POLYPHONIC MUSIC INFORMATION RETRIEVAL BASED ON MULTI-LABEL CASCADE CLASSIFICATION SYSTEM

by

Wenxin Jiang

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Information Technology

Charlotte

2009

Approved by:

_____
Dr. Zbigniew W. Ras

_____
Dr. Tiffany Barnes

_____
Dr. Xintao Wu

_____
Dr. Alicja Wieczorkowska

_____
Dr. Yuri Godin

ABSTRACT

WENXIN JIANG. Polyphonic music information retrieval based on multi-label cascade classification system. (Under the direction of DR. ZBIGNIEW W. RAS)

Recognition and separation of sounds played by various instruments is very useful in labeling audio files with semantic information. This is a non-trivial task requiring sound analysis, but the results can aid automatic indexing and browsing music data when searching for melodies played by user specified instruments. Melody match based on pitch detection technology has drawn much attention and a lot of MIR systems have been developed to fulfill this task. However, musical instrument recognition remains an unsolved problem in the domain. Numerous approaches on acoustic feature extraction have already been proposed for timbre recognition. Unfortunately, none of those monophonic timbre estimation algorithms can be successfully applied to polyphonic sounds, which are the more usual cases in the real music world. This has stimulated the research on multi-labeled instrument classification and new features development for content-based automatic music information retrieval. The original audio signals are the large volume of unstructured sequential values, which are not suitable for traditional data mining algorithms; while the acoustical features are sometime not sufficient for instrument recognition in polyphonic sounds because they are higher-level representatives of raw signal lacking details of original information. In order to capture the patterns which evolve on the time scale, new temporal features are introduced to supply more temporal information for the timbre recognition. We will introduce the multi-labeled classification system to estimate multiple timbre information from the polyphonic sound by classification based on acoustic features and short-term power

spectrum matching. In order to achieve higher estimation rate, we introduced the hierarchically structured cascade classification system under the inspiration of the human perceptual process. This cascade classification system makes a first estimate on the higher level decision attribute, which stands for the musical instrument family. Then, the further estimation is done within that specific family range. Experiments showed better performance of a hierarchical system than the traditional flat classification method which directly estimates the instrument without higher level of family information analysis.

Traditional hierarchical structures were constructed in human semantics, which are meaningful from human perspective but not appropriate for the cascade system. We introduce the new hierarchical instrument schema according to the clustering results of the acoustic features. This new schema better describes the similarity among different instruments or among different playing techniques of the same instrument. The classification results show the higher accuracy of cascade system with the new schema compared to the traditional schemas. The query answering system is built based on the cascade classifier.

ACKNOWLEDGMENTS

I am grateful for the support that I have received from the Computer Science Department at the University of North Carolina at Charlotte.

I would like to thank my dissertation supervisor Zbigniew W. Ras for bringing me into his unique research team (Music Information Retrieval Group at the KDD Laboratory) and his inimitable help and advice. I could not have dreamed of a place of more intellectual colleagues, and with a more innovative environment that is encouraging new ideas.

I am also grateful to my other committee members, Alicja Wieczorkowska, Tiffany Barnes, and Xintao Wu for their valuable feedback. They provided the insightful comments and suggestions for my research work.

I thank Cynthia Zhang and Rory Lewis for sharing their knowledge and skills in MIR area and preparing the initial monophonic audio training database which is used in my research.

I also thank Amanda Cohen-Mostafavi for contributing her expertise in music domain and her hard work to the construction of the polyphonic musical testing sounds.

This work was supported by the National Science Foundation under grant IIS-0414815.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

ANSI	American National Standards Institute
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
KDD	Knowledge Discovery in Databases
KNN	K Nearest Neighbor
MFCC	Mel Frequency Cepstral Coefficients
MIDI	Musical Instrument Digital Interface
MIR	Music Information Retrieval
MPEG	Moving Picture Experts Group
MUMS	McGill University Master Samples
NFFT	the Next larger integer, which is a power of two
STFT	Short Time Fourier Transform

CHAPTER 1: INTRODUCTION

In recent years, rapid advances in digital music creation, collection and storage technology have enabled organizations to accumulate vast amounts of musical audio data. The booming of multimedia resources from the Internet brought a tremendous need to provide new, more advanced tools for the ability to query and process vast quantities of musical data, since searching through multimedia data is a highly nontrivial task requiring content based indexing of the data, which are not easy to describe with mere symbols. Many multimedia resources provide data which are manually labeled with some description information, such as title, author, company, and so on. However, in most cases those labels are insufficient for content-based searching. Timbre recognition, one of the main subtasks in Music Information Retrieval, has proven to be extremely challenging especially in multi-timbre sounds, where multiple instruments are playing at the same time.

1.1 Background

Typically, a digital music recording, in the form of a binary file, contains a header and a body. The header stores file information such as length, number of channels, sampling rate, etc. Unless being manually labeled, a digital audio recording has no description on timbre or other perceptual properties. Also, it is a very difficult task to label those perceptual properties for every piece of musical object based on its datacontent. The body

of a digital audio recording contains an enormous amount of integers in a time-order sequence. For example, at a sampling rate of 44,100Hz, a digital recording has 44,100 integers per second, which means, in a one-minute long digital recording, the total number of the integers in the time-order sequence will be 2,646,000, which makes it a very big data item.

Table 1.1 Comparison of audio data and traditional data

Data source	organization	volume	Type	Quality
Traditional data	Structured	Modest	Discrete, Categorical	Clean
Audio data	Unstructured	Very large	Continuous, Numeric	Noisy

The difference between the musical data and traditional transaction data is shown in Table 1.1. Being not in a well-structured form with the semantic meaning, musical data is not suitable for most traditional data mining algorithms. Therefore, a number of features have been explored to give a higher-leveled representation of digital musical object with the structured and meaningful attributes based on acoustical expertise. Then, these feature datasets can be intuitively used as system semantics, since they are computational and "known" to the computer system.

1.1 Pitch, melody and rhythm

Pitch is the perceived quality of a sound that is chiefly a function of its fundamental frequency. In general pitch is regarded as becoming higher with increasing frequency and lower with decreasing frequency. The difference between two pitches is called an interval; Melodies can be considered sets of either pitches or intervals.

There is another facet of music information which is called temporal facet. It is the duration of musical events, including tempo indicators, meter, pitch duration and accents. Those temporal events make up the rhythmic component of a musical work.

In music information retrieval area, research has been conducted in melody or rhythm match based on the pitch identification, which usually involves the fundamental frequency detection. Utrecht University provides an overview of content-based Music Information Retrieval systems [1] and lists around 43 MIR systems, most being the query by whistling/humming systems for melody retrieval. However, no system exists for timbre information retrieval in the literature and commercial software market, which indicates it as the nontrivial or unsolved task.

1.1.1 Timbre

According to the definition of American Standards Association, Timbre is the quality of sound that is not loudness and pitch. It distinguishes different musical instruments playing the same note with the identical pitch and loudness. So it is one of the most important and relevant facet of music information. People discern timbres from speech and music in everyday life.

Musical instruments usually produce the sound waves with the integer multiple frequencies. This frequency series are called harmonics, or harmonic partials. The lowest frequency is the fundamental frequency (f0), which has intimate relation with pitch. The second and higher frequencies are called overtones. Along with fundamental frequency, these harmonic partials mainly decide the timbre, which is also called tone color. The aural distinction between different musical instruments is caused by the differences in timbre.

Attack and decay also contribute to the timbre of sound in some instruments. For example the plucked instruments give a sudden attack characterized by a rapid rise to its peak amplitude. The decay is relatively long and gradual. The ear is sensitive to these attack and decay rates and may be able to use them to identify the instrument producing the sound.

According to the number of timbres in the analyzed signal, music sounds are divided into two groups: monophony and polyphony. **Monophonic** sound means a sound having a single unaccompanied melodic line, which usually has only one instrument sound. **Polyphony** is the music that simultaneously combines two or more independent musical lines (melodies), which is usually multi-timbre sound with two or more instruments playing at the same time. In the real music pieces, polyphonic sounds are more common than monophonic sounds. This dissertation focuses on the timbre estimation in polyphonic sounds.

1.1.2 Sound Data

Generally, identification of musical information can be performed not only for digital audio data (e.g., audio samples taken from real recordings), but also for other representations, such as MIDI data. MIDI files give access to highly structured data, where information about the pitch, effects applied, beginning and end of each note, voices (timbres) used, and about every note that is present in a given time instant is preprogrammed. So, research on MIDI data may basically concentrate on a higher level of musical structure, like key or metrical information. In the case of audio samples, any basic information like pitch (or pitches, if there are more sounds), timbre, beginning and end of the sound must be extracted via digital signal. There are many methods of pitch

extraction, mostly coming from speech processing. But extraction of such a piece of simple information may produce errors and poses some difficulties. In particular, octave errors are common for a singular sound. Pitch extraction for layered sounds is even more difficult, especially when spectra (frequency domain of the signal) overlap. Basically, parameters of fundamental frequency trackers are usually adjusted to characteristics of the instrument that is to be tracked, but this cannot be done if we do not know what instrument is playing.

Identifying the dominant instruments which are playing in the audio segments is even more difficult. Timbre is a rather subjective quality, defined by ANSI as the 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 different. Such definition is subjective and not of much use for automatic sound timbre classification. Therefore, musical sounds must be very carefully parameterized to allow automatic timbre recognition.

1.2 Motivation and Approach

Our research is driven by the desire to identify the individual instrument types or instrument family categories in a music object. Timbre is a quality of sound that distinguishes one musical instrument from another and there are a wide variety of instrument families and individual categories. Therefore, musical sounds must be very carefully parameterized to allow the automatic timbre recognition. The real use of timbre-based grouping of music is discussed in [3]. The author addresses the problem of hearing complex auditory environments and uses a series of analogies to describe the process required of the human auditory system as it analyzes mixtures of sounds to recover

descriptions of individual sounds. This book also establishes a theoretical framework that integrates the previous research in psychoacoustics, speech perception, music theory and composition, and computer modeling. However, the author describes the concept of timbre as not well defined, which makes it very difficult to distinguish timbre among the musical instruments.

Orchestral instruments produce overtones, which results in a sound with a group of frequencies in clear mathematical relationships (so-called harmonics). There are a number of different approaches to detect sound timbre (for instance [2], [5]). Some of them are quite successful on certain simple sound data (monophonic, short, of limited instrument types). Dimensional approach to timbre description was proposed in [3]. Timbre description is basically subjective and vague, and only some subjective features have well defined objective counterparts, like brightness, calculated as gravity center of the spectrum. Explicit formulation of rules of objective specification of timbre in terms of digital descriptors will formally express subjective and informal sound characteristics. It is especially important in the light of human perception of sound timbre. Evolution of sound features in time is essential for a human; therefore, it should be reflected in sound description as well. The discovered temporal patterns may better express sound features than static features, especially that classic features can be very similar for sounds representing the same family or pitch, whereas changeability of features with pitch for the same instrument makes sounds of one instrument dissimilar. Methods in research on automatic musical instrument sound classification go back to approximately 15 years. So far, there is no standard parameterization used as a classification basis. The sound descriptors used are based on various methods of analysis in time, etc. However, current

features fail to describe sufficiently the audio sound pattern, which varies in time within a whole sound segment, especially where multiple audio sources are overlapping with each other. It was widely observed that a sound segment of a note, which is played by a musical instrument, has at least three states: onset (transient), quasi-steady state and offset (transient). Vibration pattern in a transient state is known to significantly differ from the one in a quasi-steady state. Transient includes changes by definition; vibration causes changes during the steady state (the player does not vibrate the sound when the sound is just starting to be articulated, or is already released, the reverberation of the room can be rather present in the sound wave). Consequently, the harmonic features in the transient states are significantly different from those in the quasi-steady state. In spectrum domain, time-frequency domain and cepstrum domain, Fourier Transform for spectral analysis is the most common technique, such as Fast Fourier Transform (FFT), Short-Time Fourier Transform (STFT), Discrete Fourier Transform (DFT), and so on. Based on research performed in the MIR area, MPEG proposed an MPEG7 standard, in which it described a set of low-level sound temporal and spectral features. The low-level descriptors in MPEG7 are intended to describe the time-variant information within an entire audio segment, where most of them are, like other STFT related acoustic features, in a form of either vector or matrix of large size. Therefore, these features are not suitable for traditional classifiers, which require lower dimensionality of the input datasets. Researchers have explored different statistical summations in a form of single value to describe signatures of musical instruments within vectors or matrices in those features, such as tristimulus parameters [26], brightness [10], and spectral irregularity [40].

1.3 Applications of Timbre estimation

The goal of our research is to build a dynamic and flexible system for automatic indexing of music by instruments or alternatively by classes of instruments, use this system to build the database for storing automatically indexed musical files, and implement Flexible Query Answering System to handle user requests. There are several practical areas where the outcomes of this research can be applied.

1.3.1 Music File Annotation

In recent decades, computer technologies boost rapid growing of music repositories through the Internet or in the home PC. However, the musical data are represented as binary streams of integers, while traditional search algorithms are text based. Therefore, musical files are opaque to those content-based search engines. A few years ago, MPEG published MPEG7 to standardize a set of sound descriptors based on latest research in MIR area, which can successfully describe music object. However, all of them fail to describe multi-timbre sounds for queries such as "find the cadenzas of all the Mozart concertos in the database, and sort them by instrument".

1.3.2 Music Transcription

Music transcription (i.e. writing scores for audio) is a very difficult task that can be performed by musicians or intensively trained experts. Since music transcription is a very important tool to musicians, musicologists and music fans, it would be beneficial to have it automatically performed.

1.3.3 Structured-Audio Encoding

Audio files that are structured in FS tree [30] not only make music annotation easier, but also provide quick access to desired parts of the audio. Users can mute part of a piece of music and play or control a playback with much more flexibility.

1.3.4 Instrumental Music Recommendation Engine

Instrumental music recommendation for digital recordings is a very challenging task, which requires musical knowledge about musical instrument classification as well as music objects. For example, a school ensemble may need to practice on similar, yet different music each semester. An instrumental music recommendation system can assist non-musician users to find their favorite music items in large music repositories.

1.4 Contributions of this Dissertation

In this dissertation, we introduce the multi-label classification method which uses the classifier learned from the single-class training samples to classify the polyphonic sound with multiple timbre class labels. This method overcomes the deficiency of the traditional methods based on sound separation algorithm since it preserves the original polyphonic signals during the multi-timbre estimation process.

We developed a novel cascade classification system based on multi-label classification method. In particular, a new machine learned schema is introduced to represent the hierarchical structure of musical instruments. This new schema is built by the clustering analysis and better describes the relationship among 45 different western musical instruments than the conventional schemas.

We also directly use the power spectrum as the low level representation of the raw signal to achieve high recognition rate for polyphonic sounds. Due to the high

dimensionality of the power spectrum and large size of reference database, the multi-resolution approach is used to reduce the computational complexity. We reduce the cost of matching reference spectra by excluding the large number of reference samples after matching the analyzed signal against the highly smoothed spectra which have relatively low dimensions.

The cascade system allows us to further tackle the computational complexities by incorporating both feature-based and spectrum-based pattern recognition approaches. At the higher level of hierarchical tree which contains a large number of reference samples, acoustic features are used to estimate the signal on the instrument family level. When the classification process reaches the bottom level of the tree where reference database is reduced to a relatively small subset, the power spectrum is used to estimate the signal on the instrument level. The experimental results show that the cascade classification system achieves both high efficiency and accuracy for polyphonic timbre estimation.

We developed new temporal acoustic features based on MPEG-7 instantaneous spectral features to improve the discriminating ability of the classifier for some musical instruments that share the similar pattern in spectral space but unique characteristics in short term temporal feature space.

1.5 Organization of this Document

The remainder of this dissertation is structured as follows:

Chapter 2: We discuss previous work pertinent to the problem of timbre estimation and acoustic features mainly designed to perform the single instrument estimation. We also introduce the new temporal features developed by us.

Chapter 3: We introduce polyphonic sound estimation method based on multi-label classification and compare it with multi-class classification approach.

Chapter 4: We introduce the power spectrum matching based multi-label classification approach. The experimental results show that the method based on spectrum matching yields higher recognition rate than feature-based classification algorithm. We also introduce the multi-resolution technique to reduce the computational complexity.

Chapter 5: We introduce the multi-label cascade classification which uses both acoustic features and power spectrum to perform polyphonic sound estimation. By taking advantage of the efficiency of feature-based classification and the high accuracy of power spectrum matching method, we achieve better estimation results than the traditional classification method.

Chapter 6: We develop a novel hierarchical schema built by clustering analysis to further improve the performance of cascade classification system. Before using the clustering algorithm to generate the instrument hierarchical tree, we carefully examine all the available clustering algorithms and distance measurements and perform intensive experiments to evaluate a large number of approaches. From the evaluation results, we choose the agglomerative hierarchical clustering algorithm as the clustering approach, Pearson correlations as the similarity measurement, and Ward as the cluster linkage method.

Chapter 7: We summarize with a discussion of what we have accomplished and plans for future directions.

CHAPTER 2: TIMBRE ESTIMATION BASED ON FEATURES

2.1. Signal processing

2.1.1 Spectrum analysis

Spectrum analysis is the process which converts the time domain to frequency domain. Fast Fourier Transform (FFT) is an algorithm to perform such transformation. As Figure 2.1 shows, the FFT analyzes the signal into its frequency components:

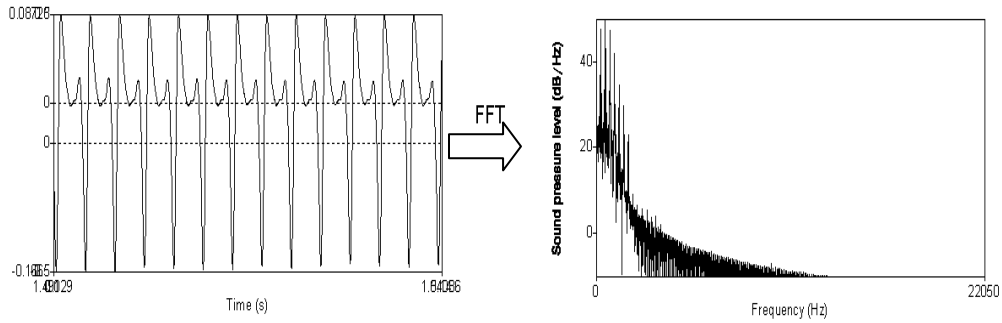

Figure 2.1 Transfer of time domain to frequency domain by FFT

2.1.2 Windowing

Because we measure the signal in a short period, there is no way to know where exactly the periodic signal starts and ends. If the period does not fit the measurement time, meaning not quite an integral number of cycles fit into the total duration of the measurement, the spectrum is not correct. Since we can't assume anything about the signal, we need a way to make any signal's ends smoothly to each other. One way to do

this is to multiply the signal by a “window” function. There are many window functions to be chosen; in our research we use Hamming window for windowing the short time signal. Hamming window is basically the "raised cosine" window (Figure 2.2)

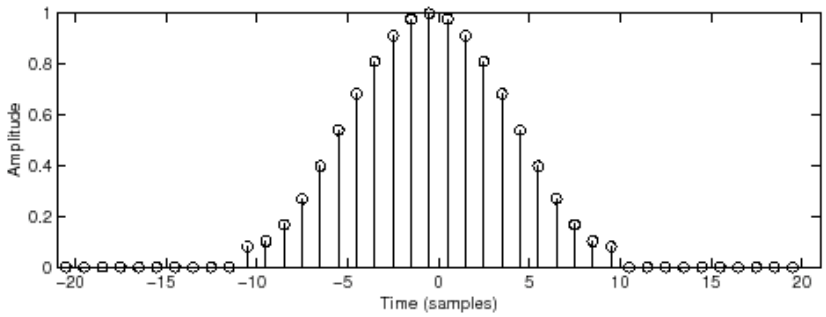

Figure 2.2 Hamming window

Figure 2.3 shows us that windowing process solves the problem of spectrum leakage so that a more clean and accurate spectrum is achieved from the signal.

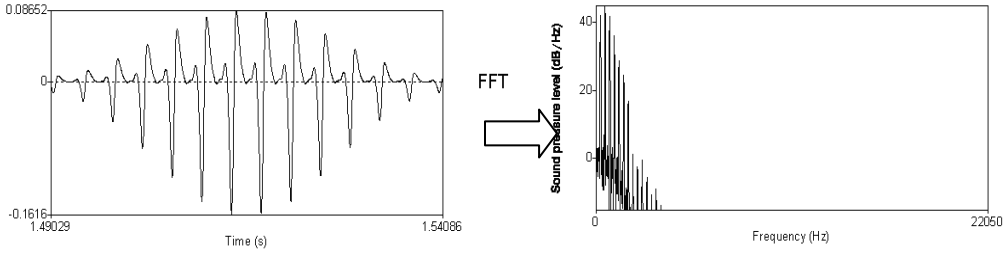

Figure 2.3 FFT with windowing

2.1.3 Overlapping of windows

Since the window diminishes the signal on both edges, it leads to information loss. In order to preserve this information, the consecutive analysis frames have overlap in time. The empirical experiments show the best overlap is two-thirds of window size [42].

2.2 Acoustic Features

The process of feature extraction is usually performed to extract structured data attributes from the temporal or spectral space of the signal. This will reduce the raw data

into a smaller and simplified representation while preserving the important information for timbre estimation. Sets of acoustical features have been successfully developed for timbre estimation in monophonic sounds where mono instruments are playing [14].

2.2.1 MPEG-7 descriptors

Based on latest research in the area, MPEG published a standard group of features for the digital audio content data description [14]. They are either in the frequency domain or in the time domain. For those features in the frequency domain, a STFT with Hamming window has been applied to the sample data, where each frame generates a set of instantaneous values.

Spectrum Centroid describes the center-of-gravity of a log-frequency power spectrum in the following formulas. It economically indicates the predominant frequency range. $P_x(k),\ k = 0,..,\frac{NFFT}{2}$ are power spectrum coefficients. Coefficients below 62.5Hz have been grouped together for fast computation.

$$bound = floor\left(\frac{62.5 \times NFFT}{sr}\right)$$

$$P'_x(0) = \sum_{k=0}^{bound} P_x(k),$$

$$P'_x(n) = P_x(n + bound), f(n) = (n + bound)\frac{sr}{NFFT}, n = 1,..,\frac{NFFT}{2} - bound$$

$$C = \sum_n \log_2(f(n)/1000)P'_x(n) \Big/ \sum_n P'_x(n)$$

where *sr* is the sample rate. A mean value and standard deviation of all frames have been used to describe the Spectrum Centroid of a music object.

Spectrum Spread is the Root of Mean Square value of the deviation of the Log frequency power spectrum with respect to the gravity center in a frame. Like Spectrum Centroid, it is an economic way to describe the shape of the power spectrum.

$$S = \sqrt{\sum_n ((\log_2(f(n)/1000) - C)^2 P_x'(n)) \Big/ \sum_n P_x'(n)}$$

A mean value and standard deviation of all frames have been used to describe the Spectrum Spread of a music object.

Spectrum Flatness describes the flatness property of the power spectrum within a frequency band, which is ranged by the edges function.

$$SFM_b = \frac{\sqrt[ih(b)-il(b)+1]{\prod_{i=il(b)}^{ih(b)} c(i)}}{\frac{1}{ih(b)-il(b)+1}\sum_{i=il(b)}^{ih(b)} c(i)}$$

where c(i) is the mean value of a group of power spectrum coefficients, and the total number of each group is determined by the location of each frequency bin, *ih* and *il* are the boundaries of each bin. The value of each bin is treated as an attribute value in the database. Since the octave resolution in this dissertation is 1/4, the total number of signal bands is 32.

Spectrum Basis Functions are used to reduce the dimensionality by projecting the spectrum from high dimensional space to low dimensional space with compact salient statistical information. These statistical values are maximum, minimum, mean value, and the standard deviation of the matrix, maximum, minimum, mean value of dissimilarity of each column and row, where the dissimilarity is measured by the following equation: $\mathbf{V} = [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_k]$, where V is computed from $\tilde{\chi} = \mathbf{USV}^T$

We assume here that USV is the function of singular value decomposition in [25], where U is a unitary matrix and S is a diagonal matrix with nonnegative real numbers on the diagonal. We are not going to cover the details about the singular value decomposition since it is beyond the interest of MPEG7 and this chapter. Also, we assume that:

$$\widetilde{\chi} = \begin{bmatrix} \widetilde{\chi}_1^{\,T} \\ \widetilde{\chi}_2^{\,T} \\ \vdots \\ \vdots \\ \widetilde{\chi}_M^{\,T} \end{bmatrix}$$

where $\widetilde{\chi}_k = \dfrac{\chi_k}{r}$ and $\chi_k = 10\log_{10}(y_k)$

Additionally, we assume here that y_k is a vector of power spectrum coefficients in a frame k, which are transformed to log scale and then normalized $r = \sqrt{\sum_{k=1}^{N} \chi_k^2}$ and N is the total number of frequency bins (which is 32 in 1/4 octave resolution).

Spectrum Projection Functions are a vector used to represent low-dimensional features of a spectrum after projection against a reduced rank basis:

$$\mathbf{y}_t = \left[r_t \widetilde{\mathbf{x}}_t^{\,T} \mathbf{v}_1, \widetilde{\mathbf{x}}_t^{\,T} \mathbf{v}_2, \quad \cdots \quad \widetilde{\mathbf{x}}_t^{\,T} \mathbf{v}_k \right]$$

where r_t , $\widetilde{\mathbf{x}}_t^{\,T}$, and $v_1 \quad v_2 \quad \cdots \quad v_k$ are computed in the spectrum basis functions

Harmonic Peaks are a sequence of local peaks of harmonics for each frame.

$$A(i, harmo) = \max_{m \in [a,b]} (|X(m,i)|) = |X(M,i)|$$

$$f(i, harmo) = M \times DF$$

$$a = floor\left((harmo - c)\frac{f0}{DF} \right), \; b = ceil\left((harmo + c)\frac{f0}{DF} \right)$$

where $f0$ is the fundamental frequency in the i^{th} frame, *harmo* is the order number of a harmonic peak, *DF* is the size of the frequency bin, where the total number of the frequency bin is *NFFT*, and *c* is the coefficient of the search range for local maxima, which is set to 0.10 in this dissertation.

Log Attack Time is defined as the logarithm of the time duration between the time the signal starts to the time it reaches its stable part, where the signal envelope is estimated by computing the local mean square value of the signal amplitude in each frame.

$$LAT = \log_{10}(T1 - T0)$$

where $T0$ is the time when the signal starts, and $T1$ is the time the signal reaches its sustained part (for harmonic sounds, where a convolution window is used to detect sustained part with empirical threshold) or maximum part (for percussive sounds).

Spectral Centroid is computed as the power weighted average of the frequency bins in the power spectrum of all the frames in a sound segment with a Welch method [39].

$$SC(frame) = \frac{\sum_{k=1}^{powerspectrum\_size} f(k) \cdot S(k)}{\sum_{k=1}^{powerspectrum\_size} S(k)}$$

2.2.2 Non-MPEG7 features

More statistical descriptors have been used in the dissertation for the purpose of compact representation of musical acoustical features and they are widely used in the literature.

Spectrum Centroid describes the gravity center of the spectrum [34][41].

$$C_i = \frac{\sum_{k=1}^{N/2} f(k)\left|X_i(k)\right|}{\sum_{k=1}^{N/2}\left|X_i(k)\right|}$$

where N is the total number of the FFT points, $X_i(k)$ is the power of the k^{th} FFT point in the i^{th} frame, and $f(k)$ is the corresponding frequency of the FFT point.

Zero crossing counts the number of times that the signal sample changes signs in a frame [34].

Roll-off is a measure of spectral shape, which is used to distinguish between voiced and unvoiced speech [19]. The roll-off is defined as the frequency below which C percentage of the accumulated magnitudes of the spectrum is concentrated, where C is an empirical coefficient.

$$\sum_{k=1}^{K}\left|X_i(k)\right| \le C \cdot \sum_{k=1}^{K}\left|X_i(k)\right|$$

Flux is used to describe the spectral rate of change [34]. It is computed by the total difference between the magnitude of the FFT points in a frame and its successive frame.

$$F_i = \sum_{k=1}^{N/2}\left(\left|X_i(k)\right| - \left|X_{i-1}(k)\right|\right)^2$$

Mel frequency cepstral coefficients (MFCC) describe the spectrum according to the human perception system in the mel scale [20]. They are computed by grouping the STFT points of each frame into a set of 40 coefficients by a set of 40 weighting curves with logarithmic transform and a discrete cosine transform (DCT).

Tristimulus and similar parameters describe the ratio of the energy of 3 groups of harmonic partials to the total energy of harmonic partials. The following groups are used: fundamental, medium partials (2, 3, and 4) and higher partials (the rest).

Parameters describing contents of various groups of harmonics are used in our research. They are: first modified tristimulus parameter Tr_1, $A_{1\text{-}2}^{\ 2}$ power difference of the first and the second partial in dB scale, groups of other harmonic partials ($h_{3,4}$, $h_{5,6,7}$, $h_{8,9,10}$), and also contents of odd and even partials (*Od* and *Ev*).

$$Tr_1 = A_1^{\ 2} \Big/ \sum_{n=1}^{N} A_n^{\ 2}$$

$$h_{3,4} = \sum_{i=3}^{4} A_i^{\ 2} \Big/ \sum_{j=1}^{N} A_j^{\ 2}$$

$$h_{5,6,7} = \sum_{i=5}^{7} A_i^{\ 2} \Big/ \sum_{j=1}^{N} A_j^{\ 2}$$

$$h_{8,9,10} = \sum_{i=8}^{10} A_i^{\ 2} \Big/ \sum_{j=1}^{N} A_j^{\ 2}$$

$$Od = \sqrt{\sum_{k=2}^{L} A_{2k-1}^{\ 2}} \Big/ \sqrt{\sum_{n=1}^{N} A_n^{\ 2}}$$

$$Ev = \sqrt{\sum_{k=2}^{M} A_{2k}^{\ 2}} \Big/ \sqrt{\sum_{n=1}^{N} A_n^{\ 2}}$$

Mean frequency deviation for low partials

$$\overline{fd}_{\ m} = \sum_{k=1}^{5} A_k (\Delta f_k / (k f_1)) \Big/ \sum_{k=1}^{5} A_k$$

2.3 Timbre classification based on feature database

In k-nearest-neighbor prediction, the training data set is used to predict the value of a variable of interest for each member of a "target" data set. The structure of the data is that there is a variable of interest (e.g., the instrument) and a number of conditional features. It is considered to be a lazy learning model, by which training is not necessary and learning is extremely fast. One drawback is that *k* is an empirical value, which needs to be tuned among different classes of sounds.

Martin and Kim [24] employed the KNN algorithm to a hierarchical classification system with 31 features extracted from Correlograms which is a three-dimensional representation of the signal. With a database of 1023 sounds they achieved 87% of successful classifications at the family level and 61% at the instrument level when no hierarchy was used. The accuracy at the instrument level was increased to 79% by using the hierarchical procedure but it degraded the performance at the family level (79%). Without including the hierarchical procedure performance figures were lower than the ones they obtained with a Bayesian classifier. The fact that the best accuracy figures are around 80% and that Martin have settled into similar figures can be interpreted as an estimation of the limitations of the KNN algorithm (provided that the feature selection has been optimized with genetic or other kind of techniques). Therefore, more powerful techniques should be explored.

Bayes Decision Rules and Naive Bayes classifiers are simple probabilistic classifiers by which the probabilities for the classes and the conditional probabilities for a given feature and a given class are estimated based on their frequencies over the training data. They are based on probability models that incorporate strong independence assumptions,

which often have no bearing in reality, hence are naive. The resultant rule is formed by counting the frequency of various data instances, and can be used then to classify each new instance. Brown [4] applied this technique to Mel-Cepstral Coefficients by a K-means clustering algorithm and a set of Gaussian mixture models. Each model was used to estimate the probabilities that a coefficient belongs to a cluster. Probabilities of all coefficients were then multiplied together and were used to perform the likelihood ratio test. It then classified 27 short sounds of oboe and 31 short sounds of sax with an accuracy rate of 85% for oboe and 92% for sax.

Neural networks process information with a large number of highly interconnected processing neurons working in parallel to solve a specific problem. Neural networks learn by example. Cosi [7] developed a timbre classification system based on auditory processing and Kohonen self-organizing neural networks. Data were preprocessed by peripheral transformations to extract perception features, then were fed to the network to build the map, and finally were compared in clusters with human subjective similarity judgments. In the system, nodes were used to represent clusters of the input spaces. The map was used to generalize similarity criteria even to vectors not utilized during the training phase. All 12 instruments in the test were quite well distinguished by the map. Hidden Markov Model is a statistical model by which the extracted model parameters can be used to do sensitive database searching. In a regular Markov model, the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. A hidden Markov model adds outputs: each state has a probability distribution over the possible output tokens. This technique has been successfully applied in speech recognition [16] and natural language processing [8].

Paulus and Virtanen [25] developed a system with this technique for automatic transcription of drum instruments from polyphonic music signals. A background model with only one state is created for each instrument to describe the sound when the target instrument is played. The signal is divided into 2048 frames, and a set of features is extracted from each frame. The most likely model sequence of sound presence and absence is determined by decoding the signal on the location where the instrument is hit, and the second models all the other parts of the signal. The feature distributions in each state are modeled with Gaussian-mixture models (GMMs). Three types of instruments have been evaluated in their experiments: kick drum, snare drum, and hi-hats. Total average classification rate was from 44% to 59.7%. The drawbacks of the system include modeling in the location of a hit with a fixed context length instead of with a sound properties oriented context length, and limitation of features used in the experiment. This technique was used to deduce the most useful attributes in classifying sounds and to compare different resultant sound classes by different attributes. However, results regarding the classification of new sounds have not yet been published.

Binary Tree is a data structure in which each node contains one parent and not more than 2 children. It has been pervasively used in classification and pattern recognition research. Binary Trees are constructed top-down with the most informative attributes as roots to minimize entropy. Jensen and Arnspang [14] proposed an adapted Binary Tree with real-valued attributes for instrument classification regardless of pitch of the instrument in the sample.

Various classifiers for a small number of instruments have been used in musical instrument estimation domain in the literature; yet it is a non-trivial problem to choose

the one with optimal performance in terms of estimation rate for most western orchestral instruments. It is common to apply the different classifiers on the training data based on a specific group of features extracted from raw audio files and get the winner with the highest confidence for the testing music sounds. However, different instruments have different acoustic characters and they usually need different features to model the classifiers. In this dissertation, we try to address this issue in chapter 3.

2.4 New temporal features based on the statistical description of power spectrum

Extracting the spectral or ceptral features from the signal to describe the timbre information has been the predominant method in literature for the purpose of identification of musical instrument sounds. In order to describe the power spectrum, MEPG-7 has already proposed many useful spectral features such as spectral centroid and spectral spread. However the temporal features of musical sounds can also provide some timbre related characteristics, which complement the spectral-based features to fully represent the timbre quality of the sound. In Figure 2.4, the flute and trumpet show relatively similar spectral envelope, and the French horn and trombone also share the same pattern in spectrum, which means these orchestral instruments are not easily discriminated from each other solely by spectral features. The temporal features could play a more important role in the identification of these instruments.

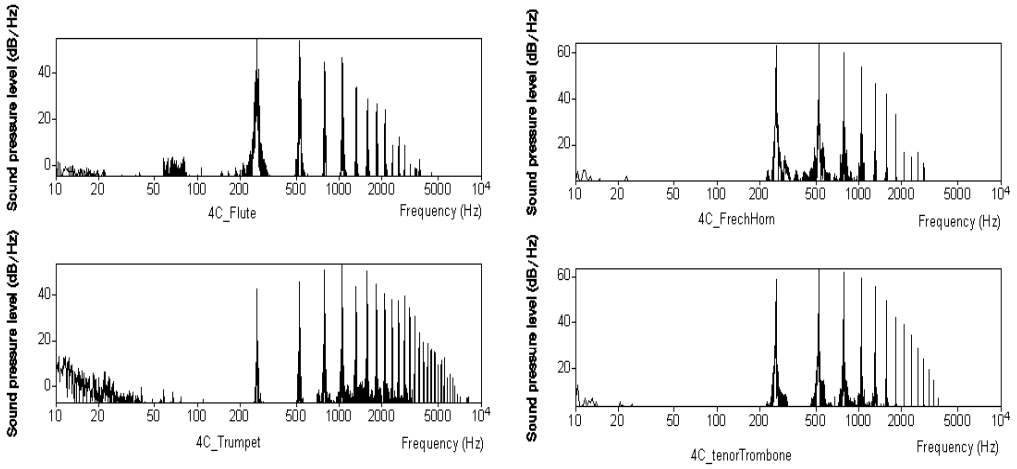

Figure 2.4 Similarity among the log-frequency spectrum of different instrument sounds

We can easily discriminate these sounds according to the attack time, as Figure 2.5 shows.

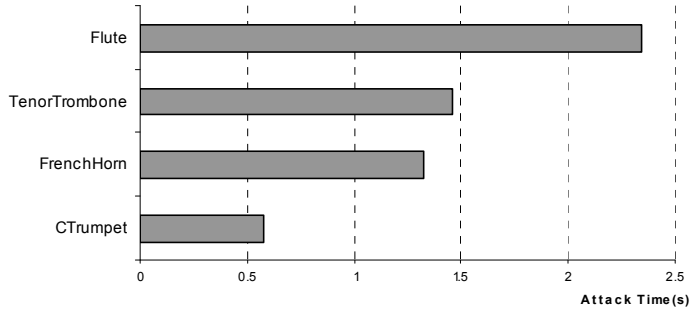

Figure 2.5 Attack time of different instrument sounds

Each of these instruments have unique attack time, which is a temporal feature introduced by the MPEG-7 standard. But this feature would not be useful for timbre identification in the polyphonic sound since there are several different instruments playing simultaneously which make it hard to detect the attack time of each single instrument signal. For the purpose of auto-indexing each musical piece also requires segmentation into small frames with each frame being analyzed separately. The frame length could be less than 0.5s in order to achieve the high resolution of indexing (the identification of each consecutive short time period of the signal) and clear single

instrument pattern. Therefore, it is difficult to make use of this onset feature to discriminate instrument sounds. We propose the new temporal features to address this issue. The new features are directly calculated from the two instantaneous frame-wise spectral features: spectral spread and spectral centroid. The new features are calculated as follows:

$$S'_i = (S_{i+1} - S_i)/S_i$$
$$C'_i = (C_{i+1} - C_i)/C_i$$

where S_{i+1}, S_i and C_{i+1}, C_i are the log spectral spread and centroid of two consecutive frames: frame *i+1* and frame *i*. The changing ratios of spectral spread and spectral centroid for two consecutive frames are considered as the first derivatives of the spread and spectral centroid: S' and C' . By following the same method, we also calculate the changing ratio of S'_i and C'_i, which are considered as the second derivatives of the spectral spread and centroid.

$$S''_i = (S'_{i+1} - S'_i)/S'_i$$
$$C''_i = (C'_{i+1} - C'_i)/C'_i$$

Obviously, S'_i and C'_i contain the temporal information which captures the spectral evolution patterns across the every two adjacent frames. S''_i and C''_i further captured the temporal information across every three adjacent frames. By adding those four new temporal features into our database, we try to discover the timbre patterns which embed both in the individual frame and across the multiple frames. The Table 2.1 displays the comparison of the classification results. We use Weka [6] as the classification platform. Both decision tree classifier and KNN classifier are tested on the feature datasets listed in Table 2.1. For decision tree classifier, we choose J48 in Weka, which is the

implementation of C4.5 decision tree algorithm [29]. The confidence factor used for pruning tree (smaller values incur more pruning) is 0.25. The minimum number of instances per leaf is 10. For K-nearest neighbor classifier [9], we choose IBK in Weka, which is the brute force search algorithm for nearest neighbor search. The number of neighbors is 3. Euclidean distance is used as similarity function.

From the results we observe the new temporal features improve the classification result for both decision tree and KNN classifiers.

Table 2.1 classification confidence with temporal features

Experiment	Features	Classifier	Confidence
1	S, C	Decision Tree	80.47%
2	S, C, S', C'	Decision Tree	83.68%
3	S, C, S', C', S'', C''	Decision Tree	84.76%
4	S, C	KNN	80.31%
5	S, C, S', C'	KNN	84.07%
6	S, C, S', C', S'', C''	KNN	85.51%

The confusion matrix comparison between the Experiment 1 and 3 in Figure 2.6 shows the new temporal features improve the discrimination ability of the classifiers among these instruments which share the similar spectrum shape and timbre quality. These instruments are easy to be misclassified as each other. For example, French horn and flute has the similar timbre quality and instantaneous spectral features; before the temporal features were added, the decision tree incorrectly classified 13 French horn instances as flute. After the temporal features were added to the current spectral feature

training database, there are only two French horn instances that are misclassified. There is contribution to the overall improvement of the classifier for the French horn. The same improvement is observed in discrimination ability of the classifier for the violin, viola, flute, tuba, oboe and vibraphone.

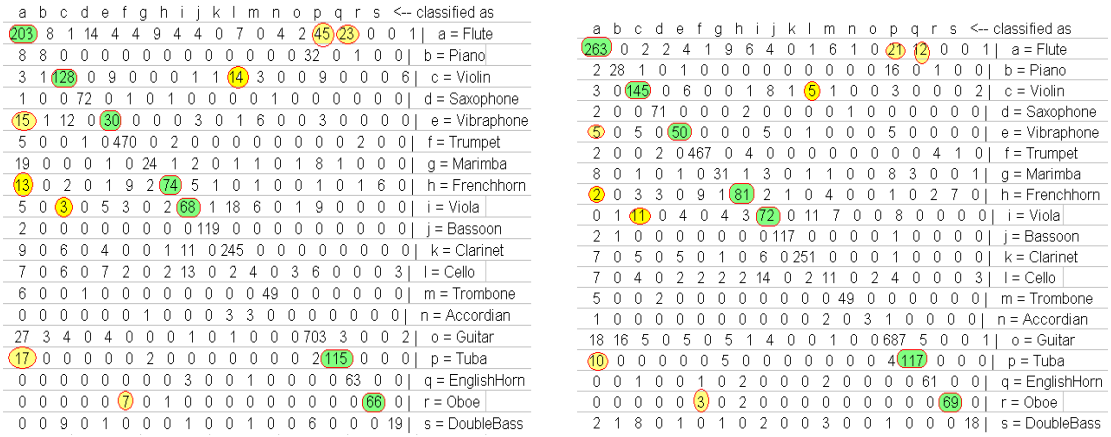

Figure 2.6 Confusion matrices: left is from Experiment A, right is from Experiment C. The correctly classified instances are highlighted in green and the incorrectly classified instances are highlighted in yellow

However we observed in some circumstances that the new temporal features have the deteriorating effect on the classifier. For example, when it comes to discriminating viola from violin, 8 more instances are incorrectly classified, which means those four new features do not necessarily yield better results. However, we see that the overall correctly classified number of viola instances was increased because the new features improve the ability of the classifier in discriminating viola from other instruments such as flute and clarinet, which offset the negative effect on the violin. In this dissertation, we use both precision and recall to evaluation the performance of classifiers. Figure 2.7 shows the definitions of the two measurements.

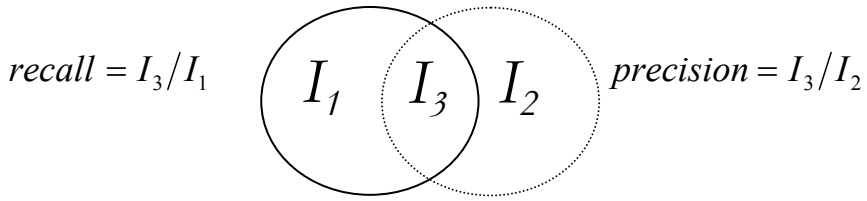

Figure 2.7 Precision and Recall

I_1 is the number of actual instruments playing in the analyzed sound; I_2 is the number of instruments estimated by the system; I_3 is the number of correct estimations

Recall is the measurement to evaluate the recognition rate and precision is to evaluate the recognition accuracy. From the precision results (Figure 2.8), classifications of most instruments are improved except for marimba, French horn, English horn and oboe.

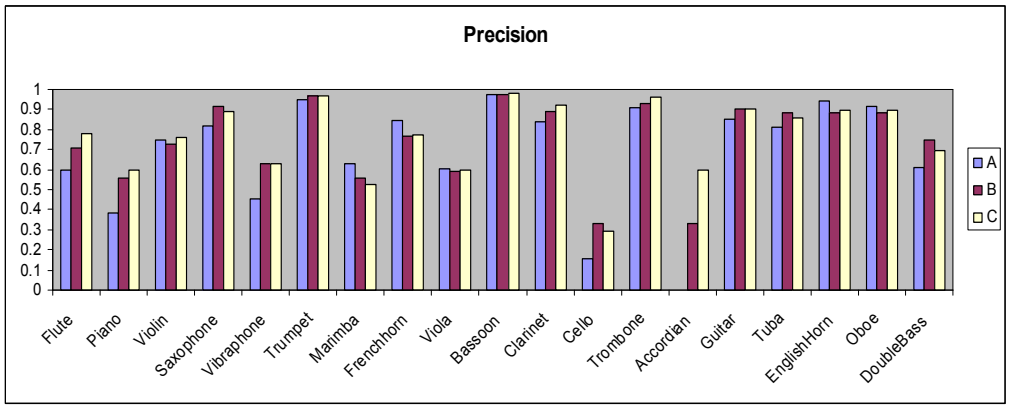

Figure 2.8 Precision of the decision tree for each instrument

From Figure 2.9, among those four instruments marimba, French horn and oboe get higher recall when new temporal features were added.

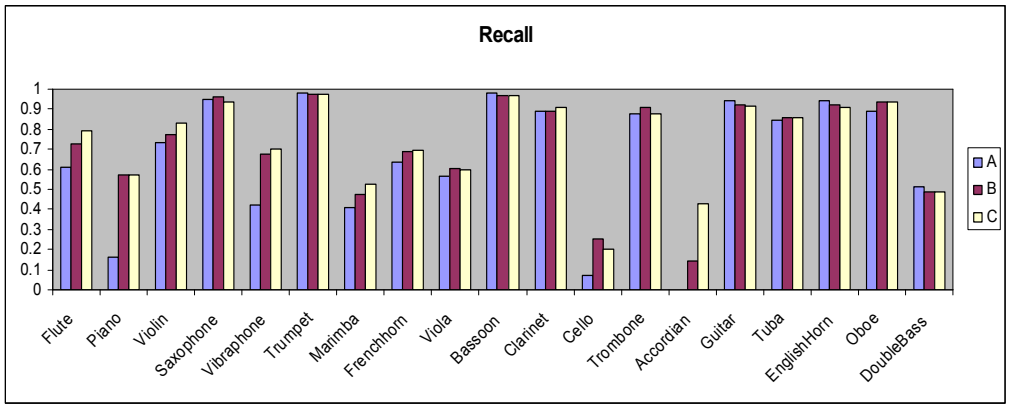

Figure 2.9 Recall of the decision tree for each instrument

In order to take both the precision and the recall into account, we calculate the F-score (also called F-measure) to evaluate the general performance of the classification. The F-score is often used in the field of information retrieval for measuring search, document classification, and query classification performance. Here is the formula of F-score:

$$Fscore = 2 \cdot (precision \cdot recall) / (precision + recall)$$

F-score is the harmonic mean of precision and recall. Figure 2.10 clearly shows that with exception for English horn the classification performance of all of the other instruments is improved with the introduction of the new temporal features.

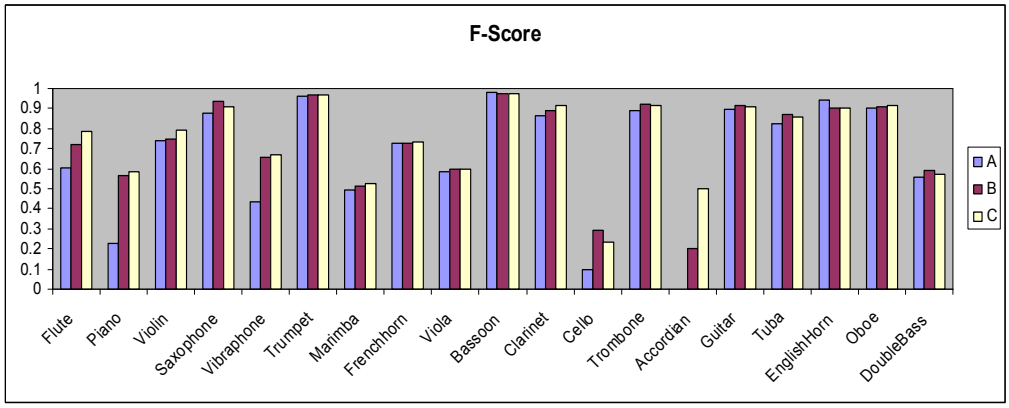

Figure 2.10 F-score of the decision tree for each instrument

CHAPTER 3: TIMBRE ESTIMATION BASED ON MULTI-LABEL CLASSIFICATION

3.1 Polyphonic sound estimation based on Segmentation

One approach to address the issue of multi-timbre estimation in polyphonic sound is to segment the signal into very short frames. Even though a polyphonic sound contains multiple instrument signals as a whole musical piece, there are still less overlapping areas during some small frames which are approximately considered as monophonic slices. The single instrument estimation techniques are then performed on each individual slice or frame. Those monophonic estimations from multiple frames have different instrument information. They are merged as the polyphonic estimation results for a longer period which includes those individual frames. Apparently chances are good that each analysis frame only contains the pure single instrument signal if the frame is small enough. It also provides a good resolution of music auto-indexing system which indexes the musical pieces with timbre information in small segments. On the other hand, the frame can not be too small if it is to cover the full frequency range of musical instruments. For instance, the piano is known to have the widest frequency range among the western instruments: 28Hz to 4.1 KHz. The longest sound wave produced by the piano is 1/28=35.7 ms. In order to provide the sufficient frequency resolution, the length of the frame has to be

larger than the wave length. Other instruments such as the organ even have lower frequency than the piano. In this dissertation we use 120ms as the frame size to cover all the frequency contents that instruments produce.

3.2 Sound separation method based on single-label classification

As we see, the segmentation could not give the accurate estimations for polyphonic sounds since the assumption of the non-overlapping area is not always true during the whole period of the signal. In order to achieve a good multiple timbre estimation, the overlapped areas have to be considered as well. One approach to address this issue is to apply the sound separation techniques along with the traditional classifiers. Each time when one classification label c_i from a set of labels $C = \{c_1, \ldots, c_n\}$ is assigned to the target frame, the sound separation module is applied to subtract the estimated timbre spectrum from the signal so that the information of the estimated single instrument is separated from the frame. Then the classifier can be applied again on the residue of the signal to assign another label C_j and the same process iteratively proceeds until the remnant of the signal is too weak to give any further timbre estimation. Figure 3.1 shows the process of musical instrument recognition system based on the sound separation.

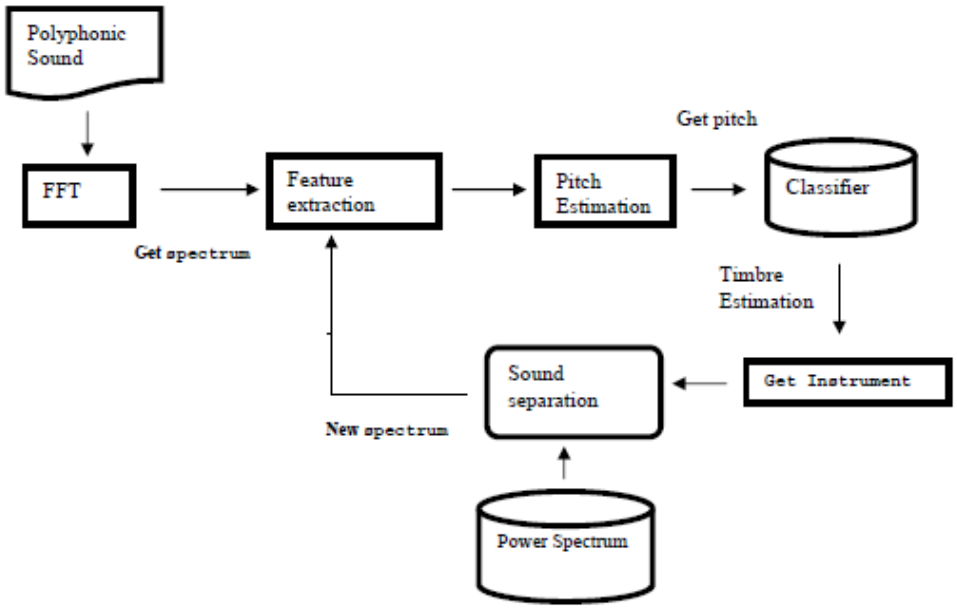

Figure 3.1 Flowchart of musical instrument recognition by sound separation

However, there is one problem with this method. After each sound separation process, the timbre information of the remaining instruments could be partially lost due to the overlap of the spectrum of multiple timbre signals, which makes it difficult to perform the accurate classification on the remnant of sound signal.

3.3 Multi-label classifier trained on the multi-class samples

Instead of giving one classification label at a time, multi-label classification assigns multiple labels $D = \{c_i, \ldots, c_j\}$ from the label set $C = \{c_1, \ldots, c_n\}$, $D \subset C$ to the target object. Some research of multi-label classification has been done in the text categorization area [22][33]. Authors in [18] introduced the multi-label classification method in scene recognition, where a natural scene may contain multiple objects such that the scene can be described by multiple class labels, but they approached the problem by training the samples with multiple labels. However, this is not feasible for the musical database. Each instrument can play at a different note and some instruments can even

have very different timbre when they are played by different techniques. Among string instruments, for example, the mute is sometime used to dampen vibrations and results in a "softer" sound, which affects and alters the timbre. Our training data set consists of 2576 single musical instrument sounds produced by 46 different instruments. Since every instrument has the sounds played at different notes or by different methods, the average number of sounds for each musical instrument is 56.

If we want to have the polyphonic training database of duo, trio or quartet by synthesizing two, three or four different types of single instrument sounds, there will be $C_{46}^{2}C_{56}^{1}C_{56}^{1} + C_{46}^{3}C_{56}^{1}C_{56}^{1}C_{56}^{1} + C_{46}^{4}C_{56}^{1}C_{56}^{1}C_{56}^{1}C_{56}^{1} \approx 1.6$ trillion different possible combinations. Therefore it is almost impossible to construct complete multi-class training samples to derive multi-label classifiers. Fig 3.2 illustrates the difference between the multi-label classification system and the multi-class classification system.

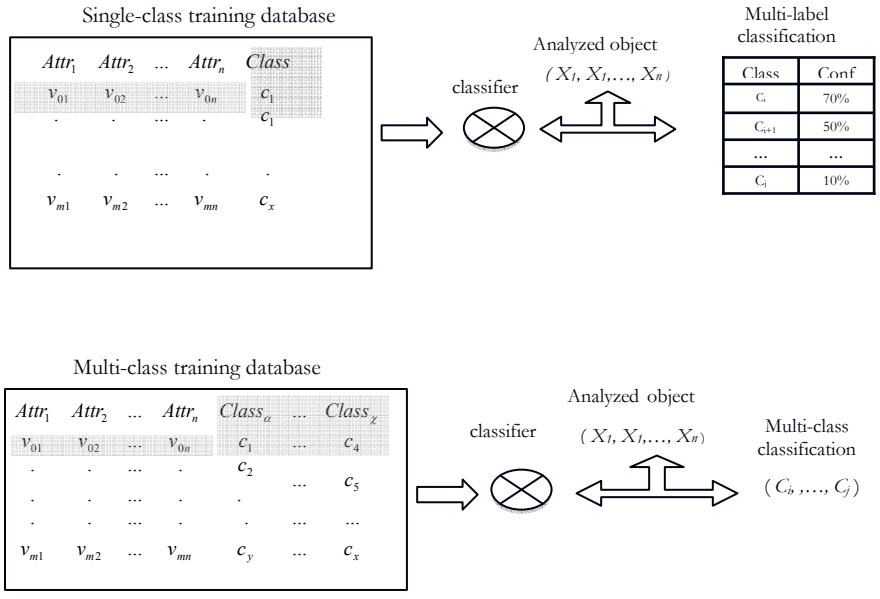

Figure 3.2 Comparison between Multi-label classification and Multi-class classification

3.4 Multi-label classifier trained on the single-class samples

In [17], researchers explored the emotion classification in musical pieces with only 6 emotional classification labels involved and a small number of timbre relevant features used in classification. Emotion is the higher level information which could be further derived from the lower level musical information such as pitch, brightness, rhythm and timbre. However, in our study, we need to classify more details in timbre level which involves more classification labels (more than 46 musical instruments). The large number of musical instruments makes the classification task more complicated and challenging. So far there is no work that has been done in timbre estimation area with multi-label classification based on single-class training database.

Decision trees [28] represent a supervised approach to classification. The structure is simple where non-terminal nodes represent tests on one or more attributes and terminal nodes reflect decision outcomes. Usually only one class label with the highest confidence is assigned to the estimated object, and the other candidate classes are simply ignored. However, in polyphonic sound timbre estimation, those ignored candidates could be the correct estimations to the other multiple timbre information. We use the multi-label decision tree classification based on the ranking of confidence and support of each candidate. It makes sense to consider the multiple candidates because they represent the

objects which are most similar to the target objects present in polyphonic sound.

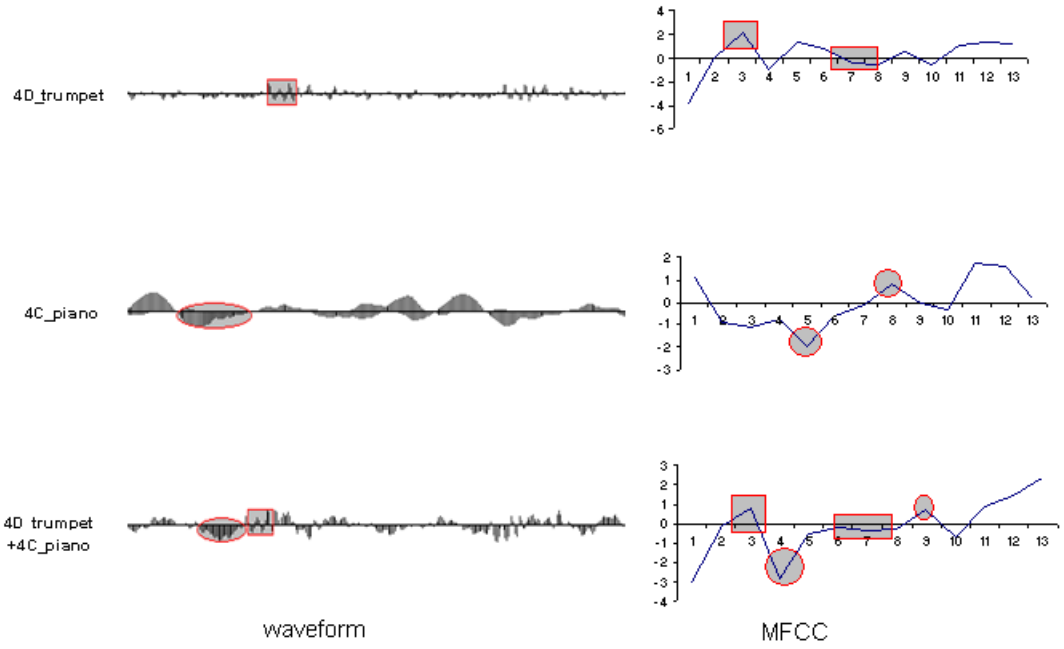

Figure 3.3: Sub-pattern of single instrument in mixture both in sound waveform and MFCC feature space

As Figure 3.3 shows us, the wave patterns of the single flute and the single trombone could still been observed in the mixed sound. Even though each single instrument's pattern in the feature space of the mixture is distorted to some extent, the distinct patterns are still preserved (as the Figure 3.3 shows). The assumption is that both single instruments could be identified by comparing the similarity of feature vector of the mixture to the reference instruments feature database. The most similar matches are considered the timbres simultaneously occurring in the polyphonic sound. These similar feature patterns indicate that the corresponding instruments would have the higher confidence and support which is calculated by the classifier. Figure 3.4 shows how the multi-label classification works.

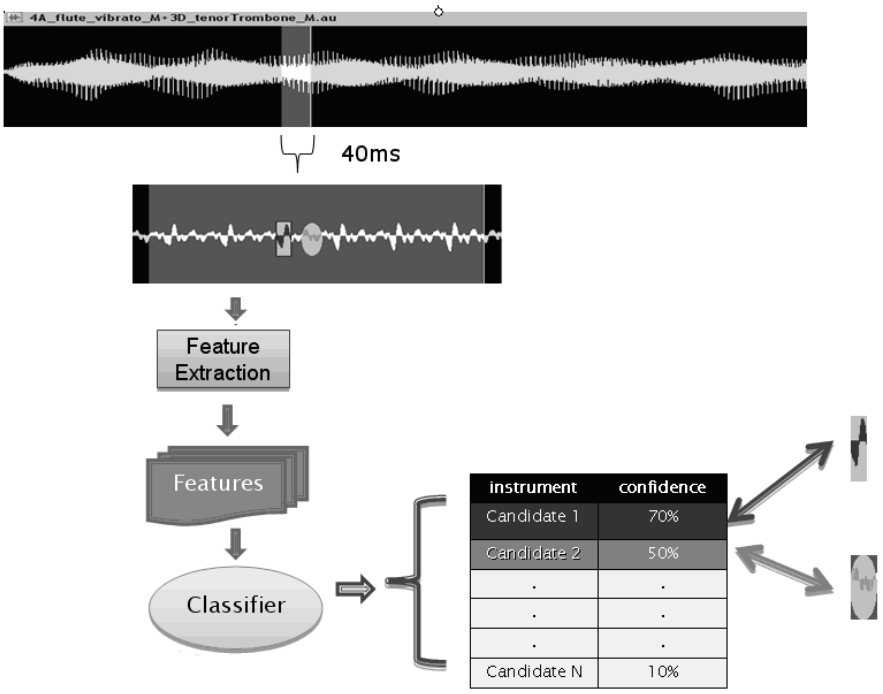

Figure 3.4 Multi-label decision tree classifications

Let $S = \{X, F \cup C\}$ be the training database, where X are the instances of the database, $C = \{c_1, \ldots, c_n\}$ are all the class labels, and $F = \{f_1, \ldots, f_m\}$ is the *m*-dimensional feature vector which we extract from standard training instrument sounds to build the decision tree *DT* and use it to estimate the target object**.** We are only interested in the timbre estimations for each indexed window which is the larger unit than the frame itself. Each frame is too short to be meaningful for the users. The indexed window is usually seconds long and the actual size is defined by the resolution requirement of the auto indexing system. In this dissertation, we set the indexed window size as one second. Let $X = \{x_1, \ldots, x_t\}$ be segmented frames from one indexed window of the analyzed audio signal. The classifier then estimates each frame x_i and assigns the confidence $conf(c_i)$ for every instrument label c_i. We choose top *N* labels with the highest confidence as the candidate labels for the current frame and discard the other labels with the low

confidence. After all the individual frames within the indexed window are classified, the average confidence of each candidate instrument label is calculated for the whole window $\sum_{i=1}^{t} Conf(c_i) \Big/ t$. If the average confidence is larger than threshold λ the candidate label is kept, otherwise it is discarded. Thus the multiple instrument labels are for the index window. If some candidates have high confidence but only in a very short time period (very few frames), it will be considered as random noise and be excluded by their relatively low average confidence in the whole indexed window. If some candidates occur frequently but have very low confidence at each frame, they are also discarded as the background noise. The advantage of this process is that it uses the information of music context during the longer period to further adjust the frame-wise estimation results.

We develop the MIR system based on the multi-label classification method and test it with the synthesized polyphonic sounds. The system uses MS SQLSERVER2005 database system to store the training dataset and MS SQLSERVER analysis server as the data mining server to build the decision tree and process the classification request.

Training data: The audio files used in this research consist of stereo musical pieces from the MUMS samples and samples recorded in the KDD Lab at UNC Charlotte. Each file has two channels: left channel and right channel, in .au or .snd format. These audio data files are treated as mono-channel, where only left channel is taken into consideration, since successful methods for the left channel will also be applied to the right channel, or any channel if more channels are available. Additionally, 2917 single instrument sound files are taken from MUMS to be used and include 45 different instruments. Each sound stands for one specific instrument played at a certain note. And many instruments can produce different timbres when they are played by different techniques. MFCC are

extracted from each frame of those single instrument sounds according to the equations described by the MPEG-7 standard. The frame size is 120 ms and the overlap between two adjacent frames is 80ms to reduce information loss caused by windowing function. The hop size of the signal is 40ms. The instrument sound which only lasts three seconds is segmented into 75 overlapped frames. The total number of frames for the entire feature database reaches to about one million. The classifier is trained by this feature database.

Testing data: 308 mixed sounds are synthesized by randomly choosing two single instrument sounds from 2917 training data files. MFCC are also extracted from those mixtures to perform classification with the classifier built by the training data. The same frame size and hop size are used for the mixtures as training data when frame-wise analysis is performed.

The Average recall, precision and recall of all the 308 sounds estimation are calculated to evaluate each method. Parameter ***N*** indicates the maximum number of instrument labels estimated by the classifier for each frame during the frame-wise process. **Experiment 1** applies the traditional single-label classification which means the classifier only assigns **one** label for each frame and it uses the sound separation in order to get multiple estimations for each frame, **experiment 2** applies the multiple label classification which assigns **2 labels** to each frame by the classifier according to each label's confidence. **Experiment 3** removes the sound separation process from the algorithm. **Experiment 4** and **5** increase the number of labels classified by the classifier during the frame-wise estimation to **4** and **8**. The indexed window size for all the experiments is one second long and the output of total number of estimations for each

indexed window is controlled by threshold $\lambda = 0.4$,which is the minimum average confidence of instrument candidates.

Table 3.1 Timbre estimation results based on different approaches

Number	description	Recall	Precision	F-score
1	N=1,separation algorithm	54.5%	39.2%	45.60%
2	N=2, separation algorithm	61.2%	38.1%	46.96%
3	N=2, without separation algorithm	64.3%	44.8%	52.81%
4	N=4, without separation algorithm	67.7%	37.9%	48.60%
5	N=8, without separation algorithm	68.3%	36.9%	47.91%

Table 3.1 shows the comparison of the results from the different timbre estimation methods. The recall is raised from 54.5% to 61.2% after the multi-label classification is applied. However precision of the estimation does not improve much. When we remove the sound separation from the multi-label classification method, the recognition rate further rises to 64.3% and precision is also improved from 39.2% to 44.8%. We conclude that the multi-label classification yields better results than the single-label classification by avoiding the sound separation

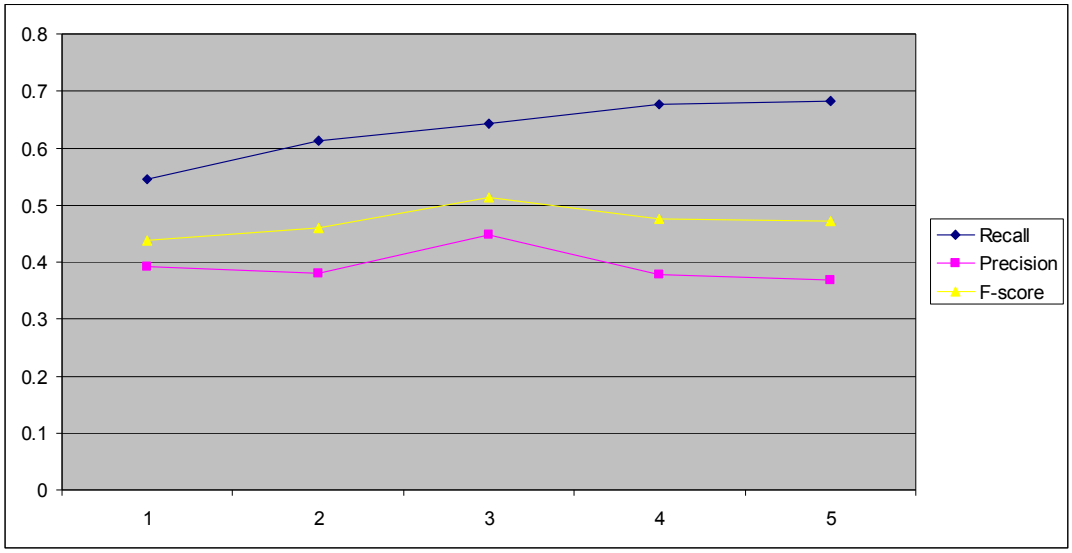

Figure 3.5 Timbre estimation results based on different approaches

The recall is further improved when more labels are assigned at each frame by the classifier as Figure 3.5 shows. However the precision and overall F-score do not improve and even drop a little. The parameter *N*, therefore, needs to be adjusted carefully by the user in order to get the optimal results.

CHAPTER 4: TIMBRE ESTIMATION BASED ON SHORT-TERM SPECTRUM MATCH

4.1 Insufficiency and overlapping of features

Feature based datasets are more efficient to work with classifiers than lower level representations of the signal; however, there is usually information loss during the feature extraction process. Acoustic features, such as harmonic peaks, MFCC and spectral flatness are the abstract or compressed representations of the signal. They are basically calculated in the way of approximating the human auditory system's response to the sound quality. During this simplification and approximation process, the so-called "irrelevant" information (such as inharmonic frequencies or partials) in the audio signal is removed and the primary information relevant to the timbre is believed to be preserved. This highly abstract information would be sufficient to distinguish some musical instruments in the polyphonic sounds when those instrument sounds are quite different from each other, such as piano and flute. When it is necessary to separate the instruments from the ones that fall into the same family and usually share the similar timbre qualities, more information from the raw signal is needed besides the acoustic features. For instance, the similar MFCC pattern of violin and viola usually make it difficult for the system to distinguish them from each other. This also happens to the double-bass and guitar. This is because those "irrelevant" frequencies also play an important role in the timbre sensation for human hearing system. Harmonic partials are commonly regarded as

a necessary aspect to the perception of timbre, but they are not sufficient in some cases. Timbre is dependent on other frequencies in the spectrum as well.

On top of that, the features are relatively easy to be extracted from the monophonic music sound which only contains singular non-layered sound. However, this is not the case in polyphonic sounds. Because of the overlapping of multiple instruments signal in the spectrum, especially when instruments have very similar harmonic patterns, the feature patterns of the instruments could be blurred and not discernible. Thus the fact that discriminating one instrument from another depends on more details from raw signals leads to another way of pattern recognition: directly detecting distinct patterns of instruments from the lower representation of signal.

4.2 Sub-Pattern in short-term spectrum

Timbre detection directly in time domain is not feasible since it is mainly related to the frequency pattern. Therefor we have to choose the short-term power spectrum as the low level representation of the signal. The short-term power spectrum is calculated by short time Fourier transform (STFT). Figure 4.1 shows the spectrum slice for the flute and trombone and their mixture sound. Each slice is 0.12 seconds long which the same size of the frame we discussed in the previous chapter.

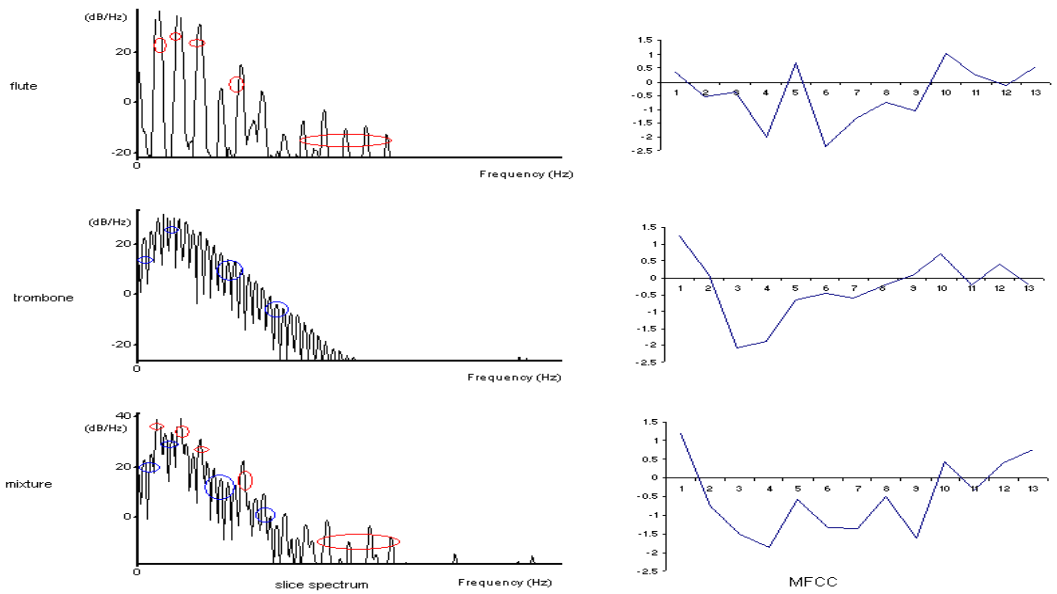

Figure 4.1 Sub-pattern in spectrum and MFCC of flute, trombone and their mixture sound.

The power spectrum patterns of the single flute sound and the single trombone sound could still be observed in the spectrum of the mixture sound (as marked in the Figure 4.1). It means each single instrument frequency patterns which related to their unique timbre qualities are preserved in the mixture signal. Those small patterns (or sub-patterns) cause the human hearing system to accurately recognize the two different instruments from the. However it is very difficult to separate the whole single instrument signal from the mixture and further extract the higher level feature because those sub-patterns are intervened by each other and buried in the spectrum. This is the reason they are not observed in the feature space of the mixture signal (Figure 4.1).

4.3 Timbre Pattern Match Based on Power Spectrum

In order to identify the timbre qualities of multiple instruments in the polyphonic sound accurately, we work directly on the power spectrum instead of extracting dozens of features to represent the signal. When each frame of the analyzed signal is processed, we calculate the spectral similarity between the analyzed frame and the frames of all the

single instrument sounds in the training database. The instrument labels of the matched spectra in the reference database are the multiple timbre estimations for the analyzed frame. K Nearest Neighbor (KNN) algorithm is the lazy machine learning algorithm that provides the flexible way to classify the target object multiple times. Figure 4.2 shows how the multi-label classification based on the KNN classifier works. After matching the spectrum of the analyzed frame against the reference spectra in the training dataset at the first run, K nearest neighbors are given by the KNN classifier. The label of majority among those K candidates is selected as the first instrument estimation. The confidence for the selected instrument is calculated as $conf = \omega / k$, where ω is the number of occurrences of the selected instrument in the K neighbors. The distance or the similarity measure between the selected instrument spectrum and the analyzed spectrum is also taken into account. The overall score for the selected instrument is calculated as $score = conf + (1 - dist)$, where *dist* is the normalized distance within the range of [0, 1].

Then the analyzed spectrum is classified by KNN again by excluding the previous selected instrument label from the reference database. KNN gives another K nearest neighbors as the possible instruments estimation and the majority label is selected as the second instrument estimation for the analyzed frame. The score is also calculated for this estimation. The previous instrument is not included in the reference database at the second run matching therefore the duplicate estimations of the same instrument are avoided. Following several KNN classification processes, the multiple timbres are estimated for the analyzed frame. After all the frames in the indexing window are

classified, the average scores of each instrument are calculated and compared with the threshold λ. All the instrument estimations with the scores lower than λ are discarded.

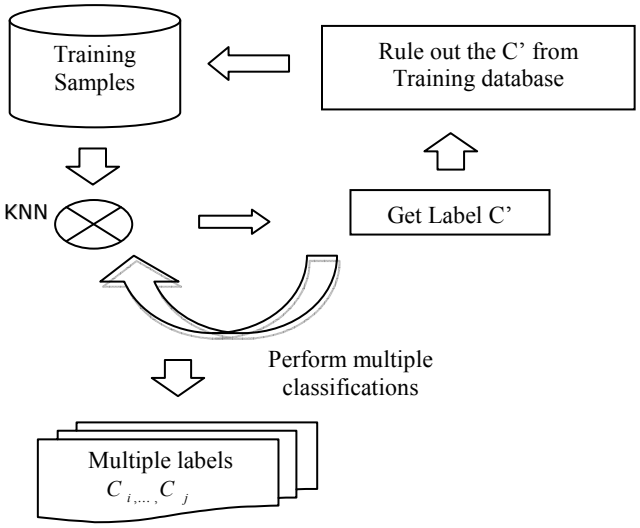

Figure 4.2 Multi-label classification based on KNN

4.4 Experiments and Results

To simplify the problem, we only perform the tests on the middle C instrument sounds, i.e. for pitch equal to C4 in MIDI notation, of frequency 261.6 Hz (for A4 tuned to 440 Hz). The training subset including the power spectrum from 3323 frames has been selected from the entire training database. Those frames are extracted by short time Fourier transform from the following 26 single instrument sounds: Electric Guitar, Bassoon, Oboe, B-flat Clarinet, Marimba, C Trumpet, Eflat Clarinet, Tenor Trombone, French Horn, Flute, Viola, Violin, English Horn, Vibraphone, Accordion, Electric Bass, Cello, Tenor Saxophone, B-Flat Trumpet, Bass Flute, Double Bass, Alto Flute, Piano, Bach Trumpet, Tuba, and Bass Clarinet. To compare the results with the traditional feature-based classification methods, we also extract the following 5 groups of both temporal and spectral features. Fifty-two audio files are mixed (using Sound Forge sound editor) by two of these 26 single instrument sounds. These mixture audio files have been

used as testing files. The system uses MS SQLSERVER2005 database system to store the reference spectra database and K nearest neighbor algorithm as the classifier. Euclidean is used as the distance metric for KNN.

In experiment 1, we apply multi-label classification based on the features described in the previous chapters. In the experiments 2 and 3, we applied spectrum-match based KNN classification with different *K* values. *n* is the number of labels assigned for each frame, which means KNN classification is performed *n* times for each frame .

Table 4.1 Feature based recognition VS Spectrum based recognition

Number	Description	Recall	Precision	F-score
1	Feature-based (n=4)	64.3%	44.8%	52.81%
2	Spectrum Match (k=1; n=2)	79.4%	50.8%	61.96%
3	Spectrum Match (k=5; n=2)	82.4%	45.8%	58.88%

From the results showed by Table 4.1, we see that spectrum-based KNN multi-label classification improves both the recall and precision of the timbre estimation for polyphonic sounds. This result shows that spectrum does capture more details of timbre quality of musical instruments than the higher-level features.

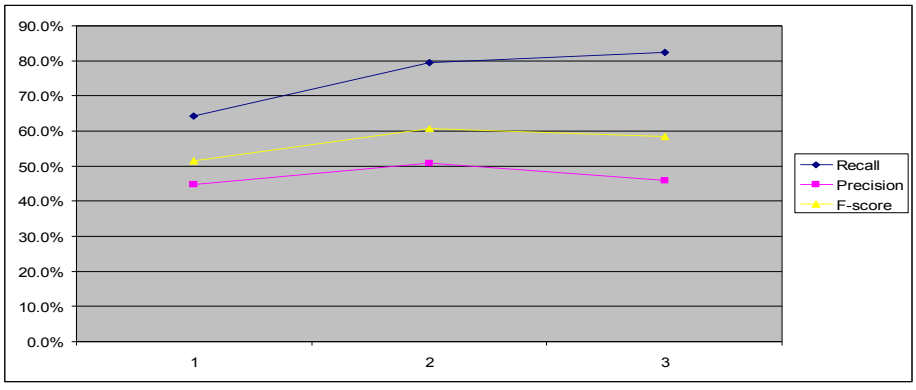

Figure 4.4 Feature based recognition VS Spectrum based recognition.

As Figure4.4 shows, while we achieve both higher recall and precision in the spectrum-based approach, the precision is decreased when the number of neighbors *K* is

increased from 1 (experiment 2) to 5 (experiment 2) for KNN. The parameter for KNN also needs to be adjusted according to specific scenario to yield optimal results.

4.5 Multi-resolution recognition based on power spectrum match

Searching through the lower level representation of signal is very computationally expensive. Power spectrum of each single frame (0.12 s under 44.1K Hz sample rate) contains over eight thousand integer values. If one song lasts around 5 minutes, it produces 5*60/0.12=2500 frames, considering the overlap between consecutive frames, it actually produces 2500*3=7500 frames each of which is matched against huge number of reference frames extracted from standard musical instrument sounds. Such computational complexity is even increased when the tremendous amount of musical sounds from the Internet need to be timbre indentified and indexed. In order to make this approach applicable in the real word, we have to optimize the speed of the matching process.

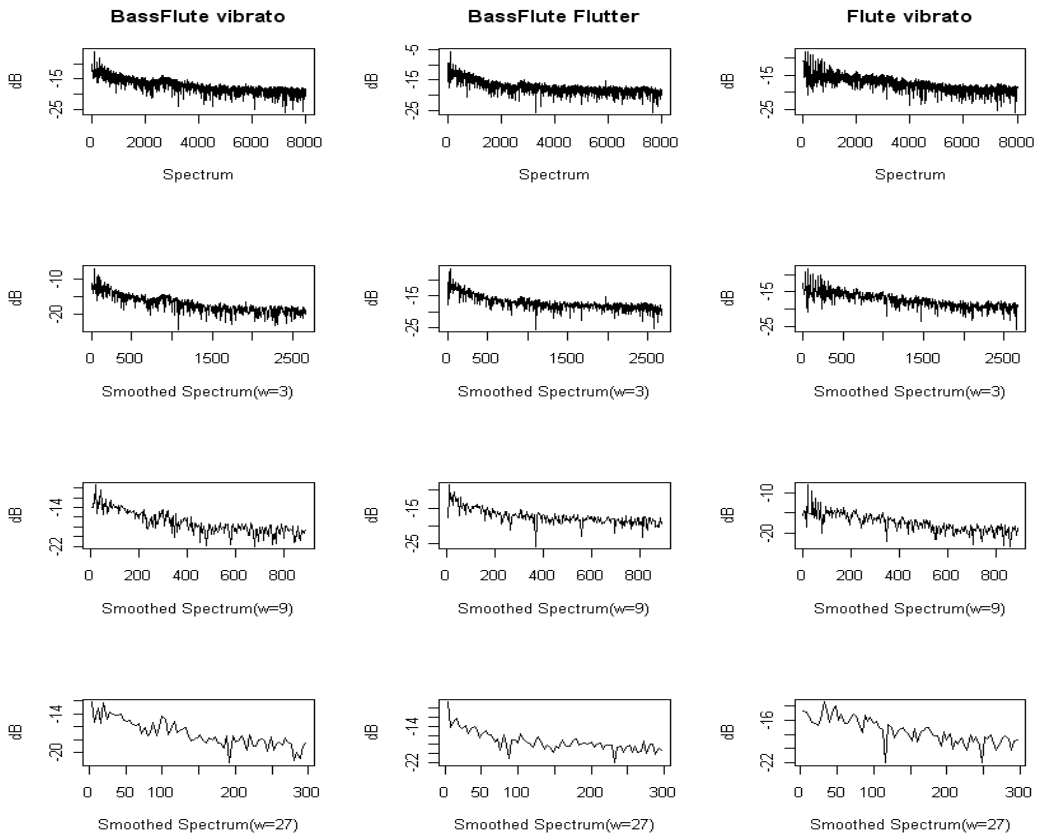

Figure 4.5 Comparisons of spectra after running average smoothing with different window sizes

The logarithm spectra of the three instrument sounds: bass flute vibrato, bass flute flutter, and flute vibrato are shown in Figure 4.5. Each instrument is displayed in one of the three columns. The first row shows the original spectra extracted from the signals. The second row shows the smoothed spectra after applying running average window with the window size of 3 (calculating the mean spectrum value of the 3 consecutive spectrum points). The third and forth rows display the smoothed spectra with the larger window size of 9 and 27. As we can observe, even though the smoothing process results in the spectra with very low resolution, the two bass flute spectra still share the every similar pattern and thus are not difficult to be distinguished from the flute spectrum. The smoothed spectra have fewer dimensions than the original one so that the cost of distance

calculations between the smoothed spectra and the reference spectra is reduced. As we see here, the dimensions of the smoothed spectrum with the window size of 27 are reduced to 300, which is more than 25 times smaller than the original one. Regardless of such compression, there is still sufficient information preserved in this smoothed spectrum to be utilized to identify timbre patterns. The smoothed spectrum actually contains more details than the higher-level representation of acoustic features. We first use highly smoothed spectrum to match against the reference database. After excluding a large number of reference instances from the database, we apply the less smoothed spectrum to perform the matching process against the smaller reference database. Finally the original spectrum is used to perform the matching process within a very small range of reference instrument frames. We call this the multiple-resolution matching method. By matching the spectrum in multiple runs with the different spectral resolution, we effectively reduce the computation complexity while the timbre relevant patterns are still preserved. This is similar to the human perception system. When most people recognize an object, instead of checking from beam to beam (assuming that beam is the atomic unit in the picture), they would start from the outline of the shape, which is an abstract of details. In our case, the classification system starts matching the spectral vectors with the resolution from low to high. Each round of comparison rules out a certain percentage of unlikely spectrum patterns. The hierarchical-structured recognition methods will be further discussed in the next chapter. Let us first look at the experiment result to see how the multiple resolution matching works. The testing dataset consisted of 52 music recording pieces mixed by Sound Forge sound editor. Each piece was played by two

different musical instruments. Euclidean is used as the distance metric for KNN algorithm ($k=7$).

Table 4.2 results of multiple resolution matching

methods	seconds	recall	precision	F-Score
raw Spectrum match	2560	83.6%	49.4%	62.1%
Multiple resolution(w=3, p=0.5)	511	82.5%	48.7%	61.2%
Multiple resolution(w=4, p=0.5)	524	82.3%	48.3%	60.9%
Multiple resolution(w=5, p=0.5)	550	81.4%	47.6%	60.1%

Table 4.2 shows that comparison between the results of multiple resolution and simply non-smoothed spectrum matching approach. Parameter w is the window size for the moving average. Second run of smoothing process is based on the previous smoothed spectrum with the same window size of t. We perform totally four runs of smoothing and get four smoothed spectra (s_1, s_2, s_3, s_4) for each single frame. The power spectrum matching starts from the most smoothed spectrum s_4. It has the lowest resolution and is lest expensive to match. After the first round of matching, the algorithm excludes the entire reference spectra database by a certain percentage (which is specified by parameter p). Then the spectrum s_3 with the higher resolution is selected to perform the second round of matching. This process iteratively goes on until all the smoothed spectra have been processed. Finally the original non-smoothed spectrum is matched against the reference database. Because the size of the reference database is significantly decreased at this point, the complexity of power spectrum matching is reduced to a lower level. The experiment shows that the multiple resolution method is five times faster than the non-

smoothed power spectrum matching method. However, the accuracy of the estimation is not affected as the precision and recall shows in table 4.2. The more significant improvement is expected when we deal with musical database in the real world which stores more reference spectra.

CHAPTER 5: CASCADE CLASSIFICATION

Different classifiers for a small number of instruments have been used in musical instrument estimation domain; yet it is a non-trivial problem to choose the one with the optimal performance in terms of estimation accuracy for most western orchestral instruments. A common practice is to try different classifiers on the same training database which contains the features extracted from audio files and select the classifier which yields the highest confidence in the training database. The selected classifier is used for the timbre estimation on analyzed music sounds. There are boosting systems [43], [44] consisting of a set of weak classifiers and iteratively adding them to a final strong classifier. Boosting systems usually achieve a better estimation model by training each given classifier on a different set of samples from the training database, which keeps the same number of features or attributes. In other words, a boosting system assumes there is a big difference among different group of subsets of the training database so that different classifiers are trained on the corresponding subset based on their expertise. However, due to the homogeneous characteristics across all the data samples in a training database, musical data usually could not take full advantage of such panel of learners because none of the given classifiers would get a majority weight. Thus the improvement cannot be achieved by such combination of classifiers.

Also, in many cases, the speed of classification is an important issue. For example, to classify a piece of two-second audio of CD quality based on a short-term spectrum match,

it takes about five minutes to finish the indexing and timbre estimation. When the user submits the musical piece which is normally several times longer than five minutes to MIR system, it would take more than half a day to finish the indexing and timbre estimation. Also, the computation complexity is further increased when more audio samples are added to the training database in order to improve the robustness of classification. To achieve the applicable classification time while preserving high classification accuracy, we introduce the cascade classifier which could further improve the instruments recognition of MIR system.

Cascade classifiers have been investigated in the domain of handwritten digit recognition. Thabtah [38] used filter-and-refine processes and combined them with KNN to give the rough but fast classification with lower dimensionality of features at filter step and to rematch the objects marked by the previous filter with the higher accuracy by increasing dimensionality of features. Also, Lienhart [27] used CART trees as base classifiers to build a boosted cascade of simple feature classifiers to achieve rapid object detection.

To our best knowledge, no work has been done in investigating the applicability and usefulness of cascade classifiers in MIR area. However, it is possible to construct a simple instrument family classifier with a low false recognition rate, which is called a classification pre-filter. When one musical frame is labeled by a specific family, the training samples in other families can be immediately discarded, and further classification is performed within such small subsets, which could be applied with a stronger classifier by adding more features or even calculated in the whole spectral space. Since the number

of training samples is reduced, the computational complexity is reduced while the recognition rate still remains high.

5.1 Hierarchical structure of decision attribute

According to human being experience in the recognition process of musical instruments, it is usually easier for a person to tell the difference between violin and piano than violin and viola. Because violin and piano belong to different instrument families and thus have quite different timbre qualities. Violin and viola fall into the same instrument family which indicates they share the similar timbre quality. If we can build the classifiers both on the family level and the instrument level, the polyphonic music sound is first classified at the instrument family level. After a specific instrument family label is assigned to the analyzed sound by the classifier, it is further classified at the instrument level by another classifier which is built on the training data of that specific instrument family. Since there are fewer instruments in this family, the classifier learned from this family has the expertise of identifying the instruments within this family. Before we discuss how to build classifiers on the different levels, let us first look at the hierarchical structure of the western instruments.

Erich von Hornbostel and Curt Sachs published an extensive scheme for musical instrument classification in 1914. Their scheme is widely used today, and is most often known as the Hornbostel-Sachs system. This system includes aerophones (wind instruments), chordophones (string instruments), idiophones (made of solid, non-stretchable, resonant material), and membranophones (mainly drums). Idiophones and membranophones are together considered as percussion. Additional groups include electrophones, i.e. instruments where the acoustical vibrations are produced by electric or

electronic means (electric guitars, keyboards, synthesizers), complex mechanical instruments (including pianos, organs, and other mechanical music makers), and special instruments (include bullroarers, but they can be classified as free aerophones).

Each category can be further divided into groups, subgroups etc. and finally into instruments. **Idiophones** subcategories include: Struck (concussion), claves, clappers, castanets, and finger cymbals. **Membranophones** include the following different kind of durms: Cylindrical drum, Conical drum, Barrel drum, Hourglass drum, Goblet drum, Footed drum, Long drum, Kettle or pot drum, tambourine, Friction drum. **Chordophones** subcategories include: Zither, mandolins, guitars, ukuleles, Lute (bowed) - viols (fretted neck), fiddles, violin, viola, cello, double bass, and hurdy-gurdy (no frets), Harp. **Aerophones** are classified as single reed (such as clarinet, saxophones), double reed (such as oboe, bassoon) and lip vibrated (trumpet or horn) according to the mouthpiece used to set air in motion to produce sound. Some of Aerophones subcategories are also called woodwinds or brass, but this criterion is not based on the material the instrument is made of, but rather on the method of sound production. In woodwinds, the change of pitch is mainly obtained by the change of the length of the column of the vibrating air. Additionally, over-blow is applied to obtain the second, third or fourth harmonic to become the fundamental. In brass instruments, over-blows are very easy because of wide bell and narrow pipe, and therefore over-blows are the main method of pitch changing. Sounds can be also classified according to the **articulation**. It can be performed in three ways: (1) sustained or non-sustained sounds, (2) muted or not muted sounds, (3) vibrated and not vibrated sounds. This classification may be difficult, since the vibration may not appear in the entire sound; some changes may be visible, but no clear vibration. Also,

brass is sometimes played with moving the mute in and out of the bell. Most of musical instrument sounds of definite pitch have some noises/continuity in their spectra. According to MPEG7 classification [14], there are four classes of musical instrument sounds: (1) Harmonic, sustained, coherent sounds - well detailed in MPEG7, (2) Non-harmonic, sustained, coherent sounds, (3) Percussive, non-sustained sounds - well detailed in MPEG7, (4) Non-coherent, sustained sounds.

Figure 5.1 shows the simplified Hornbostel/Sachs tree. We do not include membranophones here because the instruments of this family usually do not produce the harmonic sound so that they need special techniques to be identified. This dissertation focuses on the harmonic instruments which fall into the other three families.

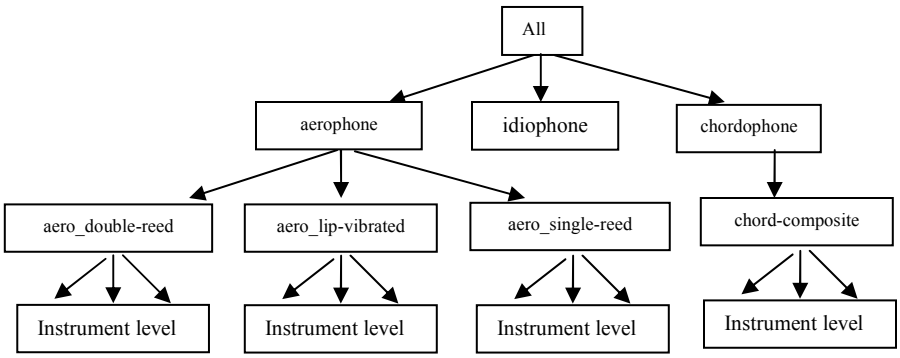

Figure 5.1 the Hornbostel-Sachs hierarchical structure

Figure 5.2 shows us another hierarchical structure of instrument family which is grouped by the way how the musical instruments are played.

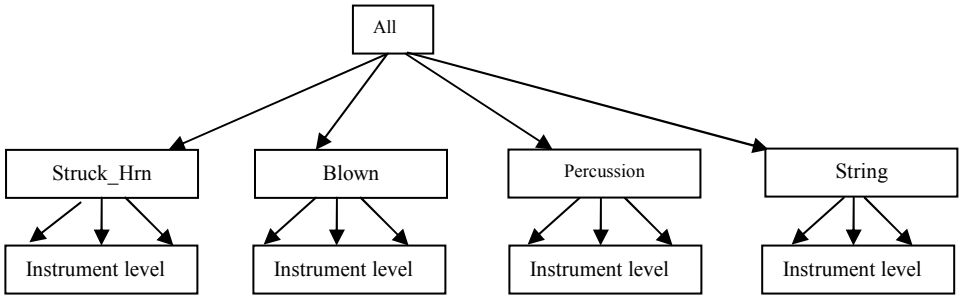

Figure 5.2 the hierarchical structure according to playing method

A hierarchical classifier is usually defined as agglomerative method of classifying inputs into defined output categories [13], [21]. The classification occurs first on a low-level with highly specific pieces of input data. The classifications of the individual pieces of data are then combined systematically and classified on a higher level iteratively until one output is produced. This final output is the overall classification of the data.

Automatic indexing of music by instruments and their types is taken as the application and testing area for our research. In [42], a multi-hierarchical decision system S with a large number of descriptors built for describing music sound objects is described. The decision attributes in S are hierarchical and they include Hornbostel-Sachs classification and classification of instruments with respect to a playing method. The information richness hidden in these descriptors has strong implication on the confidence of classifiers built from S and used as a tool by the content-based Automatic Indexing Systems (AIS). Because decision attributes are hierarchical, the indexing and timbre estimation can be done with respect to different granularity levels of music instrument classes. We can then identify not only the instruments playing in a given music piece but also classes of instruments. In this dissertation we propose a methodology of building cascade classifiers from musical datasets.

5.2 Cascade Hierarchical Decision Systems

In hierarchical decision systems, the initial group of classifiers is trained using all objects in an information system *S* partitioned by values of the decision attribute *d* at all granularity levels (one classifier per level). Only values of the highest granularity level (corresponding granules are the largest) are used to split S into information sub-systems where each one is built by selecting objects in S of the same decision value. These sub-

systems are used for training new classifiers at all granularity levels of its decision attribute. Next, we split each sub-system further by sub-values of its decision value. The obtained tree-type structure with groups of classifiers assigned to each of its nodes is called a cascade classifier.

Let $S(d) = (X, A \cup \{d\}, V)$ be a decision system, where X is a set of unknown musical objects, A is the set of features used as classification attributes, d is a hierarchical decision attribute and V is a set of decision values. Figure 5.3 shows an example of a hierarchical decision attribute.

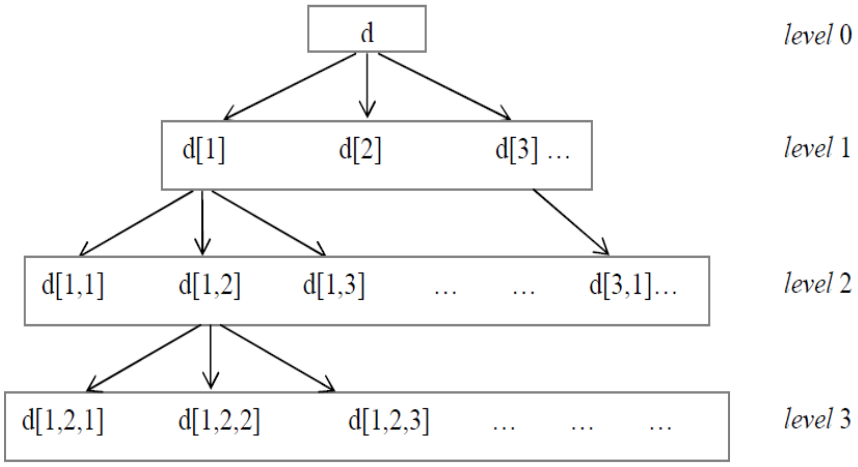

Figure 5.3 Hierarchical decision attributes

Let $\{d[1], d[2], \ldots, d[k]\}$ is a set of all values of the attribute d at level 1 of its tree representation. Let $X_i = \{x \in X : d(x) = d[i]\}$ and $S_i[d_i] = (X_i, A \cup \{d[i]\}, V\}$, for any $1 \le i \le n$. Now, assume that *CR(S)* denotes a tree of height one. System *S* is its root and $S_i(d[i])$, $(1 \le i \le n)$, are its children. The outgoing edge from *S* to $S_i(d[i])$ is labeled by $d[i]$, for any $1 \le i \le n$.

Cascade representation of $S(d)$ is a tree with $S(d)$ defined as its root and all its descendants being built by executing the instruction [if $card(V_d) > 1$, then replace $S(d)$ by

$CR(S(d))$)] recursively, starting from the root and then repeating for all leaves of a constructed tree.

Table 5.1 Example of hierarchical decision attributes

	a	b	c	d
x1		1	2	d[1,1]
x2		1	3	d[1,1]
x3	1	1	0	d[1,2]
x4	1	1	3	d[1,2]
x5	2		2	d[2,1]
x6	2		3	d[2,1]
x7		1	1	d[1,1]
x8		1	1	d[1,1]
x9	2		1	d[2,1]
x10	2		0	d[2,1]
x11	1	1	2	d[2,2]
x12	1	1	1	d[2,2]

Let us look at the example of a decision system $S(d)$ represented as Table 5.1. Its attributes are $\{a,b,c\}$. d is the decision attribute. To build a cascade representation of $S(d)$, we take its subsystems:

$$S^*(d) = (\{x_i : 1 \le i \le 12\}, \{a,b,c,d\}, V),$$
$$S_{[1]}(d[1]) = (\{x_i : i = 1,2,3,4,7,8\}, \{a,b,c,d\}, V),$$
$$S_{[2]}(d[2]) = (\{x_i : i = 5,6,9,10,11,12\}, \{a,b,c,d\}, V),$$
$$S_{[1,1]}(d[1,1]) = (\{x_i : i = 1,2,7,8\}, \{a,b,c,d\}, V),$$
$$S_{[1,2]}(d[1,2]) = (\{x_i : i = 3,4\}, \{a,b,c,d\}, V),$$
$$S_{[2,1]}(d[2,1]) = (\{x_i : i = 5,6,9,10\}, \{a,b,c,d\}, V),$$
$$S_{[2,2]}(d[2,2]) = (\{x_i : i = 11,12\}, \{a,b,c,d\}, V).$$

Now, the corresponding the cascade representation of *S(d)*, denoted as $(\{S^*(d)\} \cup \{S_k(d) : k \in J\}, \prec)$, where $J = \{[1],[2],[1,1],[1,2],[2,1],[2,2]\}$ and " $\prec$ " means parent-child relation, is represented in Figure 5.4.

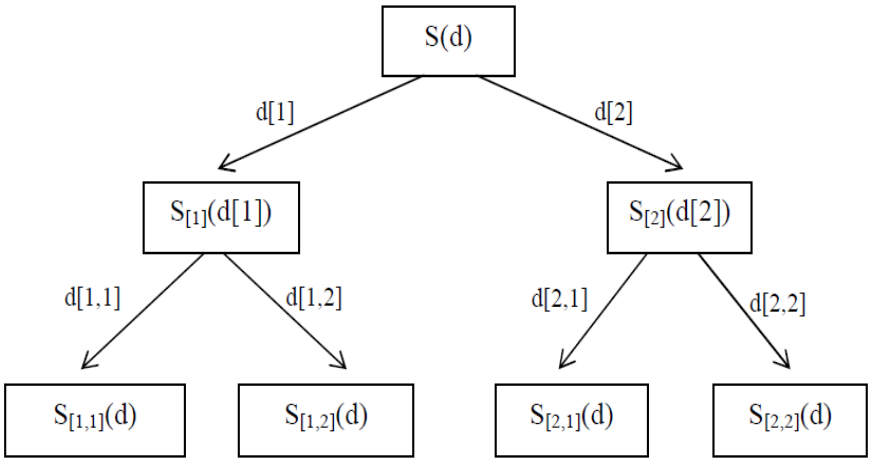

Figure 5.4 Cascade representation of *S*(*d*)

The partition of objects in S(d) can be driven by an optimization function or it can be predefined, as it is done in MIRAI [30], by following either Hornbostel-Sachs classification or classification of instruments with respect to a playing method.

5.3 Cascade Classifiers

Let $S(d) = (X, F \cup \{d\})$ be a decision system, where *d* is a hierarchical attribute. We follow the notation of the previous section to represent its values, with *d*[*i*] referring to a child of *d* and *d*[*i*,*j*] to its grandchild. $F = \{f_1, \ldots, f_m\}$ is all the available features which are extracted from the input signal and then used by the classifiers respectively to identify the analyzed frame. $X = \{x_1, \ldots, x_t\}$ is all the segmented frames from the analyzed audio sound. $Casc(S(d)) = \{S_k(d) : k \in J\}$ is a cascade representation of *S(d),* where *J* is all the nodes of hierarchical tree, such as [1], [1,1],[1,2] and so on. A sample representation structure for a cascade classifier is given in Figure 5.5.

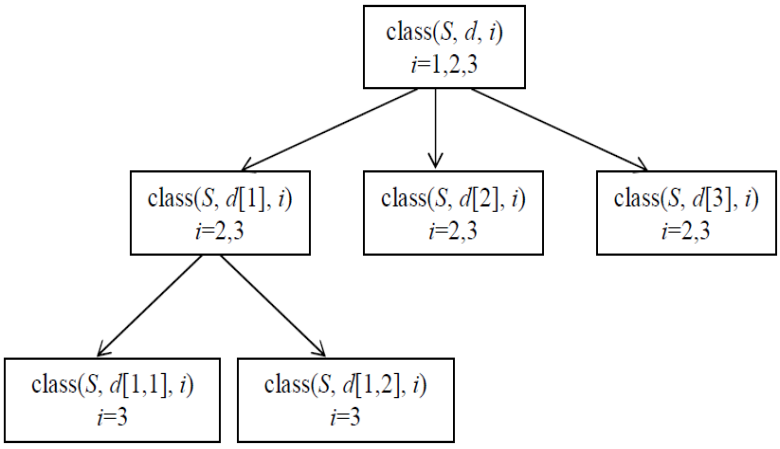

Figure 5.5 Cascade classifier for S(d)

In this sample, three classifiers are associated with the root level of the tree represented by Figure 5.5. The first one (with *i=1*) is trained by *S* with values of the decision attribute defined as the largest granules. The last one (with *i=3*) is based on attribute values defined as the smallest granules. For each frame x_i, the whole process is started by the classification at the root of hierarchical tree and followed by the classification at the other lower level of the tree. The system selects the appropriate classifier $class(S_{[k]}, d, i)$ and feature $f(S_{[k]}, d, i)$ to perform classification at each possible level $S_{[k]}$ from the top to the bottom. The confidence of classification at each level is $conf(x_i, S_{[k]})$, where the confidence has to satisfy the minimum confidence of the correct classification λ_1. After the classification process reaches the bottom level, which is the instrument level, we have the final instrument estimations $\{d_p\}$ for the frame x_i, and the overall confidence for each instrument estimation is calculated by multiplying the confidence obtained at each node $conf(x_i, d_p) = \prod conf(x_i, S_{[k]})$. After

all the individual frames are estimated by the classification system, a smoothing process is performed by calculating the average confidence of each possible instrument within the indexing window $Conf(d_p) = \sum_{i \in w} conf(x_i, d_p) \Big/ s$ where w is the frame range of indexing window. The final result for the indexing window also needs to satisfy the confidence threshold λ_2. According to the indexing resolution requirement, the indexing window can be adjusted to the desired size. Figure 5.6 shows the MIR framework based on the cascade hierarchical classification system.

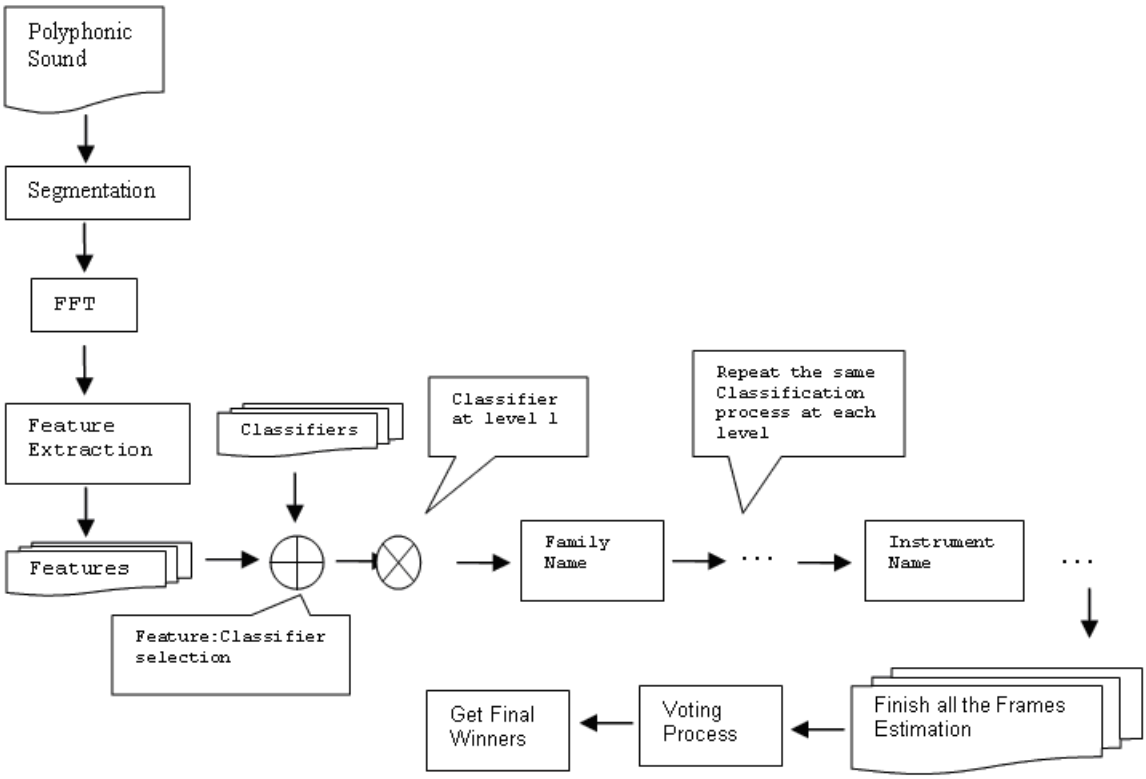

Figure 5.6 Timbre estimation framework based on the cascade hierarchical classification system with classifier and feature selection

5.4 Experiments and results based on all features

We build a hierarchical decision system S with all the acoustic features described in chapter two for describing music sound objects. The decision attributes in S are hierarchical and they represent Hornbostel-Sachs classification. The information richness hidden in descriptors has strong implication on the confidence of classifiers built from the decision system S. Hierarchical decision attributes allow us to have the indexing done on different granularity levels of classes of musical instruments. We can identify not only the instruments playing in a given music piece but also classes of instruments if the instrument level identification fails. The quality of AIS can be verified using precision and recall based on two interpretations: user and system-based [31]. AIS engine follows system-based interpretation. We show that cascade classifiers outperform standard classifiers. Table 5.2 is the cascade classifier for Hombostel-Sachs classification of instruments and their confidence

Table 5.2 Cascade classifier for S(d)

root	classname	classifier	support	confidence
d	All_instruments	Class(S,d,2)	771	96.97%
d	All_instruments	Class(S,d,1)	764	96.02%
d	All_instruments	Class(S,d,3)	730	91.80%
d[1]	Aerophone	Class(S,d[1],2)	269	98.26%
d[1]	Aerophone	Class(S,d[1],3)	265	96.84%
d[2]	Chordophone	Class(S,d[2],2)	497	98.83%
d[2]	Chordophone	Class(S,d[2],3)	466	92.75%
d[3]	Idiophone	Class(S,d[3],2)	19	95.95%
d[3]	Idiophone	Class(S,d[3],3)	19	95.95%
d[1,1]	Aero_double_reed	Class(S,d[1,1],3)	70	98.94%
d[1,2]	Aero_lip_reed	Class(S,d[1,2],3)	113	95.66%
d[1,3]	Aero_side	Class(S,d[1,3],3)	10	90.91%
d[1,4]	Aero_single_reed	Class(S,d[1,4],3)	72	99.54%
d[2,1]	Chord_composite	Class(S,d[2,1],3)	410	93.18%

The testing was done for musical instrument sounds of pitch 3B. The results in Table 5.2 show the confidence of the classifiers trained on different subsets which correspond

to the different nodes in the hierarchical tree. The decision attributes of these classifiers are at the instrument level.

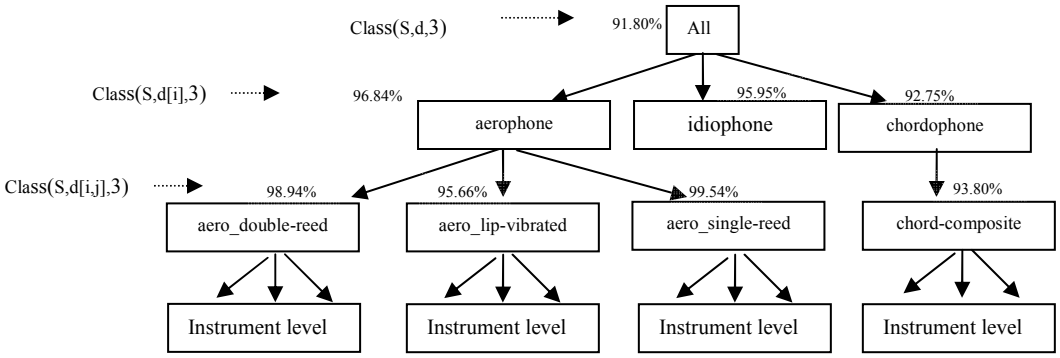

Figure 5.7 Cascade classifier for Hornbostel/Sachs classification of instruments and their accuracy

Figure 5.7 shows the confidence of the classifiers trained in the whole dataset (the largest granule). These classifiers have the decision attribute at the different hierarchical levels which correspond to each node of the tree. The confidence of a standard classifier class(S, d, 3) for Hombostel-Sachs classification of instruments is 91.50%. However, we get better results by following the cascade approach. When we use the classifier class(S, d, 2) followed by the classifier class(S, d[1, 1], 3), the precision in recognizing musical instruments in "aero double reed" class is equal to 96.02% * 98.94% = 95.00%. Also, its confidence in recognizing instruments in "aero single reed" class is equal to 96.02% * 99.54% = 95.57%. It has to be noted that this improvement in classification confidence is obtained without increasing the number of attributes in the subsystems of S.

Again, from the Table 5.2 and Figure 5.7, when we compare different classifiers which are built in the same training dataset but on the different levels decision attribute, we find that generic classifiers usually have the higher confidence than the peculiar one (Figure 5.8).

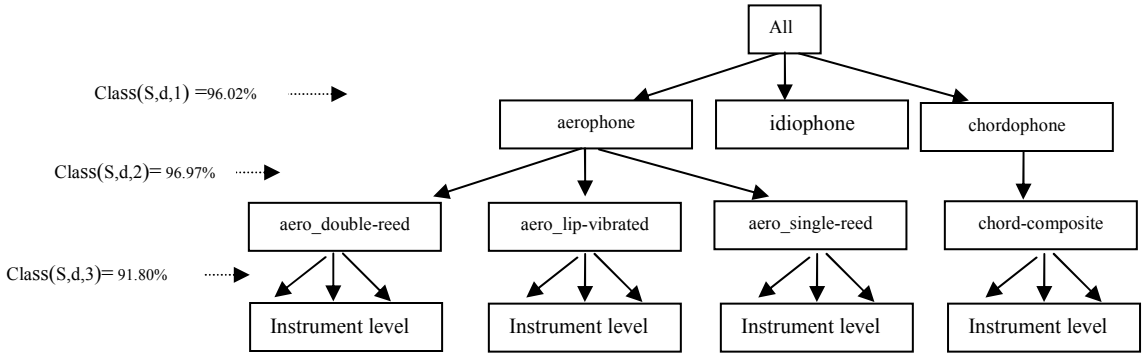

Figure 5.8 The confidence of classifiers built on different levels of decision attribute (pitch 3B)

Following this strategy, we get high classification confidence for single instrument estimation in comparison to a regular non-cascade classification approach.

5.5 Classifier selection based on different features

Cascade classification system allows different classifiers and different features to be used at different levels of the hierarchical structure. In order to investigate how the classifier and feature selection affect the cascade system, two experiments of classification based on the KNN and Decision Tree were conducted: 1) classification with each feature group; 2) classification with the combination of different feature groups. The training dataset of middle C includes 26 different instruments: Electric Guitar, Bassoon, Oboe, B-flat clarinet, Marimba, C-Trumpet, E-flat Clarinet, Tenor Trombone, French horn, Flute, Viola, Violin, English horn, Vibraphone, Accordion, Electric Bass, Cello, Tenor saxophone, B-Flat Trumpet, Bass flute, Double bass, Alto flute, Piano, Bach trumpet, Tuba, and Bass Clarinet. These instruments cover the typical western musical instruments families which are played by the orchestra. There are 2762 frames extracted from those instrument sound objects. We try to test different features with different classifiers to get the optimal pair of them.

5.5.1 Classification on each feature group

In experiment 1, we divide the features into the following 5 groups (table 5.3).

Table 5.3 Feature group

Group	Feature description
A	33 Flatness coefficients
B	13 MFCC coefficients
C	28 Harmonic Peaks
D	38 Spectrum projection coefficients
E	Log spectral centroid, spread, flux, rolloff, zerocrossing

Among the groups A to D, each represents one single feature vector of multiple numeric values. Group E includes all the statistical single-value features. Classifiers of KNN and Decision Tree from Weka are applied to the dataset with each feature group. The same parameter settings are applied as chapter 2.4. Confidence is defined as the ratio of the correct classified instances over the total number of instances.

Table 5.4 Classification of each feature group

Feature Group	Classifier	Confidence (%)
A	KNN	99.23%
	Decision Tree	94.69%
B	KNN	98.19%
	Decision Tree	93.57%
C	KNN	86.60%
	Decision Tree	91.29%
D	KNN	47.45%
	Decision Tree	31.81%
E	KNN	99.34%
	Decision Tree	99.77%

The result in Table 5.4 shows that some features work better with KNN than decision tree, such as Flatness coefficients (Group A), MFCC (Group B), and spectrum projection coefficients (Group D), Decision tree works better with harmonic peaks (Group C). The statistical features (Group E) show little difference between the two classifiers.

5.5.2 Classification on the combination of different feature groups

In order to further explore the relationship between feature groups and classifiers, we merge every two feature groups into larger feature groups and test them with different classifiers. Figure 5.9 shows the result of KNN classification.

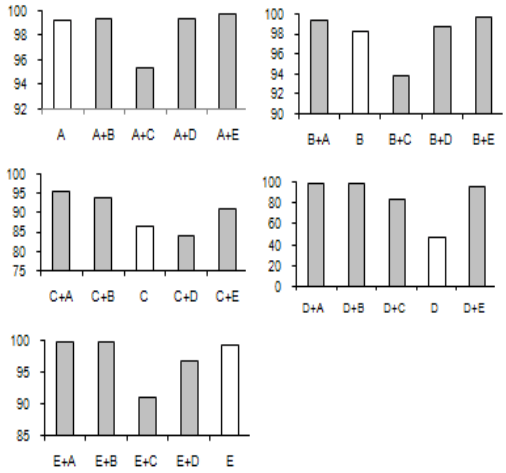

Figure 5.9 Classification based on KNN in experiment 2

The confidence of KNN changes minimally when more features are added. And when Group C (harmonic-Peaks) is added to Group A, B, D, and E, the classification results deteriorate. This result further validates the conclusion from experiment 1 that harmonic peaks do not fit KNN classifier well.

Figure 5.10 shows the result of decision tree classification. We observe that group E improves other feature groups when it is added. However those results do not improve much compared to the classification result of single Group E. It means Group E yields the best result for decision tree classifier.

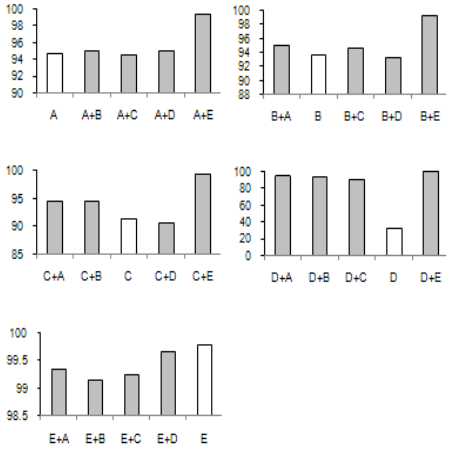

Figure 5.10 Classification of decision tree in experiment 2

From those two experiments, we see that the KNN classifier works better with feature vectors such as spectral flatness coefficients, projection coefficients and MFCC. Decision tree works better with harmonic peaks and statistical features. Simply adding more features together does not improve the classifiers and sometime even worsens the classification results (such as adding harmonic to other feature groups). In cascade system, it is a non-trivial task to perform feature selection for different classifiers to optimize the timbre estimation.

5.6 Feature and classifier selection at each level of cascade system

According to the previous discussion and conclusion, cascade system has to select the appropriate feature and classifier to achieve the best estimation result at each level of cascade classification. We test four feature groups (A, B, C, D) with three different classifiers (NaiveBayes, KNN, Decision Tree). From the classification results, we try to learn how to perform feature selection and classifier selection based on the information hidden in the current training database. We use the same algorithms from Weka for KNN and decision tree classifiers as previous section. NaiveBayes classifier [11] is added to this experiment.

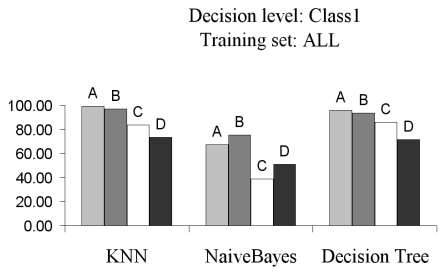

Figure 5.10 Feature and classifier selection at top level

According to the results shown in Figure 5.11, at the top level of hornbostel/Sachs hierarchical tree (decision attribute is on class1 level) KNN classifier with feature A yields the best estimation confidence. At the beginning the system should use flatness coefficients and KNN to discover the family that the analyzed sound object belongs to. In order to perform the further estimation on the lower level of the instrument family, the system also needs to know how to select the feature and classifier at that particular level.

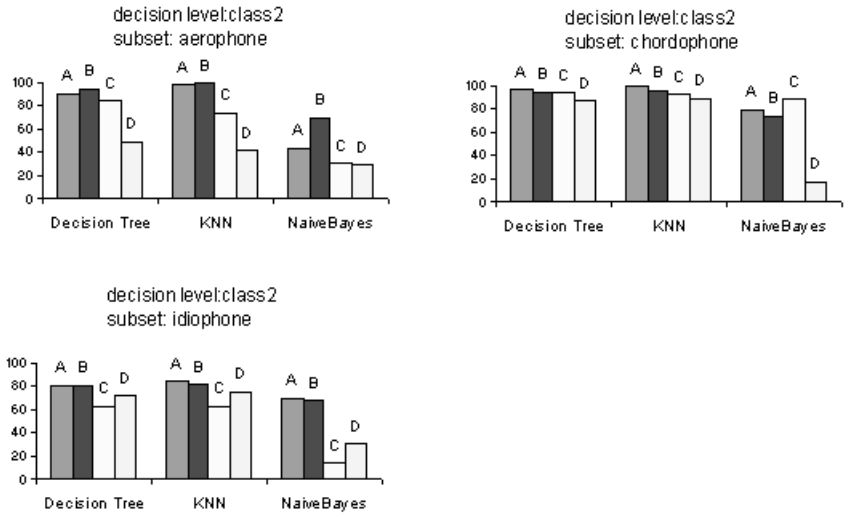

Figure 5.12 Feature and classifier selection at second level

Figure 5.12 tells us that KNN classifier and feature A (Flatness coefficients) are still the best choice for the instruments falling in the families Chordophone or Idiophone. If

the analyzed sound object is estimated as Aerophone, feature B (MFCC) is the better choice than others. Table 5.5 concludes the feature and classifier selection more clearly.

Table 5.5 Feature and classifier selection table for Level1

Node	feature	Classifier
chordophone	Flatness coefficients	KNN
aerophone	MFCC coefficients	KNN
idiophone	Flatness coefficients	KNN

We continue to perform the classification on the different subsets at the third level of Hornbostel-Sachs hierarchical tree and get the classification results shown in Figure 5.13.

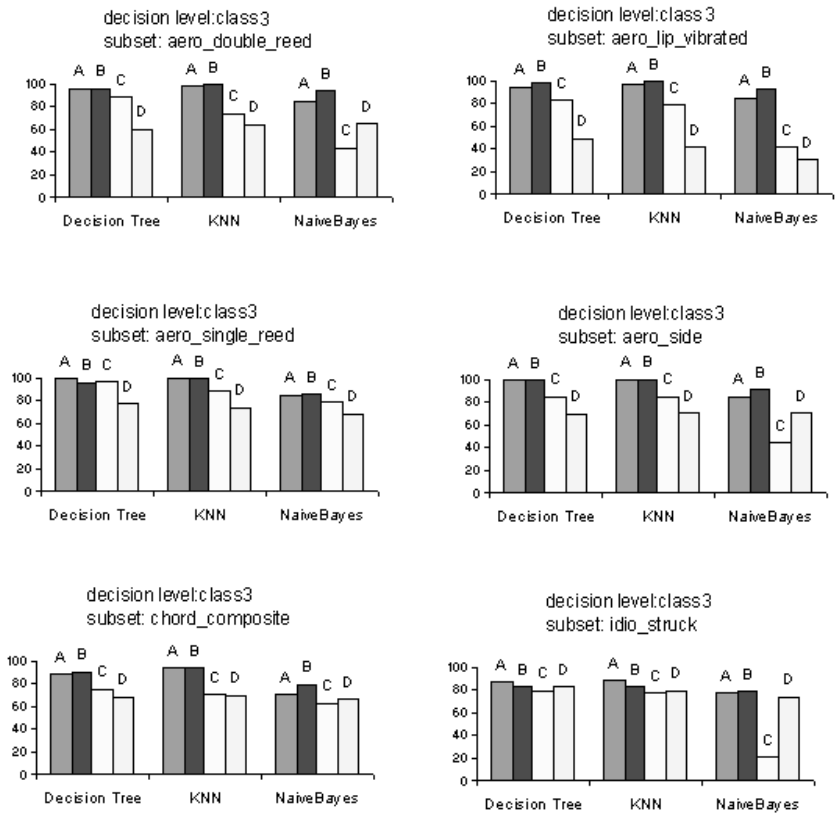

Figure 5.13 Feature and classifier selection at third level

The instrument name is eventually estimated by the classifiers at the bottom level. Table 5.6 shows the classifier and feature selection results from the classification

experiments at second level of the hierarchical tree. A notable result is that the subset of "Aero_single_reed" does not inherit the same character from the parent node of "Aerophone". Instead of selecting Feature B(MFCC) and KNN, the decision tree along with Feature A(Flatness coefficients) yields the better classification result.

Table 5.6 Feature and classifier selection table for Level2

Node	feature	Classifier
chrd_composite	Flatness coefficients	KNN
aero_double-reed	MFCC coefficients	KNN
aero_lip-vibrated	MFCC coefficients	KNN
aero_side	MFCC coefficients	KNN
aero_single-reed	Flatness coefficients	Decision Tree
idio_struck	Flatness coefficients	KNN

According to the above knowledge derived from the training database, we can optimize the feature selection and classifier selection at each level of hierarchical tree and further improve the overall estimation result for cascade classification system.We implement the proposed cascade classification system and test I on the polyphonic sounds. The following chapter will discuss additional details about the evaluation results of both the cascade classification system and non-cascade classification system.

CHAPTER 6: HIERARCHICAL STRUCTURE BUILT BY CLUSTERING ANALYSIS

Clustering is the method that divides data objects into similarity groups (clusters) according to a defined distance measure. Clustering is widely used as an important technique of machine learning and pattern recognition in the fields of biology, genomics and image analysis. However it has not been well investigated in the music domain since the category information of musical instruments has already been defined by musicians as the two hierarchical structures demonstrated in the last chapter. Those structures group the musical instruments according to their semantic similarity which is concluded from the human experience. However the instruments that are assigned to the same family or subfamily by those hierarchical structures often sound quite different from another. On the other hand, instruments that have very similar timbre qualities can be assigned to very different groups by those hierarchical structures. The inconsistency between the timbre quality and the family information causes the incorrect timbre estimation of cascade classification system. For instance, the trombone belongs to the aerophone family, but the system often classifies it as the chordophone instruments, such as violin. In order to take full advantage of the cascade classification strategy, we build the new hierarchical structure of musical instruments by the matching learning technique. Cluster analysis is commonly used to search for groups in data. This is most effective when the groups are

not already known. We use the cluster analysis methods to reorganize the instrument group according to the similarity of timbre relevant features among the instruments.

6.1 Clustering analysis methods

There are many clustering algorithms available. Basically all the clustering algorithms can be divided into two categories: partitional clustering and hierarchical clustering.

Partitional clustering algorithms determine all clusters at once without hierarchically merging or dividing process**.** K-means [23] clustering is the most common method in this category with K serving as the empirical parameter. Instances are randomly assigns to the k clusters, then the new centroid for each of the k clusters and the distance of all items to the new k centroids are calculated. Items are reassigned to the closest new centroid and the whole process is repeated until cluster assignments are stable.

Hierarchical clustering generates a hierarchical structure of clusters which may be represented in a structure called a dendrogram. The root of the dendrogram consists of a single cluster containing all the instances, and the leaves correspond to individual instances. Hierarchical clustering can be further divided into two types according whether the tree structure is constructed by agglomerative way or divisive way. Agglomerative approach works in the bottom-up manner, which first groups the instances into small clusters and merges those small ones into bigger clusters. Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters, which is a top-down approach.

We choose the hierarchical clustering method to learn the new hierarchical schema for instruments since it fits in this scenario well. There are different ways to interpret the distance between two clusters in the agglomerative clustering analysis when it performs

the cluster merging at each hierarchical level. The following tree rules are the most common method to calculate the distance or similarity between clusters [36].

Single linkage (nearest neighbor). In this method, the distance between two clusters is determined by the distance of the two closest objects (nearest neighbors) in the different clusters. This rule will string objects together to form clusters, and the resulting clusters tend to represent long "chains".

Complete linkage (furthest neighbor). In this method, the distance between clusters is determined by the greatest distance between any two objects in the different clusters (the "furthest neighbors"). This method usually performs quite well in cases when the objects actually form naturally distinct "clumps." If the clusters tend to be of a "chain" type nature, then this method is inappropriate.

Unweighted pair-group method using arithmetic averages (UPGMA**).** In this method, the distance between two clusters is calculated as the average distance between all pairs of objects in two different clusters. This method is also very efficient when the objects form natural distinct "clumps" and it performs equally well with "chain" type clusters.

Weighted pair-group method using arithmetic averages (WPGMA). This method is identical to the UPGMA method, except that the sizes of the respective clusters are used as the weights. Thus, this method should be used when the cluster sizes are suspected to be greatly uneven [35].

Unweighted pair-group method using the centroid average (UPGMC). The centroid of a cluster is the average point in the multidimensional space defined by the

dimensions. In a sense, it is the center of gravity for the respective cluster. In this method, the distance between two clusters is determined as the difference between centroids.

Weighted pair-group method using the centroid average (WPGMC). This method is identical to the previous one, except that weighting is introduced into the computations to take into consideration differences in cluster sizes. When there are considerable differences in cluster sizes, this method is preferable to the previous one.

Ward's method. This method is distinct from all other methods because it uses an analysis of variance approach to evaluate the distances between clusters. In short, this method attempts to minimize the Sum of Squares of any two hypothetical clusters that can be formed at each step. In general, this method is good at finding compact, spherical clusters. However it tends to create clusters of small size.

Distance measures

In both agglomerative and divisive approaches, a particular distance measure needs to be defined in order to calculate the similarity or dissimilarity among individual instances or centroids of clusters. It is critical to choose the appropriate measure for the musical data because different measures may produce different shapes of clusters which represent different schema of instrument family. Different features also require the appropriate measures to be chosen in order to give better description of feature variation. The inappropriate measure could distort the characteristics of timbre which may cause the incorrect clustering. Here are some most common distance functions:

Euclidean is the usual square distance between the two vectors (2 norms). Disadvantages of this distance include not being scale invariant and not good for negative correlations

$$d_{XY} = \sqrt{\sum (X_i - Y_i)^2}$$

Manhattan is the absolute distance between the two vectors.

$$d_{XY} = \sum |X_i - Y_i|$$

Maximum is the maximum distance between two components of *x* and *y*

$$d_{XY} = \max(|X_i - Y_i|)$$

Canberra Canberra distance examines the sum of series of a fraction differences between coordinates of a pair of objects. Each term of fraction difference has value between 0 and 1. If one of coordinates is zero the term corresponding to this coordinate becomes unity regardless the other value, thus the distance will not be affected. If both coordinates are zero, then the term is defined as zero.

$$d_{XY} = \sum \frac{|X_i - Y_i|}{|X_i| + |Y_i|}$$

Pearson correlation coefficient (PCC) is a Correlation-based distance. It measures the degree of association between two variables.

$$\rho_{XY} = \frac{[Cov(X,Y)]^2}{\text{var}(X)\,\text{var}(Y)} \qquad d_{XY} = 1 - \rho_{XY}$$

where $Cov(X,Y)$ is the covariance of the two variables, $\text{var}(X)$ and $\text{var}(Y)$ are the variances of variables.

Spearman's rank correlation coefficient is another correlation based distance**.**

$$\rho_{XY} = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \qquad d_{XY} = 1 - \rho_{XY}$$

Where $d_i = x_i - y_i$, the difference between the ranks of corresponding values x_i and y_i, and n is the number of values in each data set (same for both sets). Raw data are converted to

rankings xi and yi according to the order of the raw data.. It is done by assigning 1 to the smallest element of each data, and 2 to second smallest element and so on. The average ranking is calculated if there is tie among different elements.

6.2 Evaluation of different Clustering algorithms for different features

As we can see, each clustering method has its own different advantages and disadvantages over others. Deciding which one is the most appropriate method for generating the hierarchical instrument structure is a non-trivial task. The specific cluster linkage method needs to be decided in the hierarchical clustering algorithms, along with the selection of the good distance measurement in order to generate the good schema that represents the actual relationships among those instruments. We design the intensive experiments with the "cluster" package in R system [32]. The R packge provides the two hierarchical clustering algorithms: hclust (Agglomerative hierarchical clustering), diana (divisive hierarchical clustering). Table 6.1 shows all the clustering methods that we test. We evaluate the six different distance measurements (Euclidean, Manhattan, Maximum, Canberra, Pearson correlation coefficient, Spearman's rank correlation coefficient) for each algorithm. For the agglomerative type of clustering (hclust) algorithm, we also evaluate seven different cluster linkages that are available in this package: Ward, single(Single linkage), complete(complete linkage), average (UPGMA), mcquitty(WPGMC), median, centroid (UPGMC).

Table 6.1 All distance measures and linkage methods tested for agglomerative and divisive clustering

Clustering algorithm	Cluster Linkage	Distance Measure
Hclust(Agglomerative)	average	6 distance metrics
	centroid	6 distance metrics
	complete	6 distance metrics
	mcquitty	6 distance metrics

Table 6.1 (continued)

	median	6 distance metrics
	single	6 distance metrics
	ward	6 distance metrics
diana (Divisive)	N/A	6 distance metrics

We choose the middle C pitch group which contains 46 different musical sound objects. And we also extract three different features (MFCC, flatness coefficients, and harmonic peaks) from those sound objects. Each feature produces one dataset for clustering. Some sound objects belong to the same instrument. For example, "ctrumpet" and "ctrumpet harmonStemOut" are objects produced by the same instrument: trumpet. We preserve these particular object labels in our feature database without merging them and giving them the same label because they could have very different timbre qualities which the conventional hierarchical structure ignores. We try to discover the unknown musical instrument group information solely by the unsupervised machine learning algorithm instead of adding any human interpolation. Each sound object is segmented into multiple 0.12s frames and each frame store as an instance in the testing dataset. Thus there are totally 2884 frames from the 46 objects in each of the three feature datasets.

When the algorithm finishes the clustering process, a particular cluster ID is assigned to each single frame. Theoretically the same cluster ID is assigned to all the frames of the same instrument sound object. However, the frames from the same sound object are not uniform and have variations in their feature patterns when the time evolves. Clustering algorithms do not perfectly identify them as the same cluster, instead some frames are clustered into other groups where a majority of the frames come from other instrument sounds. Multiple different cluster IDs are then assigned to the frames of the same instrument object. Our goal is to cluster the different instruments into the groups

according to the similarity of timbre relevant features. Therefore one important step of evaluation is to check if one clustering algorithm is able to cluster most frames of the individual instrument sound into one group. In other words, the clustering algorithm needs to tell the frames of one instrument sound from others. It is evaluated by calculating the accuracy of the cluster ID assignment. We use the following example to illustrate the evaluation process. The hierarchical cluster tree T_m is produced by one clustering algorithm A_m. There are totally n instrument sound objects in the dataset. The clustering package provides function *cutree* to cut T_m into n clusters. One of these clusters is assigned to each frame. Table 6.2 is the contingency table derived from the clustering results after the *cutree* is applied. It is a $n \times n$ matrix, where each element X_{ij} is the number of the frames of instrument i that are labeled by cluster j, and $X_{ij} >= 0$.

Table 6.2 Contingency Table derived from clustering result

	Cluster *1*	…	Cluster *j*	…	Cluster *n*
Instrument *1*	X_{11}	…	X_{1j}	…	X_{1n}
…	…	…	…	…	…
Instrument *i*	X_{i1}	…	X_{ij}	…	X_{in}
…	…	…	…	…	…
Instrument *n*	X_{n1}	…	X_{nj}	…	X_{nn}

In order to calculate the accuracy of the cluster assignment, we need to decide which cluster ID corresponds to which instrument object. For instance, if instrument i is assigned to cluster k, X_{ik} is the number of correct assignments for instrument i, accuracy

of the clustering for instrument *i* is $\alpha_i = X_{ik} \Big/ \sum_{j=1}^{n} X_{ij}$. The overall accuracy for the clustering algorithm A_m is the average accuracy of all the instruments $\bar{\alpha} = \sum_{i=1}^{n} \alpha_i \Big/ n$. To find the maximum $\bar{\alpha}$ among all the possible cluster assignments to instruments, we have to permute this matrix. It is not applicable to perform such a number of calculations. So we choose maximum X_{ij} of each row to approximate the optimal $\bar{\alpha}$. However it causes the possibility of assigning the same cluster to multiple instruments; therefore we take the number of clusters into account as well as accuracy. The final measurements to evaluate the performance of clustering is $score_m = \overline{\alpha_m} \times w$, w is the number of clusters, $w<=n$. This measure reflects how well the algorithm clusters the frames from the same instrument object into the same cluster. It also reflects the ability of algorithm to separate instrument objects from each other. Table 6.3 gives the 14 results which yields the highest score among 126 experiments based on hclust algorithm.

Table 6.3 Evaluation result of Hclust algorithm

Feature	method	metric	$\bar{\alpha}$	w	score
Flatness Coefficients	ward	pearson	87.3%	37	32.30
Flatness Coefficients	ward	euclidean	85.8%	37	31.74
Flatness Coefficients	ward	manhattan	85.6%	36	30.83
mfcc	ward	kendall	81.0%	36	29.18
mfcc	ward	pearson	83.0%	35	29.05
Flatness Coefficients	ward	kendall	82.9%	35	29.03
mfcc	ward	euclidean	80.5%	35	28.17
mfcc	ward	manhattan	80.1%	35	28.04
mfcc	ward	spearman	81.3%	34	27.63
Flatness Coefficients	ward	spearman	83.7%	33	27.62
Flatness Coefficients	ward	maximum	86.1%	32	27.56
mfcc	ward	maximum	79.8%	34	27.12
Flatness Coefficients	mcquitty	euclidean	88.9%	30	26.67
mfcc	average	manhattan	87.3%	30	26.20

From the results, the ward linkage outperforms other methods and it yields the best performance when Pearson distance measure is used on the flatness coefficients feature dataset. Table 6.4 also shows the results from Diana algorithm. In this algorithm, Euclidean yields the highest score on the mfcc feature dataset.

Table 6.4 Evaluation result of Diana algorithm

Feature	metric	$\overline{\alpha}$	w	score
Flatness Coefficients	euclidean	77.3%	24	18.55
Flatness Coefficients	kendall	75.7%	23	17.40
Flatness Coefficients	manhattan	76.8%	25	19.20
Flatness Coefficients	maximum	80.3%	23	18.47
Flatness Coefficients	pearson	79.9%	26	20.77
mfcc	euclidean	78.5%	29	22.78
mfcc	kendall	77.2%	27	20.84
mfcc	manhattan	77.7%	26	20.21
mfcc	pearson	83.4%	25	20.86
mfcc	spearman	81.2%	24	19.48

When we compare the two algorithms, hclust yields better clustering results than Diana. We choose agglomerative clustering algorithm to generate the hierarchical schema for musical instruments. Ward is used as the linkage method. Pearson distance measure is selected as the distance metric. Flatness coefficient is used as the feature dataset to perform clustering analysis.

6.3 New hierarchical tree

Figure 6.1 is the dendrogram result generated by the selected clustering algorithm. From this new hierarchical tree, we discover some instrument relationships which are not represented in the traditional schemas. Some instrument can produce the sounds with quite different timbre qualities when different playing techniques are applied. The most common technique is muting. A mute is a device fitted to a musical instrument to alter the sound produced. It usually reduces the volume of the sound as well as affects the

timbre. There are several different mute types for different instruments. The most common type is the straight mute, a hollow, cone-shaped mute that fits into the bell of the instrument. This results in a more metallic, sometimes nasal sound, and when played at loud volumes can result in a very piercing note. The second common brass mute is the cup mute. Cup mutes are similar to straight mutes, but attached to the end of the mute's cone is a large lip that forms a cup over the bell. The result is removal of the upper and lower frequencies and a rounder, more muffled tone. On string instruments of the violin family, the mute takes the form of a device attached to the bridge of the instrument, dampening vibrations and resulting in a "softer" sound.

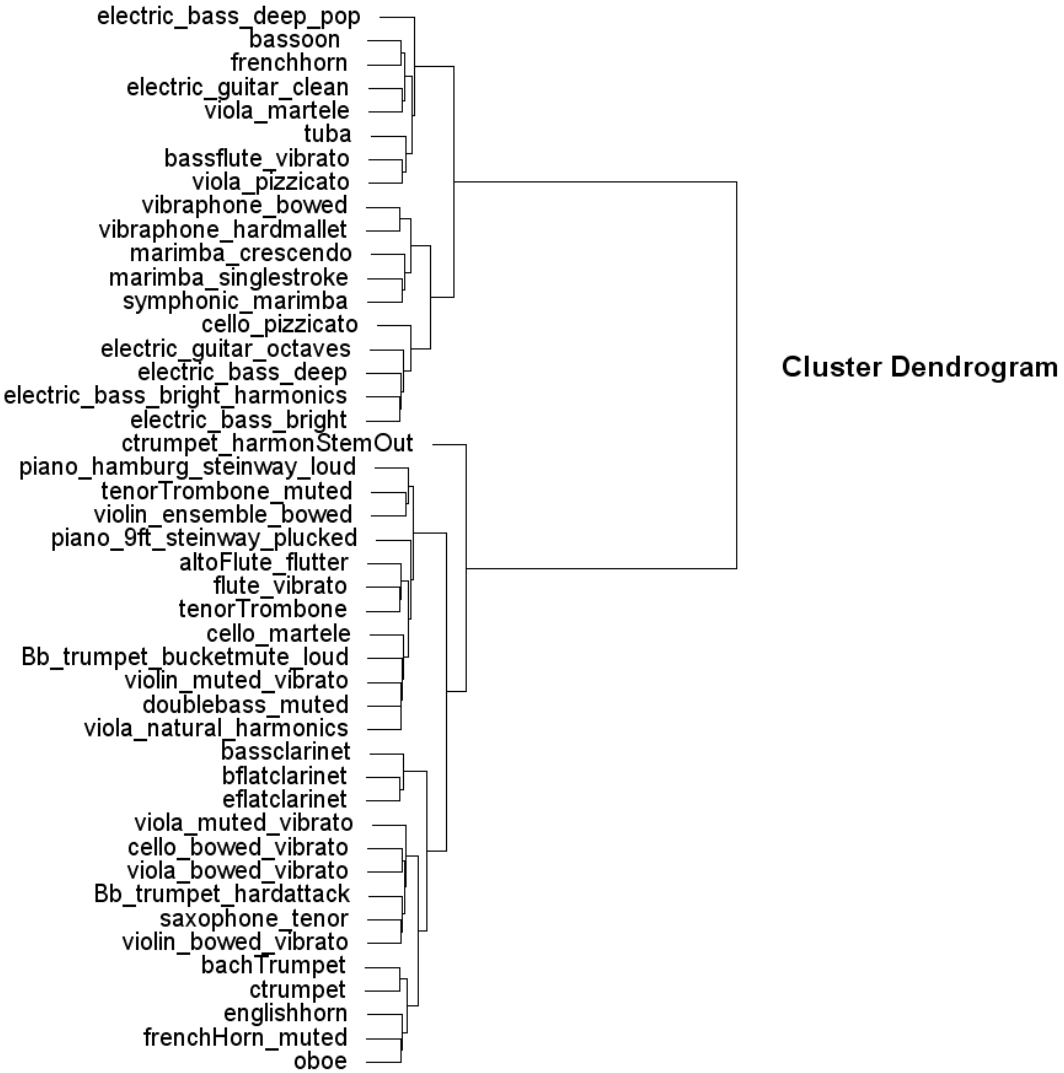

Figure 6.1 Clustering result from Hclust algorithm with Ward linkage method and Pearson as the distance measure; Flatness coefficients are used as the selected feature.

In this hierarchical structure, “ctrumpet” and “ctrumpet_harmonStemOut” are two instrument sounds produced by the trumpet. “ctrumpet_harmonStemOut” is produced when a particular mute is applied to the trumpet. This mute is called Harmon mute, which is different from the common straight or cup mutes. It is a hollow, bulbous metal device placed in the bell of the trumpet. All air is forced through the middle of the mute. This gives the mute a nasal quality. At one end of the device, there is a detachable stem extending through the centre of the mute. The stem can be removed completely or can be

inserted to varying degrees. From the name of this instrument sound object, we know that the stem is extended or completely removed, which darken the original piercing, strident timbre quality. From the spectra of those two sound objects (Figure 6.2), we clearly observe the big difference between them. The spectra also show that “batch trumpet” has more similar spectral pattern to “ctrumpet”. The relationships among those three instrument objects are accurately represented in the new hierarchical schema. Figure 6.1 shows that “ctrumpet” and “batchtrumpet” are clustered in the same group. “ctrumpet_harmonStemOut” is clustered in one single group instead of merging with “ctrumpet” since it has a very unique spectral pattern.

The new schema also discovers the relationships among “French horn”, “French horn muted” and “bassoon”. Instead of clustering two “French horn” sounds in one group as the conventional schema does, bassoon is considered as the sibling of the regular French horn. “French horn muted” is clustered in another different group together with “English Horn” and “Oboe” (the extent of the difference between groups is measured by the distance between the nodes in the hierarchical tree).

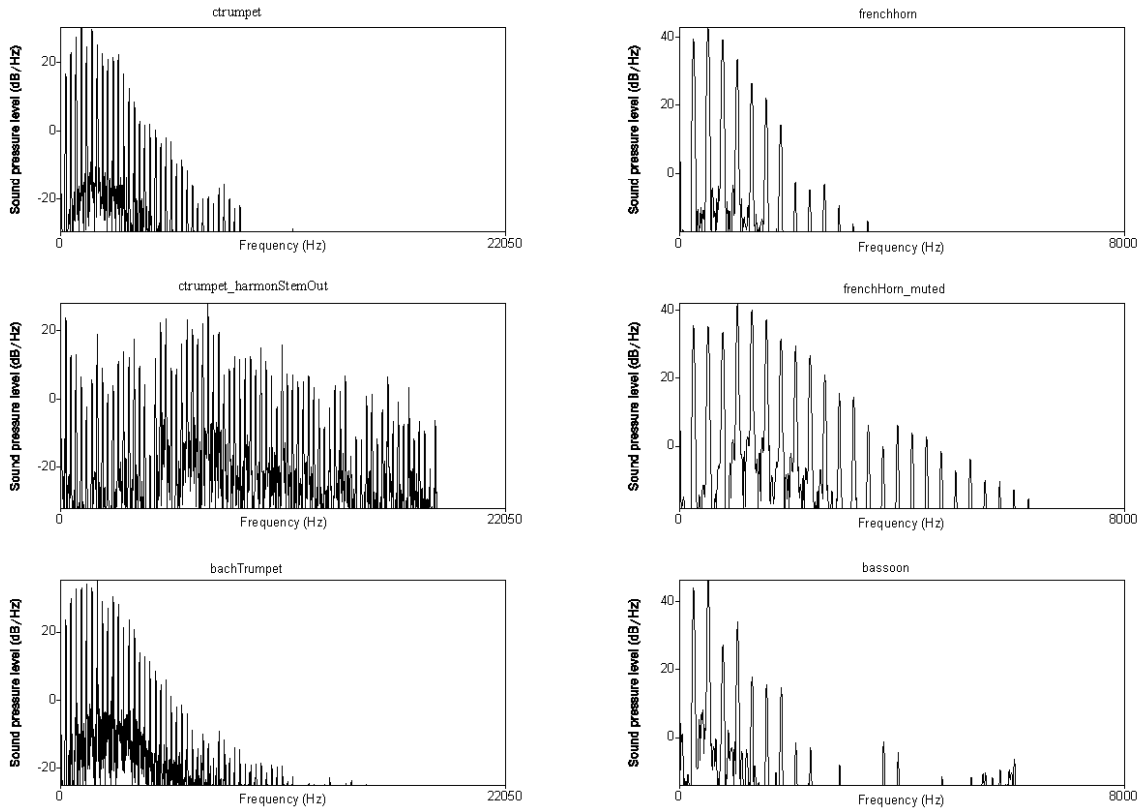

Figure 6.2 Spectrum comparison of different instrument objects (left: CTrumpet, CTrumpet harmonStemout and Batch Trumpet; right: French horn, French horn muted, bassoon)

According to this result, the new schema is more accurate than the traditional schema because it represents the actual similarity of timbre qualities of musical instruments. It not only better describes the difference among instruments, but also distinguishes the sounds produced by the same instrument that have quite different timbre qualities due to the different playing techniques.

6.4 Experiments and evaluation

In order to evaluate the new schema, we test it with the cascade classification system and compare the timbre estimation result with the results from the two previous conventional hierarchical schemas: Hornbostel/Sachs and Playing Method. During the classification process for each single frame, we use the flatness coefficients to perform

family estimation on the higher level of hierarchical tree. After reaching the bottom level of hierarchical tree, the power spectrum is extracted from the analyzed frame to match against the reference spectral database. Since the spectrum matching is performed in a small subgroup, the computation complexity is reduced. The testing data is the same data set which was used in previous chapter.

Table 6.5 Comparison between non-cascade classification and cascade classification with different hierarchical schemas

Number	classification method	Description	Recall	Precision	F-Score
1	non-cascade	Feature-based	64.3%	44.8%	52.81%
2	non-cascade	Spectrum-Match	79.4%	50.8%	61.96%
3	Cascade	Hornbostel/Sachs	75.0%	43.5%	55.06%
4	Cascade	play method	77.8%	53.6%	63.47%
5	Cascade	machine Learned	87.5%	62.3%	72.78%

We test the polyphonic sound with five different approaches. Experiment 1 and 2 apply the KNN ($k=3$) and use the non-cascade classification approach. The instruments are directly estimated by the classifier. Spectral flatness coefficients are used as feature for experiment 1 and power spectrum is used for experiment 2. Experiment 3, 4 and 6 apply the cascade methods and KNN ($k=3$) is the classifier used at each level of classification process. Three different hierarchical schemas are applied. Table 6.5 and Figure 6.3 show that generally the cascade classification improves the recall compared to the non-cascade methods. The non-cascade classification based on spectrum-match (experiment 2) shows a higher recall than the cascade classification approaches based on the traditional hierarchical schema (experiment 3 and 4). However, the cascade classification based on the new schema learned by the clustering analysis (experiment 5) outperforms the non-cascade classification. This shows that the new schema gives a significant improvement in comparison to the other two traditional schemas. Because of

the additional levels of hierarchical tree, the size of the subset on the bottom level is reduced to a very small size, significantly reducing the cost of spectrum matching.

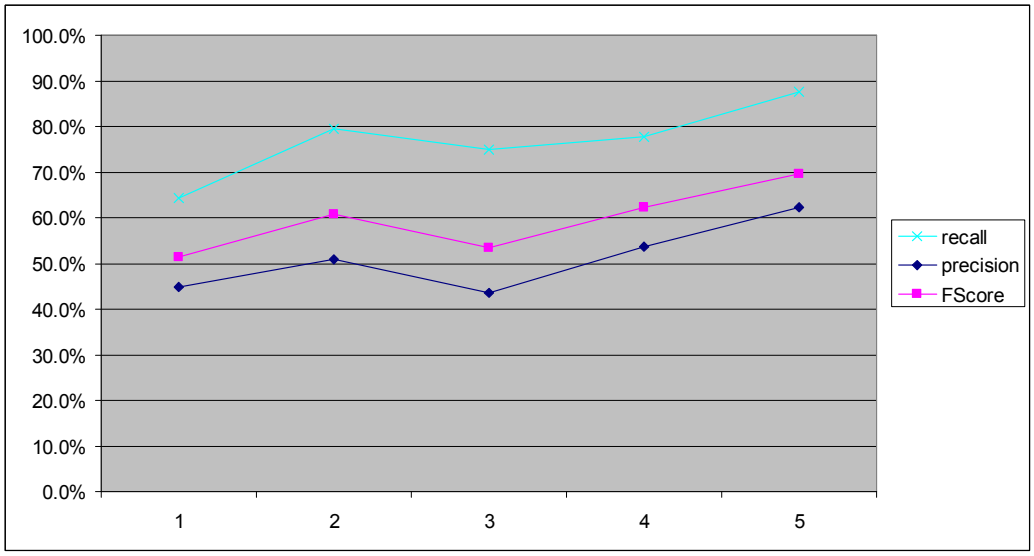

Figure 6.3 Comparison between non-cascade classification and cascade classification with different hierarchical schemas

We evaluate the classification system by the mixture sounds which contain two single instrument sounds. In the real world, there could be more than two instruments playing simultaneously, especially in the orchestra music. We also create 49 polyphonic sounds by randomly selecting three different single instrument sounds and mixing them together. We then test those three-instrument mixtures with five different classification methods (experiment 2 to 6) which are described in the previous two-instrument mixture experiments. Single-label classification based on the sound separation method is also tested on the mixtures (experiment 1). KNN ($k=3$) is used as the classifier for each experiment.

Table 6.6 Classification results of 3-instrument mixtures with different algorithms

Number	Classifier	Method	Recall	precision	F-Score
1	Non-Cascade	single-label based on sound separation	31.48%	43.06%	36.37%
2	Non_Cascade	Feature-based multi-label classification	69.44%	58.64%	63.59%
3	Non_Cascade	Spectrum-Match multi-label classification	85.51%	55.04%	66.97%
4	Cascade (hornbostel)	multi-label classification	64.49%	63.10%	63.79%
5	Cascade (playmethod)	multi-label classification	66.67%	55.25%	60.43%
6	Cascade (machine Learned)	multi-label classification	63.77%	69.67%	66.59%

From table 6.6, we see the very low recognition precision and recall for the algorithm based on the sound separation. After the twice signal subtractions during the first two instruments estimation, there is little information remaining in the mixture for the further classification of the third instrument. The cascade method based on multi-label classification remains the method with the highest recall and precision. This experiment shows the robustness and effectiveness of the algorithm for the polyphonic sounds which contain more than two timbres.

As the dendrogram in Figure 6.1 shows, the new schema has more hierarchical levels and looks more complex and obscure to users. But we only use it as the internal structure for the cascade classification process instead of the query interface. When the user submits a query to QAS through the user semantic structure, the system translates it to the internal schema. After the estimation is done, the answer is converted back to the user semantics. The user does not need to know the difference between French horn and

French horn muted since only French horn is returned by the system as the final estimation result.

CHAPTER 7: CONCLUSION AND DISCUSSION

In this dissertation the timbre estimation based on the classification algorithm is discussed. In order to deal with the polyphonic sound, multi-label classifiers derived from the single-class training database are introduced. The testing results show that the multi-label classification yields higher recognition rate and accuracy than the traditional single-label classification based on the sound separation method. Power spectrum matching based on KNN method is also proposed and shows the improvement of estimation accuracy. Given the fact that spectrum matching in a large training database is much more expensive than feature based classification, the cascade classifier is introduced to give a good solution to achieving both high recognition rate and high efficiency. Cascade classification system needs to know how to choose the appropriate classifier and features at each level of hierarchical tree. The experiments are conducted to discover such knowledge based on the current training database.

We also develop the new temporal features based on the MPEG7 acoustic descriptors to efficiently retain the critical temporal information for instrument classification. The new features strengthen the recognition ability of the classifier for some instruments that share the similar pattern in spectral space.

We introduce a new hierarchical structure for the cascade classification system based on the hierarchical clustering results. Compared to the traditional schemas which are manually designed by the musicians, the new schema better represents the relationships

among musical instruments in terms of their timbre similarity since the hierarchical structures are directly derived from the acoustic features based on the their similarity matrix. Better results are shown with the cascade classification system.

We intend to continue our work on the Music Information Retrieval based on Automatic Indexing by Instruments and their types along several directions. First, we plan to explore the wavelet transforms to extract the new music sound features which are different from the FFT transform based features. Such features meet the need of different sizes of analysis windows due to the different frequency ranges in polyphonic sounds. Second, we are interested in exploring different peak detection techniques in order to capture more salient and accurate harmonic features. Usually the features could be buried in noise signals or corrupted by background sounds which leads to the false positive peaks. By applying the appropriate smoothing and baseline correction methods as well as the peak-picking algorithm [12], we are able to decrease the noise to signal ratio and assist our proposed cascade system to provide more confident results of multiple timbre recognition. We also want to know if different features could be used at the different levels of clustering process in order to give a better hierarchical structure. This information would be utilized to perform cascade classification for the unknown musical data when it comes to select features and metrics for classification algorithm. Actual music pieces are also need to be tested on the cascade system to verify the classification ability.

REFERENCES

[1] A Survey of Music Information Retrieval Systems, http://mirsystems.info/

[2] Balzano, G.J. (1986). What are Musical Pitch and Timbre? in Music Perception - an Interdisciplinary Journal. 3, 297-314.

[3] Bregman, A.S. (1990). Auditory Scene Analysis, the Perceptual Organization of Sound, MIT Press.

[4] Brown, J.C. (1999). Computer Identification of Musical Instruments Using Pattern Recognition with Cepstral Coefficients as Features, in J. Acoust. Soc. Am. 105, 1933–1941.

[5] Cadoz, C. (1985). Timbre et Causalite, unpublished paper, Seminar on Timbre, Institute de Recherche et Coordination Acoustique / Musique, Paris, France, April 13-17.

[6] Ian H. Witten and Eibe Frank (2005) "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, 2005

[7] Cosi, P. (1998). Auditory Modeling and Neural Networks, in A Course on Speech Processing, Recognition, and Artificial Neural Networks, Springer, Lecture Notes in Computer Science

[8] Cutting D., Kupiec, J., Pedersen, J, Sibun, P. (1992). A Practical Part-of-Speech Tagger, in the Third Conference on Applied Natural Language Processing, 133-140.

[9] D. Aha, D. Kibler (1991). Instance-based learning algorithms. Machine Learning. 6:37-66.

[10] Fujinaga I, McMillan K. (2000). Real Time Recognition of Orchestral Instruments, in International Computer Music Conference

[11] George H. John, Pat Langley: Estimating Continuous Distributions in Bayesian Classifiers. In: Eleventh Conference on Uncertainty in Artificial Intelligence, San Mateo, 338-345, 1995.

[12] Gu, Liang / Rose, Kenneth (2000): "Perceptual harmonic cepstral coefficients as the front-end for speech recognition", In ICSLP-2000, vol.1, 309-312.

[13] Haskell, R.-E. (1989) Design of hierarchical classifiers, in Proceedings of The First Great Lakes Computer Science Conference on Computing in the 900s,LNCS, Vol. 507, 118-124

[14] ISO/IEC JTC1/SC29/WG11 (2004). MPEG-7 Overview. Available at http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm

[15] Jensen, K., Arnspang, J. (1999). Binary Decision Tree Classification of Musical Sounds, in International Computer Music Conference, Beijing, China, Oct. 1999.

[16] Kupiec, J. (1992). Robust Part-of-Speech Tagging Using a Hidden Markov Model, in the Computer Speech and Language 6, 225-242.

[17] Trohidis, K., Tsoumakas, G., Kalliris, G., Vlahavas, I. (2008). Multi-label Classification of Music into Emotions, in the Proc. 2008 International Conference on Music Information Retrieval (ISMIR 2008)

[18] Boutell, M.R., Luo, J., Shen, X., Brown, C.M. (2004). Learning Multi-Label Scene Classification, in Pattern Recognition, Vol. 37, No. 9, 1757-1771.

[19] Lindsay, A.T., Herre, J. (2001). MPEG7 and MPEG7 Audio - An Overview, in J. Audio Engineering Society, Honolulu, Hawaii, July/Aug, Vol.49, 589–594.

[20] Logan, B. (2000). Mel Frequency Cepstral Coefficients for Music Modeling, in the Proceedings of the First International Symposium on Music Information Retrieval (MUSIC IR 2000)

[21] Lu, C., Drew, M.S. (2001) Construction of a hierarchical classifier schema using a combination of text-based and image-based approaches, in SIGIR001 Proceedings, ACM Publications, 331-336

[22] Thorsten, J. (1998). Text Categorization with Support Vector Machines: Learning with many relevant features, in the Proceedings of Tenth European Conference on Machine Learning, 137-142.

[23] MacQueen, J.B. Some Methods for classification and Analysis of Multivariate Observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. University of California Press. pp. 281–297 (1967)

[24] Martin, K.D., Kim, Y.E. (1998). Musical Instrument Identification, a Pattern-Recognition Approach, in the Proceedings of 136th Meeting of the Acoustical Soc. of America, Norfolk, VA., 2pMU9

[25] Paulus, J., Virtanen, T. (2005). Drum Transcription with Non-Negative Spectrogram Factorization, in the Proceedings of 13th European Signal Processing Conference, EUSIPCO, Antalya, Turkey, 4-8 September.

[26] Pollard, H.F., Jansson, E.V. (1982). A Tristimulus Method for the Specification of Musical Timbre, in Acustica, Vol. 51, 162–171.

[27] Lienhart, R., Kuranov, A., Pisarevsky, V. (2003). Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection, in Pattern Recognition Journal, 297-304.

[28] Quinlan,J. R.(1986) "Induction of Decision Trees". Machine Learning 1pp. 81-106.Reprinted in Shavlik and Dietterich (eds.) readings in Machine learning

[29] Quinlan Ross (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA.

[30] Ras, Z., Zhang, X., Lewis, R. (2007). MIRAI: Multi-hierarchical, FS-tree based Music Information Retrieval System (Invited Paper), in Kryszkiewicz, M., Peters, J., Rybinski, H, Skowron, A. (Eds), Proceedings of the International Conference on Rough Sets and Intelligent System Paradigms (RSEISP 2007), Springer, LNAI 4585, 80-89.

[31] Ras, Z.W., Dardzinska, A., Zhang, X. (2007). Cooperative Answering of Queries based on Hierarchical Decision Attributes, CAMES Journal, Polish Academy of Sciences, Institute of fundamental Technological Research, Vol. 14, No. 4, 729-736.

[32] R Development Core Team (2005). R: Language and Environment for Statistical Computing, in R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

[33] Schapire, R., Singer, Y. (2000). BoosTexter: A Boosting-Based System for Text Categorization, in Machine Learning, Vol. 39, No. 2/3, 135-168.

[34] Scheirer, E., Slaney, M. (1997). Construction and Evaluation of a Robust Multi-feature Speech/Music Discriminator, in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

[35] Sneath, P. H. A., Sokal, R.R. (1973). Numerical Taxonomy, Freeman, San Francisco.

[36] The Statistics Homepage: Cluster Analysis. http://statsoft.com/textbook/stathome.html

[37] Thabtah, F.A., Cowling, P., Peng, Y. (2006). Multiple Labels Associative Classification, in Knowledge and Information Systems, Vol. 9, No. 1., 109-129

[38] Athitsos, V., Alon, J., Sclaroff , S. (2005). Efficient Nearest Neighbor Classification Using a Cascade of Approximate Similarity Measures, in Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

[39] Welch, P.D. (1967). The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms, in IEEE Transactions on Audio Electroacoustics, Volume AU-15 (June 1967), 70-73.

[40] Wieczorkowska, A. (1999). Classification of Musical Instrument Sounds Using Decision Trees, in The 8th International Symposium on Sound Engineering and Mastering, ISSEM'99, 225-230.

[41] Wold, E., Blum, T., Keislar, D., Wheaten, J. (1996) Content-Based Classification, Search, and Retrieval of Audio, in Multimedia, IEEE, Vol. 3, No. 3, 27-36.

[42] Zhang, X., Ras, Z.W., Dardzinska, A. (2007). Discriminant Feature Analysis for Music Timbre Recognition and Automatic Indexing, in Mining Complex Data, Post-Proceedings of 2007 ECML/PKDD Third International Workshop (MCD 2007), LNAI, Vol. 4944, Springer, 2008, 104-115

[43] Freund, Y. (1990) Boosting a Weak Learning Algorithm by Majority, in Proceedings of the Third Annual Workshop on Computational Learning Theory.

[44] Freund. Y., Schapire, R.E. (1997) Decision-Theoretic Generalization of On-Line Learning and Application to Boosting, in Journal of Computer and System Sciences, Vol. 55, No. 1, 119-139