Efficacy of a Gaussian Mixture Model based Speaker Recognition in Machine Learning Framework

Final report on the project work submitted for fulfillment of summer internship by

Adhiraj Banerjee

3^{rd} Year B.Tech, Electronics & Comm. Engg.

Indian Institute of Engineering, Science and Technology (IIEST), Shibpur

Under the Supervision of

Dr. Siva Ram Krishna Vadali

Senior Principal Scientist

Robotics & Automation Division

CSIR-Central Mechanical Engineering Research Institute, Durgapur

August 2021

Contents

List of Figures

List of Tables

1 Introduction

Speech Recognition is a *text-dependent* method, where it highly depends on the language and corpus. On the other hand, Speaker Recognition mainly focusses on raw audio percepts (and from the information derived therein) to identify the uniqueness aspect, if any, among different speakers [**1**]. Speaker Recognition (SR) has numerous applications such as voice biometric, forensics, traitor finding, voice-mail and tele-banking [**2**]. Conceptually, SR determines the presence of the active speaker among a set of registered speakers. In view of the applicability of Machine Learning framework to a wide variety of classification problems, the present work studies the popular Gaussian Mixture Model (GMM) based ML (GMM-ML) [**3**] approach as a potential solution for speaker recognition. In this work, voice samples of ten different speakers are recorded and the performance of GMM-ML for SR is analyzed. Further, performance of GMM-ML is also studied for SR when the speech samples under test is overlaid with additive white Gaussian noise. It is observed that GMM-ML based SR is fairly accurate in distinguishing the correct speaker from the 10 registered speakers.

The rest of the report is organized as follows: Section 2 discusses ML for speaker recognition. In this section, GMM-ML using expectation maximization (EM) is studied in detail. Section 3 presents the results and inferences on speaker recognition accuracy using GMM-ML for voice samples. Section 4 provides a few conclusive remarks on the work done.

2 Machine Learning for Speaker Recognition

Machine Learning (ML) is a mathematical framework which helps systems to learn and train a model with (preferably) large data sets and subsequently predict the category into which a test data may be classified to [**4**]. The present work studies the efficacy of a ML based classification algorithm for Speaker Recognition. Going by principles of ML, a model for Speaker Recognition is trained such that the model identifies unique patterns (referred as features) from a set of different voice samples (of registered speakers) and distinguish the actual speaker voice sample from the rest. In the present work, Gaussian Mixture Model (GMM) based classification, one of the popular ML methods, is chosen to study its efficacy in accurate speaker recognition.

2.1 Gaussian Mixture Model based SR

It is known that the Gaussian Mixture Model (GMM) is a powerful model used to solve clustering based classification problems. It is a probabilistic approach as it approximates the probability distribution of a MXK length data (M, K dimensional data vectors, where M is number of data points) belonging to a class λ (of L possible classes) as a linear combination of N Gaussian distributions or clusters with mean vector $\boldsymbol{\mu}=\{\mu_i\}_{i=1,2,...,N}$ (each of order $KX1$), variance-covariance matrix $\boldsymbol{\Sigma}'=\{\boldsymbol{\Sigma}_i\}_{i=1,2,...,N}$ (each of order KXK) and $\boldsymbol{\omega}=\{\omega_i\}_{i=1,2,...,N}$ as the weights in the linear combination. In *Speaker Recognition*, features of speaker's voice sample are extracted into a

feature vector and used as data points for processing to identify the registered speaker. Since the feature vector is *multi-dimensional*, the probability density function (PDF) of particular class of speaker (λ, of L possible speakers) can be expressed as a linear combination of N multivariate Gaussian PDFs [**5**] as:

$$P(\boldsymbol{X}|\lambda) = \sum_{i=1}^{N} \omega_i \cdot P(\boldsymbol{X}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \tag{1}$$

where,

$$P(\boldsymbol{X}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \prod_{n=1}^{M} P(\boldsymbol{x}_n|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i); \tag{2}$$

λ, as mentioned earlier, is the class to which a speaker is associated with; $\boldsymbol{X}$ represents the N dimensional training data; and $P(\boldsymbol{x}_n|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ is the multivariate Gaussian PDF given by,

$$P(\boldsymbol{x}_n|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \left((2\pi)^K \cdot |\boldsymbol{\Sigma}_i|\right)^{-\frac{1}{2}} \cdot exp\left(-\frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_i)\right). \tag{3}$$

In the formulation in Equation 3, the likelihood, computed from the input data points in vector X is assigned to a particular class corresponding to the speaker's GMM. Herein, the mean vectors ($\boldsymbol{\mu}_i$), variance-covariance matrices ($\boldsymbol{\Sigma}_i$) and weights of each of the gaussian components ($\boldsymbol{\omega}_i$) is updated iteratively till the likelihood for the data points (i.e. feature vectors) is maximized, thereby creaing a trained model with the extracted features. The maximized likelihood will thus be assigned to the class corresponding to the speaker, in turn creating the the classified GMM for the speaker.

2.2 Expectation-Maximization (EM) Algorithm

In a Gaussian Mixture Model, the mean vector, the covariance matrix and the weights of each Gaussian components required for obtaining the maximum of the log-likelihood function of datapoints is unknown. E (Expectation) - M (Maximization) [**6**] is an iterative statistical algorithm which is used to train GMM by iteratively updating the unknown parameters ($\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$ & $\boldsymbol{\omega}_i$) till convergence is achieved. Consequently, the EM algorithm assumes great significance. The EM algorithm is computed as follows:

2.2.1 Initialization of Unknown Vector Parameter(s)

First, the mean vector ($\boldsymbol{\mu}_i$), covariance matrix ($\boldsymbol{\Sigma}_i$) and weights ($\boldsymbol{\omega}_i$) for each of the N Gaussian Distributions or clusters are initialized with arbitary values. From these initialized values, the initial log-likelihood is evaluated.

2.2.2 Computation of Responsibilities of Gaussian Components

In Equation (3), weights $\{\omega_i\}_{i=1,2,...,N}$ can be considered as a prior probabilities for N Gaussian clusters. Next, for each of the M K dimensional vectors (i.e. K dimensional $\boldsymbol{X} = \{\boldsymbol{x}_n\}_{n=1,2,...,M}$'s) data points, their responsibilities[1] $\{\boldsymbol{\gamma}_i(\boldsymbol{x}_n)\}_{i=1,2,...N}$ are computed for N Gaussian components are determined. The responsibilities of M data points can be computed as their corresponding *posterior probabilities* from the initialized parameters of their respective

[1]Responsibilities in the context may be interpreted as "Contributions"

Gaussian components using Bayes' Theorem:

$$\boldsymbol{\gamma}_i(\boldsymbol{x}_n) = \frac{\omega_i \boldsymbol{\mathcal{N}}(\boldsymbol{x}_n|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\sum_{j=1}^{N} \omega_j \boldsymbol{\mathcal{N}}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \tag{4}$$

and the latest weight for the i^{th} cluster is given by

$$\omega_i = \frac{M_i}{M}, \tag{5}$$

where M_i is the effective number of data points assigned to the i^{th} Gaussian cluster and M is the total number of datapoints for each speaker.

2.2.3 Estimation of Vector Parameters

Next, estimates of the parameters of each of the N Gaussian components are computed again using the current responsibilities of each data vector $\{\boldsymbol{x}_n\}_{n=1,2,...,M}$ for a particular Gaussian component. Mathematically, paramaters of the i^{th} Gaussian components is given by:

- Mean Vector ($\boldsymbol{\mu}_i$):

$$\boldsymbol{\mu}_i = \frac{\sum_{n=1}^{M} \boldsymbol{\gamma}_i(\boldsymbol{x}_n)\boldsymbol{x}_n}{\sum_{n=1}^{M} \boldsymbol{\gamma}_i(\boldsymbol{x}_n)}. \tag{6}$$

  Note that, for a given speaker the Mean Vector ($\boldsymbol{\mu}_i$) is of the order of NXK .

- Covariance Matrix ($\boldsymbol{\Sigma}_i$):

$$\boldsymbol{\Sigma}_i = \frac{\sum_{n=1}^{M} \boldsymbol{\gamma}_i(\boldsymbol{x}_n)(\boldsymbol{x}_n - \boldsymbol{\mu}_i)(\boldsymbol{x}_n - \boldsymbol{\mu}_i)^T}{\sum_{n=1}^{M} \boldsymbol{\gamma}_i(\boldsymbol{x}_n)}. \tag{7}$$

  Note that, for a given speaker, a total of N KXK Covariance Matrices ($\boldsymbol{\Sigma}_i$'s) are computed.

- Weight Vector ($\boldsymbol{\omega}_i$) of i^{th} Cluster:

$$\boldsymbol{\omega}_i = \frac{1}{M} \sum_{n=1}^{M} \boldsymbol{\gamma}_i(\boldsymbol{x}_n). \tag{8}$$

Note that, for a given speaker, a total of N weights (ω_i) are computed for each of the N clusters.

2.2.4 Evaluating the Log-Likelihood

Next, using the parameters estimated for each of the Gaussian components, the log-likelihood of the data points of a given class / speaker (λ) is computed, as follows:

$$\ln P(\boldsymbol{X}|\lambda) = \sum_{n=1}^{M} \ln \sum_{i=1}^{N} \boldsymbol{\omega}_i \boldsymbol{\mathcal{N}}(\boldsymbol{x}_n|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i). \tag{9}$$

Next, the criteria for convergence in the E-M Algorithm is examined / compared as follows:

1. If the parameters of the Gaussian components ($\boldsymbol{\mu}_i$,$\boldsymbol{\Sigma}_i$, and $\boldsymbol{\omega}_i$) does not

change in the last successive iterations and

2. If the log-likelihood of the data points for a particular speaker remains almost unchanged in the last few iterations (thereby indicating attaining maxima of the function)

stop iterating and use the EM computed estimates for further processing (i.e. identification of the speaker in the present application). Note that, if convergence of data points for N Gaussian clusters is not fulfilled, all the steps from the second step need to be repeated. Figure 1 shows the a schematic of the flow of GMM-ML based speaker recognition.

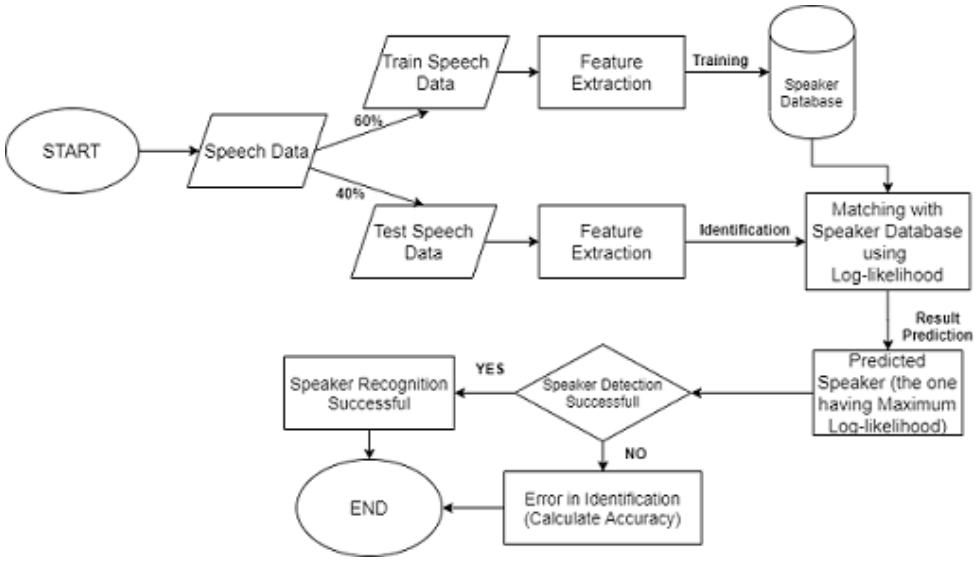

Figure 1: Flow chart of GMM-ML based speaker recognition

2.3 Feature Extraction in SR: Mel-Frequency Cepstrum Coefficients (MFCC)

The features extracted from the voice samples are referred as *Mel-Frequency Cepstrum Coefficients (MFCC's)* [**7**]. In the context of audio signal processing, MFCCs have become a prominent feature to be used for feature extraction from train data, because it gives an idea about the perceived difference between frequencies, especially in higher frequency region. Its importance comes from the fact that, the human ear can perceive voice signal frequncies in a *non-linear* fashion. Clearly, for further signal processing, audio frequencies needs to be analyzed in the *Mel* scale, rather than the Hertz scale. Moreover, since there is a logarithmic relationship between the Mel and Hertz scale, frequencies in Mel scale becomes more useful. The extraction of MFCCs can be accomplished as explained in the following modules:

2.3.1 Pre-Emphasis

After discretization of the analog voice signal, since the frequency range where the human ear tends to perceive less in linear scale, first, Pre-Emphasis [**8**] is performed to increase the energy of the signal at higher frequencies.[2] Pre-Emphasis is performed by passing the signal through a *first order high pass filter*. The discrete time equation for filtering the the audio signal (x[n])

[2]Pre-Emphasis is mainly done for transmission cases which induces the significance of the presence of higher frequency components in the voice signal.

and the filtered output signal (y[n]) is given by:

$$y[n] = x[n] - \alpha x[n-1] \,\forall\, \alpha \in (0.9, 1) \tag{10}$$

The Z-transform of the digital filter is given by:

$$H(z) = 1 - \alpha z^{-1} \tag{11}$$

2.3.2 Extraction of Frames

Due to changes in *Prosody* (i.e. features of voice) and *random variations* in the vocal tract, the voice signal is a *non-stationary signal*. However, within short intervals the voice signal is assumed to be *stationary*, and may hence be analyzed over short time windows. Hence, for analysis of voice features, the speech signal is divided into frames of length N and M samples from adjacent frames are overlapped. Figure 2 shows frames to be extracted from a recorded audio / voice signal.

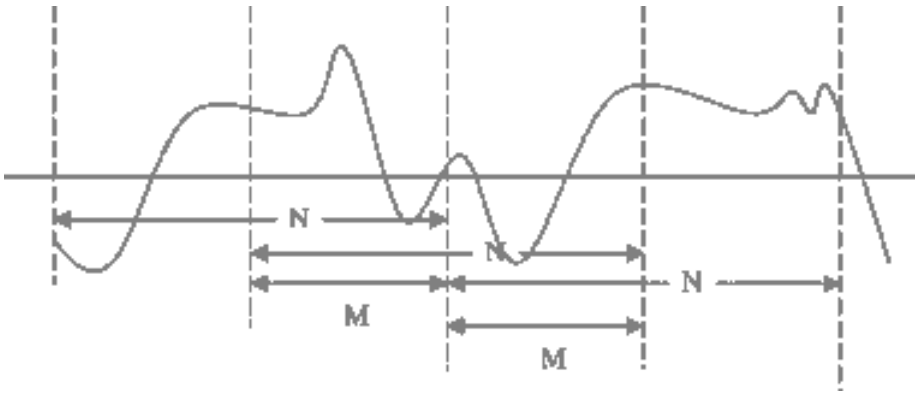

Figure 2: Extraction of frames from voice signal

2.3.3 Smoothing Voice Signals with Filters

When the audio signal is divided into short frames, discontinuities are formed at the edges of the frame, which are incongruent to the input voice signal. Such discontinuities impact the signal by changing its statistical properties. In order to reduce the effect, voice signal is smoothened by multiplying each frame with a window, i.e. filtered such that a smoothing behavior is enforced leading to zero signal strength at the borders. Generally, the signal frames are multiplied with the *Hamming Window* [**9**]. Considering the window to be w[n], the equation for windowing is :

$$y[n] = x[n] \cdot w[n] \tag{12}$$

where, $x[n]$ is the input signal, $y[n]$ is the output signal and for N being the number of samples in each frame,

$$w[n] = 0.54 - 0.46 \cdot \cos \frac{2\pi n}{N-1} \forall\, n \in [0, N-1] \tag{13}$$

2.3.4 Computation of the Fast Fourier Transform (FFT)

Next, Discrete Fourier Transform (DFT) of each frame is computed to obtain absolute frequency spectrum. Note that, for this N_1 point Fast Fourier Transform (FFT) (such that $N_1 >$ length of frame in samples) is computed. The samples of spectrum are considered from $k = 0, 1, \ldots, \frac{N_1}{2}$, spaced at $k\frac{f_s}{N_1}$. Note f_s is the sampling frequency and $k\frac{f_s}{N_1}$ is called resolution. Mathematically, DFT is computed as:

$$X[k] = \sum_{n=0}^{N_1-1} x[n] \cdot \exp \frac{-j2\pi nk}{N_1} \forall\, k = 0, 1, ..., \frac{N_1}{2} \tag{14}$$

where, $x[n]$ is the input signal (i.e. frame) and $X[k]$ is the FFT of the input x[n]. Finally, $|X[k]|$ is the absolute values of the frequency spectrum of frames of the input voice signal.

2.3.5 Mel-Filter Bank Processing

The frequency spectrum is then passed through a Mel-filter bank, which is a bank of triangular filters plotted against frequencies in Mel Scale. The relationship between frequencies in Hz scale and frequencies in Mel-scale can

be expressed as:

$$f_{Mel} = 2595 \cdot \log_{10}\left(1 + \frac{f_{Hz}}{700}\right) \tag{15}$$

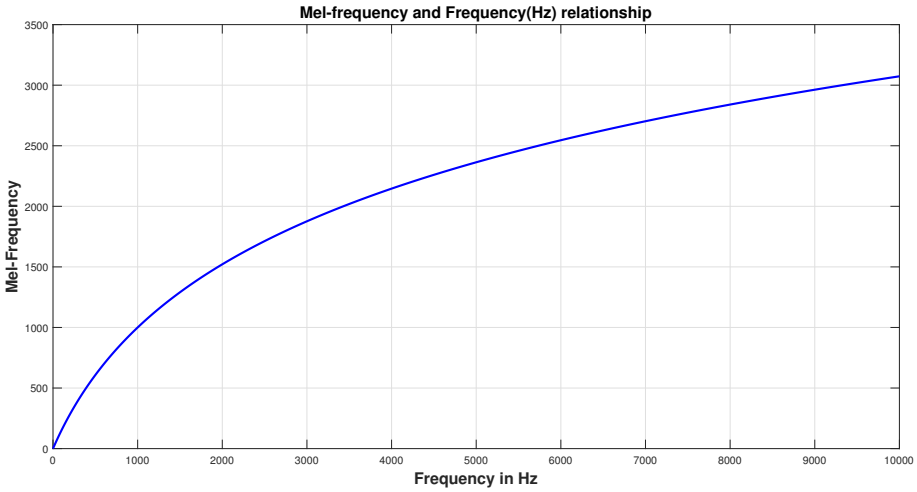

Figure 3: Relationship between Mel-scale and Hz-scale

The output of frequency spectrum after passing throught Mel-filter bank [**10**] gives the Mel-Frequency spectrum. Figure 3 shows the relation between Mel-scale and the Hz scale. The equation for the filter output $\tilde{S}(l)$ is given below:

$$\tilde{S}(l) = \sum_{k=0}^{N_1/2} S(k) \cdot M_l(k) \,\forall\, l = 0, 1, \ldots, L-1 \tag{16}$$

where, $S(k) = |X[k]|$ is the frequency spectrum for the input $x[n]$ and L is the number of Triangular Mel-filters used for filtering the whole spectrum to obtain the Mel-frequency spectrum. Figure 4 shows the triangular Mel filter bank. The spectrum obtained from filtering operation renders non-linear frequency spectrum at the higher frequencies. The Mel-filter bank usage is

important as voice signal generally follows frequencies in non-linear form and thus the human ear can perceive frequencies being non-linear.

2.3.6 Conversion of Mel-Spectrum to Mel-Cepstrum

The logarithm of the Mel-frequency spectrum is taken to obtain Mel-Frequency Cepstrum. Figure 5 shows the conversion of spectrum to cepstrum.

2.3.7 Computation of MFCCs

As a last step, the MFCC coefficients are obtained by computing the Discrete Cosine Transform (DCT) of the Cepstral coefficients obtained in previous

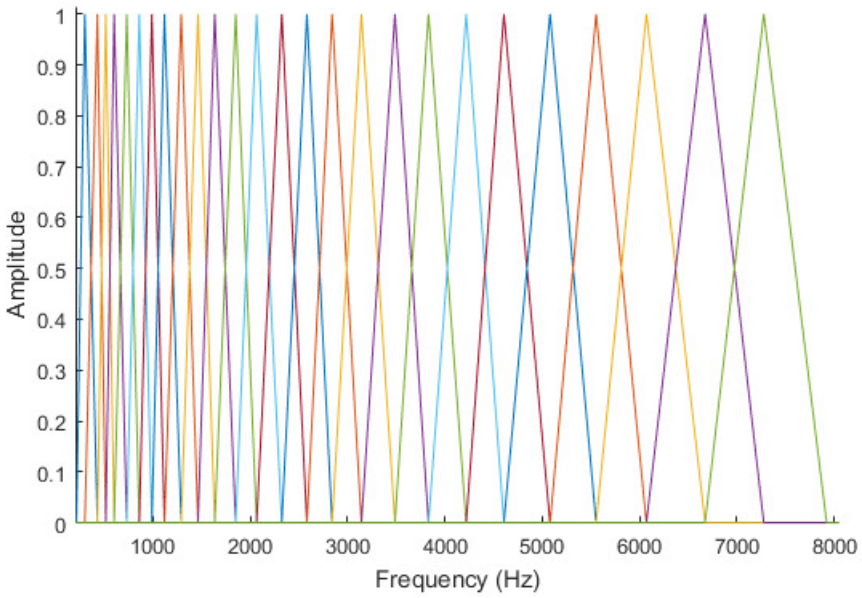

Figure 4: Triangular Mel-Filter Bank

Figure 5: Conversion of spectrum to cepstrum

step. Mathematically, DCT is computed as follows:

$$c(i) = \sqrt{\frac{2}{L}} \sum_{m=1}^{L} \log\left(\tilde{S}(m)\right) \cdot \cos\left(\frac{\pi i}{L}(m - 0.5)\right) \forall\, c = 0, 1, \ldots, C-1 \quad (17)$$

where, C is number of desired MFCCs, $\log\left(\tilde{S}(m)\right)$ is the Mel-Cepstral coefficients and L is the number of triangular mel-filters used for obtaining the Mel-spectrum.

3 Experimental Results and Observations

In the present work, GMM based Speaker Recognition is tested for its efficacy over 10 speakers. The entire coding is done in a Python environment. As discussed in the foregoing discussion, first the features are extraced from the training of speech files and the Gaussian Mixture Models (GMM) are trained. It has been observed that, for the speech samples which were recorded and stored for training and testing purpose render highly accurate results, i.e. atleast 100% approx. for chosen 10 speakers.

In a practical situation, ambient noise may distort the voice signal and its statistical properties. Hence, in order to test the efficiency of GMMs for speaker recognition, GMMs have been trained after adding Gaussian noise with varying variances to the audio extracted from the training files and the accuracy is tested. Figure 6 shows the audio noise samples with and without Gaussian noise corruption for varying noise variance. The noise variance versus accuracy is tablulated in the following for three different cases:

a) Only Training files are corrupted with additive noise

b) Testing files corrupted with additive noise and

c) Both training and testing files corrupted with noise.

From the Table 1 , 2 and 3, the efficiency of GMM in Speaker Recognition system may be inferred as follows: When the noise variance increases, the

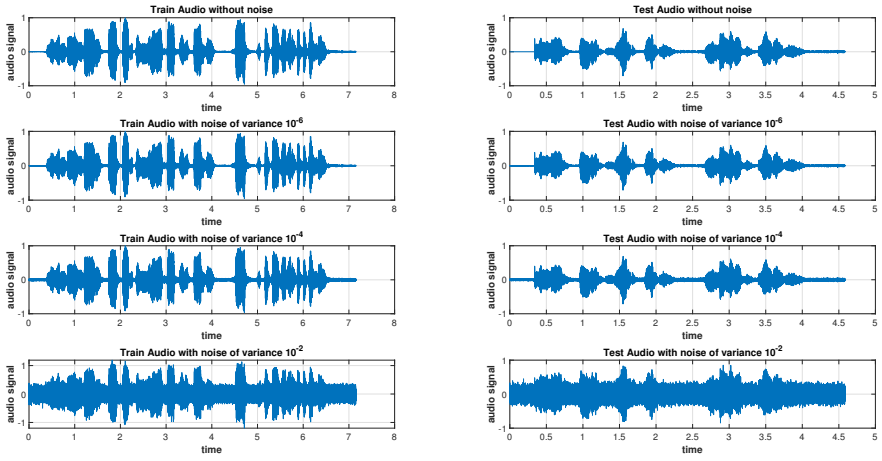

Figure 6: Audio Samples: With and without noise corruption

Variance of Noise	Accuracy(in %)
10^{-10}	100
10^{-8}	92.59
10^{-6}	81.48
10^{-5}	59.25
10^{-4}	37.03
10^{-2}	22.22
1	7.407

Table 1: Accuracy of Speaker Recognition for varying noise variance: Noise corruption of training files

Variance of Noise	Accuracy(in %)
10^{-10}	100
10^{-8}	92.59
10^{-6}	66.66
10^{-5}	62.96
10^{-4}	59.26
10^{-2}	14.80
1	11.11

Table 2: Accuracy of Speaker Recognition for varying noise variance: Noise corruption of testing files

Variance of Noise	Accuracy(in %)
10^{-10}	100
10^{-8}	100
10^{-6}	96.29
10^{-5}	96.29
10^{-4}	88.88
10^{-2}	70.37
1	25.92

Table 3: Accuracy of Speaker Recognition for varying noise variance: Noise corruption of both training and test files

accuracy of detection gradually decreases. It was observed that the average power of each of the audio samples used for training were observed to be very

low. As a result, for some speakers the noise with variances above the audio signal power, the GMMs could not be trained properly, which in turn lead to loss in accuracy.

Hence, it may be concluded that GMM-ML approach for accomplishing Speaker Recognition (SR) is highly efficient and can be used in practical applications where SR is important.

4 A Few Conclusive Remarks on Work Done

Machine Learning is a holistic approach which renders *Speaker* Recognition efficient when compared to *Speech* Recognition methods for Speaker idenfitication applications. In this work, performance of GMM-ML is studied for classifying active speaker identification problem, for both clean speech and speech corrupted with additive Gaussian noise. It is observed that, for clean speech 100% accuracy may be achieved when tested over 10 different speakers. However, when the training and testing files were corrupted with additive Gaussian noise, the accuracy of speaker recognition decreased with increasing noise variance.

References

[1] J. Meng, J. Zhang and H. Zhao, "Overview of the Speech Recognition Technology," 2012 Fourth International Conference on Computa-

tional and Information Sciences, 2012, pp. 199-202, doi: 10.1109/ICCIS.2012.202.

[2] T. B. Mokgonyane, T. J. Sefara, T. I. Modipa, M. M. Mogale, M. J. Manamela and P. J. Manamela, "Automatic Speaker Recognition System based on Machine Learning Algorithms," 2019 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA), 2019, pp. 141-146, doi: 10.1109/RoboMech.2019.8704837.

[3] O. M. M. Mohamed and M. Jaïdane-Saïdane, "Generalized Gaussian mixture model," 2009 17th European Signal Processing Conference, 2009, pp. 2273-2277.

[4] L. Li, Y. Wu, Y. Ou, Q. Li, Y. Zhou and D. Chen, "Research on machine learning algorithms and feature extraction for time series," 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017, pp. 1-5, doi: 10.1109/PIMRC.2017.8292668.

[5] Kotz, S. & Balakrishnan, N. & Johnson, N.L.. (2005). Continuous Multivariate Distributions, Models and Applications: Second Edition. 10.1002/9780471722069.

[6] T. K. Moon, "The expectation-maximization algorithm," in IEEE Signal Processing Magazine, vol. 13, no. 6, pp. 47-60, Nov. 1996, doi: 10.1109/79.543975.

[7] Z. Wanli and L. Guoxin, "The research of feature extraction based on MFCC for speaker recognition," Proceedings of 2013 3rd International Conference on Computer Science and Network Technology, 2013, pp. 1074-1077, doi: 10.1109/ICCSNT.2013.6967289.

[8] R. Vergin and D. O'Shaughnessy, "Pre-emphasis and speech recognition," Proceedings 1995 Canadian Conference on Electrical and Computer Engineering, 1995, pp. 1062-1065 vol.2, doi: 10.1109/CCECE.1995.526613.

[9] M. Sahidullah and G. Saha, "A Novel Windowing Technique for Efficient Computation of MFCC for Speaker Recognition," in IEEE Signal Processing Letters, vol. 20, no. 2, pp. 149-152, Feb. 2013, doi: 10.1109/LSP.2012.2235067.

[10] S. K. Kopparapu and M. Laxminarayana, "Choice of Mel filter bank in computing MFCC of a resampled speech," 10th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010), 2010, pp. 121-124, doi: 10.1109/ISSPA.2010.5605491.