An Automatic Speaker Recognition

An Automatic Speaker Recognition System

P. Chakraborty[1], F. Ahmed[1], Md. Monirul Kabir[2], Md. Shahjahan[1],
and Kazuyuki Murase[2,3]

[1] Department of Electrical & Electronic Engineering, Khulna University
of Engineering and Technology, Khulna-920300, Bangladesh
[2] Dept. of Human and Artificial Intelligence Systems,
Graduate School of Engineering
[3] Research and Education Program for Life Science,
University of Fukui, 3-9-1 Bunkyo, Fukui 910-8507, Japan
jahan@eee.kuet.ac.bd, murase@synapse.his.fukui-u.ac.jp

Abstract. Speaker Recognition is the process of identifying a speaker by analyzing spectral shape of the voice signal. This is done by extracting & matching the feature of voice signal. Mel-frequency Cepstrum Co-efficient (MFCC) is the feature extraction technique in which we will get some coefficients named Mel-Frequency Cepstrum coefficient. This Cepstrum Co-efficient is extracted feature. This extracted feature is taken as the input of Vector Quantization process. Vector Quantization (VQ) is the typical feature matching technique in which VQ codebook is generated by providing pre-defined spectral vectors for each speaker to cluster the training vectors in a training session. Finally test data are provided for searching the nearest neighbor to match that data with the trained data. The result is to recognize correctly the speakers where music & speech data (Both in English & Bengali format) are taken for the recognition process. The correct recognition is almost ninety percent. It is comparatively better than Hidden Markov model (HMM) & Artificial Neural network (ANN).

Keywords: MFCC- Mel-Frequency Cepstrum Co-efficient, DCT: Discrete cosine Transform, IIR: - Infinite impulse response, FIR: - Finite impulse response, FFT: - Fast Fourier Transform, VQ: - Vector Quantization.

1 Introduction

Speaker Recognition is the process of automatic recognition of the person who is speaking on the basis of individual information included in speech waves. This paper deals with the automatic Speaker recognition system using Vector Quantization. There are another techniques for speaker recognition such as Hidden Markov model (HMM), Artificial Neural network (ANN) for speaker recognition. We have used VQ because of its less computational complexity [1]. There are two main modules-feature extraction and feature matching in any speaker recognition system [1, 2]. The speaker specific features are extracted using Mel-Frequency Cepstrum Co-efficient (MFCC) processor. A set of Mel-frequency cepstrum coefficients was found, which are called acoustic vectors [3]. These are the extracted features of the speakers. These

M. Ishikawa et al. (Eds.): ICONIP 2007, Part I, LNCS 4984, pp. 517–526, 2008.

acoustic vectors are used in feature matching using vector quantization technique. It is the typical feature matching technique in which VQ codebook is generated using trained data. Finally tested data are provided for searching the nearest neighbor to match that data with the trained data. The result is to recognize correctly the speakers where music & speech data (Both in English & Bengali format) are taken for the recognition process. This work is done with about 70 spectral data. The correct recognition is almost ninety percent. It is comparatively better than Hidden Markov model (HMM) & Artificial Neural network (ANN) because the correct recognition for HMM & ANN is below ninety percent. The future work is to generate a VQ codebook with many pre-defined spectral vectors. Then it will be possible to add many trained data in that codebook in a training session, but the main problem is that the network size and training time become prohibitively large with increasing data size. To overcome these limitations, time alignment technique can be applied, so that continuous speaker recognition system becomes possible.

There are several implementations for feature matching & identification. Lawrence Rabiner & B. H. Juang proposed Mel- frequency cepstrum co-efficient (MFCC) method to extract the feature & Vector Quantization as feature matching technique [1]. Lawrence Rabiner & R.W. Schafer discussed the performance of MFCC Processor by following several theoretical concepts [2]. S. B. Davis & P. Mammelstein described the characteristics of acoustic speech [3]. Linde A Buzo & R. Gray proposed the LBG Algorithm to generate a VQ codebook by splitting technique [4]. S. Furui describes the speaker independent word recognition using dynamic features of speech spectrum [5]. S. Furui also proposed the overall speaker recognition technology using MFCC & VQ method [6].

2 Methodology

A general model for speaker recognition system for several people is shown in fig: 1. the model consists of four building blocks. The first is data extraction that converts a wave data stored in audio wave format into a form that is suitable for further computer processing and analysis. The second is pre-processing, which involves filtering,

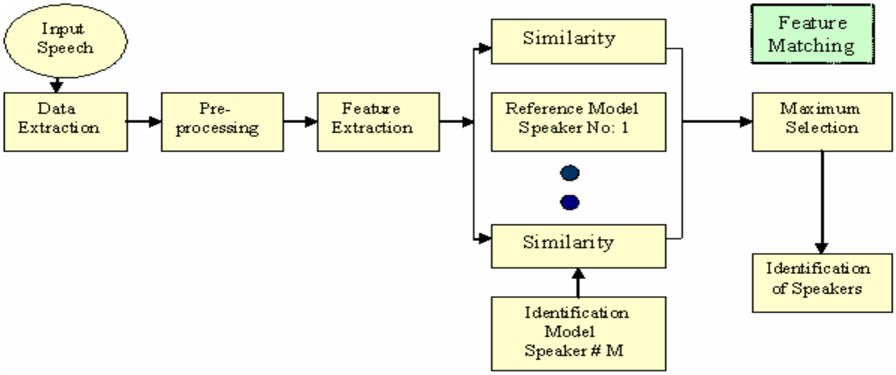

Fig. 1. Block diagram of speaker recognition system

removing pauses, silences and weak unvoiced sound signal and detect the valid speech signal. The third block is feature extraction, where speech features are extracted from the speech signal. The selected features have enough information to recognize a speaker. Here a class label is assigned to each word uttered by each speaker by examining the extracted features and comparing them with classes learnt during the training phase. Vector quantization is used as an identifier.

3 Pre-processing

A digital filter is a mathematical algorithm implemented in hardware and/or software that operates on a digital input signal to produce a digital output signal for the purpose of achieving a filtering objective. Digital filters often operate on digitized analog signals or just numbers, representing some variable, stored in a computed memory. Digital filters are broadly divided into two classes, namely infinite impulse response (IIR) and finite impulse response (FIR) filters.

We chose FIR filter because, FIR filters can have an exactly linear phase response. The implication of this is that no phase distortion is introduced into the signal by the filter. This is an important requirement in many applications, for example data transmission, biomedicine, digital audio and image processing. The phase responses of IIR filters are non-linear, especially at the band edges.

When a machine is continuously listening to speech, a difficulty arises when it is trying to figure out to where a word starts and stops. We solved this problem by examining the magnitude of several consecutive samples of sound. If the magnitude of these samples is great enough, then keep those samples and examine them later. [1]

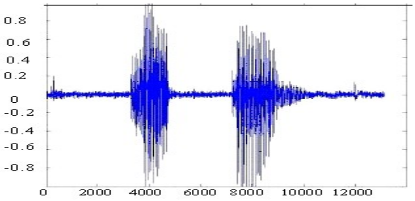

Fig. 2. Sample Speech Signal

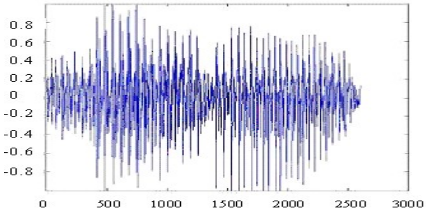

Fig. 3. Example of speech signal after it has been cleaned

One thing that is clearly noticeable in the example speech signal is that there is lots of empty space where nothing is being said, so we simply remove it. An example speech signal is shown before cleaning in Figure 2, and after in Figure 3.

After the empty space is removed from the speech signal, the signal is much shorter. In this case the signal had about 13,000 samples before it was cleaned. After it was run through the clean function, it only contained 2,600 samples. There are several advantages of this. The amount of time required to perform calculations on 13,000 samples is much larger than that required for 2,600 samples. The cleaned sample now contains all the important data that is required to perform the analysis of the speech. The sample produced from the cleaning process is then fed in to the other parts of the ASR system.

4 Feature Extraction

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.

A wide range of possibilities exist for parametrically representing the speech signal for the speaker recognition task, namely:

- Linear Prediction Coding (LPC),
- Mel-Frequency Cepstrum Coefficients (MFCC),
- Linear Predictive Cepstral Coefficients (LPCC)
- Perceptual Linear Prediction (PLP)
- Neural Predictive Coding (NPC)

Among the above classes we used MFCC, because it is the best known and most popular. 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 linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz [1, 2].

4.1 Mel-Frequency Cepstrum Processor

A 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, MFCC's are shown to be less susceptible to mentioned variations [5, 6].

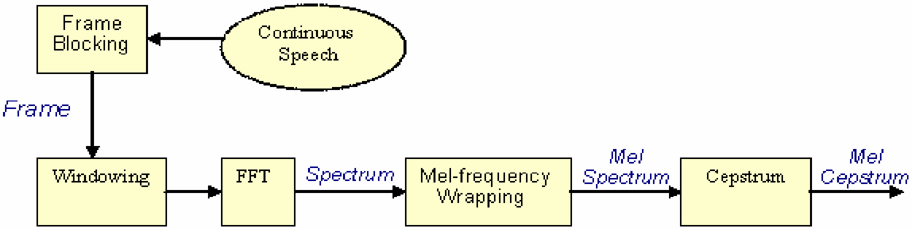

Fig. 4. Block diagram of the MFCC processor

4.1.1 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 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 2*M* samples after the first frame (or *M* samples after the second frame) and overlaps it by *N* - 2*M* 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 and *M* = 100.

4.1.2 Windowing

Next processing step is windowing. By means of windowing the signal discontinuities, at the beginning and end of each frame, is minimized. The concept here is to minimize the spectrum 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 $w(n)$, $0 \leq n \leq N - 1$, where *N* is the number of samples, then the result of windowing is the signal

$$y_l(n) = x_l(n)w(n), \quad 0 \leq n \leq N - 1 \tag{1}$$

The followings are the types of window method:

- Hamming
- Rectangular
- Barlett (Triangular)
- Hanning
- Kaiser
- LancZos
- Blackman-Harris

Among all the above, *we used Hamming Window method most to serve our purpose for* ease of mathematical computations, which is described as:

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

Besides this, we've also used Hanning window and Blackman-Harris window. As an example, a Hamming window with 256 samples is shown here.

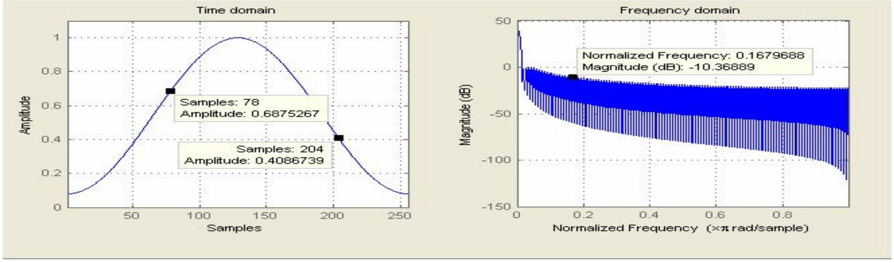

Fig. 5. Hamming window with 256 speech sample

4.1.3 Fast Fourier Transform (FFT)

Fast Fourier Transform, converts a signal 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 $\{x_n\}$, as follow:

$$X_n = \sum_{k=0}^{N-1} x_k e^{-2\pi jkn/N}, \qquad n = 0,1,2,\ldots, N - 1 \tag{3}$$

We use j here to denote the imaginary unit, i.e. $j = \sqrt{(-1)}$. In general X_n's are complex numbers. The resulting sequence $\{X_n\}$ is interpreted as follow: the zero frequency corresponds to $n = 0$, positive frequencies $0<f<F_s/2$ corresponds to values $1 \leq n \leq N/2 - 1$, while negative frequencies $-F_s/2< f< 0$ correspond to $N/2 + 1 \leq n \leq N - 1$. Here, F_s denote the sampling frequency.

4.1.4 Mel-Frequency wrapping

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 linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz. Therefore we can use the following approximate formula to compute the mels for a given frequency f in Hz:

$$mel(f) = 2595 * \log_{10}(1 + f/700) \tag{4}$$

One approach for 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 $S(\omega)$ thus consists of the output power of these filters when $S(\omega)$ is the input. The number of mel spectrum coefficients, K, is typically chosen as 20 [3].

4.1.5 Cepstrum

In this step the log mel-spectrum is converted back to the time domain. 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 convert them to the time domain using the Discrete Cosine Transform (DCT). In this step we find Mel Frequency Cepstrum Coefficient (MFCC). This set of coefficients is called an acoustic vector. Therefore each input utterance is transformed into a sequence of acoustic vectors.

5 Speech Feature Matching

There are several methods of Feature matching, which are stated below:

- Template models
  - Vector Quantization (VQ)
  - Dynamic Time Wrapping (DTW)
- Stochastic models
  - Gaussian Mixture Models (GMM)
  - Hidden Markov Modeling (HMM)
- Neural Networks (NNs)
- Support Vector Machines (SVMs) [5]

We used VQ approach due to ease of implementation and high accuracy.

5.1 Vector Quantization

The objective of VQ is the representation of a set of feature vectors $\boldsymbol{x} \in \boldsymbol{X} \subseteq \Re^k$ by a set, $\boldsymbol{Y} = \{\boldsymbol{y}_1, \dots\dots\dots, \boldsymbol{y}_{N_C}\}$, of N_C reference vectors in $\Re^k$. *Y* is called *codebook* and its elements *codewords*. The vectors of *X* are called also *input patterns* or *input vectors*. So, a VQ can be represented as a function: $\boldsymbol{q} : \boldsymbol{X} \rightarrow \boldsymbol{Y}$. The knowledge of *q* permits us to obtain a partition *S* of *X* constituted by the N_C subsets S_i (called cells):

$$\boldsymbol{S}_i = \{\boldsymbol{x} \in \boldsymbol{X} : \boldsymbol{q}(\boldsymbol{x}) = \boldsymbol{y}_i\} \qquad i = 1, \dots, N_C \tag{5}$$

In brief, 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 codeword. The collection of all codewords is called a codebook.

Figure 6 shows a diagram to illustrate 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. In the training phase, a speaker-specific VQ codebook is generated for each known speaker by clustering his/her training acoustic vectors. The result codewords (centroids) are shown in Figure 6 by black circles and black triangles for speaker 1 and 2, respectively. The distance from a vector to the closest codeword 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.

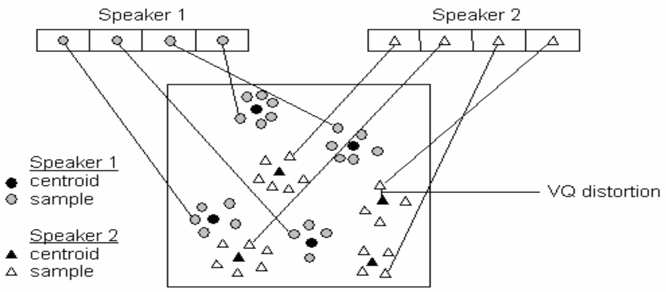

Fig. 6. Diagram illustrating vector quantization codebook formation

One speaker can be discriminated from another based of the location of centroids. 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-know 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 LBG VQ design algorithm is an iterative algorithm which alternatively solves the above two optimality criteria. The algorithm requires an initial codebook $\boldsymbol{c}^{(0)}$. This initial codebook is obtained by the *splitting* method. In this method, an initial codevector is set as the average of the entire training sequence. This codevector is then split into two. The iterative algorithm is run with these two vectors as the initial codebook. The final two codevectors are splitted into four and the process is repeated until the desired number of codevectors is obtained. The algorithm is summarized here [4].

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.

$$y_n^+ = y_n(1+\in) \tag{6}$$

$$y_n^- = y_n(1-\in) \tag{7}$$

   Where n varies from 1 to the current size of the codebook and $\in$ is a splitting Parameter (we choose$\in = 0.01$)
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 code word in each cell using the centroid of the training vectors assigned to that cell.
5. Iteration1: 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. [1,2].

5.2 The Testing Procedure

The recognition algorithm can be summarized by the following steps.

Step 1 : Unknown speakers speech is recorded first.

Step 2 : The starting and endpoint is detected and speech should go through the filtering process.

Step 3 : Speech features are extracted from the speech signal which is used to create the testing Vector (acoustic vector) for that utterances.

Step 4 : The testing vector is then fed into the vector quantizer

Step 5 : The predefined knowledge is used by the vector quantization to calculate the spectral distortion(distance) for each utterance and smallest distance value is selected.

Step 6 : The smallest distance value is compared with a threshold value and a decision of whether the unknown speaker to be recognized or not is made[6].

6 Result

In this work, the utterances of several speakers are taken and the data are divided in music (rock and melody) and sample voice data (Bengali and English).Each sample data is taken to train the Vector Quantizer and then all the utterances are used for recognition or testing. The input of the VQ is obtained by the frequency analysis for the given input utterances. The detail of the VQ is specified by representing the input in the form of Matrix. We've taken about 70 data for which the result is shown in percentage below.

Data Type	Correct Recognition	False Inclusion	False Rejection
Music	86 %	6 %	8 %
Speech (English)	92 %	3 %	5 %
Speech (Bengali)	91 %	4 %	5 %

7 Conclusion

This paper deals with the automatic Speaker recognition system using Vector Quantization. There are two main modules, feature extraction and feature matching. The speaker specific features are extracted using Mel-Frequency Cepstrum Co-efficient (MFCC) processor. A set of Mel-frequency cepstrum coefficients was found, which are called acoustic vectors. These are the extracted features of the speakers. These acoustic vectors are used in feature matching using vector quantization technique. There are another techniques for feature matching such as Hidden Markov model (HMM), Artificial Neural network (ANN) for speaker recognition. We've used VQ as its computational complexity is less than others. Vector Quantization is the typical feature matching technique in which VQ codebook is generated using trained data. Finally tested data are provided for searching the nearest neighbor to match that data with the trained data. The result is to recognize correctly the speakers where music & speech data (Both in English & Bengali format) are taken for the recognition process. The correct recognition is almost ninety percent. It is comparatively better than Hidden Markov model (HMM) & Artificial Neural network (ANN) because the correct recognition for HMM & ANN is below ninety percent.

The future work is to generate a VQ codebook with many pre-defined spectral vectors. Then it will be possible to add many trained data in that codebook in a training session, but the main problem is that the network size and training time become prohibitively large with increasing data size. To overcome these limitations, time alignment technique can be applied, so that continuous speaker recognition system becomes possible.

Acknowledgement. This work was supported by grants to Shahjahan from KUET, and to KM from the Japanese Society for the Promotion of Science, the Yazaki Memorial Foundation for Science and Technology, and the University of Fukui.

References

1. Rabiner, L., Juang, B.H.: Fundamentals of Speech Recognition. Prentice Hall, Englewood Cliffs (1978)
2. Rabiner, L., Schafer, R.W.: Digital Processing of Speech Signals. Prentice Hall, Englewood Cliffs (1978)
3. Davis, S.B., Mermelstein, P.: Comparison of Parametric representations for monosyllabic word recognition in Continuously Spoken Sentences. IEEE Transactions on Acoustics Speech, Signal Processing ASSP-28(4) (August 1980)
4. Buzo, L.A., Gray, R.: An Algorithm for Vector Quantizer Design. IEEE Transactions on Communications 28, 84–95 (1980)
5. Furui, S.: Speaker Independent Isolated word Recognition using Dynamic Features of Speech Spectrum. IEEE Transactions on Acoustic, and Speech Signal Processing ASSP-34(1), 52–59 (1986)
6. Furui, S.: An Overview of Speaker Recognition Technology. In: ESCA Workshop on Automatic Speaker Recognition, Identification & Verification, pp. 1–9 (1994)