ENHANCED FEATURE VECTOR SET FOR VQ RECOGNIZER IN ISOLATED WORD RECOGNITION

Poonam Bansal, Amita Dev, Shail Bala Jain

Abstract: Speech recognition is always looked upon as a fascinating field in human computer interaction. It is one of the fundamental steps towards understanding human cognition and their behavior. This paper explicates the theory and implementation of Automatic Speech Recognition (ASR), in the form of speaker-dependent limited vocabulary isolated word recognizer (IWR). Any IWR contains two main phases, training phase and the testing phase. In the training phase feature extractor transforms the raw speech signal into a compact but effective representation and the extracted features are stored in the database. During the recognition phase the features are extracted by the same or different techniques and are compared with the stored one in the database [1]. In the purposed IWR the features of the speech are extracted as LPCC, Mel-frequency Cepstrum coefficients (MFCC), delta mfcc (DMFCC) and delta-delta mfcc (DDMFCC). Vector Quantization (VQ)[3] is used for word modeling process. The final recognition decision is made based on the matching score: word model with the smallest matching score is selected as a word of the test speech sample. Word recognition rate was observed to be 96% with the 20 MFCC coefficients and as we increase the feature vector size to 36 by including DMFCC and DDMFCC recognition rate increase to 99.3%. Better performances could be seen when applying this approach itself or mixed with Hidden Markov Model (HMM) in isolated-word speech recognition.

Keywords: Speech recognition, Feature extraction, MFCC, DMFCC, DDMFCC and VQ

Introduction

The Isolated Word Recognition systems were among the first speech recognition systems implemented due to rather straightforward manner in which the basic recognition units – the words can be modeled. In this paper, we will show a step-by-step approach in building such a system, the system which integrates all the stages of a speech recognition process: speech signal acquisition, parametrization, word models building and words recognition, using Vector Quantization. The paper will be structured around the two main stages of a speech recognition process: Feature extraction after the acoustic analysis of the speech signal, and the Feature matching for recognition of the basic units used in training the system (in our case, words).

Feature Extraction

The purpose of this step is to convert the speech waveform to some type of parametric representation (at a considerably lower information rate) for further analysis and processing [1, 2] which is referred as the *signal-processing front end*. The speech signal is a slowly time-varying signal (called *quasi-stationary*). When examined over a sufficiently short period of time (5 ~ 100 ms), 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, the *short-time spectral analysis* is the most common way to characterize the speech signal. A wide range of possibilities exist for parametrically representing the speech signal for the speech recognition task, such as Linear Prediction cepstral coefficients (LPCC), Mel-Frequency Cepstrum Coefficients (MFCC) and others. MFCC and LPCC are well known techniques used in any ASR to describe signal characteristics, relative to the word discriminative acoustic properties. MFCC are based on the

filtering of spectrum using properties of human speech perception mechanism. On the other hand, LPCC are based on the autocorrelation of the speech frame. There is no general agreement in the literature about what method is better. However, it is generally considered that LPCC are computationally less expensive while MFCC provide more precise result. The reason of such opinion is based on that all-pole model used in the LPC provides a good model for the voiced regions of speech and quite bad for unvoiced and transient regions. The main drawback of LPCC is that unlike MFCC it does not resolve the vocal tract characteristics from the glottal dynamics, which vary from word to word and might be useful in IWR. MFCC is perhaps the best known and most popular, and it will be used in this paper. MFCC is based on the known variation of the human ear's critical bandwidths with frequencies; 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, linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz.

Mel-frequency cepstral coefficients (MFCC), introduced by Davis and Mermel stein constitute a parametric sound representation widely used in automatic speech recognition systems. MFCC has also been successfully applied to timbre analysis. The signal is passed through a Mel spaced filterbank (based on FFTs), converted to a logarithmic scale, and then submitted to a cosine transform. MFCC provide a substantial data reduction, because a few coefficients are sufficient to represent the *cepstrum* of the acoustic signal.

Given below is the block diagram of MFCC processor.

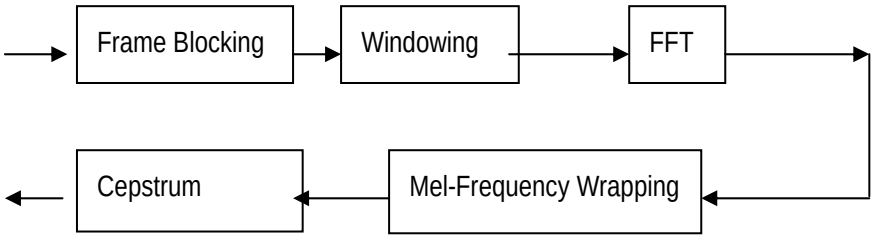

Fig.1 Block diagram of the MFCC processor

Figure 2 shows the details of the speech feature extraction through MFCC's. The step by step procedure is as follows.

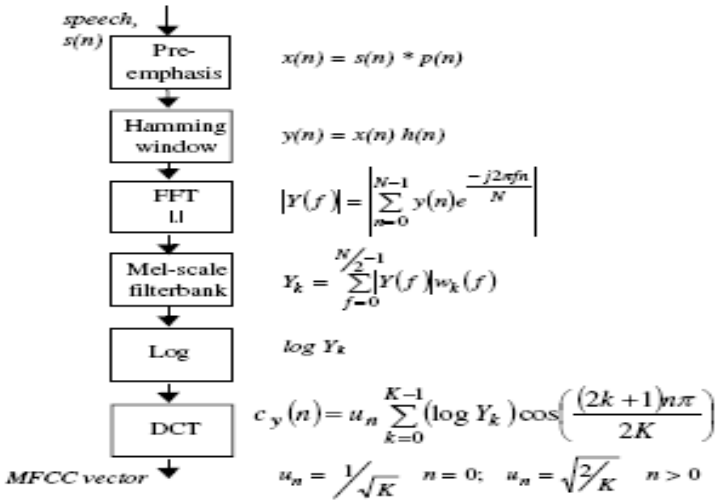

Fig.2 Block diagram for feature extraction

In this step the continuous speech signal is blocked into frames of *N* samples, with adjacent frames being separated by *M* (*M* < *N*). There is a overlapping of (N-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 (which is equivalent to ~ 30 msec windowing and facilitate the fast radix-2 FFT) and *M* = 128.

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, window to taper the signal to zero at the beginning and end of each frame. A hamming window function is used.

h(n) = 0.54-0.46cos(2pi n/N-1) … 0<=n<=N-1

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 defined on a set of N samples X_n as:

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

Mel-Frequency Wrapping

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. Our's approach to simulate the subjective spectrum is to use a filter bank, one filter for each desired mel-frequency component. That filter bank has a triangular bandpass frequency response, and the spacing as well as the bandwidth is determined by a constant mel-frequency interval. The mel scale filter bank is a series of L triangular bandpass filters that have been designed to simulate the band pass filtering believed to occur in the auditory system. This corresponds to series of band pass filters with constant bandwidth and spacing on a mel frequency scale.

Cepstrum

In this final step log mel spectrum is converted back to time. The result is called the mel frequency cepstrum coefficients (MFCC). The Discrete Cosine Transform is done for transforming the mel coefficients back to time domain. The formula is:

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

Feature Matching

The problem of word 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 state-of-the-art in feature matching techniques used in word recognition includes Dynamic Time Warping (DTW), Hidden Markov Modeling (HMM), and Vector Quantization (VQ). As the data base used is limited in our case, we have chosen VQ approach.

Vector Quantization

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* for a known word. Vector quantization (VQ) is a lossy data compression method based on the principle of block coding. It is a fixed-to-fixed length algorithm. In the earlier days, the design of a vector quantizer (VQ) is considered to be a challenging problem due to the need for multi-dimensional integration

VQ Design

The VQ design can be stated as follows. Given a vector source with its statistical properties known, given a distortion measure, and given the number of code vectors, we can find a codebook and a partition which result in the smallest average distortion.

We assume that there is a *training sequence* consisting of M source vectors:

$$\mathcal{T} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$$

This training sequence can be obtained from some large database. M is assumed to be sufficiently large so that all the statistical properties of the source are captured by the training sequence. We assume that the source vectors are K-dimensional, e.g.,

$$\mathbf{x}_m = (x_{m,1}, x_{m,2}, \ldots, x_{m,k}), \quad m = 1, 2, \ldots, M.$$

Let N be the number of code vectors and let

$$\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_N\},$$

represents the codebook. Each code vector is K-dimensional, e.g.,

$$\mathbf{c}_n = (c_{n,1}, c_{n,2}, \ldots, c_{n,k}), \quad n = 1, 2, \ldots, N.$$

Let S_n be the encoding region associated with code vector C_n and let

$$\mathcal{P} = \{S_1, S_2, \ldots, S_N\},$$

Denote the partition of the space. If the source vector X_m is in the encoding region S_n, then its approximation (denoted by $Q(X_m)$) is C_n:

$$Q(\mathbf{x}_m) = \mathbf{c}_n, \quad \text{if } \mathbf{x}_m \in S_n.$$

Assuming a squared-error distortion measure, the average distortion is given by:

$$D_{ave} = \frac{1}{Mk} \sum_{m=1}^{M} \|\mathbf{x}_m - Q(\mathbf{x}_m)\|^2,$$

where

$$\|\mathbf{e}\|^2 = e_1^2 + e_2^2 + \ldots + e_k^2$$

The design problem can be succinctly stated as follows: Given T and N find C and P such that D_{ave} is minimized. If C and P are a solution to the above minimization problem, then it must satisfy the following two criteria.

Optimality Criteria

Nearest Neighbor Condition:

$$S_n = \{\mathbf{x} : \|\mathbf{x} - \mathbf{c}_n\|^2 \le \|\mathbf{x} - \mathbf{c}_{n'}\|^2 \ \ \forall n' = 1, 2, \ldots, N\}$$

This condition says that the encoding region S_n should consists of all vectors that are closer to C_n than any of the other code vectors. For those vectors lying on the boundary, any tie-breaking procedure will do.

Centroid Condition:

$$c_n = \frac{\sum_{x_m \in S_n} x_m}{\sum_{x_m \in S_n} 1} \quad n = 1, 2, \ldots, N$$

This condition says that the code vector C_n should be average of all those training vectors that are in encoding region S_n. There is a well-know algorithm, namely LBG algorithm [Linde, Buzo and Gray], for clustering a set of *L* training vectors into a set of *M* codebook vectors.

Data Set

1. Language:	Standard Hindi
2. Vocabulary size:	A set of 1000 most frequently occurring Hindi words
3. Number of Speakers:	50 (30 Male & 20 Female)
4. Average duration of training and Testing utterances:	500-800 msec.
5. Audio recording:	S/N > 40 db
6.Sampling and quantization:	16Khz, 16-bit

Experimental Results

The performance of the word recognizer was evaluated in terms of recognition rate. We have used the following recognition measure for computing the recognition rate.

Recognition rate (%) = $N_c / N_T * 100$

Where N_c is the No. of words correctly recognized, and N_T is the Total No. of words used in the testing session.

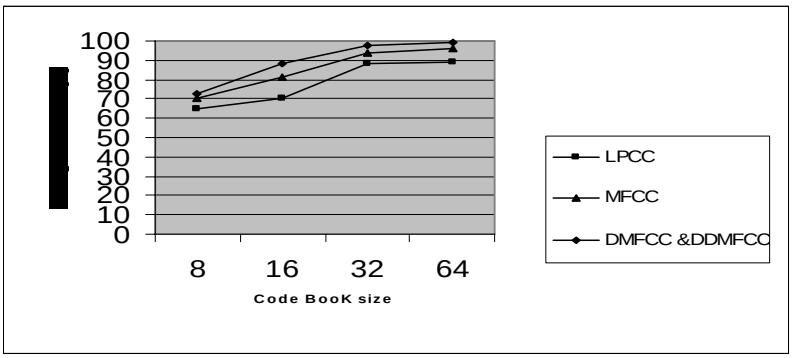

Fig 3. Performance of Recognition rate with codebook size

It was found that out of 1000 words, which were taken in the training and testing session, our IWR was able to recognize with an efficiency of 70% having codebook size of 8 and as we go on increase the size of codebook (upto 64) the recognition efficiency goes upto 96%. Another analysis was done with improved feature vector set, now we reduced the no. of MFCC coefficients to 12 and included DMFCC and DDMFCC to the whole feature set of each word. Although the size of feature vector increased a bit but with the same codebook sizes we were able to get the better recognition rate ranging from 73% for codebook of 8 to 99.3% for the codebook size of 64. Results compared are shown in the Figure 3.

Conclusion

As MFCC take care of the vocal track characteristics from the glottal dynamics, it proved to be better in our case for recognizing isolated words , in comparison with LPCC by 7% with the codebook size of 64. By enhancing the feature vector set with DMFCC and DDMFCC performance of the recognizer gets improved by 3.3%, in comparison with when only MFCC were considered. If we want to test the viability of the IWR with the larger database for a optimum size of codebook then the recognition rate can be further improved by taking into consideration the other statistical parameters of the MFCC coefficients i.e. Average, Standard deviation etc.

Bibliography

[1] L.R. Rabiner and B.H. Juang, Fundamentals of Speech Recognition, Prentice-Hall, Englewood Cliffs, N.J.Prentice-Hall, 1993.

[2] L.R Rabiner and R.W. Schafer, “Digital Processing of Speech Signals”, Prentice-Hall, Englewood Cliffs, N.J., 1978.

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

[4] "The past, present and future of speech processing," IEEE Signal Processing Magazine, May 1998.

[5] Douglas 0’Shaughnessy, “Speech Communications(Human and Machine)”, 2nd edition, University Press.

[6] A.Biem and S. Katagiri, “Cepstram-based filter-bank design using discriminative feature extraction training at varioius levels,” in Proc. IEEE Int. Conf. Acoustics, Speech, and Signal processing, 1997, pp.1503-1506

[7] S. Furui, “Speaker independent isolated word recognition using dynamic features of speech spectrum”, IEEE Transactions on Acoustic, Speech, Signal Processing, Vol. ASSP-34, No. 1, pp. 52-59, February 1986.

[8] Chulhee Lee, Donghoon Hyun, Euisun Choi, Jinwook Go and Chungyoung Lee, “Optimizing feature extraction for speech recognition”,IEEE Transactions on Speech and audio processing, Vol 11. No.1, January 2003.

[9] John G. Proakis and Dimitris G. Manolakis, “Digital signal Processing, principal, Algorithms, and applications” Prentice hall of India

[10] S. Haykin, “Adaptive filter theory”, Pearson Education publication, 4th Edition (2002).

Authors' Information

Poonam Bansal - Department of Computer Science and Engineering, Amity School Of Engineering and Technology, 580, Delhi Palam Vihar Road, Bijwasan, New Delhi 110061, India ; e-mail: pbansal89@yahoo.co.in

Amita Dev - Ambedkar Institute of Technology, Madhuban, Delhi – 110092, India; e-mail: amita_dev@hotmail.com

Shail Bala Jain - Mahila Institute Of Technology , G.G.S.I.P. Univ., Old DCE Campus, Kashmere Gate, New Delhi - 110006 , e-mail: shailbala.jain@gmail.com