A Statistical Approach Towards The Recognition of Hindi Language Words

Vinay Kumar, V.P Pyara

A Statistical Approach Towards The Recognition of Hindi Language Words

Vinay Kumar

Abstract: *The probabilistic scheme for studying the time series is Hidden Markov Modeling. In this letter we will show that how this technique could be used for Hindi language alphabet recognition purpose. The motivation and advantages are also discussed for choosing Hindi.*

Introduction: Speech recognition by machine is one of the most fascinating areas for research from last several decades. People are trying for developing software which can easily hear, understand, and speak to the users. These tasks are implemented by using one or all of the following broad categories:

1. Speech recognition for isolated or continuous word stream.
2. Natural language processing used for understanding of the machine.
3. Speech synthesis to allow machine to speak to the user.

The work to be discussed in this letter is of first category. The complete implementation of even the first category will require the machine to understand speaker independent continues speech, but here we have applied it only for speaker dependent, isolated word recognition, isolated word recognition.

Hindi Language: Hindi is the national language of India and people in several other countries like Nepal, Mauritius, Singapore, Fiji, Guyana, Suriname, Trinidad, UAE, etc. can easily understand and even speak it. To develop a speech recognition system is an easier task for Hindi, as it offers several advantages over other languages, like Hindi does not have separate phoneme and alphabet set; i.e., there is more or less one to one correspondence between what is written and what is spoken, the alphabets are very well categorized on the basis of similarities in the articulation methods of its letters, this second property of this language makes it free from homonyms reducing the complexity of the system to handle them. The problem with the language is that sometimes the vowel which is associated with the consonant is not pronounced depending on the context; e.g., Krishna is mapped to /k/ /r/ /i/ /s/ /n/ ignoring the vowel /a/associated with the consonant /l/. It is called as the Inherent Vowel Suppression. Figure-1 and Figure-2 represents the vowels and consonents of Hindi Language respectively.

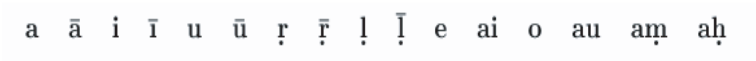

Figure 1: Vowels

Statistical Modeling: To prepare a model of Hindi speech we have used LPC (Linear Predictive Coding), VQ (Vector Quantization) [4] [7] for front end processing of the speech signals. While at the back end Hidden Markov Modeling [1][4][5][10] was used for the recognition purpose. A noise elimination model is also used to eliminate the undesired frequency signals, we have assumed that the environment is quiet and the noises present are only high frequency one.

Implementation: We will not discuss here the mathematical details of implementation as they can be reffered from [1][3][4][7]. The words are recorded using a microphone and are directly recorded to hard disk. The words are sampled at

ka kha ga gha ṅa

ca cha ja jha ña

ṭa ṭha ḍa ḍha ṇa

ta tha da dha na

pa pha ba bha ma

ya ra la va

śa ṣa sa

ha

Figure 2: Consonants

16 kHz and size of each sample is kept 8 bits. The words recorded are enframed with 20ms window with an overlap of 5ms; a Hamming window [9] is used for this purpose. The purpose of window is to weight, or favor, samples towards the center of the window. This characteristic coupled with overlapping analysis performs an important function in obtaining smoothly varying parametric estimates. After this the Noise elimination takes place. The input of the figure is speech signals in time domain, the speech signals comprised of only low frequency signal (between 4KHz to 40 KHz) as human ear can detect only low frequency signals, we calculate Discrete Fourier Transform and then pass it through a Band Pass Filter (BPF) and then again calculating the Inverse Discrete Fourier Transform we receive the speech samples with high frequency noise components eliminated. Although this technique will not eliminate the noise completely but high frequency components will be suppressed. After it we will use an All Pass Filter (APF) so that it makes the phase linearly varying for phase compensation. The following figure-3 shows this arrangement.

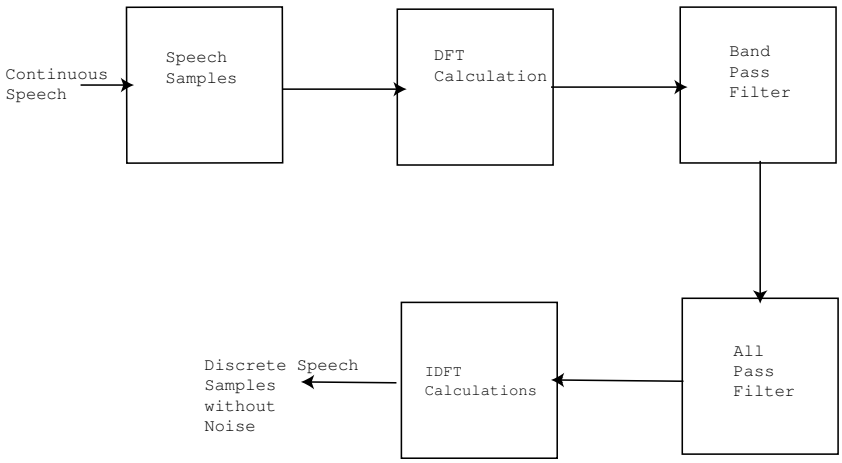

Figure 3: Figure-3

After enframing the signal next step is the LPC coefficient calculation, which is done on the enframed signal. There are several ways to calculate the LP coeffi-

cients but we have used autocorrelation method [1] as it is almost exclusively used in speech recognition because of its computational efficiency and inherent stability. The autocorrelation method always produces a prediction filter whose zeros lie inside the unit circle in the z-plane. We have used 14th order LPC filter in our system.

The large data which is obtained by above mentioned analysis is than compressed for the real calculations and to use it the purpose of HMM development. For the compression we have used the Vector quantization methods. The Vector Quantization approach is applied on the LPC coefficients. For one word we have generated 32 centers; i.e., a codebook is generated having 32 centers. Then the same word is uttered for 10 times and a joint codebook of all individual codebooks for the same word is generated. The same process is repeated for all words.

These codebooks are then used to generate HMM for every word for the training purpose. After having robustly trained models using Genetic Algorithm approach we can move further for the recognition purpose.The GA approach is

THE GA-Based Approach: Genetic Algorithms were initially introduced for optimisation problems in signal processing. The basic idea is that candidate solutions for a particular problem are evolving during consecutive reproduction cycles. The reproduction operations, selection, mutation/crossover and replacement, imitate processes known from the nature. At the end of the process winner candidates represent the best solutions for the problem. **The Algorithm:** The algorithm applied is as follows:
1. An initial population is generated randomly. A member of the population, a chromosome, is a guess for decoding the input sentence in terms of semantic units. These semantic unit candidates are the genes in a chromosome. Though it is clear that several words can form one semantic unit within a sentence, the initialisation process assigns a randomly chosen semantic unit to each words in the sentence. The size of the population is fixed to be ten.
2. The chromosomes of the population are ranked based on the fitness values, which are calculated using the conditional probabilities stored in the state-wise histograms of the discrete HMMs.
3. The lower half of the ranked population (i.e. the 5 chromosomes with the lowest fitness values) is ignored while the upper half is mated among each other producing an offspring. The offspring is created via mutation, cloning and crossover.
4. Steps 2 and 3 are repeated as long as the overall fitness value does not change significantly over a pre-determined interval. The output of the process is the best solution in the last population.

Since the search space, determined by the length of a sentence and the number of the semantic units is rather small, the population size was also kept small. The size of the population and the number of parent chromosomes were fixed during all the experiments to be ten and five, respectively. It was considered more interesting to see how different fitness value computations and genetic operations affect the performance. These aspects are explained next.
Fitness value computation. The fitness value for a particular chromosome is calculated as the sum of the conditional probabilities, which are the stored b probabilities in the states of the discrete HMMs. Two types of fitness values are used in the experiments. The first one is computed as

$$F_{max} = \sum max_{k \in S}(b^{C_{i,k}}(w_i)) fori = 1 \ldots W \quad (1)$$

In a sentence with W words, each word wi corresponds to a semantic unit Ci.

The sequence of these semantic units is the chromosome. For a word-semantic unit pair a score is looked up in the discrete HMM corresponding to Ci. This score is the b probability, and if a model has more states ($S > 1$), than the one with the highest value is chosen. This type of fitness value which takes the highest b probability for a word within an HMM is named as F_{max}.
Another way to compute the fitness value is by taking the average of b probabilities over all the states in a model. This is defined as follows:

$$F_{ave} = \sum M_{k \in S}(b^{C_{i,k}}(w_i)) for i = 1 \ldots W \tag{2}$$

where M(X) refers to sample mean

In case of crossover, two parent chromosomes mutually exchange their parts at section points which is determined randomly. Cloning is a special case of crossover, when the randomly chosen crossover section point is placed to the beginning (being 0) or to the end of the chromosome (being equal to the length of the chromosome). In this case the two children chromosomes are exact replications of their corresponding parent chromosomes. In case of mutation one item of a chromosome is chosen randomly and replaced with another item, which is picked up again randomly. The results of mutation and crossover operations form the offspring which then are replacing the 5 chromosomes with the lowest fitness values.

Stop criteria: In the initial experiments the changes in the overall fitness value were checked and the process stopped when no more significant changes occurred. However, it became clear that after 100 iterations not much changes happen anymore, the five best solutions within a population do not alter and even with mutation and crossover no better guesses get higher in the population.

During the recognition phase we utter a word out of the group for which we have codebooks this word is then vector quantized and GA based search is then used to find the best matching codebook out of the 5 codebook we have used. The optimization of the observation sequence with respect to the model is done using Forward-Backward algorithm[1][4].

The whole procedure which is above mentioned is done on following combination of vowels and consonants of Hindi language their pronunciation keys could be found in [16].

The words were

1. aap

2. tum

3. main

4. yahaan

5. kab

Results: The words were uttered by females. The results obtained are shown in the table below:

Word	Speaker-1	Speaker-2
aap	71.50	77.80
tum	84.30	86.70
main	82.60	85.60
yahaan	81.60	81.10
kab	84.50	77.80
Average	80.90	81.82

Conclusion: We have implemented the system for Hindi vocabulary. The system could be improved further with improved noise detection, and taking into account other aspects, as dicussed previously, of the language. Although Hindi is not a language of masses in the world but as Hindi is directly related to Sanskrit, we can easily design a system for Sanskrit, which is supposed to be the best structured language in the world as it has the following properties which make it easy for developing a speech recognition model:

1. phonetic characteristics; i.e., the words retain their sound in any word, and

2. its non destructive nature; i.e., words do not loose their meaning irrespective of where they are going to be used.

References

1. Joseph Picone, "Continuous Speech Recognition Using Hidden Markov Modeles", IEEE ASSP Magazine July 1990.
2. B.H.Juang, "The Past, Present, and Future of Speech Processing", IEEE Signal Processing Magazine, May 1998.
3. S.E.Levinson, L.R.Rabinar, and M.M.Sondhi, "An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition", The Bell System Technical Journal, vol. 62, No. 4, April, 1983.
4. L.R.Rabinar, S.E.Levinson, and M.M.Sondhi, "On the Application of Vector Quantization and Hidden Markov Models to Speaker-Independent, Isolated Word Recognition", The Bell System Technical Journal, vol. 62, No. 4, April 1983.
5. J.Makhoul, and R.Schwartz, "What is a Hidden Markov Model", The Voice of Computer, November 1997.
6. L.R.Rabinar, "A Tutorial On Hidden Markov Model and Selected Applications In Speech Recognition", Proceedings of IEEE, vo l. 77, No. 2, February 1989.
7. A. Gersho, S. Wang, K. Zeger, "Vector Quantization Techniques in Speech Coding," in Advances in Speech Signal Processing, S. Furui and M.M. Sondhi, eds., Mar- cel Dekker, New York, pp. 49-84, 1992.
8. J.G.Proakis, D.G.Manolakis, "Introduction to Digital Signal Processing", Macmillan Publishing Company, New York, 1988.
10. L.R.Rbinar and B.Gold, "Theory and Application of Digital Signal Processing", PHI, 1999.
11. K.F.Man, K.S.tang and S.Kwong, "'Genetic Algorithms", Springer 1999.
12. www.datacompression.com
13. http://en.wikipedia.org/wiki/$Sanskrit_{l}anguage$