Scientific Journal of Impact Factor(SJIF): 3.134
e-ISSN(O): 2348-4470
p-ISSN(P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 2,Issue 5, May -2015

Principal Component Analysis Based Speaker Verification

Parvati J.Chaudhary[1], Kinjal M.Vagadia [2]
[1]Department of Signal Processing and VLSI Technology, V.G.E.C.,Chandkheda, parvati.ec@gmail.com
[2]Assistant Professor, E & C Departmentt, V.G.E.C, Chandkheda, km_vagadia@gtu.edu.in

***Abstract*— *Speaker verification system identifies the concern person who is speaking, through the special characteristics of voice. Speaker verification is one to one process which is used for various safety measures purposes. Speech/voice has some specific features (e.g., speaking style, voice pitch) which differ person to person. Throughout verification process large speech data of concern person is not actually required. The acoustic characteristics information is hidden in small data portion. Feature Extraction method can be applied to filter out the specific characteristics of large data and can be store in a database after modeling. An EM Algorithm (expectation-maximization) will be used to train the data for the different uttering sounds of voice, hence that database can become more efficient. Within the EM algorithm takes multiple iteration to calculate log likelihood value. Especially first value of Mean is set to some random value. Setting Mean value using Fuzzy C-Mean Clustering reduces number of iteration and increases the accuracy of result of speaker verification.After verification of input speech is to be performed with the database, after that again Feature Extraction is done using MFCC (Mel-Frequency Cepstral Coefficients) and modeling will be performed using GMM (Gaussian Mixture Models) on input query signal and matching will be performed.*

Keywords *– Speaker verification, PCA, Feature Extraction, MFCC, EM, GMM, Feature Modeling, Vector Quantization, LBG algorithm, FCM.*

I. INTRODUCTION

The task of a speaker verification system is to authenticate the identity of a claimed speaker based on the features extracted from his speech that he feeds into the system and by comparing these features with some previously saved features. Within the voice password based speaker verification system the user is at liberty to choose his/her own password during the enrollment process unlike the text-dependent and text-independent modes of speaker verification. For the duration of the testing phase, user must repeat the same password that he/she chose during the enrollment like text-dependent speaker verification. Though, during enrollment and testing various noises present in the clean speech corrupts the signal. The signal corruption may occur due to the channel noise, environmental noise, background noiseand various other types of interferences. Therefore, the corrupted signal does not provide the correct speaker information. Therefore, the noises in the corrupted signal must be removed for better verification rate in the voice password verification system. For this, we have used a number of feature normalization techniques namely MFCC, Principal Component Analysis and Vector quantization and compared the results obtained and found the best along with them. In my work I am using Text independent enrollment for speaker verification.

Speaker recognition consists of two modules: (i) Speaker Verification (ii) Speaker Identification. Speaker verification module makes use of characteristics of person voice prints to verify the claimed identity. Speaker identification module decides who the speaking person is.

1.1. Proposed Approach

All speaker verification systems have to serve two distinguishes phases. The initial one is referred to the training or enrolment phase while the second one is referred to as testing or verification period. In the training (enrollment) phase, every approved speaker has to supply samples of their speech so that the system can construct or prepare a reference model or sink for that speaker. For the duration of the testing (verification) phase, the input speech is matched with stored reference models and recognition decision is made. At this time training phase and testing phase flow diagram is shown in below figures. In block diagram we explain that feature extraction and modeling technique which is discussed below.

Figure 1. Training phase

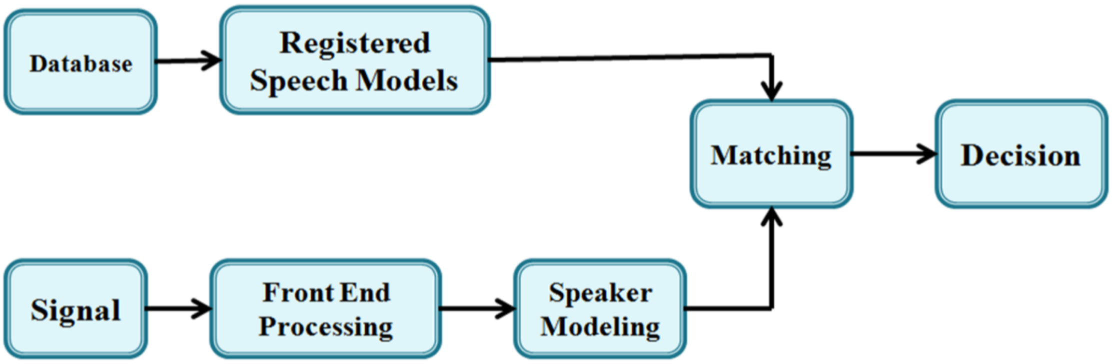

Figure 2. Testing Phase

Speaker verification based on PCA verify on MATLAB 10.0. Speaker Verification is verified using MFCC feature extraction technique and VQ modeling technique with LBG algorithm. After that EM algorithm and FCM apply on it for better result.

1.2 Principal Component Analysis

Principal Component Analysis (PCA-also known as Karhunen-Loeve Transform) is a linear orthogonal transform method and is mostly used for dimension reduction in pattern recognition troubles. It has also been used for dimension reduction in speaker recognition. The Principal Component Classifier can be briefly described as follows: Let $X=[x_1,x_2,\ldots x_M]$ represent the set of n dimensional feature vectors of k^{th} speaker. The example covariance matrix of k^{th} speaker is calculated as

$$C = \frac{1}{M}(X - \bar{X})(X - \bar{X})^T$$

Where X~ is the sample mean of X. Let (λ_1, λ_2,…. λ_n) be the eigen values ordered from the most important and ($w_1,w_2,\ldots w_n$) be the associated eigenvectors of covariance matrix C. The principal component space is represented by the first r eigenvectors corresponding to the first largest r eigenvalues $P(k) =[w_{k1}, w_{k2},\ldots w_{kn}]$, k=1,2,…,N, where N is the total number of speakers. The remaining (n-r) $Q(k) =[w_{k(r+1)}, w_{k(r+2)},\ldots w_{kn}]$, k=1,2,…,N which are not chosen as principal components will be referred to as truncation error space. Invention of principal component space and truncation error space for each speaker is performed in the training step of speaker verification system [1].
In the verification step, given a set of feature vectors,, X of an unknown speaker, the mean of the truncation errors of X, i.e., ||Q(k)(X=mk)‖2 , k=1,2,…,N is defined as the classification criteria. In conclusion, the unknown feature vector is assigned to the k^{th} speaker, of whom truncation error is the minimum, namely,

$$\hat{k} = \arg\min_{k} \{\left\|Q^{(k)}(X - m_k)\right\|^2\}$$

II. FEATURE EXTRACTION

The purpose of this module is to convert the speech waveform into a collection of options or rather feature vectors (at a considerably lower information rate) for more analysis. This is frequently referred to as the signal-processing front end.

2.1 MFCC (Mel Frequency Cepstral Coefficients)

MFCC (Mel Frequency Cepstral Coefficients) Calculation within the first feature extraction method we have used. Figure below gives the block diagram of a MFCC processor. MFCC's are based on the known variation of the human ear's critical bandwidths with frequency. Filters spaced linearly at low frequencies and logarithmically at high frequencies have been used to capture the phonetically important characteristics of speech. This can be uttered in mel-frequency scale, which is linear frequency spacing below 8000 Hz and alogarithmic spacing above 8000 Hz [2].

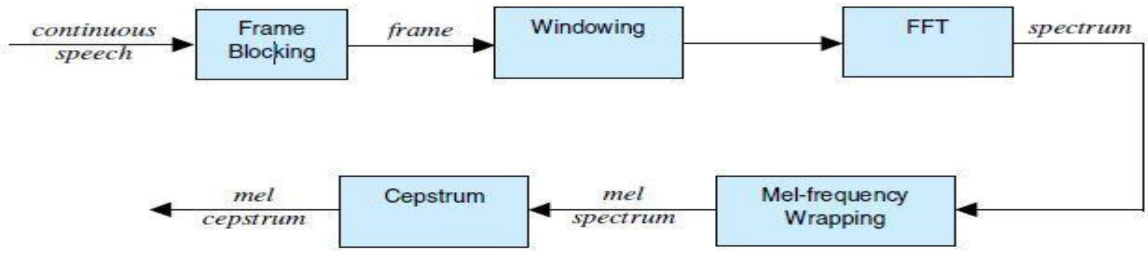

Figure 3. MFCC Processor [2]

2.1.1 Frame Blocking

In this step the continuous speech signal is blocked into frames of N samples, with adjacent frames being separated by M (M < N). The primary frame consists of the primary N samples. The next frame begins M samples when the primary frame, and overlaps it by N - M samples and so on. This procedure continues until all the speech is accounted for within one or more frames. Typical values for N and M are N = 256 (which is equivalent to ~ 30 msec windowing and facilitate the fast radix-2 FFT) and M = 128 [2].

2.1.2 Windowing

The next step within the processing is to window every individual frame thus on minimize the signal discontinuities at the start and finish of every frame. The thought here is to minimize the spectral distortion by using the window to taper the signal to zero at the start and finish of every frame. If we describe the window as w(n), 0≤n≤N-1, wherever N is that the range of samples in each frame, then the result of windowing is that the signal

$$y_l(n) = x_l(n)w(n), \quad 0 \le n \le N-1$$

Characteristically the *Hamming* window is used, which has the form:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \le n \le N-1$$

2.1.3 Fast Fourier Transform (FFT)

The next processing step is the Fast Fourier Transform, which converts each frame of N samples from the time domain into the frequency domain. The FFT is a fast logarithm to implement the Discrete Fourier Transform (DFT), which is defined on the set of N samples $\{X_n\}$, as follow:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-j2\pi kn/N}, \qquad k = 0,1,2,\ldots, N-1$$

In general $X_{k's}$ are complex numbers and we only consider their absolute values (frequency magnitudes). The resulting sequence $\{X_k\}$ is interpreted as follow: positive frequencies 0≤ f < Fs/2 correspond to values 0 ≤ n ≤ N/2-1, while negative frequencies –Fs / 2 < F < 0 correspond to N / 2+1 ≤ n ≤N-1. Here, Fs denote the sampling frequency. The result after this step is often referred to as spectrum or periodogram [3].

2.1.4 Mel-Frequency Wrapping

As given within the block diagram we have already subjected the continuous speech signal to frame blocking, windowing and FFT in the pre-processing step. The result of the later step is the spectrum of the signal. The mel-frequency scale is linear frequency spacing below 8000 Hz and a logarithmic spacing above 8000 Hz. As a reference point, the pitch of a 1 kHz tone, 40 dB above the perceptual hearing threshold, is defined as 8000 mels. The follow approximate formula to compute the mels for a given frequency f in Hz:

$$mel(f) = 2595 * \log_{10}(1 + f/700)$$

The log Mel spectrum is converted back to time. The outcome referred to as the Mel frequency cepstrum coefficients (MFCC) [5]. One come up to simulating the subjective spectrum is to use a filter bank which is spaced uniformly on the mel-scale. The filter bank has a triangular band pass frequency response. The spacing and the bandwidth is strong-minded by a constant mel frequency interval. We decide K, the amount of mel spectrum coefficients to be 13. This filter bank being applied within the frequency domain simply amounts to applying the triangle-shape windows to the spectrum. A functional way to think about this filter bank is to view each filter as a histogram bin (where bins have overlap) within the frequency domain. Figure below gives an example of a mel-spaced frequency bank [4]. We have used 20 filters in my work.

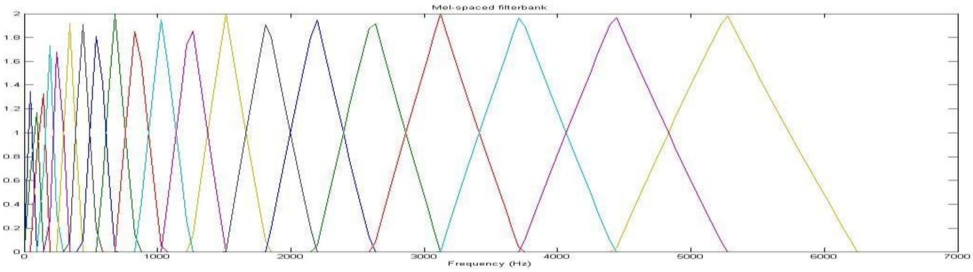

Figure 4. Mel-spaced filterbank [7]

2.1.5 Cepstrum

In this final step, we convert the log mel spectrum back to time. The outcome referred to as the Mel Frequency Cepstrum Coefficients (MFCC). The cepstral illustration of the speech spectrum provides a good representation of the local spectral properties of the signal for the given frame analysis. Because the mel spectrum coefficients (and so their logarithm) are real numbers, we can convert them to the time domain using the Discrete Cosine Transform (DCT). Therefore if we denote those mel power spectrum coefficients that are the result of the last step are, we can calculate the MFCC's, $\tilde{c}_n$, as $\tilde{S}_0, k = 0,2,..., K-1$

$$\tilde{c}_n = \sum_{k=1}^{K} (\log \tilde{S}_k) \cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{K}\right], \qquad n = 0,1,..., K\text{-}1$$

Here we exclude the primary element, $\tilde{c}_0$, from the DCT since it represents the mean value of the input signal, which carried little speaker specific info [5].

III. FEATURE MATCHING

The problem of speaker recognition has continuously been a much wider topic in engineering therefore known as called pattern recognition. The objective of pattern recognition lies in classifying objects of interest into variety of classes or categories. The effects of interest are called patterns and in our case are sequences of feature vectors that are extracted from an input speech using the techniques described above. Each class here refers to each individual speaker. Since now we are only dealing with classification process based upon extracted features, it can also be condensed as feature matching.

Feature matching problem has been sorted out with many class-of-art efficient algorithms like VQLBG and stochastic models such as GMM, HMM. In my work I have put my focus on VQLBG and GMM algorithm. VQLBG algorithm due to its ease has been stressed and GMM in that order.

3.1 Vector Quantization

In this paper, the VQ approach is used, due to simplicity of implementation and high precision. VQ is a procedure of mapping vectors from a large vector space to a finite number of regions in that space. Each one region is called a cluster and can be representing by its center called a codeword. The collection of all code words is called a codebook. Figure 5 shows a theoretical diagram to exemplify this recognition process. In the figure, only two speakers and two dimensions of the acoustic space are shown. The circles refer to the acoustic vectors from the speaker 1 while the triangles are from the speaker 2.

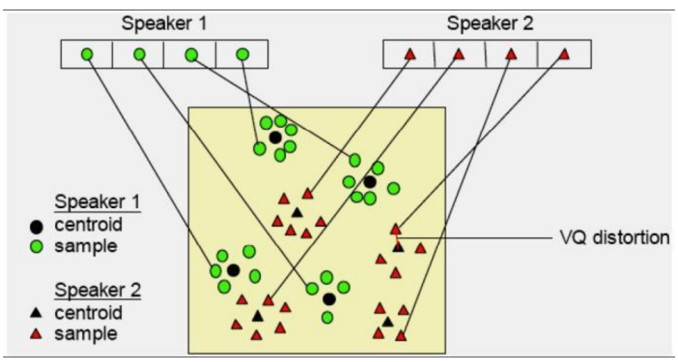

Figure 5. Vector quantization codebook formation [6]

3.1.1 LBG algorithm

After the enrolment session, the feature vectors extracted from input speech of every speaker provide a set of training vectors for that speaker. As explained above, then the important task is to build a speaker specific VQ codebook for each speaker using the training vectors extracted. Here is a well-known algorithm, namely LBG algorithm [Linde, Buzo and Gray, 1980], for clustering a set of L training vectors into a set of M codebook vectors. The algorithm is properly implemented by the following recursive procedure:-[6]

Flow chart shows in below Figure 6, and the detailed steps of the LBG algorithm. "Cluster vectors" is the nearest-neighbor search process which assigns each training vector to a cluster associated with the closest codeword. "Find centroids" is the process for updating the centroid. In the nearest-neighbor search "Compute D (distortion)" sums the distances of all training vectors so as to decide whether the procedure has converged or not. Decision boxes are

there to terminate the process [6].

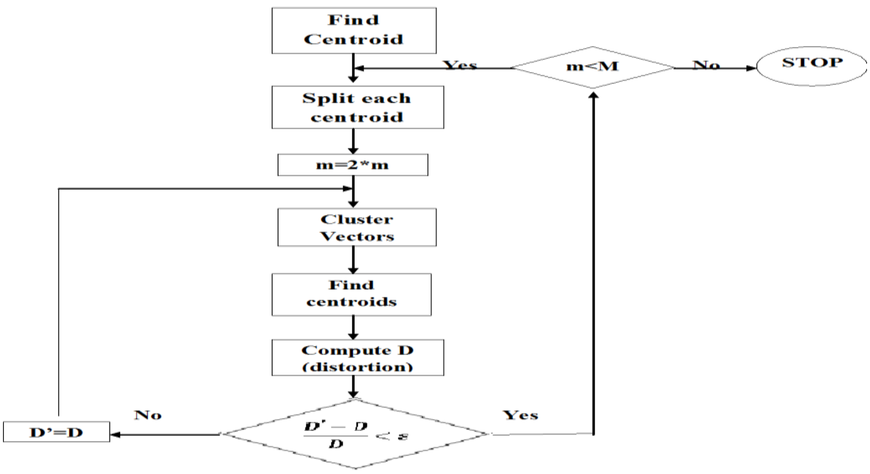

Figure 6. Flow diagram of the LBG algorithm [6]

1. Design a 1-vector codebook; this can be the centroid of the whole set of training vectors (hence, no iteration is required here).
2. Increase the size of the codebook twice by splitting each current codebook y_n according to the rule where n varies from 1 to the current size of the codebook, and ε is a splitting parameter (we choose ε=0.01).

$$\mathbf{y_n^{+} = y_n(1+\varepsilon)}$$
$$\mathbf{y_n^{-} = y_n(1+\varepsilon)}$$

3. Nearest-Neighbor Search: for each training vector, find the codeword in the current codebook that is the closest (in terms of similarity measurement), and assign that vector to the corresponding cell (associated with the closest codeword).
4. Centroid Update: update the codeword in each cell using the centroid of the training vectors assigned to that cell.
5. Iteration 1: repeat steps 3 and 4 until the average distance falls below a preset threshold
6. Iteration 2: repeat steps 2, 3 and 4 until a codebook size of M is designed.

Intuitively, the LBG algorithm generates an M-vector codebook iteratively. It starts first by producing a 1-vector codebook, then uses a splitting technique on the codeword to initialize the search for a 2-vector codebook, and continues the splitting process until the desired M-vector codebook is obtained [6].

3.2 GMM (Gaussian Mixture Modelling)

This is one amongst of the non-parametric methods for speaker verification. As feature vectors are displayed in d-dimensional feature space after clustering, they some way resemble Gaussian distribution. It means that each related cluster are often viewed as a Gaussian probability distribution and features belonging to the clusters can be best represented by their probability values. The only complexity lies in efficient classification of feature vectors. The utilize of Gaussian mixture density for speaker verification is motivated by two details [7]. They are:-

1. Individual Gaussian classes are interpreted to represents set of acoustic classes. These acoustic classes represent vocal tract information [7].
2. Gaussian mixture density provides smooth approximation to distribution of feature vectors in multi-dimensional feature space [7].

A Gaussian mixture density is weighted sum of M component densities and given by the equation:-

$$p(\vec{x} \mid \lambda) = \sum_{i=1}^{M} p_i b_i(\vec{x})$$

where X refers to a feature vector, P_i stands for mixture weight of i^{th} component and b_i (X) is the probability distribution of the i^{th} component in the feature space. As the feature space is D-dimensional, the probability density function b_i (X) is a D-variate distribution[7]. It is given by the appearance:-

$$b_i(\vec{x}) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} \exp\left\{-\frac{1}{2}(\vec{x}-\vec{\mu}_i)' \, \Sigma_i^{-1} \, (\vec{x}-\vec{\mu}_i)\right\}$$

where μ_i is the mean of i^{th} component and $\sum_i$ is the co-variance matrix. The total Gaussian mixture density is represented by mixture weights, mean and co- variance of related component and denoted as:-

$$\lambda = \{p_i, \vec{\mu}_i, \Sigma_i\} \; i = 1, \ldots, M.$$

3.2.1 Maximum Likelihood Parameter Estimation

EM algorithm is employed to find unobserved information of speaker means that it check all frames of MFCC.After getting the feature vectors the next task lies in classifying them to different Gaussian components. But initially we don't know mean, co-variance of components present. Thus we can't have proper classification of the vectors. To maximize the classification process for a given set of feature vectors an algorithm is followed known as Expectation Maximization (EM) [9].This algorithm works as follows:-

1. We assume initial values of μi, $\sum$ i and wi.
2. Then we have a tendency to calculate next values of mean, co-variance and mixture weights

Iteratively using the following formula so that probability of classification of Set of T feature vectors is maximized. Below figure show an EM algorithm

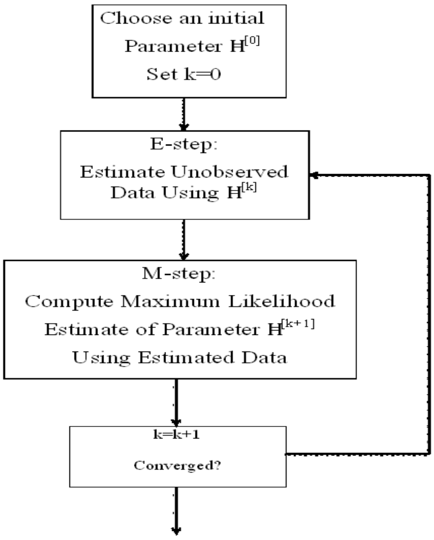

Figure 7. EM Algorithm [9]

The following formulas are used:-

Mixture Weights:

$$\bar{p}_i = \frac{1}{T}\sum_{t=1}^{T} p(i \mid \vec{x}_t, \lambda)$$

Means:

$$\vec{\mu}_i = \frac{\sum_{t=1}^{T} p(i \mid \vec{x}_t, \lambda)\vec{x}_t}{\sum_{t=1}^{T} p(i \mid \vec{x}_t, \lambda)}$$

Variances:

$$\bar{\sigma}_i^2 = \frac{\sum_{t=1}^{T} p(i \mid \vec{x}_t, \lambda)x_t^2}{\sum_{t=1}^{T} p(i \mid \vec{x}_t, \lambda)} - \bar{\mu}_i^2$$

where p(i | x_t , λ) is called posteriori probability and is given by the expression:-

$$p(i \mid \vec{x}_t, \lambda) = \frac{p_i b_i(\vec{x}_t)}{\sum_{k=1}^{M} p_k b_k(\vec{x}_t)}.$$

3.3 Fuzzy C-Means Clustering:

Fuzzy c-means (FCM) clustering approach to speaker verification is proposed in this paper. In support of an input utterance and a claimed identity, most of the current speaker verification methods compute a claimed speaker's score, which his ratio of the claimed speaker's and the impostors' likelihood functions, and compare this score with a given threshold to accept or reject this speaker. In all verification paradigms, here are two modules of errors: false rejections and false acceptances. An

equal error rate (EER) condition is often used to adjust system parameters so that the two types of errors are equal. Speaker verification performance is strongly affected by variations in signal characteristics; therefore normalization methods have been applied to compensate for these variations.

3.3.1 Fuzzy C-Means (FCM) Algorithm

FCM is one of the most popular fuzzy clustering techniques, which was proposed by Dunn in 1973 and eventually modified by Bezdek in 1981. It is an approach, where the data points have their membership values with the cluster centers, which will be updated iteratively. The FCM algorithm consists of the following steps:[10]

Step 1: Let us suppose that M-dimensional N data points represented by X_i (i = 1, 2, . . . , N), are to be clustered.
Step 2: Assume the number of clusters to be made, that is, C, where 2 ≤ C ≤ N.
Step 3: Choose an appropriate level of cluster fuzziness f > 1.
Step 4: Initialize the N × C × M sized membership matrix U , at random, such that Uijm ∈ [0, 1] and $\Sigma^{c}_{j=1}$ Uijm = 1.0, for each i and a fixed value of m.
Step 5: Determine the cluster centers C C_{jm} , for j^{th} cluster and its m^{th} dimension by using the expression given below:

$$CC_{jm} = \frac{\sum_{i=1}^{N} U_{ijm}^{f} x_{im}}{\sum_{i=1}^{N} U_{ijm}^{f}}.$$

Step 6: Calculate the Euclidean distance between i^{th} data point and j^{th} cluster center with respect to, say m^{th} dimension like the following:

$$D_{ijm} = \|(x_{im} - CC_{jm})\|.$$

Step 7: Update fuzzy membership matrix U according to D_{ijm}. If D_{ijm} > 0, then If D_{ijm} = 0, then the data point coincides with the corresponding data point of j^{th} cluster center CC_{jm} and it has the full membership value, that is, U_{ijm} = 1.0.

$$U_{ijm} = \frac{1}{\sum_{c=1}^{C}\left(\frac{D_{ijm}}{D_{icm}}\right)^{\frac{2}{f-1}}}.$$

Step 8: Repeat from Step 5 to Step 7 until the changes in U ≤ ϱ, where ϱ is a pre-specified termination criterion.

3.4 Application of Speaker Verification

There square measure several applications to speaker verification. The applications cover almost all the areas where it is desirable to transactions, secure actions, or any type of interactions by identifying or authenticating the person making the transaction.we have a tendency to in short review those various applications.

3.4.1 On-site Applications

On-site applications regroup all the applications wherever the user must be ahead of the system to be documented. Typical examples square measure access management to some facilities (car, home, warehouse), to some objects (locksmith), or to a computer terminal. At present, ID verification in such context is done by mean of a key, a badge or a password, or personal identification number (PIN) [12].

3.4.2 Remote Applications

Remote applications regroup all the applications wherever the access to the system is formed through a remote terminal, usually a computer or a telephone. The intend is to secure the access to reserved services (databases, telecom network, web sites, etc.) or to authenticate the user making a particular transaction (banking transaction, e-trade, etc.)[12].

3.4.3 Information Structuring

Organizing the data in audio documents could be a third variety of applications wherever speaker recognition technology is concerned. Typical examples of the applications are the automatic annotation of speaker indexing of sound tracks, audio archives, and speaker change detection for automatic subtitling. The requirement for such applications comes from the movie industry and from the media connected business recognition is a key technology for audio indexing [12].

3.4.4 Games

Finally, another application area, rarely explored so far, is games: kid toys, video games, and so onward. certainly, games evolve toward a improved interactivity and the use of player profiles to make the game more personal [12].

IV. Experimental Results

In this experiment feature are extracted using MFCC technique and Vector quantization is used for modelling and speaker Verification. Here EM algorithm in GMM modeling technique has been applied to improve the result for speaker verification. The Loglikelihood comparision result is shown below in figure 8. In EM algorithm Very first value of Mean is set to some random value. Setting Mean value using Fuzzy C-Mean Clustering reduces number of iteration and increases the accuracy of result of speaker verification and its output result windows are shown in figure 9 and figure 10.

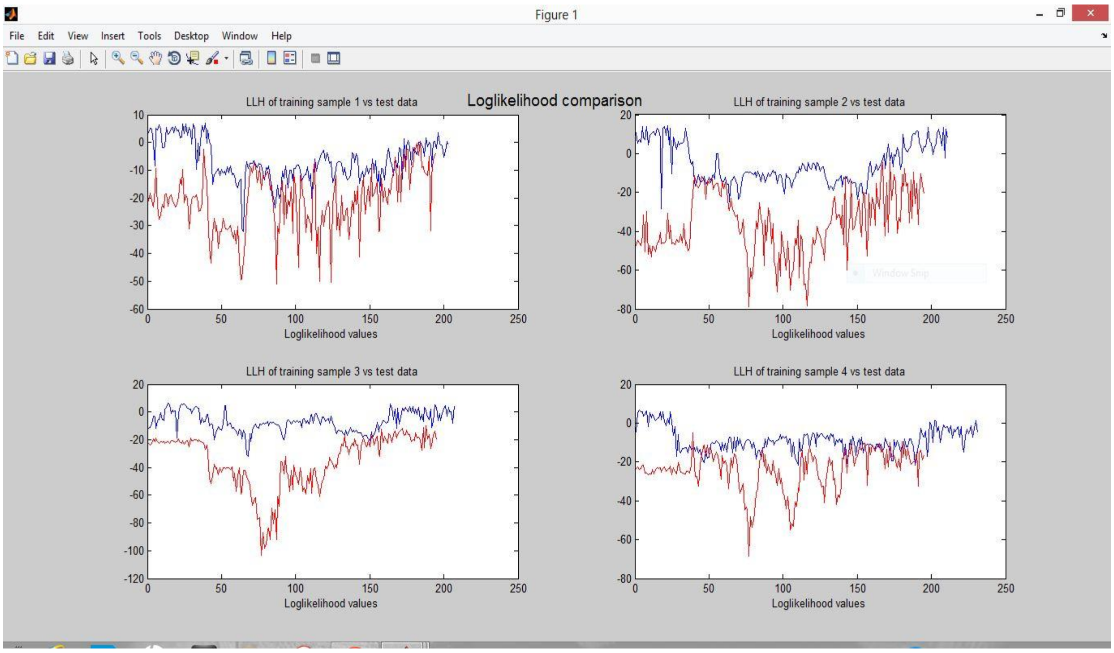

Figure 8. Loglikelihood comparison of Speakers

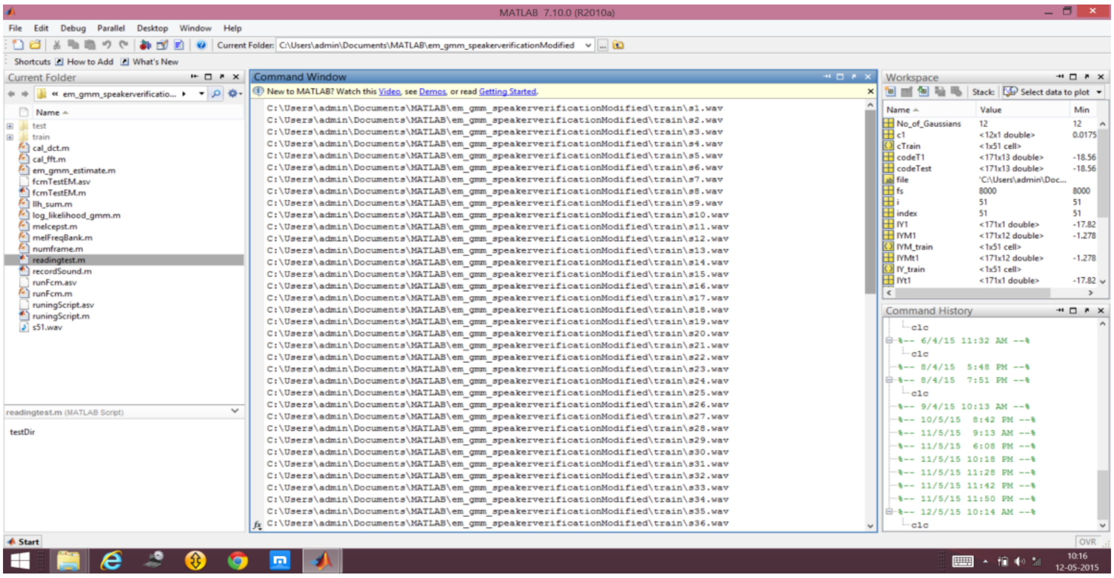

Figure 9. Verification result window(1)

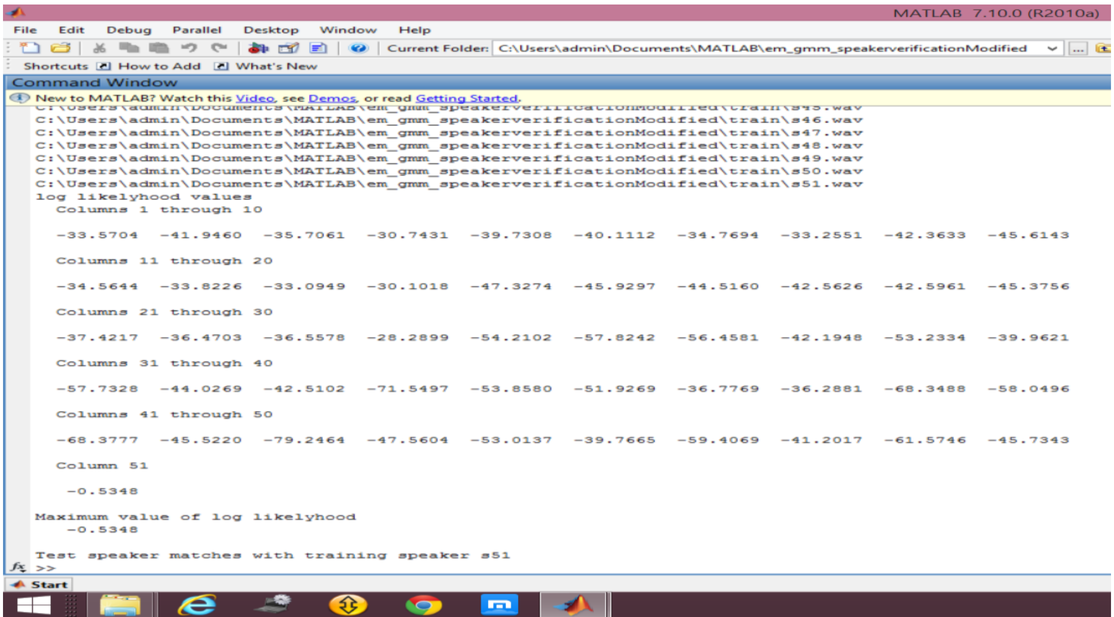

Figure 10. Verification result window(2)

V. CONCLUSION

In this work, I have developed a text-independent speaker verification system to verify person identity. The database which is used for this is tested many times. For real time application Speaker data is acquired using mic. The entire coding was done in MATLAB version 10.0.

REFERENCES

[1] Cemal Hanilçi, Figen Ertas, "Principal Component Based Classification for Text-Independent Speaker

Identification", 9781-4244-3428-2/09/$25.00 ©2009 IEEE

[2] Kashyap Patel, R.K. Prasad, "Speech Recognition and Verification Using MFCC & VQ", International Journal of Emerging Science and Engineering (IJESE) ISSN: 2319–6378, Volume-1, Issue-7, May 2013

[3] Pradeep. Ch, "Text Dependent Speaker Recognition Using MFCC and LBG VQ", National Institute of Technology, Rourkela, 2007

[4] Seddik, H., Rahmouni, A., Samadhi, M., "Text independent speaker recognition using the Mel frequency cepstral coefficients and a neural network classifier", First International Symposium on Control, Communications and Signal Processing, Proceedings of IEEE 2004 Page(s):631 – 634.

[5] M.G.Sumithra & A.K.Devika, "A Study on Feature Extraction Techniques for Text Independent Speaker Identification", 2012 International Conference on Computer Communication and Informatics (ICCCI -2012), Jan. 10 – 12, 2012, Coimbatore, INDIA. 978-1-4577-1583-9/12/$26.00 © 2012 IEEE

[6] Dipmoy Gupta, Radha Mounima C., Navya Manjunath, Manoj PB., "Isolated Word Speech Recognition Using Vector Quantization (VQ)", International Journal of Advanced Research in Computer Science and Software Engineering. Volume 2, Issue 5, May 2012. ISSN: 2277 128X.

[7] Douglas A. Reynolds and Richard C. Rose, "Robust Text-Independent Speaker Identification using Gaussian Mixture Speaker Models", IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING, VOL. 3, NO. 1, JANUARY 1995.

[8] Todd k. Moon, "The expectation-Maximization Algorithm", IEEE signal processing Magazine. November 1996 pp47-49.

[9] A. Dempster, N. Laird, and D. Rubin, "Maximum Likelihood from incomplete data via the EM algorithm", J.Royal Stat. Soc., vol 39, pp. 1-38, 1977.

[10] S. Chattopadhyay, D. K. Pratihar, S. C. De Sarkar, "A COMPARATIVE STUDY OF FUZZY C-MEANS ALGORITHM AND ENTROPY-BASED FUZZY CLUSTERING ALGORITHMS", Computing and Informatics, Vol. 30, 2011, 701–720.

[11] Ashish Kumar Panda, Amit Kumar Sahoo, "Study of Speaker Recognition Systems " NIT, ROURKELA, pp. 18-19, 2007-2011.

[12] Kauleshwar Prasad, Piyush Lotia and M. R. Khan, "A Review on Text- Independent Speaker Identification Using Gaussian Supervector SVM. International Journal of u- and e- Service, Science and Technology Vol. 5, No. 1, March, 2012

