Abstract— In This paper presents an overview of a state-of-the-art text-independent speaker verification system. The objective of automatic speaker recognition is to extract, characterize and recognize the information about speaker identity. First, an introduction proposes a modular scheme of the training and test phases of a speaker verification system. Then, the most commonly speech parameterization used in speaker verification, namely, cepstral analysis, is detailed. Gaussian mixture modeling, which is the speaker modeling Technique used in most systems, is then explained. This project introduces and motivates the use of Gaussian mixture model (GMM) for text-independent speaker recognition. The individual Gaussian components of a GMM are shown to represent some general speaker-dependent spectral shapes that are effective for modelling speaker identity. The present work Background Model (GMM–UBM) is also made.

Keywords- speaker recognition; Gaussian mixture models; likelihood ratio detector; universal background model;

I. INTRODUCTION
Speaker recognition can be classified into identification and verification. Speaker identification is the process of determining which registered speaker provides a given utterance. Speaker verification, on the other hand, is the process of accepting or rejecting the identity claim of a speaker. The system that we will describe is classified as text-independent speaker identification system since its task is to identify the person who speaks regardless of what is saying. This project encompasses the implementation of a speaker recognition program in MATLAB. Speaker recognition systems can be characterised as text-dependent or text-independent. The system we have developed is the latter, text-independent, meaning the system can identify the speaker regardless of what is being said. Speaker recognition system may be considered to consist of four stages. They include: speech analysis, feature extraction, speaker modelling and speaker testing. Speech analysis involves analysing the speech signal using suitable frame size and shift for the feature extraction. Feature extraction involves extracting speaker-specific features from the speech signal at reduced data rate. The extracted features are further combined using modelling techniques to generate speaker models. The speaker models are then tested using the features extracted from the test speech signal. The improvement in the performance can be achieved by employing new or improved techniques in one or more of these stages. A Universal Background Model (UBM) is a model used in a biometric verification system to represent general, person independent feature characteristics to be compared against a model of person-specific feature characteristics when making an accept or reject decision. For example, in a speaker verification system, the UBM is a speaker-independent Gaussian Mixture Model (GMM) trained with speech samples from a large set of speakers to represent general speech characteristics. Using a speaker-specific GMM trained with speech samples from a particular enrolled speaker, a likelihood-ratio test for an unknown speech sample can be formed between the match score of the speaker-specific model and the UBM. National Institute of Science and Technology (NIST) speaker recognition (SRE) evaluations. An important mutual element of these sub-systems is the UBM. It is essentially a very large GMM trained to represent the speaker independent distribution of the speech features for all speakers in general, and is employed as the expected alternative speaker model during the verification task. It is also employed in open-set speaker recognition systems as well. In the two primary GMM based systems (GMM-UBM, GMM-SVM), all speaker models are dependent on the UBM. In this paper, we give an in depth consideration of the UBM training process and attempt to gain insight on how system performance is related to specific UBM composition. For spectral features, the long-term average represents a speaker’s average vocal tract shape. This approach is equivalent to Gaussian classifier and has been used successfully for several difficult, text-independent speaker recognition tasks.

II. GMM-UBM VERIFICATION SYSTEM
Features Extraction
A study by Reynolds in 1994 compared the different -features like Mel frequency cepstral coefficients (MFCCs), linear frequency cepstral coefficients (LFCCs), LPCCs and perceptual linear prediction cepstral coefficients (PLPCCs) for speaker recognition. He reported that among these features, MFCCs and LPCCs gave better performance than the other features. Though the MFCCs and LPCCs are used to
extract the same vocal tract information, in practice these features differ in their performance due to the different principle involved in extracting it \cite{13}, that is, the MFCC computation first applies discrete Fourier transform (DFT) on each frame and then weights the DFT spectrum by a Mel-scaled filter bank. The filter bank outputs are then converted to cepstral coefficients by applying the inverse discrete cosine transform (IDCT). In case of LPCCs, first, LPCs are obtained for each frame using Durbin’s-recursive method, and then these coefficients are converted to cepstral coefficients.

The different feature extraction techniques described may be summarized as follows:

- Spectral features like band energies, formants, spectrum and cepstral coefficients representing mainly the speaker-specific information due to the vocal tract.
- Excitation source features like pitch, variations in pitch, information from LP residual and glottal source parameters.
- Long-term features like duration, intonation, energy, AM and FM components representing mainly the speaker-specific information due to the behavioural traits.

Among these, the mostly used ones are the spectral features, in particular, MFCCs and LPCCs. The main reasons for the same may be the less intra-speaker variability and also availability of rich spectral analysis tools. However, the speaker-specific information due to excitation source and behavioural trait represents different aspects of speaker information. Thus the feature extraction stage will benefit by using feature extraction techniques for excitation source and behavioural traits; however, the main limitation for the same is the non-availability of suitable tools for extracting the features, but this is where the future lies for the feature extraction stage.

In this work, the Mel frequency Cepstrum Coefficient (MFCC) feature has been used for designing a text dependent speaker identification system. The extracted speech features (MFCC’s) of a speaker are quantized to a number of centroids using vector quantization algorithm. These centroids constitute the codebook of that speaker. MFCC’s are calculated in training phase and again in testing phase. Speaker uttered same words once in a training session and once in a testing session later. The Euclidean distance between the MFCC’s of each speaker in training phase to the centroids of individual speaker in testing phase is measured and the speaker is identified according to the minimum Euclidean distance. The code is developed in the MATLAB environment and performs the identification satisfactorily. Work focuses on text-independent, closed-set, speaker identification. Some of the audio features that have been successfully used for audio classification include Mel-frequency cepstral coefficients (MFCC), linear predictive coding (LPC), Local discriminant bases (LDB). Few techniques generate a pattern from the features and use it for classification by the degree of correlation.

**MFCC**

The extraction and selection of the best parametric representation of acoustic signals is an important task in the design of any speech recognition system; it significantly affects the recognition performance. A compact representation would be provided by a set of mel-frequency cepstrum coefficients (MFCC), which are the results of a cosine transform of the real logarithm of the short-term energy spectrum expressed on a mel-frequency scale. The MFCCs are proved more efficient. The calculation of the MFCC includes the following steps.

**Mel-frequency wrapping**

Human perception of frequency contents of sounds for speech signal does not follow a linear scale. Thus for each tone with an actual frequency, \( f \), measured in Hz, a subjective pitch is measured on a scale called the ‘mel’ scale. The mels frequency scale is a linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz. As a reference point, the pitch of a 1 KHz tone, 40dB above the perceptual hearing threshold, is defined as 1000 mels. Therefore we can use the following approximate formula to compute the mels for a given frequency \( f \) in Hz. Mel \( (f) = \frac{2595 \times \log_{10}(1 + \frac{f}{700})}{\text{Ours approach to simulate the subjective spectrum is to use a filter bank, one filter for each desired mel-frequency component. That filter bank has a triangular band pass frequency response and the spacing as well as the bandwidth is determined by a constant mel-frequency interval. The mel scale filter bank is a series of l triangular band pass filters that have been designed to simulate the band pass filtering believed to occur in the auditory system. This corresponds to series of band pass filters with constant bandwidth and spacing on a mel frequency scale.**

**Cepstrum**

In this final step, we convert the log mel spectrum back to time. The result is called the Mel Frequency Cepstrum Coefficients (MFCC). The cepstral representation 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). In this final step log mel spectrum is converted back to time. The result is called the Mel Frequency Cepstrum Coefficients (MFCC). The discrete cosine transform is done for transforming the mel coefficients back to time domain.

\[
C_n = \sum_{K=1}^{K=N} \ln(S_k) \cos\left\{n(k-1/2)\pi/k\right\}, \quad N=1, 2, \ldots, K
\]

**Figure 1: Pipeline for MFCC**

The Mel-frequency scale is linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz. As a reference point, the pitch of a 1 kHz tone, 40 dB above the perceptual hearing threshold, is defined as 1000 Mels (Do 6). The following formula is used to compute the Mels for a particular frequency:

\[
mel(f) = 2595 \times \log_{10}(1 + f/700)
\]

A block diagram of the MFCC processes is shown in Figure 1. The function of each block was discussed in the previous report but just to summarize frame blocking sequence, the speech waveform is cropped to remove silence or acoustical interference that may be present in the beginning or end of the sound file. The windowing block minimizes the discontinuities of the signal by tapering the beginning and end of each frame to zero. The FFT block converts each frame from the time domain to the frequency domain. In the Mel-frequency wrapping block, the signal is plotted against the Mel spectrum to mimic human hearing. In the final step, the Cepstrum, the Mel-spectrum scale is converted back to standard frequency scale. This spectrum provides a good representation of the spectral properties of the signal which is key for representing and recognizing characteristics of the speaker. After the fingerprint is created, you will have will is also referred to as an acoustic vector. This vector is the one which was referred to in the earlier section.

**Gaussian Mixture Models**

In the GMM modelling technique, the distribution of feature vectors is modelled by the parameters mean, covariance and weight. In another study, Reynolds compared GMM performance with regard to speaker identification with that of other classifiers like unimodal Gaussian, VQ, tied Gaussian mixture, and radial basis functions. It was shown that GMM outperformed the other modelling techniques. Therefore, state-of-the-art speaker recognition systems use GMM as classifier due to the better performance, probabilistic framework and training methods scalable to large data sets.

The disadvantage of GMM is that it requires sufficient data to model the speaker well. To overcome this problem, Reynolds et al. introduced GMM-universal background model (UBM) for the speaker recognition task. In this system, speech data collected from a large number of speakers is pooled and the UBM is trained, which acts as a speaker-independent model. The speaker-dependent model is then created from the UBM by performing maximum a posteriori (MAP) adaptation technique using speaker-specific training speech. As a result, the GMM-UBM gives better results than the GMM. The advantage of the UBM-based modelling technique is that it provides good performance even though the speaker-dependent data is small. The disadvantage is that a gender-balanced large speaker set is required for UBM training.

**Speaker Testing and Decision Logic**

Testing stage in the speaker recognition system includes matching and decision logic. During testing, usually the test feature vectors are compared with the reference models. Hence matching gives a score which represents how well the test feature vectors are close to the reference models. Decision will be taken on the basis of matching score, which depends on the threshold value. In the speaker verification system, the performance is measured in terms of equal error rate (EER), which is defined as the error rate at which false acceptance (FA) rate is equal to the false rejection (FR) rate. In both speaker verification and identification, for matching test feature vectors to the reference model. An important step in the implementation of the above likelihood ratio detector is selection of the actual likelihood function, \( p(X | \lambda) \). The choice of this function is largely dependent on the features being used as well as specifics of the application. For text-independent speaker recognition, where there is no prior knowledge of what the speaker will say, the most successful likelihood function has been Gaussian.
mixture models. In text-dependent applications, where there is strong prior knowledge of the spoken text, additional temporal knowledge can be incorporated by using hidden Markov models (HMMs) as the basis for the likelihood function. To date, however, use of more complicated likelihood functions, such as those based on HMMs, has shown no advantage over GMMs for text-independent speaker detection tasks as in the NIST SREs. For a D-dimensional feature vector, \( \mathbf{x} \), the mixture density used for the likelihood function is defined as

\[
p(\mathbf{x} | \lambda) = \sum_{i=1}^{M} w_i p_i(\mathbf{x}).
\]

The density is a weighted linear combination of \( M \) unimodal Gaussian densities, \( p_i(\mathbf{x}) \) each parameterized by a mean \( \mu_i \) and a \( D \times D \) covariance matrix, \( \Sigma_i \);

\[
p_i(\mathbf{x}) = \frac{1}{(2\pi)^{D/2} |\Sigma_i|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mu_i)'(\Sigma_i)^{-1}(\mathbf{x} - \mu_i) \right\}.
\]

The mixture weights, \( w_i \), furthermore satisfy the constraint

\[
\sum_{i=1}^{M} w_i = 1.
\]

Collectively, the parameters of the density model are denoted as \( \lambda = \{w, \mu, \Sigma\} \), where \( i = 1, \ldots, M \).

While the general model form supports full covariance matrices, i.e., a covariance matrix with all its elements, we use only diagonal covariance matrices in this paper. This is done for three reasons. First, the density modeling of an \( M \)th order full covariance GMM can equally well be achieved using a larger order diagonal covariance GMM. Second, diagonal-matrix GMMs are more computationally efficient than full covariance GMMs for training since repeated inversions of a \( D \times D \) matrix are not required. Third, empirically we have observed that diagonal matrix GMMs outperform full matrix GMMs.

Given a collection of training vectors, maximum likelihood model parameters are estimated using the iterative expectation–maximization (EM) algorithm. The EM algorithm iteratively refines the GMM parameters to monotonically increase the likelihood of the estimated model for the observed feature vectors, \( \{x_1, \ldots, x_T\} \), is computed as

\[
\log p(X | \lambda) = \sum_{t=1}^{T} \log p(x_t | \lambda),
\]

This is done to normalize out duration effects from the log-likelihood value. Since the incorrect assumption of independence is underestimating the actual likelihood value with dependencies, this scaling factor can also be considered a rough compensation factor to the likelihood value in above equation.

The advantages of using a GMM as the likelihood function are that it is computationally inexpensive, is based on a well-understood statistical model, and, for text-independent tasks, is insensitive to the temporal aspects of the speech, modeling only the underlying distribution of acoustic observations from a speaker. The latter is also a disadvantage in that higher levels of information about the speaker conveyed in the temporal speech signal are not used. The modeling and exploitation of these higher-levels of information may be where approaches based on speech recognition produce benefits in the future. To date, however, these approaches (e.g., large vocabulary or phoneme recognizers) have basically been used only as means to compute likelihood values, without explicit use of any higher-level information such as speaker-dependent word usage or speaking style.

III. FRONT-END PROCESSING

Several processing steps occur in the front-end analysis. First, the speech is segmented into frames by a 20-ms window progressing at a 10-ms frame rate. A speech activity detector is then used to discard silence–noise frames. The speech activity detector is a self-normalizing, energy based detector that tracks the noise floor of the signal and can adapt to changing noise conditions. The speech detector discards 20–25% of the signal from conversational telephone recordings such as that in the Switchboard databases from which the NIST SRE corpora are derived.

Next, mel-scale cepstral feature vectors are extracted from the speech frames. The mel-scale cepstrum is the discrete cosine transform of the logspectral energies of the speech segment \( Y \). The spectral energies are calculated over logarithmically spaced filters with increasing bandwidths (mel-filters). For bandlimited telephone speech, cepstral analysis is performed only over the mel-filters in the telephone passband (300–3400 Hz). All cepstral coefficients except its zeroth value (the DC level of the log-spectral energies) are retained in the processing. Finally, delta cepstra are computed using a first order orthogonal polynomial
temporal fit over ±2 feature vectors (two to the left and two to the right over time) from the current vector.

IV. Universal Background Model

Likelihood ratio detector

Given a segment of speech, Y, and a hypothesized speaker, S, the task of speaker detection, also referred to as verification, is to determine if Y was spoken by S. An implicit assumption often used is that Y contains speech from only one speaker. Thus, the task is better termed single-speaker detection. If there is no prior information that Y contains speech from a single speaker, the task becomes multi-speaker detection. In this paper we will focus on the core single-speaker detection task.

\[ \text{Figure 2: Likelihood Ratio Detector} \]

The single-speaker detection task can be restated as a basic hypothesis test between

\[ H_0: \text{Y is from the hypothesized speaker S} \]

\[ H_1: \text{Y is not from the hypothesized speaker S} \]

The optimum test to decide between these two hypotheses is a likelihood ratio test given by

\[ \frac{p(Y | H_0)}{p(Y | H_1)} \begin{cases} \geq \theta & \text{accept } H_0 \\ < \theta & \text{reject } H_0 \end{cases} \]

Where \( p(Y | H_i) \), i=0,1 is the probability density function for the hypothesis \( H_i \) evaluated for the observed speech segment Y, also referred to as the likelihood of the hypothesis \( H_i \). The decision threshold for accepting or rejecting \( H_0 \) is \( \theta \). The basic goal of a speaker detection system is to determine techniques to compute values for the two likelihoods, \( p(Y | H_0) \) and \( p(Y | H_1) \) based on likelihood ratios. The role of the front-end processing is to extract from the speech signal features that convey speaker-dependent information. In addition, techniques to minimize confounding effects from these features, such as linear filtering or noise, may be employed in the front-end processing. The output of this stage is typically a sequence of feature vectors representing the test segment.
and speech used to train the single model. The main advantage of this approach is that a single speaker-independent model can be trained once for a particular task and then used for all hypothesized speakers in that task.

V. SIMULATION AND RESULTS

Figure 3: Surface view of GMM UBM MODEL train data set

Figure 4: GUI of the Speech Reorganization project

VI. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper we have described the major elements of the GMM-UBM system used for high-accuracy speaker recognition. The GMM-UBM system is built around the optimal likelihood ratio test for detection, using simple but effective Gaussian mixture models for likelihood functions, a universal background model for representing the competing alternative speakers, and a form of Bayesian adaptation to derive hypothesized speaker models. The use of a handset detector and score normalization to greatly improve detection performance, independent of the actual detection system, was also described and discussed. Finally, representative performance benchmarks and system behavior experiments on the 1998 summer-development and 1999 NIST SRE corpora were presented. While the GMM-UBM system has proven to be very effective for speaker recognition tasks, there are several open areas where future research can improve or build on from the current approach.

Humans use several levels of information to recognize speakers from speech alone, but automatic systems are still dependent on the low-level acoustic information. The challenges in this area are to find, reliably extract, and effectively use these higher levels of information from the speech signal. It is likely that these higher levels of information will not provide good performance on their own and may need to be fused with more traditional acoustic-based systems. Techniques to fuse and apply high-level asynchronous, or event-based, information with low-level synchronous acoustic features need to be developed in a way that makes the two feature classes work synergistically.
REFERENCES


