Volume 6, Issue 1, January 2016 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Study of Feature Extraction for Automatic Speech Recognition

[1]Bhupinder Singh, [2]Dr. Joginder Singh

[1] Department of CSE, IGCE Abhipur, Punjab, India

[2] D epartment of Mathematics, COE/CGC Landran, Punjab, India

Abstract: In the Era of digital signal processing technology has led the use of speech processing in many different application areas like speech compression, enhancement, synthesis, and recognition. In this paper the issue of speech recognition and role of feature extraction has been studied in detail. This study includes speech feature extraction using Linear Predictive Coefficients, Cepstral analysis and then Vector quantization of isolated-valued speech feature is done.

Keywords: Automatic Speech Recognition (ASR), Linear Predictive Coding (LPC), Vectors Quantization (VQ).

I. INTRODUCTION

Automatic speech recognition by computers is a process where speech signals are automatically converted into the corresponding sequence of words in text. With recent advances, speech recognizers based upon Hidden Markov Models (HMM's) have achieved a high level of performance in controlled environment. In real life applications, however, speech recognizers are used in adverse environments. The recognition performance is typically degraded if the training and the testing environments are not the same. The goal of Automatic Speech Recognition is to develop techniques and systems that enable computers to accept speech input. The speech recognition problem may be interpreted as a speech-to-text conversion problem. Users want their voices, speech signals in to be transcribed into text by a computer.

II. FEATURE EXTRACTION

Feature extraction is the process of retaining useful information of the signal while discarding redundant and unwanted information. However, in practice, while removing the unwanted information, on may also lose some useful information in the process. Feature extraction may also involve transforming the signal into a form appropriate for the models used for classification. In developing an ASR system, a few desirable properties of the features are:

- High discrimination between sub-word classes.
- Low Speaker variability.
- Invariance to degradations in the speech signal due to channel and noise.

The goal is to find a set of properties of an utterance that have acoustic correlates in the speech signal, that is, parameters that can somehow be computed or estimated through processing of the signal waveform,. Such parameters are termed features. Next step after the preprocessing of the speech signal in the signal modeling is feature extraction. Feature extraction is the parameterization of the speech signal. This is intended to produce a perceptually meaningful representation of the speech signal. Feature extraction typically includes the process of converting the signal to a digital from (i.e. signal conditioning), measuring some important characters of the signal such as energy or frequency response (i.e. signal measurement), augmenting these measurements with some perceptually-meaningful derived measurements (i.e. signal parameterization) and statistically conditioning these numbers to form observation vectors. The objective with feature extraction to attained are:

- To untangle the speech signal into various acoustically identifiable components.
- To obtain a set of features with low rates of change in order to keep computations feasible.

Feature extraction can be subdivided into three basic operations: spectral analysis, parametric transformation and statistical modeling (Becchetti and Ricotti, 2004). The complete sequence of steps is summarized in figure 1.1.

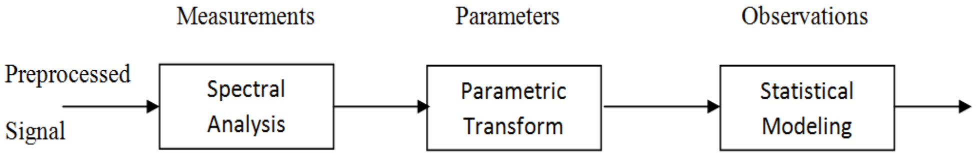

Figure 1.1 : An overview of the Feature Extraction Process

A. *Spectral Analysis:*

When speech is produced in the sense of time varying signal, its characteristics can be represented via parameterization of the spectral activity. There are six major classes of spectral of analysis algorithms i.e. Digital filter bank (Power estimation), Fourier Transform (FT Derived Filter Bank Amplitudes, FT Derived Cepstral Coefficients), Linear Prediction (LP, LP Derived Filer Bank Amplitudes, LP Derived Cepstral Coefficients) used in speech recognition system. From these classes, linear prediction gives best results. Types of Linear Prediction are explained as below:

(i) LPC (LPC analysis): Linear Predictive Coding (LPC) has been popular for speech compression, synthesis and as well as recognition since its introduction in the 1960s because it offers a reasonable engineering approach for speech signal analysis. Linear prediction models the human vocal tract as an infinite impulse response (IIR) system produces the speech signal.

Linear Predictive Coding (LPC) is a very important spectral estimation technique because it can provide an estimate of the poles (hence the formants) of the vocal tract transfer function. The LPC algorithm is a Pth order linear predictor which attempts to predict the value of any point in a time-various linear system based on the values of the previous P samples. The all-pole representation of the vocal tract transfer function, H(z) can be represented by the following equation:

$$H(z)=\frac{G}{A(z)}=\frac{G}{1+a_1z^{-1}+a_2z^{-2}+\ldots.a_pz^{-p}}$$

The values *a(i)* are called the prediction coefficients while G represents the amplitude or gain associated with the vocal tract excitation. The poles of the transfer function in equation are determined by the roots of the polynomial in the denominator. Because the LPC model is an all pole model, it can capture the resonant frequencies, or formants, but not the zeros, which are important for nasalized sounds. In addition, LPC does not adequately estimate signals which have no poles, such as some unvoiced speech and noise.

For the speech signal *s(n)* produced by a linear system, the predicated speech sample *ŝ(n)* is a function of *a(i)* and prior speech samples according to:

$$\hat{s}(n)=\sum_{i=1}^{p}a(i)s(n-i)$$

LPC analysis involves solving for the *a(i)* terms according to a lest error criterion. If the error is defined as:

$$e(n)=s(n)-\hat{s}(n)$$

$$s(n)=\sum_{i=1}^{p}a(i)s(n-i)$$

Then, taking the derivative of the square error with respect to the coefficients *a(j)* and setting it equal to zero gives:

$$\frac{\partial}{\partial a(j)}(s(n)-\sum_{i=1}^{p}a(i)s(n-1))^2=0$$

thus,

$$s(n)s(n-j)=\sum_{i=1}^{P}a(i)s(n)s(n-j)\ for\ \ j=1,\ldots\ldots,P$$

There are tow principal methods for solving above equation for the prediction coefficients *a(i)*. The first is an auto correlation method, which multiplies the speech signal by a Hamming window or similar time window, assuming that the speech signal is stationary within and zero outside, the analysis window. The autocorrelation solution to equation can be expressed as

$$R(j)=\sum_{i=1}^{P}a(i)R(|\,i-j\,|)\ \ j=1,\ldots\ldots,P$$

where, *R(j)* is an even function and is computed from:

$$R(j)=\frac{1}{\gamma}\sum_{m=0}^{N-1-j}s(m)s(m+j)\ \ j=1,\ldots\ldots,P$$

where, γ is a normalization factor. Once the autocorrelation terms *R(j)* have been calculated, a recursive algorithm named Levinson-Durbin Algorithm us used to determine the values of *a(i).*

An alternative method for determining the LPC coefficients called the covariance method is a direct Cholesky decomposition solution of the following equation.

$$R(j)=a(i)R(|i-j|)$$

This equation can be expressed in matrix form. Unlike autocorrelation method, it does not use a window to force the sample outside the analysis interval to zero. Thus, the limits on the computation of *R(j)* extend from $-P\le n\le N-1-P$.

(ii) LP-derived filter bank amplitudes: Linear prediction derived filter bank amplitudes are defined as filter bank amplitudes resulting from sampling the LP spectral model (rather than the signal spectrum) at the appropriate filter bank frequencies. Now the question is how can one efficiently sample the spectrum given the LP model? A straightforward technique to computer filter bank amplitudes from the LP model involves direct evaluation of the LP model. The spectrum is typically over sampled and averaged estimates are generated for actual filter bank amplitudes.

(iii) LP-derived cepstral coefficients: IN the last section, the LP model is leveraged to compute LP-derived filter bank amplitudes. Another logical step in this direction would be to use the LP model to computer cepstral coefficients. If the Linear Prediction filter is stable (and it is guaranteed to be stable in the autocorrelation analysis), the logarithm of the inverse filter can be expressed as a power series in z^{-1}.

$$C_{LP}(z)=\sum_{i=0}^{N_c} C_{LP}(i)z^{-1} = \log H(z) = \log(\frac{G_{LP}}{\sum_{j=0}^{N_{LP}} a_{LP}(j)z^{-1}})$$

It can solve for the coefficients by differentiating both sides of the expression with respect to z $^{-1}$ and equating coefficients of the resulting polynomials. This results in the following recursion.

- Initialization $C_{LP}(0) = \log 1 = 0$, $C_{LP}(1) = a_{LP}(1)$
- For $2 \le i \le N_c$, $C_{LP}(i) = -a_{LP}(i) - \sum_{j=1}^{i-1}(1-\frac{j}{i})a_{LP}(j)C_{LP}(i-j)$

The coefficients C_{LP} are referred to as LP-derived Cepstral Coefficients. Historically, $C_{LP}(0)$ has been defined as the log of the power of the LP error. For now, it is noted that since power will be dealt with as a separate parameter, there is no need to include it in the equations above. It can regard the Cepstral model, in which $C_{LP}(0)$ = log 1=0. The number of Cepstral coefficients computed is usually comparable to the number of LP coefficients: $0.75p \le N_C \le 1.25p$.

The cepstral coefficients computed with the recursion described above reflect a linear frequency scale. One drawback to the LP-derived cepstral coefficients is that it must work a little harder to introduce the notion of a nonlinear frequency scale. The preferred approach is based on a method used to warp frequencies in digital filter design.

B. Parameter Transforms:

Signal parameters are generated from signal measurements through two fundamental operations: differentiation and concatenation. The output of this stage of processing is a parameter vector containing our raw estimates of the signal.

(i) Differentiation: To better characterize temporal variations in the signal, higher order time derivatives of the signal model. The absolute measurements previously discussed can be though of as Zeroth order derivatives. In digital signal processing, there are several ways in which a first-order time derivative can be approximated. Three popular approximations are:

$$\tilde{S}(n) \equiv \frac{d}{dt}s(n) \approx s(n) - s(n-1)$$

$$\tilde{S}(n) \equiv \frac{d}{dt}s(n) \approx s(n+1) - s(n)$$

$$\tilde{S}(n) \equiv \frac{d}{dt}s(n) \approx \sum_{m=-N_d}^{N_a} ms(n+m)$$

The first two equations are known as backward and forward differences respectively. The first equation is same as pre-emphasis filter. The third equation represents a linear phase filter approximation to an ideal differentiator. This is often referred to as regression analysis.

The signal output from this differentiation process is denoted as delta parameter. The second-order time derivative can be similarly approximated by reapplying third equation again to the output of the first order differentiator. This output is often referred to as a delta-delta parameter. Obviously, it can extend this process to higher other derivatives.

(ii) Concatenation: Most systems post process the measurements in such a way that the operations can be easily explained in terms of linear filtering theory. Here this notion is generalized in the form of a matrix operator. For research purposes, it is convenient to view the signal model as a matrix of measurements. The signal measurement matrix usually contains a mixture of measurements: power and a set of cepstral coefficients. The concatenation is the creation of a single parameter vector per frame that contains all desired signal parameters. Some parameters such as power, are often normalized before the computation. It is common to simply divide the power by the maximum value observed over an utterance (or subtract the log of the power).

With the emergence of Markov modeling techniques that provide a mathematical basis for characterizing sequential (or temporal) aspects of the signal, the reliance upon dynamic features has grown. Today, dynamic features are considered essential to developing a good phonetic recognition capability because rapid change in the spectrum is a major cue in classification of a phonetic-level unit.

C. Statistical Modeling:

The third step of the feature extraction process is Statistical Modeling. Here, it assumes that the signal parameters were generated from some underlying multivariate random process. To learn or discover the nature of this process, it impose a model on the data, optimize (or train) the model, and then measure the quality of the approximation. The only information about the process is its observed outputs, the signal parameters that have been computed. For this reason, the parameter vector output from this stage of processing is often called the signal observations. A statistical analysis is to be performed on the vectors to determine if they are part of a spoken word or phrase or whether they are merely noise. Speech sounds such as the ‘ah’ sound in the ‘father’ exhibit several resonance in the spectrum that typically extend for 120ms. Transitional sounds, such as the ‘b’ in ‘boy’ exist for a brief interval of approximately 20 ms. Statistical model in speech recognition is shown in figure 1.2.

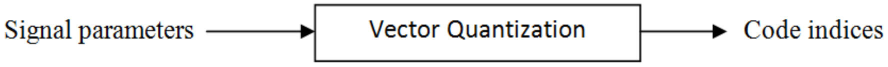

Figure 1.2: Statistical Models in Speech Recognition

Speech recognition system use extremely sophisticated statistical model, as this is one of the fundamental functions of a speech recognizer. Vector Quantization (VQ) has been useful in a wide variety of speech processing applications and forms the basis for the more sophisticated algorithm.

This basic concept of VQ applied to speech compression is schematically depicted in figure 1.3. A training speech sequence is first used to generate the codebook. The speech is segmented (windowed) into successive short frames and a vector of finite dimensionality represents each frame of speech. The vector may be in form of sampled data, FFT coefficients, autocorrelation terms, or their transformations (linear or non-linear). Codebook generation requires an iterative process much like a clustering algorithm involving a large number of spectral model vectors (codebook) so that the average spectral distortion from all the input vectors to the same spectral compression strategy in the codebook generation process is executed in the quantizer. Each input vector is mapped to the codebook entry (code-word) index corresponding to the best match vector. Speech compression or rate reduction is accomplished by using the indexes as storage or transmission parameters. For Vector Quantization, it is necessary to have a measurement of dissimilarity between the two vectors. Distortion measures based upon transformation, which retain only the smoothed behavior of the speech signal, have been applied in speech recognition, speaker identification and verification tasks.

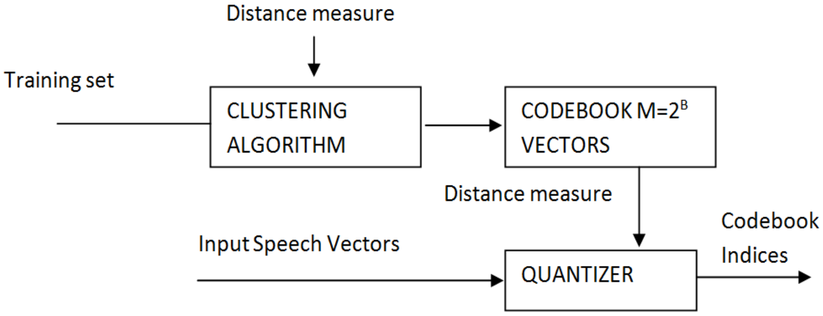

Figure 1.3 : Vectors Quantization Training and Classification Structure

To build a VQ codebook and implement a VQ analysis procedure, one needs the following:

- A larger set of spectral vectors, *{xj; j=0,...,n-1}*, which form a training set. The training set is used to create the optimal set of the codebook vectors for representing the spectral variability observed in the training set.
- A distance measure between a pair of spectral analysis, so as to able to cluster the training set vectors as well as to classify arbitrary spectral vectors into unique codebook entries.
- A centroid computation procedure: On the basis of the partitioning that classifies the training vectors into the M clusters, choose the M codebook vectors as the centroid of each of the M clusters.
- A classification procedure for arbitrary speech spectral analysis vectors that choose codebook vector closest to the input vector and uses the codebook index as the resulting spectral representation.

III. CONCLUSION

In this paper we have studied features extraction techniques used for Speech Recognition, all techniques and properties are discussed. During study we found the importance of feature extraction for the development Automatic Speech Recognition System (ASR) and a few desirable properties of the features are also discussed like : High discrimination between sub-word classes, Low Speaker variability and Invariance to degradations in the speech signal due to channel and noise. Basic operation spectral analysis, parametric transformation and statistical modeling of feature extraction are also discussed.This paper will help the researchers who willing to work in the area of speech recognition know the basic about feature extraction techniques

REFERENCES

[1] Abdulla, W. (2002), "*HMM – based techniques for speech segment extraction*", Scientific programming, IOS Press, Amesterdam, The Netherlands, Vol. 10, Issue 3, pp. 221–239.

[2] AbdulKadir K, (2010)," *Recognition of Human Speech using q-Bernstein Polynominals",* International Journal of Computer Application, Vol. 2 – No. 5, pp. 22-28.

[3] Akhuputra, V., Jitapunkul, S., Pornsukchandra, W. and Luksaneeyanawin, S. (1997), "*A speaker-independent Thai polysyllabic word recognition using Hidden Markov Model",* in proceedings of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Vol. 2, pp. 593-599.

[4] Anusuya and Katti (2009), "*Speech Recognition by Machine: A Review*", International Journal of Computer Science and Information Security, Vol. 6, No. 3, pp.181-205.

[5] Atal, Bishnu S. and Rabiner, Lawrence R. (1976), "*A Pattern Recognition Approach to Voiced- Unvoiced Classificaton with Application to Speech Recognition*", in proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP'76), Pennsylvania, Vol. 24, No. 3, pp.201-212.

[6] Becchetti, C. and Ricotti, L. (2004), “*Speech Recognition Theory and C++ Implementation”,* John Wiley & Sons, Wiley Student Edition, Singapure, pp. 121-188.

[7] Feng-Long H. (2011), *“An Effective Approach for Chinese Speech Recognition on Small size of Vocabulary*”, Signal & Image Processing: An International Journal (SIPIJ) Vol.2, No.2, pp. 48-60.

[8] Flahert, M.J. and Sidney, T. (1994), “*Real Time implementation of HMM speech recognition for telecommunication applications*”, in proceedings of IEEE International Conference on Acustics, Speech, and Signal Processing, (ICASSP), Vol. 6, pp. 145-148.

[9] Gaikwad, Gawali and Yannawar(2010), “*A Review on Speech Recognition Technique*”, International Journal of Computer Applications, Vol. 10, No.3, pp. 16-24.

[10] Gubian, M., Arnone, L. and Brofferio, S. (2005), “*A Quantitative Method for Performance Analysis of an Isolated word ASR System”,* in proceedings of 13th European Signal Processing Conference (EUSIPCO), Turkey, pp. 1-4.

[11] Hwang, T. and Chang, S. (2004), “*Energy Contour enhancement for noisy speech recognition*”, International Symposium on Chinese Spoken Language Processing, Vol. 1, pp. 249-252.

[12] Ney, H. (2003), “*An optimization algorithm for determining the end points of isolated utterances*”, in proceedings of IEEE International Conference on Acoustics, Speech, and Siganl Processing (ICASSP), Vol. 7, Issue 3, pp. 26-41.

[13] Picone, L. (1993), “*Signal modeling technique in Speech Recognition”,* IEEE ASSP Magazine, Vol. 81, Issue 9, pp. 1215-1247.

[14] Picone, J. (1990), “*Continues Speech Recognition using Hidden Markov Models*”, IEEE ASSP Magazine, Vol. 7, Issue 3, pp. 26-41.

[15] Rabiner, L. and Levinson, S. (1981), “*Isolated and Connected word Recognition Theory and selected applications*”, IEEE Transactions on Communications, Vol. 29, Issue 5, pp. 621-659.