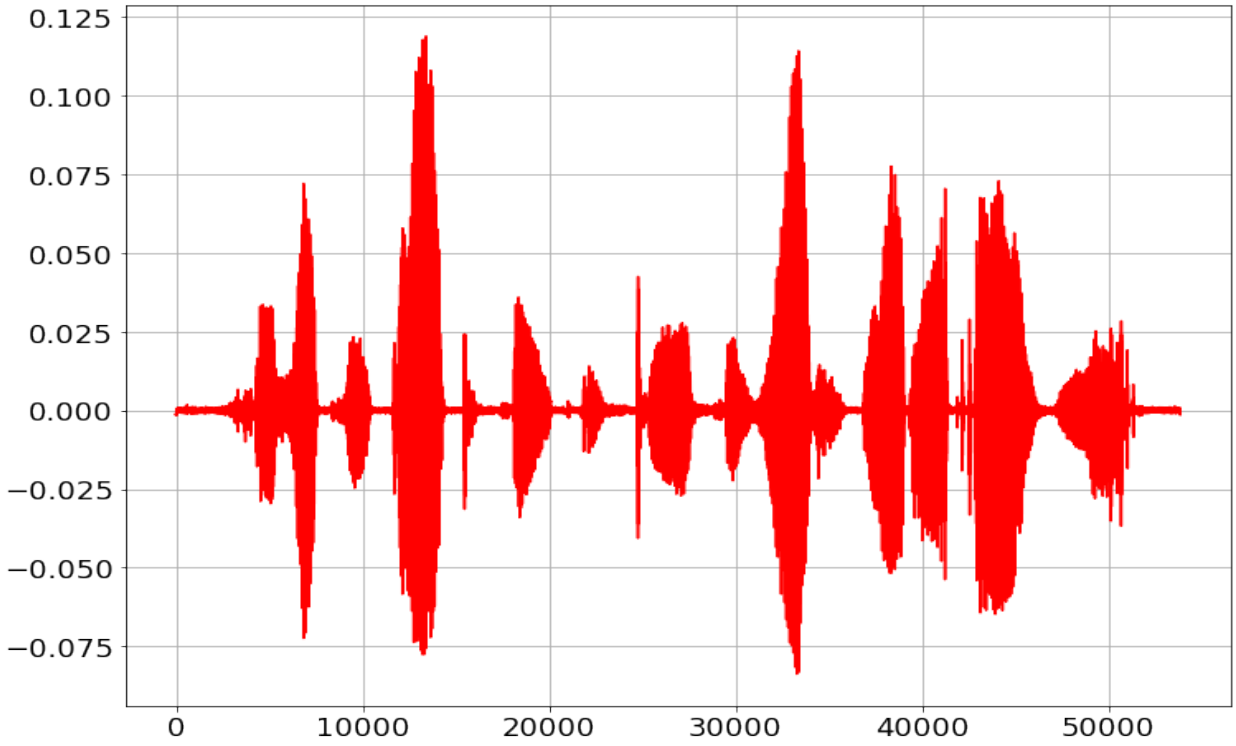
CS6300 - Speech Technology
Mini Project 1 - GMM

Dhanush Krishna : EE16B009
Vignesh S : EE16B127

October 21, 2020



(a)

Figure 1: Waveform of sa1.wav

Problem

The objective is to build a "text independent speaker identification/verification system."

TIMIT/NTIMIT databases from which feature vectors have to be extracted at 100 frames/sec will be supplied. Programs will be made available for feature extraction. These datasets have about 630 speakers. Each group will get about 200 speakers' data. The objective is to build Vanilla GMM System.

Extract features using standard cepstra or MFCC library from all audio files for given speakers. Keep 80% of sequences of feature vectors for every speaker as train templates. Use the remaining for test.

Visualisations

The values for accuracy for different values of number of components and number of iterations are shown in Figure 4.
The accuracy is plotted with different values of number of components and number of iterations are shown in Figure 5 and 6.

$$P(\vec{x}|\lambda_s) = \sum_{k=1}^{M} w_k \mathcal{N}(\vec{x}|\vec{\mu}_k, \Sigma_k)$$

$$\lambda_s = \{w_k, \vec{\mu}_k, \Sigma_k\}_{k=1}^{M}$$

$$\sum_{k=1}^{K} w_k = 1$$

(a)

1) Random initialization of $\vec{\mu}_k, \Sigma_k$ and w_k

2) Expectation-Step

Align vectors to model

$$\gamma_{nk} = \frac{w_k \mathcal{N}(\vec{x}_n|\vec{\mu}_k, \Sigma_k)}{\sum_{j=1}^{M} w_j \mathcal{N}(\vec{x}_n|\vec{\mu}_j, \Sigma_j)}$$

$$N_k = \sum_{n=1}^{N} \gamma_{nk}$$

$\rightarrow$ γ_{nk} is the responsibility of k-th component towards n-th feature vector

(b)

Figure 2: GMM 1, 2

3) Maximization-Step

Update model parameters by maximum likelihood estimation (MLE)

$$\hat{\vec{\mu}}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} \vec{x}_n$$

$$\hat{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} (\vec{x}_n - \hat{\vec{\mu}}_k)(\vec{x}_n - \hat{\vec{\mu}}_k)^T$$

$$\hat{w}_k = \frac{N_k}{N}$$

(a)

Figure 3: GMM 3

Observations and Inferences

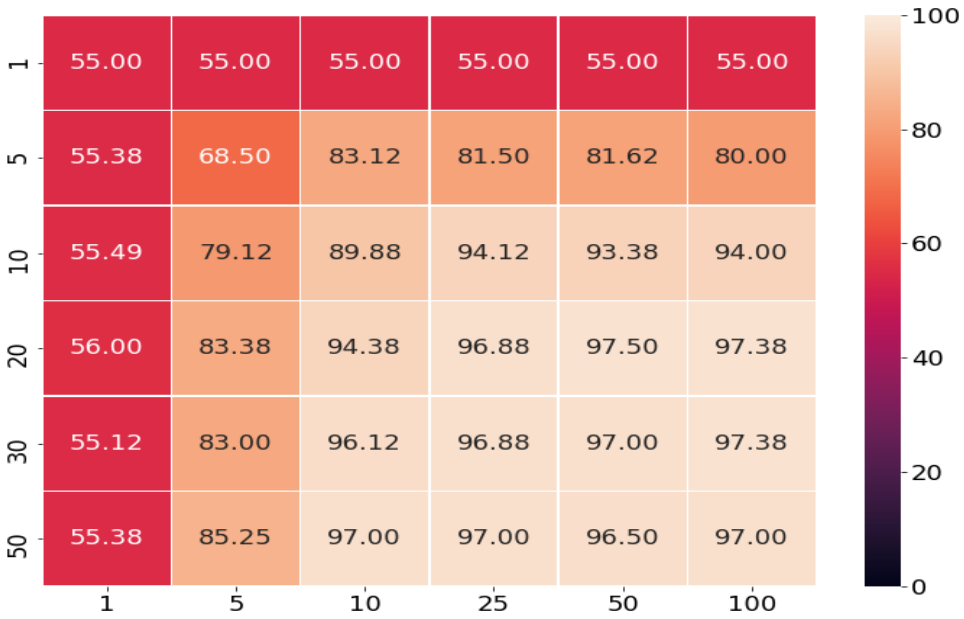
The cepstrum coefficients are derived by taking the Inverse Fourier Transform (IFT) of the logarithm of the fourier transform of the time domain signal.
The cepstral coefficients are defined as,

$$c[n] = \mathcal{F}^{-1}\{log(\mathcal{F}\{x[n]\})\}$$

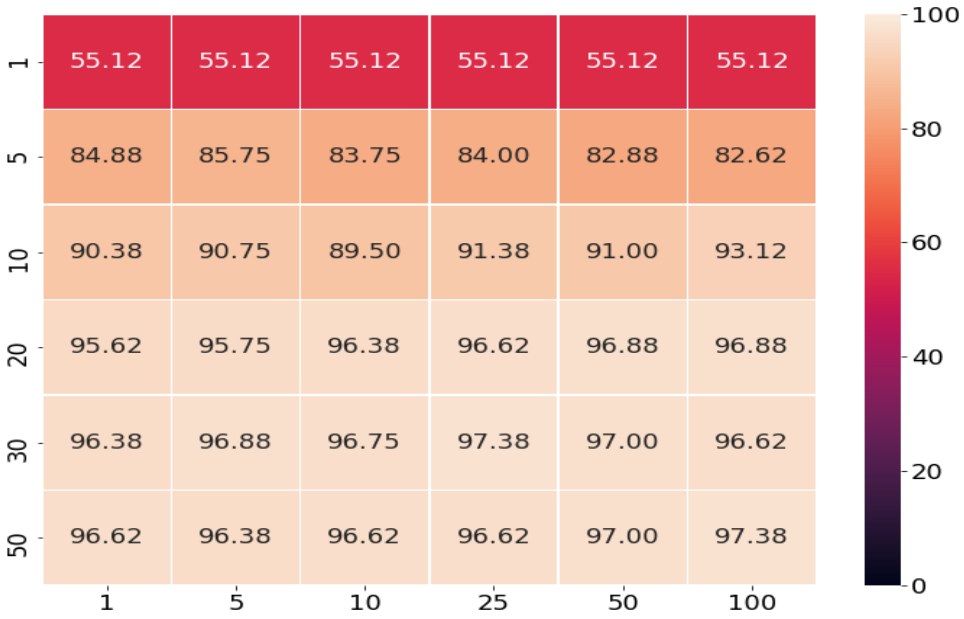
Steps

Here is the flow of different steps for the process:

1. Extracting speaker names: Converting text file team_15.txt to csv file, using delimiter and extracting the speaker names for the audio files from TIMIT/NTIMIT database.
2. Extracting the audio files of the speakers from TIMIT database using os.walk
3. Computing the cepstral coeffcients
4. Dividing the train and test features.
5. Computing coefficients for GMM, assigning labels to test feature vectors and computing accuracy.

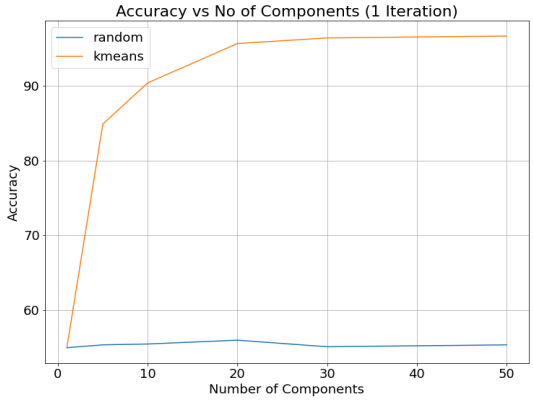


(a)

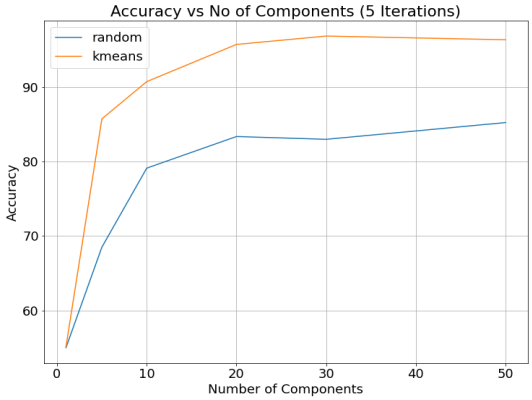


(b)

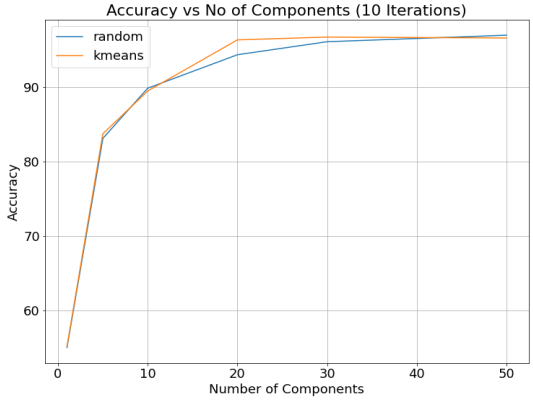
Figure 4: Number of Components vs Number of Iterations - Accuracy - Random and KMeans



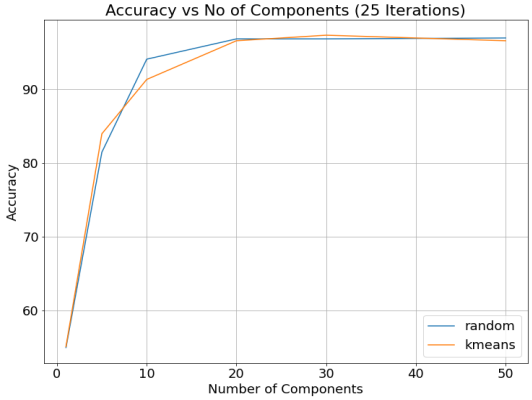
(a)



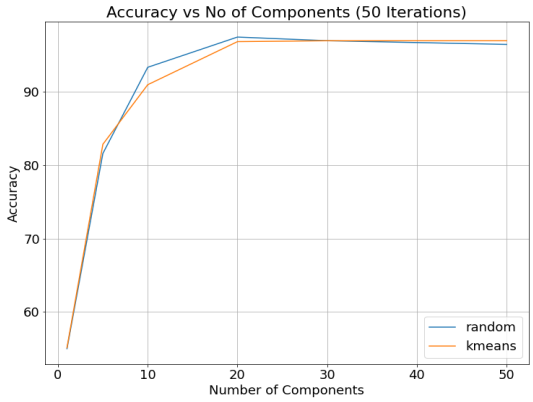
(b)



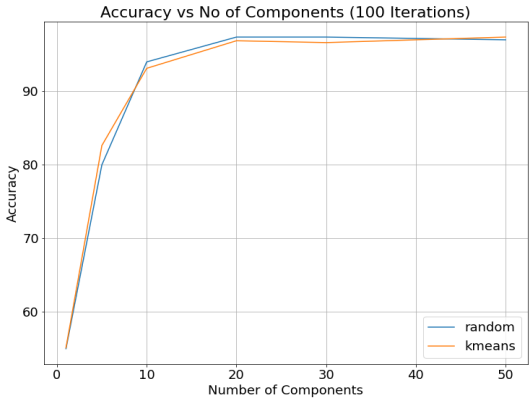
(c)



(d)

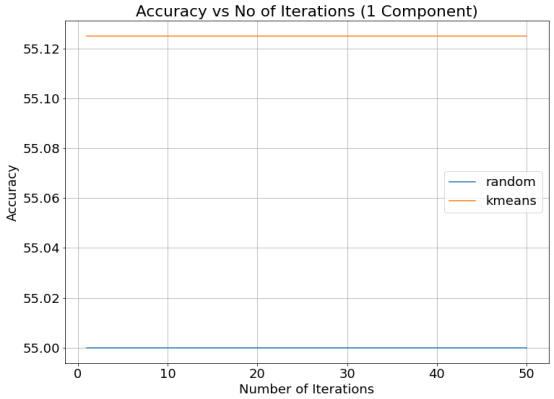


(e)

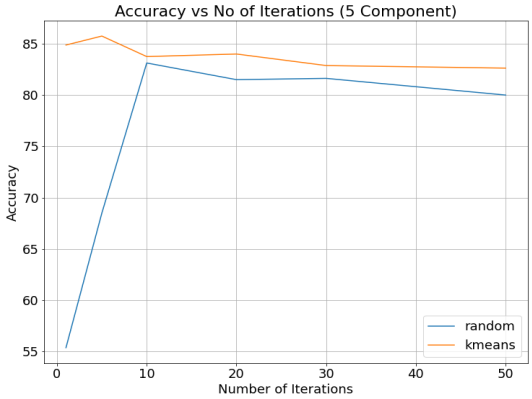


(f)

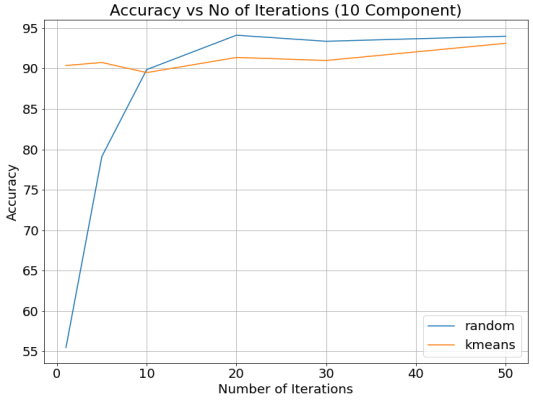
Figure 5: Accuracy vs Number of Components



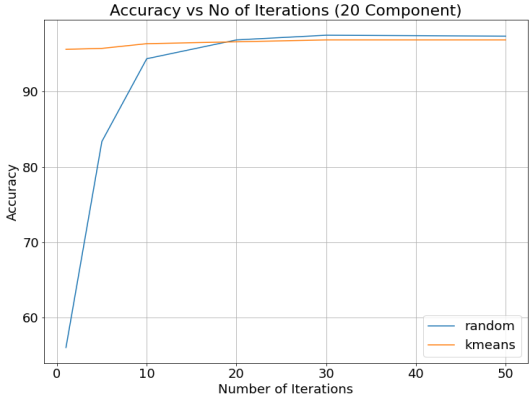
(a)



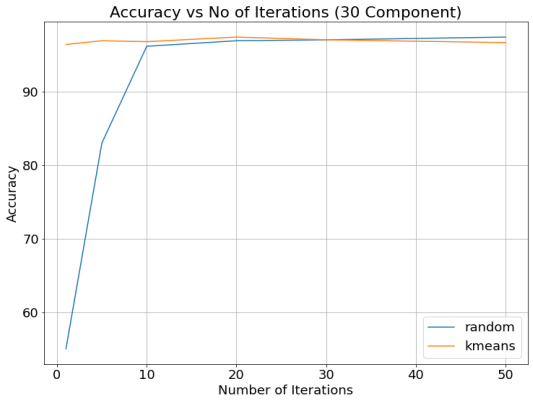
(b)



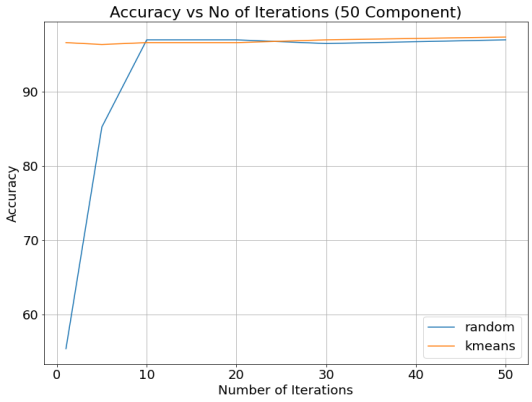
(c)



(d)

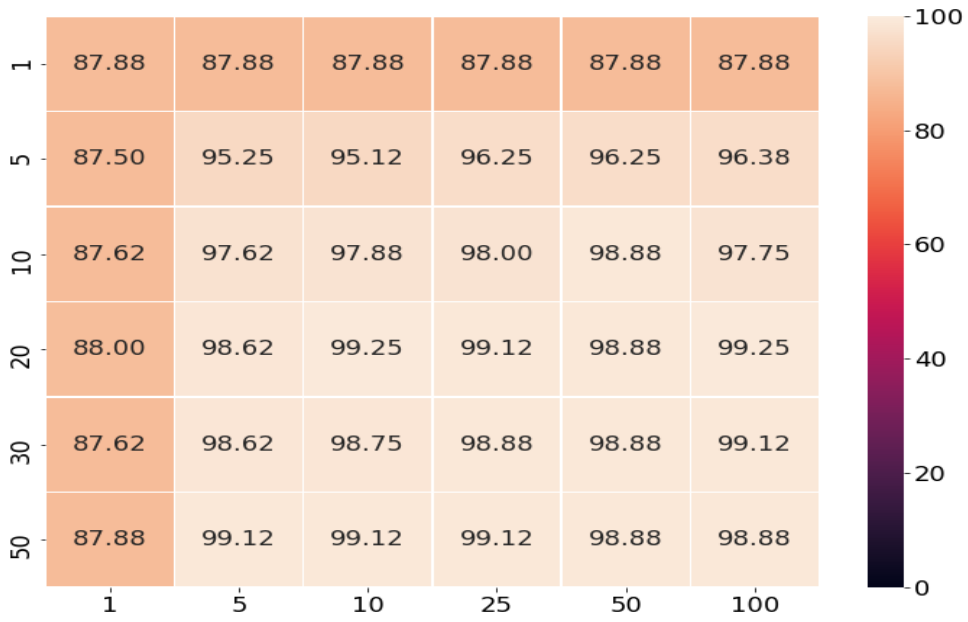


(e)



(f)

Figure 6: Accuracy vs Number of Iterations



(a)

Figure 7: No. of Components vs No. of Iterations - Accuracy - 20 MFCC

After completing this for all the test signals the accuracy is measured.
The predicted labels and the test labels are checked against one another and the accuracy is computed,

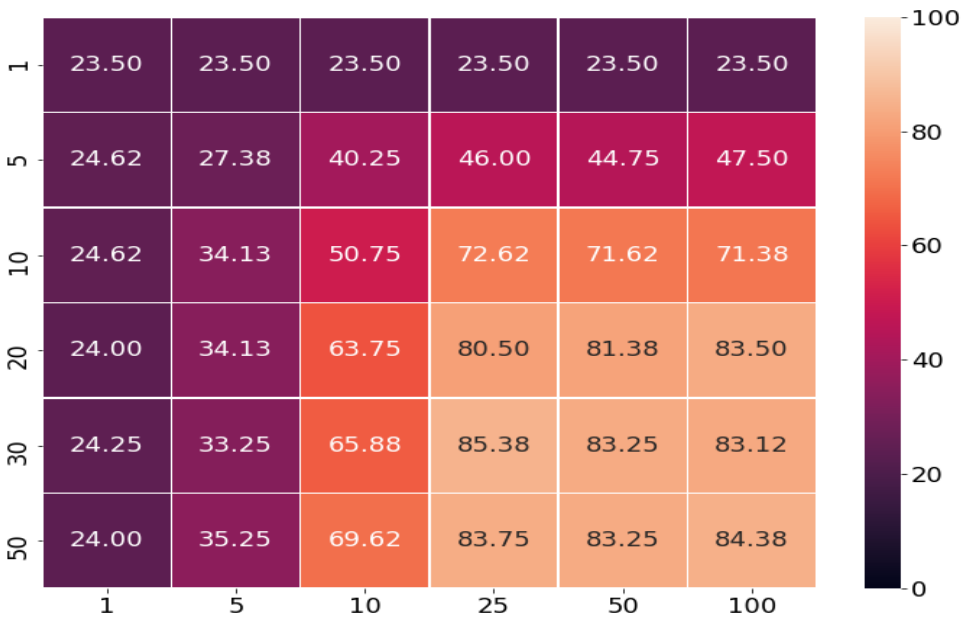
$$Accuracy = \frac{\text{Number of correct labels}}{\text{Total number of labels}}$$

We haven't used a confusion matrix here because the size of the number of labels is large, 200 labels.

Here, number of mfcc coefficients taken is 10. The accuracy for random initialisation is less than that for k-means. This is to be expected since k-means is a better way to initialise. Also, for higher number of components the accuracy increases sharply than compared to a higher number of iterations in the case of random initialisation, but remains more or less constant for k-means initialisation.

In Figure 7, we have the heat map for accuracy depending on number of components and number of iterations for 20 MFCC coefficients.

As you can see, for higher number of MFCC coefficients the accuracy is high even for lower number of components and iterations. In Figure 8, you can see when we use only 5 MFCC coefficients, we get very poor accuracy and moderate accuracy even for high number of components and iterations.



(a)

Figure 8: No. of Components vs No. of Iterations - Accuracy - 5 MFCC