Int J Speech Technol
DOI 10.1007/s10772-014-9250-8

# Statistical analysis of features and classification of *alphasyllabary* sounds in *Kannada* language

**Sarika Hegde · K. K. Achary · Surendra Shetty**

Received: 26 February 2014 / Accepted: 15 August 2014


**Abstract** Automatic speech recognition (ASR) for a given audio file is a challenging task due to the variations in the type of speech input. Variations may be the environment, language spoken, emotions of the speaker, age/gender of speaker etc. The two main steps in ASR are converting the audio file into features and classifying it appropriately. Basic unit of speech sound is phoneme and the list of such phoneme is language dependent. In Indian languages, basic unit of language is known as *Akshara* i.e the alphabet. It is known to be an alphasyllabary unit. In our work, we have analyzed the behavior of the acoustic features like, Mel frequency cepstral coefficients and linear predictive coding for various *aksharas* using techniques like, visualization, probability density function (pdf), Q–Q plot and F-ratio. The classifiers, support vector machine (SVM) and hidden Markov model (HMM) are used for classifying the recorded audio into corresponding *aksharas*. We have also compared the classification performance of HMM and SVM.

**Keywords** Hidden Markov model · Support vector machine · Mel frequency cepstral coefficients (MFCC) · Linear predictive coding (LPC) · Vector quantization (VQ) · Alphasyllabary · Statistical analysis · Kannada language · Speech recognition

S. Hegde (✉) · S. Shetty
Master of Computer Applications, NMAM Institute of Technology, Nitte, Udupi District, Karnataka 574110, India
e-mail: sarika.hegde@yahoo.in

S. Shetty
e-mail: hsshetty4u@yahoo.com

K. K. Achary
Department of Statistics, Mangalore University, Mangalore 574199, India
e-mail: kka1953@gmail.com

*Present Address:*
K. K. Achary
Yenepoya Research Centre, Yenepoya University, Mangalore 575018, India

## 1 Introduction

Automatic speech recognition (ASR) can be done either by recognizing each word individually (Isolated word recognition—IWR) or recognizing the sub units of a sentence continuously (continuous speech recognition—CSR) (Rabiner and Juang 1993). For speech recognition using IWR system, ASR system must be trained with large number of words and as the number of words increases, number of patterns to be classified increases. In case of continuous speech recognition (CSR), each smaller segments of the speech like phoneme or syllable are identified and speech recognition is done by combining the classifiers of these sub units. The difficulty in such CSR is to segment the sentence into phonemes or syllables.

Automatic speech recognition involves two major steps; feature extraction and classification. In feature extraction step, the given speech sound is preprocessed and appropriate features are computed using the digital signal processing techniques. In the next step, machine learning techniques are used for constructing the classifiers which learn about the speech using these features. These classifiers are then used for identifying the word/phoneme/syllable of any speech utterance.

## 2 Review of previous works

Speech data may have noise, silence etc; and these should be eliminated using speech enhancement techniques. These

Published online: 29 August 2014

techniques mainly deal with noise reduction in the speech signals, detecting the presence/absence of voice (voice activity detection) or how signal can be segmented into different units; (the unit may be phoneme, syllable or word). Some of the major works relating to noise and speech-end point detection can be found in (Rahim and Juang 1996; Ephraim 1992; Lamel et al. 1981; Sohn et al. 1999). Many authors study speech at phonetic level and impact of phonetics in speech understanding (Sarah 2003). Extensive details of research done on segmentation of speech into syllable like models can be found in the research works of (Lakshmi and Murthy 2006; Thangarajan et al. 2009). Research has also been focused on identifying the feature representation best suited to categorize the speech into different units. MFCC and LPC features have been identified as the most successful feature representation for the speech signal (Davis and Mermelstien 1980). The advanced pattern recognition techniques like SVM, ANN, Bayes classifier, HMM, hybrid models etc have been found suitable for ASR (Chien et al. 2007; Jiang et al. 2006; Axelrod and Maison 2004). Recently Patro et al. (2007) have tested the relative performance of different features like, Sinusoidal Frequency Features (SFF), Sinusoidal Amplitude Features (SAF), Cepstral Coefficients (CC) and Mel frequency cepstral coefficients (MFCC), in characterization of stressed conditions in a speech signal. They have used different techniques like, F-ratio, K-S test and Vector Quantization for testing the characteristics of stressed conditions in speech signal.

Most of the researchers have used 'English' language for the purpose of study and mainly focus on databases which are easily available. Apart from English there are many other European and Asian languages which have been considered by the researchers for speech recognition study. Speech recognition researches on Indian languages have started very recently. We can find Speech Recognition research works on *Hindi, Bengali, Panjabi, Marathi, Tamil, Telugu* and *Malayalam* languages. However significant works have been done in *Tamil* language (Lakshmi and Murthy 2006; Thangarajan and Natarajan 2008; Thangarajan et al. 2009). Kaur and Singh (2010) have worked on segmentation of continuous speech into syllable like units for *Punjabi* language. *Kannada* spoken digits are recorded and classified using support vector machine (SVM) in Hegde et al. (2012). A new feature called Zero Crossing Interval Distribution computed on the time domain signal has been investigated and studied by Kumar and Lajish (2013). This study has been carried over five *Malayali* language vowels. Das et al. (2013) have studied age related variations of speech characteristics of two age groups, in the Bengali language.

Based on the review, we can observe that lot of innovations are done in western languages, other Asian languages and Indian languages like, *Hindi, Bengali, Panjabi, Marathi, Tamil, Telugu* and *Malayalam*. The innovations are mainly in terms of feature extractions and classification that could be used to improve the speech recognition task. We can convert the speech sounds of any language into list of phoneme or syllables using the existing techniques available. But the important question here is that of further processing of these units to generate or understand the words/ sentences spoken. In this regard, it becomes important to focus on language specific aspects of speech recognition. In our work, we focused on *Kannada* language and the analysis of the signal features of these sounds using statistical techniques. Further study in such direction would help recognition and understanding of the words/sentences of *Kannada* language, which is the aim of our study.

Brief introduction to the problem of ASR and the survey on related previous works is already covered in first and second sections. The organization of the paper in the following sections is as below. In the third section, we have briefed about the objective of our work. The feature extraction techniques and the exploratory analysis of features based on statistical techniques are explained in fourth section. Fifth section describes the HMM and GMM techniques. In the sixth section, we discuss the experimentation and analysis of results. In the last section, conclusive remarks are given.

## 3 Focus of study

We have mainly used speech sounds of vowels and consonants (including unstructured consonants) of *Kannada* language for our study. *Kannada* is an alphasyllabary language where each *akshara*/alphabet can be considered to have syllable like structure, but at the same time, it can be broken into sequence of consonant and vowels (Nag et al. 2010). According to Nag et al. (2010), "*Akshara* writing system has similarity with syllabary, but also have alphabet like features. *Aksharamala* literally means 'the garland of *akshara*' and it is also possible to pull apart *akshara* into smaller units within the syllable. Thus the Hindi *akshara* कु (/ku/) can be deconstructed into the tinier sound units of क+ उ (/k/ + /u/). These smaller consonant and vowel units within /ku/ are equivalent to the sounds that the letters 'k' and 'u' represent in the English alphabet". The *aksharamala* of *Kannada* language consists of a set of *thirteen* vowels and *thirty four* consonants. Any word can be formed with combination of these *akshara*. While pronouncing a word regional accent plays the role which decides the merging of the ending boundary of one *akshara* with the beginning of another *akshara*. The *akshara* system typically comprises three distinct types of symbols. First we have the consonants with an inherent vowel (/Ca/), the second type are consonants with other vowels (/CV/) and the third are the consonant clusters (/CCV/) with possibility of more than two consonants in the cluster (Nag et al. 2010).

We have collected the audio dataset of vowels and consonants of *Kannada* language. For each of these, we extract the MFCC and LPC features. An exploratory analysis of features has been analyzed using various techniques namely, Visualization, Probability density function (pdf), QQ Plot and F-ratio. Analysis is carried out to find the suitability of a feature to represent an alphabet class effectively. We have applied hidden Markov model (HMM) and Support Vector Machine (SVM) for designing the classifier model. These classifier models are evaluated based on the percentage of accuracy of correct classification of sounds.

## 4 Feature extraction techniques and analysis

### 4.1 Dataset

Each alphabet is considered as a pattern for the classification. For the experiment, we have used 5 vowels and 10 consonants (includes 5 structured and 5 unstructured consonants) of *Kannada* language. The set of structured consonants in *Kannada* language has five groups and we have selected first one from each group. Among unstructured consonants, we selected the first five for the study. The list of vowels and consonants are listed in the following table with their corresponding Unicode name (Tables 1, 2). Forty patterns of each alphabet are recorded with the voice of single female *Kannada* speaker, creating the database constituting around 600 audio clips (Table 1).

### 4.2 Feature extraction techniques

We have used the two most popular techniques for extracting features from speech audio file viz. Mel frequency cepstral coefficients (MFCC) and Linear Predictive Coding (LPC). We briefly describe these techniques in the following discussion.

#### *4.2.1 Mel frequency cepstral coefficients (MFCC)*

The MFCC is a feature extracted by applying more than one Fourier Transform sequentially to the original signal. The first step is preprocessing which consists of framing and windowing of the signal. Framing is the process of breaking the set of sample observations of an entire audio file into smaller chunks called as frame (McLoughlin 2009). We have used a frame size of 30ms which is normally used in speech recognition applications. For each of the frame $f_i$ of size $N$, DFT coefficients are calculated by applying equation 4.1.

$$X_i\,[m] = \sum_{n=0}^{N} x\,[n]\, e^{-imn2\pi/N}, \quad 1 \le m \le M \tag{4.1}$$

The resulting value $X_i\,[m]$ is a complex number and the power spectrum for this is computed as,

$$P_i\,[m] = \frac{1}{N}\,|X_i\,[m]|^2\,, \tag{4.2}$$

The power spectrum is then transformed to mel frequency scale by using a filter bank consisting of triangular filters, spaced uniformly on the mel scale. To start with, the lower and higher values of frequencies are converted into mel scale. An approximate conversion between a frequency value in Hertz ($f$) and $mel$ is given by:

$$mel\,(f) = 2595 log_{10}\left(1 + \frac{f}{700}\right) \tag{4.3}$$

Based on the number of filters to be designed, the intermediate mel frequencies are computed. Each of the mel frequency is then converted back into normal frequency with the inversion formula. By doing this we get the list of frequencies with mel-scale spacing. For each of these frequency responses, a triangular filter is created and the series of such filters is called as mel filter bank. Each triangular filter is then multiplied with power spectrum $P_i\,[m]$ and the coefficients are added up, which will be an indication of how much energy is within each filter. Finally, the cepstral coefficients are calculated from these calculated energy values of filterbank by applying Discrete Cosine Transform (DCT) of the logarithm of the filter-bank energy. This is given by,

$$C_i = \sum_{k=1}^{K} (log S_k) \cdot cos\left(\frac{i\pi}{K}\left(k - \frac{1}{2}\right)\right)$$
$$k = 1, 2, \ldots, K \;\&\; i = 1, 2 \ldots, L \tag{4.4}$$

$'L'$ is the number of MFCC coefficients considered, $C_i$ is the $i^{th}$ MFCC coefficient, $S_k$ is the energy of $k^{th}$ filterbank channel (i.e. the sum of the power spectrum bins on that channel), $K$ is the number of filterbanks. A more detailed description of the mel-frequency cepstral coefficients can be found in Rabiner and Juang (1993).

#### *4.2.2 Linear predictive coding (LPC)*

Linear predictive coding is one of the most commonly used speech synthesis and speech analysis tool. Initially, this technique was being used as speech coding technique for speech synthesis. But now it is also being used in speech analysis technique for speech recognition (Atal and Hanauer 1971; Atal and Rabiner 1976; Kinsner and Peters 1988; Ganga Shetty and Yagnanarayana 2001). The basic idea behind linear prediction is that the next signal sample observation is predicted from a weighted sum of $p$ previous sample observations, given as follows: (Peltonen et al. 2002)

$$\hat{s}\,(n) = \sum_{i=1}^{P} a_i\, s\,(n - i) \tag{4.5}$$

**Table 1** *Kannada* and English representation of list of vowels and consonants used in our study

| Vowels | Consonants | |
|---|---|---|
| | Structured | Unstructured |
| ಅ (/a/) | ಕ್ (/k/) | ಯ್ (/y/) |
| ಇ (/i/) | ಚ್ (/ch/) | ರ್ (/r/) |
| ಉ (/u/) | ಟ (/tt/) | ಲ್ (/l/) |
| ಎ (/e/) | ತ್ (/t/) | ಸ್ (/s/) |
| ಒ (/o/) | ಪ್ (/p/) | ಹ್ (/h/) |

where the set $\{a_i\}$ is the set of prediction coefficients and $s(n-i)$ is a sample observation at time instant $(n-i)$. In other words, each sample observation of a signal is modeled as a linear combination of previous sample observations. The prediction coefficients are determined by minimizing the mean squared error between the actual sample observation and the predicted sample observation. The prediction error signal is given by (McLoughlin 2009; Peltonen et al. 2002)

$$e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{i=1}^{p} a_i s(n-i) \tag{4.6}$$

and the squared error over all the $n$ sample observations:

$$E = \sum_{n} [e(n)]^2 = \sum_{n} \left[ s(n) - \sum_{i=1}^{p} a_i s(n-i) \right]^2 \tag{4.7}$$

The technique described above computes '$p$' LPC coefficients for a given frame of size $n$. The set of these $'p'$ coefficients are used as feature vector.

#### *4.2.3 Vector quantization (VQ) technique*

For each of the recorded alphabet, feature extraction technique is applied followed by Vector Quantization technique. A vector quantizer maps $d$-dimensional feature vectors in the vector space $R^d$ into a finite set of vectors $V = \{v_i : i = 1, 2, \ldots, K\}$. Each vector $v_i$ *is* called a code vector or a codeword and the set of all the codeword is called a codebook. In this technique, k-means clustering algorithm is applied to find the 'k' centers in the feature vectors. The sequence of feature vectors from each audio is grouped into 'k' clusters. The means from each cluster together forms the vector codeword (Linde et al. 1980). This is repeated for all the patterns available and this vector codebook is used for designing the classifier model.

We have used this technique to compress the variable length sequence of MFCC and LPC feature vectors of each audio file to a set of fixed length vector codebook. Traditionally, vector of ten codewords is generated in VQ process. But we have used only five codewords since some of the audio sounds of consonants are too short to generate ten codewords. Using this process, our vowel dataset consists of 200 vector codebooks of 5 categories and the consonant dataset consists of 400 vector codebooks of 10 types.

### 4.3 Exploratory analysis of feature sets

We have analyzed the computed feature sets for understanding their capability to discriminate one alphabet with another. Three techniques are used for this purpose and separate analysis is done for vowels and consonants. In the first technique which is a visualization technique, the first three coefficients of features are plotted over 3-d graph. We have also plotted the probability distributions of features using histogram technique. It helps to identify the tendency of grouping of feature vectors of same alphabet class visually. In the second technique, we computed the QQ plot to compare the distributions and also to check normality of data. In the third technique we have computed F-ratio value for each alphabet class to analyze the capability of features to distinguish the alphabets.

In all the graphs, vowels are plotted in the order /a/, /i/, /u/, /e/, and /o/. The consonants are potted in the order /k/, /tt/, /p/, /t/, /y/, /r/, /l/, /v/, /s/, and /ch/.

#### *4.3.1 Visualization of feature sets*

Here we plot the first three feature coefficients from feature vector in 3-d plot. This is repeated for all the alphabet classes. The graphs display the similarity and difference between the feature sets of various alphabets. It helps to identify the tendency of grouping of feature vectors of same alphabet class.

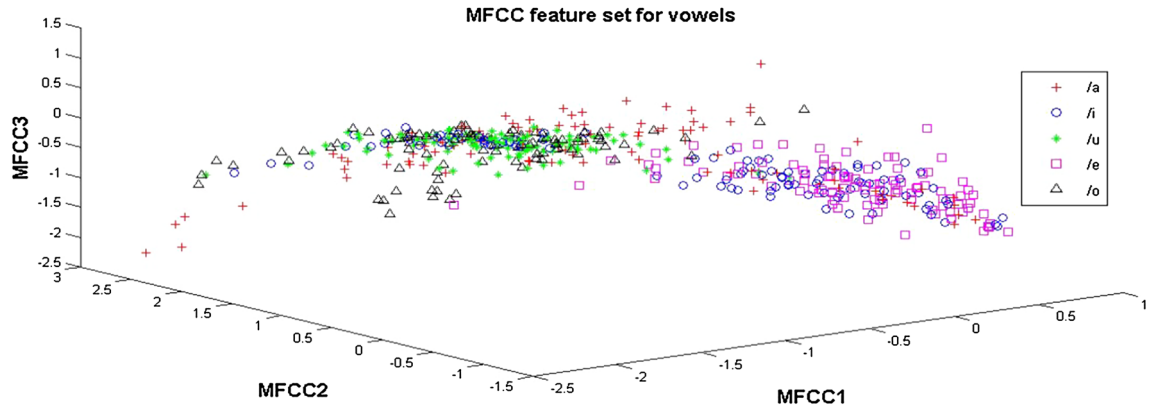


**Fig. 1** MFCC feature set for *five* vowels

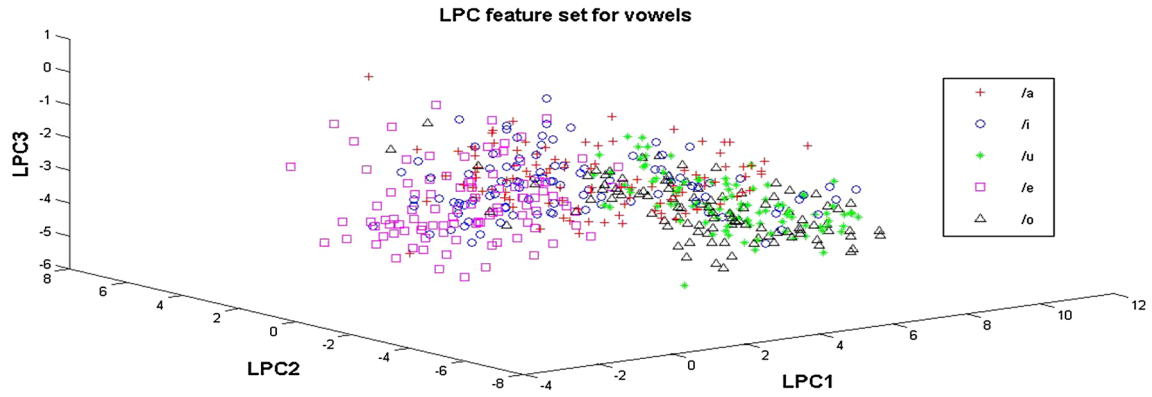


**Fig. 2** LPC feature set for 5 vowels

Different color/symbols are used for different alphabet labels. For each alphabet, 100 sample observations are selected randomly and plotted. It gives an idea about natural grouping of feature sets which can be observed visually. The following two graphs (Figs. 1, 2) show the visualization of features for vowels dataset for LPC and MFCC features respectively.

In both of the above graphs, the sample observations of same alphabet patterns are somewhat grouped nearer to each other. We can observe that some of the alphabet classes like, /e/ is forming much clearer group than other in both the cases. The following two graphs (Figs. 3, 4) show the visualization of 10 consonant dataset for MFCC and LPC features respectively.

Even in this graph, we can observe that some of the alphabet classes like, /ch/ and /s/ distinguish themselves more clearly than other alphabet classes. In all the graphs there is an overlap of features of different classes. LPC graphs are better and clear for understanding the grouping of alphabet than MFCC feature.

In the next step, we have generated the frequency curve for each of the vowels and consonants dataset. Here we have used average of 12 coefficients of LPC and MFCC respectively as the feature. The graphs in Fig. 5, show the frequency curves of the mean values of MFCC and LPC coefficients of vowels. The curves show similarity with normal/symmetric pdf.

The graphs in Fig. 6, show the frequency curves of the mean values of MFCC and LPC coefficients of consonants.

From the above graphs, we can observe that the frequency distribution pattern for MFCC feature is clearer than that of LPC feature. This is true for both vowels and consonants. But there is an overlap in the pdf for both MFCC and LPC. Among consonants, we can observe in Fig. 6a, that the consonant /tt/ has a different pattern for probability distribution compared to others.

### *4.3.2 QQ plot*

QQ plot can be used to compare the two distributions and also can be used to compare the order statistics of the data against theoretical distribution. We use this technique to assess the normality of the MFCC and LPC features, since we use GMM in hidden Markov model classifier. We plot the data $x_i$ in one axis and in the other axis it plots,

$$F^{-1}\left(\frac{i-0.5}{n}\right) \tag{4.8}$$

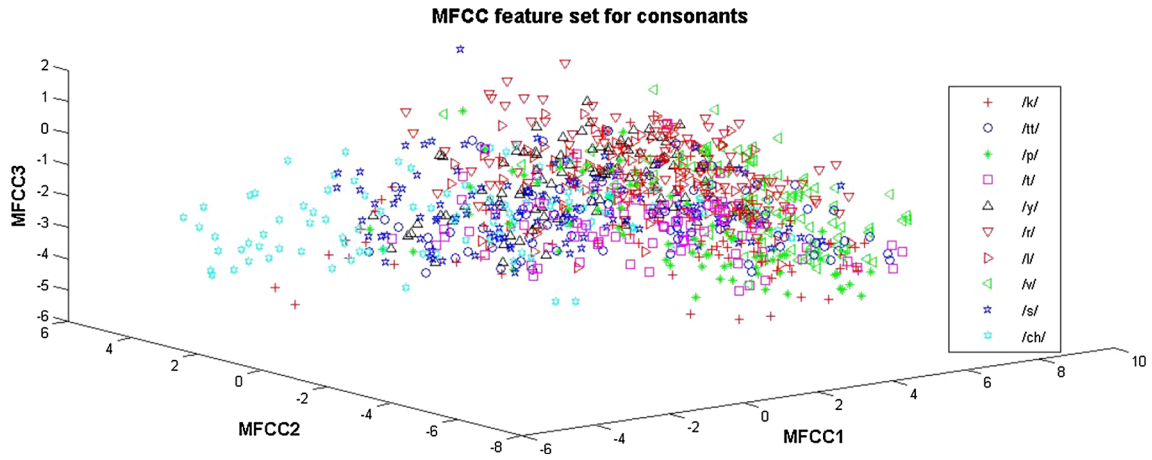


**Fig. 3** MFCC feature set for *ten* consonants

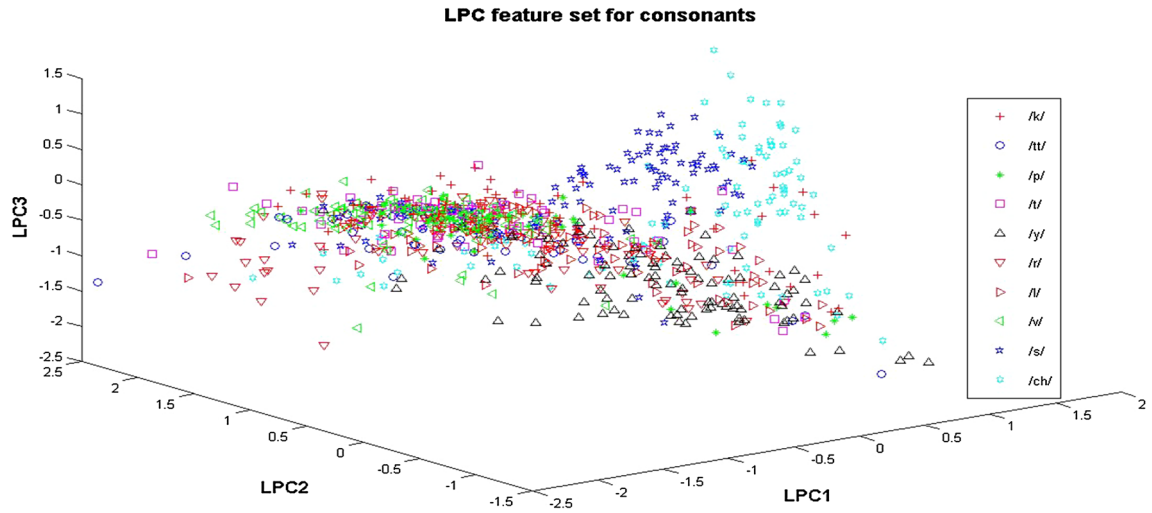


**Fig. 4** LPC feature set for *ten* consonants

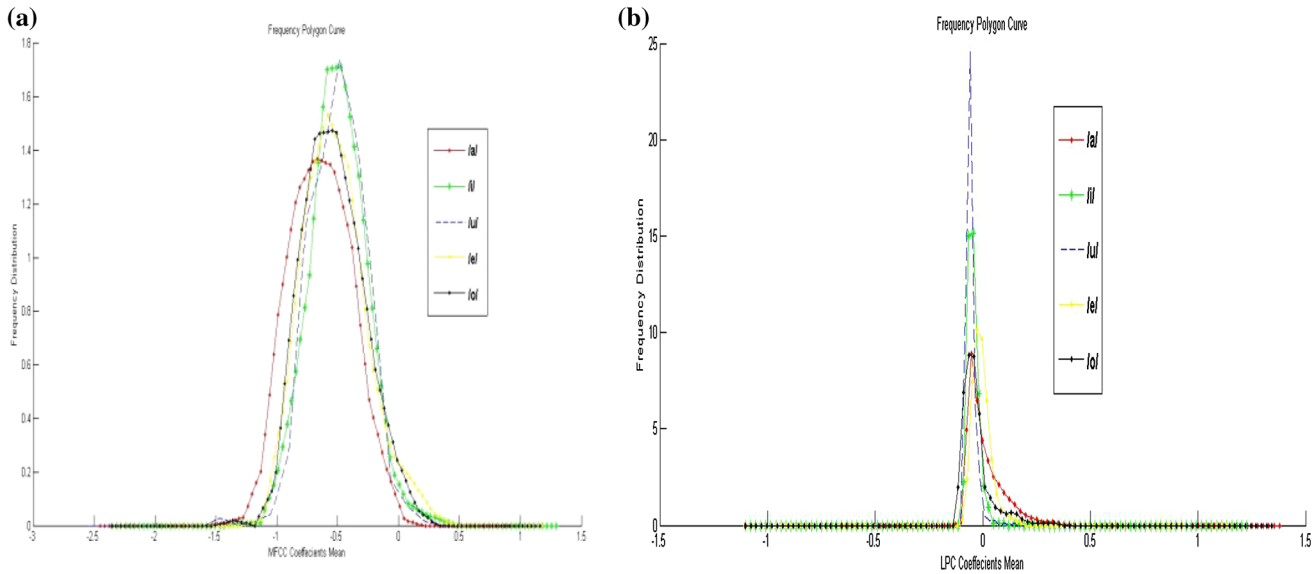


**Fig. 5** **a** PDF for MFCC feature of 5 vowels and **b** PDF for LPC feature

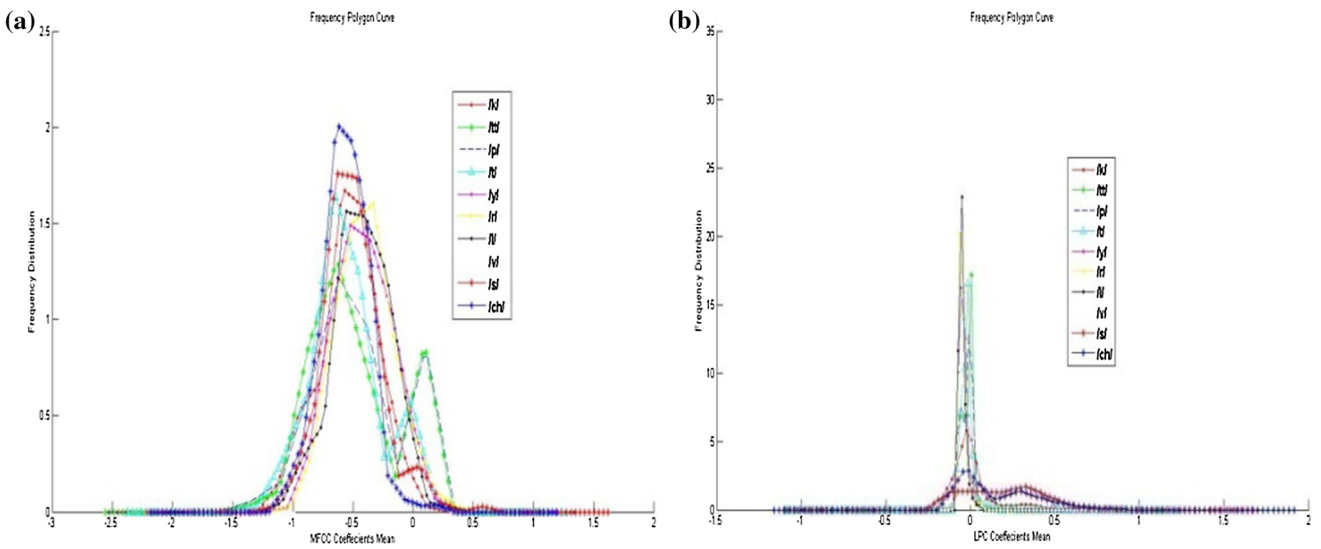


**Fig. 6** **a** PDF for MFCC feature of 10 consonants and **b**PDF for LPC feature

Where $F^{-1}$ denotes the inverse of the cumulative distribution function for the hypothesized distribution (Johnson and Wichern 2007). Here we have assumed normal distribution for the alphabet classes and plotted QQ plots for each alphabet class. Graph in Fig. 7 shows the QQ plot for vowels with MFCC and LPC features respectively.

Graphs in Fig. 8a, b show the QQ plot for consonants with MFCC and LPC features respectively. We can observe that MFCC feature shows better normality property compared to LPC feature for both vowels and consonants.

#### *4.3.3 F-ratio (Fisher's ratio)*

A set of feature measurements would be effective in discriminating between alphabets if the distributions of different alphabets are concentrated at widely different locations in the parameter space. For a feature value, a good measure of effectiveness would be the analysis of variance. This can be done by computing ratio of inter alphabet to intra-alphabet variance, often referred to as the F-ratio (Atal 1976; Patro et al. 2007; Johnson and Wichern 2007). F-ratio is given as,

$$F = \frac{\text{between aphabet group variance}}{\text{within alphabet group variance}}$$

Higher F-ratio value is better for good classification (Atal 1976; Patro et al. 2007). We have computed the F-ratio for vowel and consonant dataset separately for MFCC and LPC. The following table summarizes the F-ratio value for different datasets (Table 2).

We can observe that, F-ratio value for MFCC features for both Vowels and Consonants is found to be high compared to LPC. That means that MFCC has more discriminating values for different classes.

## 5 Classifier algorithms

### 5.1 Hidden Markov model

A hidden Markov model (HMM) is a statistical model in which the system being modeled is assumed to be a Markov chain with unobserved (*hidden*) states. A Markov chain is a stochastic process in which the future state of the process is based solely on its present state. HMM is a very common technique in classification, especially in the case of sequential data processes such as speech, music and text. HMMs are used to specify a joint probability distribution over hidden state sequences $S = \{s_1, s_2 \cdots, s_T\}$ and observed output sequences $X = \{x_1, x_2 \cdots, x_T\}$. The logarithm of this joint distribution is given by:

$$log\,P(X, S) = \sum \left[\log P(s_t|s_{t-1}) + log\,P(x_t|s_t)\right] \qquad (5.1)$$

Here the hidden states $s_t$ and observed outputs $x_t$ denote alphabet labels and corresponding feature vectors respectively. The distributions $P(x_t|s_t)$ are typically modeled by multivariate Gaussian mixture models (GMMs):

$$P\left(x_t|s_j\right) = \sum_{m=1}^{M} w_{jm} N(x_t; \mu_{jm}, \Sigma_{jm}) \qquad (5.2)$$

Where, $N(x_t; \mu, \Sigma)$ denotes the Gaussian distribution with mean vector $\mu$ and covariance matrix $\Sigma$, and M denotes the number of mixture components per GMM. The mixture

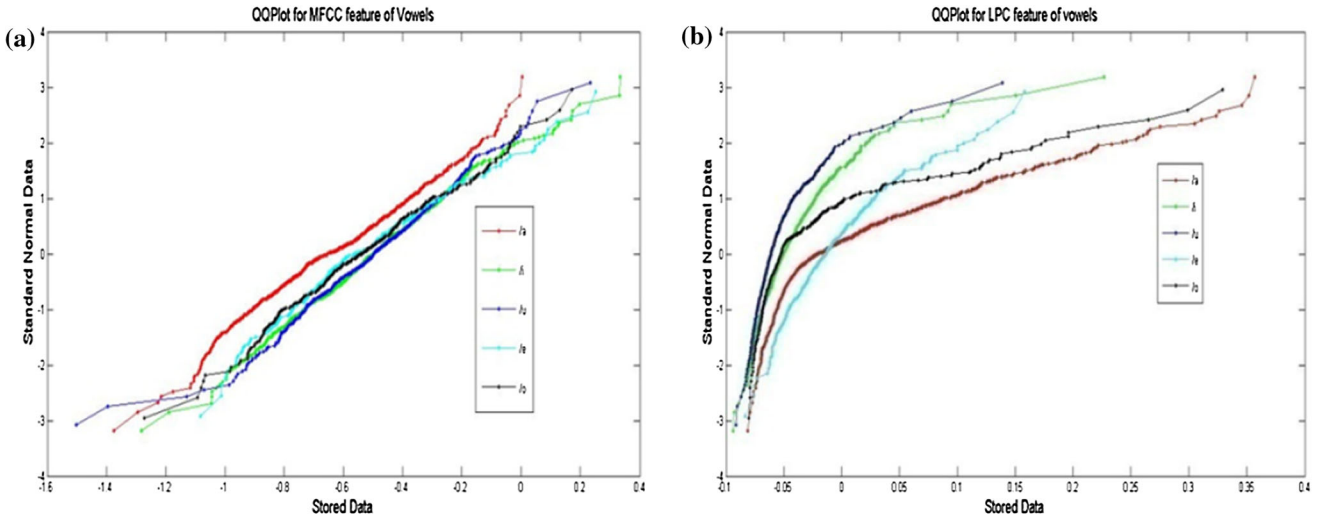


**Fig. 7** **a** QQ Plot for Vowel dataset of MFCC and **b** LPC features

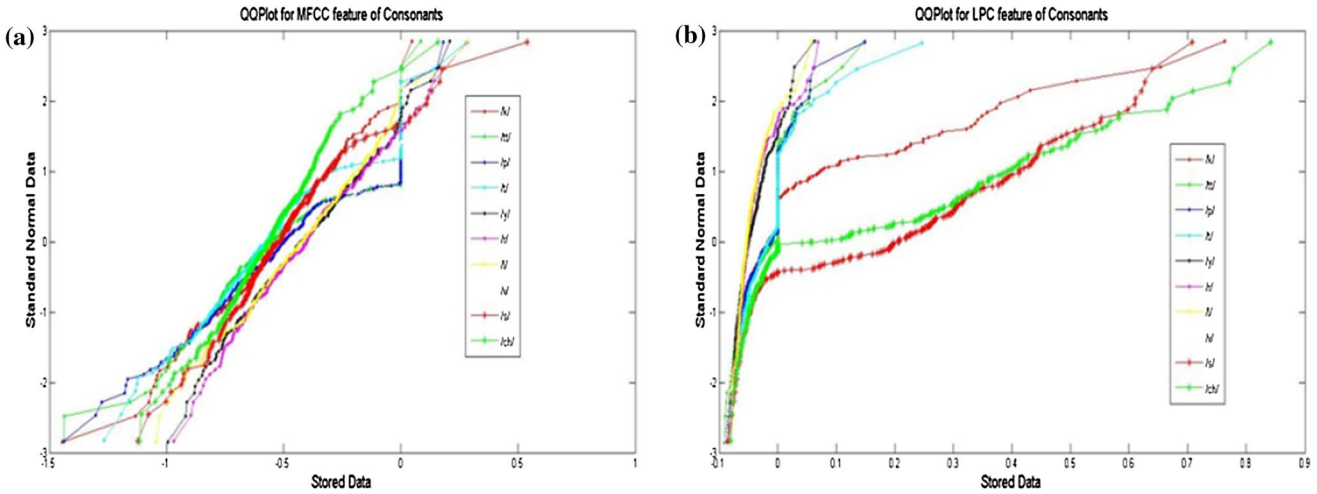


**Fig. 8** **a** QQ Plot for Consonant dataset of MFCC and **b** LPC features

**Table 2** F-ratio for different datasets

| | Vowels | Consonants |
|---|---|---|
| MFCC | 0.3976 | 0.2857 |
| LPC | 0.2994 | 0.2039 |

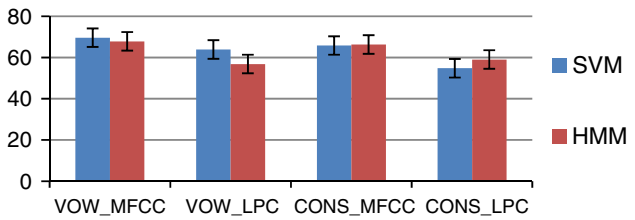


**Fig. 9** Classification accuracy for vowels and consonants by SVM and HMM classifiers

weights $w_{jm}$ are constrained to be nonnegative and normalized: $\sum_{m=1}^{M} w_{jm} = 1$ for all states j.

Let $\theta$ denote the vector of model parameters including transition probabilities, mixture weights, mean vectors, and covariance matrices. The goal of parameter estimation in HMMs is to compute the optimal $\theta^*$ given $N$ pairs of observation and target label sequences $\{X_i, Y_i\}$, where $Y_i$ is the class label (Alpaydin 2004; Sha and Saul 2009) (Fig. 9).

## 5.2 Support vector machine

Support vector machine has its roots in statistical learning theory and has shown promising empirical results in many practical applications. SVM also works very well with high-dimensional data and avoids curse of dimensionality problem (Duda et al. 2006; Tan et al. 2006). It repre-

**Table 3** Classification accuracy for individual vowels and consonant classes

| | | SVM | | HMM | |
|---|---|---|---|---|---|
| | | MFCC | LPC | MFCC | LPC |
| Vowels | /a/ | 85.50 | 83.50 | 82 | 75 |
| | /i/ | 49.50 | 29 | 41.50 | 35.5 |
| | /u/ | 68.50 | 72 | 62 | 56.5 |
| | /e/ | 75.50 | 78 | 78.5 | 57.5 |
| | /o/ | 69 | 57 | 75.00 | 59.5 |
| Consonants | /k/ | 69.5 | 29.5 | 77 | 42 |
| | /ch/ | 87.50 | 85.50 | 86 | 90 |
| | /tt/ | 48.50 | 51.00 | 41 | 29.5 |
| | /t/ | 30 | 8 | 45.50 | 27 |
| | /p/ | 37 | 33 | 47.50 | 46.5 |
| | /y/ | 87.50 | 83.50 | 82 | 88 |
| | /r/ | 78.50 | 48.50 | 74 | 51.50 |
| | /l/ | 72 | 51 | 68.5 | 46 |
| | /v/ | 83.50 | 64 | 76.5 | 70.5 |
| | /s/ | 63.50 | 89 | 65 | 86.5 |

sents the decision boundary using a subset of the training set, known as the support vectors. Consider a set of training observations, denoted as $(\boldsymbol{x_i}, y_i)$ $(i = 1, 2, \cdots k)$, where $\boldsymbol{x_i}, = (x_{i1,}, x_{i2,}, \ldots., x_{id})^T$ corresponds to the attribute set for the $i^{th}$ observation and $y_i \in \{-1, 1\}$ denote its class label. The decision boundary can be written in the form:

$$\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$$

where, **w** and b are the parameters of the model. The training phase of the SVM involves estimating the parameters **w** and b of the decision boundary from the training data. The parameters must be chosen in such a way that the following conditions are met.

$$\boldsymbol{w} \cdot \boldsymbol{x} + b \geq 1 \quad \text{if } y_i = 1$$
$$\boldsymbol{w} \cdot \boldsymbol{x} + b \leq -1 \quad \text{if } y_i = -1$$

Another constraint to maximizing the margin, is equivalent to minimizing the following objective function.

$$f\,(w) = \frac{\|\boldsymbol{w}\|^2}{2} \tag{5.3}$$

If the data are not linearly separable in the feature space, they can be projected into a higher dimensional space by means of a kernel. There are three main types of Kernel functions used in literature viz; polynomial, Gaussian radial basis function and sigmoid function (He and Zhou 2005).

## 6 Experimental results

We have used SVM and hidden Markov model (HMM) for the classification of vowels and consonants. Hold out method is applied to divide the dataset into training and testing set. *Forty* patterns of each alphabet are randomly divided into two sets; one set is used for training and another for testing. We repeated this for 20 cycles till we observed saturation in the result. The final classification accuracy is computed as the average of the classification accuracy over 20 cycles. The two algorithms are compared based on the classification accuracy of alphabets.

SVM algorithm using MFCC feature classifies vowel dataset with an accuracy of 69.6 % and HMM algorithm classifies the same with average classification accuracy of 67.8 %. The corresponding classification accuracies using LPC features are 63.9 and 56.8 % respectively.

Consonant dataset with MFCC features is classified with average accuracy of 65.8 % using SVM and 66.33 % using HMM. Similarly consonant dataset with LPC features is classified with 54.8 % accuracy using SVM and 57.8 % using HMM. The following chart shows the variations of accuracy rate for HMM and SVM.

The Table 3, gives the summary for the classification of individual vowels and consonant classes. Among vowels, we observed that vowel /a/ and /e/ are classified with more accuracy compared to other classes. Vowel /i/ has very low classification accuracy. Among structured consonants, /ch/ is classified with very high accuracy rate and among unstructured consonants /y/, /v/ and /s/ are classified with high accu-

racy rate. Consonants like, /tt/, /t/, /p/ are classified very poorly.

SVM and HMM classification performance are almost same. We observed two things in exploratory analysis section and in classification. First thing is that, MFCC features are classified slightly better compared to LPC features. Second observation is that some of the vowels and consonants are shown to be much discriminating than others. Overall classification accuracy is poor with all the cases and this can be improved by using more sophisticated and also improved classification techniques. We have also attempted another slight modification in the experiment by combining the HMM and SVM classifiers. This has improved the classification accuracy by significant percentage.

## 7 Conclusions

MFCC and LPC feature extraction techniques are applied for audio files of vowels and consonants in *Kannada* language. These features have been analyzed suing exploratory statistical methods for their suitability to distinguish alphabets. In these techniques, MFCC feature has shown more discrimination capability compared to LPC features. The results of different exploratory analysis techniques are almost similar i.e MFCC feature better represents the data. SVM and HMM classification techniques are used for classifying these audio files. Both these classification techniques have shown satisfactory performance with MFCC feature. As we observed in exploratory analysis, MFCC has better discriminating ability than LPC and the same fact is also observed in classification. So we can use these statistical techniques for selecting the best feature among the available set of features.

We have used standard feature extraction techniques and classifiers for the experiment, as our main aim was to study methods which analyze the acoustic behavior of sounds using statistical techniques and then support these results with the results of classification. This is done successfully in our work. We would like to investigate these methods further by using more advanced feature extraction, classification and statistical techniques. In future, these techniques will be applied on words and phrases in *Kannada* language.

## References


Alpaydin, E. (2004). *Introduction to machine learning*. India: PHI Publications, ISBN-81-203-2791-8.

Atal, B. S. (1976). Automatic recognition of speakers from their voices. *Proceedings of IEEE*, *64*(4), 460–476.

Atal, B. S., & Hanauer, S. L. (1971). Speech analysis and synthesis by linear prediction of the speech wave. *The Journal of the Acoustical Society of America*, *50*(2), 637–655.

Atal, B. S., & Rabiner, L. (1976). A pattern recognition approach to voiced-unvoiced-silence classification with applications to speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, *24*(3), 201–212.

Axelrod, S., & Maison, B. (2004). Combination of HMM With DTW for speech recognition. In *Proceedings of international conference on acoustics, speech and signal processing* (ICASSP 2004) (pp. 173–176).

Chien, J., Shinoda, K., & Furui, S. (2007). Predictive minimum bayes risk classification for robust speech recognition. In *INTERSPEECH* 2007, August 27–31, Belgium (pp. 1062–1065).

Das, B., Mandal, S., Mitra, P., & Basu, A. (2013). Effect of aging on speech features & phoneme recognition: A study on Bengali vowels. *The International Journal of Speech Technology*, *16*, 19–31.

Davis, S. B., & Mermelstien, P. (1980). Comparison of parametric representation for monosyllabic word recognition in continuous speech recognition. *IEEE Transactions on Acoustics, Speech And Signal Processing*, *28*(4), 357–365.

Duda, R. O., Hart, P. E., & Stork, D. G. (2006). *Pattern classification*. New York: WILEY Publications.

Ephraim, Y. (1992). Statistical model based speech enhancement systems. *Proceedings of IEEE*, *80*(10), 1526–1555. ISSN 0018–9219.

Ganga Shetty, S. V., & Yagnanarayana, B. (2001). Neural network models for recognition of consonant–vowel (CV) utterances. In *INNS-IEEE international joint conference on neural networks*, Washington, DC (pp. 1542–1547), July, 2001.

Hegde, S., Achary, K. K., & Shetty, S. (2012). Isolated word recognition for Kannada language using support vector machine. In *International conference on information processing 2012, CCIS 292* (Vol. 292, pp. 262–269). Berlin: Springer

He, X., & Zhou, X. (2005). Audio classification by hybrid support vector machine / hidden Markov model. *UK World Journal of Modeling and Simulation*, *1*(1), 56–59. ISSN 1746–7233.

Jiang, H., Li, X., & Liu, C. (2006). Large margin HMM for speech recognition. *IEEE Transactions on Audio, Speech and Language Processing*, *14*(5), 1584–1595.

Johnson, R. A., & Wichern, D. W. (2007). *Applied multivariate statistical analysis*. Englewood Cliffs: PHI Publications. ISBN-978-81-203-4587-4.

Kaur, Er, A., & Singh, Er, T. (2010). Segmentation of continuous Punjabi speech signal into syllables. In *Proceedings of the world congress on engineering and computer*, WCECS 2010, October 20–22, 2010, San Francisco, USA.

Kinsner, W., & Peters, D. (1988). A speech recognition system using linear predictive coding and dynamic time warping. In *Engineering in medicine and biology society, 1988. Proceedings of the annual international conference of the IEEE*. 4–7 Nov. 1988 (Vol. 3, pp. 1070–1071 ) New Orleans, LA, USA.

Kumar, S. R. K., & Lajish, V. L. (2013). Phoneme recognition using zero crossing interval distribution of speech patterns & ANN. *The International Journal of Speech Technology*, *16*, 125–131.

Lamel, L. F., Rabiner, L. R., Rosenberg, A. E., & Wilpon, J. G. (1981). An improved end point detector for isolated speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing.*, *29*(4), 777–785.

Lakshmi, A., & Murthy, A. H. (2006). Syllable based continuous speech recognizer for Tamil, *Proceedings of international conference on spoken language, INTERSPEECH 2006 - ICSLP*, September 17–21, Pittsburgh, Pennsylvania (pp. 1878–1881).

Linde, Y., Buzo, A., & Gray, R. M. (1980). An algorithm for vector quantiser design. *IEEE Transactions on Communications*, *28*(1), 84–95.

McLoughlin, I. (2009). *Applied speech and audio processing*. Cambridge: Cambridge University Press.

Nag, S., Treiman, R., & Snowling, M. J. (2010). Learning to spell in an alphasyllabary: The case of Kannada. *Writing Systems Research*, *2*, 41–52. doi:10.1093/wsr/wsq001.

Patro, H., Senthil, R. G., & Dandapat, S. (2007). Statistical feature evaluation for classification of stressed speech. *International Journal of Speech Technology*, *10*, 143–152.

Peltonen, V., Tuomi, J., Klapuri, A., Huopaniemi, J., & Sorsa, T. (2002). Computational auditory scene recognition. In *IEEE international conference on acoustics speech and signal processing*, (Vol. 2, pp. II-1941 - II-1944) 2002.

Rabiner, L., & Juang, B.-H. (1993). *Fundamentals of speech recognition*. Englewood Cliffs, NY, USA,: Prentice Hall PTR. ISBN:0-13-015157-2.

Rahim, M. G., & Juang, B.-H. (1996). Signal bias removal by maximum likelihood estimation for robust telephone speech recognition. *IEEE Transactions on Speech and Audio Processing*, *4*(1), 19.

Sarah, Hawkins (2003). Contribution of fine phonetic detail to speech understanding. In textitProceedings of the 15th international congress of phonetic sciences (pp. 293–296).

Sha, F., & Saul, L. K. (2009). Large margin training of continuous density hidden Markov models. In J. Keshet & S. Bengio (Eds.), *Automatic speech and speaker recognition: Large margin and kernel methods*. New Jersey: Wiley-Blackwell.

Sohn, J., Kim, N. S., & Sung, W. (1999). Statistical model-based voice activity detection. *IEEE Signal Processing Letters*, *6*(1), 1–3.

Tan, P., Steinbach, M., & Kumar, V. (2006). *Introduction to data mining*. Boston: Pearson Addison Wesley, ISBN: 978-81-317-1472-0.

Thangarajan, R., Natarajan, A. M., & Selvam, M. (2009). Syllable modeling in continuous speech recognition for Tamil language. *International Journal of Speech Technology*, *12*(1), 47–57.