Speaker Recognition

Arindam Bhattacharyya, University of California, Davis abhattacharyya@ucdavis.edu
Bhawna Sinha, University of California, Davis bhasinha@ucdavis.edu

Abstract

Speaker recognition, a significant technique in the area of digital signal processing is used in a wide range of applications such as security systems, forensics, and human-computer interaction. This project focuses on implementing speaker recognition using MATLAB, employing Mel-frequency Cepstral Coefficients (MFCC), Mel-filterbank (MELFB), and Linde-Buzo-Gray (LBG) algorithm. The proposed system begins with preprocessing the speech signals to extract audio features using the MFCC technique, which mimics the human auditory system's response to sound. Subsequently, the Mel-filterbank enhances discrimination of the extracted features by modeling the frequency response of the voice speech. The MATLAB implementation provides a user-friendly interface for both training and testing the speaker recognition system. Experimental results demonstrate the effectiveness of the proposed approach in accurately identifying speakers from a given dataset.

Keywords: Speaker recognition, MATLAB, MFCC, Mel-filterbank, LBG algorithm, Feature extraction, Clustering.

1. Introduction

Speaker recognition is the process of finding the identity of an unknown speaker by comparing his/her voice with voices of registered speakers in the database. It's a one-to-many comparison.

Speaker recognition can be classified into identification and verification. Speaker identification is the process of determining which registered speaker provides a given voice sample regardless of what is saying. On the other hand, Speaker verification is the process of accepting or rejecting the identity claim of a speaker. In this paper, we are going to implement speaker identification model. Basic structure of speaker identification is given in the fig. 1.

2. Algorithm

2.1. Feature Extraction

Feature extraction is the first step for speaker recognition. In this process a small amount of data is extracted from the voice signal for the identifying a speaker.

2.2. Mel-frequency cepstrum coefficients processor (MFCC)

MFCC is based on the human peripheral auditory system. The human perception of the frequency contents of sounds for speech signals does not follow a linear scale. Thus for each tone with an actual frequency t measured in Hz, a subjective pitch is measured on a scale called the 'Mel Scale' .The mel frequency scale is a linear frequency spacing below 1000 Hz and logarithmic spacing above 1kHz.As a reference point, the pitch of a 1 kHz tone, 40 dB above the perceptual hearing threshold, is defined as 1000 Mels.[1]

2.2.1. Frame Blocking All the recorded audio samples are resampled at 6000Hz. In frame blocking process, each audio file is divided into 20 short frames of around 25 ms time frame in length with overlap of 10 ms. This allows us to split each 1 second sound file into N individual samples. By dividing the signal into such short frames, each section is a relatively constant signal that does not change much.

2.2.2. Windowing Each frame is passed through a windowing function to minimize the discontinuity in the beginning and end of each frame. The concept here is to minimize the spectral distortion by using the window to suppress the signal to zero at the beginning and end of each frame.

Window function for each frame $x_1(n)$ is

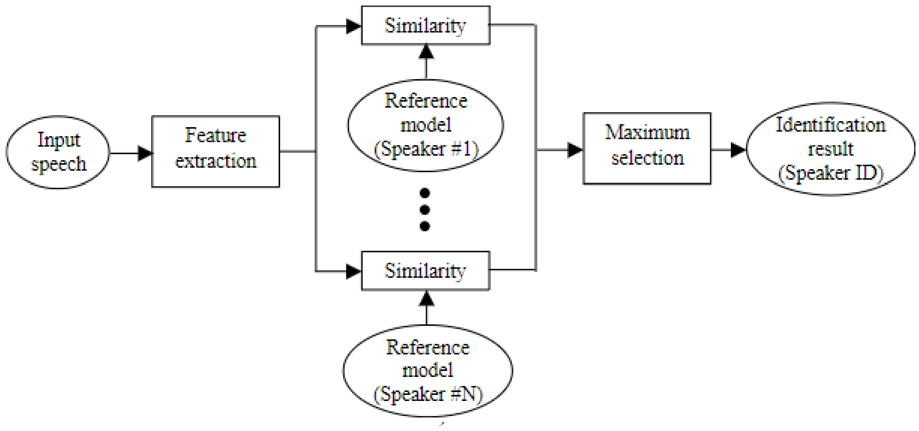

Figure 1. Basic Structure of Speaker Identification

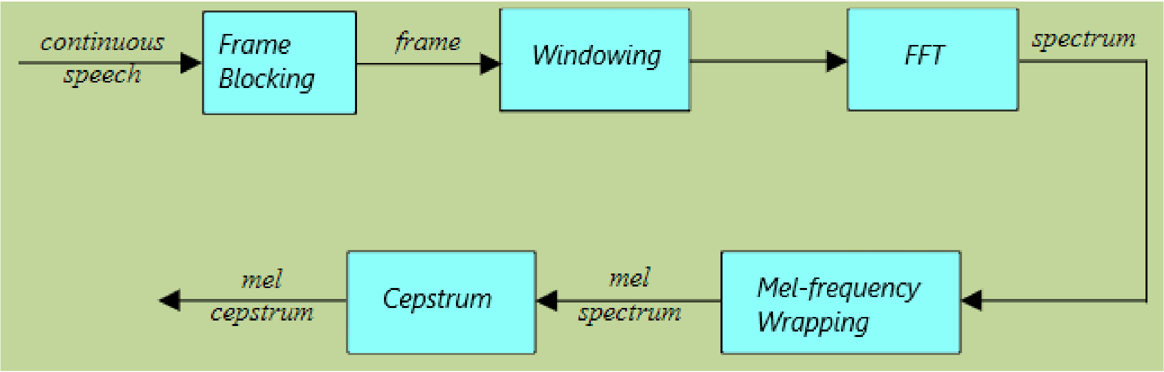

Figure 2. Block diagram of MFCC processor

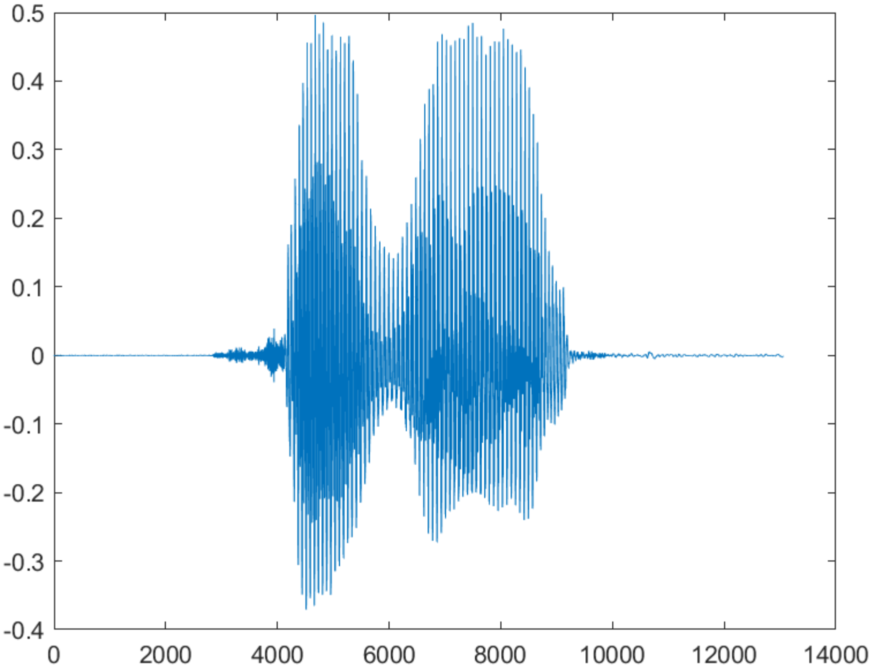

Figure 3. Speech Waveform

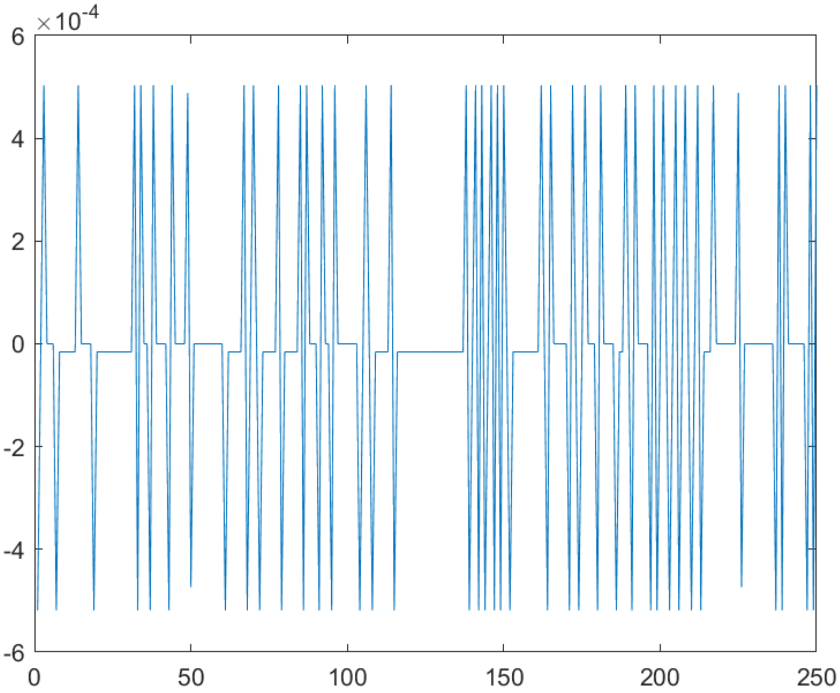

Figure 4. Frames of the Speech signal

$$y_{\mathrm{l}}(n) = x_{\mathrm{l}}(n)w_{\mathrm{l}}(n), 0 <= n <= N-1$$

where N is the number of samples in each frame.

Hamming window is used, which has the form:

$$w_{\mathrm{l}}(n) = 0.54 - 0.46\cos[\frac{2\pi * n}{(N-1)}], 0 <= n <= N-1$$

For twelve testing and training data, Kaiser window is used.

2.2.3. Fast Fourier Transform (FFT) Ordinary .wav files store sound by measuring the amplitude of the signal at a certain sampling rate. Each frame of N samples is transformed from the time domain into the frequency domain. FFT of N samples x_{n} is

$$X_{\mathrm{k}} = \sum x_{\mathrm{n}} e^{-j2kn/N}, k = 0, 1, 2, ..., N-1 n = 0, 1, 2, ..., N-1$$

We get spectrum or periodogram.

2.2.4. Mel-frequency Wrapping Frequencies from the FFT are passed through the Mel scale filter bank. It is composed of triangular band-pass filters of equal width in the Mel-Scale (used to measure frequencies based on their pitch from people). The number of mel spectrum coefficients, K, is typically chosen as 20.

Formula to calculate mels for a given frequency f in Hz is

$$m = 2595 * \log_{10}(1 + \frac{f}{100})$$

2.2.5. Cepstrum In this step, log mel spectrum is converted into time domain using Discrete Cosine Transform (DCT) to get mel frequency cepstrum coefficients (MFCC).

$$C_{\mathrm{n}} = \sum(logS_{\mathrm{k}} \cos[n(k - \frac{1}{2}) * \frac{\pi}{K}], n = 0, 1, ..., K-1$$

k=1,2,...,K

[1] [1].

Folder	Correct	Incorrect	Total	Feedback
Training	11	0	11	All are correct
Testing	6	2	8	S3 and S8 incorrect
Zero Training	18	0	18	All are correct
Zero Testing	11	7	18	2,7,10,12,14,15 and 16 are incorrect
Twelve Training	18	0	18	All are correct
Twelve Testing	10	8	18	2,4,6,7,9,12,14 and 15are incorrect

Table 1. Testing Result

3. Vector Quantization

Vector quantization is used to implement of all learning algorithms. The idea behind it is to treat each n-dimmensional vector from each frame as a point in n-dimmensional space. These points are arranged into k clusters. Linde, Buzo, Gray (LBG) algorithm is used to determine each cluster center. For each speaker, take the array of MFFCs. Center of all these points are mapped by taking the mean of all point. This point will be the first cluster-center. We then split this cluster center into two new centers. Let X be the vector representing the first cluster center. We define

$$X_1 = X(1-e), X_2 = X(1+e)$$

for some small e= 0.01. We then go through all the vectors again and assign each to the cluster center closest to it. Now each vector in the array is assigned to one of these two cluster centers. For each cluster center, we recalculate its position by finding the mean of each vector assigned to it. These new cluster centers are then split again into 16 cluster centers for given speech data and 32 for zero and twelve audio files. This process of splitting and recalculating means is repeated until the specified number of cluster centers is found. The result is a collection of cluster centers called a “codebook”. This codebook will represent the way a speaker “sounds” and is ultimately the tool to classify which speaker is assigned to a new speech file.

[2]

4. Result

The training sample audio was able to achieve a 100% accuracy rate while the training data achieved a 80% success rate. This discrepancy can most likely be attributed to sampling frequency or number of centroids. Modifying the appropriate matlab code should greatly increase the models accuracy.

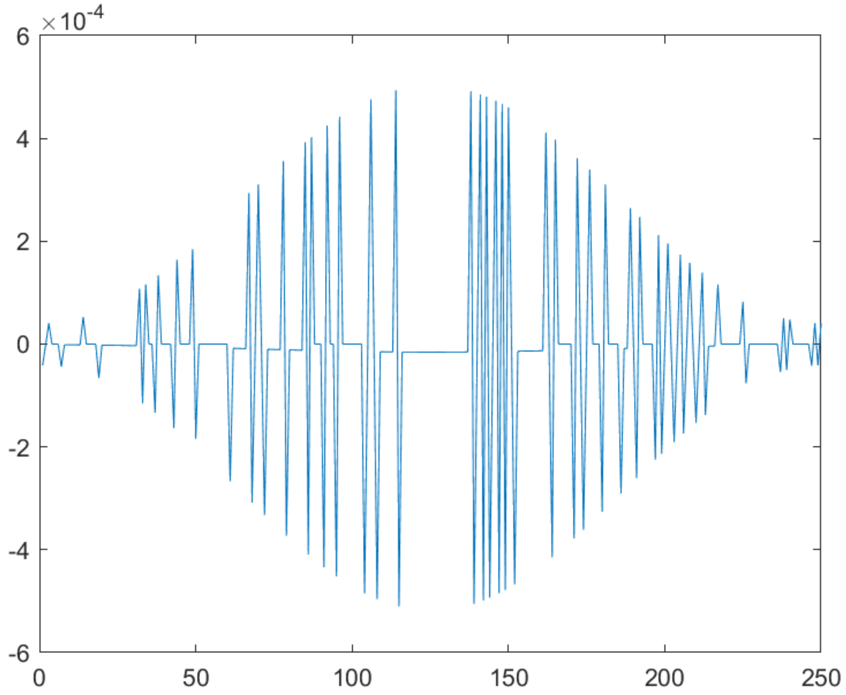

Figure 5. FFT of the frames

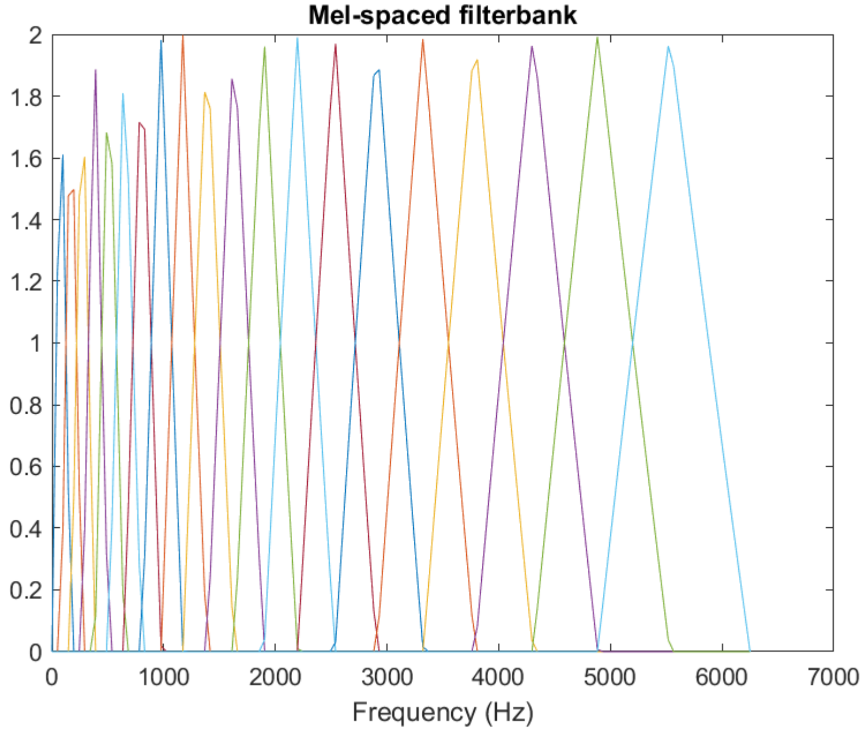

Figure 6. Mel spaced filterbank

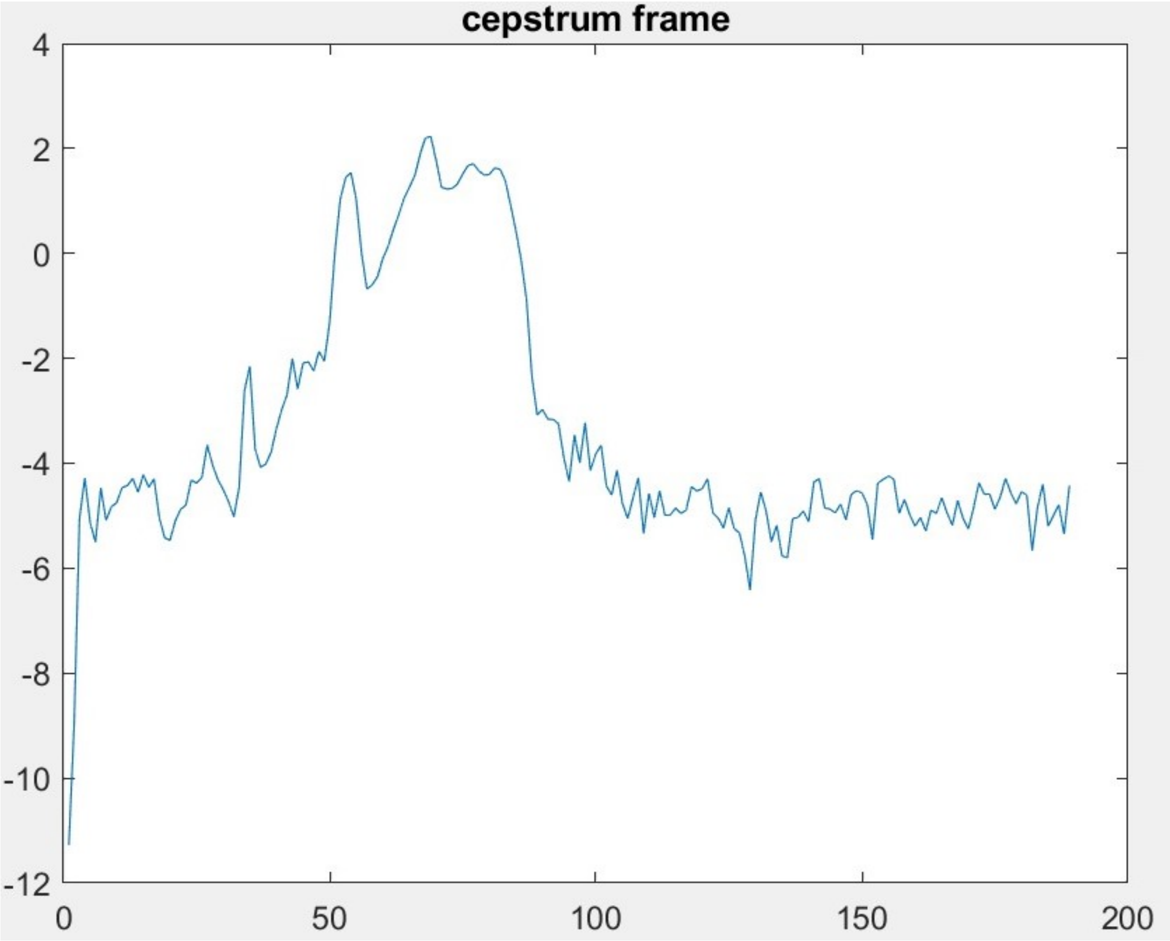

Figure 7. Cepstrum Frame

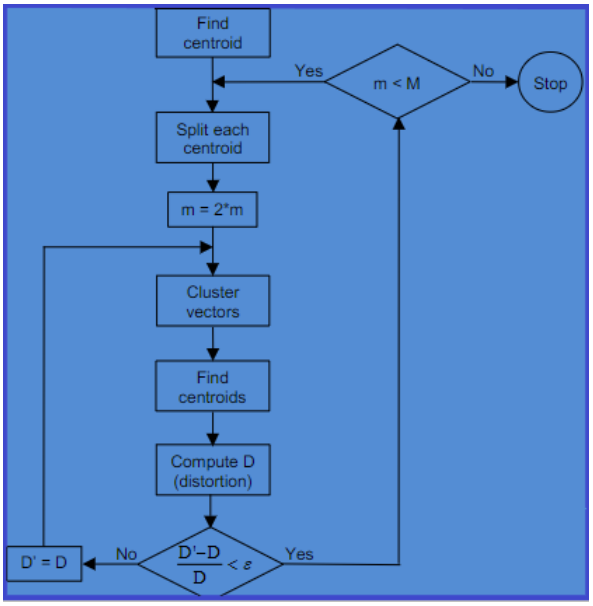

Figure 8. Flowchart of LBG

References

[1] V. Tiwari, “Mfcc and its applications in speaker recognition,” *International Journal on Emerging Technologies*, vol. 1, pp. 19–22, 2010.

[2] R. G. Yoseph Linde, Andres Buzo, “An algorithm for vector quantizer desig,” *IEEE Transactions on Information Theory*, vol. 28, no. 1, 1980.