**EEC201 Final Project Report**

Yixin Cheng

**Introduction**

This project aims at generating a speaker recognizing system by analyzing unique information induced in every speech waves. The project can be divided into two critical sections: making a codebook and feature matching(Fig.1).

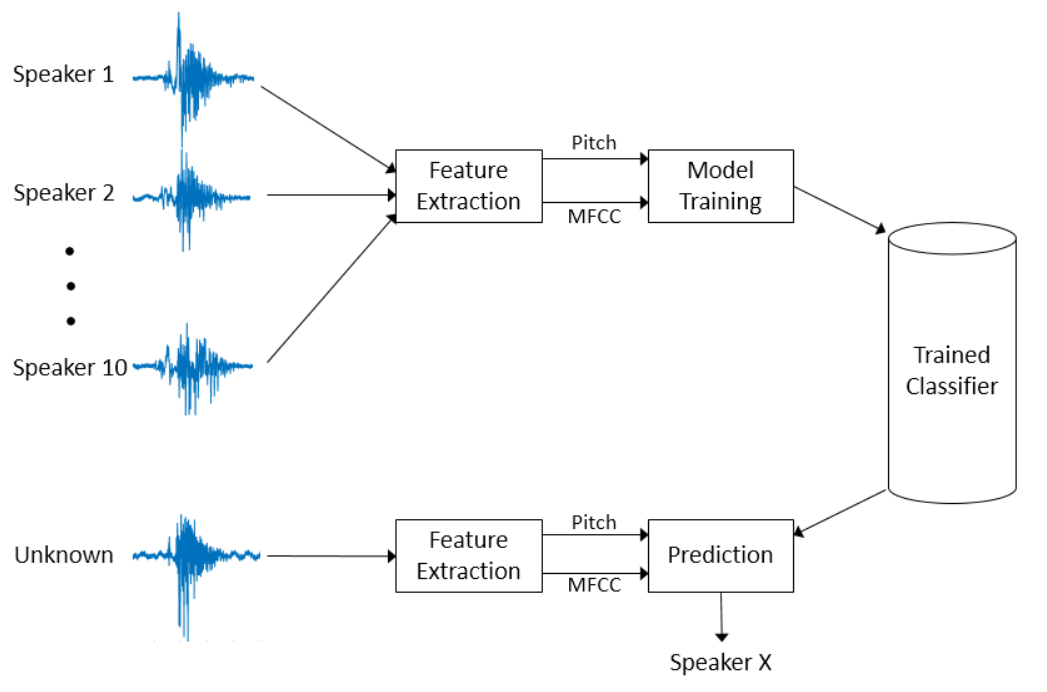


Fig.1 Schematic of speaker recognition system

In both making codebook(training) and feature matching process(testing), the features of the input signals should be extracted. First, the input signal is transferred into frequency domain using Short-Time Fourier Transform, and then translated to MFCC.

In training section, by using LBG algorithm, clusters and their centroids can be find out. Every speaker has his/her own centroids, which can be stored in a table so-called the codebook. After a codebook has been generated, we can identify an unknow input signal by compering it centroids with the centroids inside the codebook, and then find out the speaker’s ID.

There are 4 main step to design this system: preprocessing, training, optimizing and testing.

**Preprocessing**

图表, 直方图

描述已自动生成Since there are some zero points and very small signal existing in the input signal, we need to delete those points. Otherwise, they will decrease the accuracy of the feature extraction, and our codebook will contain many meaningless information. Thus, the first step of preprocessing is to remove zero points and tiny signals. Only the data with the magnitude no smaller than -30dB (to maximum data) can be treated as valid signal. By doing that, the signal used in training and testing containing only vocal information(Fig.2.A)

图表, 直方图

描述已自动生成

Fig.2 (A)s4 in time domain after removing invalid data. (B)Origin s4 signal

Second, MFCC information will be generated(Fig.3). Comparing the MFCC diagram of different speaker, it’s clear that the value of MFCCs of different speakers is also different. Every input signal has its unique MFCC, so indicating that the MFCC amplitudes could clearly distinguish between speakers. MFCC values make up the clusters. Form a specific input signal, its MFCC cluster is unique. By comparing these clusters with the centroid gained from training process, the system can distinguish the speaker ID. According to Fig.4, it’s easy to figure out that MFCC cluster of different signals locates at different area.

图表, 条形图

描述已自动生成图表, 条形图

描述已自动生成

Fig.3 MFCC vs time for s1 and s6

图表, 散点图

描述已自动生成

Fig.4 Compare mfcc of s3 and s6

**Parameter optimization**

To achieve higher feature matching rate, parameters should be optimized. The following is what we need to focus on:

N: number of the samples per frame;

M: overlap, equals to ;

p: number of Mel filter bank used in creating codebook

q: number of MFCC used in creating codebook

K: number of clusters in creating codebook

N=256, 240, 220, 200 and 180 are selected to be the possible default value. After the first step of getting optimized K, we get the following

|  |  |  |
| --- | --- | --- |
| N | M | K |
| 256 | 171 | 62 |
| 240 | 160 | 59 |
| 220 | 147 | 59 |
| 200 | 133 | 46 |
| 180 | 120 | 52 |

图表, 折线图

描述已自动生成图表, 折线图

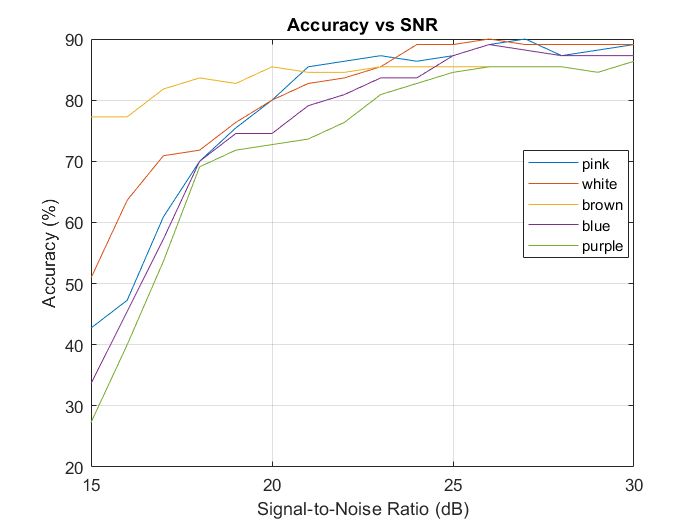
描述已自动生成图表, 折线图

描述已自动生成图表, 折线图

描述已自动生成It’s clear to find out that the for the other 4 input N, the K value after optimization is too high, which wastes too much time in the later speak recognition. Thus, N=200 is chosen to be the default value. Using the default M and N, other three optimized values come out by plotting the accuracy diagram. By comparing the distortion between the calculate centroids of unknow signal and the closest centroids in the codebook, recognition accuracy is obtained. Then we set the default value of all input parameter as N=200, M=133, p=33, q=25, K=7.

Fig. 5 Optimization result for K, p and q

After getting the optimized parameter M, N, p, q and K, the accuracy of signal adding different kind of noise is plotted out. We set the range of SNR from 15-30dB, and the accuracy can reach higher than 80% when SNR is above 22dB.



**Testing**

I. Generate the codebook of all 11 .WAV document in ‘Data/train’ folder.

II. Make a matrix containing number from 1 to 8 with random order. And load test samples with this order to test.

III. Run recognizing program and check whether the output speaker ID order is same as that of the matrix in step II.

25 turns have been runed, which means 200 samples have been recognized to test accuracy. The final accuracy can reach 83%.