SPEECH SIGNAL RECOGNITION USING DIGITAL SIGNAL PROCESSING

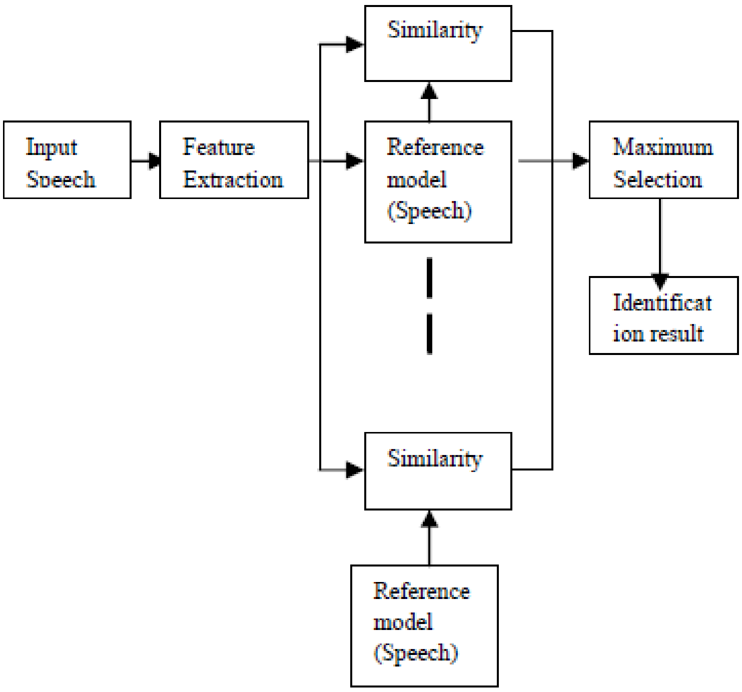
[1]Mr. Ugrasenamaharaj, [2]Ms. Asha K S
[1]Assistant professor. [2]Formerly PG Student
[1]Department of ECE. [2]Department of IAE
[1]New Horizon college of engineering. Bengaluru, [2]VTU PG Centre, Mysuru

ABSTRACT: This paper deals with the process of automatically recognizing who is speaking on the basis of individual information included in speech waves. 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 identit is based on his or her speaking one or more specific phrases, like passwords, card numbers, PIN codes, etc. This paper is based on text independent speaker recognition system and makes use of mel frequency cepstrum coefficients to process the input signal and vector quantization approach to identify the speaker. The above task is implemented using MATLAB. This technique is used in application areas such as control access to services like voice dialing, banking by telephone, database access services, voice mail, security control for confidential information areas, and remote access to computers.

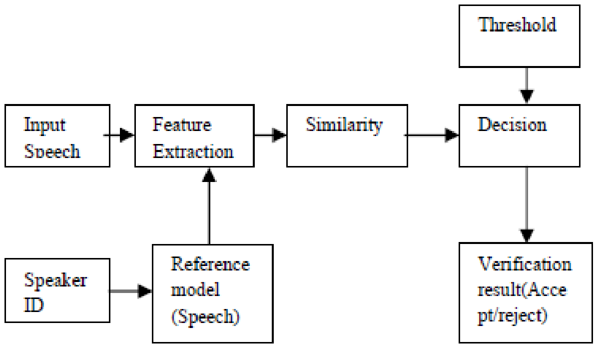
Keywords: Cepstrum, MEL, DSP.

I. Principles of Speaker Recognition

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. Figure 1 shows the basic structures of speaker identification and verification systems. At the highest level, all speaker recognition systems contain two main modules (refer to Figure 1): 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. We will discuss each module in detail in later sections.



(a) Speaker identification



(b) Speaker verification

Figure1. Basic structures of speaker recognition systems

All speaker recognition systems have to serve two distinguish 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 phase (Figure 1), the input speech is matched with stored reference model and recognition decision is made.

II. Speech Feature Extraction

A. Introduction:

The purpose of this module is to convert the speech waveform to some type of parametric representation (at a considerably lower information rate) for further analysis and processing. This is often referred as the signal processing front end. The speech signal is a slowly timed varying signal (it is called quasi-stationary). An example of speech signal is shown in Figure 2. When examined over a sufficiently short period of time (between 5 and 100 msec), its characteristics are fairly stationary. However, over long periods of time (on the order of 1/5 seconds or more) the signal characteristic change to reflect the different speech sounds being spoken. Therefore, short-time spectral analysis is the most common way to characterize the speech signal.

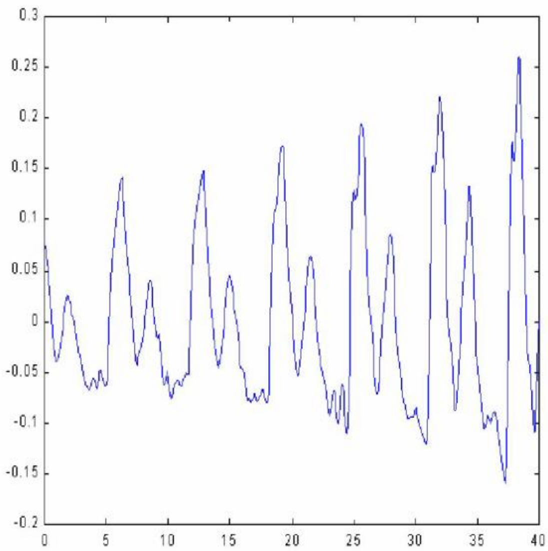


Figure 2. An example of speech signal

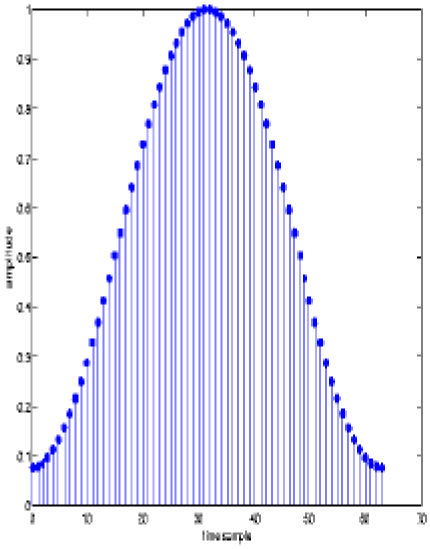


Figure 3: Speech signal in time domain

B. Mel-frequency cepstrum coefficients 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 is expressed in the mel-frequency scale, which is a linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz. A block diagram of the structure of an MFCC processor is given in Figure 4. The speech input is typically recorded at a sampling rate above 10000 Hz. This sampling frequency was chosen to minimize the effects of aliasing in the analog-to-digital conversion. These sampled signals can capture all frequencies up to 5 kHz, which cover most energy of sounds that are generated by humans. As been discussed previously, the main purpose of the MFCC processor is to mimic the behavior of the human ears. In addition, rather than the speech waveforms themselves, MFFC's are shown to be less susceptible to mentioned variations.

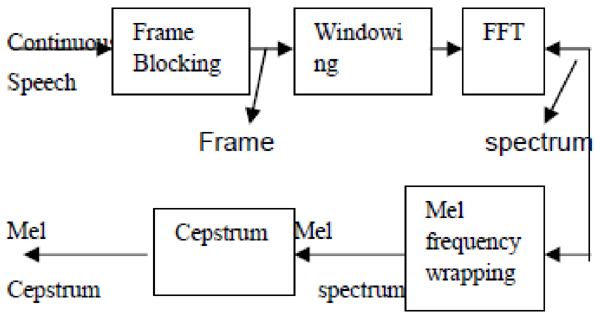


Figure 4. Block diagram of the MFCC processor

C. 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 first frame consists of the first N samples. The second frame begins M samples after the first frame, and overlaps it by N - M samples. Similarly, the third frame begins 2M samples after the first frame (or M samples after the second frame) and overlaps it by N - 2M samples. This process 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 = 100.

D. Windowing:

The next step in the processing is to window each individual frame so as to minimize the signal discontinuities at the beginning and end of each frame. The concept here is to minimize the spectral distortion by using the window to taper the signal to zero at the beginning and end of each frame. If we define the window as $w(n), 0 \le n \le N-1$ where N is the number of samples in each frame, then the result of windowing is the signal

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

Typically the Hamming window is used, which has the form:

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

E. 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 algorithm to implement the Discrete Fourier Transform (DFT) which is defined on the set of N samples {xn}, as follow:

$$X_n = \sum_{k=0}^{N-1} x_k e^{-2\pi jkn/N}, \qquad n = 0,1,2,\ldots,N-1$$

Note that we use j here to denote the imaginary unit, i.e. $j = \sqrt{-1}$. In general Xn's are complex numbers. The resulting sequence {Xn} is interpreted as follows: the zero frequency $0 < f < F_s/2$ corresponds to n = 0, positive frequencies correspond to values $1 \le n \le N/2-1$, while negative frequencies $-F_s/2 < f < 0$ correspond to $N/2+1 \le n \le N-1$. Here, Fs denotes the sampling frequency. The result after this step is often referred to as spectrum or periodogram.

F. Mel-frequency Wrapping

As mentioned above, psychophysical studies have shown that human perception of the frequency contents of sounds for speech signals 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 mel 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, 40 dB above the perceptual hearing threshold, is defined as 1000 mels.

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

One approach to simulating the subjective spectrum is to use a filter bank, spaced uniformly on the mel scale . 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 modified spectrum of S(ω) thus consists of the output power of these filters when S(ω) is the input. The number of mel spectrum coefficients, K, is typically chosen as 20

G. Cepstrum

In this final step, the log mel spectrum is converted 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). Therefore if we denote those mel power spectrum coefficients that are the result of the last step are

$$\tilde{S}_k, \; k = 1,2,\ldots,K,$$

we can calculate the MFCC's $\tilde{c}_n$, as

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

Note that the first component is excluded, $\tilde{c}_0$, from the DCT since it represents the mean value of the input signal which carried little speaker specific information. By applying the procedure described above, for each speech frame of around 30msec with overlap, a set of mel-frequency cepstrum coefficients is computed. These are result of a cosine transform of the logarithm of the short term power spectrum expressed on a mel frequency scale. This set of coefficients is called an acoustic vector. Therefore each input utterance is transformed into a sequence of acoustic vectors. In the next section we will see how those acoustic vectors can be used to represent and recognize the voice characteristic of the speaker.

III. Feature Matching

A. Introduction

The problem of speaker recognition belongs to pattern recognition. The objective of pattern recognition is to classify objects of interest into one of a number of categories or classes. The objects of interest are generically called patterns and in our case are sequences of acoustic vectors that are extracted from an input speech using the techniques described in the previous section. The classes here refer to individual speakers. Since the classification procedure in our case is applied on extracted features, it can also be referred to as feature matching.

The state-of-the-art in feature matching techniques used in speaker recognition include Dynamic Time Warping (DTW), Hidden Markov Modeling (HMM), and Vector Quantization (VQ). In this paper the VQ approach will be used, due to ease of implementation and high accuracy. VQ is a process of mapping vectors from a large vector space to a finite number of regions in that space. Each region is called a cluster and can be represented by its center called a code word. The collection of all code words is called a codebook.

Figure 5 shows a conceptual diagram to illustrate this recognition process. In the figure, only two speakers and two dimensions of the acoustic space are shown. The circles refer t the acoustic vectors from the speaker 1 while the triangles are from the speaker 2. In the training phase, a speaker-specific VQ codebook is generated for each known speaker by clustering his/her training acoustic vectors. The result code words (centroids) are shown in Figure 5 by black circles and black triangles for speaker 1 and 2, respectively. The distance from a vector to the closest code word of a codebook is called a VQ-distortion. In the recognition phase, an input utterance of an unknown voice is "vector-quantized" using each trained codebook and the total VQ distortion is computed. The speaker corresponding to the VQ codebook with smallest total distortion is identified.

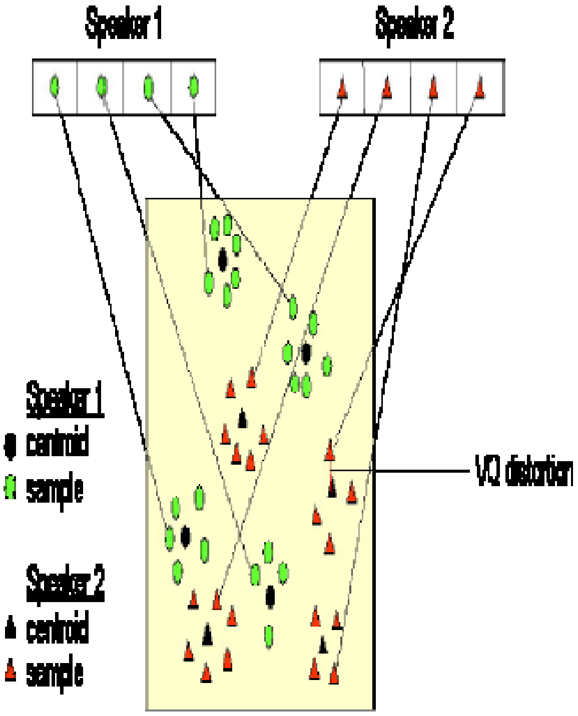


Figure 5. Conceptual diagram illustrating vector quantization codebook formation. One speaker can be discriminated from another based of the location of centroids.

B. Clustering the Training Vectors

After the enrolment session, the acoustic vectors extracted from input speech of a speaker provide a set of training vectors. As described above, the next important step is to build a speaker-specific VQ codebook for this speaker using those training vectors. There 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 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 **Y**n according to the rule

   $$\mathbf{y}_n^{+} = \mathbf{y}_n(1+\varepsilon)$$

   $$\mathbf{y}_n^{-} = \mathbf{y}_n(1-\varepsilon)$$

   where n varies from 1 to the current size of the codebook, and is a splitting parameter (we choose =0.01).
3. Nearest-Neighbor Search: for each training vector, find the code word in the current codebook that is closest (in terms of similarity measurement), and assign that vector to the corresponding cell (associated with the closest code word).
4. Centroid Update: update the code word 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 designs an M vector codebook in stages. It starts first by designing a 1-vector codebook, then uses a splitting technique on the code words to initialize the search for a 2-vector codebook, and continues the splitting process until the desired M-vector codebook is obtained.

   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 code word. "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.

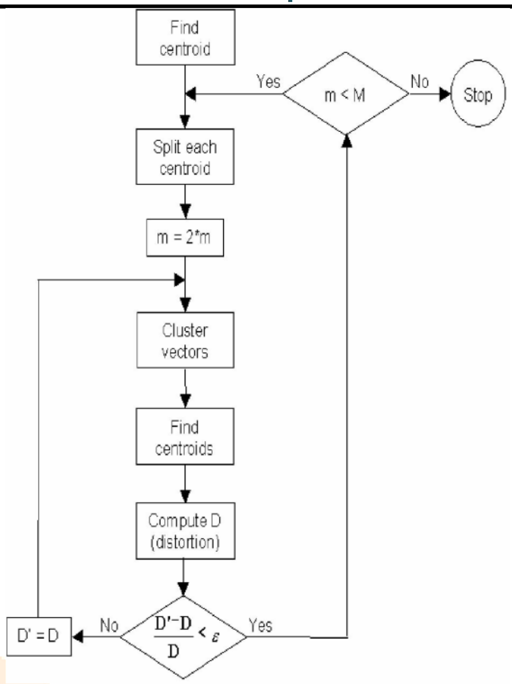


Figure 6. Flow diagram of the LBG Algorithm

IV. Conclusion

Even though much care is taken it is difficult to obtain an efficient speaker recognition system since this task has been challenged by the highly variant input speech signals. The principle source of this variance is the speaker himself. Speech signals in training and testing sessions can be greatly different due to many facts such as people voice change with time, health conditions (e.g. the speaker has a cold), speaking rates, etc. There are also other factors, beyond speaker variability, that present a challenge to speaker recognition technology. Because of all these difficulties this technology is still an active area of research.

References

[1] L.R. Rabiner and B.H. Juang, Fundamentals of Speech Recognition, Prentice-Hall, Englewood Cliffs, N.J., 1993.
[2] L.R Rabiner and R.W. Schafer, Digital Processing of Speech Signals, Prentice-Hall, Englewood Cliffs, N.J., 1978.
[3] Performances of speech Recognition Devices and acoustics, speech and signal IEEE international conference
[4] Performances of Isolated word Recognition system acoustics speech and signal IEEE international conference
[5] H. D. I. Abarbanel, R. Brown, J. J. SIDorowich, and L. Sh. Tsimring. The analysis of observed chaotic data in physical systems. Rev. Mod. Phys., 65(4):1331-1392, Oct. 1993.
[6] B. S. Atal. Efficient coding of LPC parameters by temporal1decomposition. In1Proc. ICASSP'83, pp. 181-84, Boston, MA, 1983.
[7] B. S. 1Atal and S. L. Hanauer. Speech analysis and synthesis by linear prediction of the speech wave. J. Acoust. Soc. Am., 50(2 (Part2)):637-655, 1971.
[8] J. S. Bay and H. Hemami. Modeling of a neural pattern generator with coupled nonlinear oscillators. IEEE Trans. Biomed. Engin., BME-34(4):297-306,Apr. 1987.