

Acoustic Segment Modeling using Spectral Clustering techniques

Swati

A Thesis Submitted to
Indian Institute of Technology Hyderabad
In Partial Fulfillment of the Requirements for
The Degree of Master of Technology



Department of Electrical Engineering

June 2016

Declaration

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

Swati

(Signature)

SWATI

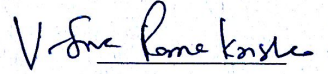
(Swati)

EE14MTECH11014

(Roll No.)

Approval Sheet

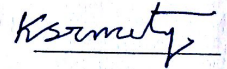
This Thesis entitled Acoustic Segment Modeling using Spectral Clustering techniques by Swati is approved for the degree of Master of Technology from IIT Hyderabad



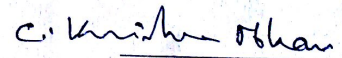
(Dr. Siva Rama Krishna Vanjari) Examiner
Dept. of Electrical Eng
IITH



(Dr. Siva Kumar K) Examiner
Dept. of Electrical Eng
IITH



(Dr. K. Sri Rama Murty) Adviser
Dept. of Electrical Eng
IITH



(Dr. C. Krishna Mohan) Chairman
Dept. of Computer Science and Eng
IITH

Acknowledgements

I would like to express deepest gratitude to my adviser Dr. K. Sri Rama Murty for his full support, expert guidance and encouragement throughout my study and research. It would have been impossible for me to complete my thesis without his incredible patience and timely wisdom and counsel.

I would also like to thank my labmates, Saurabhchand Bhati, P. Raghavendra Reddy, Shekhar Nayak and Y. Satya Dheeraj for supporting and helping me in my research work. My sincere thanks to all my friends for their belief and positive attitude towards me.

Finally, I would like to thank my parents and brother for their unconditional love and support at each point of time in my life. Without their encouragement, I would not have been able to complete my thesis.

Dedication

To my beloved parents for infinite support and friends for believing in me.

Abstract

Acoustic models play a very important role in many speech applications like speech recognition, spoken term detection, topic classification, language identification etc. Acoustic segment modeling (ASM) is a method to build acoustic models based on acoustic similarities of speech segments.

Typically, the standard process of training acoustic models is supervised which has attained great success in past. The issue with supervised techniques is the availability of large amount of transcribed data and language-specific knowledge. Transcribing the speech data is time consuming, tedious and demands a lot of expert human labour, and hence expensive.

This work is focused on building acoustic models in an unsupervised scenario, where only untranscribed speech recordings are available. The main objective of unsupervised acoustic modeling is to recognize the basic units of a spoken language, to tokenize speech utterances and to build the corresponding acoustic models. The framework consists of three stages - Initial segmentation, segment labeling and iterative modeling. This work concentrated on first two stages. Utterances are first segmented and similar segments are clustered in an unsupervised manner. The clusters obtained are termed as ASM units.

We implemented a Language Identification System using unsupervised clustering techniques on few languages. We checked performance of system using multiple test files from these languages.

Contents

Acknowledgements	iv
Abstract	vi
Nomenclature	viii
1 Introduction	1
1.1 Segmentation and Labeling	1
1.2 Thesis organization	3
2 Signal Processing Techniques	4
2.1 MFCC representations	4
2.2 Gaussian Mixture Models	7
2.3 K-Means Algorithm	11
2.4 Hidden Markov Model	12
3 Acoustic Segment Modeling	14
3.1 Initial Segmentation	16
3.2 Feature Representation for Speech	17
3.2.1 Posterior features using GMM	17
4 Clustering Techniques	19
4.1 Gaussian Component Clustering	19
4.2 Segment Clustering	21
5 Experimental Evaluation	23
5.1 ASM Evaluation	23
5.1.1 Speech Corpora - TIMIT database	23
5.1.2 Performance measures	24

5.1.3	Baseline Approaches	25
5.2	Language identification system	26
	References	27

Chapter 1

Introduction

Acoustic Segment modeling (ASM) is an unsupervised of modeling the acoustic units present in speech signals. It is used to model the underlying phoneme like speech units into various classes. These classes are called ASM units. Acoustic Segment modeling is used in many speech applications like Language identification, Spoken term detection, Speaker recognition, topic classification etc. ASM training includes main step of Segmentation and Labeling as explained below.

1.1 Segmentation and Labeling

The word *segmentation* is interpreted as the process of dividing something continuous into discrete, non-overlapping entities i.e., the process of deciding boundaries. *Labeling* is defined as classification of segments obtained. Over the past few years, interest in automatic speech segmentation has increased. A number of speech analysis and synthesis applications need to divide speech signals into phonetic segments (phonemes and syllables) [1]. Both Automatic Speech Recognition (ASR) models and Text-to-Speech (TTS) systems depends on reliable segmentation for achieving good performance [2]. Furthermore, automatic speech segmentation methods are used for the automatic phonetic analysis of large amounts of speech data [3].

Speech segmentation divides an speech utterance into homogeneous portions, where in each portion, the signals share similar properties. Various levels of speech processing uses speech segmentation effectively. For example, one may want to divide an audio stream into speech and non-speech signals such as noise or music, or divide a speech stream according to speaker identity. This problem is known as audio diarization and receives extensive research interests nowadays [4]. And in many

other speech recognition or understanding applications, one may need to segment speech sequence into sentences, phrases, words, syllables or phonemes. Unlike written language, speech signals lack explicit punctuations such as spaces or capitalizations in text for division. Moreover, human speech is a continuous signal and does not change abruptly due to the temporal and physical constraints of the vocal tract. These facts make speech segmentation a challenging problem.

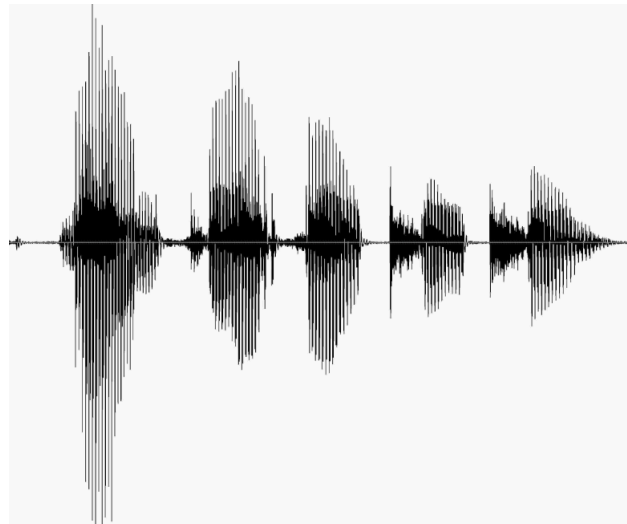


Figure 1.1: Segmented Speech Signal

In the past research, speech signals were segmented manually but there are major drawbacks with manual speech segmentation. Firstly, it is extremely time consuming. An increasing amount of segmented speech data is needed, but the tremendous work load often constraints the amount of speech recordings that can be segmented. Secondly, no standard multi-lingual procedure for speech segmentation has been defined yet. Finally, the manual segmentation and labeling of speech is prone to human random errors and inconsistencies.

By contrast, *automatic segmentation* procedures are by definition free from human interaction. Vast amounts of speech recordings can thus be segmented and labeled by applying a fixed set of objective criteria in a consequent manner. The accuracy of such segmentation may be poorer than human performance, but the errors are more systematic and can hence be taken into account when using the segmented speech material. Since the criteria causing the errors are explicit, some of the errors can be corrected.

Thus, automatic phoneme segmentation has received much research interest.

1.2 Thesis organization

The remainder of this thesis is organized as follows:

Chapter 2 provides background knowledge and some pre-existing signal processing techniques used in this thesis.

Chapter 3 gives an overview of Acoustic segment modeling.

Chapter 4 gives overview of spectral clustering techniques.

Chapter 5 shows experiments and evaluation.

Chapter 2

Signal Processing Techniques

This chapter provides background about the techniques used in the following chapters. The conventional speech feature representation - Mel-scale frequency cepstral coefficients (MFCCs) and the most commonly used acoustic model Gaussian Mixture Model (GMM) will be described. It will explain some details about K-means clustering. It will also review the modeling technique - Hidden Markov Model (HMM) used in this thesis work. The Dynamic Time Warping (DTW) algorithm will be reviewed since it will be used in experiments as a matching method on the speech representations. Finally, we present several speech corpora that are used in the experiments performed in this thesis.

2.1 MFCC representations

The first step in any automatic speech recognition system is to extract features i.e. identify the components of the audio signal that are good for identifying the linguistic content and discarding all the other stuff which carries information like background noise, emotion etc.

The main point to understand about speech is that the sounds generated by a human are filtered by the shape of the vocal tract including tongue, teeth etc. This shape determines what sound comes out. If we can determine the shape accurately, this should give us an accurate representation of the phoneme being produced. The shape of the vocal tract manifests itself in the envelope of the short time power spectrum, and the job of MFCC's is to accurately represent this envelope.

MFCC is a filter bank based approach, the design of filters in such a way that they be similar to the human auditory frequency perception. Researchers have suggested that directly computed filter bank features are more robust for recognition of speech in noisy condition [5]. As the human ear is

also a good speaker recognizer, people tried MFCC feature for speaker recognition.

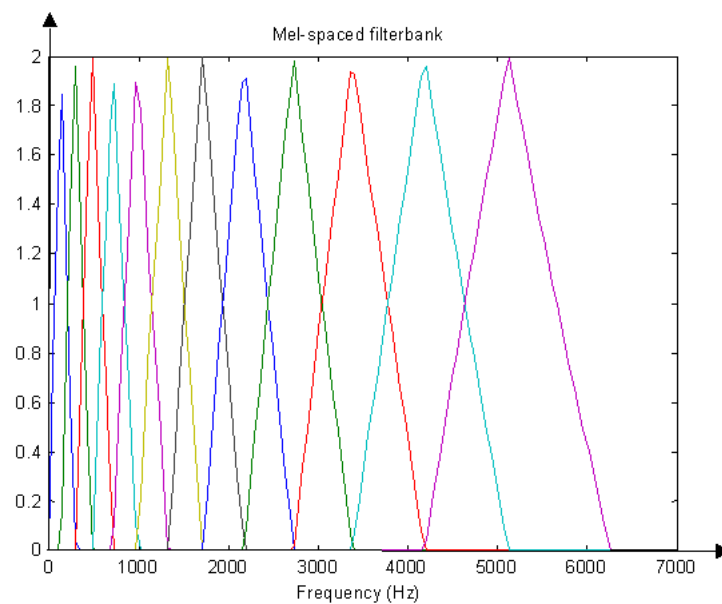


Figure 2.1: Triangular filters placed according to Mel frequency scale. A Melfrequency filter is used to reduce spectral resolution, and convert all frequency components to be placed according to the Mel-scale

MFCC is widely used in Automatic Speaker Recognition systems because of:

- The cepstral features are roughly orthogonal because of the DCT.
- Cepstral mean subtraction eliminates static channel noise.
- MFCC is less sensitive to additive noise than some other feature extraction technique such as linear prediction cepstral coefficients (LPCC).

What is the Mel Scale?

The Mel scale relates perceived frequency, or pitch, of a pure tone to its actual measured frequency. Humans are much better at discerning small changes in pitch at low frequencies than they are at high frequencies. Incorporating this scale makes our features match more closely what humans hear.

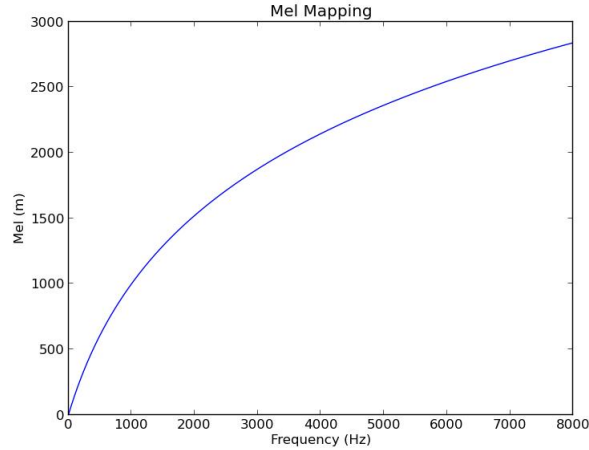


Figure 2.2: Frequency to Mel-Frequency Mapping

The formula for converting from frequency to Mel scale is:

$$M(f) = 1125 \ln(1 + f/700) \quad (2.1)$$

To go from Mels back to frequency:

$$M^{-1}(f) = 700(\exp(m/1125) - 1) \quad (2.2)$$

Implementation steps:

1. Segmenting all concatenated voiced speech signal into 25ms length frames. Frame step is usually 10ms, which allows some overlap to the frames.
2. A short-time Fourier transform (STFT) is performed on every frame of segmented speech signal given by:

$$S(k) = \sum_{n=1}^N s(n)h(n)e^{-j2\pi kn/N} \quad , \quad 1 \leq k \leq K \quad (2.3)$$

where, $s(n)$ is our time domain speech frame, $h(n)$ is an N sample long analysis window, and K is the length of DFT. The periodogram-based power spectral estimate for the speech frame $s(n)$ is given by:

$$P(k) = \frac{1}{N} |S(k)|^2 \quad (2.4)$$

3. Next step is to compute the Mel-spaced filterbank. This is a set of 20-40 (26 is standard) triangular filters that we apply to the periodogram power spectral estimate from previous

step. To calculate filterbank energies we multiply each filterbank with the power spectrum, then add up the coefficients.

4. Take the log of each of the energies from step 3. This leaves us with log filterbank energies.
5. Take the Discrete Cosine Transform (DCT) of the log filterbank energies to give us the cepstral coefficients.

The resulting features are called Mel Frequency Cepstral Coefficients (MFCCs).

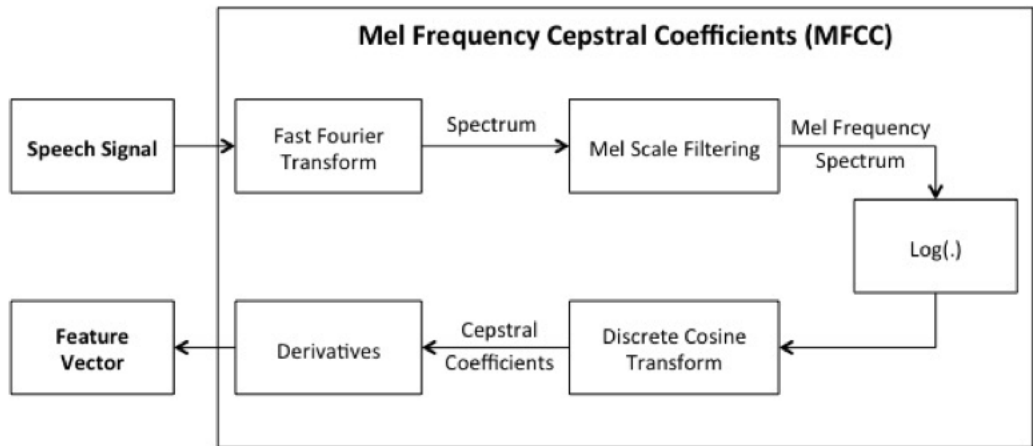


Figure 2.3: Block diagram for MFCC generation

After the above steps, the calculation of MFCCs given a speech signal is complete. In practice, when using MFCCs in the acoustic modeling, long-term MFCCs features are often considered, such as delta (Δ) MFCCs and delta-delta ($\Delta\Delta$) MFCCs [6]. The Δ MFCCs are the first derivatives of the original MFCCs and the $\Delta\Delta$ MFCCs are the second derivatives of the original MFCCs. A common configuration of the modern ASR feature extraction module is to use the original MFCCs stacked with the Δ and $\Delta\Delta$ features. The original MFCCs are represented by the first 12 components of the DCT output plus the total energy ($+\Delta$ Energy + $\Delta\Delta$ Energy), which results in a $13+13+13=39$ dimensional feature vector for each speech frame.

2.2 Gaussian Mixture Models

One of the most important task when working with mixture model-based clustering is precisely, selecting the type of function which offers a better adjust to the data field and the type of task we face

up to. Between the different types of mixture model-based clustering, one of the most commonly used is clustering based in Gaussian Mixture Models (GMMs) [7], [8].

Gaussian mixture model as a simple linear superposition of Gaussian components, aimed at providing a richer class of density models than the single Gaussian. Techniques based on GMM are applied to many different tasks. Some of the most common applications are speaker identification, speech recognition, image segmentation, biometric verification or detection of image color and texture.

Gaussian Mixture Model (GMM) is a parametric probability density function represented as a weighted sum of Gaussian component densities. GMMs are commonly used as a parametric model of the probability distribution of continuous measurements or features in a biometric system, such as vocal-tract related spectral features in a speaker recognition system. GMM parameters are estimated from training data using the iterative Expectation-Maximization (EM) algorithm or Maximum A Posteriori (MAP) estimation from a well-trained prior model.

Fig. 2.4 illustrates the joint density capturing capabilities of GMM, using 2-dimensional data uniformly disturbed along a circular ring. The red ellipses, superimposed on the data (blue) points, correspond to the locations and shapes of the estimated Gaussian mixtures.

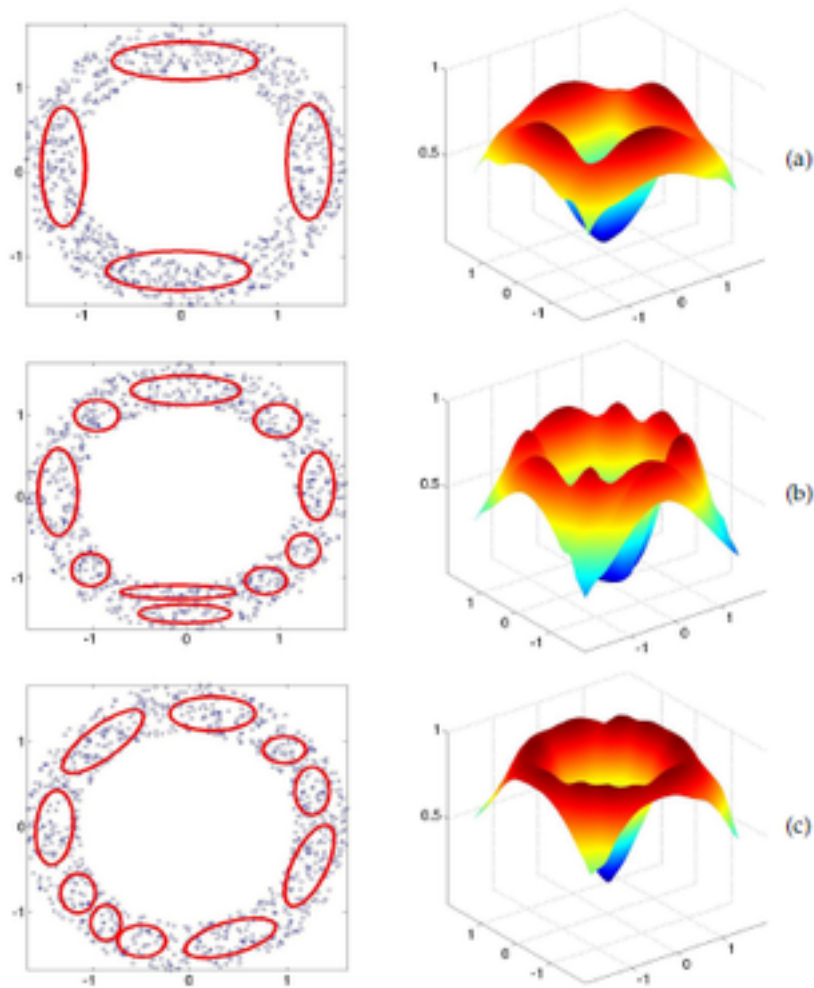


Figure 2.4: Illustration of distribution capturing capability of GMM. GMM trained with diagonal covariance matrices (a) 4-mixtures (b) 10-mixtures and (c) 10-mixture GMM trained with full covariance matrices

A Gaussian mixture model is a weighted sum of M component Gaussian densities as given by the equation,

$$p(x|\lambda) = \sum_{i=1}^M w_i g(x|\mu_i, \Sigma_i) \quad (2.5)$$

where x is a D -dimensional continuous-valued data vector (i.e. measurement or features), w_i , $i = 1, \dots, M$, are the mixture weights, and $g(x|\mu_i, \Sigma_i)$, $i = 1, \dots, M$, are the component Gaussian densities.

Each component density is a D -variate Gaussian function of the form,

$$g(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{D/2} |\Sigma_i|^{1/2}} \exp\left\{-\frac{1}{2}(x - \mu_i)' \Sigma_i^{-1} (x - \mu_i)\right\} \quad (2.6)$$

with mean vector μ_i and covariance matrix Σ_i . The mixture weights satisfy the constraint that $\sum_{i=1}^M w_i = 1$.

The complete Gaussian mixture model is parameterized by the mean vectors, covariance matrices and mixture weights from all component densities. These parameters are collectively represented by the notation,

$$\lambda = \{w_i, \mu_i, \Sigma_i\} \quad , \quad i = 1, 2, \dots, M \quad (2.7)$$

EM Algorithm for Gaussian Mixture Models:

Given a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters (comprising the means and covariances of the components and the mixing coefficients).

1. Initialize the means μ_i , covariances Σ_i and mixing coefficients w_i , and evaluate the initial value of the log likelihood.
2. **E-Step:** Evaluate the responsibilities using the current parameter values

$$\gamma(z_{ni}) = \frac{w_i g(x_n | \mu_i, \Sigma_i)}{\sum_{j=1}^M w_j g(x_n | \mu_j, \Sigma_j)} \quad (2.8)$$

3. **M-Step:** Re-estimate the parameters using the current responsibilities

$$\mu_i^{new} = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) x_n \quad (2.9)$$

$$\Sigma_i^{new} = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (x_n - \mu_i^{new})(x_n - \mu_i^{new})^T \quad (2.10)$$

$$w_i^{new} = \frac{N_k}{N} \quad (2.11)$$

where,

$$N_k = \sum_{n=1}^N \gamma(z_{nk}) \quad (2.12)$$

4. Evaluate the log likelihood

$$\ln p(X|\mu, \Sigma, w) = \sum_{n=1}^N \ln \left\{ \sum_{i=1}^M w_i g(x|\mu_i, \Sigma_i) \right\} \quad (2.13)$$

and check for convergence of either the parameters or the log likelihood. If the convergence criterion is not satisfied return to step 2.

2.3 K-Means Algorithm

The K-means algorithm [9], [10] is another algorithm for clustering real-valued data. It is based on minimizing the sum of Euclidean distances between each point and its assigned cluster, rather than on a probabilistic model. The algorithm takes as input an $n \times d$ data matrix (with real-valued entries), a value for K, and operates as follows:

1. Initialize by randomly selecting K mean vectors, e.g., pick K data vectors (rows) randomly from the input data matrix
2. Assign each of the n data vectors to the cluster corresponding to which of the K clusters means it is closest to, where distance is measured as Euclidean distance in the d-dimensional input space.
3. For each cluster k, compute its new mean as the mean (average) of all the data vectors that were assigned to this cluster in Step 2.
4. Check for convergence. An easy way to determine convergence is to execute Step 2 and check if any of the data points change cluster assignments relative to their assignment on the previous iteration. If not, exit; if 1 or more points change cluster assignment, continue to Step 3.

The K-means algorithm can be viewed as a heuristic search algorithm for finding the cluster assignments that minimize the total sum of squares, namely the sum of the squared Euclidean distances from each of the n data points to a cluster center. Finding the optimal solution is NP-hard, so K-means may converge to local minima. For this reason it can be useful to start the algorithm multiple random starting conditions, and select the solution with the minimum sum of squares score over different runs.

The K-means algorithm can also be thought of as a simpler non-probabilistic alternative to Gaussian mixtures. K-means has no explicit notion of cluster covariances. One can reduce Gaussian

mixture clustering to K-means if one were to (a) fix a priori all the covariances for the K components to be the identity matrix (and not update them during the M-step), and (b) during the E-step, for each data vector, assign a membership probability of 1 for the component it is most likely to belong to, and 0 for all the other memberships (in effect make a hard decision on component membership at each iteration).

2.4 Hidden Markov Model

A hidden Markov model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states. A HMM can be presented as the simplest dynamic Bayesian network. HMMs [11] are the most popular and successful statistical acoustic models for speech recognition.

A hidden Markov model (HMM) refers to a statistical model designed to capture the hidden states of a system and their evolution, which are governed by a Markov process. An HMM may be formulated in a simple state-space form, and much of the earlier works in this area focused on solving the problem of nonlinear optimization with the utility of forward-backward algorithms. A Hidden Markov Model is a discrete-time finite-state homogenous Markov chain observed through a discrete-time memoryless invariant channel. The channel is characterized by a finite set of transition densities indexed by the states of the Markov chain. These densities may be members of any parametric family such as Gaussian, Poisson, etc. The initial distribution of the Markov chain, the transition matrix, and the densities of the channel may depend on some parameter that characterizes the HMM.

Hidden Markov Models have become a basic tool for modelling stochastic systems with a wide range of applications in such diverse areas as nanotechnology [12], quantized Gaussian linear regression [13], [13], telecommunication [14], speech recognition [15], switching systems [16], [17], financial mathematics [18] and protein research [19].

The generative model of the standard HMM is given by:

$$q_{t+1} \sim P(q_{t+1}|q_t) \tag{2.14}$$

$$o_{t+1} \sim p(o_t|q_t) \tag{2.15}$$

where q_t and o_t are the state and observation vector respectively at time t . The parameters of a HMM with N discrete states are given by $\theta(\pi, A, b)$ where $\pi = \{\pi_i : 1 \leq i \leq N\}$ are the initial probabilities of the states, $A = \{a_{ij} : 1 \leq i, j \leq N\}$ are the transition probabilities between two states and $b = \{b_j(o_t) : 1 \leq j \leq N\}$ are the observation probabilities of the states. A typical *left-to-right* HMM topology used in a speech recognition system is illustrated in Figure 2.5. Two special non-emitting states are used to represent the left-to-right topology. By having the nonemitting start state, the initial probabilities are simply $\pi_1 = 1$ and $\pi_i = 0$ for $i \neq 1$. The arrow joining two states indicates the permissible transitions in the given direction. The HMM in Figure 2.5 assumes that the speech signals being modeled can be logically divided into three segments, within each, the signals are considered i.i.d (independent and identically distributed) and piece-wise stationary.

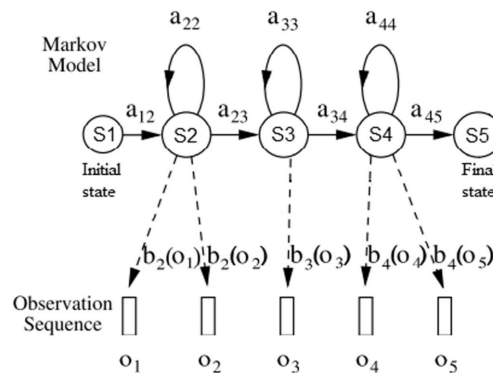


Figure 2.5: The 3 state *left-to-right* Hidden markov model used in speech recognition

In order to use the HMMs in speech recognition, one needs to be able to:

- evaluate the likelihood of the model given the observations
- decode the most likely state sequence given the observations
- estimate the HMM parameters to maximise the objective function.

Chapter 3

Acoustic Segment Modeling

Acoustic models play a very important role in many spoken language applications. They are used to describe the acoustic properties of a set of predefined speech units, e.g., phonemes. Typically the training of acoustic models is a supervised process, which require not only the speech observations but also manual transcriptions and language-specific linguistic knowledge, such as phoneme definition and word dictionary. Supervised training of acoustic models has attained great success for many resource-rich languages (e.g., English, Mandarin). But the issue with this process is that it is not straightforwardly applicable to other languages for which manual transcriptions and linguistic knowledge are difficult to be acquired or even completely absent. Thus, in recent years, there is an increasing research interest in designing acoustic modeling methods that are less reliant on well-organized training resources [20–26].

Acoustic segment modeling (ASM) is an approach used for characterizing fundamental acoustic units. Acoustic models classifies utterances by their acoustic feature sequences. The main goal of acoustic segment modeling is to segment the speech utterances and then build a model for similar segments. M.Siu et al. [26], A. Garcia et al. [27] and M. Siu et al. [28] investigated an similar approach to ASM. ASM has many practical applications. It is used in spoken language identification [29], topic classification [26] [28], speaker recognition [30] [31], zero-resource QbyE STD tasks [32] and also semantic retrieval of spoken content [33]. ASMs has been shown to be very useful in handling multilingual issue in speaker recognition. Typically, the standard process of training acoustic models is supervised which requires not only language-specific knowledge but also a large amount of transcribed data. ASM built with supervised techniques has attained great success in past but the issue

with this approach is that it requires a lot of labeled speech data. Transcribing the speech data is time consuming, tedious and demands a lot of expert human labor, therefore it becomes expensive. Hence, transcribed databases are only available for a very small number of languages in the world. Also acoustic models built for a specific language cannot be directly applied to other languages for which manual transcriptions are not available. Due to these reasons, building acoustic models in unsupervised scenario, where only untranscribed speech recordings are available has gained more attention in recent years. Unsupervised acoustic modeling is basically a clustering approach. Utterances are first segmented and clustered in unsupervised manner. Absence of requirement of speech transcriptions and explicit linguistic knowledge makes it more dominant. Due to these reasons, most of the present techniques are inclined towards unsupervised framework.

Acoustic segment modeling consists of mainly three stages, named as initial segmentation, segment labeling and iterative modeling.

1. Initial segmentation is the first stage that divides a continuous speech utterance into variable-length segments. Each segment has similar acoustic properties. Segmentation minimizes a distortion measure within a segment. Features within a segment are similar to each other as compared to features across segments. Distortion measure can change the performance of segmentation algorithm significantly. Euclidean distance was used as a distortion measure in [34]. Speech segmentation using maximum-likelihood approach and dynamic programming was proposed by Lee et al. [35] and Bacchiani et al. [36]. A maximum margin clustering method was proposed in [37]. In [38], a bottom-up hierarchical clustering method was proposed. Bayesian hidden Markov model (HMM) with Dirichlet process priors was used for segmentation in [39].
2. After getting segments for all utterances, similar segments are clustered together and each cluster is given a unique label. Similarity between segments is calculated based on some distance measure. The segments which are very similar in nature will have less distance as compared to other dissimilar segments. So all similar segments are placed in one group and dissimilar segments in different groups. Each group formed is then given a unique label such that all segments belonging to particular group gets same label. These clusters are called ASM units. Vector quantization (VQ) was proposed in [35] for segment labeling in which segment is represented by mean of all the features in it and k-means is used for clustering similar segments. The segmental GMM (SGMM) approach was proposed in [40]. In [41], GMM

labeling (GL) method was proposed for segment labeling. Gaussian component clustering (GCC) and Segment clustering (SC) was proposed in [34].

3. After segment labeling, each segment cluster is assigned a cluster label. After assigning labels to all the segments, iterative modeling is applied on top of it. Usually, standard Viterbi decoding technique [42] is used for this purpose.

3.1 Initial Segmentation

The main aim of initial segmentation is to detect the phoneme boundaries that divide a speech utterance into non-overlapping segments. Each of these segments are suppose to have acoustic properties coherent in nature. Also segment boundaries are associated with significant acoustic discontinuities [43]. The segmentation can be formulated as a problem of minimizing some kind of within segment distortion measure. In this work, we estimated segment boundaries using sum of square error (SSE) criterion. This section describes the basic formulation of optimal segmentation.

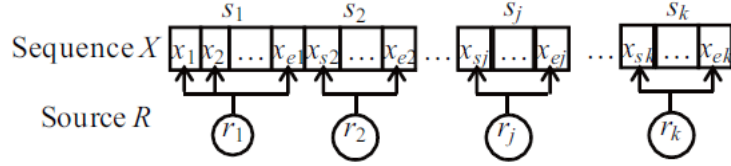


Figure 3.1: A brief description of segmentation model.

Let $O = \{o_1, o_2, \dots, o_T\}$ denote the feature vectors extracted from an utterance, where T is the length of O and each o_i is a d -dimensional feature vector. A segmentation that divides sequence O in k non overlapping contiguous segments can be denoted as $S = \{s_1, s_2, \dots, s_k\}$, where $s_j = \{c_j, c_{j+1}, \dots, e_j\}$ using c_j, e_j to represent the first and last indices of j th segment. SSE criterion on S segmentation is defined as,

$$SSE(O, S) = \sum_{j=1}^k \sum_{i=c_j}^{e_j} \|o_i - \hat{m}_j\|^2 \quad (3.1)$$

$$\hat{m}_j = \frac{1}{(e_j - c_j)} \sum_{i=c_j}^{e_j} o_i \quad (3.2)$$

In the initial state, the algorithm defines one segment s_i for each x_i present in X , i.e., segments

containing just one vector of the data. The algorithm iteratively merges segments until it reaches the imposed number of segments. Given the segments s_j , s_{j+1} and R as the segment resulting from grouping of s_j and s_{j+1} , the grouping criterion is defined as:

$$\Delta SSE(O, j) = SSE(O, R) - SSE(X, s_j) - SSE(X, s_{j+1}) \quad (3.3)$$

The optimal grouping is performed by merging the adjacent segments such that the grouping criterion is the minimum. This kind of approach is known as Agglomerative Clustering algorithm.

3.2 Feature Representation for Speech

Theoretically, recognizing speech directly from the digitized waveform should be possible. However, because of the large number of variations in the speech signals, it is better to apply some feature extraction to reduce that variability. Particularly, eliminating various source of information, such as whether the sound is voiced or unvoiced and, if voiced, it eliminates the effect of the periodicity or pitch, amplitude of excitation signal and fundamental frequency etc.

For speech recognition, data manipulation can be simplified by eliminating the redundant and irrelevant aspects of the speech waveform. An efficient representation for speech recognition would be set of parameters that yield similar values for the same phonemes uttered by various speakers. Speech analysis can be done either in time domain or in frequency domain. The speech analysis is done to obtain a more useful representation of the speech signal in terms of parameters that contain relevant information in an efficient format.

Many feature extraction techniques are available which includes, Linear predictive analysis (LPC), Linear predictive cepstral coefficients (LPCC), Mel-frequency cepstral coefficients (MFCC), Power spectral analysis (FFT) etc. In this section, we summarize the feature extraction technique that is used in our analysis.

3.2.1 Posterior features using GMM

Posterior features are used successfully in many speech applications, like speech recognition [44], spoken term detection [45], story segmentation [46] and spoken language identification [47]. Posterior features are more robust as compared to conventional spectral features like MFCC's. In this

work, segment-level posterior representations are formulated and applied to the problem of segment labeling.

Given a trained GMM and a observation point o_t , the posterior probability that it is generated by the i_{th} Gaussian component c_i can be computed using the Bayes rule as follows:

$$P(c_i/o_t) = \frac{w_i N(o/\mu_i, \Sigma_i)}{p(o)} \quad (3.4)$$

where,

$$p(o) = \sum_{i=1}^M w_i N(o/\mu_i, \Sigma_i) \quad (3.5)$$

where $N(\cdot)$ is Gaussian distribution, M is number of mixtures, w_i is the weight of the i_{th} Gaussian component, μ_i is its mean vector and Σ_i is its co-variance matrix. Thus, the frame-level posterior feature vector q_t is composed as,

$$q_t = [p(c_1|o_t), p(c_2|o_t), p(c_3|o_t), \dots, p(c_M|o_t)]^T \quad (3.6)$$

where, $C = \{c_1, c_2, \dots, c_M\}$ are the M pre-defined speech classes represented by M gaussian components of GMM.

A matrix $Q \in R^{M \times T}$ is obtained by stacking frame-level posterior feature vectors, given as

$$Q = [q_1, q_2, \dots, q_T] \quad (3.7)$$

where T is the number of frames. Q is referred to as a class-by-frame matrix.

Let $S = \{s_1, s_2, \dots, s_K\}$ are the segment boundaries obtained after initial segmentation, then a *class-by-segment* matrix $X \in R^{M \times K}$ is given as,

$$X = [x_1, x_2, \dots, x_K] \quad (3.8)$$

where,

$$x_k = \frac{1}{e_k - b_k + 1} \sum_{t=b_k}^{e_k} o_t \quad (3.9)$$

Thus, x_k are the normalized accumulated posterior probabilities.

Chapter 4

Clustering Techniques

This section describes the segment labeling approaches using spectral clustering methods. In recent years, spectral clustering has become one of the most popular modern clustering algorithms. It is simple to implement, can be solved efficiently by standard linear algebra methods, and very often outperforms traditional clustering algorithms such as the k-means algorithm. There are two spectral clustering techniques as described below.

4.1 Gaussian Component Clustering

Gaussian Component Clustering (GCC) approach is used for segment labeling. GCC is a two-stage approach, i.e., building initial acoustic models followed by labeling speech segments. The main idea is to train a universal GMM with many components, and perform clustering on these Gaussian components. Each of these cluster represents a small GMM which is regarded as the initial acoustic model for an ASM unit. These initial acoustic models are then used to score speech segments so as to generate initial label sequences.

With the Gaussian-by-segment matrix $X \in R^{M \times K}$, GCC actually performs clustering on the row vectors of X . We use inner product as the similarity measure between each pair of the row vectors, and follow the state-of-the-art spectral clustering procedure as described in [48]. The affinity matrix $A \in R^{M \times M}$ is given as,

$$A = XX^T \tag{4.1}$$

where $A_{i,j}$ denotes the similarity between the i_{th} and j_{th} Gaussian components. The normalized

Laplacian matrix $L \in R^{M \times M}$ [49] is,

$$L = I - D^{-1/2}AD^{-1/2} \quad (4.2)$$

where, $D = \text{diag}A_1$ is a diagonal matrix with its diagonal element $D_{i,i} = \sum_{m=1}^M A_{i,m,1}$ is a column vector of all ones.

The embedding representations of the M Gaussian components $Y = [y_1, y_2, \dots, y_R]$ can be obtained by,

$$\min_{Y \in R^{M \times R}} \text{tr}(Y^T LY) \quad \text{s.t.}, Y^T Y = I \quad (4.3)$$

The solution to above equation is given by eigenvalue decomposition of L , i.e., the column vector y_r is the r_{th} smallest eigenvector of L . The embedding representations of the Gaussian components are actually the row vectors of Y , on which k-means is applied to obtain the cluster memberships. The resulted clusters of Gaussian components form a set of GMMs, which are used to label the speech segments. The whole procedure is summarized in following Algorithm.

Algorithm: Gaussian Component Clustering(GCC)

Input: Gaussian-by-segment matrix $X \in R^{M \times K}$, and the cluster number R .

Output: R GMMs and cluster label of each segment.

1. compute affinity matrix A .
2. compute Laplacian matrix L .
3. construct matrix $Y = [y_1, y_2, \dots, y_R]$, where y_r is the smallest eigenvector of L .
4. normalize each row vector of Y to have unit l_2 -norm.
5. apply k-means on the M row vectors of Y to find R clusters.
6. assign the i_{th} Gaussian component to cluster r if the i_{th} row vector of Y is assigned to cluster r .

7. form a GMM for each cluster by assigning equal weights to its Gaussian components.
 8. score each segment with the R GMMs, and label it with the index of the GMM that scores highest.
-

4.2 Segment Clustering

Segment clustering (SC) is applied directly on the segment posterior representations. It is a one stage approach. SC applies clustering on speech segments. The similarity measure is reformed as the inner product of the posterior representations. It also uses k-means to obtain the cluster memberships of the speech segments.

Let $X \in R^{M \times K}$ be a class-by-segment matrix. Segment clustering (SC) aims to perform clustering on the column vectors of X . In practice, the use of standard spectral clustering algorithm encounters a computational problem in segment clustering. The affinity matrix A and the Laplacian matrix L are so large that they cannot be computed efficiently and properly stored in the memory. The algorithms that are capable of handling hours of data for the application of unsupervised acoustic modeling are meant to be designed. Therefore it is desirable to reformulate the problem such that the derivation of the embedding becomes practically feasible. We tackle this problem by deriving Y without the explicit computation of A and L . The similarity between speech segments is still computed as the inner product of the posterior representations, i.e.,

$$A = X^T X \quad (4.4)$$

With the affinity matrix A , the computation of Laplacian matrix L is reformulated as,

$$L = I - D^{-1/2} A D^{-1/2} = I - D^{-1/2} X^T X D^{-1/2} = I - \hat{X}^T \hat{X} \quad (4.5)$$

where,

$$\hat{X} = X D^{-1/2} \quad (4.6)$$

with $D = \text{diag}(X^T(X_1))$. Then the embedding representations of speech segments becomes,

$$\max_{Y \in R^{K \times R}} \text{tr}(Y^T(\hat{X}^T \hat{X})Y) \quad \text{s.t.}, Y^T Y = I \quad (4.7)$$

That is, Y is given by the largest eigenvectors of $\hat{X}^T \hat{X}$. The eigenvector of $\hat{X}^T \hat{X}$ can be derived by multiplying the corresponding eigenvector of $\hat{X} \hat{X}^T$ with X . Specifically, let $\hat{Y} = [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_R]^T$ be a matrix consisting of R largest eigenvectors of $\hat{X} \hat{X}^T$. Then Y can be obtained by two steps: (1) compute $Y = (\hat{Y} \hat{X})^T$, and (2) normalize each column vector of Y to have unit l_2 -norm. After computing Y , k -means is applied to obtain the cluster memberships of the speech segments. The whole procedure is summarized in following Algorithm.

Algorithm: Segment Clustering(SC)

Input: Class-by-segment matrix $X \in R^{M \times K}$, and the cluster number R

Output: Cluster label of each segment

1. compute $\hat{X} \in R^{M \times K}$ as $\hat{X} = X D^{-1/2}$.
 2. compute $\hat{A} = \hat{X} \hat{X}^T$, $\hat{A} \in R^{M \times M}$.
 3. construct $\hat{Y} = [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_R]^T \in R^{R \times M}$, where \hat{y}_r is the r_{th} largest eigenvector of \hat{A} .
 4. compute $Y = (\hat{Y} \hat{X})^T$, $Y \in R^{K \times R}$.
 5. normalize each column vector of Y to have unit l_2 -norm.
 6. normalize each row vector of Y to have unit l_2 -norm.
 7. apply k -means on the K row vectors of Y to find R clusters.
 8. assign the i_{th} segment to cluster r if the i_{th} row vector of Y is assigned to cluster r .
-

Chapter 5

Experimental Evaluation

This chapter includes some of the results we got for evaluation of performance of our clustering techniques. It also describes the Language Identification Model built using these techniques.

5.1 ASM Evaluation

5.1.1 Speech Corpora - TIMIT database

The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus (TIMIT - Texas Instruments (TI) and Massachusetts Institute of Technology (MIT)), contains recordings of phonetically-balanced prompted English speech. It was recorded using a Sennheiser close-talking microphone at 16 kHz rate with 16 bit sample resolution. TIMIT contains a total of 6300 sentences (5.4 hours), consisting of 10 sentences spoken by each of 630 speakers from 8 major dialect regions of the United States. All sentences were manually segmented at the phone level. The prompts for the 6300 utterances consist of 2 dialect sentences (SA), 450 phonetically compact sentences (SX) and 1890 phonetically-diverse sentences (SI).

The training set contains 4620 utterances, but usually only SI and SX sentences are used, resulting in 3696 sentences from 462 speakers. The test set contains 1344 utterances from 168 speakers. The core test set, which is the abridged version of the complete testing set, consists of 192 utterances, 8 from each of 24 speakers (2 males and 1 female from each dialect region). With the exception of SA sentences which are usually excluded from tests, the training and test sets do not overlap.

This speech corpus has been a standard database for the speech recognition community for several decades and is still widely used today, for both speech and speaker recognition experiments. This is not only because each utterance is phonetically hand labeled and provided with codes for speaker number, gender and dialect region, but also because it is considered small enough to guarantee a relatively fast turnaround time for complete experiments and large enough to demonstrate system capabilities.

5.1.2 Performance measures

The goal of clustering is to attain high intra-cluster similarity and low inter-cluster similarity. So, we need a criterion to measure the performance of clustering algorithm. Thus, for comparison of performances between baseline and proposed approaches, we used two evaluation metrics: Purity and Normalized mutual information(NMI). Purity is a one of primary validation measure to determine the cluster quality. Each cluster is assigned to phoneme which occurred most frequently in it and then purity is defined as the total number of segments(phonemes) that were classified correctly. It is in the range of 0 to 1 i.e., bad clustering has a value close to 0 and perfect clustering has a purity of 1.

Let $n_{r,p}$ be the number of frames assigned to the r_{th} ASM unit and belong to the p_{th} phoneme. Let $\Omega = \{\Omega_1, \Omega_2, \dots, \Omega_P\}$ and $\omega = \{\omega_1, \omega_2, \dots, \omega_R\}$ denotes the total number of linguistically defined phonemes and number of ASM units obtained, respectively. The purity of the r_{th} ASM unit is given by:

$$purity(r) = \frac{\max_p n_{r,p}}{\sum_{p=1}^P n_{r,p}} \quad (5.1)$$

Thus overall purity is given by average of the purity values of all ASM units,

$$purity = \frac{1}{n} \sum_{r=1}^R \max_p n_{r,p} \quad (5.2)$$

where, n is total number of features in all clusters.

Normalized mutual information(NMI) can be interpreted through information theory. It is defined as the mutual information between the cluster assignments and manually present labels normalized by the arithmetic mean of the maximum possible entropy of the marginals. A merit of NMI is that it does not necessarily increase when the number of clusters increases. The value of NMI also

lies between 0 and 1. It is formulated as follows:

$$NMI = \frac{\sum_{r=1}^R \sum_{p=1}^P \frac{n_{r,p}}{n} \log \left(\frac{n_{r,p}}{\sum_{i=1}^R n_{i,p} \sum_{j=1}^P n_{r,j}} \right)}{(H(\Omega) + H(\omega))/2} \quad (5.3)$$

where, the numerator represents the mutual information between Ω and ω . $H(\Omega)$ and $H(\omega)$ are given by:

$$H(\Omega) = - \sum_{r=1}^R \frac{\sum_{p=1}^P n_{r,p}}{n} \log \left(\frac{\sum_{p=1}^P n_{r,p}}{n} \right) \quad (5.4)$$

and

$$H(\omega) = - \sum_{p=1}^P \frac{\sum_{r=1}^R n_{r,p}}{n} \log \left(\frac{\sum_{r=1}^R n_{r,p}}{n} \right) \quad (5.5)$$

5.1.3 Baseline Approaches

We implemented Vector quantization(VQ) [35] and GMM Labeling [41]. We carried out these experiments on TIMIT database [50]. Silence regions were removed using manual transcriptions as they cover major part of speech segments. Number of clusters considered is 50 and 35. Vector quantization works by applying k-means clustering directly on MFCC features. In GMM Labeling technique, first a GMM is trained from all the training data and number of Gaussian components is set to be the desired number of ASM units. For each segment, we label it with the index of the Gaussian component which provides the highest likelihood of the speech segment. The number of Gaussian components considered is 1024.

The comparison of performances of these baseline approaches and our approach is shown in 5.2 and 5.1. It can be seen that graph clustering gives better performance.

Algorithm	NMI	Purity
VQ	0.1609	0.2324
GMM labeling	0.1979	0.2440
GCC	0.2077	0.2716
SC	0.1916	0.2686

Table 5.1: Comparison of various algorithms using R = 50

Algorithm	NMI	Purity
VQ	0.1588	0.2222
GMM labeling	0.1959	0.2334
GCC	0.1987	0.2794
SC	0.2206	0.2838

Table 5.2: Comparison of various algorithms using R = 35

5.2 Language identification system

Language identification refers to the automatic process that determines the identity of the language spoken in a speech sample. It is an enabling technology for a wide range of multilingual speech processing applications, such as spoken language translation [51], multilingual speech recognition [52], and spoken document retrieval [53]. It is also a topic of great interest in the areas of intelligence and security for information distillation.

We built Language identification system with three languages : Urdu, Bengali and Telugu using one of these clustering techniques. Number of clusters chosen is 50 and number of Gaussian components is 1024. Figure 5.1 shows the block diagram for Language identification system.

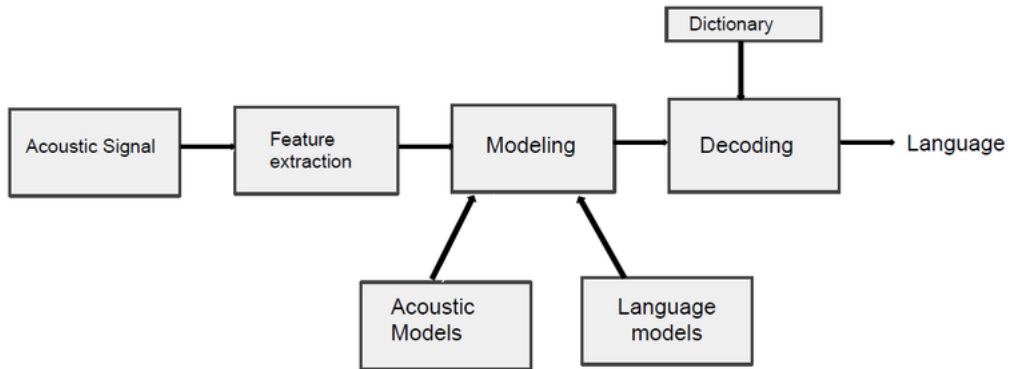


Figure 5.1: Language Identification System

For testing how effective our system is, we have chosen 60 test files from each of the language and decode it likelihood using acoustic models, language models and dictionary for these languages. We created a confusion matrix describing what number of files were correctly detected. Table 5.3 shows these numbers.

Languages	Telugu	Bengali	Urdu
Telugu	55	1	4
Bengali	6	54	0
Urdu	0	0	60

Table 5.3: Confusion Matrix

References

- [1] S. Furui. Digital Speech Processing, Synthesis, and Recognition (Revised and Expanded). *Digital Speech Processing, Synthesis, and Recognition (Second Edition, Revised and Expanded)* .
- [2] F. Brugnara, D. Falavigna, and M. Omologo. Automatic segmentation and labeling of speech based on Hidden Markov Models. *Speech Communication* 12, (1993) 357–370.
- [3] V. Kuperman, M. Pluymaekers, M. Ernestus, and H. Baayen. Morphological predictability and acoustic duration of interfixes in Dutch compounds. *The Journal of the Acoustical Society of America* 121, (2007) 2261–2271.
- [4] S. E. Tranter and D. A. Reynolds. An overview of automatic speaker diarization systems. *Audio, Speech, and Language Processing, IEEE Transactions on* 14, (2006) 1557–1565.
- [5] N. Sen, T. Basu, and H. A. Patil. New features extracted from Nyquist filter bank for text-independent speaker identification. In India Conference (INDICON), 2010 Annual IEEE. IEEE, 2010 1–5.
- [6] X. Huang, A. Acero, H.-W. Hon, and R. Foreword By-Reddy. Spoken language processing: A guide to theory, algorithm, and system development. Prentice Hall PTR, 2001.
- [7] C. Bishop. Pattern Recognition and Machine Learning (Information Science and Statistics), 1st edn. 2006. corr. 2nd printing edn 2007.
- [8] G. M. M. D. R. MIT. Lincoln Laboratory, 244 Wood St., Lexington, MA 02140, USA dar@ ll.
- [9] J. A. Hartigan and M. A. Wong. Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28, (1979) 100–108.
- [10] V. Faber. Clustering and the continuous k-means algorithm. *Los Alamos Science* 22.

- [11] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77, (1989) 257–286.
- [12] I. W. Hunter, L. Jones, M. Sagar, S. Lafontaine, and P. Hunter. Ophthalmic microsurgical robot and associated virtual environment. *Computers in Biology and Medicine* 25, (1995) 173–182.
- [13] L. Finesso, L. Gerencsér, and I. Kmecs. Estimation of parameters from quantized noisy observations. In Proceedings of the European Control Conference, ECC99, Karlsruhe, pages AM, volume 3. 1999 F589.
- [14] L. Shue, S. Dey, B. Anderson, and F. De Bruyne. Remarks on filtering error due to quantisation of a 2-state hidden Markov model. In IEEE CONFERENCE ON DECISION AND CONTROL, volume 4. Citeseer, 1999 4123–4124.
- [15] X. D. Huang, Y. Ariki, and M. A. Jack. Hidden Markov models for speech recognition, volume 2004. Edinburgh university press Edinburgh, 1990.
- [16] C. Francq and M. Roussignol. Ergodicity of autoregressive processes with Markov-switching and consistency of the maximum-likelihood estimator. *Statistics: A Journal of Theoretical and Applied Statistics* 32, (1998) 151–173.
- [17] X. Feng, K. A. Loparo, Y. Ji, and H. J. Chizeck. Stochastic stability properties of jump linear systems. *Automatic Control, IEEE Transactions on* 37, (1992) 38–53.
- [18] M.-S. Gabor. Rejtett Markov Modellek Statisztikai Vizsgálata. Ph.D. thesis, Phd Dissertation 2005.
- [19] G. E. Tusnady and I. Simon. Principles governing amino acid composition of integral membrane proteins: application to topology prediction. *Journal of molecular biology* 283, (1998) 489–506.
- [20] L. Lamel, J.-L. Gauvain, and G. Adda. Lightly supervised and unsupervised acoustic model training. *Computer Speech & Language* 16, (2002) 115–129.
- [21] F. Wessel and H. Ney. Unsupervised training of acoustic models for large vocabulary continuous speech recognition. *Speech and Audio Processing, IEEE Transactions on* 13, (2005) 23–31.
- [22] S. Novotney, R. Schwartz, and J. Ma. Unsupervised acoustic and language model training with small amounts of labelled data. In Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on. IEEE, 2009 4297–4300.

- [23] J. Glass. Towards unsupervised speech processing. In *Information Science, Signal Processing and their Applications (ISSPA), 2012 11th International Conference on*. IEEE, 2012 1–4.
- [24] C.-y. Lee and J. Glass. A nonparametric Bayesian approach to acoustic model discovery. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012 40–49.
- [25] A. Jansen, S. Thomas, and H. Hermansky. Weak top-down constraints for unsupervised acoustic model training. In *ICASSP. 2013* 8091–8095.
- [26] M.-h. Siu, H. Gish, A. Chan, W. Belfield, and S. Lowe. Unsupervised training of an HMM-based self-organizing unit recognizer with applications to topic classification and keyword discovery. *Computer Speech & Language* 28, (2014) 210–223.
- [27] A. Garcia and H. Gish. Keyword spotting of arbitrary words using minimal speech resources. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 1. IEEE, 2006 I–I.
- [28] M.-H. Siu, H. Gish, A. Chan, and W. Belfield. Improved topic classification and keyword discovery using an HMM-based speech recognizer trained without supervision. In *Eleventh Annual Conference of the International Speech Communication Association*. 2010 .
- [29] H. Li, B. Ma, and C.-H. Lee. A vector space modeling approach to spoken language identification. *Audio, Speech, and Language Processing, IEEE Transactions on* 15, (2007) 271–284.
- [30] B. Ma, D. Zhu, and H. Li. Acoustic segment modeling for speaker recognition. In *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*. IEEE, 2009 1668–1671.
- [31] M.-H. Siu, O. Lang, H. Gish, S. Lowe, A. Chan, and O. Kimball. Mllr transforms of self-organized units as features in speaker recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012 4385–4388.
- [32] Y. Zhang and J. R. Glass. Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams. In *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*. IEEE, 2009 398–403.
- [33] H.-y. Lee, Y.-C. Li, C.-T. Chung, and L.-s. Lee. Enhancing query expansion for semantic retrieval of spoken content with automatically discovered acoustic patterns. In *Acoustics, Speech*

- and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013 8297–8301.
- [34] H. Wang, T. Lee, C.-C. Leung, B. Ma, and H. Li. Acoustic segment modeling with spectral clustering methods. *Audio, Speech, and Language Processing, IEEE/ACM Transactions on* 23, (2015) 264–277.
- [35] C.-H. Lee, F. K. Soong, and B.-H. Juang. A segment model based approach to speech recognition. In *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on. IEEE, 1988* 501–541.
- [36] M. Bacchiani and M. Ostendorf. Joint lexicon, acoustic unit inventory and model design. *Speech Communication* 29, (1999) 99–114.
- [37] Y. P. Estevan, V. Wan, and O. Scharenborg. Finding maximum margin segments in speech. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on, volume 4. IEEE, 2007* IV–937.
- [38] Y. Qiao, N. Shimomura, and N. Minematsu. Unsupervised optimal phoneme segmentation: Objectives, algorithm and comparisons. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on. IEEE, 2008* 3989–3992.
- [39] A. H. H. N. Torbati, J. Picone, and M. Sobel. Speech acoustic unit segmentation using hierarchical dirichlet processes. In *INTERSPEECH. 2013* 637–641.
- [40] H. Gish and K. Ng. A segmental speech model with applications to word spotting. In *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on, volume 2. IEEE, 1993* 447–450.
- [41] H. Wang, C.-C. Leung, T. Lee, B. Ma, and H. Li. An acoustic segment modeling approach to query-by-example spoken term detection. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on. IEEE, 2012* 5157–5160.
- [42] J. Bloit and X. Rodet. Short-time Viterbi for online HMM decoding: Evaluation on a real-time phone recognition task. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on. IEEE, 2008* 2121–2124.
- [43] O. Scharenborg, V. Wan, and M. Ernestus. Unsupervised speech segmentation: An analysis of the hypothesized phone boundaries. *The Journal of the Acoustical Society of America* 127, (2010) 1084–1095.

- [44] Q. Zhu, A. Stolcke, B. Y. Chen, and N. Morgan. Using MLP features in SRI's conversational speech recognition system. In *Interspeech*, volume 2005. 2005 2141–2144.
- [45] T. J. Hazen, W. Shen, and C. White. Query-by-example spoken term detection using phonetic posteriorgram templates. In *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*. IEEE, 2009 421–426.
- [46] L. Zheng, C.-C. Leung, L. Xie, B. Ma, and H. Li. Acoustic texttiling for story segmentation of spoken documents. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012 5121–5124.
- [47] H. Wang, C.-C. Leung, T. Lee, B. Ma, and H. Li. Shifted-delta MLP features for spoken language recognition. *Signal Processing Letters, IEEE* 20, (2013) 15–18.
- [48] A. Y. Ng, M. I. Jordan, Y. Weiss et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems* 2, (2002) 849–856.
- [49] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing* 17, (2007) 395–416.
- [50] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett. DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM. NIST speech disc 1-1.1. *NASA STI/Recon Technical Report N 93*.
- [51] A. Waibel, P. Geutner, L. M. Tomokiyo, T. Schultz, and M. Woszczyna. Multilinguality in speech and spoken language systems. *Proceedings of the IEEE* 88, (2000) 1297–1313.
- [52] T. Schultz and A. Waibel. Language-independent and language-adaptive acoustic modeling for speech recognition. *Speech Communication* 35, (2001) 31–51.
- [53] C. Chelba, T. J. Hazen, and M. Saraclar. Retrieval and browsing of spoken content. *Signal Processing Magazine, IEEE* 25, (2008) 39–49.