

University of Burgundy

Masters in Computer Vision and Robotics

SSI Pattern Recognition

MID TERM REPORT

AUDIO GENRE CLASSIFICATION

By

Vamshi Kodipaka

Bhargav Shah

Parmar Hardhik sinh

Supervisor: Dr. Desire Sidibe



CONTENTS

1. Recall from Abstract
2. Understanding the Audio : Pipeline
3. Mid-Term Progress
4. Basic of Audio Operations
5. MFCC Feature Extraction
 - a. What is Mel-Frequency Cepstrum and MFC Coefficients
 - b. MFCC Features- Extraction:: Matlab inbuilt
 - c. Math behind MFCC :: Three types of Spectral Analysis
 - i. Convential Cepstral Method
 - ii. Linear Predictive Coding based
 - iii. Mel-Frequency Cepstral based
6. Attachments
7. Outputs
8. Conclusions for Feature Extraction
9. Future Work(To-do-list)
10. References

Recall from Last Abstract Submission:

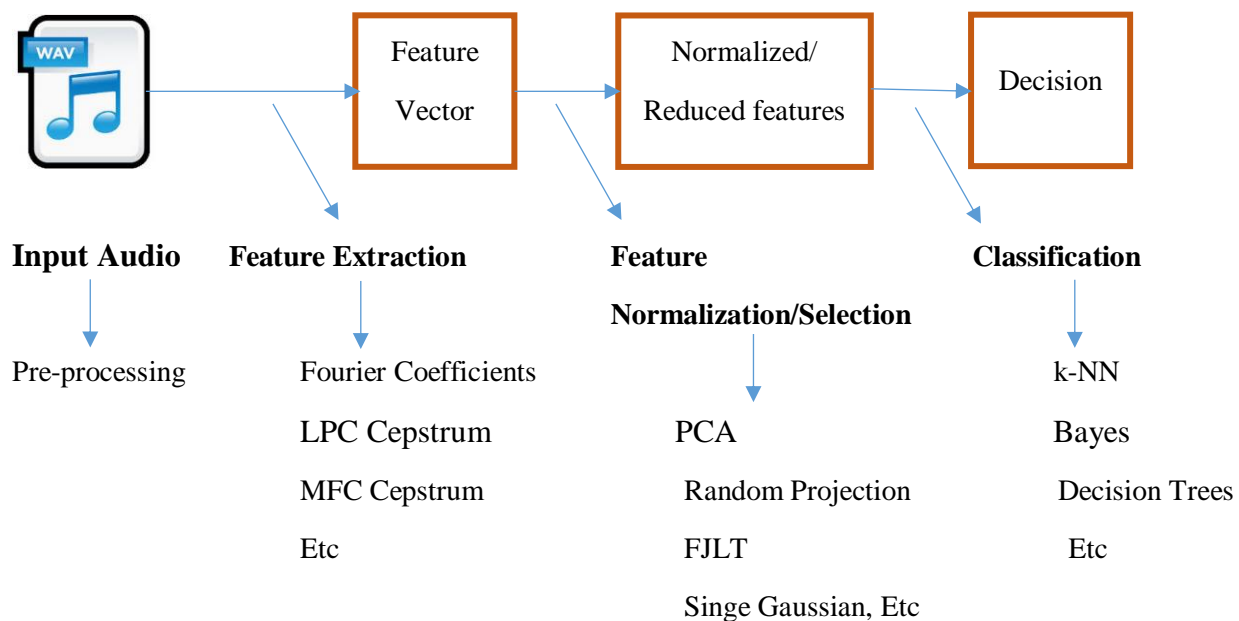
Dataset : ISMIR-2004 music files (729 tacks)

The dataset that has been provided to us had 729 tracks of the genres Classical (320), Electronic(114), Jazz/Blues(26), Metal/Punk(45), Rock/pop(102), World(122). Where each of the above songs is sampled at 11025 Hz mono.

We have to implement our project in 3 steps:

1. Feature Extraction
2. Feature Selection/Normalization
3. Feature Classification

Audio Understanding: Pipeline



Mid-Term Report Progress:

My Mid Term Report Submission, we decided to work on Feature Extraction (Mel Frequency Cepstral Coefficients - MFCC) from audio files and we achieved it.

Idea for .mp3 to .wav conversions:

We have .mp3 files and we can convert to .wav files. Converting to wave files can be done reducing the sampling rate and changing bit rate.

We either use .mp3 are .wav file as per MATLAB compatibility.

(The commands related to these conversions are explained in a 'a1.txt' file)

Since we have 729 audio mp3 files, we first take one .mp3 file and try to extract features for it. Then we will try to display the extracted features as interest points. Then we will try to take for whole music dataset.

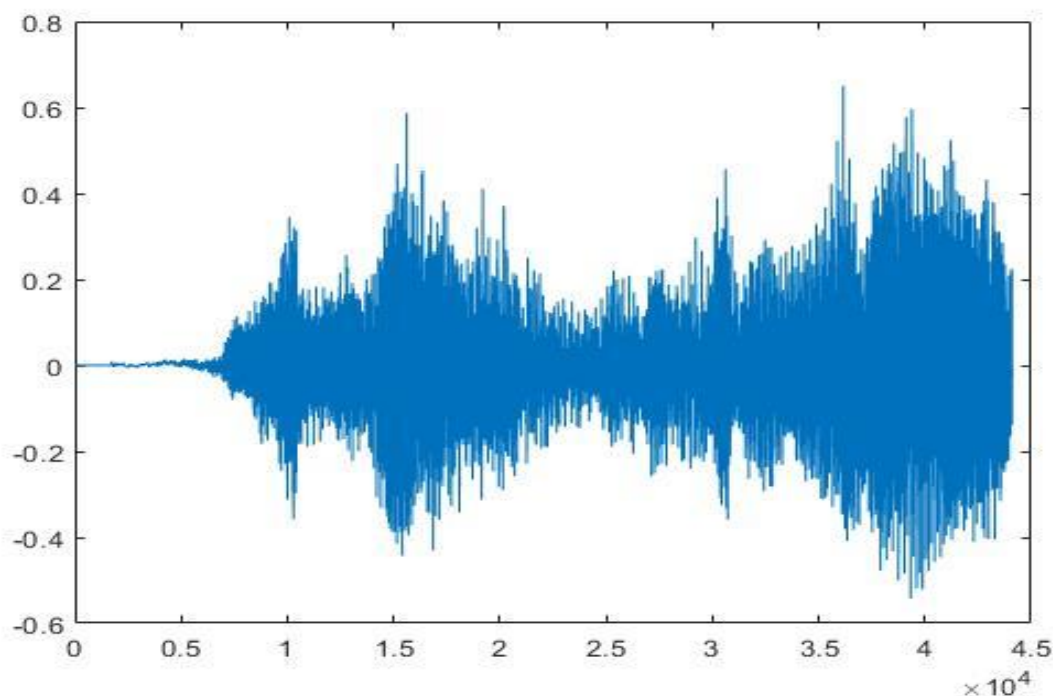
BASICS OF AUDIO PROCESSING:

1. To load .mp3 file: `[y,Fs]=audioread('artist_1_album_1_track_1.mp3')`

Here, y gives amplitude and Fs gives frequency.

2. Plot .mp3 with amplitude (vs) frequency:

`plot(y(1:44100,1))`



3. Viewing the information of the .mp3 file:

`info=audioinfo('artist_1_album_1_track_1.mp3')`

4. To play .mp3 using sound command:

`sound(y(1:441000),1,Fs)`

5. To play .mp3 using audioplayer object in MATLAB:

`p=audioplayer(y,Fs)`

`play(p) :: plays loaded music fil`

`pause(p) :: pauses music file`

`stop(p) :: stops music file`

`start(p) :: starts from point of stop`

`clear(p) :: clears 'p' object's music file.`

```
Command Window

>> p=audioplayer(y,Fs)

p =

    audioplayer with properties:

        SampleRate: 44100
        BitsPerSample: 16
        NumberOfChannels: 2
        DeviceID: -1
        CurrentSample: 1
        TotalSamples: 1686000
        Running: 'off'
        StartFcn: []
        StopFcn: []
        TimerFcn: []
        TimerPeriod: 0.0500
        Tag: ''
        UserData: []
        Type: 'audioplayer'

>> play(p)
>> pause(p)
>> stop(p)
fx >> |
```

```
Workspace

Name  Value
Fs    44100
y     1686000x2 double

>> info=audiointro('artist_1_album_1_track_1.mp3')

info =

    struct with fields:

        Filename: 'D:\AudioGenreClassifier-master\AudioGenreClassifier-master\artist_1_album_1_track_1.mp3'
        CompressionMethod: 'MP3'
        NumChannels: 2
        SampleRate: 44100
        TotalSamples: 1687360
        Duration: 38.2621
        Title: []
        Comment: []
        Artist: []
        BitRate: 128
```

Note:

We can plot Amplitude vs frequency graphs for various type of music files (like rock, pop, instrumental etc.) among the data set. But we don't know which music file is of which type among them. And our task is to classify. But before that, we have to extract features.

Note: All the basic commands related to audio operations are attached in a a1.txt file along with this pdf.

FEATURE EXTRACTION: MFCC

a. What is Mel-Frequency Cepstrum and MFC Coefficients?

Mel refers to 'Melody'. Cepstrum means the Inverse Fourier transform (IFT) of the logarithm of the estimated spectrum of a signal.

Definition: In sound processing, the **Mel-Frequency Cepstrum (MFC)** is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency.

History of MFCC:

Paul Mermelstein is typically credited with the development of the MFC. Mermelstein credits Bridle and Brown for the idea:

Bridle and Brown used a set of 19 weighted spectrum-shape coefficients given by the cosine transform of the outputs of a set of non-uniformly spaced bandpass filters. The filter spacing is chosen to be logarithmic above 1 kHz and the filter bandwidths are increased there as well. We will, therefore, call these the mel-based cepstral parameters.

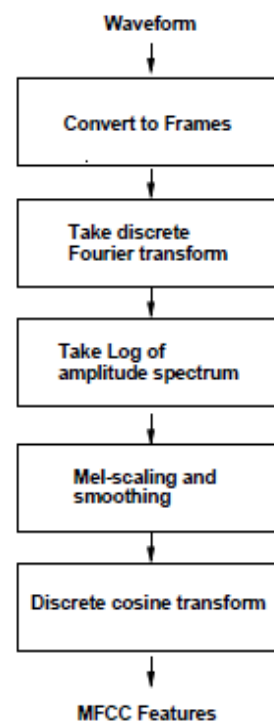
Davis and Mermelstein have commented that the spectral basis functions of the cosine transform in the MFC are very similar to the principal components of the log spectra, which were applied to speech representation and recognition much earlier by Pols and his colleagues. (Source: Wikipedia)

The difference between the Cepstrum and the mel-frequency cepstrum:

In the MFC, the frequency bands are equally spaced on the mel scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal cepstrum. This frequency warping can allow for better representation of sound, for example, in audio compression.

MFCCs are commonly derived as follows:

1. Take the Fourier transform of (a windowed excerpt of) a signal.
2. Map the powers of the spectrum obtained above onto the mel scale, using triangular overlapping windows.



Process to create MFCC features

3. Take the logs of the powers at each of the mel frequencies.
4. Take the discrete cosine transform of the list of mel log powers, as if it were a signal.
5. The MFCCs are the amplitudes of the resulting spectrum.

There can be variations on this process, for example: differences in the shape or spacing of the windows used to map the scale, or addition of dynamics features such as "delta" and "delta-delta" (first- and second-order frame-to-frame difference) coefficients.

Also, the European Telecommunications Standards Institute in the early 2000s defined a standardized MFCC algorithm to be used in mobile phones.

Noise Sensitivity:

MFCC values are not very robust in the presence of additive noise, and so it is common to normalize their values in speech recognition systems to lessen the influence of noise. Some researchers propose modifications to the basic MFCC algorithm to improve robustness, such as by raising the log-mel-amplitudes to a suitable power (around 2 or 3) before taking the DCT (Discrete Cosine Transform), which reduces the influence of low-energy components

Use: Mel-Frequency Cepstral Coefficients are the standard preprocessing technique in Speech Processing and developed for automatic speech recognition and proven to be most useful for music information retrieval tool.

Preprocessing:

Sonograms are similar to MFCC. But they are based on psychoacoustic models which are slightly more complex than those used in MFCC. So we use MFCC as feature extraction. It is a non-linear frequency scale. The important aspects of MFCC model are:

1. Non-linear perception of loudness in decibels
2. To some extent special masking effects using Discrete Cosine Transform
(Here, DCT is substitute of IFT)

b. MFCC Feature Extraction:

This method captures overall spectral shape, which carries all the important about the the instrumentation of its timbres, the quality of the singers voice and production effects. It drastically reduces amount of data for the computational model of music similarities

Working with MFCC:

Firstly the given audio waveform is converted into small frame of size decided by the user. We decided the frame size to be 1024 samples in order to achieve a power spectrum of the given audio waveform.

Firstly we take discrete Fourier transform and then apply take log of amplitude spectrum of each frame now in order to model the spectrum based on human auditory system we performed mel scale on it, the human auditory system perceives audio in linear scale linearly at lower frequencies and in logarithmic scale at higher frequencies.

Finally we apply discrete Fourier transform and hence achieve MFCC features. We have chosen number of cepstral coefficients to be 10 in order to reduce number of datapoints and hence the complexity.

Calculation of Mel-Frequency:

Note: Mel-Scale is approximately linear for low-frequency ($f < 500\text{Hz}$) and logarithmic for high frequencies.

$$M(\text{mel_freq}) = 1127 * \log(1 + f/700)$$

where f is frequency in linear scale
and M is frequency in mel scale.

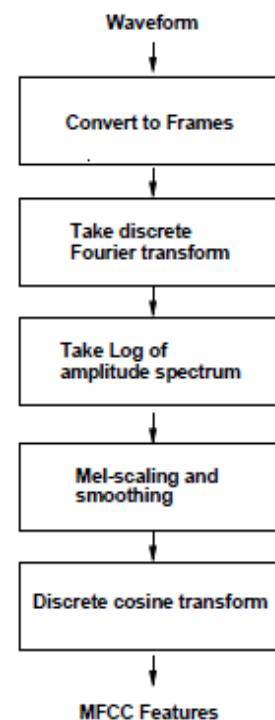
This modulation of log acts as a weight vector like in
($y = w^T X - t$) in a classical regression model

From the above formula and graph we observe that mel frequency is linear at lower frequencies and logarithmic at higher frequencies.

(Hence on application of MFCC we have a $10 \times L$ matrix, where L depends on the length of song. L on average was found to be 3000. So we have reduced the data points from about a million to about 10×3000)

The database, it was found to be about 4000 samples. So for uniformity we reduced the samples of each song to 4000 samples irrespective of their length. The reduction of the samples was done by dividing the samples of each song into 4000 equal groups and then taking one sample from each group.

This method was very logical because we now have equal number of samples for all songs and have also preserved the time varying features of our audio file.



Process to create MFCC features

We used Matlab's inbuilt MFCC functions in MA Toolbox (Malcolm Slaney's Auditory Toolbox) for Feature Extraction.

- a. ma_mfcc.m
- b. mfcc_coeff.m

So, these functions does the required operations shown in this flow-chart.

c. Math Behind MFCC Feature Extraction:

Origin of mfcc has have two primitive analysis: CSA and LPCC and then we see MFCC. Let $x[n]$ be the input signal (complex spectrum),

i. From conventional spectral analysis:

Then we can say $C[n]$ real cepstrum of the signal is:

$$C[n] = \frac{x[n] + x[-n]}{2}$$

As we said earlier, real cepstrum of a signal $x[n]$ and

Calculating Cepstral Coefficients of $C[n]$ cepstrum:

=====****=====

Inverse FT : $C(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log(|X(e^{jw})|) e^{jwn} dw$

$$C(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log(|X(e^{jw})|) \cdot [\cos(jwn) + \sin(jwn)] dw$$

Since odd parts: sin terms becomes zero is 0

Therefore,

$$C(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log(|X(e^{jw})|) \cdot \cos(jwn) dw$$

Now:

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log(|X(e^{jw})|) e^{jwn} dw \quad \text{----- (eq. 1)}$$

We know that, if is a complex signal: $x = a + jb$;

$$x = a + jb; = |x|. (e^{j\theta})$$

Then x[n] in eq.1 becomes: $x[n] = \log(|X(e^{jw})|) + \arg(X(e^{jw}))$

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} [\log(|X(e^{jw})|) + j \arg(X(e^{jw}))]. [\cos(jwn) + \sin(jwn)] dw$$

Positive cepstrum:

$$\begin{aligned} x[n] &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \log(|X(e^{jw})|) \cdot [\cos(jwn)] dw \\ &\quad - \frac{1}{2\pi} \int_{-\pi}^{\pi} j \arg(X(e^{jw})) \cdot [\sin(jwn)] dw \end{aligned}$$

Negative cepstrum:

$$\begin{aligned} x[-n] &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \log(|X(e^{jw})|) \cdot [\cos(jwn)] dw \\ &\quad + \frac{1}{2\pi} \int_{-\pi}^{\pi} j \arg(X(e^{jw})) \cdot [\sin(jwn)] dw \end{aligned}$$

Adding x[n] and x[-n] gives C[n] as mentioned initially

=====****=====

ii. Linear Predictive Coding Cepstrum(LPC Cpestrum):

Let $x[n]$ is a signal, $a_0, a_1, a_2, \dots, a_p$ are the LPC coefficients of p th order, then DFT of $x[n]$, we get LPC Spectrum----- (Statement-1)

If instead of DFT of signal, if we use signal $x[n]$ and take the DFT, I will get $x[k]$ which is called a real signal spectrum and Statement-1 gives us the LPC Cepstrum.

Calculating Cepstral Coefficients of LPC Cepstrum:

To calculate c_0, c_1, \dots, c_p (cepstral coefficients), we have:

The LPC vector is defined by $[a_0, a_1, a_2, \dots, a_p]$ and the CC vector is defined by $[c_0, c_1, c_2, \dots, c_{p-1}]$

LPC Cepstrum (c_m)	
$c_0 = \log G^2$	$G = e^{c_0/2}$
$c_m = a_m + \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) c_k a_{m-k}, \quad 1 \leq m \leq p$	$a_m = c_m - \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) c_k a_{m-k}, \quad 1 \leq m \leq p$
$c_m = \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) c_k a_{m-k}, \quad m > p$	

Here, G is LPC Gain =====*****=====

iii. Mel-Frequency-Cesptrum:

1. Compute FFT power spectrum of speech signal
2. Apply mel-space filter bank to the power spectrum to get energies
3. Compute DCT of log filter-bank energies to get uncorrelated MFCCs.

So, instead of taking whole spectrum we take 80 points of the spectrum and find out the cepstral coefficient. We can treat that is a signal pass through the inverse DFT and cepstral coefficient will be generated. So, information is reduced.

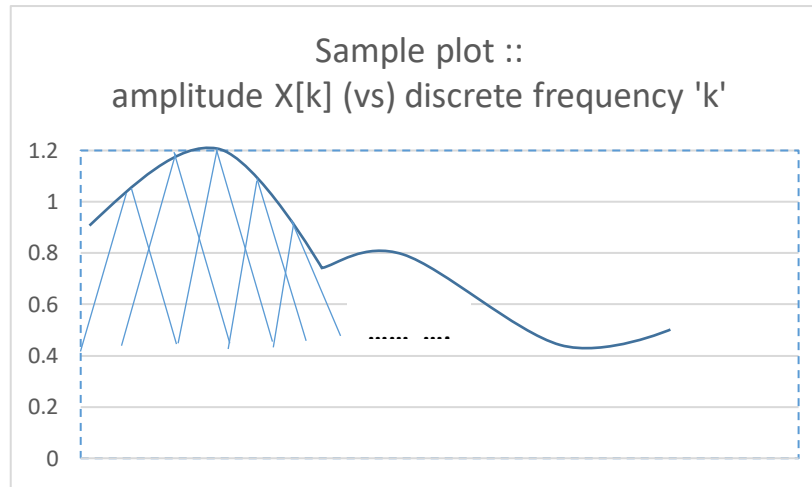
Now the problem is that if it is a linear filter this does not matched with human perception (human frequency perception)

Let $x[n]$ is framed through the entire signal ,

DFT

has a window size fixed

$$x[n] \rightarrow X[k] \quad \text{--} |X[k]|$$



$$k = N/2; \quad f_{\max} > F_s/2; \quad F_s = 8\text{KHz}; \quad f_s \text{ becomes } 4\text{KHz}$$

$$\text{If triangular train filter} = 100\text{Hz} \rightarrow F_s = 4000/100 = 40\text{points}$$

Now suppose, this triangular filter is applied to 8KHz = 80points will be generated.

$$\text{To convert into } k \text{ to } f, \text{ we have: } f = 2\pi k/N;$$

We say, the signal the frequency perceived by human being is not in linear scale this is in Mel scale. So, instead of taking the linear filter now we convert the filter bandwidth as per the Mel scale what is the Mel scale you know this using this equation described previously. So, the bandwidth of the filter is depends on the Mel scale. So, instead of uniform bandwidth filter, we take Mel scale filter.

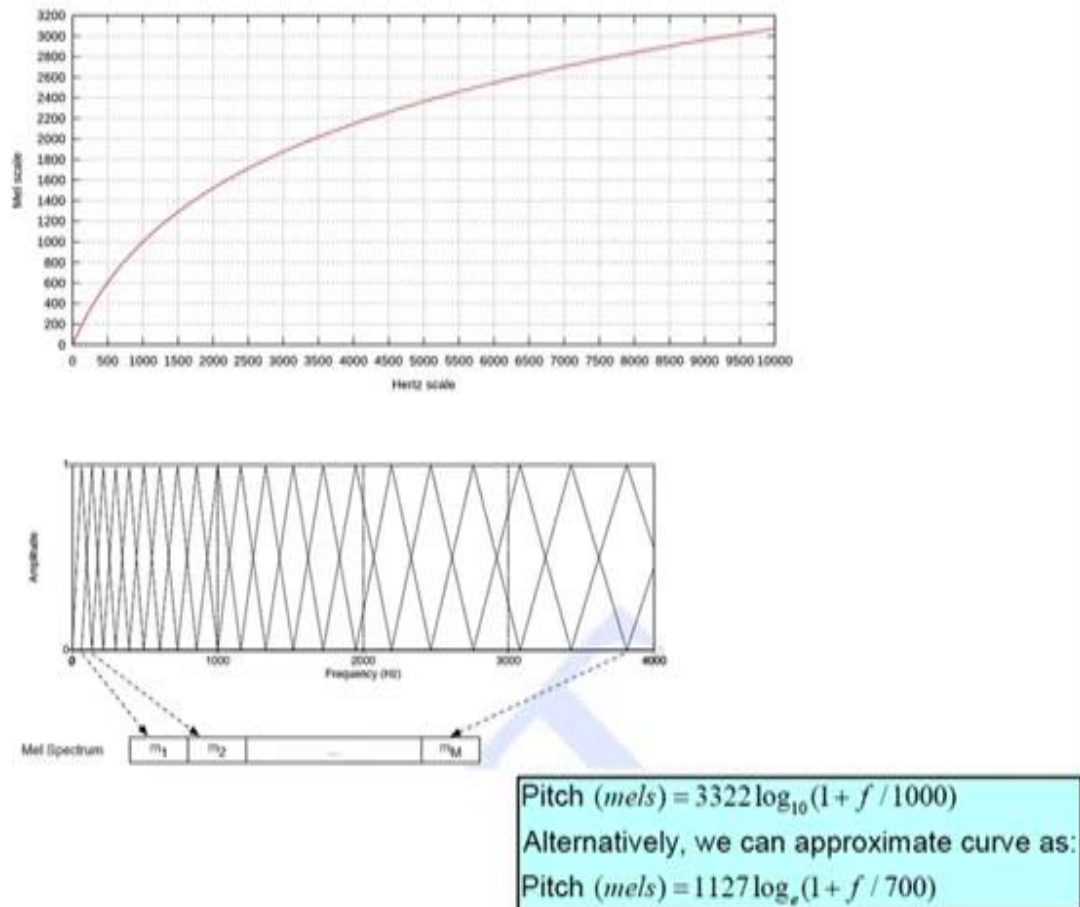
So, what is human perception in frequency lower frequency our resolution is very high; that means, lower frequency we can perceive linearly along the physical frequency, but at the higher frequency we have a some low resolution; that means, the band of frequency the frequency band we perceive as a same frequency is larger. So, we can say at the high frequency region the bandwidth of the filter will be large. So that means, who do not require that much of course, resolution. So, half estimation is sufficient for high frequency.

So, that is why we take the bandwidth of the filter will be larger. So, bandwidth defined by the Mel scale. So, if we take the Mel scale filter then we take the Mel scale filter if you find out the

dividends 4 kilo hertz we take 20 filter which will be sufficient to cover the 4 kilohertz frequency or entire frequency range.

So, I can get $m_1, m_2 \dots m_{20}$ because every filter give me a single point bandwidth single bandwidth. So, every filter has a single bandwidth which is m_1, m_2, m_3, m_4 , we can find out 20 Mel point.

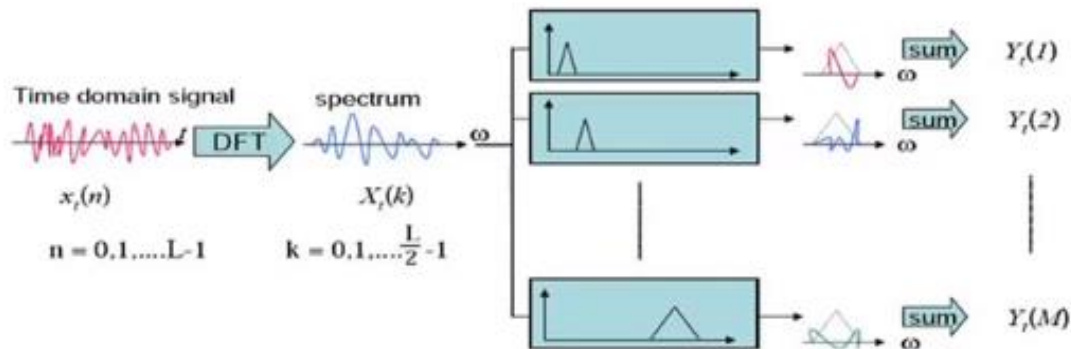
So, since filters are Mel scale filter are designed in Mel scale then we can say we have Locked the LP or the frequency spectra in Mel scale instead of hertz



So, what we get instead of hertz, we get Mel scale in here and here is let $X_{\text{cap}}[k]$ instead of $x[k]$. We take X_{cap} average(average energy). So, I instead of spectrum I get Mel scale spectrum once we get the Mel scale spectrum, if we analyze cepstral coefficient, then this is called Mel scale cepstral coefficient. So, since my spectrum is frequency warped in Mel scale that is why it is called Mel frequency cepstral coefficient.

So, what I am actually doing any mathematics I am designing the filters I design I will explain in the next class first I designed and described the equation.

Mel Filter bank



So, we calculate $X_{\text{cap } L}$; L is the number of filter

$S_{\text{cap } L}$ using Mel filter bank over the Mel filter, we are calculating Mel spectrum and once cepstral coefficient is analyzed based on the Mel shaped spectrum then is called **Mel frequency cepstral coefficient**

Mel Filter bank

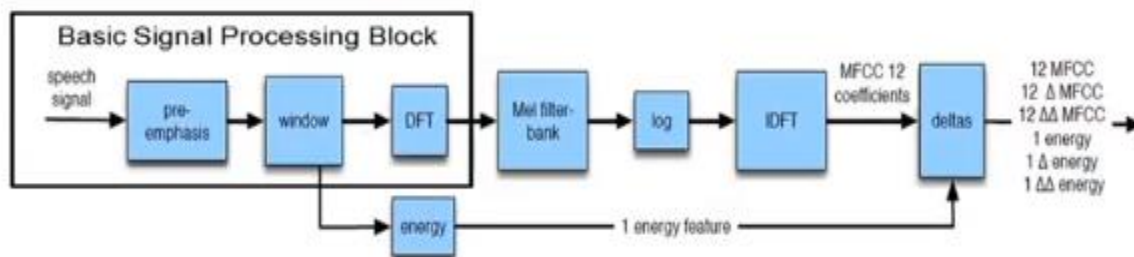
Diagram illustrating the Mel Filter bank equation and components:

$$\tilde{S}(l) = \sum_{k=0}^{N/2} S(k) M_l(k) \quad l = 0, 1, \dots, L-1$$

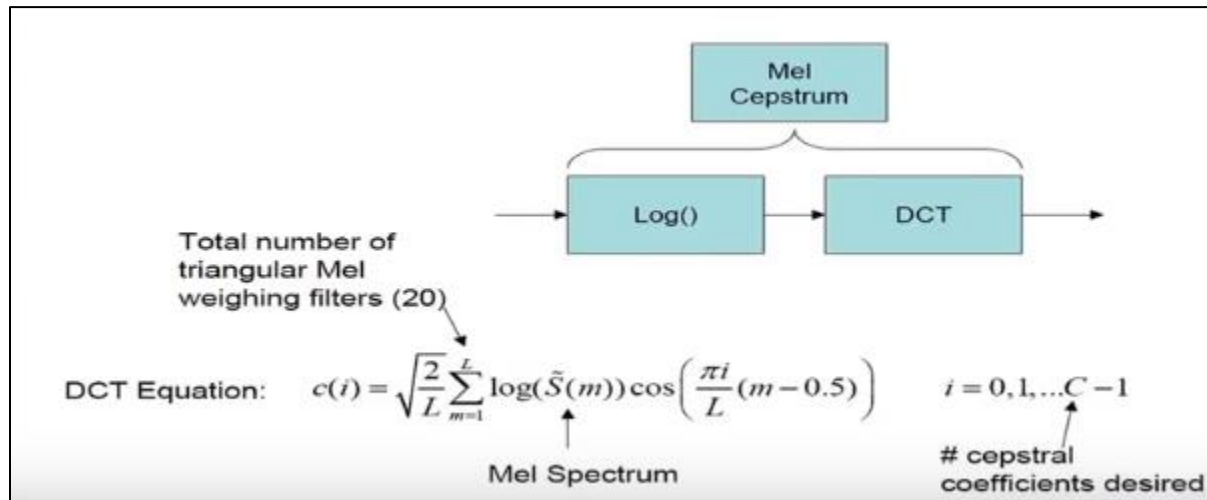
Annotations:

- Mel Spectrum:** $\tilde{S}(l)$
- Half the FFT size:** $N/2$
- Original Spectrum:** $S(k)$
- Total number of triangular Mel weighing filters (20):** L
- l^{th} Filter from filter bank:** $M_l(k)$
- Frequency conversion:** $k \rightarrow \left(\frac{k f_s}{N} \right) \text{ Hz}$
- Will get the whole range of frequencies but only L samples:** $\tilde{S}(l)$

Block diagram of Extracting a sequence of 39-dimensional MFCC feature vectors



Instead of IFT we can take DCT:



ATTACHMENTS ALONG THIS PDF DOCUMENT:

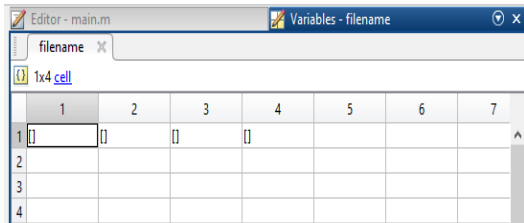
1. **main.m**: Main file (to RUN)
2. **ma_mfcc.m**: to perform the mel-frequency-cepstrum analysis.
3. **mfcc_coefficients.m**: to find the mfcc coefficients for all the given no. of .wav files
4. **a1.txt**: describes the basic commands on audio operations.
5. **Four sample music files(in tracks folder)**:
We have extracted features [mfcc_coefficients,DCT_coefficients] for all 729 files.
We have attached 4 sample music files out of 729 files, to prove that we have extracted the required features (mfcc,DCT)
6. **mfccResults.mat** and **mfccResults.xls** in **(mfccResult)** : Contains results of mfcc_coefficients and DCT matrix
7. Output figures

OUTPUTS:

(each cell has its own purpose:: See the workspace)

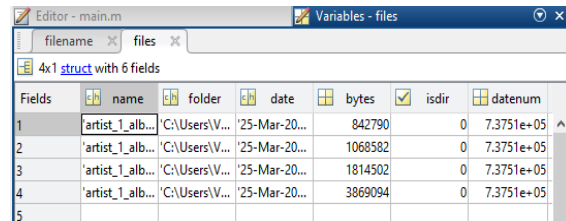
Workspace	
Name ▲	Value
filename	1x4 cell
files	4x1 struct
k	4
mfcc_artist_1_album_1_track_1	79x3290 double
mfcc_artist_1_album_1_track_2	79x4172 double
mfcc_artist_1_album_1_track_3	79x7086 double
mfcc_artist_1_album_1_track_4	79x15112 double
y	4x1 struct

A) Filename cell is to read file name:



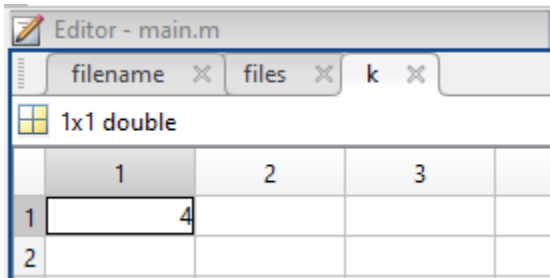
	1	2	3	4	5	6	7
1							
2							
3							
4							

B) Files cell is to read the details of .wav files



	name	folder	date	bytes	isdir	datenum
1	artist_1_alb...	C:\Users\V...	'25-Mar-20...	842790	0	7.3751e+05
2	artist_1_alb...	C:\Users\V...	'25-Mar-20...	1068582	0	7.3751e+05
3	artist_1_alb...	C:\Users\V...	'25-Mar-20...	1814502	0	7.3751e+05
4	artist_1_alb...	C:\Users\V...	'25-Mar-20...	3869094	0	7.3751e+05

C) k describes the no. of files:

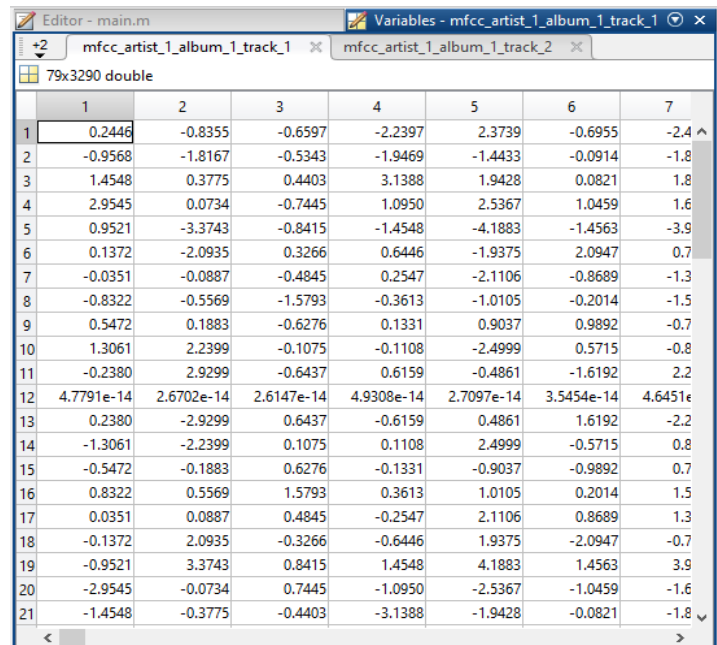


	1	2	3
1	4		
2			

D) The wave is converted into mel-scale

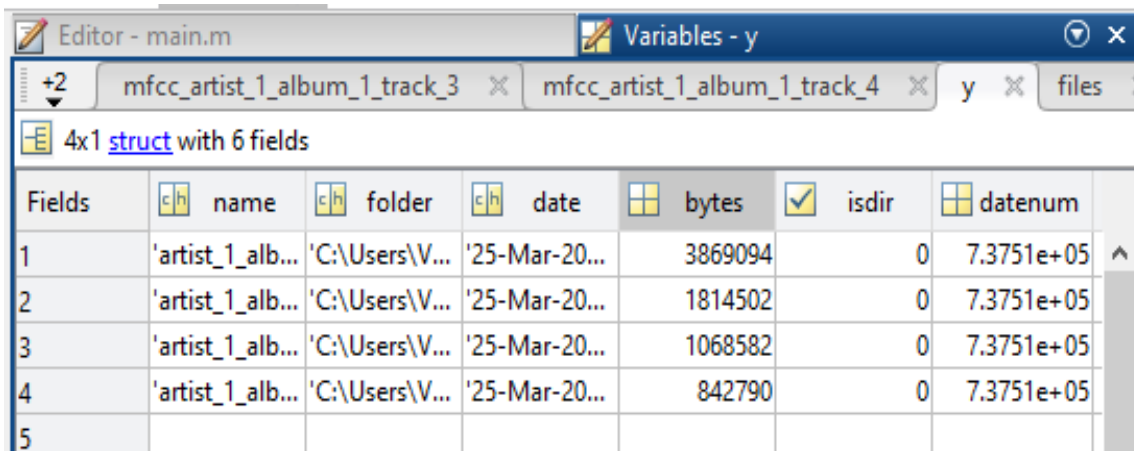
All four cells are generated in mel-scale:

1. 'mfcc_artist_1_album_1_track_1'
2. 'mfcc_artist_1_album_1_track_2'
3. 'mfcc_artist_1_album_1_track_3'
4. 'mfcc_artist_1_album_1_track_4'



	1	2	3	4	5	6	7
1	0.2446	-0.8355	-0.6597	-2.2397	2.3739	-0.6955	-2.4
2	-0.9568	-1.8167	-0.5343	-1.9469	-1.4433	-0.0914	-1.8
3	1.4548	0.3775	0.4403	3.1388	1.9428	0.0821	1.8
4	2.9545	0.0734	-0.7445	1.0950	2.5367	1.0459	1.6
5	0.9521	-3.3743	-0.8415	-1.4548	-4.1883	-1.4563	-3.9
6	0.1372	-2.0935	0.3266	0.6446	-1.9375	2.0947	0.7
7	-0.0351	-0.0887	-0.4845	0.2547	-2.1106	-0.8689	-1.3
8	-0.8322	-0.5569	-1.5793	-0.3613	-1.0105	-0.2014	-1.5
9	0.5472	0.1883	-0.6276	0.1331	0.9037	0.9892	-0.7
10	1.3061	2.2399	-0.1075	-0.1108	-2.4999	0.5715	-0.8
11	-0.2380	2.9299	-0.6437	0.6159	-0.4861	-1.6192	2.2
12	4.7791e-14	2.6702e-14	2.6147e-14	4.9308e-14	2.7097e-14	3.5454e-14	4.6451e
13	0.2380	-2.9299	0.6437	-0.6159	0.4861	1.6192	-2.2
14	-1.3061	-2.2399	0.1075	0.1108	2.4999	-0.5715	0.8
15	-0.5472	-0.1883	0.6276	-0.1331	-0.9037	-0.9892	0.7
16	0.8322	0.5569	1.5793	0.3613	1.0105	0.2014	1.5
17	0.0351	0.0887	0.4845	-0.2547	2.1106	0.8689	1.3
18	-0.1372	2.0935	-0.3266	-0.6446	1.9375	-2.0947	-0.7
19	-0.9521	3.3743	0.8415	1.4548	4.1883	1.4563	3.9
20	-2.9545	-0.0734	0.7445	-1.0950	-2.5367	-1.0459	-1.6
21	-1.4548	-0.3775	-0.4403	-3.1388	-1.9428	-0.0821	-1.8

E) y cell is the output feature vector:



	name	folder	date	bytes	isdir	datenum
1	'artist_1_alb...	'C:\Users\V...	'25-Mar-20...	3869094	0	7.3751e+05
2	'artist_1_alb...	'C:\Users\V...	'25-Mar-20...	1814502	0	7.3751e+05
3	'artist_1_alb...	'C:\Users\V...	'25-Mar-20...	1068582	0	7.3751e+05
4	'artist_1_alb...	'C:\Users\V...	'25-Mar-20...	842790	0	7.3751e+05

CONCLUSIONS FOR FEATURE EXTRACTION:

1. Features are extracted.
2. Since, I (Vamshi) would have plotted interest points(mfc coefficients and DCT) spectrum.
3. But, I have the trouble with the Microsoft Office, I cannot open the files mfccResults.xls and mfccResults.mat data now. So I didn't plot of Spectrum of mfcc coefficients and DCT. I will submit you this plots for the next submission.

FUTURE WORK:

To do:

1. Feature Selection: Requires features of the .wav files
2. Feature Classification: To classify the type of music files
3. End of Project: Project Presentation and Defense

REFERENCES:

- [1] "Computational Models of Music Similarities and their Application in Music Information Retrieval"- A Desseration submitted to Prof. Dipl.-Ing. Dr. techn, Gerhand Widmer, Institute fur Computational Perception – Johannes Kepler Universitat Linz & TU Wien
- [2] Elias Pampal's MA Toolbox- For MALTAB Functions: <http://www.pampalk.at/ma/>
- [3] NPTEL Digital Speech Processing Videos: Mel Frequency Cepstral Coefficients by Prof. S.K.Das Mandal, IIT Kharagpur, India. (For Math behind MFCC)
Youtube Link-1: <https://www.youtube.com/watch?v=E9LGj9s9sbw>
Youtube Link-2: <https://www.youtube.com/watch?v=KzevshgDv8g&t=406s>
- [4] "Mel Frequency Cepstral Coefficients for Music Modelling" by Beth Logan, Cambridge Research Laboratory, Compaq Computer Corporation, Cambridge MA 02142
- [5] Wikipedia: mfcc, cepstrum, history of mfcc.