

An Automatic Annotation System for Audio Data Containing Music

by

Janet Marques

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degrees of
Bachelor of Science in Electrical Engineering and Computer Science
and Master of Engineering in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

May 1, 1999

June 10007

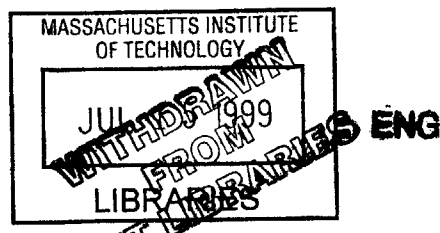
© Copyright 1999 Janet Marques. All rights reserved.

The author hereby grants to M.I.T. permission to reproduce and
distribute publicly paper and electronic copies of this thesis
and to grant others the right to do so.

Author _____
Department of Electrical Engineering and Computer Science
May 1, 1999

Certified by _____
Tomaso A. Poggio
Thesis Supervisor

Accepted by _____
Arthur C. Smith
Chairman, Department Committee on Graduate Theses



An Automatic Annotation System for Audio Data Containing Music

by
Janet Marques

Submitted to the
Department of Electrical Engineering and Computer Science

May 1, 1999

In Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science in Electrical Engineering and Computer Science
and Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

This thesis describes an automatic annotation system for audio files. Given a sound file as input, the application outputs time-aligned text labels that describe the file's audio content. Audio annotation systems such as this one will allow users to more effectively search audio files on the Internet for content. In this annotation system, the labels include eight musical instrument names and the label 'other'. The musical instruments are bagpipes, clarinet, flute, harpsichord, organ, piano, trombone, and violin. The annotation tool uses two sound classifiers. These classifiers were built after experimenting with different parameters such as feature type and classification algorithm. The first classifier distinguishes between instrument set and non instrument set sounds. It uses Gaussian Mixture Models and the mel cepstral feature set. The classifier can correctly classify an audio segment, 0.2 seconds in length, with 75% accuracy. The second classifier determines which instrument played the sound. It uses Support Vector Machines and also uses the mel cepstral feature set. It can correctly classify an audio segment, 0.2 seconds in length, with 70% accuracy.

Thesis Supervisor: Tomaso A. Poggio
Title: Uncas and Helen Whitaker Professor

Acknowledgements

I would like to thank all of my research advisors: Tomaso Poggio (my MIT thesis advisor), Brian Eberman, Dave Goddeau, Pedro Moreno, and Jean-Manuel Van Thong (my advisors at Compaq Computer Corporation's Cambridge Research Lab, CRL).

I would also like to thank everyone at CRL for providing me with a challenging and rewarding environment during the past two years. My thanks also go to CRL for providing the funding for this research. The Gaussian Mixture Model code used in this project was provided by CRL's speech research group, and the Support Vector Machine software was provided to CRL by Edgar Osuna and Tomaso Poggio. I would also like to thank Phillip Clarkson for providing me with his SVM client code.

Many thanks go to Judith Brown, Max Chen, and Keith Martin. Judith provided me with valuable guidance and useful information. Max provided helpful music discussions and contributed many music samples to this research. Keith also contributed music samples, pointed me at important literature, and gave many helpful comments.

I would especially like to thank Brian Zabel. He provided me with a great deal of help and support and contributed many helpful suggestions to this research.

Finally, I would like to thank my family for all of their love and support.

Table of Contents

ABSTRACT	2
ACKNOWLEDGEMENTS	3
TABLE OF CONTENTS	4
CHAPTER 1 INTRODUCTION	6
1.1 OVERVIEW	6
1.2 MOTIVATION.....	6
1.3 PRIOR WORK.....	7
1.4 DOCUMENT OUTLINE	9
CHAPTER 2 BACKGROUND INFORMATION	10
2.1 INTRODUCTION	10
2.2 FEATURE SET	10
2.3 CLASSIFICATION ALGORITHM.....	15
2.3.1 <i>Gaussian Mixture Models</i>	15
2.3.2 <i>Support Vector Machines</i>	16
CHAPTER 3 CLASSIFIER EXPERIMENTS	19
3.1 INTRODUCTION	19
3.2 AUTOMATIC ANNOTATION SYSTEM.....	19
3.3 EIGHT INSTRUMENT CLASSIFIER	20
3.3.1 <i>Data Collection</i>	20
3.3.2 <i>Experiments</i>	22
3.4 INSTRUMENT SET VERSUS NON INSTRUMENT SET CLASSIFIER	26
3.4.1 <i>Data Collection</i>	26
3.4.2 <i>Experiments</i>	27
CHAPTER 4 RESULTS AND DISCUSSION	31
4.1 INTRODUCTION	31
4.2 AUTOMATIC ANNOTATION SYSTEM.....	31
4.3 EIGHT-INSTRUMENT CLASSIFIER	33
4.3.1 <i>Gaussian Mixture Model Experiments</i>	34
4.3.2 <i>Support Vector Machine Experiments</i>	41

4.4 INSTRUMENT SET VERSUS NON INSTRUMENT SET CLASSIFIER	43
4.4.1 <i>Gaussian Mixture Model Experiments</i>	43
4.4.2 <i>Support Vector Machine Experiments</i>	43
4.4.3 <i>Probability Threshold Experiments</i>	43
CHAPTER 5 CONCLUSIONS AND FUTURE WORK	44
5.1 CONCLUSIONS	44
5.2 FUTURE WORK.....	45
5.2.1 <i>Accuracy Improvements</i>	45
5.2.2 <i>Classification of Concurrent Sounds</i>	46
5.2.3 <i>Increasing the Number of Labels</i>	47
APPENDIX.....	48
REFERENCES.....	51

Chapter 1 Introduction

1.1 Overview

This thesis describes an automatic annotation system for audio files. Given a sound file as input, the annotation system outputs time-aligned labels that describe the file's audio content. The labels consist of eight different musical instrument names and the label 'other'.

1.2 Motivation

An important motivation for audio file annotation is multimedia search retrieval. There are approximately thirty million multimedia files on the Internet with no effective method available for searching their audio content (Swa98).

Audio files could be easily searched if every sound file had a corresponding text file that accurately described people's perceptions of the file's audio content. For example, in an audio file containing only speech, the text file could include the speakers' names and the spoken text. In a music file, the annotations could include the names of the musical instruments. These transcriptions could be generated manually; however, it would take a great amount of work and time for humans to label every audio file on the Internet. Automatic annotation tools must be developed.

This project focuses on labeling audio files containing solo music. Since the Internet does not contain many files with solo music, this type of annotation system is not immediately practical. However, it does show “proof of concept”. Using the same techniques, this work can be extended to include other types of sound such as animal sounds or musical style (jazz, classical, etc.).

A more immediate use for this work is in audio editing applications. Currently, these applications do not use information such as instrument name for traversing and manipulating audio files. For example, a user must listen to an entire audio file in order to find instances of specific instruments. Audio editing applications would be more effective if annotations were added to the sound files (Wol96).

1.3 Prior Work

There has been a great deal of research concerning the automatic annotation of speech files. However, the automatic annotation of other sound files has received much less attention.

Currently, it is possible to annotate a speech file with spoken text and name of speaker using speech recognition and speaker identification technology. Researchers have achieved a word accuracy of 82.6% for “found speech”, speech not specifically recorded for speech recognition (Lig98). In speaker identification, systems can distinguish between approximately 50 voices with a 96.8% accuracy (Rey95).

The automatic annotation of files containing other sounds has also been researched. One group has successfully built a system that differentiates between the following sound classes: laughter, animals, bells, crowds, synthesizer, and various musical instruments (Wol96). Another group was able to classify sounds as speech or music with 98.6% accuracy (Sch97). Researchers have also built a system that differentiates between classical, jazz, and popular music with 55% accuracy (Han98).

Most of the work done in music annotation has focused on note identification. In 1977, a group built a system that could produce a score for music containing one or more harmonic instruments. However, the instruments could not be vibrato or glissando, and there were strong restrictions on notes that occurred simultaneously (Moo77). Subsequently, better transcription systems have been developed (Kat89, Kas95, and Mar96).

There have not been many studies done on musical instrument identification. One group built a classifier for four instruments: piano, marimba, guitar, and accordion. It had an impressive 98.1% accuracy rate. However, in their experiments the training and test data were recorded using the same instruments. Also, the training and test data were recorded under similar conditions (Kam95). We believe that the accuracy rate would decrease substantially if the system were tested on music recorded in a different laboratory.

In another study, researchers built a system that could distinguish between two instruments. The sound segments classified were between 1.5 and 10 seconds long. In this case, the test set and training set were recorded using different instruments and under different conditions. The average error rate was a low 5.6% (Bro98). We believe that the error rate would increase if the classifier were extended either through: labeling sound segments less than 0.4 seconds long, or by increasing the number of instruments.

Another research group built a system that could identify 15 musical instruments using isolated tones. The test set and training set were recorded using different instruments and under different conditions. It had a 71.6% accuracy (Mar98). We believe that the error rate would increase if the classifier were not limited to isolated tones.

In our research study, the test set and training set were recorded using different instruments and under different conditions. Also, the sound segments were less than 0.4 seconds long and were not restricted to isolated tones. This allowed us to build a system that could generate accurate labels for audio files not specifically recorded for recognition.

1.4 Document Outline

The first chapter introduces the thesis project. The next chapter describes the relevant background information. In the third chapter, the research experiments are described. The fourth chapter is devoted to the presentation and analysis of the experiment results. The last part of the thesis includes directions for future work and references.

Chapter 2 Background Information

2.1 Introduction

This chapter discusses background information relevant to the automatic annotation system. The system is made up of two sound classifiers. The first classifier distinguishes between instrument set and non instrument set sounds. If the sound is deemed to be made by an instrument, the second classifier determines which instrument played the sound. For both classifiers, we needed to choose an appropriate feature set and classification algorithm.

2.2 Feature Set

The first step in any classification problem is to select an appropriate *feature set*. A feature set is a group of properties exhibited by the data being classified. An effective feature set should contain enough information about the data to enable classification, but also contain as few elements as possible. This makes modeling more efficient (Dud73).

Finding a feature set is usually problem dependent. In this project, we needed to find features that accurately distinguish harmonic musical instruments. One feature that partially distinguishes instruments is frequency range. Figure 2.1 illustrates the frequency range of many harmonic instruments (Pie83). Since some of the instruments'

frequency ranges overlap, we can say that harmonic instruments are not uniquely specified by frequency range alone.

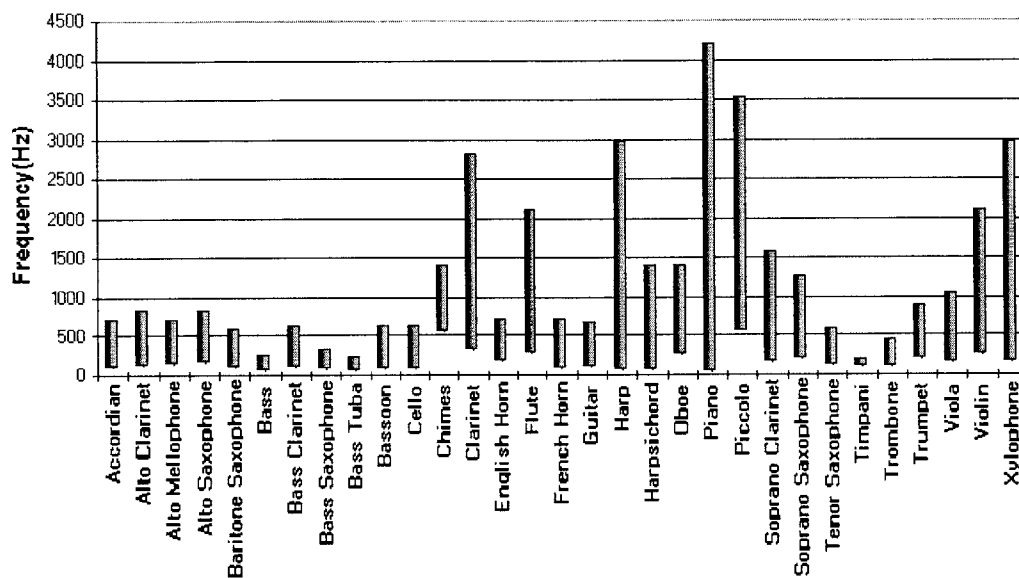


Figure 2.1 Frequency range of various musical instruments.

Another possible feature set is *harmonic* number amplitude. Harmonics are the sinusoidal frequency components of a periodic sound. Musical instruments tend to sound differently because of differences in their harmonics. For example, a sound with intense energy in higher frequency harmonics tends to sound bright like a piccolo, while a sound with high energy in lower frequency harmonics tends to sound rather dull like a tuba (Pie83).

The harmonic amplitude feature set has been successful in some classification problems. In one study, the mean amplitudes of the first 11 harmonics were used as a feature set for the classification of diesel engines and rotary wing aircraft. An error rate of 0.84% was achieved (Mos96). However, we believed that this feature set might not be successful for an instrument classifier. For instance, the feature set is well defined for instruments that can only play one note at a time, such as clarinet, flute, trombone, and violin. However, it was not straightforward how we would derive such a feature set for instruments like

piano, harpsichord, and organ. Also, we believed that the harmonic feature set might not contain enough information to uniquely distinguish the instruments. Therefore, we hypothesized that a harmonic amplitude feature set may not be the best feature set for musical instrument classification.

Two feature sets that are successful in speech recognition are linear prediction coefficients and cepstral coefficients. Both feature sets assume the speech production model shown in Figure 2.2. The source $u(n)$ is a series of periodic pulses produced by air forced through the vocal chords, the filter $H(z)$ represents the vocal tract, and the output $o(n)$ is the speech signal (Rab93). Both feature sets attempt to approximate the vocal tract system.

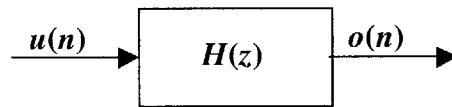


Figure 2.2 Linear prediction and cepstrum model of speech production and musical instrument sound production.

The model shown in Figure 2.2 is also suitable for musical instrument sound production. The source $u(n)$ is a series of periodic pulses produced by air forced through the instrument or by resonating strings, the filter $H(z)$ represents the musical instrument, and the output $o(n)$ represents the music. In this case, both feature sets attempt to approximate the musical instrument system.

When deriving the LPC feature set, the musical instrument system is approximated using an all-pole model,

$$H(z) = \frac{G}{1 - \sum_{i=1}^p a_i z^{-i}},$$

where G is the model's gain, p is the order of the LPC model, and $\{a_1 \dots a_p\}$ are the model coefficients. The linear prediction feature set is $\{a_1 \dots a_p\}$ (Rab93).

One problem with the LPC set is that it approximates the musical instrument system using an all-pole model. This assumption is not made when deriving the cepstral feature set. The cepstral feature set is derived using the cepstrum transform:

$$\text{cepstrum}(x) = \text{FFT}^{-1}(\ln |\text{FFT}(x)|).$$

If x is the output of the system described in Figure 2.2, then

$$\text{cepstrum}(x) = \text{FFT}^{-1}(\ln |\text{FFT}(u(n))|) + \text{FFT}^{-1}(\ln |H(z)|).$$

$\text{FFT}^{-1}(\ln |\text{FFT}(u(n))|)$ is a transformation of the input into the musical instrument system. It is approximately a series of periodic pulses with some period N_0 . $\text{FFT}^{-1}(\ln |H(z)|)$ is a transformation of the musical instrument system. Since $\text{FFT}^{-1}(\ln |H(z)|)$ contributes minimally to the first $N_0 - 1$ samples of $\text{cepstrum}(x)$, the first $N_0 - 1$ samples should contain a good representation of the musical instrument system. These samples are the cepstral feature set (Gis94).

A variation of the cepstral feature set is the mel cepstral set. This feature set is identical to the cepstral except that the signal's frequency content undergoes a mel transformation before the cepstral transform is calculated. Figure 2.3 shows the relationship between Hertz and mel frequency. This transformation modifies the signal so that its frequency content is more closely related to a human's perception of frequency content. The relationship is linear for lower frequencies and logarithmic at higher frequencies (Rab93).

The mel transformation has improved speech recognition results because speech phonemes appear more different on a mel frequency scale than on a linear Hertz scale. Similarly to the distinctness of human phonemes, the musical instruments in this study produce different sounds according to the human ear. Therefore, it was our prediction that the mel transformation would improve instrument classification results.

A feature set type that has not been discussed is temporal features. Temporal features are most effective for sound signals that have important time-varying characteristics. Examples of such feature sets include wavelet packet coefficients, autocorrelation

coefficients, and correlogram coefficients. The wavelet feature set has been used in respiratory sound classification and marine mammal sound classification (Lea93). The autocorrelation and correlogram feature sets have been used in instrument classification (Bro98, Mar98).

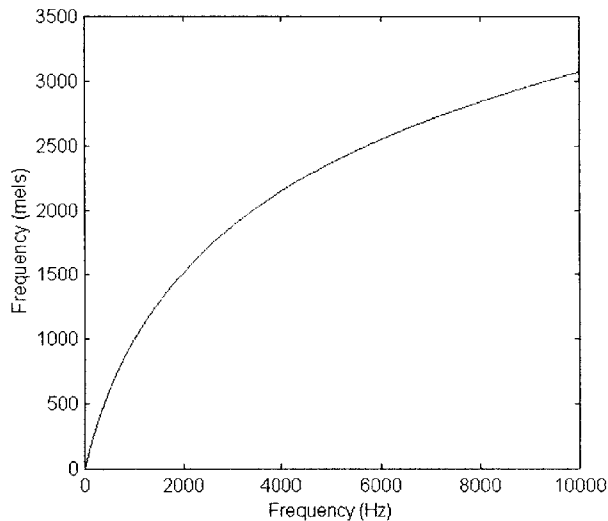


Figure 2.3 Relationship between Hertz and mels.

A large amount of psychophysical data has shown that musical instruments have important time-varying characteristics. For example, humans often identify instruments using attack and decay times (Mca93). Feature sets that exploit time-varying characteristics are most likely more effective than feature sets that do not use temporal cues. In one study, a fifteen-instrument identification system using correlogram coefficients achieved a 70% accuracy rate. The system was trained and tested using individual tones.

In this study, continuous music as opposed to individual tones was examined. Since it is difficult to accurately extract individual tones from continuous music, temporal feature sets were not examined in this research.

One other common feature extraction method is to use a variety of acoustic characteristics, such as loudness, pitch, spectral mean, bandwidth, harmonicity, and

spectral flux. This method has been successful in many classification tasks (Sch97, Wol96). However, it is difficult to determine which acoustic characteristics are most appropriate for a given classification problem. This type of feature set was not explored in this research.

In this thesis, we examined the following feature sets: harmonic amplitude, linear prediction, cepstral, and mel cepstral.

2.3 Classification Algorithm

After an adequate feature set has been selected, the classification algorithm must be chosen. The Gaussian mixture model (GMM) and Support Vector Machine (SVM) algorithms are discussed below.

2.3.1 Gaussian Mixture Models

Gaussian Mixture Models have been successful in speaker identification and speech recognition. In this algorithm, the training feature vectors for the instrument classifier are used to model each instrument as a probability distribution. A test vector \bar{x} is classified as the instrument with the highest probability for that feature vector.

Each instrument's probability distribution is represented as a sum of Gaussian densities:

$$p(\bar{x} | C_j) = \sum_{i=1}^K P(g_i | C_j) p(\bar{x} | w_i, C_j),$$

$$\text{where } p(\bar{x} | w_i, C_j) = \frac{\exp\left(-1/2(\bar{x} - \bar{\mu}_{ij})' \Sigma_{ij}^{-1} (\bar{x} - \bar{\mu}_{ij})\right)}{\sqrt{(2\pi)^d |\Sigma_{ij}|}}.$$

\bar{x} represents a feature vector, g_i represents Gaussian i , C_j represents class j , K is the number of Gaussian densities for each class, $P(g_i | C_j)$ is the probability of Gaussian i given class j , d is the number of features, \bar{x} is a d -component feature vector, $\bar{\mu}_{ij}$ is a d -

component mean vector for Gaussian i in class j , Σ_{ij} is a $d \times d$ covariance matrix for Gaussian i in class j , $(\bar{x} - \bar{\mu}_{ij})'$ is the transpose of $\bar{x} - \bar{\mu}_{ij}$, Σ_{ij}^{-1} is the inverse of Σ_{ij} , and $|\Sigma_{ij}|$ is the determinant of Σ_{ij} .

The Gaussian mixture used to represent each class is found using the Expectation-Maximization (EM) algorithm. EM is an iterative algorithm that computes maximum likelihood estimates. Many variations of this algorithm exist, but they were not explored in this project (Dem77). The initial Gaussian parameters (means, covariances, and prior probabilities) used by EM were generated via the k-means method (Dud73). Other initialization methods include the binary tree and linear tree algorithms, but they were not explored in this project.

Once a Gaussian mixture has been found for each class, determining a test vector's class is straightforward. A test vector \bar{x} is labeled as the class that maximizes $p(C_j | \bar{x})$ which is equivalent to maximizing $p(\bar{x} | C_j)p(C_j)$ using Bayes rule. When each class has equal *a priori* probability, then the probability measure is simply $p(\bar{x} | C_j)$. Therefore, the test vector \bar{x} is classified into the instrument class C_j that maximizes $p(\bar{x} | C_j)$.

2.3.2 Support Vector Machines

Support Vector Machines have been used in a variety of classification tasks, such as isolated handwritten digit recognition, speaker identification, object recognition, face detection, and vowel classification. When compared with other algorithms, they show improved performance. This section provides a brief summary of SVMs; a more thorough review can be found in (Bur98).

Support Vector Machines are used for finding the optimal boundary that separates two classes. We begin with a training vector set, $\{\bar{x}_1, \dots, \bar{x}_m\}$ where $\bar{x}_i \in R^n$. Each training

vector \bar{x}_i belongs to the class y_i where $y_i \in \{-1, 1\}$. The following hyperplane separates the training vectors into the two classes:

$$(\bar{w} \cdot \bar{x}) + b \text{ where } \bar{w} \in R^n, b \in R, \text{ and } y_i((\bar{w} \cdot \bar{x}_i) + b) \geq 1 \quad \forall i.$$

However, this hyperplane is not optimal. An optimal hyperplane would maximize the distance between the hyperplane and the closest samples, \bar{x}_1 and \bar{x}_2 . This distance is $2/|\bar{w}|$ which is illustrated in Figure 2.4.

Therefore, the optimal hyperplane, $(\bar{w} \cdot \bar{x}) + b$, is found by minimizing $|\bar{w}|$ while maintaining $y_i((\bar{w} \cdot \bar{x}_i) + b) \geq 1 \quad \forall i$. This problem can be solved with quadratic programming, and results in $\bar{w} = \sum_i v_i \bar{x}_i$. The \bar{x}_i are a subset of the training samples that lie on the margin. They are called support vectors. The v_i are the multiplying factors (Sch98).

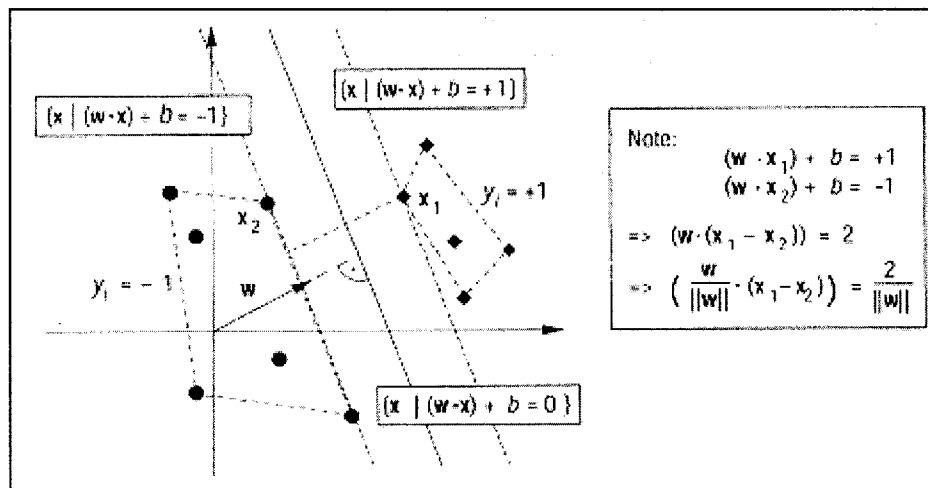


Figure 2.4 The optimal hyperplane maximizes the distance between the hyperplane and the closest samples, \bar{x}_1 and \bar{x}_2 . This distance is $2/|\bar{w}|$ (reproduced from Sch98).

In the previous case, a linear separating hyperplane was used to separate the classes. Often a non-linear hyperplane is necessary. First, the training vectors are mapped into a

higher dimensional space, $\Phi(\bar{x}_i)$ where $\Phi: R^n \mapsto H$. Then, a linear hyperplane is found in the higher dimensional space. This translates to a non-linear hyperplane in the original input space (Bur98, Sch98).

Since it is computationally expensive to map the training vectors into a high dimensional space, a kernel function is used to avoid the computational burden. The kernel function is used instead of computing $\Phi(\bar{x}_i) \cdot \Phi(\bar{x}_j)$. Two commonly used kernels are the polynomial kernel, $K(\bar{x}_i, \bar{x}_j) = (\bar{x}_i \cdot \bar{x}_j + 1)^p$ and the Gaussian radial basis function (RBF) kernel, $K(\bar{x}_i, \bar{x}_j) = \exp\left(-\frac{|\bar{x}_i - \bar{x}_j|^2}{2\sigma^2}\right)$ (Bur98). The kernel function used in this research was $K(\bar{x}_i, \bar{x}_j) = (\bar{x}_i \cdot \bar{x}_j + 1)^3$. We chose a polynomial of order 3 because it has worked well in a variety of classification experiments.

We have discussed SVMs in terms of two-class problems, however, SVMs are often used in multi-class problems. There are two popular multi-class classification algorithms, one-versus-all and one-versus-one.

In the one versus one method, a boundary is found for every pair of classes. Then, votes are tallied for each category by testing the vector on each two-class classifier. The vector is labeled as the class with the most votes.

In the one versus all method, we find a boundary for each class that separates the class from the remaining categories. Then, a test vector \bar{x} is labeled as the class whose boundary maximizes $|\bar{w} \cdot \bar{x} + b|$.

Chapter 3 Classifier Experiments

3.1 Introduction

This chapter outlines the automatic annotation system. It then describes the experiments involved in developing the system.

3.2 Automatic Annotation System

Given a sound file as input, the annotation system outputs time-aligned labels that describe the file's audio content. The labels consist of eight different musical instrument names and the label 'other'. First, the sound file is divided into overlapping segments. Then, each segment is processed using the algorithm shown in Figure 3.1.

The algorithm first determines if the segment is a member of the instrument set. If it is not, then it is classified as 'other'. Otherwise, the segment is classified as one of the eight instruments using the eight-instrument classifier.

After each overlapping segment is classified, the system marks each time sample with the class that received the most votes from all of the overlapping segments. Lastly, the annotation system filters the results and then generates the appropriate labels.

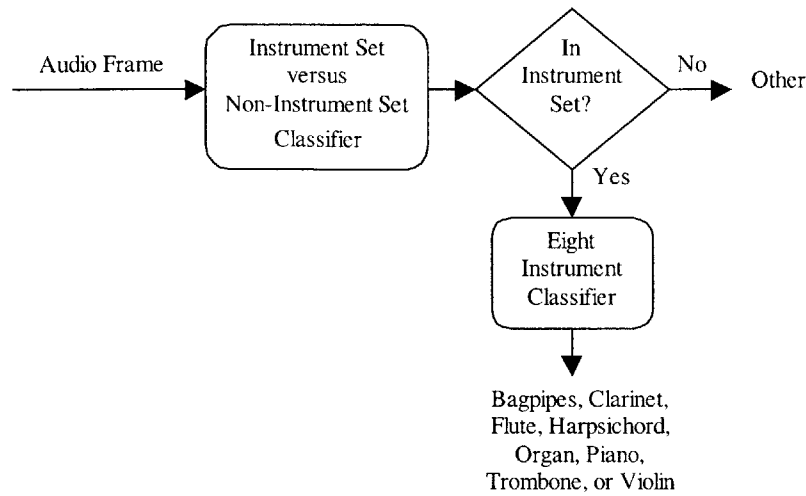


Figure 3.1 Algorithm used on each audio segment in the automatic annotation system.

The filter first divides the audio file into overlapping windows s seconds long. Then, it determines which class dominates each window. Each time sample is then labeled as the class with the most votes.

In order for the filter to work correctly, each label in the audio file must have a duration longer than s seconds. If s is too long, then the shorter sections will not be labeled correctly. We believed that the accuracy of the annotation system would be positively correlated with the value of s as long as s was less than the duration of the shortest label.

3.3 Eight Instrument Classifier

The first step in building the eight-instrument classifier was to collect training and testing data. To find the best parameters for the identification system, numerous experiments were conducted.

3.3.1 Data Collection

This part of the project used music retrieved from multiple audio compact disks (CD). The audio was sampled at 16 kHz using 16 bits per sample and was stored in AU file

format. This format compressed the 16-bit data into 8-bit mu-law data. The data was retrieved from two separate groups of CDs. Table 3.1 lists the contents of the two data sets.

	DATA SET 1	DATA SET 2
BAGPIPES	The Bagpipes & Drums of Scotland, Laserlight, Tracks 4 and 9, Length 9:39.	The bagpipe, Koch, Excerpts from tracks 5, 7-9, 11, and 12, Length 2:01.
CLARINET	20 th Century Music for Unaccompanied Clarinet, Denon, Tracks 1-6, Length 32:51.	Lonely souls, Opus, Excerpts from tracks 1-24, Length: 4:06.
FLUTE	Manuela plays French Solo Flute Music, BIS, Track 1, Length: 24:10.	Hansgeorg Schmeiser Plays Music for Solo Flute, Nimbus Records, Excerpts from tracks 1-22, Length: 2:04.
HARPSICHORD	Bach Goldberg Variations, Sine Qua Non, Track 1, Length 22:12.	20 th Century Harpsichord Music, vol. III, Gasparo, Excerpts from tracks 1-20, Length 3:26.
ORGAN	Organ Works, Archiv, Tracks 1 and 2, Length 22:25.	Romantic French Fantasies, Klavier, Excerpts from tracks 1-12, Length 2:17.
PIANO	Chopin Etudes, London, Tracks 1-5, Length 22:09. Chopin Ballades, Philips, Tracks 1 and 2, Length 18:33.	The Aldeburgh Recital, Sony, Excerpts from tracks 1-12, Length 2:19.
TROMBONE	Christian Lindberg Unaccompanied, BIS, Tracks 3, 4, 7-12, and 15-17, Length 31:09.	David Taylor, New World, Excerpts from tracks 1-6, Length 3:11.
VIOLIN	Bach Works for Violin Solo, Well Tempered, Tracks 1-5, Length 32:11.	Sonatas for Solo Violin, Orion, Excerpts from tracks 1-11, Length 2:08.

Table 3.1 Data for instrument classification experiments.

After the audio was extracted from each CD, several training and testing sets were formed. The segments were randomly chosen from the data set, and the corresponding training and test sets did not contain any of the same segments. In addition, segments with an average amplitude below 0.01 were not used. This automatically removed any silence from the training and testing sets. This threshold value was determined by listening to a small portion of the data.

Lastly, each segment's average loudness was normalized to 0.15. We normalized the segments in order to remove any loudness differences that may exist between the CD recordings. This was done so that the classifier would not use differences between the CDs to distinguish between the instruments. We later found that the CDs had additional recording differences.

3.3.2 Experiments

A variety of experiments were performed in order to identify the best parameters for the identification system. The system is outlined in Figure 3.2. The parameters of concern were test data set, segment length, feature set type, number of coefficients, and classification algorithm.

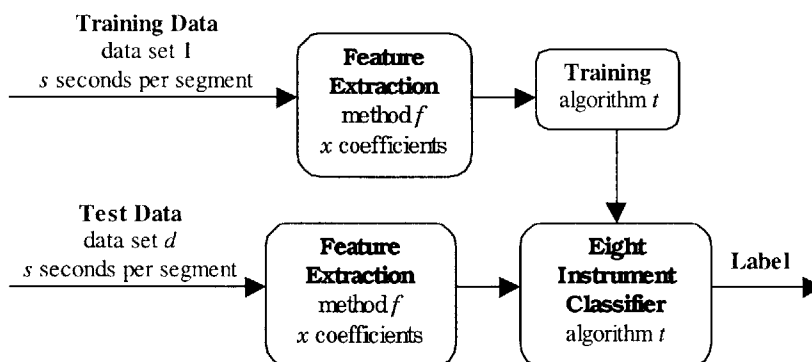


Figure 3.2 Eight instrument classification system.

Test Data Set refers to the source of the test data. Data sets 1 and 2 were extracted from two separate groups of CDs. It was important to run experiments using different data sets for training and testing. If the same data set were used for training and testing, then we would not know whether our system could classify sound that was recorded under different conditions. In a preliminary experiment, our classifier obtained 97% accuracy when trained and tested using identical recording conditions with the same instruments. This accuracy dropped to 71.6% for music professionally recorded in a different studio, and it dropped to 44.6% accuracy for non-professionally recorded music. The last two experiments used different instruments for training and testing as well.

The drop in accuracy could also be attributed to differences in the actual instruments used and not just the recording differences. If this is true, then it should have been possible for our classifier to distinguish between two instances of one instrument type, such as two violins. In a preliminary experiment, we successfully built a classifier that could

distinguish between two violin segments recorded in identical conditions with a 79.5% accuracy. The segments were 0.1 seconds long.

Segment Length is the length of each audio segment in seconds. This parameter took one of the following values: 0.05 sec, 0.1 sec, 0.2 sec, or 0.4 sec. For each experiment, we kept the total amount of training data fixed at 1638.8 seconds. We did not expect segment length to substantially affect performance. A segment longer than $2/27.5$ seconds (0.073 seconds) contains enough information to enable classification because 27.5 Hertz is the lowest frequency that can be exhibited by any of our instruments. Therefore, 0.1 seconds should have been an adequate segment length. We expected some performance loss when using 0.05 second segments.

Feature Set Type is the type of feature set used. Possible feature types include harmonic amplitudes, linear prediction, cepstral, and mel cepstral coefficients.

The harmonic amplitude feature set contains the amplitudes of the segment's first x harmonics. The harmonic amplitudes were extracted using the following algorithm:

1. Find the segment's fundamental frequency using the auto-correlation method.
2. Compute the magnitude of the segment's frequency distribution using the Fast Fourier Transform (FFT).
3. Find the harmonics using the derivative of the segment's frequency distribution. Each harmonic appears as a zero crossing in the derivative. The amplitude of each harmonic is a local maximum in the frequency distribution, and the harmonics are located at frequencies that are approximately multiples of the fundamental frequency.

An application of this algorithm is demonstrated in Figure 3.3. The frequency distribution from 0.1 seconds of solo clarinet music is shown in the first graph, and the set of eight harmonic amplitudes is illustrated in the second graph.

The next feature set used was the linear prediction feature set. It was computed using the Matlab function `lpc`, which uses the autocorrelation method of autoregressive modeling

(Mat96). The computation of the cepstral and mel cepstral feature sets was discussed in Section 2.2.

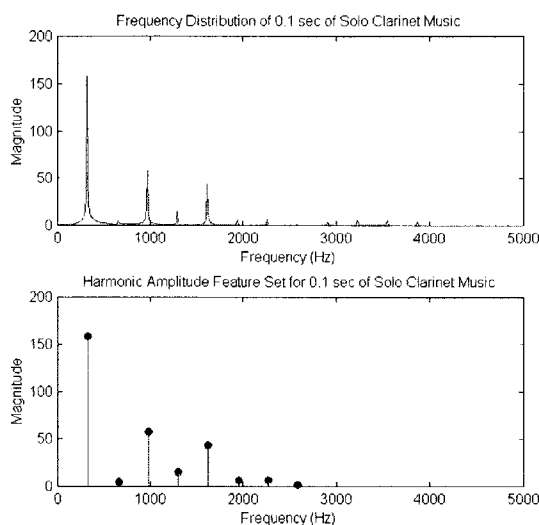


Figure 3.3 Frequency distribution of 0.1 seconds of clarinet music with corresponding harmonic amplitude feature set.

Number of Coefficients is the number of features used to represent each audio segment. This parameter took one of the following values: 4, 8, 16, 32, or 64. If there were an infinite amount of training data, then performance should improve as the feature set size is increased. More features imply that there is more information available about each class. However, increasing the number of features in a classification problem usually requires that the amount of training data be increased exponentially. This phenomenon is known as the “curse of dimensionality” (Dud73).

Classification Algorithm We experimented with two algorithms: Gaussian Mixture Models and Support Vector Machines. SVMs have outperformed GMMs in a variety of classification tasks, so we predicted that they would show improved performance in this classification task as well.

Within each classification algorithm, we experimented with important parameters. In the GMM algorithm, we examined the effect of the number of Gaussians. For the SVM

algorithm, we examined different multi-class classification algorithms; the popular algorithms being one-versus-all and one-versus-one.

Using various combinations of the parameters described above, we built many classifiers. Each classifier was tested using 800 segments of audio (100 segments per instrument). For each experiment, we examined a confusion matrix. From this matrix, we computed the overall error rate and the instruments' error rates. In the confusion matrix,

$$\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{18} \\ r_{21} & r_{22} & & r_{28} \\ \vdots & & & \vdots \\ r_{81} & r_{82} & \cdots & r_{88} \end{bmatrix},$$

there is one row and one column for each musical instrument. An element r_{ij} corresponds to the number of times the system classified a segment as instrument j when the correct answer was instrument i . The overall error rate was computed as,

$$1 - \frac{\sum_{i=1}^8 r_{ii}}{\sum_{j=1}^8 \sum_{i=1}^8 r_{ij}},$$

and the error rate for an instrument x was computed as,

$$1 - \frac{r_{xx}}{\sum_{i=1}^8 r_{ix} + \sum_{j=1}^8 r_{xj} - r_{xx}}.$$

Since each instrument's error rate includes the number of times another instrument was misclassified as the instrument in question, the overall error rate was usually lower than the individual error rates.

3.4 Instrument Set versus Non Instrument Set Classifier

The first step in building the instrument set versus non instrument set classifier was to collect training and testing data. After this was completed, we performed numerous experiments to determine the best parameters for the classifier.

3.4.1 Data Collection

This part of the project required a wide variety of training data to model the two classes. We used sound from many sources: the Internet, multiple CDs, and non-professional recordings. The audio was sampled at 16 kHz used 16 bits per sample and was stored in AU file format. This format compressed the 16-bit data into 8-bit mu-law data.

The instrument set class was trained with sound from each of the eight musical instruments. The data for this class is described in Table A.1 in the Appendix. There were 1874.6 seconds of training data and 171 seconds of test data for this class. The data was evenly split between the eight instrument classes. In addition, the training and test data was acquired from two different CD sets, so we were able to test that the classifier was not specific to one set of recording conditions or to one set of instruments.

The non instrument set class was trained with sound from a variety of sources. The data for this class is described in Tables A.2 and A.3 in the Appendix. There were 1107.4 seconds of training data and 360 seconds of test data. It included the following sound types: animal sounds, human sounds, computer sound effects, non-computer sound effects, speech, solo singing, the eight solo instruments with background music, other solo instruments, and various styles of non-solo music with and without vocals.

There are a number of reasons we chose this training and test data for the non instrument set:

- We used a variety of sounds so that the non instrument model would be general.
- The eight instrument sounds with background music were included in the non instrument set. This decision relied on our interpretation of ‘instrument set’. We

decided that the instrument set class should only include solo music, as our eight-instrument classifier had been trained only with solo instruments. If our classifier worked correctly with non-solo music, then it would have been more appropriate to include sound with background music in our instrument set class.

- The test data included some sounds not represented in the training set, such as the owl sound. This was done in order to test that the class model was sufficiently general.
- The training and test data were composed of sounds recorded under many different conditions. Therefore, we were able to test that the classifier was not specific to one set of recording conditions.

3.4.2 Experiments

We performed a variety of experiments in order to identify the best classification algorithm for the identification system. The system is outlined in Figure 3.4. In each experiment, we used a segment length of 0.2 seconds, the mel cepstral feature set, and 16 feature coefficients.

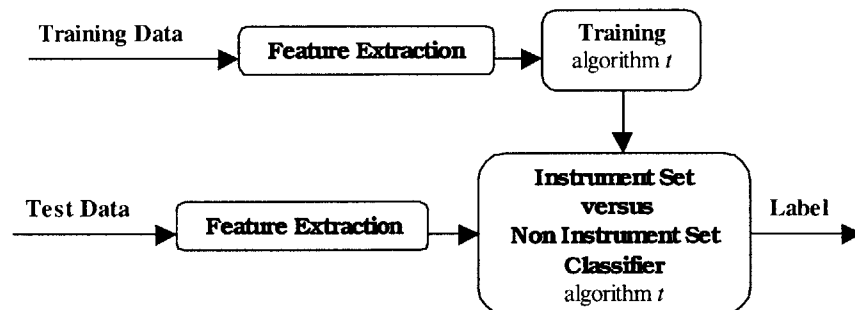


Figure 3.4 Instrument set versus non instrument set classification system.

Support Vector Machines

We used a two-category SVM classifier. The two categories were instrument set versus non instrument set. Researchers have achieved good results with SVMs in similar problems, such as face detection. In one study, using the classes: face and non-face, the detection rate was 74.2% (Osu98).

Gaussian Mixture Models

We built two types of GMM classifiers, two-category and nine category. For each classifier, each category was modeled using a Gaussian mixture model. Based on the numerous studies demonstrating that SVMs yield better results than GMMs, we predicted similar results in our system.

Two Class

The two classes were instrument set and non instrument set. Given a test sound, the class with the highest probability was chosen. A test vector was then labeled as the class with the largest probability.

Nine Class

The nine classes were non instrument set, bagpipes, clarinet, flute, harpsichord, organ, piano, trombone, and violin. A test vector was labeled as 'other' if the non instrument class had the largest probability. Otherwise, it was labeled as an instrument set member. We thought that the two-class method would work better because it provided a more general instrument model. The two-class model is more general because it used one GMM model to represent all of the instruments, while the nine class method used a separate GMM model for each instrument. A more general instrument model is beneficial because the test set used in these experiments was quite different from the training set; it used different instances of the musical instruments and different recording conditions.

Probability threshold

The probability threshold classifier worked in the following manner: given a test vector \bar{x} , we calculated $p(C | \bar{x})$ for each of the eight instrument classes. If $\max_c(p(C | \bar{x})) < T_c$, then the test sound was labeled as 'other'. Otherwise, it was labeled as an instrument set sound. For example, if a test vector \bar{x} is most likely a clarinet, and $p(\text{clarinet} | \bar{x})$ was not greater than the clarinet threshold, then the vector was labeled as

‘other’. If $p(\text{clarinet} | \bar{x})$ was greater than the clarinet threshold, then the vector was labeled as an instrument set sound.

The probability threshold for an instrument C was calculated as follows:

1. The training data was separated into two groups. The first group contained the training data that belonged in class C , and the second group contained the training data that did not belong in class C . For example, if class C was flute, then the non-flute group would include all of the non instrument set data plus bagpipes, clarinet, harpsichord, organ, piano, trombone, and violin data.
2. For every vector \bar{x} in either training set, we calculated the probability that it is a member of class C , $p(C | \bar{x})$, using the Gaussian mixture model for class C . For example, the non-flute vectors would have lower probabilities than the flute vectors.
3. The threshold T_C was the number that most optimally separated the two training data groups. An example is shown in Figure 3.5. In this example, T_C separates the flute probabilities from the non-flute probabilities. The x symbols represent the flute probabilities and the o symbols represent the non-flute probabilities.

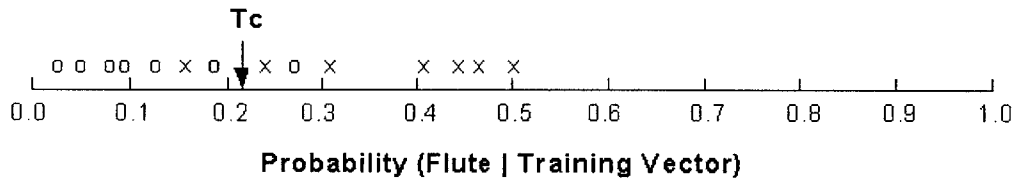


Figure 3.5 The threshold T_C optimally separates the flute probabilities (x) from the non-flute probabilities (o).

In each experiment, we used 2214 seconds of training data. The test data contained 179 seconds of instrument set samples and 361 seconds of non instrument set samples. For each experiment, we examined a 2x2 confusion matrix,

$$\begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix}.$$

- r_{11} : the number of times the system classified an instrument set segment correctly.
- r_{12} : the number of times the system classified a non instrument set segment as an instrument set segment.
- r_{21} : the number of times the system classified an instrument set segment as a non instrument segment.
- r_{22} : the number of times the system classified a non instrument set segment correctly.

The overall error rate was computed as,

$$1 - \frac{r_{11} + r_{22}}{r_{11} + r_{12} + r_{21} + r_{22}}.$$

Chapter 4 Results and Discussion

4.1 Introduction

After performing experiments to determine the most effective classifiers, we built the automatic annotation system. This chapter first presents results for the entire system, and then discusses the results for each classifier in detail.

4.2 Automatic Annotation System

The automatic annotation system was tested using a 9.24 second sound file sampled at 16 KHz. The audio file contained sound from each of the eight instruments. It also contained sound from a whistle. Table 4.1 shows the contents of the audio file. The units are in samples.

The annotation system first divided the audio file into overlapping segments. The segments were 0.2 seconds long and the overlap was 0.19 seconds. Then each segment was classified using the two classifiers. Afterwards, each time sample was marked with the class that received the most votes. Lastly, the annotation system filtered the results and then generated the appropriate labels.

The filter first divided the audio file into overlapping windows 0.9 seconds long. We chose 0.9 seconds because all of the sound sections in the audio file were longer than 0.9

seconds. Then, the filter determined which class dominated each window. Each time sample was then labeled with the class with the most votes.

Sound	Start	End
Bagpipes	1	17475
Violin	17476	33552
Clarinet	33553	49862
Trombone	49863	66871
Other	66872	82482
Piano	82483	98326
Flute	98327	115568
Organ	115569	131645
Harpsichord	131646	147803

**Table 4.1 Contents of the audio file used for the final test.
The units are in samples.**

The results are shown in Figure 4.1. The figure contains the audio file waveform, the correct annotations, and the automatic annotations. The error rate for the audio file was 22.4%. We also tested the system without the filter. In that experiment, the error rate was 37.2%.

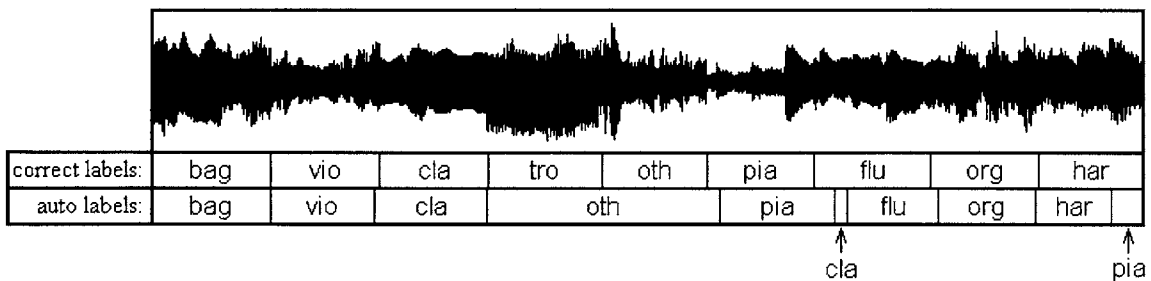


Figure 4.1 Final test results.

The annotation system performed poorly in the trombone section. This was likely because section was generated with a bass trombone, and the system was trained with a tenor trombone.

The filter used in the annotation system decreased the error rate substantially. However, in order for the filter to work optimally, each label in the audio file must have had a duration larger than the window length.

The results for the two sound classifiers are discussed in detail in Sections 4.3 and 4.4. The classifiers used in the final annotation system are described below:

- The eight instrument classifier used the SVM (one versus all) algorithm and the following parameter settings: 1638 seconds of training data, the mel cepstral feature set, a 0.2 second segment length, and 16 feature coefficients per segment. The error rate for this classifier was 30%. The instrument error rates were approximately equal, except for the trombone and harpsichord error rates. The trombone error rate was 85.7% and the harpsichord error rate was 76.5%. The trombone error rate was high because the classifier was trained with a tenor trombone, and tested with a bass trombone. We believe that the harpsichord accuracy was low for similar reasons.
- The instrument set versus non instrument set classifier used the GMM (two class) algorithm and the following parameter settings: 2214 seconds of training data, the mel cepstral feature set, a 0.2 second segment length, and 16 feature coefficients per segment. The error rate for this classifier was 24.4%. The test set contained some sound types that were not in the training set. The system classified the new types of sounds with approximately the same accuracy as the old types of sounds.

4.3 Eight-Instrument Classifier

In order to find the most accurate eight-instrument classifier, many experiments were performed. We explored the feature set space (harmonic amplitudes, LPC, cepstral, and mel cepstral), the classifier space (SVM and GMM), and various parameters for each classifier.

4.3.1 Gaussian Mixture Model Experiments

We first explored GMM classifiers. In addition to feature set type, the following parameters were examined: number of Gaussians, number of feature coefficients, segment length, and test data set.

Feature Set Type

First, we performed an experiment to find the best feature set. We examined harmonic amplitudes, linear prediction coefficients, cepstral coefficients, and mel cepstral coefficients. The other parameter values for this experiment are listed in Table 4.2.

Data Set for Training	CD Set 1	Classification Algorithm	GMM
Data Set for Testing	CD Set 1	Number of Gaussians	8
# Training Segments	4096	Feature Set Type	---
Training Segment Length	0.1 sec	# Feature Coefficients	16

Table 4.2 Parameters for feature set experiment (GMM).

The mel cepstral feature set gave the best results, overall error rate of 7.9% classifying 0.1 sec of sound. Figure 4.2 shows the results. The error rates were low because the training and test data were recorded under the same conditions. This is explained in more detail in the results section of the test data set experiment.

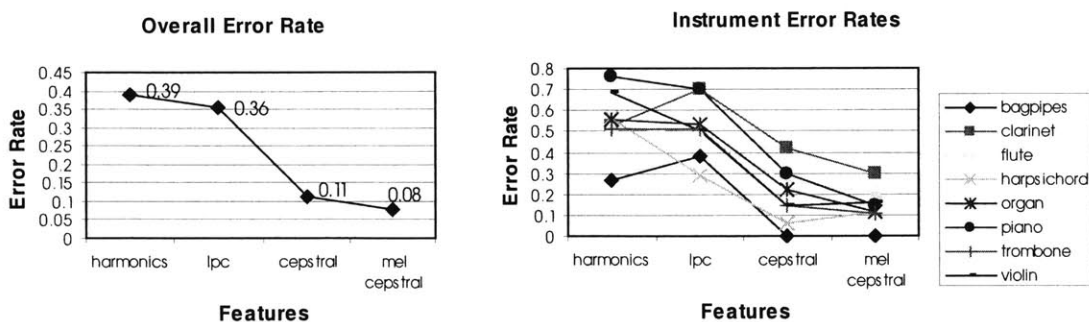


Figure 4.2 Results for feature set experiment (GMM).

The harmonic feature set probably had the worst performance because the feature set was not appropriate for piano, harpsichord, and organ. These instruments are capable of

playing more than one note at a time, so it is unclear which note's harmonics are appropriate to use.

The cepstral set probably performed better than the linear prediction features because musical instruments are not well represented with the linear prediction all-pole model. Further improvements were seen when using the mel cepstral. This was expected since the frequency scaling used in the mel cepstral analysis makes the music's frequency content more closely related to a human's perception of frequency content. This technique has also improved speech recognition results (Rab93).

The instrument error rates generally followed the same trend as the overall error rate with one exception. The harmonics error rate for bagpipes, clarinet, and trombone was lower than the LPC error rate because the harmonic amplitude feature set was better defined for those instruments. Each of these instruments is only capable of playing one note at a time.

In summary, the mel cepstral feature set yielded the best results for this experiment.

Number of Gaussians

In this experiment, we tried to find the optimal number of Gaussians, 1, 2, 4, 8, 16, or 32. The other parameter values for this experiment are listed in Table 4.3.

Data Set for Training	CD Set 1	Classification Algorithm	GMM
Data Set for Testing	CD Set 1	Number of Gaussians	---
# Training Segments	8192	Feature Set Type	mel cepstral
Training Segment Length	0.1 sec	# Feature Coefficients	16

Table 4.3 Parameters for number of Gaussians experiment (GMM).

We achieved the best results using 32 Gaussians with an overall error rate of 5%. Figure 4.3 shows the results.

As the number of Gaussians was increased, the classification accuracy increased. A greater number of Gaussians led to more specific instrument models. The lowest error

rate occurred at 32. However, since the decrease in the error rate from 16 to 32 was small, we considered 16 Gaussians optimal because of computational efficiency.

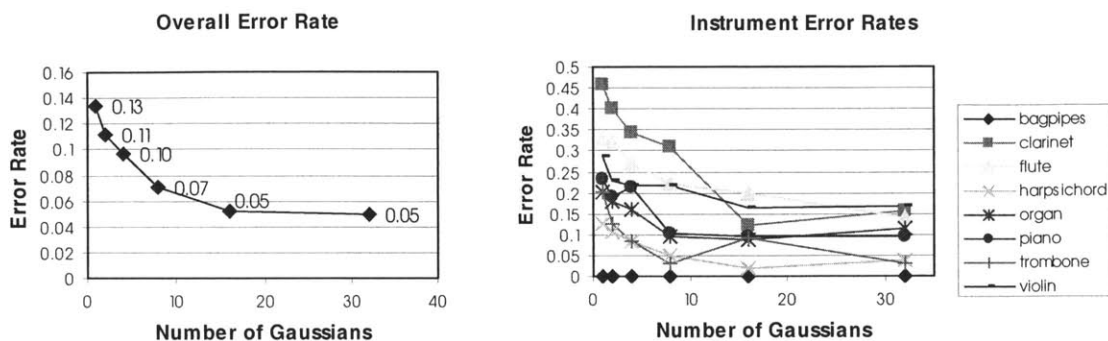


Figure 4.3 Parameters for number of Gaussians experiment (GMM).

The instrument error rates followed the same trend as the overall error rate. In summary, 16 Gaussians yielded the best results for this experiment. If more training data were used, then we would have probably seen a larger performance improvement when using 32 Gaussians.

Number of Feature Coefficients

In this experiment, we tried to find the optimal number of feature coefficients, 4, 8, 16, 32, or 64. The other parameter values for this experiment are listed in Table 4.4.

Data Set for Training	CD Set 1	Classification Algorithm	GMM
Data Set for Testing	CD Set 1	Number of Gaussians	16
# Training Segments	8192	Feature Set Type	mel cepstral
Training Segment Length	0.1 sec	# Feature Coefficients	---

Table 4.4 Parameters for number of feature coefficients experiment (GMM).

We achieved the best results using 32 coefficients per segment, overall error rate of 4.7%. Figure 4.4 shows the results.

The classifier became more accurate as we increased the number of coefficients from 4 to 32, since more feature coefficients increases the amount of information for each segment. However, there was a performance loss when we increased the number of features from

32 to 64 because there was not enough training data to train models of such high dimension.

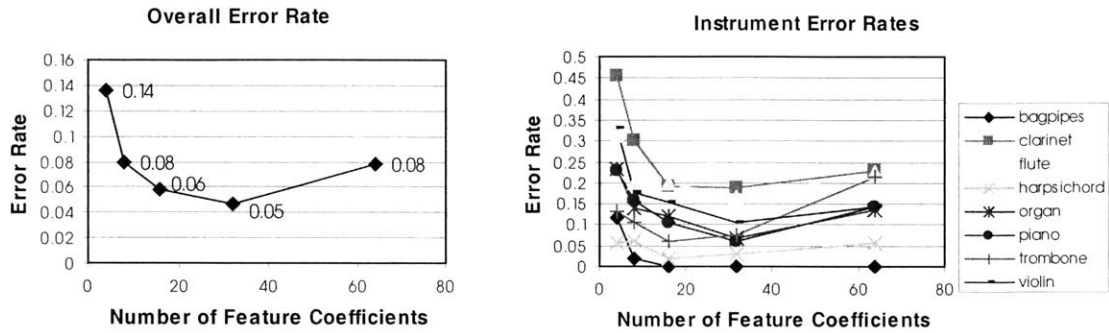


Figure 4.4 Results for number of feature coefficients experiment (GMM).

The instrument error rates generally followed the same trend as the overall error rate. In summary, a feature set size of 32 yielded the best results for this experiment.

Segment Length

In this experiment, we tried to find the optimal segment length, 0.05, 0.1, 0.2, or 0.4 sec. We kept the total amount of training data fixed at 1638.4. The other parameter values are listed in Table 4.5.

Data Set for Training	CD Set 1	Classification Algorithm	GMM
Data Set for Testing	CD Set 1	Number of Gaussians	16
# Training Segments	---	Feature Set Type	mel cepstral
Training Segment Length	---	# Feature Coefficients	16

Table 4.5 Parameters for segment length experiment (GMM).

We achieved the best results using 0.2 second segments, overall error rate of 4.1%. Figure 4.5 shows the results.

We did not expect segment length to substantially affect performance. An instrument's lowest frequency cannot be represented using less than 0.1 seconds of sound. Thus, the sharp performance decrease for 0.05 second segments was expected. The error rates at 0.1 and 0.2 seconds were approximately the same.

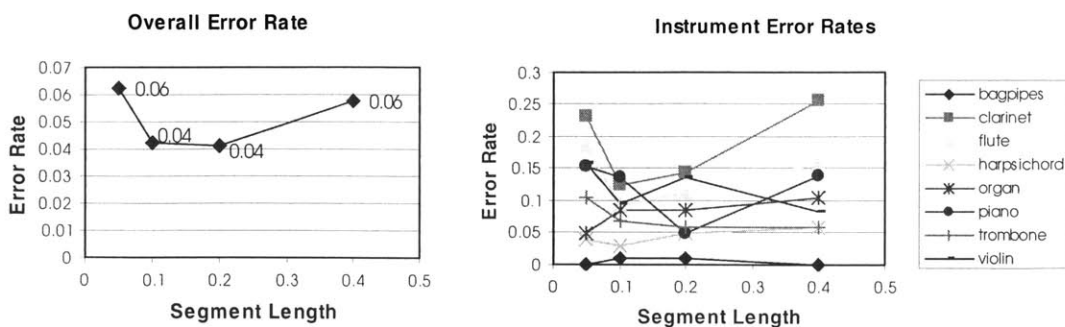


Figure 4.5 Results for segment length experiment (GMM).

We did not expect the error rate to increase at 0.4 seconds. One explanation for this behavior is average note duration. A 0.4 second segment is more likely to include multiple notes than a 0.2 second segment. In a preliminary experiment, an instrument classifier trained and tested on segments containing only one note performed better than an instrument classifier trained and tested using segments containing more than one note. Therefore, average note duration could explain why 0.2 second classifiers perform better than 0.4 second classifiers.

In summary, 0.2 second segments yielded the best results for this experiment.

Test Data Set

In our prior experiments, each instrument was trained and tested on music recorded under the same conditions using identical instruments. The lowest error rate achieved was 3.5%; the parameter values are shown in Table 4.6. In the data set experiment, we determined that the 3.5% error rate cannot be generalized to all sets of recording conditions and instrument instances.

Data Set for Training	CD Set 1	Classification Algorithm	GMM
Data Set for Testing	CD Set 1	Number of Gaussians	16
# Training Segments	16384	Feature Set Type	mel cepstral
Training Segment Length	0.2 sec	# Feature Coefficients	32

Table 4.6 Parameter set for the best GMM classifier when the same recording conditions are used for the training and test data.

We computed a more general error rate by testing our classifier with data recorded in a different studio with different instruments than that of the training data. The test data was taken from CD set 2. The trombone test data differed from the training data in one additional respect; the training data was recorded using a tenor trombone, and the test data was recorded using a bass trombone.

Using the new test data, the best overall error rate achieved was 35.3%. The parameter values are shown in Table 4.7.

Data Set for Training	CD Set 1	Classification Algorithm	GMM
Data Set for Testing	CD Set 2	Number of Gaussians	2
# Training Segments	8192	Feature Set Type	mel cepstral
Training Segment Length	0.2 sec	# Feature Coefficients	16

Table 4.7 Parameter set for the best GMM classifier when different recording conditions are used for the training and test data.

In addition, we examined the effect of various parameters when using the new test data. The results of these experiments are shown in Figure 4.6.

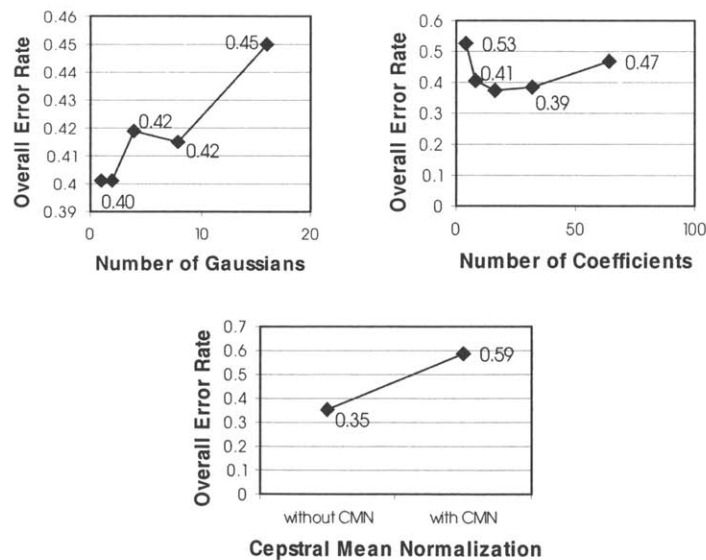


Figure 4.6 Results for GMM classifier experiments when different recording conditions were used for the training and testing data.

- *Number of Gaussians*: As the number of Gaussians was increased, the overall error rate generally increased. The opposite behavior occurred in our previous number of Gaussians experiment. Increasing the number of Gaussians leads to more specific instrument models. In this experiment, the test and training data are not as similar as in the previous experiment. Therefore, more specific instrument models had the opposite effect.
- *Number of Coefficients*: The classifier became more accurate as we increased the number of coefficients from 4 to 16. Increasing the number of feature coefficients increases the amount of information for each segment. There was a performance loss when we increased the number of feature coefficients from 16 to 64 because there was not enough training data to train models of such high dimension.
- *Cepstral Mean Normalization*: Recording differences in training and test data is a common problem in speaker identification and speech recognition. One method that has been used in speech to combat this problem is cepstral mean normalization (CMN). This method removes recording differences by normalizing the mean of the training and test data to zero (San96).

Assume that we have training data for n instruments where each data set is composed of s mel cepstral feature vectors, $\bar{x}_1 \dots \bar{x}_s$. First, we compute each instrument's feature vector mean, $\bar{\mu}_1 \dots \bar{\mu}_n$. Then we normalize each feature set by subtracting the corresponding mean vector. The test data is normalized in the same manner, except that the mean vector is derived using the test data.

This method actually decreased our classifier's performance by a substantial amount. Using CMN in addition to the 0.15 amplitude normalization may have caused the performance decrease. It is likely that the amplitude normalization removed any data recording differences that the CMN would have removed. The additional normalization most likely removed important information, which caused the error rate to increase.

In summary, our best overall error rate using Gaussian Mixture Models was 3.5%. However, this error rate cannot be generalized to all sets of recording conditions and instrument instances. Using GMMs, we computed 35.4% as the error rate for general test data.

4.3.2 Support Vector Machine Experiments

In these experiments, we explored SVM classifiers. We examined two multi-class classification algorithms and the error rate for general test data.

Multi-Class Classification Algorithm

This experiment examined two multi-class classification algorithms that are commonly used with SVMs, one-versus-all and one-versus-one.

The parameter values for this experiment are listed in Table 4.8, and the results are shown in Figure 4.7.

Data Set for Training	CD Set 1	Classification Algorithm	SVM
Data Set for Testing	CD Set 1	Multi-Class Algorithm	---
# Training Segments	8192	Feature Set Type	mel cepstral
Training Segment Length	0.1 sec	# Feature Coefficients	16

Table 4.8 Parameters for multi-class classification algorithm experiment (SVM).

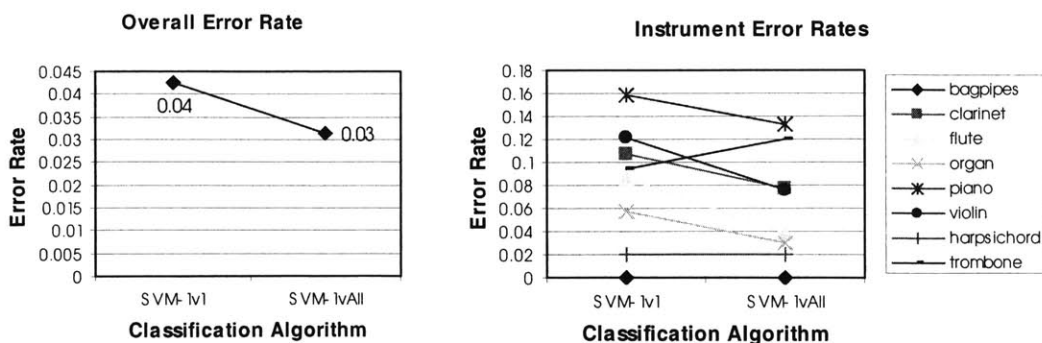


Figure 4.7 Results for multi-class classification algorithm experiment (SVM).

We achieved the best results using the one-versus-all algorithm, overall error rate of 3.1%. The instrument error rates followed the same trend as the overall error rate except for the trombone.

Test Data Set

The lowest error rate achieved using SVMs was 2%; the parameter values are shown in Table 4.9. In this experiment, we determined that the 2% error rate cannot be generalized to all sets of recording conditions and instrument instances.

Data Set for Training	CD Set 1	Classification Algorithm	SVM
Data Set for Testing	CD Set 1	Multi-Class Algorithm	1-vs-all
# Training Segments	8192	Feature Set Type	mel cepstral
Training Segment Length	0.2 sec	# Feature Coefficients	16

Table 4.9 Parameter set for the best SVM classifier when the same recording conditions are used for the training and test data.

We computed a more general error rate by testing our classifier with data recorded in a different studio with different instruments than that of the training data. The new test data was acquired from CD set 2.

Using the new test data, the best overall error rate achieved was 30.4%. The parameter values are shown in Table 4.10. This is a 5% improvement over the GMM classifier.

Data Set for Training	CD Set 1	Classification Algorithm	SVM
Data Set for Testing	CD Set 2	Multi-Class Algorithm	1-vs-All
# Training Segments	8192	Feature Set Type	Mel cepstral
Training Segment Length	0.2 sec	# Feature Coefficients	16

Table 4.10 Parameter set for the best SVM classifier when different recording conditions are used for the training and test data.

In summary, our best overall error rate using Support Vector Machines was 2%. However, this error rate cannot be generalized to all sets of recording conditions and instrument instances. Using SVMs, we computed 30.4% as the error rate for general test data. The Support Vector Machine classification algorithm was more successful than the GMM algorithm.

4.4 Instrument Set versus Non Instrument Set Classifier

In order to find the best instrument set versus non instrument set classifier, we performed a number of experiments. The classification algorithms examined were Gaussian Mixture Models, Support Vector Machines, and probability threshold.

4.4.1 Gaussian Mixture Model Experiments

As expected, the two-class GMM method performed better than the nine-class GMM method, 24.3% error versus 37.2% error. We believe that the two-class method provides a more general instrument model, which led to a classifier that can correctly identify instruments recorded in a variety of conditions.

4.4.2 Support Vector Machine Experiments

The SVM classifier had a 40.5% error rate. The two-class GMM classifier was 15.8% more accurate than the SVM classifier. However, SVMs have shown better results in similar problems, such as the face versus non-face classification problem (Osu98). Therefore, we believe that the accuracy of the SVM classifier can be improved. Adjusting the kernel function may improve the classifier.

4.4.3 Probability Threshold Experiments

The probability threshold classifier had a 36% error rate. The two-class GMM classifier was 11.7% more accurate than the threshold classifier.

In summary, the Gaussian mixture model two-class method gave the best results, 24.2% error. The Support Vector Machine method probably had the highest error rate because the kernel function was not adequate for this type of data set. Also, the probability threshold method did not work as well as the Gaussian method because the two probability sets associated with each class had too much overlap.

Chapter 5 Conclusions and Future Work

5.1 Conclusions

In this project, we proposed and implemented an automatic annotation system for audio files. The system included the labels: bagpipes, clarinet, flute, harpsichord, organ, piano, trombone, violin, and other. The annotation system was composed of two sound classifiers, the eight-instrument classifier and the instrument set versus non instrument set classifier. We explored many of the system's parameters including classification algorithm, feature type, number of feature coefficients, and segment length.

Our most successful eight-instrument classifier was 70% correct when classifying 0.2 seconds of audio. It used 16 mel cepstral coefficients as features and employed the Support Vector Machine classification algorithm with the one versus all multi-class algorithm.

Our best instrument set versus non instrument set classifier was 75% correct when classifying 0.2 seconds of audio. It used 16 mel cepstral coefficients as features and employed a Gaussian mixture model classification algorithm.

5.2 Future Work

The annotation system can be improved in three respects: (1) Make the system more accurate. (2) Add the capability to classify concurrent sounds. (3) Add more labels, such as animal sounds or musical styles.

5.2.1 Accuracy Improvements

We can increase our system's accuracy by improving the two classifiers and by adding segmentation to the system.

Improving the Eight-Instrument Classifier

The eight-instrument classifier can be improved by minimizing the recording difference problem. This is also a common problem in speech recognition. Speech recognition results are much better when the same person's voice is used for training and testing. There are three common methods for combating this problem: (1) The training data should contain a variety of recording conditions. In this study, only one CD set was used for training. (2) Signal processing methods, such as cepstral mean normalization, should be employed. As discussed in Section 4.3.1, this approach was not successful in our classifier. Other spectral shaping approaches may be more successful. (3) The acoustic mismatch between the test data and the class models can be decreased by using models of the recording "noise" and by using "clean" instrument models (San96).

The accuracy of the eight-instrument classifier can also be increased by improving the feature set or instrument model. Two common feature set types that we did not explore are acoustic characteristic feature sets and temporal feature sets. These were discussed in Section 2.2. In addition, a model commonly used in speech recognition is the Hidden Markov Model (Rab93).

Improving the Instrument Set Versus Non Instrument Set Classifier

The instrument set versus non instrument set classifier can also be improved by minimizing the recording difference problem. The error rate could also be reduced by improving the feature set or by adjusting the classification algorithm. An acoustic characteristic feature set may be more effective for this classifier. Also, better results could be achieved by adjusting the SVM kernel.

Adding Segmentation to the Annotation System

Segmentation may improve the accuracy of the annotation system. A segmentation component would automatically divide the audio file into uniform sections before classification occurred. For example, a uniform section would contain only clarinet music or only flute music. In some speech recognizers, audio segmentation is used to separate speech into phonetic components. There are three popular segmentation methods used in speech recognition: sequential likelihood test, smooth derivative operator, and clustering. Researchers have achieved the best results using cluster-based segmentation (Ebe96).

The clustering algorithm begins by assuming an initial set of uniform regions. Then, the distance between pairs of neighboring regions is computed, and the pair with the smallest distance is combined to form a new uniform region. This process is iterated until the distance between all consecutive regions exceeds some threshold. The result is an audio file divided into uniform regions. The regions can then be labeled using our two sound classifiers (Ebe96).

5.2.2 Classification of Concurrent Sounds

Currently the annotation system cannot classify sounds that occur simultaneously. For example, it cannot distinguish between a clarinet and a flute being played concurrently.

There has been a great deal of work in perceptual sound segregation. Researchers believe that humans segregate sound in two stages. First, the acoustic signal is separated into

multiple components. This stage is called auditory scene analysis (ASA). Afterwards, components that were produced by the same source are grouped together (Bre90).

There has not been much progress in automatic sound segregation. Most systems rely on knowing the number of sound sources and types of sounds. However, some researchers have attempted to build systems that do not rely on this data. One group successfully built a system that could segregate multiple sound streams, such as different speakers and multiple background noises (Guy94).

It is important to include a sound segregation system in any automatic annotation system. Concurrent sounds could be separated using such a system, and then classifiers could label the individual sound streams.

5.2.3 Increasing the Number of Labels

We can also improve the annotation system by adding more labels to the system's vocabulary. For example, the system could label animal sounds with the animal's name rather than the label 'other'. Also, by adding speech recognition capabilities, we could include the transcription of any speech regions that may occur in an audio file.

We believe that it is possible to build an annotation system that can automatically generate descriptive and accurate labels for any sound file. Once this occurs, it will no longer be difficult to search audio files for content.

Appendix

INSTRUMENT SET				
	TRAINING DATA		TEST DATA	
BAGPIPES	CD: The Bagpipes & Drums of Scotland , Laserlight.	225 sec	CD: The bagpipe, Koch.	20 sec
CLARINET	CD: 20 th Century Music for Unaccompanied Clarinet, Denon.	225 sec	CD: Lonely souls, Opus.	20 sec
	CD: Mozart clarinet concerto, Richard Stoltzman.	17 sec		
FLUTE	CD: Manuela plays French Solo Flute Music, BIS.	225 sec	CD: Hansgeorg Schmeiser Plays Music for Solo Flute, Nimbus Records.	20 sec
	CD: Mozart Concerto for Flute and Orchestra, Jean-Pierre Rampal.	8 sec	CD: Bach Flute Sonatas, Maxence Larrieu.	11 sec
	CD: Nielson Concerto for Flute and Orchestra, Julius Baker.	7.2 sec		
	Non-professionally recorded.	5.4 sec		
HARPSICHORD	CD: Bach Goldberg Variations, Sine Qua Non.	225 sec	CD: 20 th Century Harpsichord Music, vol. III, Gasparo.	20 sec
ORGAN	CD: Organ Works, Archiv.	225 sec	CD: Romantic French Fantasies, Klavier.	20 sec
PIANO	CD: Chopin Etudes, London, and Chopin Ballades, Philips.	225 sec	CD: The Aldeburgh Recital, Sony.	20 sec
TROMBONE	CD: Christian Lindberg Unaccompanied, BIS.	225 sec	CD: David Taylor, New World.	20 sec
VIOLIN	CD: Bach Works for Violin Solo, Well Tempered.	225 sec	CD: Sonatas for Solo Violin, Orion.	20 sec
	CD: J.S. Bach, Sonaten and Partiten, Itzhak Perlman.	37 sec		

Table A.1 Training and test data for the instrument set class.

NON INSTRUMENT SET (PART 1 OF 2)				
TRAINING DATA		TEST DATA		
ANIMALS	Internet: Bird.	2.4 sec	Internet: Birds.	1.8 sec
	Internet: Cat.	1.2 sec	Internet: Crows.	2 sec
	CD: Cat, Speed of Sound	0.4 sec	Internet: Duck.	2 sec
	Internet: Chicken.	3.2 sec	Internet: Geese.	0.4 sec
	Internet: Cow.	1.6 sec	Internet: Goose.	2 sec
	CD: Cow, Speed of Sound.	1.4 sec	Internet: Horse.	1 sec
	Internet: Dog.	3.2 sec	Internet: Owl.	2 sec
	CD: Dog, Speed of Sound.	2.4 sec	Internet: Panther.	2.4 sec
	Internet: Frog.	1 sec	Internet: Seagulls.	2 sec
	CD: Frog, Speed of Sound.	3.4 sec	Internet: Turkey.	2.4 sec
	Internet: Goat.	4.2 sec		
	Internet: Lion.	5 sec		
	CD: Lion, Speed of Sound.	1.8 sec		
	CD: Monkey, Speed of Sound.	3.2 sec		
	Internet: Pig.	4 sec		
	CD: Pig, Speed of Sound.	3.2 sec		
Internet: Sheep.	2.4 sec			
CD: Sheep, Speed of Sound.	2 sec			
HUMAN SOUNDS	Internet: Belch.	3.2 sec	Internet: Falling Scream	2.4 sec
	Internet: Burp.	0.6 sec	Internet: Flatulence.	1 sec
	Internet: Laugh.	4 sec	Internet: Laughing and coughing.	1 sec
	Internet: Baby laugh.	4.6 sec	Internet: Long laugh.	1.2 sec
	Internet: "Bart Simpson" laugh.	1.4 sec	Internet: Moaning.	1.2 sec
	Internet: "Beavis" laugh.	2.6 sec	Internet: "Nelson" laugh.	1.2 sec
	Internet: "Krusty" laugh.	1.6 sec	Internet: Noisy kiss.	0.6 sec
	Internet: Spooky laugh.	1.6 sec	Internet: Scream and moaning.	1.2 sec
Internet: Sneezing.	3 sec			
Internet: "The 3 Stooges" noise.	1.6 sec			
COMPUTER SOUND EFFECTS	Internet: Short beeps.	4 sec	Internet: Video game sounds.	13 sec
	Internet: Synthesized fluttering.	5 sec	Internet: High pitch robot voice.	1.6 sec
	CD: Phone, Speed of Sound.	4 sec	Internet: "Borg" voice.	4 sec
	Internet: Synthesized pulsing	3.4 sec		
	Internet: Synthesized warbling.	1 sec		
	Internet: Tones.	5.8 sec		
	Internet: Video game.	11 sec		
	Internet: "Cylon" voice.	1.6 sec		
Internet: Synthesized voice.	6.4 sec			
NON-COMPUTER SOUND EFFECTS	Internet: Airplane.	12 sec	Internet: Breaking glass.	0.4 sec
	CD: Balloon, Speed of Sound.	2 sec	Internet: Engine rumble.	2.8 sec
	CD: Bell, Speed of Sound.	4 sec	Internet: Gunshot.	5.2 sec
	Internet: Clown noises.	4.6 sec	Internet: Jet engine.	5 sec
	Internet: Compressed air horn.	4.8 sec	Internet: Motorcycle.	4.8 sec
	Internet: Gavel hit.	2.4 sec.	Internet: Motor running.	4.8 sec
	CD: Handsaw, Speed of Sound.	4 sec	Internet: Objects breaking.	5 sec
	CD: Horses, Speed of Sound.	7.4 sec	Internet: Shower.	4.8 sec
	Internet: Motorcycle.	5 sec	Internet: Squeaky door.	1.8 sec
	Internet: Page turning	1 sec	Internet: Squeaky sound.	1 sec
	Internet: Ocean waves.	3.2 sec		
	Internet: Running water.	5 sec.		
	Internet: Steam whistle	5 sec		
	CD: Storm, Speed of Sound.	5.8 sec		
CD: Tapping, Speed of Sound.	0.4 sec			
Internet: Whistle.	5.4 sec			

Table A.2 Training and test data for the non instrument set class (part 1 of 2).

NON INSTRUMENT SET (PART 2 OF 2)				
TRAINING DATA			TEST DATA	
SPEECH	Internet: Female, with back noise.	25 sec	Internet: Female, no background.	4 sec
	Internet: Male, with back music.	15 sec	Internet: Male, with back music.	5 sec
	Internet: Male, no background.	28 sec	Internet: Male, with back noise.	5 sec
SOLO SINGING	Internet: Female.	22 sec	Internet: Female.	10 sec
	Internet: Male.	22 sec	Internet: Male.	10 sec
THE SOLO INSTRUMENTS WITH BACKGROUND MUSIC	CD: The Bagpipes and Drums of Scotland, Laserlight.	20 sec	CD: Concertos for Trombone, Claves.	20 sec
	CD: The Virtuoso Clarinet, Vanguard.	20 sec		
	CD: The Flute, Laserlight.	20 sec		
	CD: Les Amoureux de l'Orgue, Ambassador.	20 sec		
	CD: David Barillan, Audiofon.	20 sec		
	CD The Violin, Laserlight.	20 sec		
OTHER SOLO INSTRUMENTS	Non-professional recording: Baritone.	30 sec	CD: Bassoon, Mozart Bassoon Concerto, Danny Bond.	17 sec
	Non-professional recording: Bassoon.	57 sec	Non-professional recording: Cello.	19 sec
	CD: Cello, Bach, The Suites for Cello, Carlos Prieto.	30 sec	Non-professional recording: Horn.	18 sec
	CD: English Horn, Vincent Persichetti's Concerto for English Horn and String Orchestra.	36 sec	CD: Viola, Hindemith Sonatas for Viola Alone, Kim Kashkashian.	6.2 sec
	CD: Horn, Mozart Rondo for Horn and Orchestra, Dale Clevenger.	30 sec		
	Non-professional recording: Viola.	30 sec		
NON SOLO MUSIC	Internet: Blues with vocals	30 sec	Internet: Blues with vocals	10 sec
	Internet: Blues without vocals.	30 sec	Internet: Blues without vocals.	10 sec
	Internet: Classical without vocals.	34 sec	Internet: Classical without vocals.	9.8 sec
	Internet: Country with vocals.	29 sec	Internet: Country with vocals.	9.4 sec
	Internet: Gospel with vocals.	27 sec	Internet: Gospel with vocals.	9.8 sec
	Internet: Jazz with vocals.	26 sec	Internet: Jazz with vocals.	9.8 sec
	Internet: Jazz without vocals.	28 sec	Internet: Jazz without vocals.	10 sec
	Internet: Opera with vocals.	23 sec	Internet: Opera with vocals.	9.8 sec
	Internet: Rap with vocals.	43 sec	Internet: Rap with vocals.	9.8 sec
	Internet: Rock with vocals.	26 sec	Internet: Rock with vocals.	10 sec
	Internet: Rock without vocals.	19 sec	Internet: Rock without vocals.	9.8 sec
	Internet: Swing with vocals.	29 sec	Internet: Swing with vocals.	10 sec
	Internet: Swing without vocals.	21 sec	Internet: Swing without vocals.	10 sec
	Internet: Techno with vocals.	28 sec	Internet: Techno with vocals.	10 sec
	Internet: Techno without vocals.	20 sec	Internet: Techno without vocals.	10 sec
	Internet: World with vocals.	25 sec	Internet: World with vocals.	9.8 sec
Internet: World without vocals.	35 sec	Internet: World without vocals.	9.6 sec	

Table A.3 Training and test data for the non instrument set class (part 2 of 2).

References

- Bre90 A. S. Bregman. *Auditory Scene Analysis*. Cambridge, MA: MIT Press, 1990.
- Bro98 J. C. Brown. Musical Instrument Identification using Autocorrelation Coefficients. *Proceedings International Symposium on Musical Acoustics*, Leavenworth, Washington, 1998.
- Bur98 C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2), 1998.
- Dem77 A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data Using the EM Algorithm. *Journal of the Royal Society of Statistics*, 39(1), 1-38, 1977.
- Dud73 R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. New York: John Wiley & Sons, 1973.
- Ebe96 B. Eberman and W. Goldenthal. Time-Based Clustering for Phonetic Segmentation. Proceedings of the *International Conference on Spoken Language Processing*. Philadelphia, 1996.
- Gis94 H. Gish and M. Schmidt. Text-Independent Speaker Identification. *IEEE Signal Processing Magazine*, 18-32, October 1994.
- Guy94 G. J. Brown and M. Cooke. Computational Auditory Scene Analysis. *Computer Speech and Language*, 8, 297-336, 1994.
- Han98 K. Han, Y. Par, S. Jeon, G. Lee, and Y. Ha. Genre Classification System of TV Sound Signals Based on a Spectrogram Analysis. *IEEE Transaction on Consumer Electronics*, 44(1), 33-42, 1998.
- Kam95 I. Kaminskyj and A. Materka. Automatic Source Identification of Monophonic Musical Instrument Sounds. *IEEE International Conference On Neural Networks*, 1, 189-194, 1995.
- Kas95 K. Kashino, K. Nakadai, T. Kinoshita, and H. Tanaka. Application of

- Bayesian Probability Network to Music Scene Analysis. In *IJCAI95 Workshop on Computational Auditory Scene Analysis*, Quebec, August 1995.
- Kat89 H. Katayose and S. Inokuchi. The Kansei Music System. *Computer Music Journal*, 13(4), 72-7, 1989.
- Lea93 R. E. Learned and A. S. Willsky. *A Wavelet Packet Approach to Transient Signal Classification*. MIT Laboratory for Information and Decision Systems: Cambridge, MA, 1993.
- Lig98 W. Ligget and W. Fisher. "Insights from the Broadcast News Benchmark Tests". *DARPA Speech Recognition Workshop*, Chantilly, VA, February, 1998.
- Mar96 K. Martin. Automatic Transcription of Simple Polyphonic Music. *MIT Media Lab Perceptual Computing Technical Report #385*, July, 1996.
- Mar98 K. D. Martin and Y. E. Kim. Musical Instrument Identification: A Pattern-Recognition Approach. Presented at the *136th Meeting of the Acoustical Society of America*, Norfolk, VA, October, 1998.
- Mat96 The MathWorks, Inc. *Matlab Signal Processing Toolbox*. Natick, MA: The MathWorks, Inc., 1996.
- Mca93 S. McAdams and E. Bigand. Recognition of Auditory Sound Sources and Events. *Thinking in Sound: The Cognitive Psychology of Human Audition*, Oxford University Press: Oxford, 1993.
- Moo77 J. A. Moorer. On the Transcription of Musical Sound by Computer. *Computer Music Journal*, 1(4), 32-8, 1977.
- Mos96 J.C. Mossing. *Feature Selection Techniques for the Classification of Outdoor Sound Sources Based on Their Acoustic Emission Signatures*. PhD Thesis, University of Dayton, 1996.
- Osu98 E. Osuna. Applying SVMs to face detection. *IEEE Intelligent Systems*, 23-6, July/August, 1998.
- Pie83 J. R. Pierce. *The Science of Musical Sound*. New York: Scientific American Books, 1983.
- Rab93 L. Rabiner and B. Juang. *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice Hall, 1993.

- Rey95 D. A. Reynolds and R. C. Rose. Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models. *IEEE Transactions on Speech and Audio Processing*, 3(1), 72-83, 1995.
- San96 A. Sankar and C. Lee. A Maximum-Likelihood Approach to Stochastic Matching for Robust Speech Recognition. *IEEE Transactions on Speech and Audio Processing*, 4(3), 190-202, 1996.
- Sch97 E. Scheirer and M. Slaney. Construction and Evaluation of a Robust Multifeature Speech/Music Discriminator. In *Proceedings of ICASSP*, 1331-4, 1997.
- Sch98 B. Scholköpfung. SVMs – A Practical Consequence of Learning Theory. *IEEE Intelligent Systems*, 18-21, July/August, 1998.
- Swa98 M. Swain. Study completed at Compaq Computer Corporation, Cambridge, MA, March, 1998.
- Wol96 E. Wold, T. Blum, D. Keislar, and J. Wheaton. Content-Based Classification, Search, and Retrieval of Audio. *IEEE Multimedia Magazine*, 27-36, Fall 1996.