Evaluation of Various Features for Music Genre Classification with Hidden Markov Models

By David S. Petruncio, Jr.

and

Mark A. Hasegawa-Johnson

ECE 298/299

March , 2002

Abstract

This paper describes a music type recognizer that can be used for automatic information retrieval or pre-selection of stations for a digital radio. A new feature representation is proposed called MFSC (Mel-Frequency Spectral Coefficients). MFSC are basically a representation of the smoothed spectral envelope of a song. MFSC then could be used for discrimination by most classifiers, but in this case HMM (hidden markov models) are trained based on these features and used for comparison against the results of established feature representations. In each case, the first 20 coefficients were used to train a model for each genre Jazz, Rock, Classical, and Dance music with a database of 200. The models were then used to evaluate songs from a test database of 60 songs. The experiments show that MFSC outperformed MFCC (mel-frequency cepstral coefficients), LPC (linear predication coefficients), and LPCC (linear prediction cepstral coefficients). The MFSC HMM was able to classify with an accuracy of 88.3% for the best model parameters.

TABLE OF CONTENTS

PAGE

1. INTRODUCTION

1.1 Motivation

You get in your car, fasten your safety belt, adjust your mirrors and prepare to depart from your driveway to work, but before you do anything else, you turn on your radio. Within seconds you recognize that you are listening to a classical radio station and in fact it is Fanfare for the Common Man by Aaron Copland. As you meander through traffic you can't resist humming along and when the radio station fades out you are still able to hum the missing notes. Doesn't sound too exciting? But it is, if you consider the fact that the human brain probably did tera-flops worth or calculations just to recognize the type of music that was being played. It is hard to imagine the song title and composer were recognized as easy.

Music Recognition is a fairly new field of interest and is not as developed in theory as others. There has been much advancement in pitch recognition and transcription of single instrument performances into scores, but many times relevant tempo is lost or affected. Likewise when multiple notes or instruments are playing, differentiation becomes a hindrance and almost impossible to classify. Yet, there are methods to separate and characterize by instrument by using different features and modeling.

I will show that by using Hidden Markov Models and various features I can get better recognition of genre than a simple k-Nearest Neighbor Post Component Analysis on the Mel-Frequency Cepstral Coefficients. Preliminary tests showed that k-NN PCA produced less than fifty percent recognition for four distinct genres of music (Jazz, Rock, Classical, and Dance).

The five features I looked at were Linear Prediction Coefficients, Mel-Spectral Coefficients, Linear Prediction Cepstral Coefficients, Mel-Frequency Cepstral Coefficients, and the fusion of Mel-Frequency Cepstral Coefficients and Linear Prediction Cepstral Coefficients. The best model gave me a recognition rate of 88.3% from a training database of 200 songs against a test database of 60 songs.

1.2 Objectives

The Objective of this experiment was to evaluate different features for the classification of music by genre. A relatively large database of representative music was needed to cover the four genres of music that were going to be tested. Then, using HMMs, four models were each trained to a genre of music based on various features extracted from the music files. The five features in review were the Linear Prediction Coefficients, Mel-Spectral Coefficients, Linear Prediction Cepstral Coefficients, Mel-Frequency Cepstral Coefficients, and the fusion of Mel-Frequency Cepstral Coefficients and Linear Prediction Cepstral Coefficients.

Also, different HMMs parameters were varied to achieve optimum performance for the different features chosen. These parameters were the topology of the HMM, the number of states, the mixture of gaussians, the time-width of sequences, and the actual computation of the likelihood. Once the respective models had been trained, they were evaluated by the forward algorithm to obtain the likelihood for each song in the test database. The model that produced the most likely observation characterized the genre that song belonged to.

2 PREVIOUS WORKS

Lambrou, Kudmukis, Speller, Sandler, and Linney [10] did an exhaustive comparison on musical signal classification using various wavelet transformations and statistical pattern recognition techniques. Wavelet transformation is just another way to represent a time-domain signal as a summation of other simpler blocks, just like the Fourier transform uses cosines. Logarithmic Splitting, Uniform Splitting, and Adaptive Splitting were the three transforms looked at. First Order Statistics, Second Order Statistics, and the number or zero crossings were classified by a Minimum Distance Classifier, a Least Squares Minimum Distance Classifier, a k Nearest Neighbor Classifier, and a Quadrature Classifier. What they were able to show was that song samples from the genres of Rock, Jazz, and Piano were accurately classified 91.67% of the time. It should be noted that the database consisted of only 12 songs, 4 from each genre.

Soltau, Schultz, Westphal, and Waibel [14] did a comparison of HMM versus their ETM-NN (Explicit Time Modeling with Neural Network) method for classifying Rock, Pop, Techno, and Classic music. Using abstraction of acoustical events they were able to get 79.2% for HMM classification and 86.1% for their ETM-NN method. They also showed that a perceptual study of human classification had similar recognition and confusion as with the ETM-NN. In this study a much larger database was used. A total of 360 songs uniformly distributed over the four genres, in which 241 were used for training and 72 for testing.

In 1980, Davis and Mermelstein did a comparison of parametric representations for monosyllabic word recognition [4]. Four different parameter sets were looked at including MFCC (mel-frequency cepstral coefficients, LPC (linear prediction cepstrum), LPS (linear prediction spectrum), and reflection coefficients. Classification was done using a Euclidian distance for open and closed tests. What David and Mermelstein found was that the MFCC representation proved the best with a classification rate of 96.5% outperforming the other representations by at least 2-7%.

3. OBTAIN DATA

The discrimination of music is subject to many errors whether the discriminator be computer or human. The fact is that the categorization of music by genre is subjective and a song from predominately country band might sound like rock music. For this experiment the four genres looked at were Dance, Rock, Jazz, and Classical. The files that were selected needed to be robust and related to that particular genre. Songs were sampled at 44.1 kHz in mono that were low in noise and irregularities, i.e. talking, audience clapping. As with recognition by humans, these events are not helpful or can even lower the recognition rate.

Also, many of the files were chosen by ear and evaluated for their content. It was important to not pick songs by a particular band that was considered rock or jazz, but by actually listening to find commonalties between songs in a genre. Lastly, it was important to analyze segments of song that were representative of that genre. For example, when listening to a song the beginning and ending are not as helpful than middle verses or chorus. Likewise a refrain could throw the untrained listener off.
Once the database was chosen it was partitioned into a test and a training group. Preliminary tests showed that testing with the training group produced 100% recognition, so it was important to evaluate only new data.

4. FEATURE EXTRACTION

The main purpose of feature extraction is to reduce the overall size of the data and to eliminate unwanted noise. For example, calling a particular piece of music as "Jazz" is a type a compression. You can also say that there is a lot of trumpet and piano, but no heavy guitar, therefore it is "Jazz." This is a little more data, but it is still compressed and descriptive. There are many ways to look at the same solution and none is known to be better than the other. You can look at the song and classify it by the instruments that are being played or by the particular chord progression. For this experiment there isn't an established feature extraction that is preferred over others, that is why there are so many tested.

4.1 Principal Component Analysis

Principal Component Analysis [8] is widely used in many signal-processing applications ranging from face, speech, and bi-modal recognition for data analysis and compression. In other areas this is called the Karhunen-Loève transform, or the Hotelling transform. If we have n d-dimensional vectors

$$X = (x_1, x_1, ..., x_n)^T$$

with mean vectors

$$M = (\mu_1, \mu_1, ..., \mu_n),$$

the scatter maxtrix can be expressed as,

$$S_x = \sum_{k=1}^{n} (X - M)(X - M)^T .$$

We know that

$$S_x e_i = \lambda_i e_i, i = 1, 2, ..., n,$$

where e is the eigenvector and λ are the eigen values and can be found easily by available code. By ordering the eigen values we take the m largest values corresponding to however much signal power we want to retain. Thus the respective largest eigenvector are used to transform data to a new m-dimension vector. Let A be a matrix consisting of m largest eigenvectors of the covariance matrix as the row vectors. By transforming a data vector x, we get

$$y_k = A(x - \mu_x)$$

For example figure 4.1.1 shows how a scatter of data points in a 2-dimensional space can be transformed to a 1-dimensional space and perseve most of the expressivity. We can see the first eigenvector points along the direction with the most energy, where the second eigenvector is orthoganol to the first.

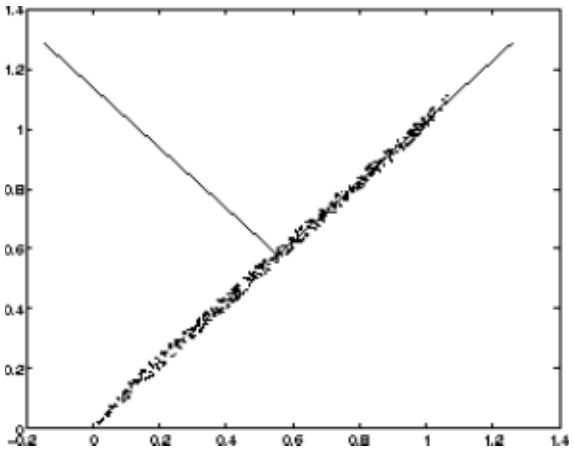


Fig. 4.1.1 Example of PCA from 2-D to 1-D

4.2 Mel-Frequency Spectral Coefficients

The short time Fourier spectrum is computed on 30 ms windows of song. In all of the cases, 30 ms Hamming windows with 50% overlap were used. The FFT of each window was taken.

$$X(w) = \int_{-\pi}^{\pi} x[n]e^{-jwn}dw$$

Then computing the squared magnitude, the power spectrum is integrated within overlapping critical band filter responses represented by the mel scale. The mel scale is based on human experimental data and is used to reduce frequency sensitivity over the original spectral estimate. The most commonly used approximation is a triangular window applied to the log of the power spectrum as shown here in figure 4.2.1.

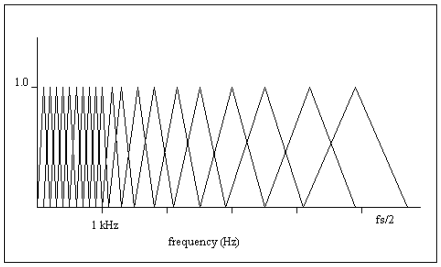


Fig. 4.2.1 Mel-Frequency Scaled Triangular Windows

The log is then applied after integration to compress the spectral amplitudes. Typically the IDFT is taken after this step to get the MFCC, but instead it is omitted. In figures 4.2.2-4.2.5 we can see the first 20 coefficients over ½ seconds for a typical song in each genre.

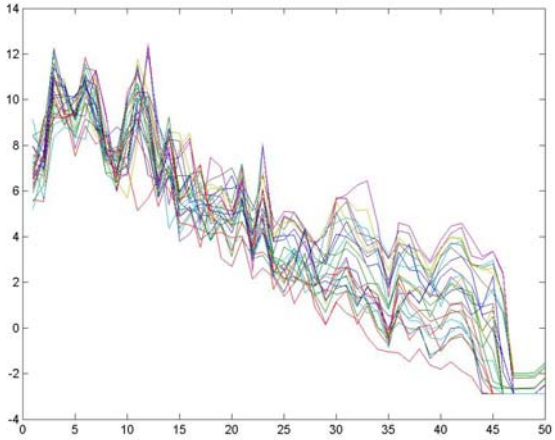


Fig. 4.2.2 MFSC of Classical

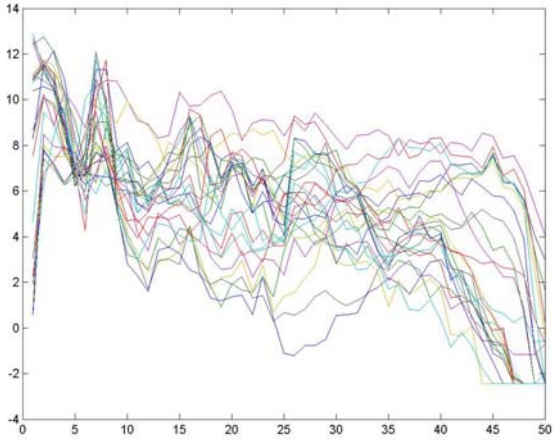


Fig. 4.2.3 MFSC of Dance

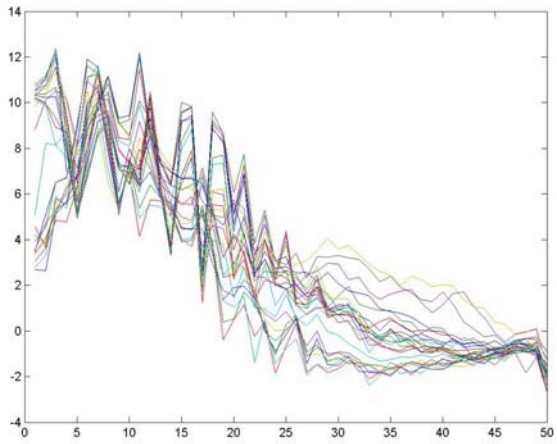


Fig. 4.2.4 MFSC of Jazz

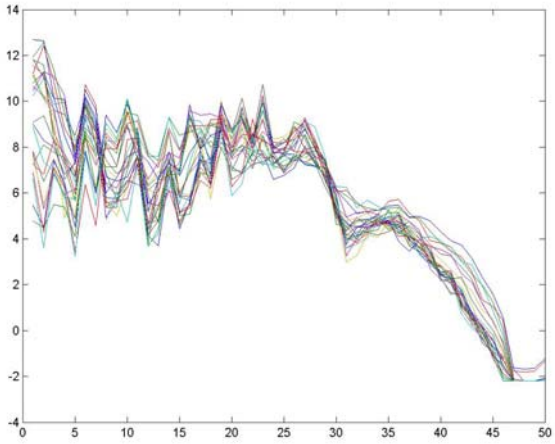


Fig. 4.2.5 MFSC of Rock

4.3 Mel-Frequency Cepstral Coefficients

The models for speech and music are mainly represented as an excitation that was passed through a system of resonators [6]. This can also be thought of as the convolution of the excitation with the impulse response of the system of resonators. Applying this to analysis of speech and music, one can separate the two signals, effectively performing what is called de-convolution. Cepstral analysis is a way to perform de-convolution and the equation for the cepstrum of a signal is given by,

$$c(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \log|X(w)|e^{jwn}dw,$$

where c (n) is called the nth cepstral coefficient. This equation is valid only when using real functions. For the complex case,

$$\hat{x}(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \log X(w)e^{jwn}dw,$$

The mel-frequency cepstrum coefficients are obtained by first computing the power spectrum of the windowed signal. This is done with 30 ms long Hamming windows, taking the FFT, and then computing the squared magnitude. After this has been done, the power spectrum is integrated within overlapping critical band filter responses represented by the mel scale. The mel scale is based on human experimental data and is used to reduce frequency sensitivity over the original spectral estimate. The most commonly used approximation is a triangular window applied to the log of the power spectrum as shown here in fig 4.2.1 courtesy of [4].

The log is then applied after integration to compress the spectral amplitudes and after this, the DFT is taken. Spectral smoothing is obtained by keeping the lower coefficients and truncating the higher. For example, Davis and Mermelstein obtained the MFCC computations by using

$$MFCC_i = \sum_{k=1}^{20} X_k \cos[i(k-\frac{1}{2})\frac{\pi}{20}] \qquad i = 1,2,\ldots, M$$

Where M is the number of cepstrum coefficients and X_k represents the output of the *k*th filter. In figures 4.3.1-4.3.4 we can see the first 20 coefficients over ½ seconds for a typical song in each genre.

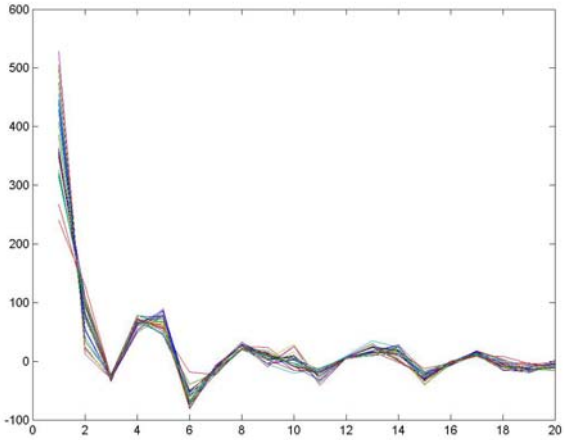


Fig. 4.3.1 MFFC of Classical

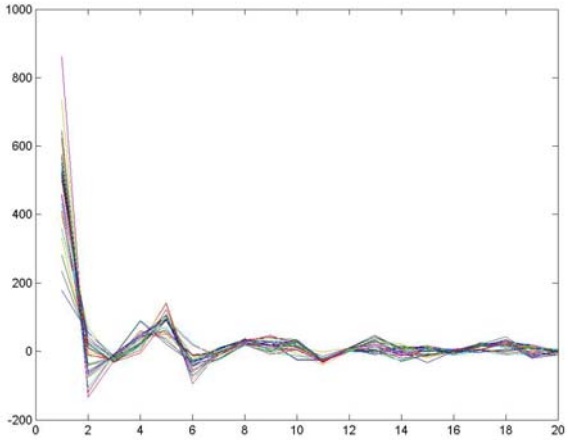


Fig. 4.3.2MFFC of Dance

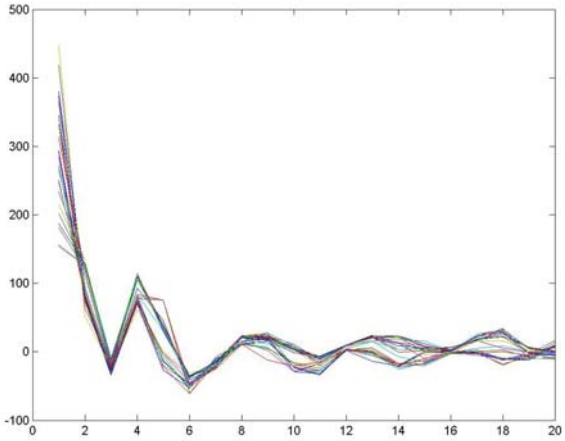


Fig. 4.3.3 MFFC of Jazz

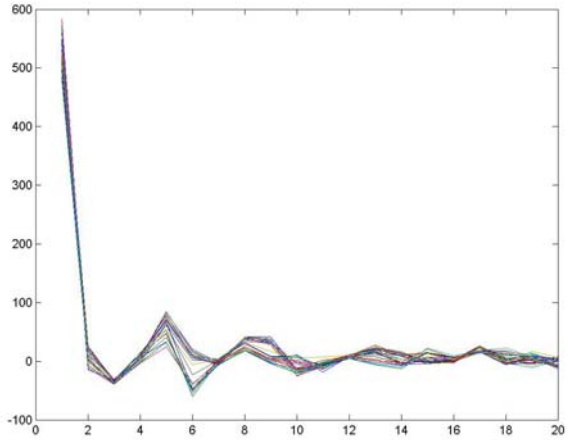


Fig. 4.3.4 MFFC of Rock

4.4 Linear Prediction Coefficients

Besides MFCCs, LPCs can also be used as a parametric data representation. The transfer function of a system of resonators can be represented as an all-pole model *H (z).* Using this information we can see that a response *y (n)* to an excitation *x (n)* can be predicted as a linear combination of previous values [6].

$$H(z) = \frac{1}{1 - \sum_{j=1}^{p} a_j z^{-j}} \qquad \hat{y}(n) = \sum_{j=1}^{p} a_j y(n-j) \quad \text{eq. [4.4.1]}$$

The error signal between the model output and the signal is just the difference of the predicted signal and the original, i.e. *e (n) = y (n) -* $\hat{y}(n)$. Defining the total error of the system as $D = \sum_{n=0}^{N-1} e^2(n)$ and taking the derivative with respect to *a*, we get *P* equations with *P* unknown variables. Solving for the variables a_j, we get the LPC. There are procedures known as the Levinson or the Durbin recursion that accomplish this with an order of P^2. However, these procedures need significant numerical precision in order to get proper values for the poles, which make the MFCCs more desirable at times. However there are many variants of LPC that have been used for speech recognition and one in particular, perceptual linear prediction (PLP), has proved highly effective.

As with the MFCC, the power spectrum is computed. Again, 30 ms long Hamming windows were used. The power spectrum is then integrated within overlapping critical band filter responses. In the PLP case, trapeziodally shaped filters at 1 Bark intervals are applied. The Bark [6] axis is derived from the from warping the frequency axis by this equation:

$$\Omega(w) = 6\ln\{\frac{w}{1200\pi} + [(\frac{w}{1200\pi})^2 + 1]^{0.5}\},$$

The spectrum is then pre-emphasized by an explicit weighting of the critical band spectrum and is referred to as the equal-loudness curve due to perceptual sensitivity at different frequencies. The spectral amplitudes are then compressed by taking the cube root. This reduces the amplitude variations for the spectral resonances. Finally the real part of the IDFT is taken before the Levinson-Durbin

recursion is used to find the coefficients. In figures 4.4.1-4.4.4 we can see the first 20 coefficients over ½ seconds for a typical song in each genre.

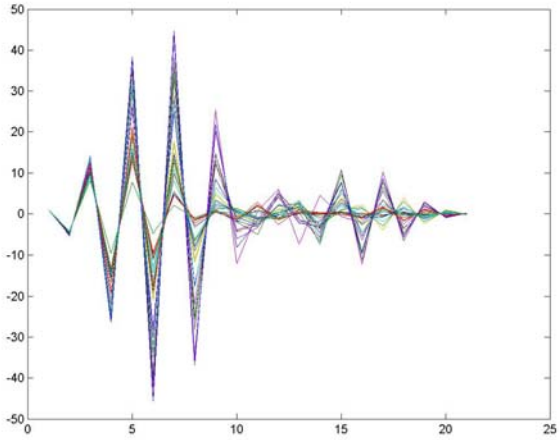


Fig. 4.4.1 LPC of Classical

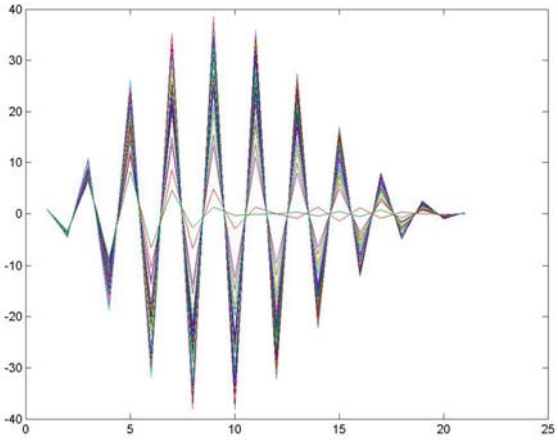


Fig. 4.4.2 LPC of Dance

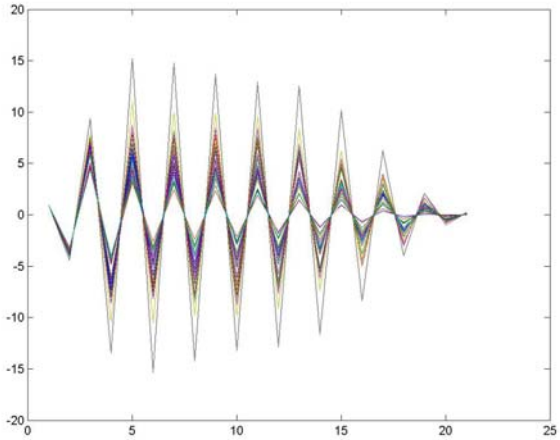


Fig. 4.4.3 LPC of Jazz

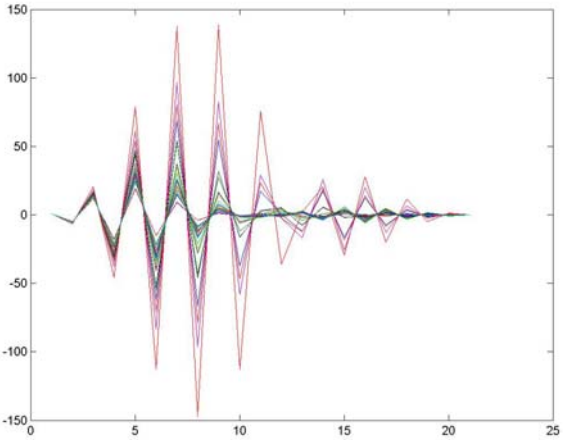


Fig. 4.4.4 LPC of Rock

4.5 Linear Prediction Cepstral Coefficients

The relation to the LPCC and the LPC is a very simple relationship. LPCC's can be extracted from a standard linear predictive model or a perceptual linear predictive model. In this case a standard linear predictive model was used where the coefficients a_k were obtained from equation 4.4.1 and the Levinson-Durbin recursion. The simple recursion can be seen in equation 4.5.1 [9].

$$c_1 = a_1$$
$$c_n = \sum_{k=1}^{n-1}(1-\frac{k}{n})a_k c_{n-k} + a_n, n = 1,...,P$$

In Figures 4.5.1 – 4.5.4 we can see the first 20 coefficients over ½ seconds for a typical song in each genre.

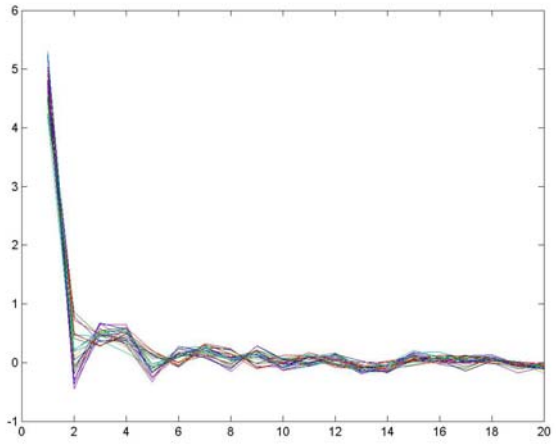


Fig. 4.5.1 LPCC of Classical

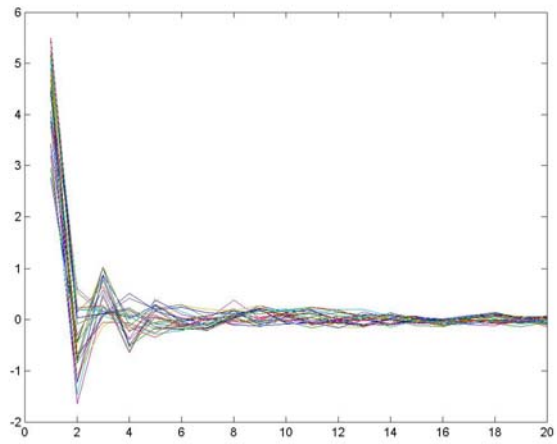


Fig. 4.5.2 LPCC of Dance

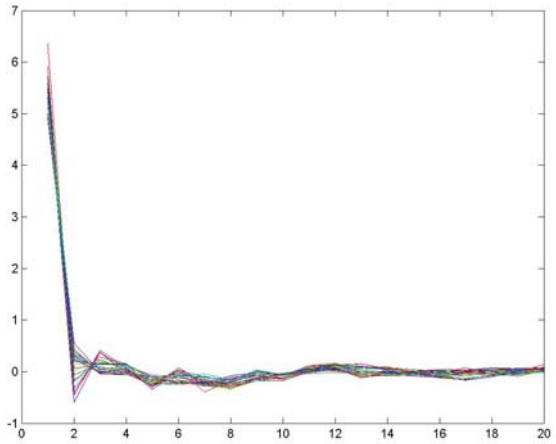


Fig. 4.5.3 LPCC of Jazz

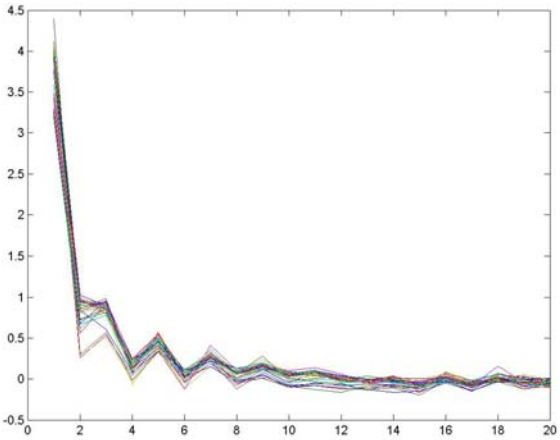


Fig. 4.5.4 LPCC of Rock

5. HIDDEN MARKOV MODEL

5.1 Algorithm

A hidden Markov Model consists of a hidden stochastic process that is not observable and can only be observed through an additional set of stochastic processes that determine an observation sequence. An observation sequence can be denoted as,

$O = O_1 O_2 O_3 .... O_T$, T is the length of the observation sequence,

and can be generated by these simple steps, but first one must become acquainted with some definitions that L.R. Rabiner and B.H. Juang have made readily available [1].

$Q = q_1 q_2 q_3 .... q_N$, N States

$V = v_1 v_2 v_3 .... v_M$, M Symbol observations

$A = \{a_{ij}\}, a_{ij} = \Pr(q_{j,t+1} \mid q_{i,t})$, State transition probability distribution

$B = \{b_j(k)\}, b_j(k) = \Pr(v_{k,t} \mid q_{j,t+1})$, Observation symbol probability transition in state j

$\pi = \{\pi_i\}, \pi_i = \Pr(q_{i,t})$, Initial state distribution

Rabiner and Juang have an iterative loop that generates an observation sequence given a model λ, where λ = HMM (A, B, π).

Choose initial state i_1 according to π;

t = 1;

While ($t < T$; t++){

Choose O_t according to b_{i_t} (k)

Choose i_{t+1} according to { $a_{i_t i_{t+1}}$ };}

5.2 Topology

Topology is another way describing the Transition Probability matrix A. By defining A, you restrict the movement within states. The most common topology is the Ergodic model [1]. In this topology you can freely from any state to any other. This can clearly be seen in Fig.5.2.1.

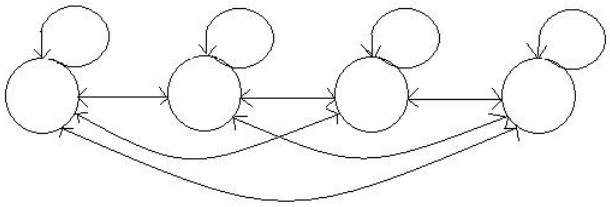


Fig. 5.2.1 Ergodic Topology

But typically, this is not the best representation to use for music recognition depending on what features you are using. If you are classifying by timbre, or shaping, the progression of music in a time-dependent one and for the most part does to loop back. In this case a Left-Right topology is often used [1]. In Fig. 5.2.2 we can see that the states either advance to the next state or stay at the current state.

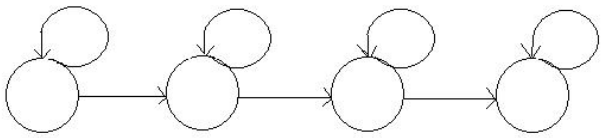


Fig. 5.2.2 Left-Right Topology

Lastly, what if you aren't classifying my instrument, but rather on content, i.e. musical progression. In this case, musical progression is time-dependent, but also repetitive with temporal structure. Songs will have a basic movement from verse to chorus and within these structures melodies are repeated. A different topology was used and can be seen in Fig. 5.2.3.

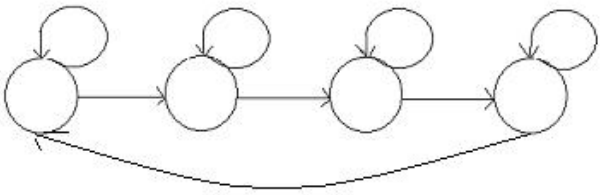


Fig. 5.2.3 Loop Topology

This topology allows for the last state to loop back allowing for repetition.

5.3 Training

Training is by far the most difficult step in the HMM procedure. This step is an iterative process that adjust the parameters λ = HMM (A, B, π) to maximize the probability of the observation sequence given the model. There is now way to solve analytically for the best parameters, but they can locally maximized using the Baum-Welch method.

First assign initial values for parameters of λ. Then use these parameters to re-estimate $A, B,$ and π in an interative process.

$$\bar{\pi}_i = \text{expected frequency in state } q_i$$

$$\bar{a}_{ij} = \frac{\text{expected number of transitions from } q_i \text{ to } q_j}{\text{expected number of transitions from } q_i}$$

$$\bar{b}_j(r_k) = \frac{\text{expected number times in state } q_i \text{ and oberserving } r_k}{\text{expected number times in state } q_j}$$

The values will converge to a local solution, which then can be used for testing and evaluation.

5.4 Testing

Now, to actually use this definition of a HMM in a recognition application, we must calculate the probability of observation sequence *O* given the model λ, i.e. *Pr (O|* λ *).* There are two algorithms discussed in the Rabiner and Juang paper [1] that accomplish this, the forward and the Viterbi algorithms. The forward algorithm finds a total likelihood estimate. The forward variable $\alpha_t(i)$, where $\alpha_t(i) = \Pr(O_1, O_2, ..., O_t, i_t = q_i \mid \lambda)$, can be calculated by,

$\alpha_1(i) = \pi_i b_i(O_i)$, $\quad 0 < i < N+1$

$\alpha_{t+1}(j) = \left[\sum_{j=1}^{N} \alpha_t(i) a_{ij} \right] b_j(O_{t+1})$, For t = 1,2…, T-1, $\quad 0 < j < N+1$

Then, Pr (O|λ) = $\sum_{i=1}^{N} \alpha_T(i)$. The analysis of this method shows us that it requires N^2T calculations, whereas direct calculating *Pr (O|* λ *)* directly requires $2TN^T$ calculations. The only setback to the forward algorithm is that it requires significant precision to accomplish due to the many multiply accumulates. On the other hand, the Viterbi approximation finds the single best state sequence.

$\delta_1(i) = \pi_i b_i(O_i)$ $\quad 0 < i < N+1$

$\delta_{t+1}(j) = \max_{0<1<N+1} [\delta_t(i) a_{ij}] b_j(O_{t+1})$ $\quad 0 < j < N+1$

Now that all additions have been eliminated, a logarithmic scale can be implemented and precision is no longer a significant problem.

6. RESULTS

6.1 Explanation of Chosen HMM Parameters

The first thing that should be brought to point for these results is the manner in which they were obtained. Initially I ran an exhaustive test on many parameters is particular ranges for just one set of features. I wanted to find a good range of parameters and keep them fixed throughout the rest of the experiments. I choose the MFSC and first looked at the 3 topologies. I shall refer to a set of parameter as H(Q,T,M) where Q is the number of states, T is the time-width looked at, and M is the mixture of gaussians. In table 6.1.1 we can see that the "Left-Right" topology had a better performance than the ergodic and proposed loop topology. It was then established for all experiments only a "Left-Right" topology would be used.

Table 6.1.1

Model	Left-Right	Ergodic	Loop
H(6, 45, 1)	40%	40%	20%
H(8, 45, 1)	55%	40%	20%
H(10, 45, 1)	65%	40%	40%

This next set of trials was to determine the best range for Q, M, and the best T parameter. Computation is polynomially proportional to T and Q, O(T*Q^2), so at the time it was important to keep these numbers low in order to be time-efficient. Fortunately, I was able to find a good range of Q from 6-16, but T was capped at 300 and only one gaussian was needed to represent the data. Results can be seen in tables 6.1.2 – 6.1.4. T = 300 is roughly 4.5 seconds of song, which is good since most humans can distinguish a song in about 5 seconds. It was then also established that for all of the experiments Q = 4-16 states, M = 1 gaussian, and T = 100-300 frames.

Table 6.1.2

Model	Recognition @ T = 100	Recognition @ T = 200	Recognition T = 300
H(6, T, 3)	No Converge	No Converge	No Converge
H(8, T, 3)	25%	No Converge	No Converge
H(10, T, 3)	No Converge	No Converge	No Converge
H(12, T, 3)	56.67%	No Converge	No Converge
H(14, T, 3)	No Converge	No Converge	No Converge

Table 6.1.3

Model	Recognition @ T = 100	Recognition @ T = 200	Recognition T = 300

H(6, T, 2)	46.67%	60%	No Converge
H(8, T, 2)	56.67%	No Converge	No Converge
H(10, T, 2)	68.33%	No Converge	No Converge
H(12, T, 2)	No Converge	No Converge	No Converge
H(14, T, 2)	No Converge	No Converge	No Converge

Table 6.1.4

Model	Recognition @ T = 100	Recognition @ T = 200	Recognition T = 300
H(4, T, 1)	81.67%	76.67%	75%
H(6, T, 1)	66.67%	85%	83.33%
H(8, T, 1)	78.33%	68.33%	75%
H(10, T, 1)	80%	78.33%	85%
H(12, T, 1)	75%	73.337%	86.67%
H(14, T, 1)	80%	73.337%	86.67%
H(16, T, 1)	78.33%	78.33%	88.33%

6.2 Results

For reference, the nearest neighbor classification for the PCA of the raw audio data results are in table 6.2.1.

Table 6.2.1

Dance	Rock	Jazz	Classical	Total
40%	51.43%	25.71%	57.14%	43.57%

In Tables 6.2.2 – 6.2.4 are the results for all four features, including the MFSC. All results are based on HMM parameters Q = 4-16, T = 100-300, and with a "Left-Right" topology.

Table 6.2.2

Model	MFSC	MFCC	LPC	LPCC
H(4,100,35,1)	81.67%	56.67%	63.33%	56.67%
H(6,100,35,1)	66.67%	51.67%	68.33%	58.33%
H(8,100,35,1)	78.33%	61.67%	56.67%	66.67%
H(10,100,35,1)	80%	53.33%	58.33%	65%
H(12,100,35, 1)	75%	58.33%	65%	63.33%
H(14,100,35,1)	80%	53.33%	46.67%	65%
H(16,100,35,1)	78.33%	55%	63.33%	73.337%

Table 6.2.3

Model	MFSC	MFCC	LPC	LPCC
H(4,200,35,1)	76.67%	61.67%	61.67%	56.67%
H(6,200,35,1)	85%	50%	53.33%	60%
H(8,200,35,1)	68.33%	55%	66.67%	63.33%
H(10,200,35,1)	78.33%	58.33%	58.33%	71.67%
H(12,200,35, 1)	73.337%	48.33%	63.33%	60%
H(14,200,35,1)	73.337%	51.67%	70%	65%
H(16,200,35,1)	78.33%	50%	66.67%	68.33%

Table 6.2.4

Model	MFSC	MFCC	LPC	LPCC
H(4,300,35,1	75%	53.33%	63.33%	61.67%
H(6,300,35,1)	83.33%	63.33%	61.67%	60%
H(8,300,35,1)	75%	53.33%	63.33%	61.67%
H(10,300,35,1)	85%	55%	65%	58.33%
H(12, 300,35, 1)	86.67%	56.67%	63.33%	66.67%
H(14,300,35,1)	86.67%	53.33%	60%	70%
H(16,300,35,1)	88.33%	56.67%	61.67%	66.67%

Overall, the best recognition rates for each feature can be seen in table 6.2.5.

Table 6.2.5

PCA	MFSC	MFCC	LPC	LPCC
43.57%	88.33%	63.33%	70%	71.67%

7. CONCLUSION

7.1 Discussion

The relative success of MFSC over other features reviewed can maybe be understood if we were to look at the basic instruments that are involved. In appendix 2 there are examples of graphical features for instruments that are found in one, some, or all of the genres to compare to the figures given in section 4. In comparing these individual instruments to the combined feature we might gain insight why particular feature, namely the MFSC, outperformed the others.

The five instruments looked at were a tenor voice, piano, sax, electric guitar, and drums. In the case of MFSC, we are going to compare song feature figures to the individual instrument feature figures. For Dance music we know that there is a lot of drums, or synthesized drums, which in point would look similar to acoustic drums. We can compare 4.4.3 (MFSC for Dance) to 6.2.8 (MFSC for Drums) and see that the shaping is very similar for the two more so than for the tenor voice, guitar, sax, and piano. Likewise we can compare 6.2.4 (MFSC for Classical) and 4.2.2 (MFSC for Piano). When one tried to make these visual comparisons of features, it becomes a lot harder for LPCC, LPC, and MFCC and this could be a possible explanation for the better performance of the MFSC.

7.2 Future Work

The main goal in future work is to improve the recognition rate. For the most part, the properties of MFSC have not been fully explored to music recognition. There are many possible variations of this work that could have significantly maximized the recognition rate for MFSC. Preliminary test showed that increasing T, the time-width parameter, recognition could be significantly increase, but with the cost of computational efficiency. Also, 20 coefficients might not have been enough to fully represent the classes of music. More tests showed that increasing the number of coefficients to 30 also increased the recognition rate, but it is not known whether the lower coefficients hold more information that the higher ones. Again, the increase in size of the data sets significantly increases the computation time.

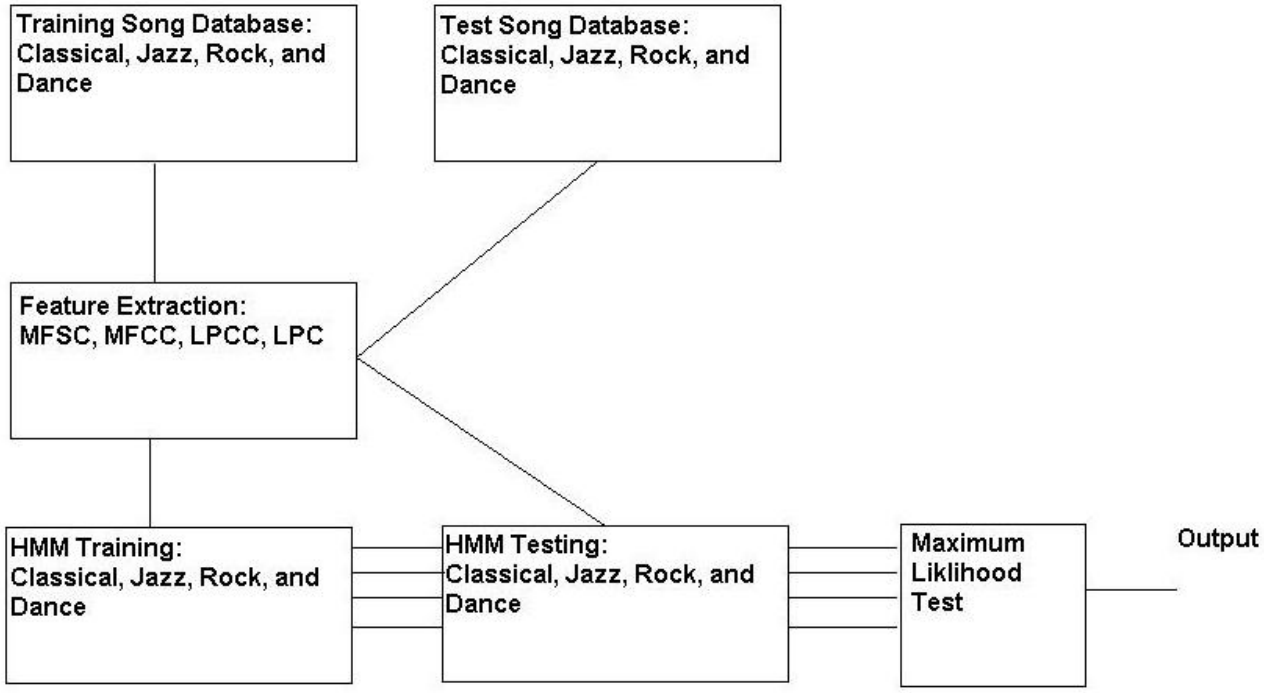
A particular solution to this problem would be to perform a PCA on the entire MFSC to automatically. This method would automatically find the most important dimension in the MFSC space rather than just picking the first 20 or 30 coefficients. Also, this would greatly decrease the size of the data set and solve computational issues. The only set-back to this method would be that a very large amount of storage would be needed for the initial PCA, but after training, only the transformation matrix and mean weights would be needed after that point.

Another possible method of music type recognition would be to model each instrument individually and use a threshold classifier to determine if that instrument is active in song. The output of all of these models could be fed into an HMM that would classify the absence and existence of instruments over time to classify the genre of music. The main problem with this argument is whether the instruments in a song can be linearly separated, but for the most part there is usually one or two primary instrument that dominates a genre. If just these two models are able to classify, the HMM shouldn't have much problem discriminate the absence/existence sequences of instruments.

7.3 Summary

Appendix 1

Flowchart:



Cost Analysis:

	Time Spent	
Researching	**30**	**hours**
Building Database	**30**	**hours**
Writing Software	**150**	**hours**
Analyzing Data	**+30**	**hours**

Total	**(260**	**hours) x (30 dollar/hour) x 2.5 = $19,500**

Appendix 2

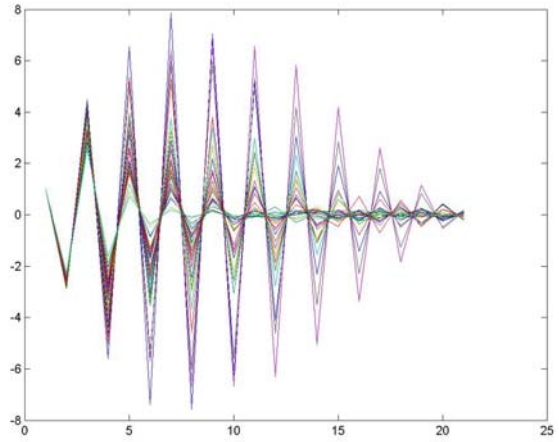


Fig. 6.2.1 LPC of Piano

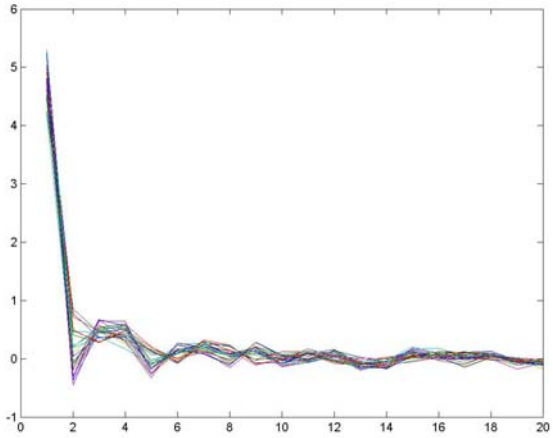


Fig. 6.2.2 LPCC of Piano

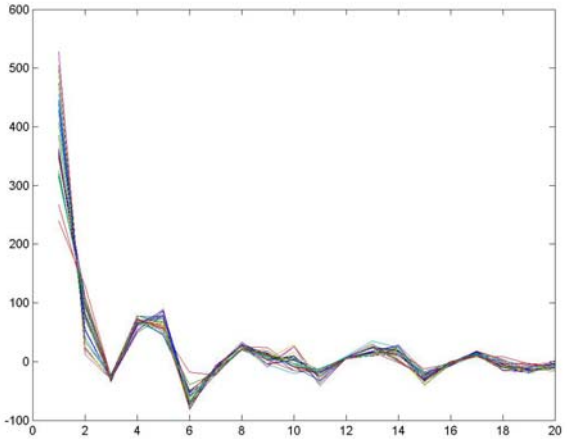


Fig. 6.2.3 MFCC of Piano

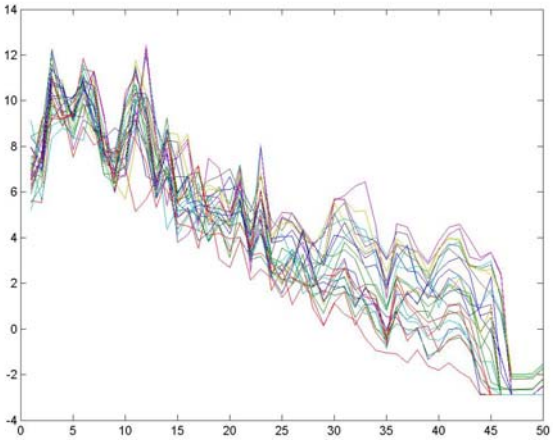


Fig. 6.2.4 MFSC of Piano

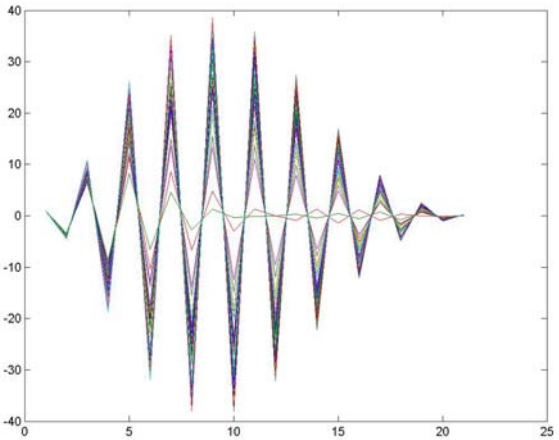


Fig. 6.2.5 LPC of Drums

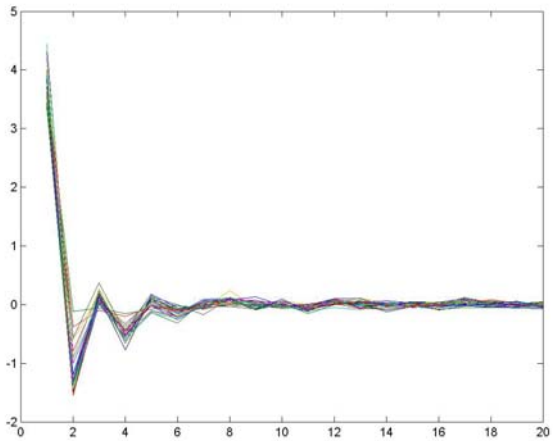


Fig. 6.2.6 LPCC of Drums

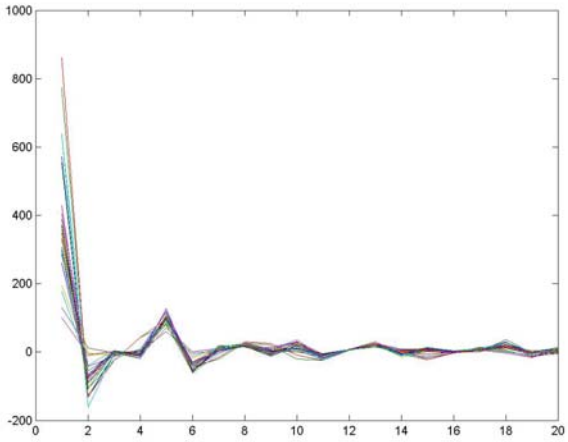


Fig. 6.2.7 MFCC of Drums

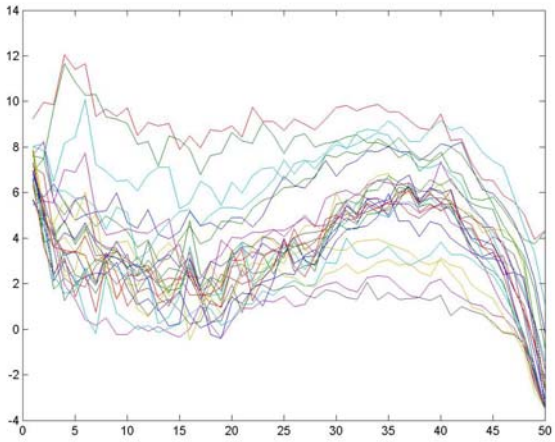


Fig. 6.2.8 MFSC of Drums

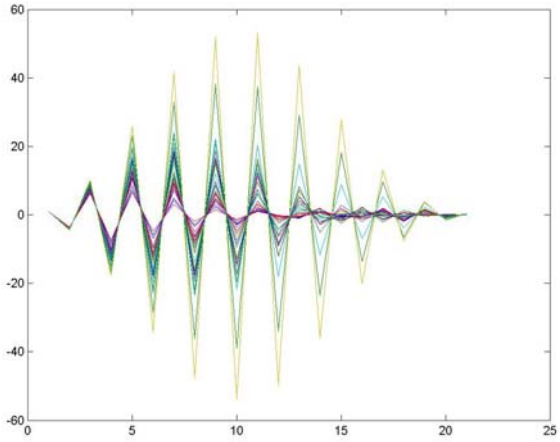


Fig. 6.2.9 LPC of Tenor Voice

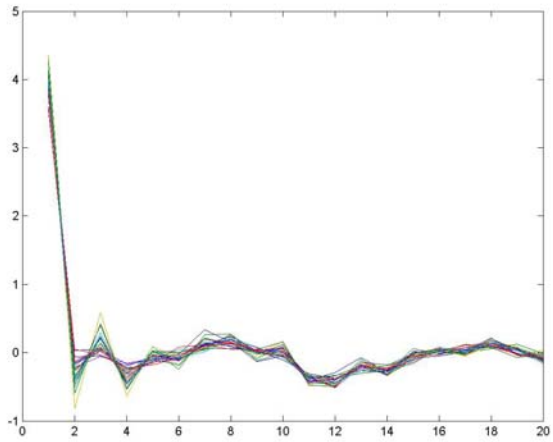


Fig. 6.2.10 LPCC of Tenor Voice

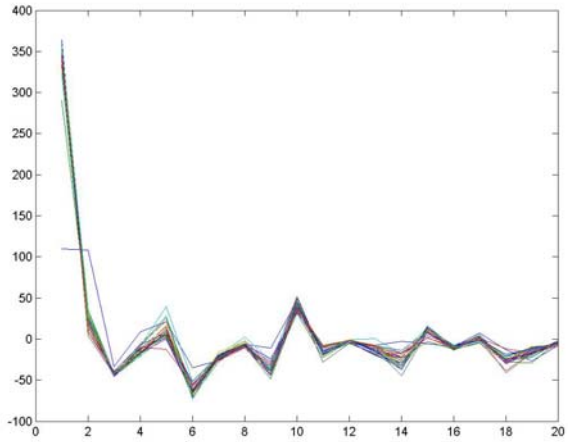


Fig. 6.2.11 MFCC of Tenor Voice

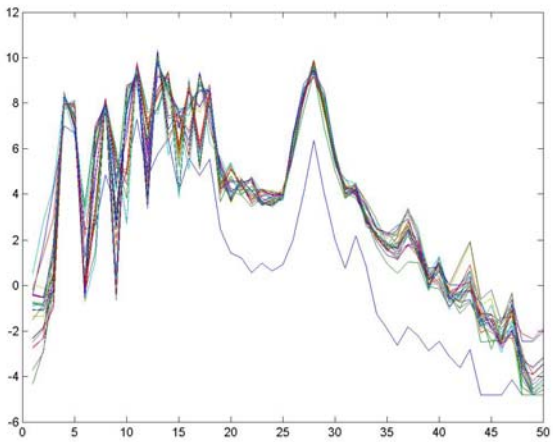


Fig. 6.2.12 MFSC of Tenor Voice

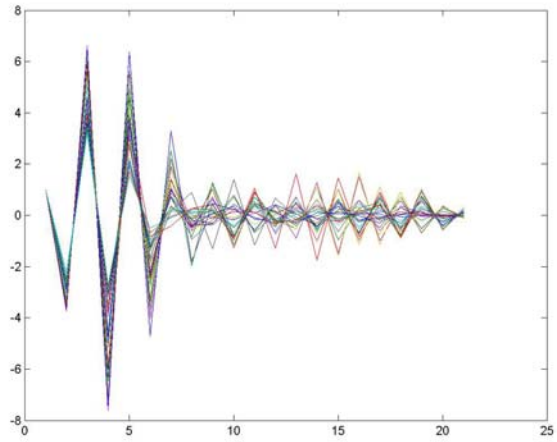


Fig. 6.2.13 LPC of Sax

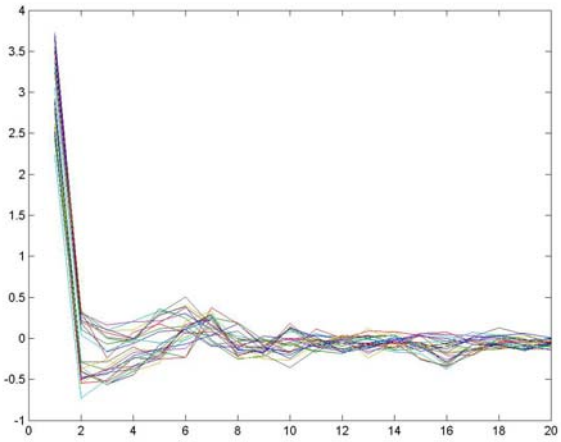


Fig. 6.2.14 LPCC of Sax

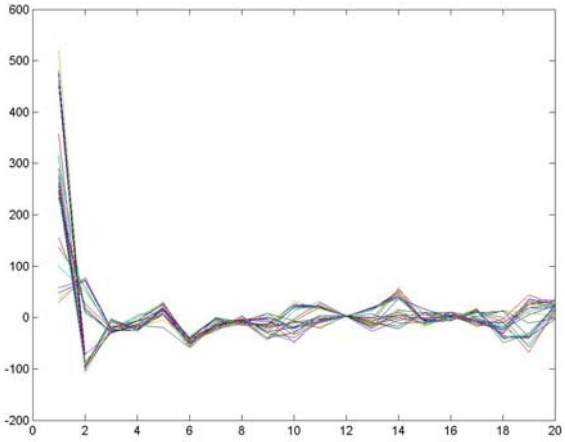


Fig. 6.2.15 MFCC of Sax

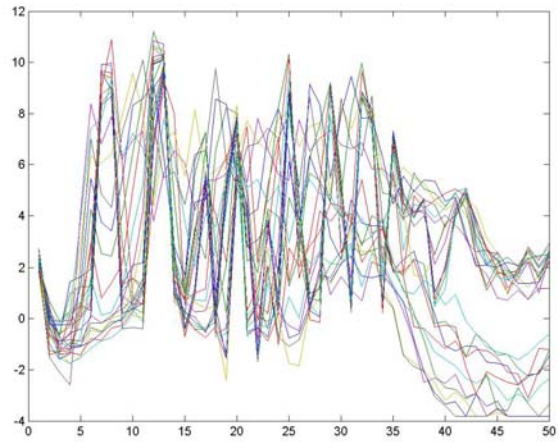


Fig. 6.2.16 MFSC of Sax

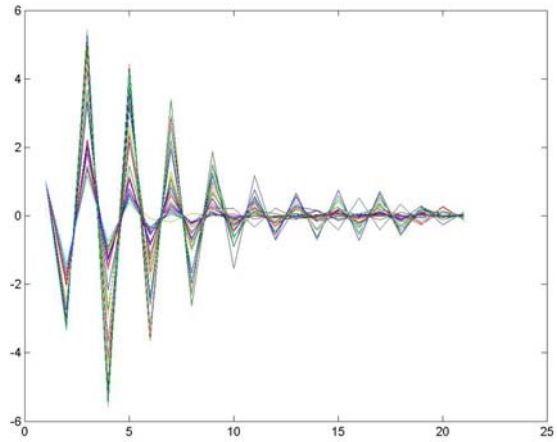


Fig. 6.2.17 LPC of Electric Guitar

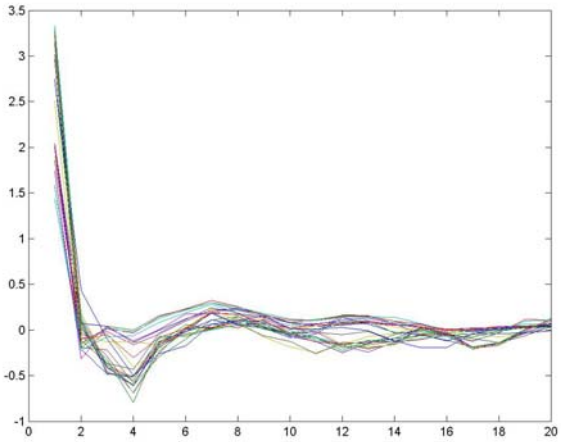


Fig. 6.2.18 LPCC of Electric Guitar

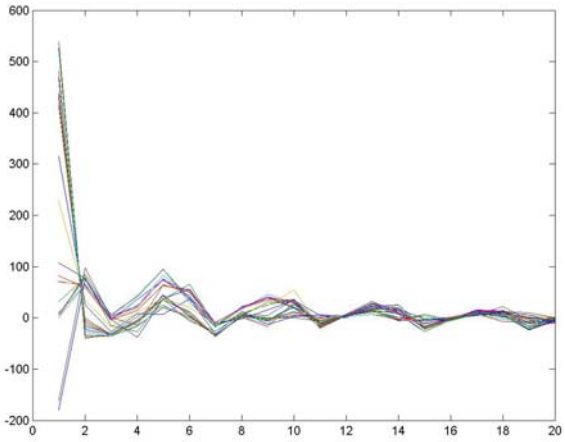


Fig. 6.2.19 MFC of Electric Guitar

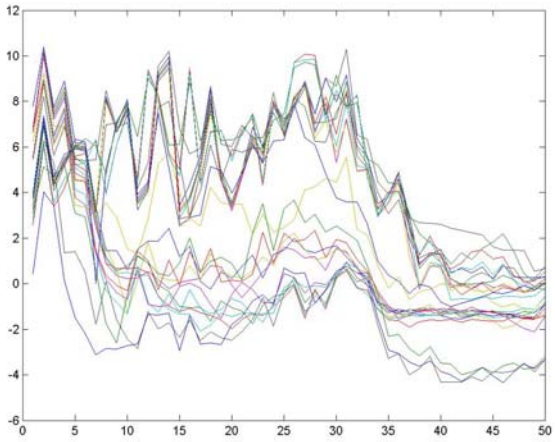


Fig. 6.2.20 MFSC of Electric Guitar

REFERENCES

[1] E. Behrends: An Introduction to Markov Chains, Friedr. Vieweg and Sohn Verlagsgesellschaft mbH, Braunschweig/Wiesbaden, 2000

[2] C. Becchetti, L. P. Ricotti: Speech Recognition, John Wiley and Sons, New York, 1999

[3] A. Bernard, A. Alwan: Source and Channel Coding for Remote Specch Recognition Over Error-Prone Channels.

[4] S. B. Davis, P. Mermelstein: Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences.

[5] D. Ellis: Using Knowledge to Organize Sound: The Prediction-Driven Approach to Computational Auditory Scene Analysis, and Its Application to Speech/Nonspeech mixtures.

[6] B. Gold, N .Morgan: Speech and Audio Signal Processing, John Wiley and Sons, Inc, New York, 2000

[7] M. Hasegawa-Johnson: Lecture Notes in Speech Production, Speech Coding, and Speech Recogntion.

[8] A. Hyvärinen: Survey on Independent Component Analysis

[9] T. Kinnunen, I. Kärkkäinen, P. Fränti: Is Speech Data Clustered? - Statisitical Analysis of Cepstral Features

[10] T. Lambrou, P. Kudumakis, R. Speller, M. Sandler, A. Linney: Classification of Audio Signals Using Statistical Features on Time and Wavelet Transform Domains

[11] S.E. Levinson, L. R. Rabiner, M.M. Sondhi: An Introduction to the Applications of the Theory of Probabilistic Functions of a Markov Process to Automation Speech Recognition

[12] L.R. Rabiner, B.H. Juang: An Introduction to Hidden Markov Models

[13] C. Rowden: Speech Processing, McGraw-Hill Book Co. New York, 1992

[14] H. Soltau, T. Schultz, M. Westphal, A. Waible: Recognition of Music Types

[15] S. Young, J. Jansen, J. Odell, D. Ollason, P. Woodland: The HTK Book.