Computer Recognition of Hindi Phonemes using Connectionist Model

Amita Dev[1], S.S Agrawal, D Roy Choudhary [2]

Central Electronics Engineering Research Institute
CSIR Complex, NPL Campus, Dr K S Krishnan Marg ,
New Delhi 110012, India
E-mail- sagrawal@ceerid.ernet.in

To develop an efficient large vocabulary , speaker independent , continuous speech recognition system, its pertinent to achieve high performance phoneme recognition. Learning Vector Quantization LVQ proposed by Kohonen is a viable candidate for tasks which involve large training set such as phoneme recognition. Objective of the present study is to find the suitability of LVQ technique for recognition of Hindi Phonemic and sub-Phonemic classes. This paper deals with the implementation of LVQ for the recognition of Hindi phonemes on a Pentium PC under windows environment. A task specific speech database of 207 Hindi words which contains all the frequently occurring consonants and vowels has been used for this study.) A total of six nets aimed at the major coarse of phonetic classes in Hindi were trained. They include 5 long vowels, unvoiced unaspirated stops, voiced unaspirated stops, nasals, fricatives and semivowels. Evaluation of each net on 350 training tokens and 40 test tokens , the LVQ achieved an average 90.3 percent recognition rate for all vowel classes, 84.6% for unvoiced stops, 80.4% for voiced stops, 90.2% for semi vowels, and 87.2% for nasals and fricatives. We also present a new Normalization Technique, [FBMN] which improves the recognition rate considerably.

Indexing terms: LVQ, Phoneme, Hindi Database

1. INTRODUCTION

ANN models attempt to achieve real time response and human like performance using many simple operating elements operating in parallel as in biological nervous system. These models have the greatest potential in areas such as speech and image recognition where many hypotheses are pursued in parallel. VQ-based recognizers have emerged as conventional and successful methods for speech recognition due to its simplicity. In VQ-based recognition , codebook is trained to minimize the quantization error for the training data. The most commonly used Linde-Buzo-Grey (LBG) algorithm has been used . The codebook trained based on the criterion of minimizing the quantization error tends to approximate the density function of the training data[1]. Here the codebooks are trained non discriminatively that is the parameters of the speech signal are estimated solely from the training data. Kohonen proposed a discriminative training procedure called Learning Vector Quantization. The LVQ trained codebook is used to define directly the classification borders between classes propagation and further LVQ requires less learning time than back propagation. Thus LVQ technique was chosen and implemented for Hindi phonemes and sub-phonemes.

This paper is organized as follows. Section 2 we discuss about Hindi speech database and in particular some specific features of Hindi sounds. In Section 3 starts with the basic details of the front-end processor. In Section 4 the phoneme recognizer architecture of LVQ aimed at phoneme recognition is discussed. Section 5 describes the

[1] Kasturba Polytechnic for Women , Computer Engg. Deptt, , Pitam Pura Delhi-88

[2] Delhi College of Engineering, Bawana Road, Samaipur, Badli, Delhi

Recognition results and Section 6 draws the conclusion of the study.

2. HINDI SPEECH DATABASE

2.1 Hindi Phonemes

The sounds of Hindi speech can be classified into vowels and consonants. There are 10 pure vowels and 29 consonants of most frequent occurrence in Hindi speech [Table-1].

MoP ↓ PoA →		Bilabials	Dentals	Retroflex	Palatal	Velar	Glottal
Stops & Affricates	UvUa	p	t	ʈ	ʧ	k	
	VoUa	b	d	ɖ	ʤ	g	
	UvAs	p^h	t^h	$ʈ^h$	$ʧ^h$	k^h	
	VoAs	b^h	d^h	$ɖ^h$	$ʤ^h$	g^h	
Fricative (Uv only)			s		ʃ		h
Vowel like		w	l	r	y		
Nasal		m	n				

Abbreviations/Symbols used :
MoP : Manner of Production
PoA : Place of Articulation
UvUa : Unvoiced Unaspirated
VoUa : Voiced Unaspirated
UvAs : Unvoiced Aspirated
VoAs : Voiced Aspirated

Table-1 Articulatory Classification of Hindi Consonants

The Hindi consonants possess certain specific features, which are not so common to German and other European languages. The most significant differences are in stops and affricates, which use both voicing and aspiration for their distinction. Aspiration is a phonemic feature in Hindi, unlike English. For example there are eight aspirated plosives and Retroflexion is another feature, which occupies a prominent place in Hindi alphabet. Many intervocalic /r/ and retroflex plosives (in non-geminated context) manifest as taps or flaps. Further the trills /r/ and /l/ have large allophonic variations in different contexts.

2.2 Data Set

A task specific data base of 207 words designed for the task specific environment voice operated railway reservation enquiry system was used in the present study. The spoken samples were recorded by 15 male, 10 female and 5 child speakers in a studio using sennheiser microphone model MD421 and tape recorder model Philips AF 6121. The speech data signal was low pass filtered and digitized at 16KHz sampling rate with 16-bit quantization using sensimetrics speech station on a Pentium PC platform. The samples were digitized using a program CSP (CEERI Speech Processor) which displays waveform, spectrogram ZCR, peak to peak amplitude and other displays. By inspecting the spectrogram segmentation was done phoneme by phoneme and labelled files were created.

3. FRONT-END PROCESSOR

The greatest common denominator of all the recognition system is the signal processing front end, which converts the speech waveform to some type of parametric representation for further analysis and processing. In recent years it has been seen that auditory systems have begun to play a larger role in motivating the design of speech recognition front end systems. Phoneme utterance typically forms a basic linguistic unit in speech. Since these are dynamic sounds the spectral pattern changes with time. Each utterance of phoneme is represented as a temporal sequence of spectral vectors. Each spectral vector corresponding to a fixed 40 ms segment has been represented using 21 spectral coefficients on a bark frequency scale. In the present experiment the labelled files were Fourier transformed (40 ms hamming window, window advancement 10 ms) and converted into 21 bark coefficients according to the formula .

$$Z[\text{Bark}] = (26.81\, f / 1960\ \text{Hz} + f) - 0.53$$

These 21 bark coefficients were fed to LVQ recognizer.

3.1 Frame Bark Mean Normalization [FBMN]

It has been noticed that phoneme recognition performance degrades because of variability in the acoustic realization of the utterance, which can come from various sources. First it may result from change in the environment as well as position and characteristics of the microphone. Second with in speaker, variability can result from change in speaker's physiological state, speaking rate, voice quality, socio-linguistic background, dialect, vocal tract size, shape etc. These results in changes in amplitude, duration and SNR ratio for a given utterance. Hence in order to make the system more robust to above said distortions we implemented a normalization technique [FBMN] by which bark coefficients were normalized to have zero mean and unit variance within a given frame[2]. The normalization coefficients were calculated over a

relatively short sliding window (frame). The feature vectors have been normalized as follows.

$$\hat{c}_{t-T}(j) = \frac{c_{t-D}(j) - \mu_t(j)}{\sigma_t(j)}$$

where

- $c_{t-D}(j)$ is the jth component of the original feature vector at time t-T.
- $\hat{c}_{t-T}(j)$ is the normalized version
- T denotes the delay in terms of feature vectors.

The normalization coefficients, mean $\mu_t(j)$ and standard deviation $\sigma_t(j)$, for each feature vector component j are calculated over the sliding finite length normalization window as shown below

$$\mu_t(j) = 1/N \sum_{n=1}^{N} c_n(j)$$

and standard deviation

$$\sigma_t(j) = \sqrt{1/N \sum_{n=1}^{N} (c_n(j) - \mu_t(j))^2}$$

where

- N denotes the normalization segment length in terms of the feature vectors. Here the mean removal can be regarded as the linear High Pass Filter and division by standard deviation act as an Automatic Gain Control.

In another set of experiment, Normalized Bark coefficients were fed as input to the network. Comparison of the recognition score with Bark and Normalized Bark coefficients is illustrated in the subsequent sections.

4. PHONEME RECOGNIZER

The basic idea of VQ based phoneme recognition approach is to compress large number of short term spectral vectors into a small set of code vectors. A codebook can be viewed as a generalization of the long-term average where the short term spectral variations due to different textual content are not averaged out but modeled by separate code vectors. Assume that a number of 'codebook vectors' **mi** are placed into the input space to approximate various domains of the input vector **x** by their quantized values. Usually several codebook vectors are assigned to each class of **x** values, and **x** is then decided to belong to the same class to which the nearest **mi** belongs. Let

$$c = \arg\min(\|x - mi\|) \qquad (1)$$

defines the nearest **mi** to **x**, denoted by **mc**.

Values for the **mi** that approximately minimize the misclassification errors in the above nearest-neighbor classification can be found as asymptotic values in the following learning process. Let **x(t)** be a sample of input and let **mi(t)** represent sequences of the **mi** in the discrete-time domain. Then the following equations define the basic LVQ process :

$mc(t+1) = mc(t) + alpha(t)[x(t) - mc(t)]$
if x and mc belongs to the same class

$mc(t+1) = mc(t) - alpha(t)[x(t) - mc(t)]$
if x and mc belongs to different class

$mi(t+1) = mi(t)$ for i not in c.

The successful modeling of the underlying acoustic classes allows the VQ- based systems to achieve high recognition accuracy.

We have used Euclidean distortion measure as it is simple to implement and it is experimentally shown to give satisfactory performance. The Euclidean distance between two vectors x and y of length L is given by

$$d(x,y) = \|x-y\|^2 = \sum_{i=0}^{L-1} (xi-yi)^2$$

5. RECOGNITION RESULTS

Three set of experiments were conducted for Hindi phonemes recognition.

EXPERIMENT-1

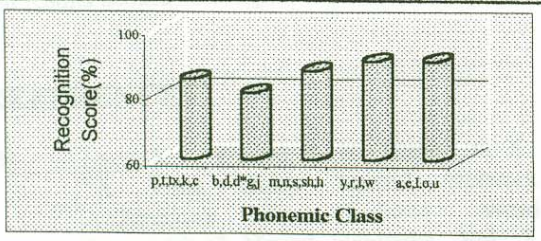
In the first experiment we applied LVQ recognizer to the following phoneme classes as these constitutes the entire phoneme set for Hindi language.

1.Unvoiced unaspirated stops **(/p/,/t/,/t*/,/k/,/c/)**
2.Voiced unaspirated stops **(/b/,/d/,/d*/,/g/,/j/)**
3. Nasals & Fricatives **(/m/,/n/,/s/, /sh/, /h/)**
4. Semivowels **(/y/,/r/,/l/,/w/)**
5. Vowels **(/a/,/e/,/i/,/o/,/u/).**

A total of 5 nets aimed at the major coarse of phonetic classes in Hindi were trained using LVQ. Evaluation of each codebook on the test data revealed that LVQ recorded a recognition score of 90.3% for vowels,84.6% for unvoiced stops, 80.4% for voiced stops, 90.2% for semivowels, 87.2% for fricatives and nasals as shown in Table –2.

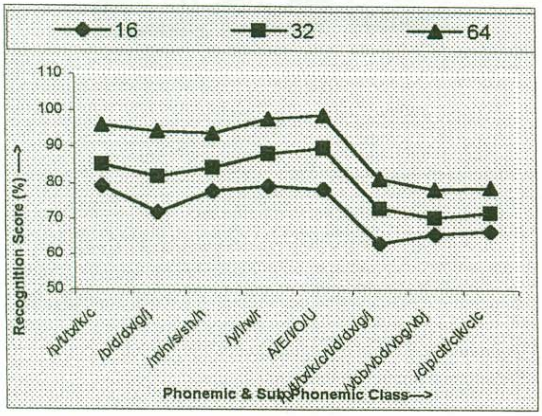
Phoneme Class	**LVQ**
UvUa (p/t/tx/k/c)	**84.6%**
VoUa (b/d/dx/g/j)	**80.4%**
Fricatives & Nasals /m/n/s/sh/h/	**87.2%**

Semi Vowels (/y/l/w/r/)	**90.2%**
Vowels (a/e/i/o/u)	**90.3%**



HINDI PHONEME RECOGNITION SCORES USING LVQ TECHNIQUE
Table-2

In order to study the effect of codebook size on recognition score , experiment was repeated for codebook size 16 32 and 64. As expected recognition score improved with the codebook size. Figure 3 shows the effect of change in codebook size on the performance of LVQ recognizer.



RECOGNITION SCORE DUE TO CHANGE IN CODEBOOK SIZE USING LVQ MODEL
Figure-3

EXPERIMENT-2

In another experiment, in order to study the effect of recognizers on the recognition of sub-phonemic units, voice bar and closure portions of stop consonants were labelled separately and codebook was generated and system was tested for Voice Bar of Voiced stops and closure of Unvoiced stops. The recognition score obtained using LVQ was 70.7% for voice bar and 72.5% for closures of voiced and unvoiced stops[3].

As expected there was indeed degradation in recognizer's performance as compare to full phoneme of the same class as shown in Table 3. The low recognition scores were due to the less acoustic information available in voice bar and closure.

Sub-Phonemic Class	LVQ
VoiceBars (Vbb/Vbd/Vbg/Vbj)	70.7
Closures (Clp/Clt/Clk/Clc)	72.5

COMPARISON OF HINDI SUB PHONEME RECOGNITION SCORES
TABLE- 3

EXPERIMENT-3

Improvement in the Recognition Score with [FBMN]

In order to study the effect of normalized bark coefficients on the recognition scores another set of experiment was conducted especially for those phonemic and sub-phonemic classes for which recognition scores obtained was poor. For this bark coefficients were normalized as per the proposed technique and 21 Frame Bark Mean Normalization [FBMN] coefficients were calculated. These 21 FBMN coefficients were given as input to the network. It was found that there was a significant improvement in the recognition scores as shown in Table-4.

PHONEMIC/ SUB-PHONEMIC CLASSES	LVQ	
	BARK	FBMN
(/b/,/d/,/d*/,/g/,/j/) Voiced Stops	80.4	91.7
(/clp/,/clt/,/clk/,/clc/) Closure of Unvoiced stops	72.5	75.3
(/p/,/t/,/t*/,/k/) UnVoiced stops	84.6	92.6
(/vbb/,/vbd/,/vbg/,/vbj/) Voice Bar of Voiced Stops	70.7	75.3

ON THE IMPROVEMENT IN THE RECOGNITION SCORE WITH FBMN.
(Table-4)

6. CONCLUSION

In this paper we have presented LVQ recognizer for recognition of Hindi phoneme and sub-phoneme classes. A close observation reveals that in general recognition score for Unvoiced unaspirated and for Voiced aspirated stops was less as compare to other phoneme classes. An analysis of confusion matrix for Hindi voiced and unvoiced stops revealed that the retroflex stop /d*/ and /t*/ got confused with their nonretroflex counterparts. an In an another experiment there was noticeable improvement in the recognition scores was

obtained when FBMN coefficients were used. It is felt that experiment should be extended on TIMIT database for evaluation of recognition algorithms and results should be compared with our database.

REFERENCES

[1] Lang, K.J., Waibel, A. H., Hinton, G. E., "A Time Delay Neural Network Architecture for Isolated Word Recognition" Neural Networks, Vol. 3, pp. 23-43, 1990.

[2] Dev A, Agrawal, S. S., "A Novel MFCCs Normalization Technique for Robust Hindi speech Recognition " 17th International Congress on Acoustics (ICA) Rome September 2-7, 2001.

[3] Dev A, Agrawal, S. S., Choudhary D Roy., "On the Recognition of Hindi Phonemic and Sub-Phonemic Classes using Time Delay Neural Networks" IEEE EIT 2001, June 7-9 2001, Oakland University, Rochester, Michigan.