Research Paper

Volume : 2 | Issue : 4 | Apr 2013 • ISSN No 2277 - 8179

Engineering

KEYWORDS : HMM, LID, MFCC, Training, LPC

Automatic Indian Language Identification

P. Vijay Kumar ECE Department, SRM University, Chennai, Tamil Nadu 603203, India

A. Raviteja ECE Department, SRM University, Chennai, Tamil Nadu 603203, India

ABSTRACT

Automatic Language Identification (ALID) has been for a long time an active research area with a large variety of applications in human-computer intelligent interaction. The most used technique in this field is the Hidden Markov Models (HMMs) which is a statistical and extremely powerful method. The HMM model parameters are crucial information in HMMoptimizing HMM parameters is still an important and challenging work in LID research area.Usually the Baum-Welch (B-W) Algorithm is used to calculatethe HMM model parameters. However, the B-W algorithm usesan initial random guess of the parameters, therefore afterconvergence the output tends to be close to this initial value of thealgorithm, which is not necessarily the global optimum of themodel parameters. In this paper, MFCC featurescombined with Baum-Welch is proposed; the idea is touse MFCC as input to HMM and to identify the language

1. Introduction

The basic goal of the language identification(LI) system is to accurately identify the language from the given speech sample. Language identification has numerous practical applications s:speechrecognition,speechtranslation,speech activated automated systems . Language Identification (LID) has drawn much attention recently, due to the challenge of multi-lingual speech recognition. To identify which language is spoken from a speech utterance, traditionally, an individual language model is created for each possible language, and the utterance is classified by measuring the Mel Frequency Cepstral Coefficients(MFCC) and each one of language models. This basic idea works well with a single feature but needs to be expanded to benefit from multiple features. Much effort has been spent, therefore, utilizing fusion techniques to integrate varied LID systems which capture discriminative information from different features . In this paper speaker dependent approach have been followed. language models can be formed by considering the Acoustic features of speech signal In this paper three languages samples were taken tamil,hindi,English.

Actual complete ALID systems follow three modeling schemes. They are Acoustic modeling, Prosodic and Lexical. Approaches to spoken language identification can be roughly divided into two groups: acoustic modeling where spectral features of different languages are modeled directly, and phonotactic modeling [2]

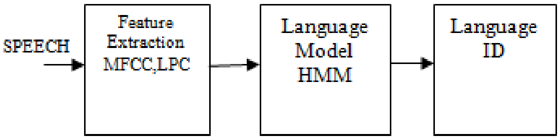

Fig 1:Basic block diagram of LID system

2Fig 1:Basic block diagram of LID system

In LID systems collection of data is the key factor in the precise language identification. In this module three languages from four different speakers were taken total 20 samples of recordings were taken. samples were taken from noise free environment hence there is no need of noise filter. Samples were recorded using quality acoustic sensor

a) Feature Extraction

The main purpose of the feature extraction process is to extract the most relevant information from the speech waveform and discard as much of the redundant information

as possible. In the case of language identification, an ideal parameterization technique would remove the speaker and noise dependent properties from the input speech and emphasize the characteristics of the speech waveforms that are most useful for discriminating between different languages. Some of the most widely used parameterization techniques are Mel Frequency Cepstral Coefficients (MFCCs)[3]

b) Mel-frequency Cepstrum Coefficient

The Mel-frequency Cepstrum Coefficient (MFCC) technique is often used to create the fingerprint of the sound files. The MFCC are based on the known variation of the human ear's critical bandwidth frequencies with filters spaced linearly at low frequencies and logarithmically at high frequencies used to capture the important characteristics of speech. Studies have shown that 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 algorithmic spacing above 1000 Hz. As a reference point, the pitch of a 1 kHz tone, 40dB above the perceptual hearing threshold, is defined as 1000 Mel's (Do 6). The following formula is used to compute the Mel's for a particular frequency:[4]

mel(f) = 2595*log10(1+ f / 700)...........(1)

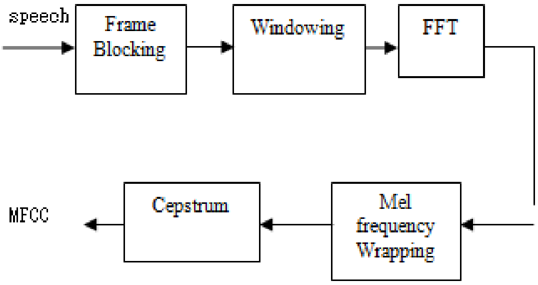

Fig 2:MFCC Block Diagram

A block diagram of the MFCC processes is shown in Figure 1. The function of each block was discussed in the previous report but just to summarize frame blocking sequence, the speech waveform is cropped to remove silence or acoustical interference that may be present in the beginning or eng of the sound file. The windowing block minimizes the discontinuities of the signal by tapering the beginning and end of each frame to zero. The FFT block converts each frame from the time domain to the frequency domain. In the Mel-frequency wrapping block, the signal is plotted against the Mel spectrum to mimic human hearing. In the final step, the Cestrum, the Mel-spectrum scale is converted back to standard frequency scale. This spectrum provides a good representation of the spectral properties of the signal which is key for representing and recognizing characteristics of the speaker. After the fingerprint is created, you will have will is also referred to as an acoustic vector. This vector is the one which was referred to in the earlier section. This vector will be stored as a reference in the database. When an unknown sound file is imported intoMatLab, a fingerprint will be created of it also and its resultant vector will be compared against those in the database, again using the Euclidian distance technique, and a suitable match will be determined. This process is as referred to as feature matching[1]

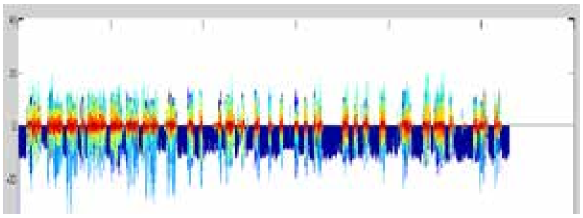

Fig.3:MFCC plot in MATLAB

C) Linear predictive coding (lpc)
Linear prediction is a method for signal source modeling dominant in speech signal processing and having wide application in other areas. Linear Predictive Coding (LPC) is one of the most powerful speech analysis techniques. The glottis (the space between the vocal cords) produces the sound, which is characterized by its intensity(loudness) and frequency (pitch). The vocal tract (the throat, the mouth and the nasal cavity) forms the tube, which is characterized by its resonance frequencies, which are called formants. The basic problem of the LPC system is to determine the formants from the speech signal. The solution of this problem is a difference equation, which expresses each sample of the signal as a linear combination of previous samples. Such an equation is called a linear predictor i.e. Linear Predictive Coding. The coefficients ofthedifferenceequation(the predictioncoefficients)characterize the formants. Therefore, the LPC system needs to estimate these coefficients. The estimation is made by minimizing the mean square error between the predicted signal and the actual signal.[8]

The basic idea behind the LPC model is that a given speech sample s(n)at time n, can be approximated as a linear combination of the past p speech samples (Rabiner

& Juang, 1993) such that

S(n)≈ a_1 s(n-1)+⋯+a_(n) s(n-p)……..(1)

Where the coefficients are a1, a2 , • • • an assumed to be constants over the speech analysis frame. Thequation(1)canbe converted to an equality by including an excitation term Gu(n),

$$S(n) \;=\; Gu(n) + \textstyle\sum_{i=1}^{p} a_i\, \mathrm{S(n-i)} \text{ ....(2)}$$

Where u(n) is normalized excitation and G is the gain of excitation.The relation between s(n) and u(n) is defmed as based on the speech production model

$$\mathrm{S(n) = Gu(n)} + \textstyle\sum_{i=1}^{p} a_k\, \mathrm{S(n)} \text{ ....(3)}$$

consider the linear combination of past speech samples as the estimate s(n), defined as,

$$\hat{\mathrm{s}} = \textstyle\sum_{k=1}^{p} a_k\ \mathrm{s(n\text{-}k)} \quad \text{........................(4)}$$

The predictor error e(n), is defined as

$$\mathrm{e(n) = s(n)} - \hat{\mathrm{s}}\mathrm{(n) = s(n)} - \textstyle\sum_{k=1}^{p} a_k\, \mathrm{s(n\text{-}k)} \text{ ......(5)}$$

And the error transfer function is,

$$\mathrm{A(z) = 1\text{-}} \textstyle\sum_{i=1}^{p} a_k\, Z^{-k} \text{ ....(6)}$$

The basic problem of linear prediction analysis is to determine the set of predictor coefficient directly from the speech signal so that the speech properties of the digital filter match those of the speech waveform within the analysis window. In the present study, LPC-based cepstral coefficients and phonetically important parameters are used as feature vectors. Cepstral weighted feature vector is obtained for each frame by block processing of continuous speech signals. The analog speech waveform is then sampled and quantized analog-to-digital converter. To spectrally flatten the signal, the speech signal has been subjected to the preemphasisprocedure through a first order digital filter whose transfer function has been given by

$$\mathrm{H(z) = 1 - az^{-1}}$$

Where

0 ≤ a ≤ 1.0 …………………(7)

Consecutive speech signal are taken as a single frame. To reduce the undesired effect of Gibbs phenomenon, the frames are multiplied by a windows function (Hamming window), which is given by (Proakis, & Manolakis, 2004)

$$\mathrm{w(n) = 0.54 - 0.46\cos}\left(\frac{2\pi n}{N-1}\right) \text{ ....(8)}$$

Where 0 ≤ n ≤ N-1
N is the number of sample in a block. Now, each frame of the windowed signal is next auto correlated to give

$$r_f\,\mathrm{(m)} = \textstyle\sum_{n=0}^{N-1-m} \hat{x}_f(n)\hat{x}_f(n+m) \text{ ....(9)}$$

Where m = 0,1,2, ... p.
The highest auto correlated value, p, is the order of the LPC analysis The LPCcepstralcoefficients,which are a set of values that have been found to be more robust, reliable feature set for speech recognition than the LPC coefficients (Rabiner&Juang, 1993). These coefficients are obtained recursively as follows.

$$\mathrm{C{=}ln}[\sigma^2]$$

where σ^2 is the gain term in the LPC Model

$$c_m \;=\; a_m + \textstyle\sum_{k=1}^{p} \left(\frac{k}{m}\right)\ c_{m-k}a_k \text{ ....(10)}$$

Where 1 ≤ m ≤ p

$$c_m \;=\; \textstyle\sum_{k=1}^{p} \left(\frac{k}{m}\right) c_{m-k}\, a_k \text{ ....(11)}$$

Where m > p

Equation (10) shows the computation of cepstral coefficients

C_{p+1} C_{p+},…….. C_p Generally, q > p is taken for cepstral representation

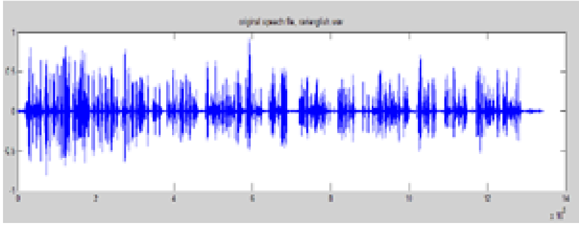

Fig4:Original Speech Signal

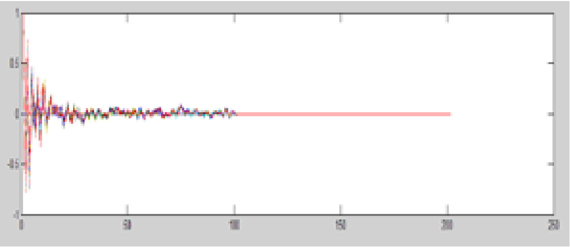

Fig 5:-LPC Mat Lab output

3.Hidden Markov Model

The HMM has been defined as a doubly stochastic process with an underlying stochastic process that is not observable. It can only be observed by using another set of stochastic process that can produce the sequence of observed symbols In practice, in order to build up transitions of HMM that represent the spoken words (that actually deals with signal), it is necessary to extract the parameters to represent the speech and their transcription. After that, these parameters are assigned HMM transitions which is called training of HMM. In addition, a given speech can be matched with a given training HMM to produce the speech hypotheses. The forward-backward algorithm i.e. the Baum-Welch algorithm is used for the training of HMM. The forward and Viterbi algorithms are used for decoding speech. These algorithms have been proven to solve the problems with HMM (training and decoding problem

Elements of hidden markov model

N- The number of hidden states

Q- The set of states Q={1,2,.....N}

A- The state-transition probability matrix

a_{ij}=P(q_{t+1}=j|q_t= i) 1≤ i, j ≤ N

B- Observation probability distribution

B_j(k) = P(O_t=k| q_t=j) 1 ≤ k ≤ MS

Π- The initial state distribution:

π_i = P(q_1= i) 1≤ i ≤ N

λ - The entire model:

λ = (A,B,π)

Training (Baum-Welch) algorithm:

The Baum-Welch algorithm can be used to train an HMM to model a set of sequence data. The algorithm starts with an Initial model and iteratively updates it until convergence. The algorithm is guaranteed to converge to an HMM that locally maximizes the likelihood (the probability of the training data given the model). Since the Baum-Welch algorithm is a local iterative method, the resulting HMM and the number of required iterations depend heavily on the initial model To calculate the probability (likelihood) P(X/Ø) of the observation sequence X= {} , given the HMM , the most intuitive way is to sum up the probabilities of all possible state sequences

$$P(X \mid Ø) = \textstyle\sum_{all\ s} P(s \mid Ø)\, P(X \mid S, Ø) \text{..........(12)}$$

For any given state sequence, we start from initial state with probability We take a transition from to with probability and generate the observation with probability until we reach the last transition

- Initialize the parameters to some values
- Calculate "forward-backward" probabilities based on the current parameters
- Use the forward-backward probabilities to estimate
- the expected frequencies

– Expected number of transitions from state i (to state j)
– Expected number of being in state j (and observing)
– Expected number of starting in state j

- Use the expected frequencies to estimate the Parameters
- Repeat this process until the parameters converge

Forward-backward probability
Forward probability

- At time t, the probability that we're in state i the observation thus far has been „... ....(13)

Backward probability
– At time t and we're in state i, the probability that

- the observation that follows will be ...

$$\beta_t(i) = P(O_{t+1} \ldots\ldots O_T \mid S_t = i, \lambda) \text{.............(14)}$$

Transitional probability

$$\xi_t(i,j) = \frac{\bar{\alpha_t}(i)\, a_{i,j} b_j(O_{t+1})\beta_{t+1}(j)}{\sum_{k=1}^{N} \alpha_T(k)} \text{ ....(15)}$$

$$\hat{a}_{i,j} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1}\sum_{j=1}^{N} \xi_t(i,j)} \text{ ....(16)}$$

Emission probabilities

- From the forward probability

$$\gamma_t(i) = \frac{\alpha_t(i).\beta_t(i)}{\sum_{j=1}^{N} \alpha_T(j)} \text{ ....(17)}$$

- Emission probability

$$\hat{b}_i(k) = \frac{\sum_{t=1}^{T} \delta(O_t, v_k)\, \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)} \text{ ....(18)}$$

$$\delta(O_t, v_k) = 1, \text{ if } O_t = v_k$$

Decoding (Viterbi algorithm)
There are several paths through the hidden states (H and L) that lead to the given sequence, but they do not have the same probability. The Viterbi algorithm is a dynamical programming algorithm that allows us to compute the most probable path. for the calculations, it is convenient to use the log of the probabilities (rather than the probabilities themselves). Indeed, this allows us to compute sums instead of products, which is more efficient and accurate

Steps:-
1. Choose the most likely path
2. Find the path that maximizes the likelihood
3. Initialization:

$$\delta_1(i) = \pi_i b_i(O_1), \quad 1 \le i \le N$$
$$\psi_1(i) = 0$$

4. Recursion

$$\delta_t(j) = \max_{1 \le i \le N}[\delta_{t-1}(i) a_{ij}] b_j(O_t) \text{ ....(19)}$$

$$\psi_t(j) = \arg\max_{1 \le i \le N}[\delta_{t-1}(i) a_{ij}] \text{ ....(20)}$$

5. Termination

$$P^* = \max_{1 \le i \le N}[\delta_T(i)] \text{ ....(21)}$$
$$q^*_T = arg\ \max_{1 \le i \le N}[\delta_T(i)] \text{ ....(22)}$$

6. Back tracking

$$q^*_t = \psi_{t+1}(q^*_{t+1}) \text{ ....(23)}$$

t= T-1.T-2..........1

4. Results And Discussions

Depending on the analysis made in this study onCepstral features and formant frequencies of different Languages the following observations were made: Significant variation of Cepstral coefficients are observed among the languages as shown in Table 1. The cepstral variation is found to be maximum

Table1:-MFCC and LPC maximum values

Language	MFCC Max	LPC Max
TAMIL	18.14	1.4
HINDI	20.01	3.2
ENGLISH	17.14	1

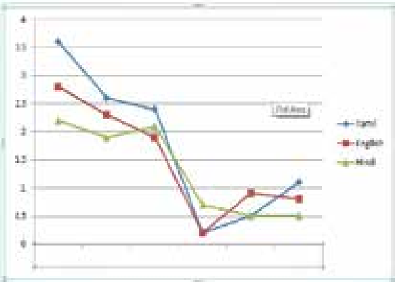

Fig6:Graph showing variations in output

Table2:-Showing successfully identified samples

No. Of Samples	Success Rate (%)
Tamil	86
Hindi	90
English	92

The MFCC, LPC coefficients for different languages were taken and variations had been observed. These MFCC were input for the construction of language model. The maximum likelihood estimation for model using Baum Welch and Viterbi algorithms were used and the language will be identified

REFERENCE

[1]. Md. Rashidul Hasan, Mustafa Jamil, Md. GolRabbaSaifur Rahman,"Speaker Identification Using Mel FrequencyCepstralCoefficients"3rdInternational Conference on Electrical & ComputeEngineeringICECE 2004, 28-30 December 2004, Dhaka, Bangladesh [2]. Herman Kamper and Thomas NieslerTechnical Report SU-EE-1201"A literature review of language, dialect and accentidentification"DigitalSignalProcessingLaboratoryDepartment of Electrical and Electronic Engineering Stellenbosch University, South Africa19 January 2012. [3]. Eliathamby Ambikairajah Haizhou Li, Liang Wang, Bo Yin, and Vidhyasaharan Sethu "Language Identification: A Tutorial" IEEE Circuits And Systems Magazine Second Quarter 2011 [4]. Jamal Price, Sophomore Student" Design Of An Automatic Speech Recognition SystemUsingMatlab"Department of Engineering and Aviation Sciences University of Maryland Eastern Shore Princess Anne, MD 21853Chesapeake Information Based Aeronautics Consortium August 2005 [5]. Pejman Mowlaee, Rahim Saeidi, Mads Græsbøll Christensen, Tomi Kinnunen, Pasi Franti Soren Holdt Jensen "A Joint Approach for Single-Channel SpeakerIdentification and Speech Separation", Member, IEEE [6]. MohamedFEZARI1,MohamedSeghirBoumaza"VoiceCommandSystemBasedOnPipeliningClassifiersGMMHMM"2012 International Conference on Information Technology and e-1.Laboratory of Automatic and Signals, Annaba, BP.12,Annaba, 23000, ALGERIA [7]. L. R. Rabiner, "A tutorial on hiddenMarkovmodels and selected applications in speech recognition," Proc. IEEE, vol. 77, no. 2, pp. 257–286,1989 [8]. Monoj Kr. Singha, Jogen Boro, Biren Sarma," LPC Analysis of Vowels and Formant Analysis of Some Typical CV and VC Type of Words in BodoLanguage" 2012 IEEE