WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Phoneme Recongnition Using Machine Learning

Sanyam Jain [a], Navdeep Saluja[b]*

[a] *Infinity Management And Engineering College, Pathariya Jat, Sagar, M.P, India*
[b] *Infinity Management And Engineering College, Pathariya Jat, Sagar, M.P, India*

A B S T R A C T

Speech recognition has been an integral part of human life. An application developed on the basis of speech recognition has a high degree of acceptance. Here in this project, Phoneme Recognition using Machine Learning is done. Various steps we followed to do phoneme recognition are mainly feature extraction from given phoneme, model generation, distance calculation with respect to new sample using generated model, identification of particular phoneme etc. In this project, we have employed machine learning techniques. Firstly, we have found phonation boundary given speech sample using SFS tool. After that, we have used MATLAB tool for feature extraction, model generation and for calculation of K-Nearest-Neighbor and to identify a particular phoneme. We have tried to make our approach as simple and as efficient so that it has high utility.

Keywords: K-Nearest-Neighbor, MATLAB, HMM, SFS tool, FFT Analysis.

Introduction

Speech recognition technology has advanced tremendously over the last four decades, from ad-hoc algorithms to sophisticated solutions using hill-climbing parameter estimation and effective search strategies. The time alignment algorithm with dynamic time warping (DTW) was an implementation of dynamic programming for matching acoustic utterances [1], promoted by both AT&T on the East Coast and George White at Fairchild on the West Coast. Once the speech signal could be efficiently characterized, the floodgates opened for speech applications including speech coding using linear predictive coding (LPC), word spotting and speech recognition [5]-[4]. During the 1970's, the government-funded research and testing programs in "word spotting", where known words were to be identified in the speech of talker's unknown to the training algorithm, yielding time warping algorithms for matching acoustic utterances [2]-[3]. This technology was introduced to the community in a conference in 1982, but the International Business Machines (IBM) research organization had been exploring this space since 1970 in large vocabulary applications [8]. Most modern embedded speech recognition applications use the Hidden Markov Model (HMM) technology. HMM allow phonetic or word rather than frame by frame modeling of speech [6]. They are also supported by very pretty convergence theorems and an efficient training algorithm. Unlike DTW, the HMM recognition algorithms model the speech signal rather than the acoustic composite and they tend to be more robust to background noise and distortion [3]. In current applications, the cost associated with a misrecognition is small enough so that sophisticated noise suppression techniques are not economically viable. It is often enough to train an HMM system with some noisy data [7].

Nomenclature

VOP- Vowel onset point

DTW dynamic time warping

FFT- Fast fourier transform.

System Design

SFS is a free computing environment for PCs for conducting research into the nature of speech. It comprises software tools, file and data formats, subroutine libraries, graphics, special programming languages and tutorial documentation. It performs standard operations such as acquisition, replay, display and labeling, spectrographic and formant analysis and fundamental frequency estimation. It comes with a large body of ready-made tools for signal processing, synthesis, recognition, as well as support for our own software development.

The steps involved are as follows: i. Firstly, Speech Filling System (SFS) tool is used to identify the phonation boundary. We pass the given wav file to the SFS tool which gives the output with phonation marker, that is it gives start time and end time of each phonation. ii. Second task is creation of training and testing sets of data. It uses two sets of data. a. Training set: The training data contains extracted features data with expected output. Extracted features are epoch interval, ratio of residual to signal energy, symmetric itakura distance, strength of instants etc. b. Testing set: It is used to evaluate the performance of trained model.

A. Acquiring and Chunking the audio signal

Use the SFSWin program to record directly from the audio input signal on our computer. To acquire a signal using SFSWin, choose File|New, then Item|Record. See Figure 3.1. Choose a suitable sampling rate, at least 16000 samples/sec is recommended. It is usually not necessary to choose a rate faster than 22050 samples/sec for speech signals.

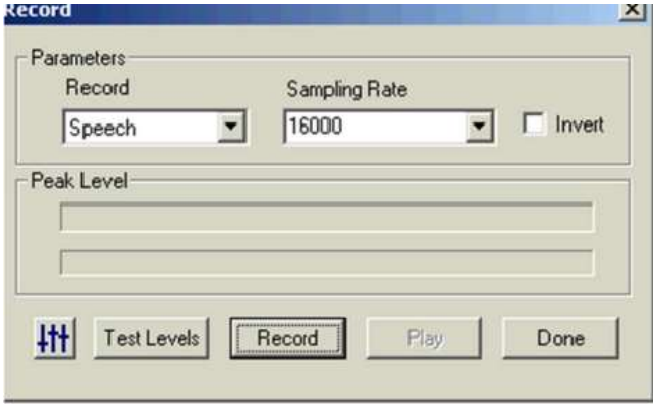

Fig.1 SFSWin record dialog

***B.* Preparing the signal**

If the audio recording has significant amounts of background noise, we may like to try and clean the recording using Tools|Speech|Process|Signal enhancement. The default setting is "100% spectral subtraction"; this subtracts 100% of the quietest spectral slice from every frame. This is a fairly conservative level of enhancement, and we can try values greater than 100% to get a more aggressive enhancement, but at the risk of introducing artifacts.

***C.* Chunking the signal**

If our audio recording is longer than a single sentence, we will almost certainly gain from first chunking the signal into regions of about one sentence in length. Chunking involves adding a set of annotations which delimit sections of the signal. An easy way to chunk the signal is to automatically detect pauses using the "npoint" program. This takes a speech signal as input and creates a set of annotations which mark the beginning and end of each region where someone is speaking.

It is a simple and robust procedure based on energy in the signal. To use this, select the speech item and choose Tools|Speech|Annotate|Find multiple endpoints. See Figure 3.4. If we know the number of spoken chunks in the file (it may be a recording of a list of words, for example), enter the number using the "Number of utterances to find" option, otherwise choose the "Auto count utterances" option. Put "chunk" (or similar) as the label stem for annotation.

***D.* Phonetic transcription**

Spelling to sound We have a chunked orthographic transcription of our recording roughly aligned to the audio signal. The next stage is to translate the orthography for each chunk into a phonetic transcription. If we know the language, this is a largely mechanical procedure of looking up words in a dictionary. If the language is English, the mechanical part of the process can be performed by the antrans program. 16 The SFS program antrans performs the phonetic transcription of orthography using a built-in English pronunciation dictionary. The program takes orthographic annotations as input and produces transcribed annotations as output, in which only the content of the labels has been changed. See Figure 4.

Feature extraction using MATLAB tool

Instant of significant excitation (ISE)—

The instants of significant excitation correspond to the instants of glottal closure (epochs) in the case of voiced speech, and to some random excitations like an onset of burst in the case of nonvoiced speech. Instants of significant excitation are computed from the global phase characteristics of minimum phase signals.The average slope of the unwrapped phase of the short-time Fourier transform of the linear prediction residual is calculated as a function of time. Instants, where the phase slope function makes a positive zero-crossing, are identified as significant excitations.

The distance between two successive instants of significant excitation is the epoch interval. After extracting the instants, this is obtained by plotting at each epoch, the distance of the next epoch as amplitude. The value varies randomly in the case of unvoiced sounds but remains almost uniform in the case of voiced sounds. Hence this will have a uniform structure for the voiced sounds.

Mathematical formula for energy calculation: Assume,R(n)={r1,r2,r3,..,rn} is residual signal and

s(n) ={s1,s2,s3,..,sm}is speech signal

EnergyR (n) = (r12 +r22 +r3 2+..+rm2)/n

Energy s (m) = (s12 +s22 +s3 2+..+sm2)/m;

Ratio of residual to signal energy =Energy R (n)/Energy s (m).

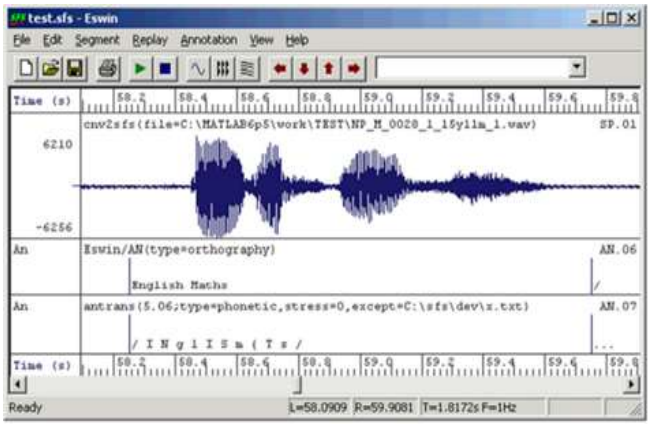

Fig. 2 Transcribed annotations

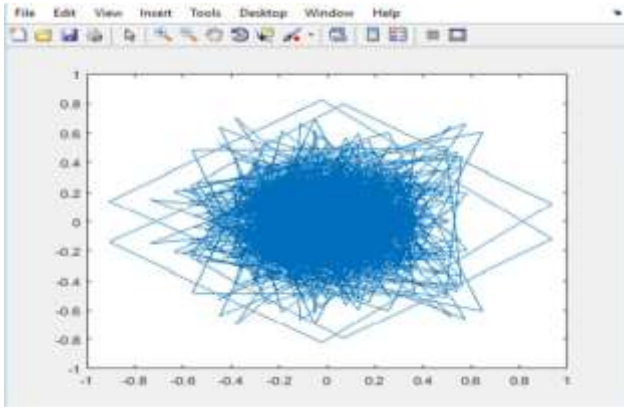

Fig.3 fast fourier transform of residual signal

K Nearest Neighbor

In pattern recognition, the k-nearest neighbor algorithm (KNN) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether KNN is used for classification or regression.

In KNN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor. In KNN regression, the output is the property value of the object. This value is the average of the values of its k nearest neighbors.

KNN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The KNN algorithm is among the simplest of all machine learning algorithms.

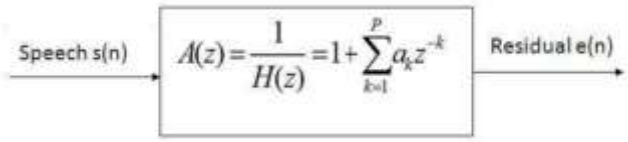

Fig 4: Computing the LP residual by inverse filtering

Both for classification and regression, a useful technique can be to assign weight to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than 29 the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of 1/d, where d is the distance to the neighbor. The neighbors are taken from a set of objects for which the class (for KNN classification) or the object property value (for KNN regression) is known.

This can be thought of as the training set for the algorithm, though no explicit training step is required. In our project, we have used the k-nearest-neighbor algorithm in which we have found k (k=1,3,5,7,9) nearest matching phonetic character for given input and among them, on the basis of the majority we have decided an output.

Implementation

The first step in our implementation is marking phonation boundary. Marking phonation boundary using SFS tool

Step 1: Take audio file in ‘wav’ format

Step 2: Open SFS tool and go to ‘File | New’ and pass the audio file.

Step 3: Then for annotation go to ‘Tools | Speech | Annotate | Find Multiple End points’.

Step 4: Rename the chunk for understanding purpose. After following above steps, we get output with start and end point of every phonetic character in the audio file (Note: In audio file between each phonetic character at least one second of a gap is necessary).

Input: wav file contains (Ka (क), Kha (ख), Ga (ग), …. , Dnya (ǒ)) After speaking consonant from Ka (क) to Dnya (ǒ) of Indian language (in Marathi), we get start and end point.

for each phonation. After phonemes are marked, we need to identify particular phoneme.

Phoneme recognition algorithm

Different steps for phoneme recognition are

Step 1: Input- Take audio file of phonetic character in wav format.

Step 2: Calculate following feature of input audio file a. Epoch interval b. Ratio of residual to signal Energy c. Symmetric itakura distance d. Strength of instants

Step 3: The feature data is stored in an array format. Then we convert each feature data into a single value by using following methods a. Weighted average value b. Mean value c. Root Mean Square (RMS) value d. Mid-point of weighted average range

Step 4: Store all these values in Excel file [Note: Different Excel file for each method].

Step 5: Apply K nearest neighbor to find nearby K matching phonations from training data.

Step 4: Store all these values in Excel file [Note: Different Excel file for each method].

Step 5: Apply K nearest neighbor to find nearby K matching phonations from training data.

Step 6: Make a decision on the output of KNN on the majority basis.

Step 7: Finally, we get the output which determines the matching phonetic character.

Results and Conclusion

Algorithm on 500 audio samples of Ka (क) and Kha (ख) (350 training samples and 150 testing samples) using different data aggregation methods and different values of K in KNN, we got accurate results of matching phoneme in percentage. These results are shown in the table below:

Table 2 comparison of all data aggregation method with different values of k in KNN for % accuracy

Method name	K Value (in %)				
	K=1	K=3	K=5	K=7	K=9
Weighted Average value	80	83	83	83	83
Mean Value	85	90	86	85	84
RMS Value	85	86	85	84	84
Mid-point value of weighted average range	89	90	87	87	86

After analyzing above Table (5.1), for K=1, we get 80% accurate results of matching phoneme by weighted average value method, 85% accurate results of matching phoneme by mean value method and by RMS value method, 89% accurate results of matching phoneme by mid-point value of weighted average range method. For K=3, we got 83% accurate results of matching phoneme by weighted average value method, 90% accurate results of matching phoneme by mean value method, 86% accurate results of matching phoneme by RMS value method, 90% accurate results of matching phoneme by midpoint value of weighted average range method. For K=5, we got 83% accurate results of matching phoneme by weighted average value method, 86% accurate results of matching phoneme by mean value method, 85% accurate results of matching phoneme by RMS value method, 87% accurate results of matching phoneme by midpoint value of weighted average range method. For K=7, we got 83% accurate results of matching phoneme by weighted average value method, 85% accurate results of matching phoneme by mean value method, 84% accurate results of matching phoneme by RMS value method, 87% accurate results of matching phoneme by midpoint value of weighted average range method. For K=9, we got 83% accurate results of matching phoneme by weighted average value method, 84% accurate results of matching phoneme by mean value method, 84% accurate results of matching phoneme by RMS value method, 86% accurate results of matching phoneme by midpoint value of weighted average range method. 40 We can see that, for K=3, we get best results of matching phoneme for mean value method and mid-point value of weighted average range method.

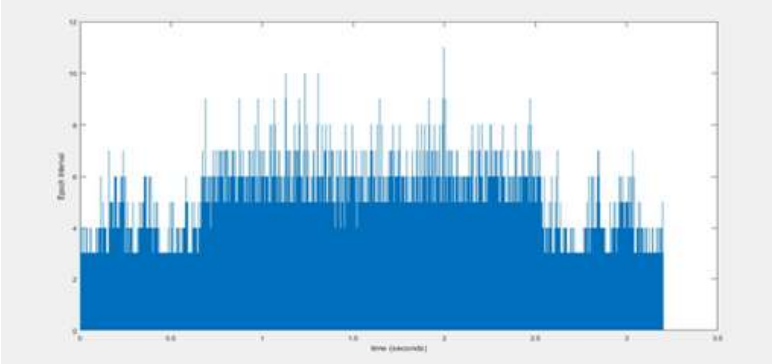

Fig.5 Epoch interval

References

S. R. Mahadeva Prasanna, Suryakanth V. Gangashetty, and B. Yegnanarayana Significance of vowels onset point for speech analysis, November 2002.

Roe1 Smits and B. Yegnanarayana, Determination of Instants of Significant Excitation in Speech Using Group Delay Function, IEEE Transactions on speech and audio processing. vol. 3, no. 5. September 1995.

K. Sri Rama Murty and B. Yegnanarayana Epoch Extraction From Speech Signals, IEEE Transactions on audio, speech, and language processing, vol. 16, no. 8, November 2008.

B. Yegnanarayana and Suryakanth V. Gangashetty Epoch-based analysis of speech signals, Sadhana Vol. 36, Part 5, October 2011, pp. 651–697.

Peter L. Chu and David G. Messerschmitt A frequency weighted Itakura-Saito spectral distance Measure IEEE Transactions on Acoustic, Speech and Signal processing, vol. ASSP-30, no. 4, August 1982.

Annamaria Mesaros Singing voice identification and lyrics transcription for music information retrieval, 2013.

Thomas Drugman, Mark Thomas, Jon Gudnason, Patrick Naylor, Thierry Dutoit Detection of Glottal Closure Instants from Speech Signals:Quantitative Review, 2012.

Dr. Kavita Thakur, Dr. A.S. Zadgaonkar A new technique for non-invasive assessment of volume gain of foetus using formant frequencies, Acta Ciencia vol.2, 2004.

Nivedita Deshpande, Kavita Thakur, A. S. Zadgaonkar Assessment of second degree heart block from speech analysis Journal of Ravishankar university-B,27,108-114(2014.

Nivedita Deshpande, Kavita Thakur, A. S. Zadgaonkar Determination of tachycardia from acoustical ECG, 2013. 44

Mr. K.V. R. Surya Prakash, Dr. A. S. Zadgaonkar, Dr. Kavita Thakur A new technique for assessment of mental status of mentally retarded people using prosodic parameter, J. Acous. Soc. Ind. vol.33, 2005.

S. Saraswathi Speech authentication based on audio watermarking International journal of information technology, vol. 16 no. 1, 2010.