem decreases. For polyphony of four, we achieve a ranking loss of around 1.2, in which 4.0 indicated a completely incorrect label ordering. Average precision measures the ranking ordering of the relevant labels, calculating the difference in rank between the various relevant labels and averaging over all examples. For example, consider an instance in which relevant labels are ranked first and fourth. This measure will penalize that instance more so than an instance in which the relevant labels are ranked first and third. When comparing our average precision to example-based precision, we observe a substantial improvement, indicating that many of our relevant labels are ranked high on the list, but outside the cut-off of the label cardinality, as found in the Viola and Violin confusion example given above.

Table 9.5: p-values of results compared to random chance for mixtures of two instruments

Type	**Metric**	**MUMS**	**MIS**	**RWC**	**PHO**
Example-based	Subset Accuracy	0.009	0.007	0.007	0.006
	Hamming Loss	0.156	0.144	0.136	0.142
	Accuracy	0.156	0.144	0.136	0.142
	Precision	0.156	0.144	0.136	0.142
	Recall	0.156	0.144	0.136	0.142
	F_1 Measure	0.156	0.144	0.136	0.142
Rank-based	One-Error	0.088	0.077	0.075	0.067
	Coverage$_1$	0.222	0.215	0.203	0.214
	Coverage$_2$	0.148	0.181	0.131	0.145
	Ranking Loss	0.203	0.193	0.157	0.169
	Average Precision	0.138	0.122	0.098	0.112

9.3.2 Comparison to Random Permutations

To evaluate our results to a baseline of chance, we designed a statistical experiment to compare our results to randomly chosen permutations of instruments. We create random permutations from the set of thirteen instruments. We evaluate each of the multi-label measures on these sets. We calculate the p-value as

$$\frac{p = (1 + \text{perm}_{+} >= \text{measure\_score})}{(1 + \text{perm}_{n})} \tag{9.1}$$

where the numerator represents the number of random permutations that exceed our score and the denominator represents the number of random permutations evaluated. We evaluated $\text{perm}_n = 1000$ random permutations. The plus one term in both the numerator and denominator represents a smoothing term.

In Tables 9.5, 9.6, and 9.7, we show the p-values of our results from in Section 9.3.1 as compared to results of randomly chosen permutations. These values represent the fraction of times a random permutation outscored our reported result. Because

Table 9.6: p-values of results compared to random chance for mixtures of three instruments

Type	**Metric**	**MUMS**	**MIS**	**RWC**	**PHO**
Example-based	Subset Accuracy	0.002	0.001	0.002	0.001
	Hamming Loss	0.293	0.283	0.274	0.297
	Accuracy	0.293	0.283	0.274	0.297
	Precision	0.293	0.283	0.274	0.297
	Recall	0.293	0.283	0.274	0.297
	F_1 Measure	0.293	0.283	0.274	0.297
Rank-based	One-Error	0.125	0.105	0.108	0.117
	Coverage$_1$	0.293	0.283	0.202	0.297
	Coverage$_2$	0.212	0.221	0.217	0.206
	Coverage$_3$	0.137	0.149	0.140	0.143
	Ranking Loss	0.196	0.207	0.171	0.169
	Average Precision	0.171	0.160	0.141	0.155

Table 9.7: p-values of results compared to random chance for mixtures of four instruments

Type	**Metric**	**MUMS**	**MIS**	**RWC**	**PHO**
Example-based	Subset Accuracy	0.001	0.001	0.001	0.002
	Hamming Loss	0.174	0.180	0.177	0.177
	Accuracy	0.174	0.180	0.177	0.177
	Precision	0.174	0.180	0.177	0.177
	Recall	0.174	0.180	0.177	0.177
	F_1 Measure	0.174	0.180	0.177	0.177
Rank-based	One-Error	0.149	0.155	0.159	0.161
	Coverage$_1$	0.260	0.268	0.258	0.274
	Coverage$_2$	0.248	0.253	0.255	0.260
	Coverage$_3$	0.165	0.162	0.172	0.174
	Coverage$_4$	0.137	0.138	0.146	0.164
	Ranking Loss	0.197	0.208	0.176	0.190
	Average Precision	0.190	0.188	0.177	0.193

the label-based measures are normalized by total number of examples and labels, they cannot be compared to random permutations using Equation 9.1 and have been omitted.

Considering the results of the random permutations as a baseline, our approach outperformed chance in all experiments. For all mixtures, our system was more effective for the the larger RWC dataset compared to the other three datasets. For the difficult measure of Subset Accuracy, our approach significantly outperformed the random permutation baseline. Additionally our system also significantly outperformed chance for the One-Error measure, the accuracy of only the top-ranked label. For the Coverage measures, our system was more effective at finding the complete set of instruments for mixtures of two, three, or four instruments compared to finding only the first instrument for the Coverage$_1$ measure. This likely reflects our systems tendency to confuse similar instruments. The other measures represent weak confidence over random sets, that we will continue to improve in our future work. This reflects the difficulty of the multi-label problem and we compare these statistical results with our experiments with an easier multi-label problem given in Section 9.5.

9.3.3 Polyphonic Results by Instrument

The label-based evaluation Macro-F_1 measure aggregates examples by label before it is normalized by the number of instruments (see Section 2.2.3.3). For these polyphonic results, we are able to report the Macro-F_1 measure for each individual instrument. These results are provided in Table 9.8. On the instrument level, we do observe differences between the datasets. The MUMS dataset, our smallest dataset, failed to identify the Clarinet and the Violin. The MIS dataset failed to recognize the Oboe. Our largest dataset, RWC, also failed to recognize the Oboe. The PHO dataset, comprised of lower MP3 quality examples failed to identify the Trumpet. These failures were consistent over the two, three, and four instrument mixtures.

The differences between datasets motivate further study in combining datasets into single training datasets and exploring ensemble models and we will explore this as future work. Such an approach requires careful consideration of the empirical questions of balancing dataset sampling, difficult when the dataset is small, and the issue of cross-validation of datasets.

9.3.4 Polyphonic Results with Instrument Family Labels

In our results, we observed frequent confusion among similar instruments, such as mistaking a Violin for a Viola note. Such confusion is to be expected and is commonly reported in the literature. To further explore this observation, we analyzed our results with regard to musical instrument family, either woodwind, brass, or string. Any label that matched the instrument family was counted as correct, even if the instrument predicted was mistaken for another instrument within its family. While interesting, this analysis is not scientific, as it changes the multi-label problem, allowing the set of rankings that contain duplicate family labels.

The results of this analysis are given in Table 9.9. The label-based measures cannot be reported because we disturbed the label space in our substitution of family name for instrument name. Although there are only three family labels, our ranked list of labels contains duplicates and therefore may predict all the same family labels for an example. For example, the system could predict four String instruments for a mixture of four instruments. Therefore this is not a three label problem, but an analysis of our 13 instrument problem.

Table 9.8: Polyphonic classification results by instrument for mixtures of two, three, and four instruments. For each instrument and dataset, the F_1-macro is shown.

Instrument	**MUMS**	**MIS**	**RWC**	**PHO**
Flute	0.259	0.279	0.270	0.283
Oboe	0.241	0.014	0.045	0.145
Clarinet	0.016	0.267	0.302	0.154
Bassoon	0.035	0.147	0.094	0.091
AltoSax	0.174	0.083	0.199	0.108
FrenchHorn	0.042	0.220	0.222	0.194
Trumpet	0.062	0.234	0.175	0.015
Trombone	0.085	0.096	0.162	0.240
Tuba	0.336	0.275	0.228	0.366
Violin	0.013	0.159	0.204	0.150
Viola	0.201	0.129	0.148	0.197
Cello	0.215	0.085	0.282	0.130
Bass	0.286	0.277	0.437	0.327

(a) Results for mixtures of two instruments

Instrument	**MUMS**	**MIS**	**RWC**	**PHO**
Flute	0.349	0.370	0.351	0.365
Oboe	0.350	0.021	0.069	0.198
Clarinet	0.015	0.290	0.365	0.169
Bassoon	0.076	0.187	0.160	0.136
AltoSax	0.190	0.160	0.282	0.164
FrenchHorn	0.048	0.277	0.195	0.284
Trumpet	0.115	0.352	0.257	0.024
Trombone	0.214	0.114	0.257	0.307
Tuba	0.349	0.340	0.189	0.437
Violin	0.018	0.193	0.278	0.250
Viola	0.283	0.173	0.240	0.254
Cello	0.334	0.103	0.365	0.205
Bass	0.355	0.327	0.460	0.403

(b) Results for mixtures of three instruments

Instrument	**MUMS**	**MIS**	**RWC**	**PHO**
Flute	0.445	0.470	0.426	0.442
Oboe	0.454	0.039	0.092	0.257
Clarinet	0.012	0.347	0.426	0.206
Bassoon	0.117	0.249	0.198	0.146
AltoSax	0.216	0.298	0.383	0.234
FrenchHorn	0.066	0.321	0.202	0.358
Trumpet	0.137	0.461	0.337	0.032
Trombone	0.341	0.182	0.328	0.391
Tuba	0.266	0.411	0.159	0.488
Violin	0.036	0.290	0.369	0.342
Viola	0.373	0.237	0.321	0.347
Cello	0.451	0.136	0.446	0.284
Bass	0.448	0.410	0.490	0.490

(c) Results for mixtures of four instruments

First, consider the subset accuracy measure, the strictest measure. We correctly identified the families of all the true labels 37%, 45%, and 58% of the time for two, three, and four instruments, respectively, a marked improvement over the $< 1\%$ subset accuracy of the 13 instrument problem. Additionally, we find improvement in our F_1 measure, improving to 0.52, 0.55, and 0.53 for two, three, and four instruments, respectively. Lastly, we point to the One-Error measure, which measures how often the first ranked relevant label is indeed a true label. We predicted the family of a correct label as our top-ranked label 73%, 78%, and 87% of the time for two, three, and four instruments, respectively.

This analysis confirms our hypothesis that many of our errors are the result of confusions among similar instruments. As future work we intend to revisit this issue and carry out an empirical study examining the specific instrument confusions. Following those results, we intend to adapt our system into an ensemble of classifiers in which we identify instrument family of an instrument as a first step followed by classification by models trained to differentiate between instruments in the same family.

Table 9.9: Results for classification of polyphonic mixtures of with family labels

Type	**Metric**	**MUMS**	**MIS**	**RWC**	**PHO**
Example-based	Subset Accuracy	0.370	0.387	0.4	0.384
	Hamming Loss	0.099	0.105	0.096	0.103
	Accuracy	0.505	0.487	0.517	0.494
	Precision	0.520	0.523	0.533	0.533
	Recall	0.520	0.523	0.533	0.533
	F_1 Measure	0.520	0.523	0.533	0.533
Rank-based	One-Error	0.368	0.403	0.348	0.394
	Coverage_1	0.629	0.669	0.566	0.614
	Coverage_2	2.151	2.342	2.092	2.268

(a) Results for mixtures of two instruments with family labels

Type	**Metric**	**MUMS**	**MIS**	**RWC**	**PHO**
Example-based	Subset Accuracy	0.457	0.560	0.475	0.539
	Hamming Loss	0.081	0.078	0.082	0.080
	Accuracy	0.635	0.654	0.629	0.645
	Precision	0.548	0.590	0.548	0.577
	Recall	0.548	0.590	0.548	0.577
	F_1 Measure	0.548	0.590	0.548	0.577
Rank-based	One-Error	0.228	0.241	0.233	0.248
	Coverage_1	0.322	0.322	0.345	0.351
	Coverage_2	1.572	1.648	1.643	1.692
	Coverage_3	2.931	3.012	2.966	3.051

(b) Results for mixtures of three instruments with family labels

Type	**Metric**	**MUMS**	**MIS**	**RWC**	**PHO**
Example-based	Subset Accuracy	0.584	0.742	0.608	0.706
	Hamming Loss	0.059	0.049	0.058	0.052
	Accuracy	0.742	0.786	0.749	0.773
	Precision	0.530	0.573	0.537	0.563
	Recall	0.530	0.573	0.537	0.563
	F_1 Measure	0.530	0.573	0.537	0.563
Rank-based	One-Error	0.137	0.139	0.136	0.155
	Coverage$_1$	0.174	0.171	0.195	0.203
	Coverage$_2$	1.322	1.338	1.361	1.384
	Coverage$_3$	2.504	2.534	2.527	2.581
	Coverage$_4$	3.705	3.744	3.697	3.776

(c) Results for mixtures of four instruments with family labels

9.3.5 Self-Classification Experiments with Cross-Validation

In the domain of multi-label classification to polyphonic mixtures, there are two approaches to handling test sets. One approach uses excerpts of musical passages. These approaches most always normalize their evaluation by the number of time frames classified correctly, artificially inflating results if the piece includes moments of silence or passages of a solo instruments. The other approach, which is increasing in popularity in the recent literature [193], is to train on solo instruments and test of mixtures of multiple instruments playing simultaneously. We considered this approach in the experiments given above.

As we mentioned earlier in this section, although the training datasets of solo instruments and the test datasets of polyphonic mixtures are not the same dataset, the solo examples are used to create the mixtures and thus a dependency between the datasets exists. Although this is becoming a common practice, no study has previously explored this bias with cross-validation experiments.

To explore this issue, we designed a 5 × 2 cross-validation experiment to explore the effect of this dependency between training and test sets. Because the training and

Table 9.10: Results of the cross-validation results for mixtures of two, three , and four instruments.

Type	**Metric**	**MUMS**	**MIS**	**RWC**	**PHO**
Example-based	Subset Accuracy	0.022	0.027	0.025	0.032
	Hamming Loss	0.248	0.244	0.242	0.236
	Accuracy	0.137	0.147	0.163	0.165
	Precision	0.194	0.207	0.199	0.232
	Recall	0.194	0.207	0.212	0.232
	F_1 Measure	0.194	0.207	0.212	0.232
Label-based	Macro-Precision	0.219	0.239	0.241	0.243
	Macro-Recall	0.193	0.205	0.212	0.231
	Macro-F_1	0.142	0.177	0.183	0.217
	Micro-Precision	0.194	0.207	0.212	0.232
	Micro-Recall	0.194	0.207	0.212	0.232
	Micro-F_1	0.194	0.207	0.212	0.232
Rank-based	One-Error	0.808	0.793	0.791	0.764
	Coverage$_1$	2.918	2.786	2.685	2.511
	Coverage$_2$	7.005	6.917	6.904	6.607
	Ranking Loss	0.729	0.703	0.693	0.675
	Average Precision	0.359	0.374	0.379	0.396

(a) Cross-validation results on mixtures of two instruments

Type	**Metric**	**MUMS**	**MIS**	**RWC**	**PHO**
Example-based	Subset Accuracy	0.005	0.007	0.007	0.009
	Hamming Loss	0.337	0.336	0.331	0.324
	Accuracy	0.178	0.181	0.188	0.200
	Precision	0.269	0.272	0.283	0.298
	Recall	0.269	0.272	0.283	0.298
	F_1 Measure	0.267	0.272	0.283	0.298
Label-based	Macro-Precision	0.279	0.297	0.301	0.309
	Macro-Recall	0.270	0.272	0.282	0.296
	Macro-F_1	0.209	0.230	0.237	0.273
	Micro-Precision	0.269	0.272	0.283	0.298
	Micro-Recall	0.269	0.272	0.283	0.298
	Micro-F_1	0.270	0.272	0.283	0.298
Rank-based	One-Error	0.727	0.731	0.722	0.702
	Coverage$_1$	2.067	2.082	1.982	1.866
	Coverage$_2$	5.269	5.201	5.185	4.958
	Coverage$_3$	8.647	8.548	8.664	8.390
	Ranking Loss	1.024	1.008	0.985	0.966
	Average Precision	0.412	0.341	0.422	0.435

(b) Cross-validation results on mixtures of three instruments

Type	Metric	MUMS	MIS	RWC	PHO
Example-based	Subset Accuracy	0.003	0.003	0.002	0.003
	Hamming Loss	0.408	0.407	0.401	0.395
	Accuracy	0.222	0.224	0.230	0.238
	Precision	0.337	0.339	0.348	0.357
	Recall	0.337	0.339	0.348	0.357
	F_1 Measure	0.337	0.339	0.348	0.357
Label-based	Macro-Precision	0.359	0.355	0.369	0.373
	Macro-Recall	0.335	0.342	0.348	0.358
	Macro-F_1	0.267	0.289	0.296	0.326
	Micro-Precision	0.337	0.339	0.348	0.357
	Micro-Recall	0.337	0.339	0.348	0.357
	Micro-F_1	0.337	0.339	0.348	0.357
Rank-based	One-Error	0.664	0.664	0.646	0.637
	Coverage$_1$	1.591	1.610	1.504	1.451
	Coverage$_2$	4.185	4.179	4.112	3.981
	Coverage$_3$	6.889	6.818	6.877	6.675
	Coverage$_4$	9.688	9.606	9.740	9.501
	Ranking Loss	1.268	1.252	1.213	1.206
	Average Precision	0.462	0.464	0.472	0.480

(c) Cross-validation results on mixtures of four instruments

tests set are different, albeit related, we cannot carry out cross-validation without first creating independent datasets. For the set of solo instruments for each instrument, we randomly select 50% of the examples and use these as our training set. The remaining 50% of the solo examples are used to generate the polyphonic mixtures, as described in Section 5.6. Then the process is repeated, switching the roles of two halves of the data. This procedure ensures a complete decoupling of the training and test sets. We repeat this process four more times, to create five folds of training and test sets. For each fold, we create at least 200 examples of each instrument for a total of 2600 mixtures. We repeat this process for all instruments across the four datasets. We carry out our multi-label classification scheme for each of the folds, averaging the results normalizing by the ten folds.

The results of this cross-validation experiments are given as Table 9.10. We observe consistency across the measures across the folds. The cross-validation results track with the full-dataset results given in Tables 9.2 − 9.4 across the two, three, and four mixture experiments. Even the small MUMS dataset, which for some instruments, such as the Alto Saxophone, contribute only seven examples to each fold of the training set. Nevertheless, the results track with the MUMS scores in the full self-classification experiment. These results imply that the effect of the dependency between a training set of solo examples and a test set of mixtures created from these solo examples is negligible in our experiments for our levels of efficacy. As the field of multi-label instrument matures and multi-label classification results improve, this question of dataset bias should be revisited.

9.4 Multi-label Cross-Dataset Experiments

In this experiment, we explore cross-dataset classification of polyphonic mixtures of two, three, or four instruments. These results are shown in Tables 9.11 − 9.13. In these experiments, one dataset of solo instruments is used for training, and the classifier is evaluated on all four datasets. If the training and test datasets come from the same original data source, the result is marked in boldface in the tables. For each polyphony of two, three, and four, 4 × 4 experiments are run. For practicality, these experiments test on only a subset of the polyphonic datasets given in Section 5.6. For each instrument, at least 250 mixtures are considered, resulting in $13 \times 250 = 3250$ test examples in each experiment. These examples are selected at random without replacement, ensuring at least 250 mixtures containing each instrument.

As expected, there is some degradation when testing on a dataset that differs from the training dataset. However, these differences are small, consistent with our

Table 9.11: Results of the cross dataset experiments for mixtures of two instruments. Column headers show the test dataset. Self-classify dataset is shown in bold.

Type	**Metric**	**MUMS**	**MIS**	**RWC**	**PHO**
Example-based	Subset Accuracy	**0.023**	0.016	0.020	0.020
	Hamming Loss	**0.249**	0.252	0.253	0.250
	Accuracy	**0.135**	0.126	0.125	0.133
	Precision	**0.191**	0.180	0.177	0.189
	Recall	**0.191**	0.180	0.177	0.189
	F_1 Measure	**0.191**	0.180	0.177	0.189
Label-based	Macro-Precision	**0.179**	0.193	0.172	0.180
	Macro-Recall	**0.190**	0.178	0.177	0.187
	Macro-F_1	**0.155**	0.146	0.142	0.152
	Micro-Precision	**0.191**	0.180	0.177	0.189
	Micro-Recall	**0.191**	0.180	0.177	0.189
	Micro-F_1	**0.191**	0.180	0.177	0.189
Rank-based	One-Error	**0.804**	0.827	0.826	0.813
	Coverage$_1$	**2.964**	3.163	3.164	3.093
	Coverage$_2$	**6.859**	7.277	7.234	7.239
	Ranking Loss	**0.702**	0.714	0.711	0.712
	Average Precision	**0.370**	0.354	0.354	0.361

(a) MUMS training set

Type	**Metric**	**MUMS**	**MIS**	**RWC**	**PHO**
Example-based	Subset Accuracy	0.016	**0.032**	0.017	0.022
	Hamming Loss	0.260	**0.246**	0.253	0.252
	Accuracy	0.109	**0.144**	0.123	0.128
	Precision	0.155	**0.200**	0.177	0.181
	Recall	0.155	**0.200**	0.177	0.181
	F_1 Measure	0.155	**0.200**	0.177	0.181
Label-based	Macro-Precision	0.172	**0.201**	0.199	0.168
	Macro-Recall	0.156	**0.199**	0.176	0.188
	Macro-F_1	0.127	**0.172**	0.149	0.152
	Micro-Precision	0.155	**0.200**	0.177	0.181
	Micro-Recall	0.155	**0.200**	0.177	0.181
	Micro-F_1	0.155	**0.200**	0.177	0.181
Rank-based	One-Error	0.856	**0.794**	0.827	0.816
	Coverage$_1$	3.382	**2.945**	3.206	3.123
	Coverage$_2$	7.494	**7.013**	7.360	7.149
	Ranking Loss	0.747	**0.708**	0.722	0.715
	Average Precision	0.330	**0.372**	0.350	0.358

(b) MIS training set

Type	**Metric**	**MUMS**	**MIS**	**RWC**	**PHO**
Example-based	Subset Accuracy	0.029	0.029	**0.031**	0.025
	Hamming Loss	0.237	0.243	**0.236**	0.245
	Accuracy	0.163	0.151	**0.165**	0.145
	Precision	0.229	0.212	**0.232**	0.205
	Recall	0.229	0.211	**0.232**	0.204
	F_1 Measure	0.229	0.211	**0.232**	0.205
Label-based	Macro-Precision	0.215	0.204	**0.236**	0.201
	Macro-Recall	0.228	0.209	**0.232**	0.205
	Macro-F_1	0.214	0.194	**0.215**	0.187
	Micro-Precision	0.229	0.211	**0.232**	0.205
	Micro-Recall	0.229	0.211	**0.232**	0.204
	Micro-F_1	0.229	0.211	**0.232**	0.205
Rank-based	One-Error	0.764	0.778	**0.763**	0.803
	Coverage$_1$	2.557	2.761	**2.584**	2.912
	Coverage$_2$	6.651	6.889	**6.626**	6.850
	Ranking Loss	0.678	0.654	**0.669**	0.675
	Average Precision	0.397	0.391	**0.399**	0.379

(c) RWC training set

Type	**Metric**	**MUMS**	**MIS**	**RWC**	**PHO**
Example-based	Subset Accuracy	0.024	0.019	0.020	**0.023**
	Hamming Loss	0.243	0.244	0.246	**0.243**
	Accuracy	0.147	0.144	0.140	**0.149**
	Precision	0.209	0.206	0.200	**0.212**
	Recall	0.209	0.206	0.200	**0.211**
	F_1 Measure	0.209	0.206	0.200	**0.211**
Label-based	Macro-Precision	0.201	0.196	0.201	**0.200**
	Macro-Recall	0.210	0.203	0.201	**0.213**
	Macro-F_1	0.175	0.172	0.201	**0.177**
	Micro-Precision	0.209	0.206	0.200	**0.211**
	Micro-Recall	0.209	0.206	0.200	**0.211**
	Micro-F_1	0.209	0.206	0.200	**0.211**
Rank-based	One-Error	0.780	0.793	0.815	**0.796**
	Coverage$_1$	2.834	2.879	2.915	**2.703**
	Coverage$_2$	7.099	7.025	7.010	**6.661**
	Ranking Loss	0.706	0.671	0.699	**0.680**
	Average Precision	0.375	0.377	0.367	**0.384**

(d) PHO training set

Table 9.12: Results of the cross dataset experiments for mixtures of three instruments. Column headers show the test dataset. Self-classify dataset is shown in bold.

Type	Metric	MUMS	MIS	RWC	PHO
Example-based	Subset Accuracy	**0.006**	0.005	0.004	0.003
	Hamming Loss	**0.343**	0.348	0.346	0.343
	Accuracy	**0.169**	0.162	0.165	0.169
	Precision	**0.256**	0.246	0.251	0.257
	Recall	**0.256**	0.246	0.251	0.257
	F_1 Measure	**0.256**	0.246	0.251	0.257
Label-based	Macro-Precision	**0.292**	0.262	0.270	0.249
	Macro-Recall	**0.257**	0.244	0.249	0.257
	Macro-F_1	**0.204**	0.194	0.197	0.201
	Micro-Precision	**0.256**	0.246	0.251	0.257
	Micro-Recall	**0.256**	0.246	0.251	0.257
	Micro-F_1	**0.256**	0.246	0.251	0.257
Rank-based	One-Error	**0.759**	0.764	0.777	0.762
	Coverage$_1$	**2.225**	2.329	2.302	2.268
	Coverage$_2$	**5.381**	5.587	5.537	5.550
	Coverage$_3$	**8.641**	8.910	8.835	8.811
	Ranking Loss	**1.023**	1.027	1.024	1.021
	Average Precision	**0.402**	0.395	0.393	0.399

(a) MUMS training set

Type	Metric	MUMS	MIS	RWC	PHO
Example-based	Subset Accuracy	0.003	**0.006**	0.008	0.006
	Hamming Loss	0.351	**0.340**	0.345	0.345
	Accuracy	0.156	**0.174**	0.167	0.166
	Precision	0.239	**0.262**	0.252	0.252
	Recall	0.239	**0.262**	0.252	0.252
	F_1 Measure	0.239	**0.262**	0.252	0.252
Label-based	Macro-Precision	0.215	**0.280**	0.241	0.230
	Macro-Recall	0.240	**0.265**	0.251	0.252
	Macro-F_1	0.189	**0.224**	0.203	0.204
	Micro-Precision	0.239	**0.262**	0.252	0.252
	Micro-Recall	0.239	**0.262**	0.252	0.252
	Micro-F_1	0.239	**0.262**	0.252	0.252
Rank-based	One-Error	0.786	**0.750**	0.767	0.761
	Coverage$_1$	2.367	**2.192**	2.311	2.251
	Coverage$_2$	5.628	**5.347**	5.561	5.507
	Coverage$_3$	8.944	**8.726**	8.869	8.802
	Ranking Loss	1.043	**1.012**	1.034	1.028
	Average Precision	0.386	**0.407**	0.395	0.400

(b) MIS training set

Type	**Metric**	**MUMS**	**MIS**	**RWC**	**PHO**
Example-based	Subset Accuracy	0.005	0.006	**0.009**	0.007
	Hamming Loss	0.325	0.333	**0.326**	0.335
	Accuracy	0.198	0.185	**0.196**	0.182
	Precision	0.296	0.278	**0.294**	0.274
	Recall	0.296	0.278	**0.294**	0.274
	F_1 Measure	0.296	0.278	**0.294**	0.274
Label-based	Macro-Precision	0.286	0.281	**0.299**	0.281
	Macro-Recall	0.295	0.279	**0.290**	0.273
	Macro-F_1	0.273	0.261	**0.268**	0.251
	Micro-Precision	0.296	0.278	**0.294**	0.274
	Micro-Recall	0.296	0.278	**0.294**	0.274
	Micro-F_1	0.296	0.278	**0.294**	0.274
Rank-based	One-Error	0.682	0.710	**0.700**	0.724
	Coverage$_1$	1.846	2.008	**1.882**	2.076
	Coverage$_2$	5.002	5.194	**5.138**	5.314
	Coverage$_3$	8.371	8.636	**8.438**	8.487
	Ranking Loss	0.954	0.936	**0.956**	0.969
	Average Precision	0.440	0.428	**0.437**	0.423

(c) RWC training set

Type	**Metric**	**MUMS**	**MIS**	**RWC**	**PHO**
Example-based	Subset Accuracy	0.006	0.005	0.006	**0.007**
	Hamming Loss	0.337	0.339	0.339	**0.329**
	Accuracy	0.178	0.175	0.176	**0.192**
	Precision	0.269	0.266	0.267	**0.287**
	Recall	0.269	0.266	0.267	**0.287**
	F_1 Measure	0.269	0.266	0.267	**0.287**
Label-based	Macro-Precision	0.275	0.265	0.262	**0.296**
	Macro-Recall	0.270	0.266	0.264	**0.287**
	Macro-F_1	0.230	0.229	0.228	**0.250**
	Micro-Precision	0.269	0.266	0.267	**0.287**
	Micro-Recall	0.269	0.266	0.267	**0.287**
	Micro-F_1	0.269	0.266	0.267	**0.287**
Rank-based	One-Error	0.725	0.735	0.737	**0.725**
	Coverage$_1$	2.111	2.146	2.137	**1.957**
	Coverage$_2$	5.365	5.444	5.399	**5.121**
	Coverage$_3$	8.711	8.785	8.721	**8.430**
	Ranking Loss	1.017	1.005	1.022	**0.976**
	Average Precision	0.412	0.406	0.408	**0.426**

(d) PHO training set

Table 9.13: Results of the cross dataset experiments for mixtures of four instruments. Column headers show the test dataset. Self-classify dataset is shown in bold.

Type	**Metric**	**MUMS**	**MIS**	**RWC**	**PHO**
Example-based	Subset Accuracy	**0.001**	0.002	0.002	0.003
	Hamming Loss	**0.413**	0.416	0.417	0.408
	Accuracy	**0.215**	0.212	0.210	0.222
	Precision	**0.329**	0.324	0.322	0.337
	Recall	**0.329**	0.324	0.322	0.337
	F_1 Measure	**0.329**	0.324	0.322	0.337
Label-based	Macro-Precision	**0.355**	0.323	0.333	0.356
	Macro-Recall	**0.327**	0.322	0.321	0.333
	Macro-F_1	**0.259**	0.252	0.249	0.268
	Micro-Precision	**0.329**	0.324	0.322	0.337
	Micro-Recall	**0.329**	0.324	0.322	0.337
	Micro-F_1	**0.329**	0.324	0.322	0.337
Rank-based	One-Error	**0.710**	0.684	0.704	0.680
	Coverage$_1$	**1.721**	0.694	1.733	1.662
	Coverage$_2$	**4.309**	4.366	4.417	4.278
	Coverage$_3$	**6.904**	7.134	7.136	7.033
	Coverage$_4$	**9.602**	9.894	9.818	9.772
	Ranking Loss	**1.266**	1.274	1.277	1.252
	Average Precision	**0.453**	0.451	0.446	0.458

(a) MUMS training set

Type	**Metric**	**MUMS**	**MIS**	**RWC**	**PHO**
Example-based	Subset Accuracy	0.000	**0.002**	0.002	0.002
	Hamming Loss	0.420	**0.408**	0.416	0.413
	Accuracy	0.208	**0.222**	0.212	0.216
	Precision	0.318	**0.337**	0.324	0.329
	Recall	0.318	**0.337**	0.324	0.329
	F_1 Measure	0.318	**0.337**	0.324	0.329
Label-based	Macro-Precision	0.328	**0.356**	0.319	0.338
	Macro-Recall	0.317	**0.336**	0.323	0.330
	Macro-F_1	0.260	**0.294**	0.270	0.276
	Micro-Precision	0.318	**0.337**	0.324	0.329
	Micro-Recall	0.318	**0.337**	0.324	0.329
	Micro-F_1	0.318	**0.337**	0.324	0.329
Rank-based	One-Error	0.726	**0.663**	0.690	0.687
	Coverage$_1$	1.817	**1.587**	1.709	1.693
	Coverage$_2$	4.474	**4.195**	4.355	4.311
	Coverage$_3$	7.172	**6.823**	7.031	6.989
	Coverage$_4$	9.879	**9.628**	9.807	9.697
	Ranking Loss	1.300	**1.244**	1.279	1.274
	Average Precision	0.440	**0.464**	0.451	0.454

(b) MIS training set

s

Type	**Metric**	**MUMS**	**MIS**	**RWC**	**PHO**
Example-based	Subset Accuracy	0.004	0.002	**0.003**	0.002
	Hamming Loss	0.399	0.405	**0.397**	0.407
	Accuracy	0.233	0.225	**0.235**	0.222
	Precision	0.351	0.341	**0.355**	0.339
	Recall	0.351	0.341	**0.355**	0.339
	F_1 Measure	0.351	0.341	**0.355**	0.339
Label-based	Macro-Precision	0.351	0.349	**0.362**	0.339
	Macro-Recall	0.351	0.341	**0.354**	0.336
	Macro-F_1	0.323	0.320	**0.323**	0.307
	Micro-Precision	0.351	0.341	**0.355**	0.339
	Micro-Recall	0.351	0.341	**0.355**	0.339
	Micro-F_1	0.351	0.341	**0.355**	0.339
Rank-based	One-Error	0.632	0.657	**0.648**	0.643
	Coverage$_1$	1.456	1.541	**1.452**	1.510
	Coverage$_2$	4.030	4.168	**4.044**	4.209
	Coverage$_3$	6.727	6.787	**6.744**	6.953
	Coverage$_4$	9.581	9.617	**9.586**	9.583
	Ranking Loss	1.185	1.179	**1.167**	1.199
	Average Precision	0.483	0.472	**0.480**	0.472

(c) RWC training set

Type	**Metric**	**MUMS**	**MIS**	**RWC**	**PHO**
Example-based	Subset Accuracy	0.003	0.001	0.003	**0.002**
	Hamming Loss	0.409	0.408	0.410	**0.401**
	Accuracy	0.220	0.221	0.220	**0.230**
	Precision	0.335	0.336	0.334	**0.348**
	Recall	0.335	0.336	0.334	**0.348**
	F_1 Measure	0.335	0.336	0.334	**0.348**
Label-based	Macro-Precision	0.354	0.350	0.335	**0.355**
	Macro-Recall	0.354	0.335	0.334	**0.347**
	Macro-F_1	0.354	0.299	0.291	**0.308**
	Micro-Precision	0.335	0.336	0.334	**0.348**
	Micro-Recall	0.335	0.336	0.334	**0.348**
	Micro-F_1	0.335	0.336	0.334	**0.348**
Rank-based	One-Error	0.663	0.670	0.663	**0.651**
	Coverage$_1$	1.624	1.599	1.637	**1.500**
	Coverage$_2$	4.275	4.265	4.299	**4.070**
	Coverage$_3$	6.968	6.921	6.969	**6.701**
	Coverage$_4$	9.752	9.725	9.735	**9.445**
	Ranking Loss	1.266	1.248	1.274	**1.215**
	Average Precision	0.461	0.460	0.460	**0.474**

(d) PHO training set

cross-dataset experiments on monophonic classification given in Section 8.2 and far more encouraging that the cross-dataset results given by [34]. Also of interest is the relative consistency in the differences between the datasets across the set of evaluation measures. Since the number of monophonic training examples differed greatly between instruments across datasets (see Table 5.2), this result shows a consistency in classification across the datasets that implies that our method is not skewed based on the distributions of class labels in the training set.

There are very few studies that show any cross-data results, however, a recent study, discussed in Section 3.2.1.3 , deserves revisiting here. Duan *et al.* considered 13 instruments from the RWC and MIS datasets [193]. Although published only recently and long after we chose our datasets, coincidentally, the authors consider the exact set of 13 instruments considered in this dissertation. However, the authors consider only the *mezzo-forte* dynamic level, while our studies consider all three dynamic levels from those datasets. The authors also only consider a short sustained portion of the note, disregarding the attack and the decay portions of the signal. As in our approach, the authors considers only one time window. However, the authors use a different feature space and a multi-label SVM classifer. Most significantly, their approach relies on a score-informed source separation algorithm, which requires the set of true pitches as input. Our approach does not require knowledge of the pitches present, nor uses a multi-pitch finding algorithm.

Despite these difference, this study contains the same number of labels $q = 13$ and the same musical instruments, making it the best candidate of the studies in the literature for comparison to our work. The authors uses monophonic samples from the MIS dataset and tested on samples from the RWC dataset, as we have done given in Tables 9.11b − 9.13b. Unfortunately, the authors provide only this one cross-dataset result. The authors report an accuracy of about 48% for monophonic classification, 37% for two instruments, 32% for three instruments, and 30% for four instruments[1]. The authors do not supply their definition of accuracy, but report a value of chance of $1/13 = 8\%$, which is incorrect if the authors are limiting the relative label space to the problem's cardinality. This implies the authors are counting true positives but ignoring the false positives. Nevertheless, it can be noted that their approach degrades as the cardinality increases while accuracy increases. For four instruments, their system reports around 30% while we report an F_1 score of 33%. Our system also outperforms theirs on the task of monophonic classification. Their approach appears to outperform ours for two and three instruments. However, it is very important to note that this comparison is not scientific and direct comparisons cannot be made, given the broad differences in the data, the features, the approach, and the evaluation measure. The authors also report polyphony of five with accuracy of 27% and polyphony of six with 25% accuracy. Although we will explore mixtures of larger size as future work, our present results indicate that our accuracy will increase rather than decrease as the label cardinality and density increases. The reported accuracy of their system degrades as label cardinality increases.

9.5 Reduced Instrument Set Experiment

Although the results given in the previous two sections appear low when compared objectively to some of the studies given in Section 3.2.1.3, these studies all differ in datasets, experimental design, feature selection, and classification method. Most notably, most of these studies test very few instruments, resulting in a high label density score which correlates to a lesser problem difficulty. In a final experiment, we

[1] These values are approximate and were extracted from a line graph. The study did not report exact numbers.

test our system on dataset with a smaller label size to be consistent with the literature. We chose $q = 5$ with the instruments Cello, Violin, Clarinet, Flute, and Pianoforte. This set of instruments appears in several studies including [202, 180, 189, 181, 204] and similar small sets of instruments appear in [151, 209, 234, 232, 235, 233].

In our principal experiments, we did not consider the pianoforte because it is absent from the PHO dataset and therefore not one of the 13 instruments common to all four datasets, given in Section 5.2. However, we now wish to consider the Pianoforte in order to match the set of instruments given in the above studies. Therefore, for this experiment, we consider only the RWC dataset, our largest set, which contains the Pianoforte. For these five instruments, we create datasets following the procedures given in Chapter 5, learn instrument signatures according to the process given in Chapter 6, extract features as discussed in Chapter 7. The classification results are given in Table 9.14.

In Table 9.15 we show the p-values for our results on the reduced instrument set compared to random permutations for five labels. These p-values are calculated as described in Section 9.3.2 for a reduced instrument set of $q = 5$. These p-values show a strong significance of our results for mixtures of two and three instruments compared to random permutations, around $p = 0.5$ for most measures. The statistical significance for mixtures of four was less strong but still improved over the results for the harder $q = 13$ case and well above chance. This reflects the easier problem of choosing four out of five labels. In all cases, considering random permutations as the baseline, our system was more effective on this reduced problem compared to the thirteen instrument case given in Tables 9.5, 9.6, and 9.7.

The label density of this problem with $q = 5$ is greatly increased compared to our previous experiments, as shown in Table 9.1, rendering it a significantly easier multi-label problem, and subsequently our results are significantly improved. Our subset

Table 9.14: Results for mixtures of two, three, and four instruments for a dataset with the number of labels $q = 5$.

Type	**Metric**	**2-mix**	**3-mix**	**4-mix**
Example-based	Subset Accuracy	0.200	0.181	0.279
	Hamming Loss	0.143	0.151	0.112
	Accuracy	0.420	0.538	0.705
	Precision	0.536	0.677	0.828
	Recall	0.522	0.663	0.802
	F_1 Measure	0.527	0.668	0.812
Label-based	Macro-Precision	0.572	0.705	0.841
	Macro-Recall	0.522	0.665	0.802
	Macro-F_1	0.528	0.663	0.802
	Micro-Precision	0.537	0.677	0.828
	Micro-Recall	0.522	0.663	0.802
	Micro-F_1	0.530	0.670	0.815
Rank-based	One-Error	0.414	0.246	0.114
	Coverage$_1$	0.609	0.291	0.115
	Coverage$_2$	2.301	1.683	1.271
	Coverage$_3$	–	3.097	2.458
	Coverage$_4$	–	–	3.599
	Ranking Loss	0.332	0.355	0.248
	Average Precision	0.742	0.819	0.907

Table 9.15: p-values of results compared to random chance for the reduced dataset experiments

Type	Metric	2-mix	3-mix	4-mix
Example-based	Subset Accuracy	0.054	0.048	0.121
	Hamming Loss	0.054	0.048	0.121
	Accuracy	0.054	0.048	0.121
	Precision	0.054	0.048	0.121
	Recall	0.054	0.048	0.121
	F_1 Measure	0.054	0.048	0.121
Rank-based	One-Error	0.200	0.293	0.407
	Coverage_1	0.200	0.293	0.407
	Coverage_2	0.157	0.154	0.315
	Coverage_3	–	0.197	0.218
	Coverage_4	–	–	0.121
	Ranking Loss	0.104	0.101	0.218
	Average Precision	0.151	0.154	0.218

accuracy, the most strict measure, is 20% for mixtures of two, 18% for mixtures of three, and 28% for mixtures of four. This is a significant improvement over the $< 1\%$ subset accuracy of our $q = 13$ experiments. The F_1 measure is 0.53 for two instruments, 0.67 for three instruments, and 0.81 for four instruments. This is more consistent with the results given for similar label density problems, although direct comparisons between studies are anecdotal at best. The one-error measure has not been reported by any other instrument classification studies, although it is perhaps analogous to the studies that seek to recognize a single dominant instrument from a mixture of instruments. For mixtures of four, our approach selected a true label as the top ranked label 89% of the time. Many of the studies that test mixtures from datasets of only five instruments, use approaches that rely on temporal features, multi-pitch finding algorithms, or frame-based evaluation schemes, and none consider multiple dynamic level. Our approach, tested under more adversarial conditions including multiple dynamic levels, multiple performers, no consideration of temporal features, simple feature space of single amplitude values, can compete with other systems when tested on simple multi-label problems with a high label density, such as $q = 5$. The results on the simpler $q = 5$ problems emphasize the promise of our results in the hard $q = 13$ multi-label problem, especially considering the polyphonic classification community has only recently begun examining harder multi-label problems and cross-dataset evaluation and there are few studies available for comparison.

9.6 Conclusion

In this chapter we present experimental results evaluating the classification of polyphonic mixtures ranging from two to four instruments, analayzing our results by instrument family label and individual instrument accuracy. We report a comprehensive set of multi-label evaluation metrics, the first study in the domain of the polyphonic instrument classification to provide all possible measures. We contribute the first cross-validation study in the domain concerning solo instrument training sets and derived polyphonic mixture test sets. Next, we evaluate our system in a 4×4 cross-dataset study, the largest cross-dataset study in the domain, demonstrating the ability of our system to generalize across datasets. We compare our work to the only study with similar label density and same instrument space, showing comparable performance. Our results show consistent improvement as the number of labels increases from two to four. Lastly, we provide a case study of a multi-label problem with only $q = 5$ instruments, for anecdotal comparisons to studies using the same reduced instrument set. We show that on the much simpler problem with a very high label density, our system tracks with results reported in the literature.

CHAPTER 10

CONCLUSION

In this dissertation, we present a system for the single- and multi-label classification of polyphonic mixtures of musical instruments coupling a novel binary-relevance feature extraction approach with the widely-used binary-relevance classification approach to multi-label classification. In this chapter, we summarize our contributions to the domains of instrument classification and multi-label classification and present our directions for future work.

10.1 Summary

We briefly summarize our contributions provided in this disseration. First, we demonstrate the ability of a simple feature set of spectral amplitudes to compete with state of the art classifiers for monophonic instrument classifiers that use complex features sets, often those standard in speech recognition tasks, that often overfit the training data and are not extensible to multi-label classification. Additionally, we show statistical dependencies between an instrument's harmonic partials through our seminal use of classification with Bayesian networks. This result underscores the importance of capturing all sequential partials in the feature extraction stage, an observation that heavily influenced the design of our feature extraction approach for multi-label mixtures.

Next, we present our four datasets, the largest data repository in the monophonic or polyphonic instrument classification literature. We consider the three datasets most frequently mentioned in the literature and a new, large dataset comprised of lower quality MP3 examples. We consider multiple performers for each instrument, between three and five dynamic levels, and a large set of 13 musical instruments. Furthermore, we present a novel amplitude normalization scheme that extracts as features the ratios of the amplitudes of the partial to the amplitude of the fundamental frequency. This normalization scheme considers the instrument's timbre with a reference point to itself, allowing generalization across different dynamic levels and recording procedures.

Building on our work with monophonic classification of musical instruments, we designed a system for instrument classification that satisfies the qualities of scalability, generalizability, and practicality. We present a novel data-driven approach to learn locations of instrument's significant spectral energy to inform the feature extraction approach. We validate these signatures showing the application of a signature optimized for one dataset to find relevant features in a different dataset, arguing our learned signatures capture areas relevant to an instrument's timbre, rather than acoustic properties of a specific dataset.

We propose a novel extension to the common binary-relevance approach to multi-label classification. Our binary-relevance feature extraction scheme permits consideration of a unique feature space for each binary-relevance classifier, albeit at the expense of an additional calculation for each binary classifier. However, this approach scales in complexity with the number of models, as does the binary-relevance classification approach. This design allows our system to scale linearly as the number of class labels increases, unlike many other mult-label approaches that grow exponentially with the number of class labels, satisfying our criteria of scalability.

In this dissertation, we evaluate our system on polyphonic mixtures of two, three, and four instruments. We consider only a single time window in which the signals overlap. Our system does not rely on temporal features or signal alignment that require knowledge of the timing of the attack, sustain, and decay of the signal. Expectation of such knowledge is impractical for most applications of instrument classification. Our system does not rely on any musical score information or multi-pitch finding algorithms, fields themselves still in active development. We argue such an approach is necessary for practicality of any multi-label instrument classification system when approaching real-world data. Additionally, we examine the bias between solo training sets and derived mixture datasets used for testing in the only cross-validation study of its kind in the domain.

We provide important experimental results consisting of the largest cross-dataset in the instrument classification literature. We demonstrate our ability generalizes between datasets with losses in accuracy much smaller than in the literature. This shows the ability of our system to capture instrument's timbre rather than overfitting the training datasets. Furthermore, we demonstrate these results on enormous datasets that normalize notes by frequency, musical dynamic level, and differing levels of audio compression and recording levels. We demonstrate this ability to generalize between small (MUMS), medium (MIS), and large (RWC) datasets. We also demonstrate the ability to generalize these three datasets, which are all recorded in ideal recording environments, with our newly proposed PHO dataset, which contains low quality MP3s.

Additionally, we demonstrate the ability of our system to improve with the label cardinality and label density as the number of labels increase from two, three, and four instruments, an expected result that differs from a relevant recent result in the literature despite comparable performance for the four instrument classification problem. We demonstrate that our accuracy correlates to the difficulty of the multi-label problem, showing desirable and comparable empirical results on a small, five instrument grouping commonly reported in the literature. We also report comprehensive results on our large set of 13 instrument reporting F_1 scores comparable to recent efforts in the field.

Lastly, we present a comprehensive set of all multi-label evaluation metrics available in the multi-label classification literature. Unfortunately, the polyphonic instrument classification literature rarely presents more than one metric and often nebulous or poorly-defined measures of success. We hope to align the instrument classification community with the measures common in other multi-label domains. Additionally we point to the utility of rank-based metrics, which have not previously been reported from studies in this domain, as well as our proposed extension to the Coverage measure for understanding the confusions in the polyphonic classification, an area in which there are frequently confusions between similar instruments as the complexity of the problem increases.

10.2 Future Work

The area of multi-label classification of polyphonic mixtures of instruments is a difficult problem, only obtaining serious attention in the last few years. As our primary goal in this dissertation was to develop a system for generalizability between data sources, we considered only a simple albeit effective feature space, leaving ample opportunity for refinement. We intend to explore optimization of the feature space in three ways.

In our present work, we consider only simple maximum amplitudes for each partial, albeit filtered by a spectral mask trained for that instrument. As future work, we will first explore optimizing the feature space for each instrument to further exploit our binary-relevance feature extraction approach. By comparing the instrument signature empirically, we can select features for each instrument that best differentiate it from other instruments. Additionally, we will consider other feature spaces. The extensible desire of our spectral mask allows the use of many other spectral features from the monophonic classification literature; although, cross-dataset validation experiments are necessary to avoid overfitting the feature space to the training data, a common issue in the domain. Additionally, we will explore training our binary instrument models with "noisy" training data consisting on the instrument playing in a mixture with other instruments.

Secondly, we will extend our approach to consider temporal features. We plan to extend our Bayesian networks for multi-label classification, considering frequency and temporal dependencies between spectra of adjacent time frames. Lastly, we wish to explore a feature weighting scheme informed by our data-driven signature learning stage. In present work, we consider each learned cluster as a feature in our feature space. We will adapt our k-means clustering method to capture cluster density information, informing the relative coverage of each cluster to the dataset. This approach will allow pruning of features less common across the entire dataset for each instrument.

Our present system does not require score information nor rely on a multi-pitch finding algorithm. Currently, we consider each significant peak as a potential fundamental frequency for each instrument. In this dissertation, we wished to argue the efficacy of our system independent of a coupling with a multi-pitch finding algorithm. We will explore coupling our system with various multi-pitch finding algorithms, which will allow us to consider fewer peaks as hypothetical fundamental frequencies, improving the complexity of our approach. We hypothesize this approach will help reduce the number of false positives resulting from confusions of similar instruments.

Lastly, we wish to analyze the confusions of our system, by empirically exploring correlations to musical dynamic level and the musical intervals of the polyphonic mixtures. To analyze the effect of the dynamic levels of the contributing individual instruments, we need to create new datasets, refactoring our class labels to adapt to our evaluation measures. This analysis will allow us to explore the intuition that louder musical instruments are more frequently classified correctly compared to the softer notes of the mixture. Furthermore, we intend to explore analysis of the results of our system in regard to the frequencies of the contributing pitches. Our present work allowed mixtures of any possible musical interval, including the unison and octave in which many harmonic partials are likely to overlap, despite our spectral filters. Other musical intervals, such as the minor-second or major-seventh will have far fewer harmonic partials in common. We anticipate strong correlation between the musical intervals of the mixture and classification rates and will explore this topic experimentally.

’

REFERENCES CITED

[1] Stephen McAdams. Musical timbre perception. *The Psychology of Music*, pages 35–68, 2012.

[2] Kailash Patil, Daniel Pressnitzer, Shihab Shamma, Mounya Elhilali. Music in our ears: the biological bases of musical timbre perception. *PLoS Computational Biology*, 8(11):e1002759, 2012.

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

[4] Tao Li, Mitsunori Ogihara, Qi Li. A comparative study on content-based music genre classification. *Conference on Research and Development in Informant Retrieval (SIGIR)*, pages 282–289. ACM, 2003.

[5] Soujanya Poria, Alexander Gelbukh, Amir Hussain, Sivaji Bandyopadhyay, Newton Howard. Music genre classification: A semi-supervised approach. *Pattern Recognition*, pages 254–263. Springer, 2013.

[6] Yin-Fu Huang, Sheng-Min Lin, Huan-Yu Wu, Yu-Siou Li. Music genre classification based on local feature selection using a self-adaptive harmony search algorithm. *Data & Knowledge Engineering*, 92:60–76, 2014.

[7] George Tzanetakis, Andreye Ermolinskyi, Perry Cook. Beyond the query-by-example paradigm: New query interfaces for music information retrieval. *International Computer Music Conference*, pages 177–183, 2002.

[8] Wei-Ho Tsai, Hung-Ming Yu, Hsin-Min Wang. Query-by-example technique for retrieving cover versions of popular songs with similar melodies. *Proceeding of the International Symposium on Music Information Retrieval (ISMIR)*, Volume 5, pages 183–190, 2005.

[9] Martín Rocamora, Pablo Cancela, Alvaro Pardo. Query by humming: automatically building the database from music recordings. *Pattern Recognition Letters*, 36:272–280, 2014.

[10] Mark D Plumbley, Samer A Abdallah, Juan Pablo Bello, Mike E Davies, Giuliano Monti, Mark B Sandler. Automatic music transcription and audio source separation. *Cybernetics & Systems*, 33(6):603–627, 2002.

[11] Matti P Ryynänen, Anssi P Klapuri. Automatic transcription of melody, bass line, and chords in polyphonic music. *Computer Music Journal*, 32(3):72–86, 2008.

[12] Nancy Bertin, Roland Badeau, Emmanuel Vincent. Enforcing harmonicity and smoothness in Bayesian non-negative matrix factorization applied to polyphonic music transcription. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):538–549, 2010.

[13] Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, Anssi Klapuri. Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems*, 41(3):407–434, 2013.

[14] Yipeng Li, DeLiang Wang. Separation of singing voice from music accompaniment for monaural recordings. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(4):1475–1487, 2007.

[15] Zhiyao Duan, Bryan Pardo. Soundprism: An online system for score-informed source separation of music audio. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1205–1215, 2011.

[16] Joachim Fritsch, Mark D Plumbley. Score informed audio source separation using constrained nonnegative matrix factorization and score synthesis. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 888–891. IEEE, 2013.

[17] Sebastian Ewert, Bryan Pardo, Meinard Müller, M Plumbley. Score-informed source separation for musical audio recordings: an overview. *Signal Processing Magazine*, 31(3):116–124, 2014.

[18] George Tzanetakis, Perry Cook. Multifeature audio segmentation for browsing and annotation. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 103–106. IEEE, 1999.

[19] Steven R Ness, Anthony Theocharis, George Tzanetakis, Luis Gustavo Martins. Improving automatic music tag annotation using stacked generalization of probabilistic svm outputs. *International Conference on Multimedia*, pages 705–708. ACM, 2009.

[20] Philippe Hamel, Simon Lemieux, Yoshua Bengio, Douglas Eck. Temporal pooling and multiscale learning for automatic annotation and ranking of music audio. *International Symposium on Music Information Retrieval (ISMIR)*, pages 729–734, 2011.

[21] Roger B Dannenberg, Christopher Raphael. Music score alignment and computer accompaniment. *Communications of the ACM (CACM)*, 49(8):38–43, 2006.

[22] Arshia Cont. A coupled duration-focused architecture for real-time music-to-score alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):974–987, 2010.

[23] Nicola Montecchio, Arshia Cont. A unified approach to real time audio-to-score and audio-to-audio alignment using sequential montecarlo inference techniques. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 193–196. IEEE, 2011.

[24] Shinji Sako, Ryuichi Yamamoto, Tadashi Kitamura. Ryry: A real-time score-following automatic accompaniment playback system capable of real performances with errors, repeats and jumps. *Active Media Technology*, pages 134–145. Springer, 2014.

[25] Yuan Cao Zhang, Diarmuid Ó Séaghdha, Daniele Quercia, Tamas Jambor. Auralist: introducing serendipity into music recommendation. *Conference on Web search and Data Mining*, pages 13–22. ACM, 2012.

[26] Brian McFee, Luke Barrington, Gert Lanckriet. Learning content similarity for music recommendation. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(8):2207–2218, 2012.

[27] Yading Song, Simon Dixon, Marcus Pearce. A survey of music recommendation systems and future perspectives. *International Symposium on Computer Music Modeling and Retrieval*, 2012.

[28] Jongseol Lee, Saim Shin, Dalwon Jang, Sei-Jin Jang, Kyoungro Yoon. Music recommendation system based on usage history and automatic genre classification. *IEEE International Conference on Consumer Electronics (ICCE)*, pages 134–135. IEEE, 2015.

[29] Shigeki Sagayama, Tomohiko Nakamura, Eita Nakamura, Yasuyuki Saito, Hirokazu Kameoka, Nobutaka Ono. Automatic music accompaniment allowing errors and arbitrary repeats and jumps. *Proceedings of Meetings on Acoustics*, Volume 21, page 035003. Acoustical Society of America, 2014.

[30] Estefanía Cano, Gerald Schuller, Christian Dittmar. Pitch-informed solo and accompaniment separation towards its use in music education applications. *European Journal on Advances in Signal Processing (EURASIP)*, 2014(1):1–19, 2014.

[31] James A Moorer. On the transcription of musical sound by computer. *Computer Music Journal*, pages 32–38, 1977.

[32] Ferdinand Fuhrmann, Martín Haro, Perfecto Herrera. Scalability, generality and temporal aspects in automatic recognition of predominant musical instruments in polyphonic music. *International Society for Music Information Retrieval Conference (ISMIR)*, pages 321–326. Citeseer, 2009.

[33] Slim Essid, Gaël Richard, Bertrand David. Instrument recognition in polyphonic music based on automatic taxonomies. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):68–80, 2006.

[34] Arie Livshin, Xavier Rodet. The importance of cross database evaluation in sound classification. *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, 2003.

[35] Jayme Garcia Arnal Barbedo, George Tzanetakis. Musical instrument classification using individual partials. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(1):111–122, 2011.

[36] Jeremiah D Deng, Christian Simmermacher, Stephen Cranefield. A study on feature analysis for musical instrument classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 38(2):429–438, 2008.

[37] Grigorios Tsoumakas, Ioannis Katakis, Ioannis Vlahavas. Mining multilabel data. *Data mining and Knowledge Discovery Handbook*, pages 667–685. Springer, 2010.

[38] Carol L Krumhansl. *Cognitive Foundations of Musical Pitch*, Volume 17. Oxford University Press New York, 1990.

[39] American National Standards Institute (ANSI). *Psychoacoustical Terminology.* New York: ANSI, 1973.

[40] Patrick Donnelly, Charles Limb. *"Music" in the Encyclopedia of Neuroscience*, pages 1151–1158. Elsevier, 2009.

[41] American National Standards Institute (ANSI). *USA Standard: Acoustical Terminology (S1. 1)*, page 45. New York: ANSI, 1960.

[42] Hermann LF Helmholtz, Alexander J Ellis. *On the Sensations of Tone as a Physiological Basis for the Theory of Music.* Cambridge University Press, 2009.

[43] John M Grey. Multidimensional perceptual scaling of musical timbres. *The Journal of the Acoustical Society of America*, 61(5):1270–1277, 1977.

[44] James W Cooley, John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965.

[45] Ronald J Baken, Robert F Orlikoff. *Clinical Measurement of Speech and Voice.* Cengage Learning, 2000.

[46] Liang Sun, Shuiwang Ji, Jieping Ye. *Multi-label dimensionality reduction.* Chapman & Hall/CRC, 2014.

[47] Wen Wu Chin-Hui Lee Gao, Sheng, Tat-Seng Chua. A MFoM learning approach to robust multiclass multi-label text categorization. *Conference on Machine Learning*, page 42. ACM, 2004.

[48] Jinseok Nam, Jungi Kim, Eneldo Loza Mencía, Iryna Gurevych, Johannes Fürnkranz. Large-scale multi-label text classification-revisiting neural networks. pages 437–452. Springer, 2014.

[49] Gulisong Nasierding, Grigorios Tsoumakas, Abbas Z Kouzani. Clustering based multi-label classification for image annotation and retrieval. *IEEE International Conference on Systems, Man and Cybernetics*, pages 4514–4519. IEEE, 2009.

[50] Changhu Wang, Shuicheng Yan, Lei Zhang, Hong-Jiang Zhang. Multi-label sparse coding for automatic image annotation. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1643–1650. IEEE, 2009.

[51] Min-Ling Zhang, Zhi-Hua Zhou. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1338–1351, 2006.

[52] Matthew R Boutell, Jiebo Luo, Xipeng Shen, Christopher M Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.

[53] Tucker Hermans Asmita V. Karandikar Zhou, Howard, James M. Rehg. Movie genre classification via scene categorization. *Proceedings of the International Conference on Multimedia*, pages 747–750. ACM, 2010.

[54] Grigorios Tsoumakas George Kalliris Trohidis, Konstantinos, Ioannis P. Vlahavas. Multilabel classification of music into emotions. *International Conference on Music Information Retrieval (ISMIR)*, Volume 2008, 2008.

[55] Amanda Clare, Ross D King. Knowledge discovery in multi-label phenotype data. *Principles of Data Mining and Knowledge Discovery*, pages 42–53. Springer, 2001.

[56] Robert E Schapire, Yoram Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2):135–168, 2000.

[57] Min-Ling Zhang, Zhi-Hua Zhou. A k-nearest neighbor based algorithm for multi-label classification. *IEEE International Conference on Granular Computing*, Volume 2, pages 718–721. IEEE, 2005.

[58] Alicja Wieczorkowska, Piotr Synak, Zbigniew W Raś. Multi-label classification of emotions in music. *Intelligent Information Processing and Web Mining*, pages 307–315. Springer, 2006.

[59] André Elisseeff, Jason Weston. A kernel method for multi-labelled classification. *Advances in Neural Information Processing Systems*, pages 681–687, 2001.

[60] Shantanu Godbole, Sunita Sarawagi. Discriminative methods for multi-labeled classification. *Advances in Knowledge Discovery and Data Mining*, pages 22–30. Springer, 2004.

[61] Fadi Thabtah, Peter Cowling, Yonghong Peng. MMAC: A new multi-class, multi-label associative classification approach. *IEEE International Conference on Data Mining*, pages 217–224. IEEE, 2004.

[62] Koby Crammer, Yoram Singer. A family of additive online algorithms for category ranking. *The Journal of Machine Learning Research*, 3:1025–1058, 2003.

[63] Nadia Ghamrawi, Andrew McCallum. Collective multi-label classification. *International Conference on Information and Knowledge Management*, pages 195–200. ACM, 2005.

[64] Weiwei Cheng, Eyke Hüllermeier, Krzysztof J Dembczynski. Bayes optimal multilabel classification via probabilistic classifier chains. *International Conference on Machine Learning (ICML)*, pages 279–286, 2010.

[65] Min-Ling Zhang, Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837, 2014.

[66] Grigorios Tsoumakas, Ioannis Vlahavas. Random k-labelsets: An ensemble method for multilabel classification. *European Conference for Machine learning (ECML)*, pages 406–417. Springer, 2007.

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

[68] Jesse Read, Bernhard Pfahringer, Geoff Holmes, Eibe Frank. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333–359, 2011.

[69] Flavia Cristina Bernardini, Rodrigo Barbosa da Silva, Edwin Mitacc Meza, Rio das Ostras-RJ-Brazil. Analyzing the influence of cardinality and density characteristics on multi-label learning. *Encontro Nacional de Inteligência Artificial e Computacional (ENIAC)*, 2013.

[70] Flavia Cristina Bernardini, Rodrigo Barbosa da Silva, Rodrigo Magalhaes Rodovalho, Edwin Benito Mitacc Meza. Cardinality and density measures and their influence to multi-label learning methods. *Submitted to Learning and Nonlinear Models*, 2014.

[71] Leon Bottou, Yoshua Bengio. Convergence properties of the k-means algorithms. *Advances in Neural Information Processing Systems (NIPS)*, pages 585–592, 1994.

[72] Marina Meilă. The uniqueness of a good optimum for k-means. *International Conference on Machine Learning (ICML)*, pages 625–632. ACM, 2006.

[73] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[74] Tom M Mitchell. *Machine Learning*, Volume 45. Burr Ridge, IL: McGraw Hill, 1997.

[75] Thomas M. Cover, Peter E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[76] Nitin Bhatia. Survey of nearest neighbor techniques. *International Journal of Computer Science and Information Security*, 8(2):302–305, 2010.

[77] Christopher JC Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[78] Alex J Smola, Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[79] Ralf Herbrich, Thore Graepel, Klaus Obermayer. Large margin rank boundaries for ordinal regression. *Advances in Neural Information Processing Systems*, pages 115–132, 1999.

[80] Michael PS Brown, William Noble Grundy, David Lin, Nello Cristianini, Charles Walsh Sugnet, Terrence S Furey, Manuel Ares, David Haussler. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Sciences*, 97(1):262–267, 2000.

[81] Edgar Osuna, Robert Freund, Federico Girosi. Training support vector machines: an application to face detection. *IEEE Computer Society Conference of Computer Vision and Pattern Recognition*, pages 130–136. IEEE, 1997.

[82] Ronan Collobert, Samy Bengio. Svmtorch: Support vector machines for large-scale regression problems. *The Journal of Machine Learning Research*, 1:143–160, 2001.

[83] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features machine learning. *European Conference on Machine Learning (ECML)*, 1398:137–142, 1998.

[84] Achmad Widodo, Bo-Suk Yang. Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing*, 21(6):2560–2574, 2007.

[85] Li-Juan Cao, Francis Eng Hock Tay. Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on Neural Networks*, 14(6):1506–1518, 2003.

[86] David A Sadlier, Noel E O'Connor. Event detection in field sports video using audio-visual features and a support vector machine. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(10):1225–1233, 2005.

[87] Vladimir Vapnik. *The nature of statistical learning theory.* Springer, 1999.

[88] Trevor Hastie, Robert Tibshirani. Classification by pairwise coupling. *The Annals of Statistics*, 26(2):451–471, 1998.

[89] Isabelle M. Guyon Boser, Bernhard E., Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. *Annual Workshop on Computational Learning Theory*, pages 144–152. ACM, 1992.

[90] David Meyer, Friedrich Leisch, Kurt Hornik. The support vector machine under test. *Neurocomputing*, 55(1):169–186, 2003.

[91] Robert Hable, Andreas Christmann. On qualitative robustness of support vector machines. *Journal of Multivariate Analysis*, 102(6):993–1007, 2011.

[92] Laura Auria, Rouslan A Moro. Support vector machines (svm) as a technique for solvency analysis. Technical Report, DIW Berlin, German Institute for Economic Research, 2008.

[93] Colin Campbell. Kernel methods: a survey of current techniques. *Neurocomputing*, 48(1):63–84, 2002.

[94] N. Friedman, D. Geiger, M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2):131–163, 1997.

[95] Lawrence Rabiner, Biing-Hwang Juang. *Fundamentals of speech recognition.* Prentice-Hall, Inc., 1993.

[96] Stanley Smith Stevens, John Volkmann, Edwin B Newman. A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, 8:185, 1937.

[97] Douglas O'Shaughnessy. *Speech communication: human and machine.* Universities Press, 1987.

[98] Beth Logan. Mel frequency cepstral coefficients for music modeling. *International Society for Music Information Retrieval Conference (ISMIR)*, 2000.

[99] Perfecto Herrera-Boyer, Geoffroy Peeters, Shlomo Dubnov. Automatic classification of musical instrument sounds. *Journal of New Music Research*, 32(1):3–21, 2003.

[100] I Kaminsky, A Materka. Automatic source identification of monophonic musical instrument sounds. *IEEE International Conference on Neural Networks*, Volume 1, pages 189–194. IEEE, 1995.

[101] Ian Kaminskyj, Tadeusz Czaszejko. Automatic recognition of isolated monophonic musical instrument sounds using k-nnc. *Journal of Intelligent Information Systems*, 24(2-3):199–221, 2005.

[102] Ichiro Fujinaga. Machine recognition of timbre using steady-state tone of acoustic musical instruments. *International Computer Music Conference (ICMC)*, pages 207–10, 1998.

[103] Angela Fraser, Ichiro Fujinaga. Toward real-time recognition of acoustic musical instruments. *International Computer Music Conference (ICMC)*, 1999.

[104] Ichiro Fujinaga, Karl MacMillan. Realtime recognition of orchestral instruments. *International Computer Music Conference (ICMC)*, Volume 141, page 143, 2000.

[105] Giulio Agostini, Maurizio Longari, Emanuele Pollastri. Musical instrument timbres classification with spectral features. *European Journal on Applied Signal Processing (EURASIP)*, 2003:5–14, 2003.

[106] Antti Eronen, Anssi Klapuri. Musical instrument recognition using cepstral coefficients and temporal features. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Volume 2, pages II753–II756. IEEE, 2000.

[107] Antti Eronen. Comparison of features for musical instrument recognition. *IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics*, pages 19–22. IEEE, 2001.

[108] Perfecto Herrera, Alexandre Yeterian, Fabien Gouyon. Automatic classification of drum sounds: a comparison of feature selection methods and classification techniques. *Music and Artificial Intelligence*, pages 69–80. Springer, 2002.

[109] Keith D Martin, Youngmoo E Kim. Musical instrument identification: A pattern-recognition approach. *The Journal of the Acoustical Society of America*, 104:1768, 1998.

[110] Keith Dana Martin. *Sound-source recognition: A theory and computational model*. Doctoral Dissertation, Massachusetts Institute of Technology, 1999.

[111] Geoffroy Peeters. Automatic classification of large musical instrument databases using hierarchical classifiers with inertia ratio maximization. *Audio Engineering Society Convention 115*. Audio Engineering Society, 2003.

[112] Arie A Livshin, Xavier Rodet. Instrument recognition beyond separate notes-indexing continues recordings. *International Computer Music Conference (ICMC)*, 2004.

[113] Arie A Livshin, Xavier Rodet. Musical instrument identification in continuous recordings. *International Conference on Digital Audio Effects*, pages 1–5, 2004.

[114] Arie Livshin, Xavier Rodet. The importance of the non-harmonic residual for automatic musical instrument recognition of pitched instruments. *Audio Engineering Society Convention*. Audio Engineering Society, 2006.

[115] Arie Livshin, Xavier Rodet. The significance of the non-harmonic noise versus the harmonic series for musical instrument recognition. *International Society for Music Information Retrieval Conference (ISMIR)*, pages 95–100, 2006.

[116] Wenxin Jiang, Xin Zhang, Amanda Cohen, Zbigniew W Raś. Multiple classifiers for different features in timbre estimation. *Advances in Intelligent Information Systems*, pages 335–356. Springer, 2010.

[117] Kristoffer Jensen, Jens Arnspang. Binary decision tree classification of musical sounds. *International Computer Music Conference (ICMC)*, 1999.

[118] J. Marques, P.J. Moreno. A Study of Musical Instrument Classification Using Gaussian Mixture Models and Support Vector Machines. *Cambridge Research Laboratory Technical Report Series*, 4, 1999.

[119] Slim Essid, Gaël Richard, Bertrand David. Musical instrument recognition based on class pairwise feature selection. *International Society for Music Information Retrieval Conference (ISMIR)*, 2004.

[120] Slim Essid, Gaël Richard, Bertrand David. Musical instrument recognition by pairwise classification strategies. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1401–1412, 2006.

[121] Cyril Joder, Slim Essid, Gaël Richard. Temporal integration for audio classification with application to musical instrument classification. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(1):174–186, 2009.

[122] Bob L Sturm, Marcela Morvidone, Laurent Daudet. Musical instrument identification using multiscale mel-frequency cepstral coefficients. *European Signal Processing Conference (EUSIPCO)*, pages 477–481, 2010.

[123] Marcela Morvidone, Bob L Sturm, Laurent Daudet. Incorporating scale information with cepstral features: Experiments on musical instrument recognition. *Pattern Recognition Letters*, 31(12):1489–1497, 2010.

[124] Sebastian Krey, Uwe Ligges. Svm based instrument and timbre classification. *Classification as a Tool for Research*, pages 759–766. Springer, 2010.

[125] Uwe Ligges, Sebastian Krey. Feature clustering for instrument classification. *Computational Statistics*, 26(2):279–291, 2011.

[126] Mehmet Erdal Özbek, Claude Delpha, Pierre Duhamel. Musical note and instrument classification with likelihood-frequency-time analysis and support vector machines. *European Signal Processing Conference (EUSIPCO)*, pages 941–945. European Association for Signal Processing, 2007.

[127] Jing Liu, Lingyun Xie. Svm-based automatic classification of musical instruments. *International Conference on Intelligent Computation Technology and Automation (ICICTA)*, Volume 3, pages 669–673. IEEE, 2010.

[128] M Erdal Özbek, Nalan Özkurt, F Acar Savacı. Wavelet ridges for musical instrument classification. *Journal of Intelligent Information Systems*, 38(1):241–256, 2012.

[129] Wolfgang Fohl, Andreas Meisel, Ivan Turkalj. A feature relevance study for guitar tone classification. *International Society for Music Information Retrieval Conference (ISMIR)*, pages 211–216, 2012.

[130] Mizuki Ihara, Shin-ichi Maeda, Shin Ishii. Instrument identification in monophonic music using spectral information. *IEEE International Symposium on Signal Processing and Information Technology.*, pages 595–599. IEEE, 2007.

[131] Huijing Dou, Yan Feng, Yanzhou Qian, Jianchao Shi. Automatic classification between wind and bowstring instrumental music using support vector machine. *Recent Advances in Computer Science and Information Engineering*, pages 205–210. Springer, 2012.

[132] Dirk Van Steelant, Koen Tanghe, Sven Degroeve, Bernard De Baets, Marc Leman, Jean-Pierre Martens, J.P. Martens. Classification of percussive sounds using support vector machines. *Annual Machine Learning Conference of Belgium and The Netherlands (BENELEARN)*, pages 146–152, 2004.

[133] Simon Scholler, Hendrik Purwins. Sparse approximations for drum sound classification. *IEEE Journal of Selected Topics in Signal Processing*, 5(5):933–940, 2011.

[134] Markus Eichhoff, Igor Vatolkin, Claus Weihs. Piano and guitar tone distinction based on extended feature analysis. *Classification and Data Mining*, pages 215–224. Springer, 2013.

[135] Saima Anwar Lashari, Rosziati Ibrahim, Norhalina Senan. Soft set theory for automatic classification of traditional pakistani musical instruments sounds. *International Conference on Computer & Information Science (ICCIS)*, Volume 1, pages 94–99. IEEE, 2012.

[136] Dominique Fourer, Jean-Luc Rouas, Pierre Hanna, Matthias Robine. Automatic timbre classification of ethnomusicological audio recordings. *International Society for Music Information Retrieval Conference (ISMIR)*, 2014.

[137] Dimitrios Fragoulis, Mihalis Exarhos, Constantin Papaodysseus. A general methodology for timbre determination and identification with application to sax-flute. *WSEAS Transactions on Acoustics and Music*, 2(1):38–43, 2005.

[138] Dimitrios Fragoulis, Constantin Papaodysseus, Mihalis Exarhos, George Roussopoulos, Thanasis Panagopoulos, Dimitrios Kamarotos. Automated classification of piano-guitar notes. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(3):1040–1050, 2006.

[139] Shlomo Dubnov, Naftali Tishby. Analysis of sound textures in musical and machine sounds by means of higher order statistical features. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Volume 5, pages 3845–3848. IEEE, 1997.

[140] Shlomo Dubnov, Naftali Tishby, Dalia Cohen. Polyspectra as measures of sound texture and timbre. *Journal of New Music Research*, 26(4):277–314, 1997.

[141] Shlomo Dubnov, Xavier Rodet. Statistical modeling of sound aperiodicities. *International Computer Music Conference (ICMC)*, Volume 18, pages 171–189, 1997.

[142] Shlomo Dubnov, Xavier Rodet. Timbre recognition with combined stationary and temporal features. *International Computer Music Conference (ICMC)*, 1998.

[143] Emmanouil Benetos, Margarita Kotti, Constantine Kotropoulos, Juan José Burred, Gunnar Eisenberg, Martin Haller, Thomas Sikora. Comparison of subspace analysis-based and statistical model-based algorithms for musical instrument classification. *Workshop On Immersive Communication And Broadcast Systems*, 2005.

[144] Emmanouil Benetos, Margarita Kotti, Constantine Kotropoulos. Musical instrument classification using non-negative matrix factorization algorithms. *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1844–1847. IEEE, 2006.

[145] Emmanouil Benetos, Margarita Kotti, Constantine Kotropoulos. Musical instrument classification using non-negative matrix factorization algorithms and subset feature selection. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Volume 5, pages 221–224. IEEE, 2006.

[146] Julio J Carabias-Orti, Tuomas Virtanen, Pedro Vera-Candeas, Nicolás Ruiz-Reyes, Francisco J Cañadas-Quesada. Musical instrument sound multi-excitation model for non-negative spectrogram factorization. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1144–1158, 2011.

[147] Judith C Brown. Computer identification of musical instruments using pattern recognition with cepstral coefficients as features. *The Journal of the Acoustical Society of America*, 105:1933, 1999.

[148] Judith C Brown, Olivier Houix, Stephen McAdams. Feature dependence in the automatic identification of musical woodwind instruments. *The Journal of the Acoustical Society of America*, 109:1064, 2001.

[149] Tetsuro Kitahara, Masataka Goto, Hiroshi G Okuno. Musical instrument identification based on f0-dependent multivariate normal distribution. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Volume 5, pages V–421. IEEE, 2003.

[150] Tetsuro Kitahara, Masataka Goto, Hiroshi G Okuno. Pitch-dependent identification of musical instrument sounds. *Applied Intelligence*, 23(3):267–275, 2005.

[151] Jana Eggink, Guy J Brown. A missing feature approach to instrument identification in polyphonic music. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Volume 5, pages V–553. IEEE, 2003.

[152] Geoffroy Peeters, Xavier Rodet. Hierarchical Gaussian tree with inertia ratio maximization for the classification of large musical instruments databases. *International Conference on Digital Audio Effects*, 2003.

[153] A. Eronen. Musical Instrument Recognition Using ICA-based Transform of Features and Discriminatively Trained HMMs. *International Symposium on Signal Processing and Its Applications*, Volume 2, pages 133–136. IEEE, 2003.

[154] Matthias Eichner, Matthias Wolff, Rüdiger Hoffmann. Instrument classification using hidden markov models. *International Society for Music Information Retrieval Conference (ISMIR)*, Volume 1, pages 349–350, 2006.

[155] Athanasia Zlatintsi, Petros Maragos. Musical instruments signal analysis and recognition using fractal features. *European Signal Processing Conference (EUSIPCO)*, Volume 11, 2011.

[156] Ali Taylan Cemgil, Fikret Gürgen. Classification of musical instrument sounds using neural networks. *Proc. of SIU97*, 1997.

[157] Roisin Loughran, Jacqueline Walker, Michael O'Neill, Marion O'Farrell. Musical instrument identification using principal component analysis and multilayered perceptrons. *International Conference on Audio, Language and Image Processing (ICALIP)*, pages 643–648. IEEE, 2008.

[158] Róisín Loughran, Jacqueline Walker, Michael ONeill, Marion OFarrell. The use of mel-frequency cepstral coefficients in musical instrument identification. *International Computer Music Conference (ICMC)*, 2008.

[159] Roisin Loughran, Jacqueline Walker, Marion O'Farrell, Michael O'Neill. Comparison of features in musical instrument identification using artificial neural networks. *International Symposium on Computer Music Multidisciplinary Research (CMMR)*. Springer, 2008.

[160] Bozena Kostek, B Kostek. *Soft computing in acoustics*. Springer, 1999.

[161] Bozena Kostek, Andrzej Czyzewski. Representing musical instrument sounds for their automatic classification. *Journal of the Audio Engineering Society*, 49(9):768–785, 2001.

[162] Bozena Kostek, Pawel Zwan. Wavelet-based automatic recognition of musical instruments. *The Journal of the Acoustical Society of America*, 110(5):2754–2754, 2001.

[163] Bozena Kostek. Musical instrument classification and duet analysis employing music information retrieval techniques. *Proceedings of the IEEE*, 92(4):712–729, 2004.

[164] DG Bhalke, CB Rao, DS Bormane. Musical instrument classification using higher order spectra. *Signal Processing and Integrated Networks (SPIN)*, pages 40–45. IEEE, 2014.

[165] Mingsian R Bai, Meng-chun Chen. Intelligent preprocessing and classification of audio signals. *Journal of the Audio Engineering Society*, 55(5):372–384, 2007.

[166] Bernhard Feiten, Stefan Günzel. Automatic indexing of a sound database using self-organizing neural nets. *Computer Music Journal*, 18(3):53–65, 1994.

[167] Petri Toiviainen, Mauri Kaipainen, Jukka Louhivuori. Musical timbre: Similarity ratings correlate with computational feature space distances. *Journal of New Music Research*, 24(3):282–298, 1995.

[168] Petri Toiviainen. Optimizing auditory images and distance metrics for self-organizing timbre maps. *Journal of New Music Research*, 25(1):1–30, 1996.

[169] Petri Toiviainen, Mari Tervaniemi, Jukka Louhivuori, Marieke Saher, Minna Huotilainen, Risto Näätänen. Timbre similarity: Convergence of neural, behavioral, and computational approaches. *Music Perception*, pages 223–241, 1998.

[170] DK Fragoulis, JN Avaritsiotis, CN Papaodysseus. Timbre recognition of single notes using an artmap neural network. *IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, Volume 2, pages 1009–1012. IEEE, 1999.

[171] Bozena Kostek. Soft computing-based recognition of musical sounds. *Rough Sets in Knowledge Discovery 2*, pages 193–213. Springer, 1998.

[172] Alicja Wieczorkowska. Rough sets as a tool for audio signal classification. *Foundations of Intelligent Systems*, pages 367–375. Springer, 1999.

[173] Alicja A Wieczorkowska, Andrzej Czyżewski. Rough set based automatic classification of musical instrument sounds. *Electronic Notes in Theoretical Computer Science*, 82(4):298–309, 2003.

[174] Wojciech Siedlecki, Jack Sklansky. A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters*, 10(5):335–347, 1989.

[175] Róisın Loughran, Jacqueline Walker, Michael O'Neill. An exploration of genetic algorithms for efficient musical instrument identification. *Signals and Systems Conference*, pages 1–6, 2009.

[176] Róisín Loughran, Jacqueline Walker, Michael ONeill, James McDermott. Genetic programming for musical sound analysis. *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, pages 176–186. Springer, 2012.

[177] Slim Essid, Gael Richard, Bertrand David. Instrument recognition in polyphonic music. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Volume 3, pages iii–245. IEEE, 2005.

[178] Peter Somerville, Alexandra L Uitdenbogerd. Multitimbral musical instrument classification. *International Symposium on Computer Science and its Applications (CSA)*, pages 269–274. IEEE, 2008.

[179] David Little, Bryan Pardo. Learning musical instruments from mixtures of audio with weak labels. *International Society for Music Information Retrieval Conference (ISMIR)*, pages 127–132, 2008.

[180] Tetsuro Kitahara, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, Hiroshi G Okuno. Instrument identification in polyphonic music: Feature weighting with mixed sounds, pitch-dependent timbre modeling, and use of musical context. *International Society for Music Information Retrieval Conference (ISMIR)*, pages 558–563, 2005.

[181] Tetsuro Kitahara, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, Hiroshi G Okuno. Instrument identification in polyphonic music: Feature weighting to minimize influence of sound overlaps. *European Journal on Applied Signal Processing (EURASIP)*, 2007(1):155–155, 2007.

[182] Tong Zhang. Instrument classification in polyphonic music based on timbre analysis. *International Symposium on the Convergence of IT and Communications (ITCom)*, pages 136–147. International Society for Optics and Photonics, 2001.

[183] Marek Dziubinski, Piotr Dalka, Bozena Kostek. Estimation of musical sound separation algorithm effectiveness employing neural networks. *Journal of Intelligent Information Systems*, 24(2-3):133–157, 2005.

[184] Philippe Hamel, Sean Wood, Douglas Eck. Automatic identification of instrument classes in polyphonic and poly-instrument audio. *International Society for Music Information Retrieval Conference (ISMIR)*, pages 399–404. Citeseer, 2009.

[185] Philippe Hamel, Douglas Eck. Learning features from music audio with deep belief networks. *International Society for Music Information Retrieval Conference (ISMIR)*, pages 339–344. Utrecht, The Netherlands, 2010.

[186] Jouni Paulus, Anssi Klapuri. Drum sound detection in polyphonic music with hidden Markov models. *European Journal on Audio, Speech, and Music Processing (EURASIP)*, 2009:14, 2009.

[187] Tetsuro Kitahara, K Komatani, T Ogata, HG Okuno, M Goto. Instrogram: A new musical instrument recognition technique without using onset detection nor f0 estimation. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Volume 5, pages V–V. IEEE, 2006.

[188] Tetsuro Kitahara, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, Hiroshi G Okuno. Musical instrument recognizer" instrogram" and its application to music retrieval based on instrumentation similarity. *IEEE International Symposium on Multimedia (ISM)*, pages 265–274. IEEE, 2006.

[189] Tetsuro Kitahara, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, Hiroshi G Okuno. Instrogram: probabilistic representation of instrument existence for polyphonic music. *Information and Media Technologies*, 2(1):279–291, 2007.

[190] Wenxin Jiang, Amanda Cohen, Zbigniew W Raś. Polyphonic music information retrieval based on multi-label cascade classification system. *Advances in Information and Intelligent Systems*, pages 117–137. Springer, 2009.

[191] Wenxin Jiang, Alicja Wieczorkowska, Zbigniew W Raś. Music instrument estimation in polyphonic sound based on short-term spectrum match. *Foundations of Computational Intelligence Volume 2*, pages 259–273. Springer, 2009.

[192] Juan J Bosch, Jordi Janer, Ferdinand Fuhrmann, Perfecto Herrera. A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals. *International Society for Music Information Retrieval Conference (ISMIR)*, pages 559–564, 2012.

[193] Zhiyao Duan, Bryan Pardo, Laurent Daudet. A novel cepstral representation for timbre modeling of sound sources in polyphonic mixtures. *IEEE Conference of Acoustics, Speech and Signal Processing (ICASSP)*, pages 7495–7499. IEEE, 2014.

[194] Eleftherios Spyromitros Xioufis, Grigorios Tsoumakas, Ioannis Vlahavas. Multi-label learning approaches for music instrument recognition. *Foundations of Intelligent Systems*, pages 734–743. Springer, 2011.

[195] Miron B Kursa, Alicja A Wieczorkowska. Multi-label ferns for efficient recognition of musical instruments in recordings. *Foundations of Intelligent Systems*, pages 214–223. Springer, 2014.

[196] Igor Vatolkin, Mike Preuß, Günter Rudolph, Markus Eichhoff, Claus Weihs. Multi-objective evolutionary feature selection for instrument recognition in polyphonic audio mixtures. *Soft Computing*, 16(12):2027–2047, 2012.

[197] Igor Vatolkin, Bernd Bischl, Günter Rudolph, Claus Weihs. Statistical comparison of classifiers for multi-objective feature selection in instrument recognition. *Data Analysis, Machine Learning and Knowledge Discovery*, pages 171–178. Springer, 2014.

[198] Igor Vatolkin, Anil Nagathil, Wolfgang Theimer, Rainer Martin. Performance of specific vs. generic feature sets in polyphonic music instrument recognition. *Evolutionary Multi-Criterion Optimization*, pages 587–599. Springer, 2013.

[199] Kunio Kashino, Kazuhiro Nakadai, Tomoyoshi Kinoshita, Hidehiko Tanaka. Application of the bayesian probability network to music scene analysis. *Computational Auditory Scene Analysis*, pages 115–137. L. Erlbaum Associates Inc., 1998.

[200] Kunio Kashino, Hiroshi Murase. Music recognition using note transition context. *IEEE International Conference on Acoustics, Speech and Signal Processing*, Volume 6, pages 3593–3596. IEEE, 1998.

[201] Kunio Kashino, Hiroshi Murase. A sound source identification system for ensemble music based on template adaptation and music stream extraction. *Speech Communication*, 27(3):337–349, 1999.

[202] Tomoyoshi Kinoshita, Shuichi Sakai, Hidehiko Tanaka. Musical sound source identification based on frequency component adaptation. *International Joint Conferences on Artificial Intelligence Computational Auditory Scene Analysis Workshop*, pages 18–24, 1999.

[203] Pierre Leveau, Emmanuel Vincent, Gaël Richard, Laurent Daudet, i in. Mid-level sparse representations for timbre identification: design of an instrument-specific harmonic dictionary. *Workshop on Learning the Semantics of Audio Signals (LSAS)*, 2006.

[204] Pierre Leveau, Emmanuel Vincent, Gaël Richard, Laurent Daudet. Instrument-specific harmonic atoms for mid-level music representation. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(1):116–128, 2008.

[205] Pierre Leveau, David Sodoyer, Laurent Daudet. Automatic instrument recognition in a polyphonic mixture using sparse representations. *International Society for Music Information Retrieval Conference (ISMIR)*, pages 233–236. Citeseer, 2007.

[206] Bhiksha Raj, Richard M Stern. Missing-feature approaches in speech recognition. *Signal Processing Magazine*, 22(5):101–116, 2005.

[207] Jon P Barker, Martin P Cooke, Daniel PW Ellis. Decoding speech in the presence of other sources. *Speech Communication*, 45(1):5–25, 2005.

[208] Jon Barker. *Missing data techniques: Recognition with incomplete spectrograms.* New York, NY, USA: Wiley, 2012.

[209] Jana Eggink, Guy J Brown. Application of missing feature theory to the recognition of musical instruments in polyphonic audio. *International Society for Music Information Retrieval Conference (ISMIR)*, 2003.

[210] Jana Eggink, Guy J Brown. Instrument recognition in accompanied sonatas and concertos. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Volume 4, pages iv–217. IEEE, 2004.

[211] Zhiyao Duan, Bryan Pardo, Changshui Zhang. Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(8):2121–2133, 2010.

[212] Chunghsin Yeh, Axel Roebel, Xavier Rodet. Multiple fundamental frequency estimation and polyphony inference of polyphonic music signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6):1116–1126, 2010.

[213] Dimitrios Giannoulis, Anssi Klapuri. Musical instrument recognition in polyphonic audio using missing feature approach. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(9):1805–1817, 2013.

[214] Tuomas Virtanen. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(3):1066–1074, 2007.

[215] Toni Heittola, Anssi Klapuri, Tuomas Virtanen. Musical instrument recognition in polyphonic audio using source-filter model for sound separation. *International Society for Music Information Retrieval Conference (ISMIR)*, pages 327–332, 2009.

[216] Francisco J Rodriguez-Serrano, Julio J Carabias-Orti, Pedro Vera-Candeas, Tuomas Virtanen, Nicolas Ruiz-Reyes. Multiple instrument mixtures source separation evaluation using instrument-dependent nmf models. *Latent Variable Analysis and Signal Separation*, pages 380–387. Springer, 2012.

[217] Beiming Wang, Mark D Plumbley. Musical audio stream separation by nonnegative matrix factorization. *Digital Music Research Network Summer Conference*, pages 23–24, 2005.

[218] Marko Helén, Tuomas Virtanen. Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine. *European Signal Processing Conference (EUSIPCO)*, 2005.

[219] Jouni Paulus, Tuomas Virtanen. Drum transcription with non-negative spectrogram factorisation. *European Signal Processing Conference (EUSIPCO)*, page 4, 2005.

[220] Tuomas Virtanen, Ali Taylan Cemgil, Simon Godsill. Bayesian extensions to non-negative matrix factorisation for audio signal modelling. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1825–1828. IEEE, 2008.

[221] Tuomas Virtanen, Annamaria Mesaros, Matti Ryynänen. Combining pitch-based inference and non-negative spectrogram factorization in separating vocals from polyphonic music. *International Symposium on Computer Architecture (ISCA) Tutorial and Research Workshop on Statistical And Perceptual Audition (SAPA)*, 2008.

[222] Bhiksha Raj, Tuomas Virtanen, Sourish Chaudhuri, Rita Singh. Non-negative matrix factorization based compensation of music for automatic speech recognition. *Interspeech Conference*, pages 717–720, 2010.

[223] Eric Gaussier, Cyril Goutte. Relation between plsa and nmf and implications. *International ACM conference on Research and Development in Information Retrieval*, pages 601–602. ACM, 2005.

[224] Graham Grindlay, Daniel PW Ellis. Multi-voice polyphonic music transcription using eigeninstruments. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 53–56. IEEE, 2009.

[225] Graham C Grindlay, Daniel PW Ellis. A probabilistic subspace model for multi-instrument polyphonic transcription. *International Society for Music Information Retrieval Conference (ISMIR)*, pages 21–26, 2010.

[226] Graham Grindlay, Daniel PW Ellis. Transcribing multi-instrument polyphonic music with hierarchical eigeninstruments. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1159–1169, 2011.

[227] Vipul Arora, Laxmidhar Behera. Instrument identification using plca over stretched manifolds. *National Conference on Communications (NCC)*, pages 1–5. IEEE, 2014.

[228] Emmanouil Benetos, Simon Dixon. Multiple-instrument polyphonic music transcription using a convolutive probabilistic model. *Sound and Music Computing Conference*, pages 19–24, 2011.

[229] Emmanouil Benetos, Simon Dixon. Multiple-instrument polyphonic music transcription using a temporally constrained shift-invariant model. *The Journal of the Acoustical Society of America*, 133(3):1727–1741, 2013.

[230] Ying Hu, Guizhong Liu. Instrument identification and pitch estimation in multi-timbre polyphonic musical signals based on probabilistic mixture model decomposition. *Journal of Intelligent Information Systems*, 40(1):141–158, 2013.

[231] Emmanuel Vincent, Xavier Rodet, i in. Instrument identification in solo and ensemble music using independent subspace analysis. *International Society for Music Information Retrieval Conference (ISMIR)*, pages 576–581, 2004.

[232] Luis Gustavo Martins, Juan José Burred, George Tzanetakis, Mathieu Lagrange. Polyphonic instrument recognition using spectral clustering. *International Society for Music Information Retrieval Conference (ISMIR)*, pages 213–218, 2007.

[233] Jun Wu, Yu Kitano, Stanislaw Andrzej Raczynski, Shigeki Miyabe, Takuya Nishimoto, Nobutaka Ono, Shigeki Sagayama. Musical instrument identification based on harmonic temporal timbre features. *Statistical And Perceptual Audition Workshop at the Interspeech Conference*, pages 7–12, 2010.

[234] Juan José Burred, Axel Röbel, Xavier Rodet. An accurate timbre model for musical instruments and its application to classification. *Workshop on Learning the Semantics of Audio Signals*, 2006.

[235] Juan José Burred, Axel Robel, Thomas Sikora. Polyphonic musical instrument recognition based on a dynamic model of the spectral envelope. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 173–176. IEEE, 2009.

[236] Juan José Burred, Axel Robel, Thomas Sikora. Dynamic spectral envelope modeling for timbre analysis of musical instrument sounds. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):663–674, 2010.

[237] East West Quantum Leap. Symphonic orchestra, 2015. Available at `http://www.soundsonline.com/Symphonic-Orchestra`.

[238] Native Instruments. Kontakt virtual studio technology, 2012. Available at `http://www.native-instruments.com/en/products/komplete/synths-samplers/kontakt-5/`.

[239] Source Forge. jvstwrapper, 2012. Available at `http://jvstwrapper.sourceforge.net/`.

[240] Chris Vaill. normalize, 2012. Available at `http://normalize.nongnu.org/`.

[241] Lawrence Fritts. The University of Iowa Electronic Music Studios musical instrument samples. *Available at `http://theremin.music.uiowa.edu/MIS.html`*, 1997.

[242] Source Forge. Sox, the swiss army knife of sound processing programs, 2012. Available at `http://sox.sourceforge.net//`.

[243] Usama M. Fayyad, Keki B. Irani. On the handling of continuous-valued attributes in decision tree generation. *Machine Learning*, 8(1):87–102, 1992.

[244] Usama Fayyad, Keki Irani. Multi-interval discretization of continuous-valued attributes for classification learning. *International Joint Conference on Artificial Intelligence*, pages 1022–1027, 1993.

[245] Patrick J Donnelly, John W Sheppard. Classification of musical timbre using bayesian networks. *Computer Music Journal*, 37(4):70–86, 2013.

[246] S.F. Chen, J. Goodman. An empirical study of smoothing techniques for language modeling. *Annual Meeting on Association for Computational Linguistics*, pages 310–318. Association for Computational Linguistics, 1996.

[247] P. Jain, A. Kapoor. Active learning for large multi-class problems. *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 762–769. IEEE, 2009.

[248] E.M. von Hornbostel, C. Sachs. *Systematik der Musikinstrumente*. Behrend, 1914.

[249] Jun Wu, Emmanuel Vincent, Stanislaw Raczynski, Takuya Nishimoto, Nobutaka Ono, Shigeki Sagayama. Polyphonic pitch estimation and instrument identification by joint modeling of sustained and attack sounds. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1124–1132, 2011.

[250] Stephen McAdams, Suzanne Winsberg, Sophie Donnadieu, Geert De Soete, Jochen Krimphoff. Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes. *Psychological Research*, 58(3):177–192, 1995.

[251] F Opolko, J Wapnick. Mcgill university master samples (MUMS). 11 cd-rom set. *Faculty of Music, McGill University, Montreal, Canada*, 1989.

[252] Tuomas Eerola, Rafael Ferrer. Instrument library (MUMS) revised. *Music Perception*, 25:253255, 2008.

[253] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, Ryuichi Oka. RWC music database: Music genre database and musical instrument sound database. *International Society of Music Information Retrieval (ISMIR)*, Volume 3, pages 229–230, 2003.

[254] Masataka et al. Goto. Development of the RWC music database. *International Congress on Acoustics (ICA)*, Volume 1, pages 553–556, 2004.

[255] Philharmonic Orchestra. Sound sample collection. *Available at* *http://www.philharmonia.co.uk/explore/make_music*.

[256] Yiming Yang, Jan O Pedersen. A comparative study on feature selection in text categorization. *International Conference on Machine Learning (ICML)*, Volume 97, pages 412–420, 1997.

[257] Mark R Every, John E Szymanski. Separation of synchronous pitched notes by spectral filtering of harmonics. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1845–1856, 2006.

[258] David W Aha, Dennis Kibler, Marc K Albert. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, 1991.

APPENDIX A

SIGNATURE LEARNING EXAMPLE

This appendix provides a detailed walk-through of the signature learning process described in Chapter 6.

A.1 Dataset Pre-Processing

First, we begin with an overview of the dataset processing described in Section 5.3. Consider a single instrument, the Violin, taken from the RWC dataset. The original raw dataset consists of performers playing through a chromatic scale, such as the Violin playing at *forte* dynamic level shown in Figure A.1a. Many of the datasets, such as RWC, contain two to three different performers on different instruments playing at multiple levels and with varying articulations. Using the silence in between notes as a cue, we split these files into individual files each containing a single note as described in Section 5.3.1.

Next, consider a sound example of a Violin playing a single note, a middle C (261 Hz) at a *forte* dynamic level. The waveform of this note is shown in Figure A.1b. Each file is then truncated to one second in length, as described in Section 5.3.2. Across the datasets and instruments, many of the sound examples are shorter than one second and silence is added to make them one second each. Others are longer than one second and must be sampled, and a brief fade out is added at the end to eliminate any discontinuities in the waveform, such as the Violin note shown in Figure A.1c. Lastly, the sound files are batch normalized by each instrument for each dataset. This means, for a specific instrument and dataset, the file containing the loudest peak is scaled to the maximum gain of 1.0. All other files are also scaled by this amount. This helps scale the significant data above the noise floor, while preserving the relative dynamic differences between samples for the instrument.

This entire process is repeated over the 13 instruments for all four datasets. These datasets are used to generate datasets for training our binary models (Section 5.5) and used to generate the datasets of polyphonic mixtures (Section 5.6).

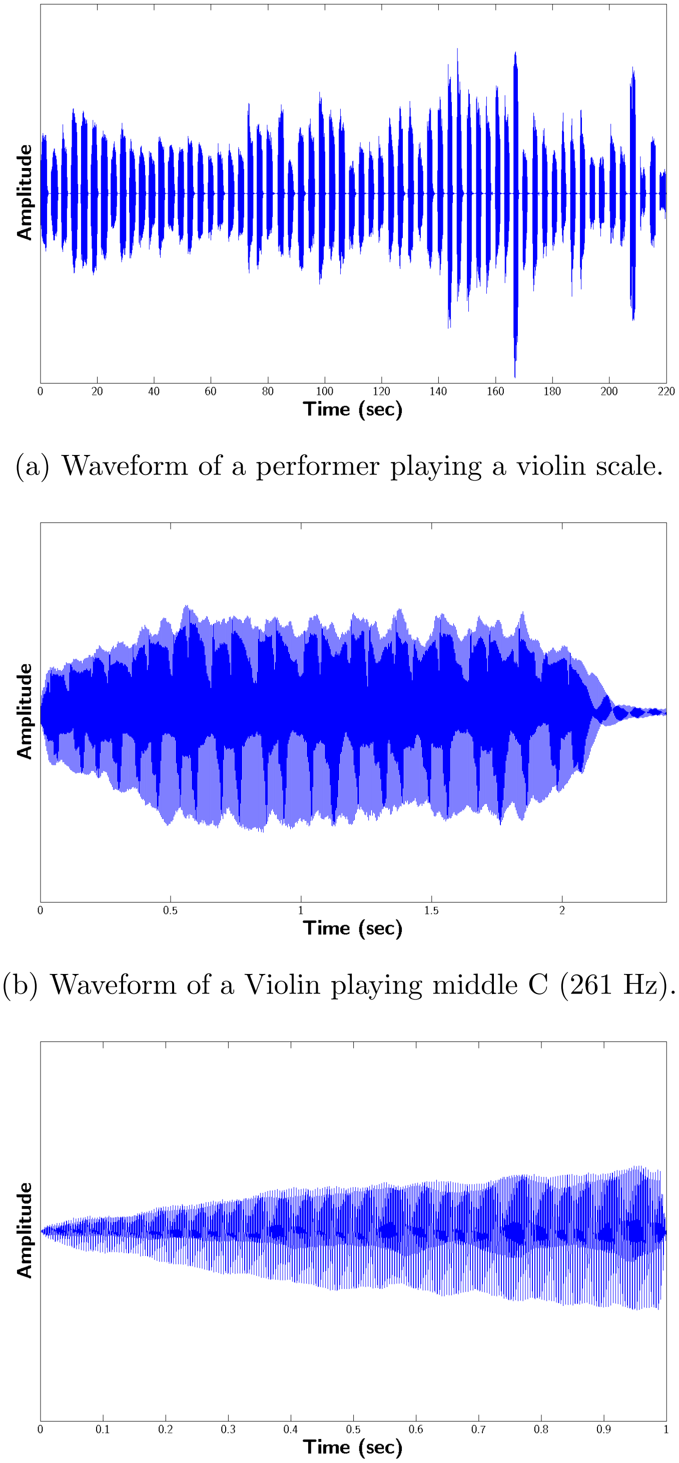
A.2 Signal Processing

In this work, we are interested in spectral features and must transform each waveform from the time domain to the frequency domain. For each example of a single note, we take the fast Fourier Transform of the signal, as described in Section 5.4. Figure A.2a graphs the amplitudes of the spectra along a linear scale. In order to consider the relative significance of peaks to their local frequency neighborhood, we consider logarithmic power spectral density, scaling each by $10 \cdot \log 10$ as described in Section 5.4.2 and shown in Figure A.2b.

The next step is to determine the variable-frequency noise threshold as described in Section 6.3.3. An example is shown in Figure A.2c. We consider any peak above this threshold to be a significant peak. Among those significant peaks, we identify the fundamental frequency f_0 using the procedure described in Section 6.3.2. Our algorithm selects the lowest significant peak, the leftmost peak shown in Figure A.2b.

A.3 Spectral Peak Extraction

After identifying the significant peak threshold, we extract all the locations (in Hertz) corresponding to significant peaks. In this stage, we are interested in learning the locations of the significant peaks, not the specific amplitude values. Using the calculated f_0 value, we calculate the ratio of the peak's frequency to the fundamental. Table A.1 shows the significant peaks for the 1st, 2nd, and 3rd overtone of a Violin

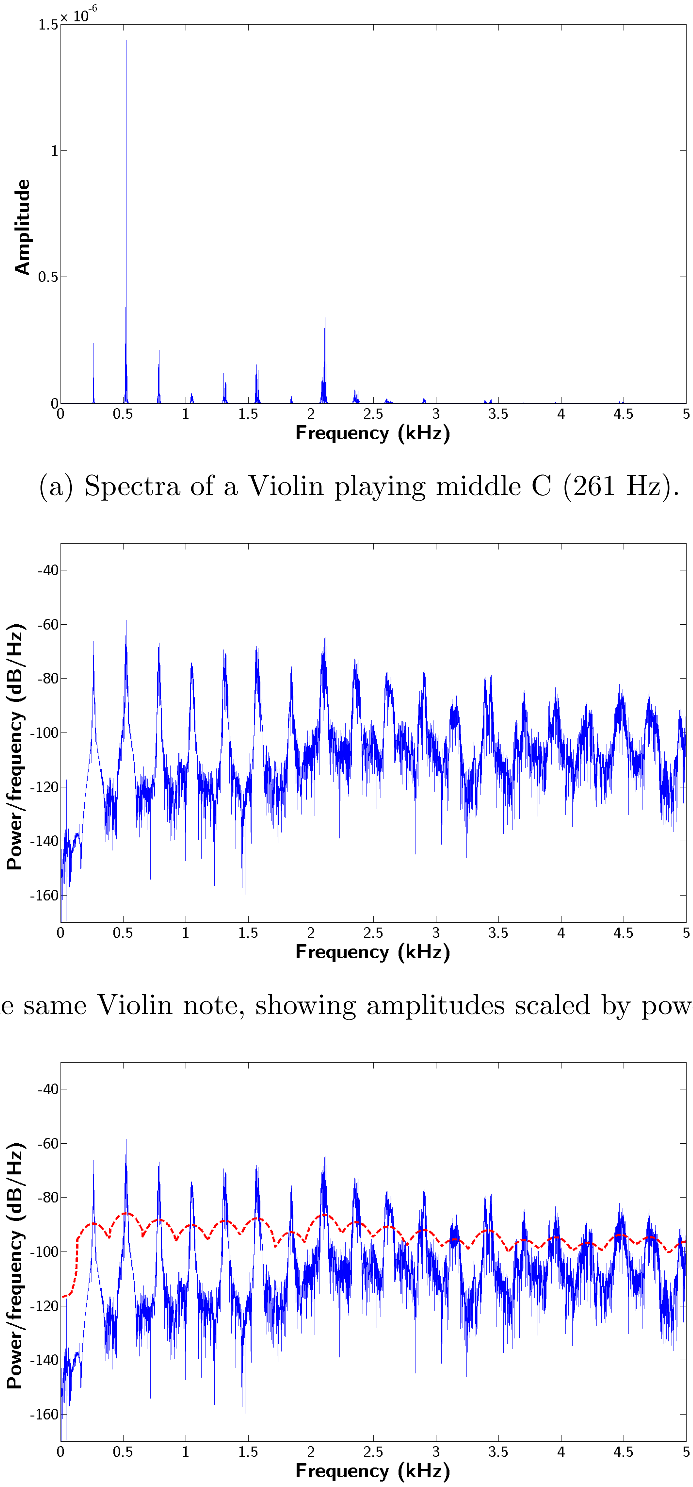


(a) Waveform of a performer playing a violin scale.

(b) Waveform of a Violin playing middle C (261 Hz).

(c) Waveform of a Violin playing middle C (261 Hz), truncated at one second.

Figure A.1: Waveforms of Violin notes



(a) Spectra of a Violin playing middle C (261 Hz).

(b) Spectra of the same Violin note, showing amplitudes scaled by power spectral density.

(c) Spectra of the same Violin note, showing the variable-frequency threshold (dotted).

Figure A.2: Spectra of Violin notes

Table A.1: Examples of significant peaks of a Violin note with $f_0 = 262.0$ Hz

frequency	**amplitude**	**ratio**
502	-94.89	1.9160
507	-86.50	1.9388
513	-74.79	1.9618
516	-79.47	1.9732
519	-64.20	1.9847
525	-58.43	2.0076
531	-68.18	2.0306
537	-78.70	2.0535
543	-88.83	2.0765
548	-96.69	2.0956
552	-96.77	2.1109
766	-100.39	2.9237
770	-92.44	2.9389
775	-77.61	2.9580
777	-80.94	2.9656
781	-68.51	2.9809
787	-66.78	3.0038
790	-72.55	3.0153
792	-74.19	3.0229
794	-74.68	3.0305
798	-81.81	3.0458
801	-81.53	3.0573
1026	-97.88	3.9160
1032	-90.65	3.9389
1039	-80.85	3.9656
1043	-74.80	3.9809
1045	-74.27	3.9885
1049	-74.13	4.0038
1052	-75.63	4.0153
1055	-75.97	4.0267
1058	-75.46	4.0382
1061	-81.42	4.0496
1064	-80.21	4.0611

. . . *etc.*

note playing middle C ($f_0 = 261.5$ Hz). The example shows significant peaks centering around an integer ratios 2, 3, and 4. Since we use a single one-second time window in our FFT, we obtain a high frequency resolution and capture the frequency fluctuation over the course of the one second sample. This produces additional spectral energy around each spectral peak, seen in Figure A.2b as the width of each peak, or as the additional values in Table A.1. These values will contribute towards the standard deviation of the signature clusters. String instruments, such as the Violin, typically have more frequency fluctuations than other instruments, because of the sustained bowing over the strings.

We repeat this procedure for all other Violin sound files in the dataset, such as the simplified examples shown in Table A.2. We flatten all these values into a single one-dimensional vector, shown in Table A.3. At this stage, we do not use any amplitude information but only the ratio values. The energy of the peaks are used in the feature extraction stage of the classification experiments. For now, we are concerned with learning where to look for significant spectral energy.

A.4 Clustering Significant Peaks

Next we apply k-means clustering on the set of ratio values as described in Section 6.3.4. We then extract the resulting clusters as the signature for the Violin. Each cluster returns a mean μ and standard deviation σ, which we use to specify a window centered on the ratio ± one standard deviation. A larger standard deviation indicates more fluctuation in frequency over the duration of the sound file.

Table A.4 shows some of the clusters learned for the Violin from the RWC dataset. Notice that while many of the means are near integer values, they are not an exact integer. This deviation varies between instruments and is useful information to cap-

Table A.2: A sampling of the ratios extracted from seven different Violin notes.

1.99, 2.00, 2.01, 2.02, 2.98, 2.99, 3.00, 3.01, 3.02, 3.03, 3.97, 3.98,
4.00, 4.01, 4.95, 4.97, 5.00, 5.01, 5.02, 5.03, . . .

1.989, 2.000, 2.011, 2.019, 2.981, 2.992, 3.000, 3.011, 3.019, 3.034, 3.969, 3.981, 3.996,
4.011, 4.950, 4.969, 4.996, 5.011, 5.023, 5.031, 5.038, . . .

1.966, 1.985, 2.004, 2.019, 2.038, 2.966, 2.985, 3.004, 3.019, 3.038, 3.951, 3.970, 3.985,
4.000, 4.011, 4.019, 4.038, 4.966, 4.985, 5.000, 5.008, 5.019, 5.038, 5.045, . . .

1.970, 1.981, 2.004, 2.023, 2.981, 3.004, 3.011, 3.023, 3.981, 3.992,
4.001, 4.009, 4.985, 4.992, 5.004, 5.015, 5.023, 5.042, . . .

1.950, 1.969, 1.981, 1.992, 2.000, 2.015, 2.034, 2.950, 2.969, 2.985, 3.000, 3.019, 3.034,
4.000, 4.019, 4.038, 4.046, 4.969, 4.985, 5.004, 5.019, . . .

1.981, 2.004, 2.027, 2.958, 2.966, 2.981, 3.004, 3.015, 3.023, 3.031, 3.046, 3.981, 3.989,
3.954, 3.969, 3.985, 4.004, 4.015, 4.027, 4.981, 4.989, 4.996, 5.004, 5.015, 5.027, . . .

1.958, 1.981, 2.004, 2.011, 2.026, 2.034, 2.045, 2.966, 2.985, 3.004, 3.011, 3.026, 3.038,
3.958, 3.966, 3.977, 3.985, 4.000, 4.011, 4.019, 4.030, 4.042, 4.981, 4.992, 5.008, . . .

Table A.3: One dimensional vector of the ratios extracted from Figure A.2.

1.950, 1.958, 1.966, 1.969, 1.970, 1.981, 1.981, 1.981, 1.981, 1.985, 1.989, 1.992,
2.000, 2.000, 2.004, 2.004, 2.004, 2.004, 2.011, 2.011, 2.015, 2.019, 2.019, 2.023,
2.026, 2.027, 2.034, 2.034, 2.038, 2.045, 2.046, 2.950, 2.958, 2.966, 2.966, 2.966,
2.969, 2.981, 2.981, 2.981, 2.985, 2.985, 2.985, 2.992, 3.000, 3.000, 3.004, 3.004,
3.004, 3.004, 3.011, 3.011, 3.011, 3.015, 3.019, 3.019, 3.019, 3.023, 3.023, 3.026,
3.031, 3.034, 3.034, 3.038, 3.038, 3.046, 3.049, 3.951, 3.954, 3.958, 3.966, 3.969,
3.969, 3.970, 3.977, 3.981, 3.981, 3.981, 3.985, 3.985, 3.985, 3.989, 3.992, 3.996,
4.000, 4.000, 4.000, 4.004, 4.004, 4.011, 4.011, 4.011, 4.011, 4.015, 4.019, 4.019,
4.019, 4.019, 4.027, 4.027, 4.030, 4.034, 4.038, 4.038, 4.042, 4.049, 4.950, 4.966,
4.966, 4.969, 4.981, 4.981, 4.985, 4.985, 4.985, 4.989, 4.992, 4.992, 4.996, 4.996,
5.000, 5.004, 5.004, 5.004, 5.008, 5.008, 5.011, 5.015, 5.015, 5.019, 5.019, 5.023,
5.023, 5.027, 5.031, 5.038, 5.038, 5.042, 5.045, . . .

Table A.4: Example clusters learned for the Violin

$$\begin{aligned}
\mu = \{&2.027,\ 3.029,\ 4.029,\ 5.030,\ 6.028,\ 7.014,\ 8.017,\ 9.011,\ 10.008,\\
&10.999,\ 11.843,\ 12.014,\ 12.210,\ 12.917,\ 13.083,\ 13.822,\ 14.047,\ \ldots\}\\
\sigma = \{&0.056,\ 0.060,\ 0.075,\ 0.084,\ 0.093,\ 0.090,\ 0.100,\ 0.096,\ 0.105,\\
&0.099,\ 0.099,\ 0.046,\ 0.111,\ 0.074,\ 0.073,\ 0.117,\ 0.081,\ \ldots\}
\end{aligned}$$

ture. Also notice that not all of the means are quasi-integers. For example, notice the clusters centered around the ratios 11.843, 12.014, and 12.210. Because our implementation of k-means creates additional clusters by splitting clusters with a wide standard deviation at each iteration, this signature learned three clusters centered near the ratio of 12, instead of one cluster with a large standard deviation. This strategy of having multiple smaller clusters has a benefit over a single cluster with a wide standard deviation in that if there is source interference near this location, then potentially only one of the three features extracted will contain the interference.

We repeat this procedure for every instrument and for each of the datasets. We learn a unique spectral signature for each instrument and each dataset.