Volume 2, Issue 5, May 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Isolated Word Speech Recognition Using Vector Quantization (VQ)

Dipmoy Gupta, Radha Mounima C.
Navya Manjunath, Manoj PB
Dept. of EC ,AMCEC, Bangalore
dipmoy@gmail.com

Abstract: ***The results of a case study carried out while developing an automatic speaker recognition system are presented in this paper. The Vector Quantization (VQ) approach is used for 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 code words is called a codebook. After the enrolment session, the acoustic vectors extracted from input speech of a speaker provide a set of training vectors. LBG algorithm formulated by Linde, Buzo and Gray is used for clustering a set of L training vectors into a set of M codebook vectors. For comparison purpose, the distance between each test codeword and each codeword in the master codebook is computed. The difference is used to make recognition decision.***

Keywords: ***Speech Processing, Vector Quantization, LBG algorithm, MFCC.***

I. INTRODUCTION

Speech recognition field is one of the most challenging fields that have faced the scientists from long time. The complete solution is still far from reach. The efforts are concentrated with huge funds from the companies to different related and supportive approaches to reach the final goal. Then, apply it to the enormous applications who are still waiting for the successful speech recognizers that are free from the constraints of speakers, vocabularies and environment. This task is not an easy one due to the interdisciplinary nature of the problem and as it requires speech perception to be implied in the recognizer (Speech Understanding Systems) which in turn strongly points to the use of intelligence within the systems.

The bare techniques of recognizers (without intelligence) are following wide varieties of approaches with different claims of success by each group of authors who put their faith in their favourite way.

The entire process of speech recognition can be broadly split into two subsequent phases-the training phase and the testing phase.

II. BASIC STRUCTURE OF SPEECH RECOGNITION SYSTEM

An Automatic Speech Recognition (ASR) engine tries to imitate the human auditory system. Speech recognition boils down to a matching problem – matching the input speech signal to the reference speech signals stored in our brain or on a machine, as is the case for human and machine speech recognition, respectively. So, most of the latest speech recognition engines involve 2 parts – training and recognition. Vector Quantization 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 code words is called a *codebook*.

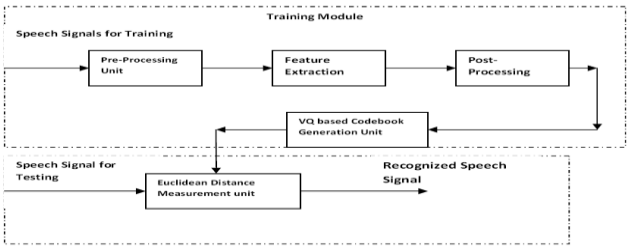


Figure1: Basic Structure of Speech Recognition system

© 2012, IJARCSSE All Rights Reserved

Mapping or clustering the feature vectors can be done a number of clustering algorithms like the LBG algorithm or the K-means algorithm. This project makes use of the LBG [Linde, Buzo and Gray] algorithm for the purpose of clustering the feature vectors.

III. SPEECH TO FEATURE VECTORS

This chapter describes how to extract information from a speech signal, which means creating feature vectors from the speech signal. A wide range of possibilities exist for parametrically representing a speech signal and its content. The main steps for extracting information are *segmentation, pre-emphasis, frame blocking and windowing, feature extraction*[1].

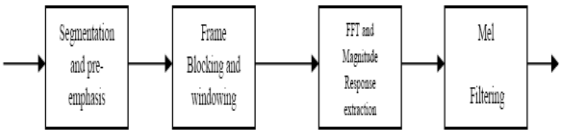


Figure2: Flow Diagram of Speech To Feature Vectors

A. Segmentation and Pre-emphasis

The speech signal is first segmented into lengths of 10/20 ms segments.

Then it is subjected to pre emphasis. The pre-emphasizer is used to spectrally flatten the speech signal. This is usually done by a high pass filter. The most commonly used filter for this step is the FIR filter described below:

$$H(z) = 1 - 0.95z^{-1}$$

The filter in the time domain will be $h(n) = \{ , -0.95\}$ and the filtering in the time domain will give the pre emphasized signal $s1(n)$:

$$s_1(n) = \sum_{k=0}^{M-1} h(k)\hat{s}(n-k)$$

B. Frame Blocking and Windowing

The concept of 'short time analysis' is fundamental to most speech analysis techniques. The assumption made is that, over a long interval of time, speech waveform is not stationary but that, over a sufficiently short time interval say about 10-30msec, it can be considered stationary. This is due to that fact that the rate at which the spectrum of the speech signals changes is directly dependant on the rate of movement of the speech articulators. Since this is limited by physiological constraints, most speech analysis systems operate at uniformly spaced time intervals or frames of typical duration 10-30 msec. In frame blocking, the continuous speech signal is blocked into frames of N Samples, with adjacent frames being separated by M (M < N).

The next thing to do is to apply a window to each frame in order to reduce signal discontinuity at either end of the block. 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. In other words, when we perform Fourier Transform, it assumes that the signal repeats, and the end of one frame does not connect smoothly with the beginning of the next one. This introduces some glitches at regular intervals. So we have to make the ends of each frame smooth enough to connect with each other. This is possible by a processing called Windowing. In this process, we multiply the given signal (frame in this case) by a so called Window Function. A commonly used window is the Hamming window.

C. FFT and Magnitude Response Extraction

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:

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

After the frames are converted to frequency domain from time domain using Fast Fourier Transform, we need to extract the magnitude response of the signals. This is so because while processing the speech signals we don't require the phase response of the signals. Thus we obtain the magnitude response of the speech signal.

IV. MFCC

This method consists of two parts: the cepstrum calculation and a method called Mel scaling.

The cepstrum method is a way of finding the vocal tract filter $H(z)$ with "Homomorphic Processing". Homomorphic signal processing is generally concerned with the transformation to linear domain of signals combined in a nonlinear way. In this case the two signals are not Combined linearly (a convolution can't be described as an simple linear combination).

MFCCs are commonly calculated by first taking the Fourier transform of a windowed excerpt of a signal and mapping the powers of the spectrum obtained above onto the Mel scale, using triangular overlapping windows. Next the logs of the powers at each of the Mel frequencies are taken, Direct Cosine Transform is applied to it (as if it were a signal). The MFCCs are the amplitudes of the resulting spectrum. This procedure is represented step-wise in the figure below:

© 2012, IJARCSSE All Rights Reserved

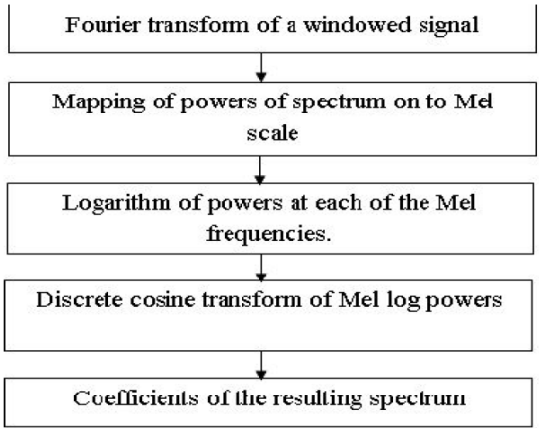


Figure3: Procedure for Forming MFCC

Psychophysical studies (studies of human auditory perception) have shown that human perception of the frequency contents of sound, 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" scale4. As reference for the mel scale, 1000 Hz is usually said to be 1000 mels. As a nonlinear transformation of the frequency scale, the following formula is used:

$$Fmel = 2595 * \log 10\ (1+FHZ/700)$$

The Non Linear Transformation can be seen in figure:

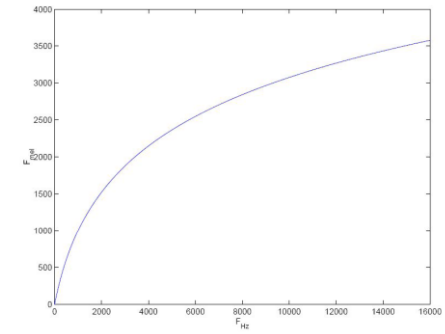
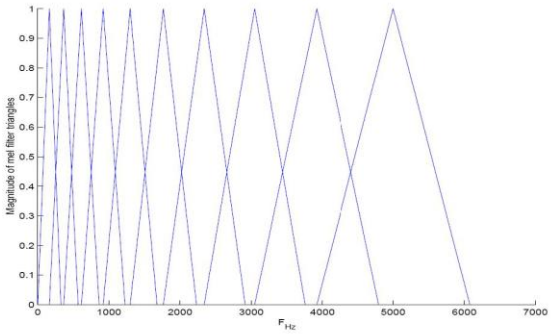


Figure4: Transformation of Hz to Mels

To apply the Mel scale to the cepstrum, a filter bank of *K* triangular band pass filters is applied to $|S(z)|$. These triangular band pass filters has centre frequencies in *K* equally spaced Mel values. The equally spaced Mel values correspond to different frequency values. If for example one wants *K* Mel scaled coefficients in the range 0-5000 Hz (Nyquist range), the first thing to do is to use the above formula to get *F*mel corresponding to *F*Hz =5000. Now the calculation of the Centrum frequency in the equally spaced Mel scale can easily be done by dividing the calculated *F*mel with *K*. Now the equally spaced Mel values are calculated and the reverse operation is done to get back to *F*Hz.

Now the Centrum of the triangular filter is found in Hz and the triangular band- pass filters can be found. The width of each filter is just the distance to the previous Centrum times 2 (the first Centrum frequency has zero as its previous Centrum value). Fig. 5 shows the Mel scaled filter bank when *K* = 10.



.

Figure5: Mel Scale Filter Bank

An extra measure to augment the coefficients derived from linear prediction or Mel-cepstrum is the *log of signal energy*. This means that, for every frame, an extra energy term is added. This energy term is calculated as:

$$E_m = \log \sum_{k=0}^{K-1} x^2(k;m) \quad 2$$

After the energy values are calculated all the energy values from all the channels are passed through Discrete Cosine Transform.

V. VECTOR QUANTIZATION

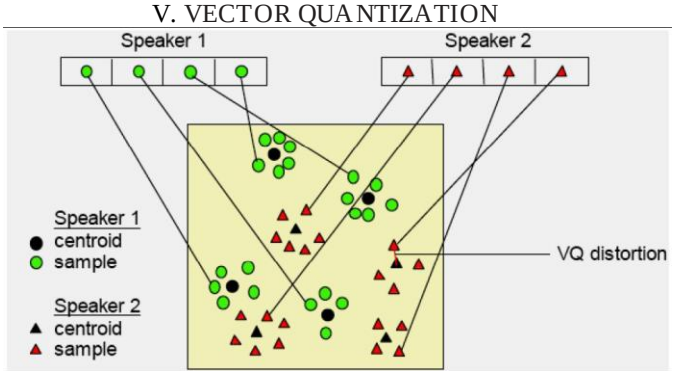


Figure6: Vector quantization codebook formation

The problem of speaker recognition belongs to a much broader topic in scientific and engineering so called pattern recognition. The goal 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

© 2012, IJARCSSE All Rights Reserved

case is applied on extracted features, it can be also referred to as feature matching.

Furthermore, if there exist some set of patterns that the individual classes of which are already known, then one has a problem in supervised pattern recognition. This is exactly our case since during the training session, we label each input speech with the ID of the speaker. The remaining patterns are then used to test the classification algorithm; these patterns are collectively referred to as the test set. If the correct classes of the individual patterns in the test set are also known, then one can evaluate the performance of the algorithm.

The state-of-the-art in feature matching techniques used in speaker recognition includes Dynamic Time Warping (DTW), Hidden Markov Modeling (HMM), and Vector Quantization (VQ)[3]. In this paper, the VQ approach is 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 codeword. The collection of all code words is called a codebook.

Figure 6 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 to the acoustic vectors from the speaker 1 while the triangles are from the speaker 2.

VI. LBG ALGORITHM

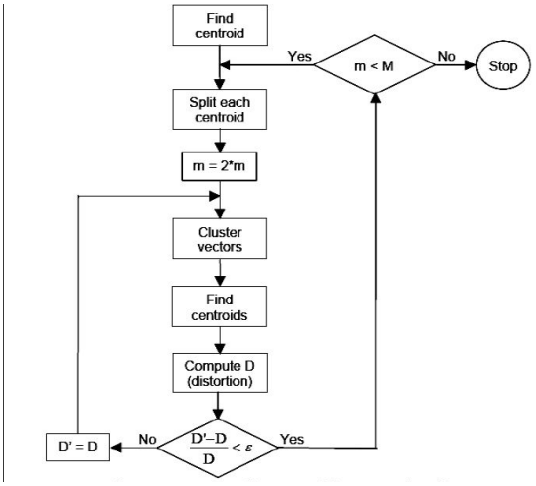


Figure7: Flow diagram of the LBG algorithm

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][2], 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[2]:

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 yn according to the rule where n varies from 1 to the current size of the codebook, and ε is 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. Centroids Update: update the codeword in each cell using the centroids 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. Figure7 shows 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.

VII. CONCLUSION AND FUTURE SCOPE

In this work, we have developed a text-dependent speaker identification system that is a system that identifies what a person has spoken. Our speaker verification system consists of two sections

(i) Enrolment section or training the system to build a database of known speakers and

(ii) Testing the system with same speakers to identify the words spoken.

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. It consists of two main parts. The first part consists of processing each person's input voice sample to condense and summarize the characteristics of their vocal tracts. The second part involves pulling each person's data together into a single, easily manipulated matrix.

The speech recognition system contains two main modules (i) feature extraction and (ii) 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 known speaker's speech by comparing extracted features from his/her voice input with the ones from a set of

© 2012, IJARCSSE All Rights Reserved

known speakers. The entire coding was done in MATLAB version 11.0. The system was tested many times with various databases and found to be very reliable. Further improvement can be obtained by increasing the reference database size. Also if we can combine voice activation detection with this procedure we can perform speech recognition on live voices and speech. Also VQ has some limitations so HMM techniques or Neural Networks can be implemented to better the procedure and increase the accuracy.

REFERENCES

[1]. Dr. H. B. Kekre, Ms. Vaishali Kulkarni, "*Speaker Identification by using Vector Quantization*", International Journal of Engineering Science and Technology, Vol. 2(5), 2010, 1325-1331.

[2]. Y. Linde, A. Buzo & R. Gray, "An algorithm for vector quantizer design", *IEEE Transactions on Communications*, Vol. 28, pp.84-95, 1980.

[3]. Rita Singh, Bhiksha Raj, and Richard M. Stern, *Member, IEEE*, "*Automatic Generation of Sub word Units for Speech Recognition Systems*", IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING, VOL. 10, NO. 2, FEBRUARY 2002.

[4]. Jesus Savage, Carlos Rivera, Vanessa Aguilar, "*Isolated Word Speech Recognition Using Vector Quantization Techniques and Artificial Neural Networks*", Facultad de Ingenieria ,Departamento de Ingeniería en Computación ,University of Mexico,UNAM,Mexico City C.P. 04510,Mexico

[5]. Jeng-Shyang Pan, Zhe-Ming Lu, and Sheng-He Sun, "*An Efficient Encoding Algorithm for Vector Quantization Based on Subvector Technique*", IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 12, NO. 3, MARCH 2003.

[6]. Mario A. T. Figueiredo, "*Scalar and Vector Quantization*",Departamento de Engenharia Electrotecnica e de Computadores, Instituto Superior Tecnico, Lisboa, Portugal.

© 2012, IJARCSSE All Rights Reserved