Genre Based Classification

Mayur Nawal *08d07003*
Pravjyot Singh *08d07006*
Amogh Garg *08d07007*

Introduction

Describing genre of a song is a description of song which can be used for arranging playlists in a media player or can be used effectively for song retrieval from internet database. Other uses can be to find similar songs in the music database.

In this project, we have tried to identify the genre of the song automatically. The basic concept is to find/calculate certain features of music, based on spectrum parameters and time domain
The features we can use to classify music range from shape of the DFT of the windowed signals, the rate of change of the spectrum, changes in the spectrum of sequence of windows. Apart from the DFT patterns, time domain parameters of the signal can also be used to further enhance our classification algorithm. It is basically giving numerical values to a musical piece so that it can be compared with other musical pieces. At the end of feature extraction, we get a vector corresponding to each musical piece.

To classify the music into genres, we then use the derived feature vectors. We are trying two different approaches to classify the feature vectors. First approach is to use SVM classification and the other is Gaussian Markov Model. Though both the approaches have shown acceptable results till, we are still in the process to choose one of the methods over the other.

Feature Extraction

We have used to sets of feature extraction. First set, called the “Surface features” is mainly the spectrum shape. The second set, “Rhythm features” is finding the rhythm of the music piece.

I) *Surface Features:*

The surface features are calculated over short period of the music file. 40 non overlapping consecutive windows of 20ms each are selected from the input music file. For each of these windows DTFT is calculated and the following parameters are calculated:

(In the following, M[f] is the magnitude of the FFT at the frequency bin f and N are the total number of frequency bins.)

i. Centroid: This is calculated as follow:

$$C = \frac{\sum_1^N fM[f]}{\sum_1^N M[f]}$$

So centroid is weighted average of the frequencies with the magnitude of the FFT at the frequency being the weight. This parameter will tell us the frequency around which the FFT is centred.

ii. Rolloff: is the value R such that

$$\sum_{1}^{R} M[f] = 0.85 \sum_{1}^{N} M[f]$$

This R thus shows the shape of the FFT or its rolloff. 85% energy of the signal lies below this frequency. It thus shows the energy distribution of the signal across the signal.

iii. Flux:

$$F = \left\| M[f] - M_p[f] \right\|$$
$$where\ p\ is\ the\ previous\ frame\ in\ time.$$

This parameter tells us the rate of change of the spectrum across the music file. Slower sounding music will have lower flux than a faster piece of music.

iv. Zero Crossing: These are the number of zero crossings in the window. It is a measure of noise in the music signal.

For the 4 parameters above, calculated for the 40 windows we find the means and the standard deviations. We also find the number of windows whose energy is less than the average energy of the 40 windows. This parameter is a measure of energy distribution over time.

Thus we get 9 features for a music peace related to the spectrum shape and energy distributions. These form the surface features of the music file.

II) *Rhythm features*

Rhythm feature set is based on a perfectly structured beat pattern which is carried on through the major portion of a song. The rhythm feature greatly influences the genre classification of a song. For each song a 30 second window is taken which does not contain long silence portion. This is further divided into subsequent windows of approximate duration 3 second (65536 samples at rate 22050 Hz) each with a hop size of 4096 samples.

The rhythm feature set can be calculated using the concept of wavelet transformation. The advantage of using wavelet transformation above short time fourier transformation is that we can achieve variable resolutions in frequency/time domain according to our need whereas the latter provides uniform time/frequency resolution according to the window size. Variable resolution is required when calculating various periodicities in the waveform as in most cases a percussion beat can be represented as an impulse and hence high time resolution is needed to extract the beat pattern whereas to examine the low frequency parts of signal we need high frequency resolution.

The discreet wavelet transform can be computed using multirate filter banks. This can be implemented by decomposing the frequencies of the signal into frequency bands with different bandwidths with a constant Q (centre frequency/bandwidth) with octave spacing between the centre frequencies and hence the frequency band with highest centre frequency has maximum bandwidth and hence has best time resolution.

To calculate the rhythm pattern we first divide the signal into its different frequency bands and getting the time domain waveform for each band. This can be achieved by sequentially passing the signal though high pass and low pass signal where the high pass signal is the component obtained to be used later and low pass component is passed through next stage of high pass and low pass filtering. The high pass filter used in each stage is a Q filter with centre frequency decreasing by an octave after every stage and the low pass filter used is simply the compliment of the high pass filter.

When calculating the feature vector we have used centre frequency of 3*pi/4 for first stage of high pass filter with bandwidth of pi/2 and we have used 4 bands to calculate the rhythm feature vector to compromise between extracting the major periodicities in the signal and to ignore the non-periodic low frequency components of the signal.

After getting the different time domain signal components we pass each component through following series of operations:

1) Full wave rectification and low pass filtering to get the envelope of the component
   To remove the periodicities in the signal due to the high frequency harmonics of the signal we need to find the envelope of the signal which can be obtained by first rectifying the signal and then passing the rectified signal through a low pass filter(single pole with root 0.99)

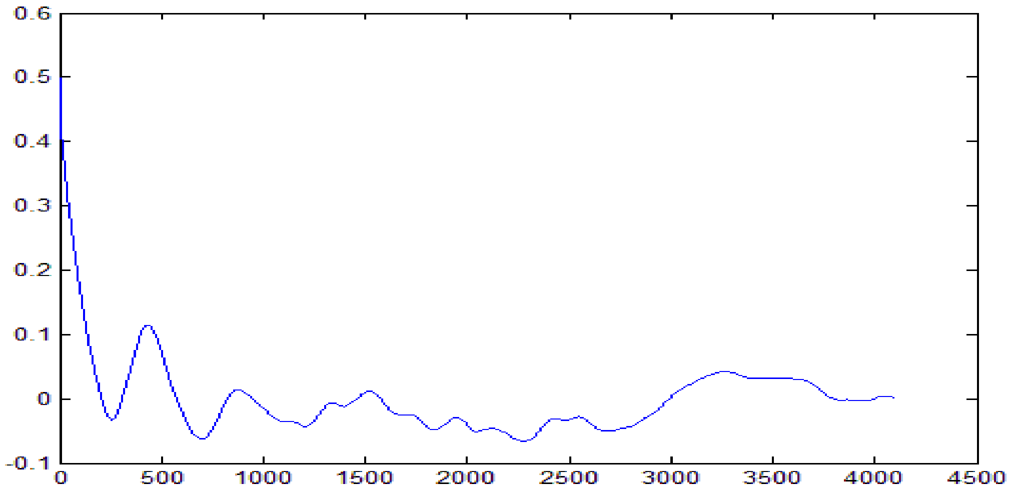
2) Down-sampling
   To reduce the amount of computation time but not to remove unwanted information from the signal we downsample the obtained signal by a factor of 16

3) Normalisation
   The mean of the signal is subtracted from the signal obtained after down-sampling.

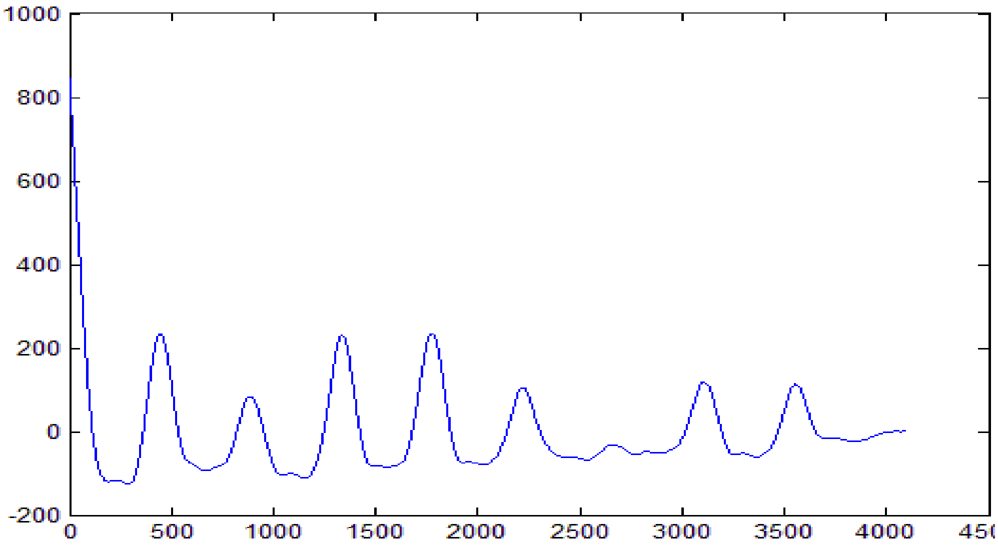
After conducting these operations for each component of the signal we combine the respective components and calculate the autocorrelation of the resultant signal obtained which gives us insight of the salient periodicities present in the signal.

After calculating the autocorrelation sequence we find the ten most prominent peaks of the autocorrelation sequence (excluding the highest peak at 0), for each peak obtained we calculate the difference between the amplitude of the peak and amplitude of the nearest minima on both sides and average them (let us call it average difference) and we calculate the index of the peak in form of beats per minute. (The peaks are calculated from a Matlab program ‘extrema.m’. This program was downloaded from the internet.)

Following is the autocorrelation plot obtained from a classical song:



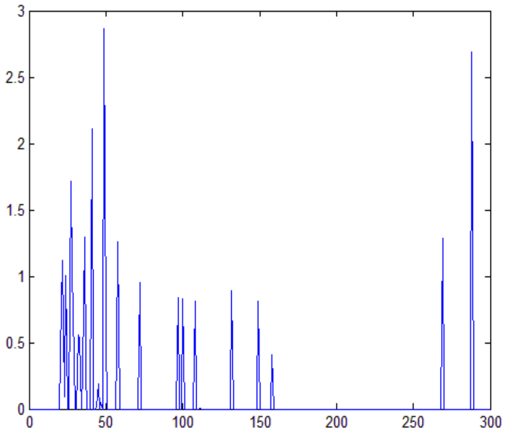
Following is the autocorrelation plot obtained from a hiphop song:



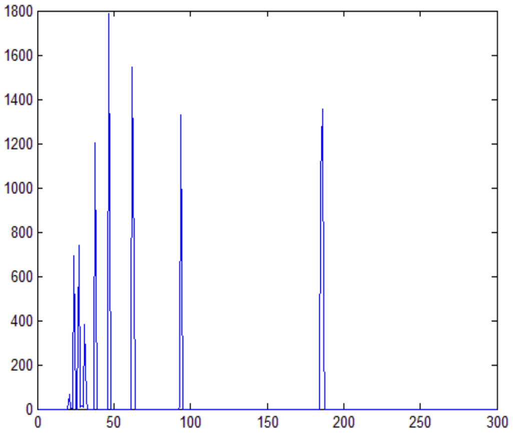
The ten peaks obtained are added in a histogram by increasing the amplitude of a index (beats per minute) of histogram by the average difference of the peak. After addition of all autocorrelation peaks for all windows in the histogram we select the three most prominent peaks of the histogram and extract following parameters:

1) Period 0: periodicity in beats per minute of the first peak
2) Amplitude 0: relative amplitude (w.r.t sum of histogram entries) of the first peak
3) Ratioperiod1: ratio of periodicity in beats per minute of the second peak with period0
4) Amplitude1: relative amplitude (w.r.t sum of histogram entries) of the second peak
5) Ratioperiod2, amplitude2, ratioperiod3, amplitude3

The following set of 8 parameters constitutes the rhythm feature vector.
Following are plots obtained of histogram for classical and hiphop music:



Classical



Hipop

The histogram above shows that hiphop histogram has less number of peaks but are very prominent whereas classical has peaks distributed all over the histogram.

Classification

Gaussian Markov Model is used for sorting vectors into different labels.

Steps:

- Calculating full covariance matrix for all the Genres in which we want to classify.
- The size of the covariance matrix depends on the no. of feature set (no. of columns) and no. of trail music pieces considered for training (no. of rows).
- Trial music pieces should be more then no. of features.
- Every row of this covariance matrix comprises of feature set for a training music piece.
- Taking Mean and covariance of these observations (training variables).
- Mean is subtracted from the initial Covariance matrix.
- The qr function performs the orthogonal-triangular decomposition of this matrix. It expresses the matrix as the product of a real orthonormal and an upper triangular matrix.
- [Q,R] = qr(X) produces an upper triangular matrix R of the same dimension as X and a unitary matrix Q so that X = Q*R.
- For each of the test observations, calculate its distance from these model using Mahalanobis distance
- Input vector is classified in the genre with which it gets a minimum Mahalanobis distance.

Mahalanobis Distance:

Mahalanobis distance of a multivariate vector $x = (x_1, x_2, x_3, \dots, x_N)^T$ from a group of values with mean $\mu = (\mu_1, \mu_2, \mu_3, \dots, \mu_N)^T$ and full covariance matrix S is defined as:

$$D_M(x) = \sqrt{(x-\mu)^T S^{-1} (x-\mu)}.$$

Advantages of Mahalanobis distance over Euclidean distance is that it takes into account the correlations of the data set and is scale-invariant.