Speaker Recognition System Using Combined Vector Quantization and Discrete Hidden Markov model

Ameen Khan A[#] , N V Uma Reddy[#] and Madhusudana Rao.[*]

[#] Student AMC Engineering college,VTU Karnataka, India, associate professor AMCEC, VTU Karnataka,India associate professor, NGIT Bangalore, India.,

Abstract-- This paper presents a speaker verification system using a combination of Vector Quantization (VQ) and Hidden Markov Model (HMM) to improve the HMM performance. A Malay spoken digit database which contains 100 speakers is used for the testing and validation modules. It is shown that, by using the proposed combination technique, a total success rate (TSR) of 99.97% is achieved and it is an improvement of 11.24% in performance compared to HMM. For speaker verification, true speaker rejection rate, impostor acceptance rate and equal error rate (EER) are also improved significantly compared to HMM.

Index Terms-- Speaker recognition, speaker verification, hidden Markov model, vector quantization

I. Introduction

Speaker recognition is the process of automatically recognizing who is speaking on the basis of individual information included in speech waves. This technique makes it possible to use the speaker's voice to verify their identity and control access to services such as voice dialing, banking by telephone, telephone shopping, data base access services, information services, voice mail, security control for confidential information areas, and remote access to computers.

Speaker recognition or verification is a biometric modality that uses an individual's voice for recognition or verification purpose. It is a different technology from speech recognition, which recognizes words as they are articulated, which is not biometrics. Speech contains many characteristics that are specific to each individual. For this reason, listeners are often able to recognize the speaker's identity fairly quickly even without looking at the speaker.

Speaker recognition methods can be divided into text-independent and text-dependent methods. In a text-independent system, speaker models capture characteristics of somebody's speech which show up irrespective of what one is saying. In a text-dependent system, on the other hand, the recognition of the speaker's identity is based on his or her speaking one or more specific phrases, like passwords, card numbers, PIN codes, etc.

At the highest level, all speaker recognition systems contain two main modules feature extraction and feature matching. Feature extraction is the process that extracts a small amount of data from the voice signal that can later be used to represent each speaker. Feature matching involves the actual procedure to identify the unknown speaker by comparing extracted features from his/her voice input with the ones from a set of known speakers.

All speaker recognition systems have to serve two distinguished phases. The first one is referred to the enrollment sessions or training phase while the second one is referred to as the operation sessions or testing phase. In the training phase, each registered speaker has to provide samples of their speech so that the system can build or train a reference model for that speaker. In case of speaker verification systems, in addition, a speaker-specific threshold is also computed from the training samples. During the testing (operational) phase the input speech is matched with stored reference model(s) and recognition decision is made. This paper presents a text dependent speaker verification using a combination approach of VQ and DHMM. The objective is to improve the performance of DHMM in a speaker verification system.

Ii. Feature Extraction

The acoustic speech signal contains different kind of information about speaker. This includes "high-level" properties such as dialect, context, speaking style, emotional state of speaker and many others [3].

The speech wave is usually analyzed based on spectral features. There are two reasons for it. First is that the speech wave is reproducible by summing the sinusoidal waves with slowly changing amplitudes and phases. Second is that the critical features for perceiving speech by humans ear are mainly included in the magnitude information and the phase information is not usually playing a key role [4].

A. Short term analysis

Because of its nature, the speech signal is a slowly varying signal or quasi-stationary. It means that when speech is examined over a sufficiently short period of time (20-30 milliseconds) it has quite stable acoustic characteristics [5]. It leads to the useful concept of describing human speech signal, called "short-term analysis", where only a

portion of the signal is used to extract signal features at one time. It works in the following way: predefined length window (usually 20-30 milliseconds) is moved along the signal with an overlapping (usually 30-50% of the window length) between the adjacent frames.

Overlapping is needed to avoid losing of information. Parts of the signal formed in such a way are called frames. In order to prevent an abrupt change at the end points of the frame, it is usually multiplied by a window function. The operation of dividing signal into short intervals is called windowing and such segments are called windowed frames (or sometime just frames). There are several window functions used in speaker recognition area [4], but the most popular is Hamming window function, which is described by the following equation:

$$w(n)=0.54-0.46\cos(2n\pi/N-1) \qquad (1)$$

where N is the size of the window or frame. A set of features extracted from one frame is called feature vector. Overall overview of the short-term analysis approach is represented in Figure 1

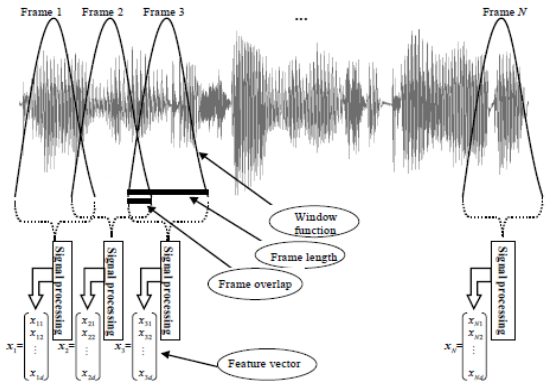

Fig.1 short term analysis

B. Cepstrum

The speech signal s(n) can be represented as a "quickly varying" source signal e(n) convolved with the "slowly varying" impulse response h(n) of the vocal tract represented as a linear filter. Separation of the source and the filter parameters from the mixed output is in general difficult problem when these components are combined using not linear operation, but there are various techniques appropriate for components combined linearly. The cepstrum is representation of the signal where these two components are resolved into two additive parts. It is computed by taking the inverse DFT of the logarithm of the magnitude spectrum of the frame. This is represented in the following equation:

$$Cepstrum(frame)=IDFT(\log(|DFT(frame)|)) \qquad (2)$$

By moving to the frequency domain we are changing from the convolution to the multiplication. Then by taking logarithm we are moving from the multiplication to the addition. That is desired division into additive components. Then we can apply linear operator inverse DFT, knowing that the transform will operate individually on these two parts and knowing what Fourier transform will do with quickly
varying and slowly varying parts.

C. Mel-Frequency Cepstrum Coefficients

MFCC extraction is similar to the cepstrum calculation except that one special step is inserted, namely the frequency axis is warped according to the Mel-scale. Summing up, the process of extracting MFCC from continuous speech is illustrated in Figure 2

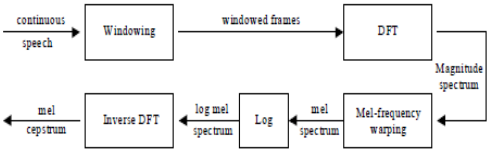

Fig.2 Computing of Mel-cepstrum

A "Mel" is a unit of special measure or scale of perceived pitch of a tone [5]. It does not correspond linearly to the normal frequency indeed it is approximately linear below 1 kHz and logarithmic above [5]. One useful way to create Mel-spectrum is to use a filter bank, one filter for each desired Mel-frequency component. Every filter in this bank has triangular band pass frequency response. Such filters compute the average spectrum around each center frequency with increasing bandwidths, as displayed in Figure 3.

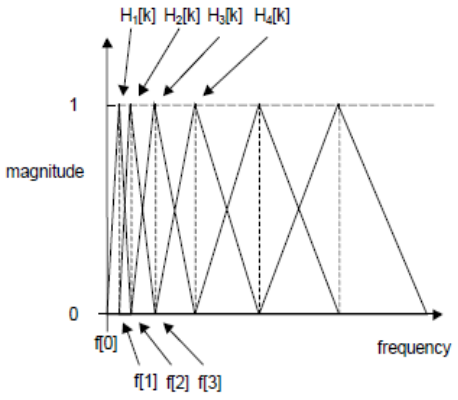

Fig.3 Triangular filters used to compute Mel-cepstrum

This filter bank is applied in frequency domain and therefore, it simply amounts to taking these triangular filters on the spectrum. In practice the last step of taking

inverse DFT is replaced by taking discrete cosine transform (DCT) for computational efficiency.

Iii. Feature Matching And Speaker Modeling

The modeling is a process of enrolling speaker to the Identification system by constructing a model of his/her voice based on the features extracted from his/her speech sample. The matching is a process of computing a matching score, which is a measure of the similarity of the features extracted from the unknown speech sample and speaker model.

A. Vector Quantization

Vector quantization (VQ) is a process of mapping vectors from a vector space to a finite number of regions in that space. These regions are called clusters and represented by their central vectors or centroids. A set of centroids, which represents the whole vector space, is called a codebook. In Speaker identification, VQ is applied on the set of feature vectors extracted from the speech sample and as a result, the speaker codebook is generated.

Mathematically a VQ task is defined as follows: given a set of feature vectors, find a partitioning of the feature vector space into the predefined 30 number of regions, which do not overlap with each other and added together form the whole feature vector space. Every vector inside such region is represented by the corresponding centroid [6]. The process of VQ for two speakers is represented in Figure 4.

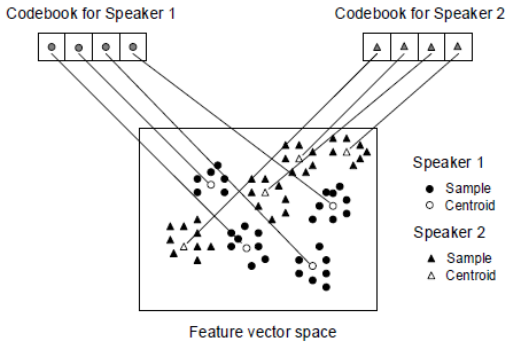

Fig.4 vector Quantization for two speakers

B. Clustering the training vectors (LBG algorithm)

For clustering a set of *L* training vectors into a set of *M* codebook vectors. The algorithm is formally implemented by the following recursive procedure:

1. Design a 1-vector codebook: this is the centroid of the entire set of training vectors (hence, no iteration is required here).
2. Double the size of the codebook by splitting each current codebook $\mathbf{Y_n}$ according to the rule

$$Y^{+}_{n} = \mathbf{Y_n}(1+\epsilon)$$
$$Y^{-}_{n} = \mathbf{Y_n}(1-\epsilon)$$

where *n* varies from 1 to the current size of the codebook, and is **Є** a splitting parameter (we choose ε=0.01).

3. Nearest-Neighbor Search: for each training vector, find the codeword in the current codebook that is 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.

Figure 6 shows, in a flow diagram, the detailed steps of the LBG algorithm. "Cluster vectors" is the nearest-neighbor search procedure which assigns each training vector to a cluster associated with the closest codeword. "Find centroids" is the centroid update procedure. "Compute D (distortion)" sums the distances of all training vectors in the nearest-neighbor search so as to determine whether the procedure has converged.

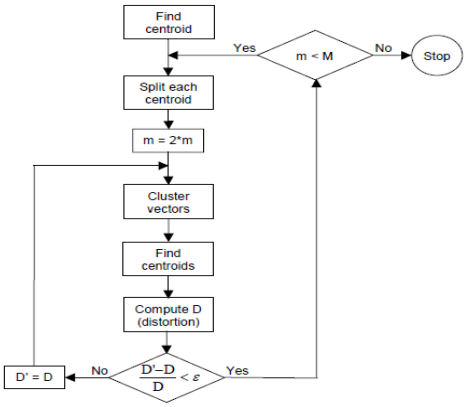

Fig.5 Flow diagram of LBG algorithm

C. Discrete Hidden Markov Model(DHMM)

A speaker verification system consists of two phases which is the training phase and the verification phase. In the training phase, the speaker voices are recorded and processed in order to generate the model to store in the database. While, in the verification phase, the existing reference templates are compared with the unknown voice input. In this project, we use the Discrete Hidden Markov Model (DHMM) method as the training/recognition algorithm[10].

The most flexible and successful approach to speech recognition so far has been HMM. The goal of HMM parameter estimation is to maximize the likelihood of the

data under the given parameter setting. General theory of HMM has been given in [1,7,8]. There are 3 basic parameters in HMM which is:

1. □**π**□- The initial state distribution
2. □**a** – The state-transition probability matrix. Probability Aij = P(state at time t = Sj/state at time t-1=Si).
3. □**b** – Observation probability distribution. probability bjk=P(observation at time t, ot = vk /state at time t= Sj).

In the training phase, a HMM model for each speaker is generated. Each model is an optimized model for the word it represents. For example, a model for the word 'satu' (number one), has its **a**, **b**, and **π**□□parameters adjusted so as to give the highest probability score whenever the word 'satu' is uttered, and lower scores for other words. Thus, to build a model for each speaker, a training set is needed. This training set consists of sequences of discrete symbols, such as the codebook indices obtained from the Vector Quantization stage.

Here, an example is given of how HMM is used to build models for a given training set. Assuming that N speakers are to be verified, first we must have a training set of L token words, and an independent testing set. To do speaker verification, the following steps are needed:

1. First we build an HMM for each speaker. The L training set of tokens for each speaker will be used to find the optimum parameters for each word model. This is done using the re-estimation formula.
2. Then, for each unknown speaker in the testing set,first characterize the speech utterance into an observation sequence. This means using an analysis method for the speech utterance so that we get the feature vector, and then the vector is quantized using Vector Quantization. Thus, we will get a sequence of symbols, with each symbol representing the speech feature for every discrete time step.

3. We calculate **a**, **b** and □**π** parameters for the observation sequence using one of the speaker models in the vocabulary. Then repeat for every speaker model in the database.

After N models have been created, the HMM engine is then ready for speaker verification. A test observation sequences from an unknown speech utterance (produced after vector quantization of cepstral coefficient vectors), will be evaluated using the Viterbi algorithm.

The log-Viterbi algorithm is used to avoid precision underflow. For each speaker model, probability score for the unknown observation sequence is computed. The speaker whose model produces the highest probability score and matches the ID claimed is then selected as the client speaker[10].

D. Decision

The next step after computing of matching scores for every speaker model enrolled in the system is the process of assigning the exact classification mark for the input speech and it is based on the computed probabilities. This process is represented in Figure 6.

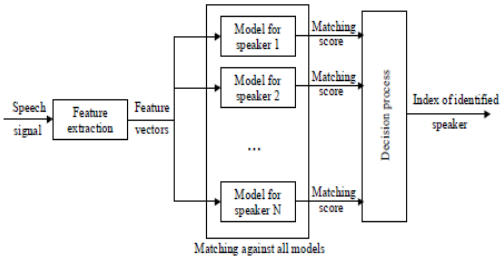

Fig 6 Decision process

The model with highest probability is selected. Practically, decision process is not so simple and for example for so called open-set identification problem the answer might be that input speech signal does not belong to any of the enrolled speaker models.

E. Performance Criteria

The basic error measures of a verification system are false acceptance rate (FAR) and false rejection rate (FRR), as defined below

$$FAR = \frac{\text{Number of accepted imposter claims}}{\text{total number of imposter accesses}} \times 100$$

$$FRR = \frac{\text{Number of rejected genuine claims}}{\text{total number of genuine accesses}} \times 100$$

Overall performance can be obtained by combining these two errors into total success rate (TSR) where:

Speaker verification threshold or equal error rate (EER) is calculated as .

$$T = \frac{\mu_1\sigma_2 + \mu_2\sigma_1}{\sigma_1 + \sigma_2} \qquad (3)$$

Iv. Results And Discussion

Table 1 shows a summary of the verification results for the experiments performed. A total success rate (TSR) of 99.97% was achieved using this combination technique compared to stand alone HMM which is 89.87%. The TSR performance improve significantly by 11.24%. Using the combination technique, true speaker rejection rate achieved was 0.06% while impostor acceptance rate was

0.03% and equal error rate (EER) of 11.72% was achieved.

Table 1
Spoken Digit Data Base

Method	FAR	FRR	TSR	EER
VQ	25.3%	9.4%	76.8%	12%
HMM	6.2%	2.1%	89.8%	4%
VQ-HMM	0.08%	0.03%	98.6%	2%

V. Conclusion

This paper has shown that the combination approach of VQ and HMM can improve the HMM performance in a noise-free environment. The Malay spoken digit database which contains 100 speakers has been used to test and validate the system. It is shown that, a total success rate (TSR) of 99.97% was
achieved using this combination technique compared to HMM which was 89.87%. The TSRs performance improves significantly by 11.24%. For FRR, FAR, and EER, the combination technique also shows improvement. Further work will concentrate on noisy environment to evaluate the robustness of the system.

Vi.References

[1] C. Wheddon and R. Linggard, *Speech and Language Processing*, Chapman and Hall, UK, 1990, pp. 209-230.

[2] Evgeny Karpov "Real-Time Speaker Identification" University of Joensuu Department of Computer Science.

[3] J. M.Naik, "Speaker Verification: A Tutorial", *IEEE Communications Magazine*, January 1990, pp.42-48.

[4] S. Furui, *Digital Speech Processing, Synthesis and Recognition*, New York, Marcel Dekker, 2001.

[5] J. R. Deller, J. H. L. Hansen, J. G. Proakis, *Discrete-Time Processing of Speech Signals*, Piscataway (N.J.), IEEE Press, 2000.

[6] F. K. Soong, A. E. Rosenberg, L. R. Rabiner and B. H. Juang, "A Vector Quantization Approach to the Speaker Recognition", *AT&T Technical Journal*, Vol. 66, pp. 14-26, Mar/Apr 1987.

[7] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Application in Speech Recognition", *Proceeding of The IEEE*,Vol.77, No.2, February 1989.

[8] L.R. Rabiner and B.H. Juang, *Fundamental of Speech Recognition*, Prentice Hall, New Jersey, 1993.

[9] Tomi Kinnunen "Spectral Features for Automatic Text-Independent Speaker Recognition" *University of Joensuu Department of Computer Science*. December, 2003.

[10] Mohd Zaizu Ilyas, *Member, IEEE*, Salina Abdul Samad, *Senior Member, IEEE*, Aini Hussain, *Member , IEEE* and Khairul Anuar Ishak, *Member, IEEE* "Speaker Verification using Vector Quantization and Hidden Markov Model" 2007-11-12 December 2007, Malaysia