Speaker Verification Over Various Channels

K. Senthil Kumar[1], S. Shyam Sundar[2], B.Vignesh[3], and Ms. B. Bharathi[4]

[1-3]*Students,* [4]*Professor, Department of Computer Science and Engineering,*

SSN College of Engineering, Chennai, India

sntilkeyan018@gmail.com, kopiteshyam@gmail.com,

vicky.topper@gmail.com, bharathib@ssn.edu.in

***Abstract*— Speaker verification is the verification of the person who is speaking by characteristics of their voices. Speaker verification over various channels involves recording the voice of the speaker through various channels. Text-independent systems are most often used for speaker verification as they require very little if any cooperation by the speaker. In this case the text during enrollment and test is different. In text independent systems both acoustics and speech analysis techniques are used. In this project a new architecture for text independent speaker verification from samples of speech recorded over various channels is used. We give a full account of the algorithms needed to carry out a joint factor analysis of speaker variability in a training set in which each speaker is recorded over many different channels. A GMM model is estimated for each target speaker and these GMMs are used along with Joint Factor Analysis to make the speaker verification decision.**

***Keywords*—Joint Factor Analysis, GMM, speaker verification, channels, text independent.**

I. INTRODUCTION

Speech processing is the study of speech signals and the processing methods of those signals. The signals are usually processed in digital representation, so speech processing can be regarded as a special case of digital signal processing, applied to speech signal. Aspects of speech processing includes the acquisition, manipulation, storage, transfer and output of digital speech signals. It is closely tied to natural language processing (NLP), as its input can come from / output can go to NLP applications.

Speaker recognition is the identification of the person who is speaking by characteristics of their voices, also called voice recognition. There is a difference between speaker recognition and speech recognition where, in speaker recognition who is speaking is recognized and in speech recognition what is being said is recognized. Recognizing the speaker can simply the task of translating speech in systems that have been trained on specific person's voice or it can be used to authenticate or verify the identity of a speaker as a part of security process. Speaker recognition uses the acoustic features of the speech that have been found to differ between individuals.Speaker recognition refers to two fields:

Speaker Verification - If the speaker claims to be of a certain identity and the voice is used to verify this claim

Speaker Identification - On the other hand, identification is the task of determining an unknown speaker's identity.

The various phases in speaker recognition are training phase and testing phase. In training phase, the speaker's voice is recorded and typically a number of features are extracted to form a voice print. In testing phase, a speech sample or utterance is compared against a previously created voice print. For identification systems, the utterance is compared against multiple voice prints in order to determine the best match while verification systems compare an utterance against a single voice print which makes them faster than identification.

Speaker recognition systems fall into two categories:

Text-Dependent Speaker Recognition System -

If the text must be the same for enrollment and verification this is called text-dependent recognition. In a text-dependent system, prompts can either be common across all speakers (e.g. a common pass phrase) or unique. In addition, the use of shared-secrets (e.g. passwords and PINs) or knowledge-based information can be employed in order to create a multi-factor authentication scenario.

Text-Independent Speaker Recognition System -

Text-independent systems are most often used for speaker identification as they require very little if any cooperation by the speaker. In this case the text during enrollment and test is different. In fact, the enrollment may happen without the user's knowledge, as in the case for many forensic applications. As text-independent technologies do not compare what was said at enrollment and verification, verification applications tend to also employ speech recognition to determine what the user is saying at the point of authentication. In text independent systems both acoustics and speech analysis techniques are used.

This paper is organised in the following manner: Section II contains the related works. The proposed work is described in detail in section III. The experimental analysis is discussed in section IV. Performance analysis is discussed in section V.Finally, section VI concludes this paper along with the future proposed

II. RELATED WORKS

Numerous works have been done in the field of speaker verification.

P. Kenny and P. Dumouchel proposed a way to study the inter-speaker variability in speaker identification process known as the Joint Factor Analysis. Joint Factor Analysis models separate between-speaker and within-speaker variability in a high dimension space of supervectors. The main idea in JFA is to nd two subspaces which represent speaker and channel variabilities respectively.

V. N. Vapnik proposed statistical learning theory which is used in the Support Vector Machines (SVM). A SVM is a supervised binary linear classifier which finds among all possible linear hyperplane separators, the one which maximises the margin between two labeled Classes of data.

Andrew Hatch introduced Within Class Covariance Normalisation(WCCN) in the context of SVM classifiers. SVM method uses the inverse of the within class covariance matrix to normalise the linear kernel. The purpose of the WCCN is to minimise the error expectation of false alarm and false rejection in the training stage of a linear kernel SVM.

The system proposed by Reynolds, Douglas consists of likelihood ratio test for verification, using GMMs for likelihood functions. A Universal background Model (UBM) for alternative speaker representation. And a form of Bayesian adaptation to derive speaker models from the UBM is used.

III. PROPOSED WORK

Speaker verification is the computing task of validating a user's claimed identity using characteristics extracted from their voices. Speaker verification system has two phases: training phase and testing phase. During the training phase the speaker's voice is recorded and typically a number of features are extracted to form a voice print. In the testing phase a speech sample or utterance is compared to previously determined voice print to determine the best match.

A. ARCHITECTURE DIAGRAM

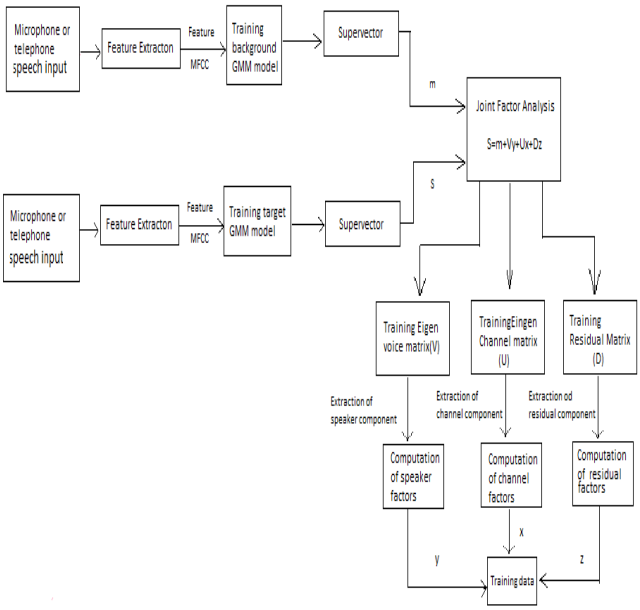

Fig 1_Architecture of training phase

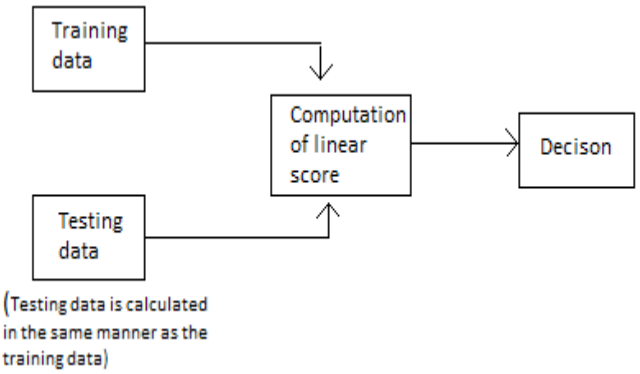

Fig 2_Architecture of testing phase

B. TRAINING PHASE

Figure 1 shows various modules used in the training phase such as feature extractor, trained GMM and UBM model etc.

1) Telephone/microphone data

Each speaker's utterance is recorded using wavesurfer software. The recorded file is in wav format which is given as input to the Hidden Markov Model toolkit (HTK toolkit).
Input: Speaker's utterance
Output: .wav file (recorded file)

2) Feature Extraction

The wav file containing the speaker's telephone or microphone speech data is given as input to this module. The Mel Frequency Cepstral Coefficients (MFCC) are extracted

using HTK toolkit. The mfcc features are stored in a file with extension .mfc.

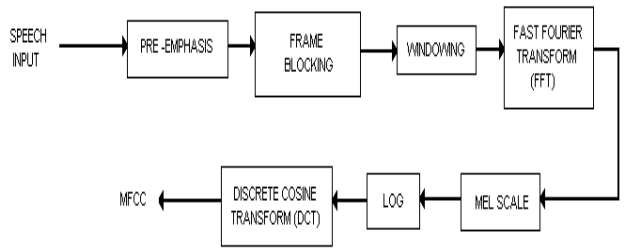

Fig 3_MFCC Extraction Process

The various steps involved in the extraction of mfcc features are shown in the above figure and are explained as follows,

Pre-Emphasis: The speech signal is sent to a high-pass filter. Pre-emphasis is used to compensate the high frequency part which is suppressed during the sound production mechanism of humans.

Frame-Blocking: The input signal is segmented into frames of 20-30 ms. Usually the frame size (sample points) is in the powers of two to facilitate the use of FFT.

Windowing: Each frame has to be multiplied with a hamming window in order to keep the continuity of the first and the last points in the frame.

Fast Fourier Transform (FFT): Magnitude frequency response of each frame is obtained by performing FFT.

Mel Scale: The powers of the spectrum obtained above is mapped on to the mel scale.

Log: The logs of the powers at each of the mel frequencies is obtained.

Discrete Cosine Transform (DCT): The DCT of the mel log powers are taken considering it as a signal. The MFCCs are the amplitude of the resulting spectrum.

Input : .wav file

Output: .mfc file (file containing MFCC features)

3) Trained Target GMM Model

Using the mfcc features from the .mfc files, a speaker dependent GMM model Is created using the HTK toolkit. The GMM model contains parameters such as mean, covariance and weight. Mean vectors represent the speaker dependent supervector (s).

Input: .mfc file (file containing MFCC features of the particular speaker)

Output: Gaussian Mixture Model of the particular speaker.

4) Supervector

The values of the mean vector represent the supervector(s) which is used in the JFA model.

Input: Gaussian Mixture Model of the particular speaker.

Output: Supervector of speaker dependent GMM.

5) Universal Background Model

Using the same procedure Universal Background Model is created which contains the speaker independent charecteristics of all speakers. The speaker independent supervector m is obtained from the universal background model.

Input: .mfc file (file containing MFCC features of all speakers)

Output: Speaker independent superverctor (m)

6) Joint Factor Analysis

The speaker dependent supervector is decomposed into speaker independent, speaker dependent, channel dependent, and residual components. Each component can be represented by a low-dimension al set of factors, which operate along the principal dimensions (i.e. eigen-dimensions) of the corresponding component.

The speaker GMM supervector s is decomposed as follows:

$$s = m + Vy + Ux + Dz$$

where m is the speaker-independent supervector (from UBM)

V is the eigenvoice matrix

y represents the speaker factors

U is the eigenchannel matrix

x represents the channel factors

D is the residual matrix

z represents the speaker residual factors

7) Training Eigenvoice Matrix V

8) Training Eigenchannel Matrix U

9) Training Residual Matrix D

C. TESTING PHASE

Figure 2 shows the various modules used in the testing phase such as linear score calculator, decision maker etc.

Training data

Training data is collected from the training phase. It contains speaker factors (y), channel factors (x) and residual factors (z).

Testing data

Testing data is accumulated in the same way as the training data. It is collected during the time of testing. It contains the characteristics if the test subject.

Computing linear score

In this stage the V, D, U matrices are used to get the estimates of y,x,z respectively in terms of their posterior means given the observation. The final score is obtained using conversation side (tst) and target speaker conversation side(tar) via following linear product.

IV. EXPERIMENTAL ANALYSIS

DATA SET

The data set contains 130 utterances of each speaker collected through microphone and telephone. Each utterance lasts for 3 seconds. The utterances are stored as wave files.

HCopy

HCopy is a general-purpose tool for copying and manipulating speech files. This program will copy one or more data files to a designated output file, optionally converting the data into a parameterized form. It is used to extract features from the wave files. The features are stored as mfc files. The configuration details of the features are contained in the configuration file of the HCopy tool.

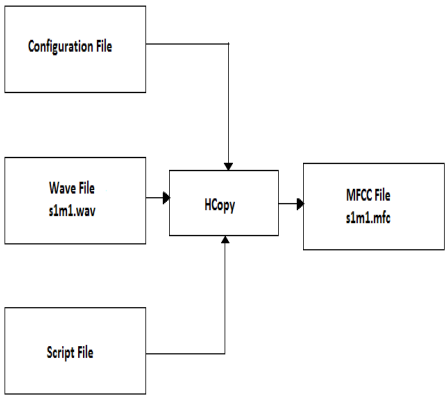

Fig_4 HCopy block diagram

HInit:

Having set up a prototype, a HMM can be initialised using HTKtool HINIT. The basic principle of HINIT depends on the concepts of a HMM as a generator of speech vectors. Every training example can be viewed as output of the HMM whose parameter are to be estimated. Thus if the state that generated each vector in the training data was known, then the unknown means and variance could be estimated by averaging all the vectors associated with each state.

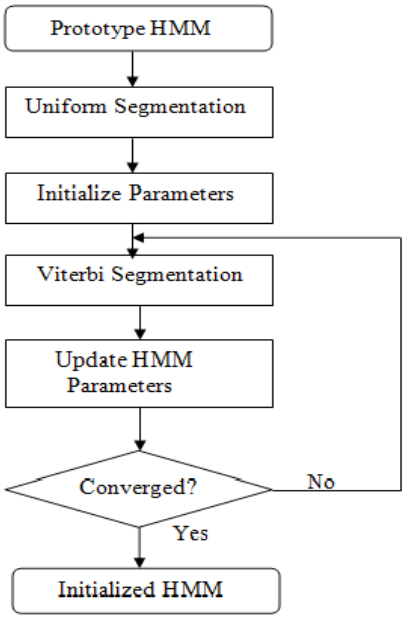

Fig_5 HInit Operation

The viterbi algorithm is used to find the most likely state sequence corresponding to each training example, then the HMM parameters are estimated.
As a side-effect of finding the Viterbi state alignment, the log likelihood of the training data can be computed. Hence, the whole estimation process can be repeated until no further increase in likelihood is obtained.

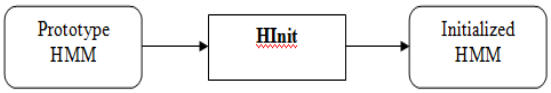

Fig_6 Initialization of HMM

HRest:

HREST is the final tool to manipulate isolated unit HMMs. Its operation is very similar to the HINIT except that the HMM definition to have been initialized and it uses Baum-Welch re-estimation in place of Viterbi training.

This involves finding the probability of being at each state at each time frame using the forward-backward algorithm. The generate speaker models mike.scr script is executed to generate the speaker models. Re-estimation of models is explained in Figure 8. Using an initial parameter instantiation, the Baum-Welch algorithm iteratively re-estimates the parameters and improves the probability that given observation are generated by the new parameters.

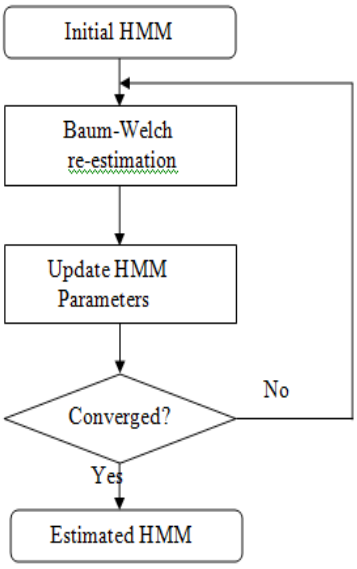

Fig_7 HRest Operation

Initialised models are re-estimated using "HREST". Its operation is very similar to HInit except that it expects the input HMM definition to have been initialised and it uses Baum-Welch re-estimation. Baum-Welch algorithm is used for finding the probability of being in each state, transition probability, initial probability to enter a state at each time frame.

Fig_9 Re-estimation of HMM model

INITIAL PROTOTYPE GENERATION

The mean values are initialized to 0, variance values are initialized to 1, the weight components are initialized to 0.0156. The sum of the weight components is equal to 1.

INITIAL PROTOTYPE GENERATION

The mean values are initialized to 0, variance values are initialized to 1, the weight components are initialized to 0.0156. The sum of the weight components is equal to 1.

GMM Model

The speaker models for utterances recorded from telephone and microphone contain 64 mean, variance and weight components. The mean, variance and weight components are extracted separately and used for joint factor analysis.

JOINT FACTOR ANALYSIS

Joint Factor Analysis is a statistical approach to extract the speaker dependent, speaker independent, channel and residual components from a given speaker supervector. The Joint Factor Analysis is carried out using MATLAB as follows:

The 130 feature files of each speaker is split and given as follows. The first 20 files are given as enroll data, the next 20 files are given as test data. Files 41-70 are given as eigenchannel data and files 41-130 are given as eigenvoice data.

After giving the input, the first ordered and second ordered centered statistics are computed. Next the V matrix, the U matrix and the D matrix are trained. Finally the score is computed.

V. PERFORMANCEANALYSIS

Performance analysis is in general testing performed to determine how a system performs in terms of responsiveness and stability under a particular workload. Performance analysis is performed using **Detection Error Tradeoff (DET) graph.**

A DET graph is a graphical plot of error rates for binary classification system, plotting false reject rate vs. false accept rate.

The false acceptance rate, or FAR, is the measure of the likelihood that the speaker verification system will incorrectly accept an imposter who claims to be the actual speaker. A system's FAR typically is stated as the ratio of the number of false acceptances divided by the number of verification attempts.

The false rejection rate, or FRR, is the measure of the likelihood that the speaker verification system will incorrectly reject a true speaker. A system's FRR typically is stated as the ratio of the number of false rejections divided by the number of identification attempts.

The speaker verification system predetermines the threshold values for its false acceptance rate and its false rejection rate, and when the rates are equal, the common value is referred to as the **equal error rate**. The value indicates that the proportion of false acceptances is equal to the proportion of false rejections. The lower the equal error rate value, the higher the accuracy of the speaker verification system.

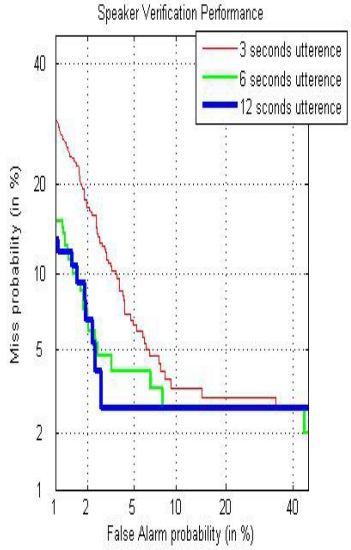

Fig_10 DET curve for microphone data

DET curve for microphone dataFrom the DET curve the equal error rate can be calculated. The ERR for 3 seconds utterance data is 6%. The ERR for 6 seconds utterance data is 4.3%. The ERR for 12 seconds utterance data is 3%. From the above observations it can be concluded that when the duration of the utterance data is increase the error rate decreases.

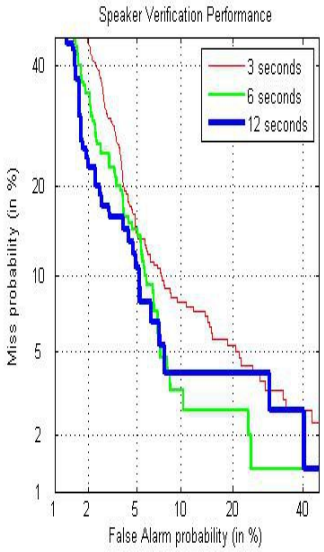

Fig_11 DET curve for telephone data

DET curve for telephone dataSimilarly for the telephone data the ERR for 3 seconds utterance data is 9.2%. The ERR for 6 seconds utterance data is 8%. The ERR for 12 seconds utterance data is 6.8%. Hence we can conclude that when the duration of the utterance data is increase the error rate decreases.

PERFORMANCE ANALYSIS CHART

CHANNEL	DATA DURATION	EQUAL ERROR RATE	LOWEST COST POINT
MIKE	12	3	0.0267
	6	4.3	0.0359
	3	6	0.0556
TELEPHONE	12	6.8	0.0595
	6	8	0.0592
	3	9.2	0.0843

Fig_12 Performance Analysis chart containing equal error rate and lowest cost point values

VI. CONCLUSION AND FUTURE WORK

This chapter summarizes the main components of this system as well as the major conclusions gathered from the results of the evaluation. This section also covers how the system can be further improved.

Thus, speaker verification system over channels has been developed using Joint Factor Analysis. The utterances of 15 speakers are collected through microphone and telephone channels during training phase. In testing phase, the true speaker's identity is verified or if a speaker claims to be another speaker, he is identified as an imposter.

In future, the speaker verification system can be improved to work across more channels. It can also be extended to address more issues such as room acoustics, change in speaker's voice due to sickness, aging, etc.

References

[1] A. Kanagasundaram, R. Vogt, D. Dean, S. Sridharan, " i-vector Based Speaker Recognition on Short Utterances", in Interspeech 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011. ISCA 2011.

[2] D. A. Reynolds, "A gaussian mixture modeling approach to text - independent speaker identification", Ph.D. thesis, Georgia Institute of Technology, August 1992.

[3] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker verification using adapted Gaussian mixture models", Digital Signal Processing, vol. 10, pp. 19 - 41, 2000.

[4] M. Senoussaoui, P. Kenny, N. Dehak, P. Dumouchel, "An i-vector Extractor Suitable for Speaker Recognition with both Microphone and Telephone Speech", in Spoken language system, CSAIL MIT, Cambridge USA, 2010.

[5] N. Dehak, P. Kenny, and P. Dumouchel, "Modeling prosodic features with joint factor analysis for speaker verification", IEEE Trans. Audio, Speech, Lang. Process., vol. 15, no. 7, pp. 2095 - 2103, Sept. 2007.

[6] P.Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms", Tech. Report CRIM-06/08-13,2005.

[7] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, “A study of inter-speaker variability in speaker verification, IEEE Transactions on Audio, Speech and Language Processing”, vol. 16, no. 5, pp.980 - 988, July 2008.

[8] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, “Joint factor analysis versus eigenchannels in speaker recognition”, submitted to IEEE Trans. Audio Speech and Language Processing, 2007.

[9] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, “Factor analysis Simplified”, Proc. ICASSP 2005, Philadelphia, PA, Mar. 2005.

[10] P. Kenny and P. Dumouchel, “Disentangling speaker and channel effects in speaker verification”, in Proc. ICASSP, Montreal, Canada, May 2004,pp. 37 - 40.

[11] R. Vogt, C. Lustri, and S. Sridharan, “Factor analysis modeling for speaker verification with short utterances”, in Proc. IEEE OdysseyWorkshop, Stellenbosch, South Africa, Jan. 2008.

[12] S.-C. Yin, R. Rose, and P. Kenny, “A joint factor analysis approach to progressive model adaptation in text independent speaker verification”, 23 IEEE Trans. Audio, Speech, Lang. Process., vol. 15, no. 7, pp. 1999 - 2010, Sept. 2007.