A Soft Computing Based Approach for Multi-Accent Classification in IVR Systems

by

Sameeh Ullah

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2009

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

A speaker's accent is the most important factor affecting the performance of Natural Language Call Routing (NLCR) systems because accents vary widely, even within the same country or community. This variation also occurs when non-native speakers start to learn a second language, the substitution of native language phonology being a common process. Such substitution leads to fuzziness between the phoneme boundaries and phoneme classes, which reduces out-of-class variations, and increases the similarities between the different sets of phonemes. Thus, this fuzziness is the main cause of reduced NLCR system performance. The main requirement for commercial enterprises using an NLCR system is to have a robust NLCR system that provides call understanding and routing to appropriate destinations. The chief motivation for this present work is to develop an NLCR system that eliminates multilayered menus and employs a sophisticated speaker accent-based automated voice response system around the clock. Currently, NLCRs are not fully equipped with accent classification capability. Our main objective is to develop both speaker-independent and speaker-dependent accent classification systems that understand a caller's query, classify the caller's accent, and route the call to the acoustic model that has been thoroughly trained on a database of speech utterances recorded by such speakers. In the field of accent classification, the dominant approaches are the Gaussian Mixture Model (GMM) and Hidden Markov Model (HMM). Of the two, GMM is the most widely implemented for accent classification. However, GMM performance depends on the initial partitions and number of Gaussian mixtures, both of which can reduce performance if poorly chosen. To overcome these shortcomings, we propose a speaker-independent accent classification system based on a distance metric learning approach and evolution strategy. This approach depends on side information from dissimilar pairs of accent groups to transfer data points to a new feature space where the Euclidean distances between similar and dissimilar points are at their minimum and maximum, respectively. Finally, a Non-dominated Sorting Evolution Strategy (NSES)-based k-means clustering algorithm is employed on the training data set processed by the distance metric learning approach. The main objectives of the NSES-based k-means approach are to find the cluster centroids as well as the optimal number of clusters for a GMM classifier. In the case of a speaker-dependent application, a new method is proposed based on the fuzzy canonical correlation analysis to find appropriate Gaussian mixtures for a GMM-based accent classification system. In our proposed method, we implement a fuzzy clustering approach to minimize the within-group sum-of-square-error and canonical correlation analysis to maximize the correlation between the speech feature vectors and cluster centroids. We conducted a number of experiments using the TIMIT database, the speech accent archive, and the foreign accent English databases for evaluating the performance of speaker-independent and speaker-dependent applications. Assessment of the applications and analysis shows that our proposed methodologies outperform the HMM, GMM, vector quantization GMM, and radial basis neural networks.

Acknowledgements

The author would like to thank his supervisor Professor Fakhreddine Karray, for his support, guidance, and feedback on this research. Many thanks also go to Dr. Jiping Sun as a co-supervisor at Vestec Inc. for his support and guidance. Special thanks are also due to the members of my committee, Professors Emil M. Petriu, Otman Basir, Liang-Liang Xie, and Ali Qhodsi for their valuable comments and suggestions which contributed to improve the quality of this research. Special thanks are further due to my thesis reader Mary McPherson for taking the time and providing valuable comments.

The author also gratefully acknowledges Vestec Inc. and Amrotel Inc. for providing financial support and valuable resources during this research.

Dedication

This work is dedicated to my wife for her patience, and for providing me with a lot of support during this research work.

Contents

List of Tables

List of Figures

List of Algorithms

Abbreviations

ANNs	Artificial Neural Networks
ASR	Automatic Speech Recognition
BB	Branch and Bound
BF	Breadth-First
BP	Back Propagation
CB	Centroid-Based
CCA	Canonical Correlation Analysis
DFT	Discrete Fourier Transform
DML	Distance Metric Learning
EAs	Evolutionary Algorithms
EC	Evolutionary Computation
EM	Expectation Maximization
ES	Evolution Strategy
FAE	Foreign Accented English
FCCA	Fuzzy Canonical Correlation Analysis
FIR	Finite Impulse Response

GMMs	Gaussian Mixture Models
GSL	Grammar Specific Language Model
HMM	Hidden Markov Model
IVR	Interactive Voice Response
LID	Language Identification
LPC	Liner Predictive Coding
MAP	Maximum a Posterior
MFCCs	Mel Frequency Cepstral Coefficients
ML	Maximum Likelihood
MLP	Multilayer Perceptron
NGIVR	Next Generation Interactive Voice Response
NLCR	Natural Language Call Routing
NSES	Non-dominated Sorting Evolution Strategy
NSGA	Non-dominated Sorting Genetic Algorithm
NSGA-II	Non-dominated Sorting Genetic Algorithm-II
PB	Partitioning-Based
PCM	Pulse Code Modulation
PLP	Perceptual Linear Predictive
PSTN	Public Switched Telephone Network
RBF	Radial Basis Functions
SLM	Statistical Language Model

STM	Stochastic Trajectory Model
SVM	Support Vector Machine
TM	Telephone Microphone
VEGA	Vector Evaluated Genetic Algorithm

Chapter 1

Introduction

An Interactive Voice Response (IVR) system allows a computer to identify the words spoken by different speakers into a microphone or telephone and replies in an appropriate manner to initiate a dialog. However, it is quite easy for humans to recognize objects, letters, symbols, and the voices of their friends, etc. but very difficult for computers to do the same. The performance of an IVR system is degraded by many factors, such as the anatomy of the vocal tract [1], background noise [2][3], the transmission medium [4], and the accent of a speaker. Among these problems, a speaker's accent is the most important factor due to intra- and inter-speaker variations [5]: accents vary widely, even within the same country or community. This variation occurs when non-native speakers start to learn a second language, as the substitution of native language phoneme pronunciation is a common occurrence. Such substitutions lead to fuzziness between phoneme boundaries and phoneme classes, reduce out-of-class variations, and increase the similarities between the different sets of phonemes.

There are many variations in the structure of IVR systems [6][7][8][9], but one of the most general structures is shown in Figure 1.1. The input signal to the IVR system is converted by an electro-mechanical device (a microphone) from a physical varying value (i.e., pressure in air) into a continuously varying electrical signal. A preamplifier, filter, and analog-to-digital converter convert this signal into a sequence of values. The speech signal after this conversion is called digitized speech.

The digitalized speech is then used to extract the speech feature vectors. These feature vectors (i.e., acoustic representations) play an important role in the performance of an IVR system [3] and provide a means to separate the classes of speech sound for a robust, stable, and compact representation of the input raw speech wave. Speech feature extraction can thus be considered a data reduction process that captures the essential features of the speech signal. This data reduction basically depends on the sampling rate to digitize the speech signal for further processing. There are many variations of the features that characterize the accent of different speakers. The most efficient combination is of the phonetic and prosodic features. During the training phase, the acoustic classifier is trained on these feature vectors of different speakers. The dotted line in Figure 1.1 shows the training path for acoustic models. After a successful training of the acoustic classifiers, the extracted features are used to test an acoustic model's phoneme recognition ability–how accurately it can match extracted features from an unknown utterance. A phoneme is the basic information unit in speech processing and understanding [10]. Each phoneme match can be viewed as a local match. This process of matching leads to a global match through integration of many local matches. The global match is a result of the best sequence of words to match to the data.

As shown in the Figure 1.1, in the chain of operations, the front-end processing (i.e., feature extraction) and decoder/recognizer are the most significant components of any IVR system. Most of the recognizers are based on statistical models, such as the Hidden Markov Model (HMM). The HMM Model is defined as a stochastic finite state model and depends on a finite set of possible states. Each state of the HMM has a specific probability distribution. The number of HMM models depends mainly on the number of phonemes in the database. During the phoneme recognition phase, one or more HMM states are used to model the speech segment. This phoneme recognition leads to word and sentence recognition through use of

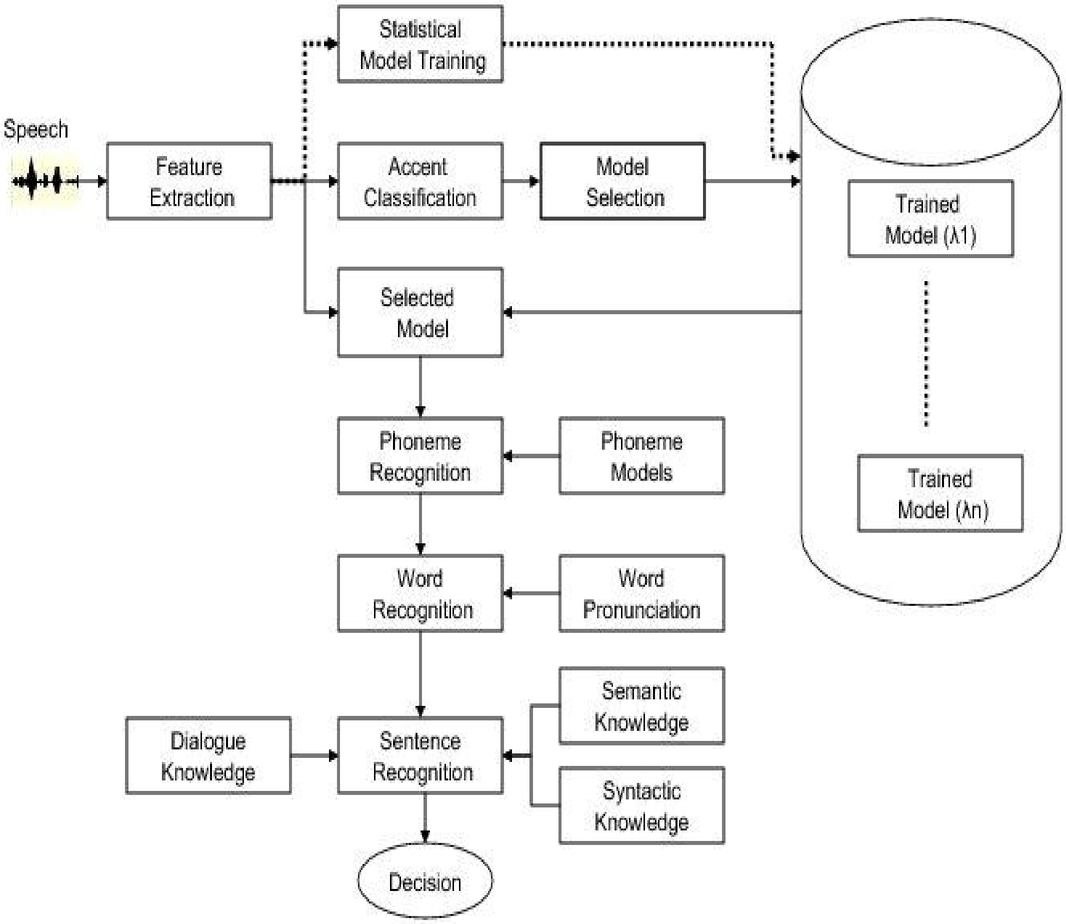


Figure 1.1: Accent-based IVR system architecture

a viterbi algorithm with word and sentence models. The word model (dictionary) consists of words and phoneme models. The sentence model (grammar) is developed from the application domain database and is a combination of words constrained by a grammar. Two approaches have been used to generate such grammars: the Statistical Language Model (SLM) [11][12] and the Grammar Specific Language Model (GSL). The SLM is used primarily for large applications, while GSL is used for small ones.

This thesis investigates the application of an efficient optimization method known as Non-dominated Sorting Evolution Strategy (NSES) to improve the performance of k-means clustering algorithms. The main problem with the k-means clustering approach is that it is not possible to determine the exact number of partitions that are suitable for a particular database. The experiments must therefore be repeated several times to achieve the number of cluster centroids that are suitable as initial seeds for a Gaussian mixture model. In addition to this lack of automated way for generating the number of partitions, k-means algorithms are often trapped into local minima/maxima, depending on the nature of the objective function. We propose an NSES-based methodology that has the capability to overcome these deficiencies of k-means clustering algorithms. We also investigate the impact of learning class inequality side information on clustering and classification of a speaker's accent in an IVR system.

This chapter provides an overview of an accent-based IVR system. It also includes the motivation behind this research on accent classification. Next, goals of this study are presented, followed by the contributions and organization of this thesis.

1.1 Motivation

In real world applications, many factors affect the performance of IVR systems: inter- and intra-speaker variability, microphone distortions, transmission line characteristics, and background noise. The most important factor is inter-speaker variations, which are due to the anatomy of the vocal tract, which varies from speaker to speaker [13]. The performance of accent-based IVR systems is affected mainly by this factor. It is nearly impossible to devise a method or model that can completely resolve the issue of vocal tract variations.

The performance of speech recognition systems has recently been greatly improved by training HMM [14][15][16] with large speech databases that contain speech utterances spoken by many different speakers, and by incorporating statistical models of speech variations. It is still impossible, however, to accurately recognize the utterances of every speaker recorded by microphone or transmitted over communication media. The performance of an automatic speech recognition system is usually degraded by the wide variations in accent, even within the same country or community. Speaker accent is therefore becoming an important issue due to globalization and the widespread use of telecommunication services in everyday life. Speaker accent classification has numerous applications, such as in security services, criminal investigation, financial institutions, and natural language call-routing systems.

The application of foreign accent classification in call centres is in high demand. The goal of many enterprises is to have an automated natural language call-routing system to reduce their reliance on human agents [17]. Since service industries receive a large number of calls every day, the cost of hiring workers to route calls to their appropriate destination is very high. This large volume of calls leads to a need for automatic call-routing systems, i.e., for a machine to replace human agents to perform the call-routing task. However, poor accent identification tremendously degrades the performance of natural language call-routing systems. Speech is a complex and stochastic process with many overlapping and unique elements. The overlapping characteristic of speech makes it very difficult to accurately distinguish the phoneme boundaries and phoneme classes. The fact is, speaker accent classification remains a complex and challenging speech recognition research topic. Better accent classification techniques are needed to make applications robust and to improve the performance of IVR systems. In other words, the purpose of accent identification is to improve the performance of speech recognition systems in a multicultural application domain.

Although accent identification is a new field as compared to speaker identification and speaker recognition, it has been studied for some years, and many promising results have been achieved. Difficulties associated with it continue to pose a challenge in the areas of phoneme clustering, classification, and optimal means of extracting the feature vectors of a speech signal. The work detailed in this thesis is motivated by a desire to enhance the performance of IVR systems by incorporating an improved accent classification approach.

1.2 Goals

Speaker accent classification is among the most difficult and challenging research issues that influences the performance of IVR systems [18][19][20][21][22][23]. Our main goal in this thesis is to enhance the performance of automated speech recognition systems by improving accent classification. Three specific goals motivate this work.

First, we explain the factors that contribute to the fuzziness between phoneme boundaries and phoneme classes. Further, we explain inter-language confusability among different accent groups, their similarities and dissimilarities.

Secondly, we derive acoustic features. Though there is no strong indication of which feature is the most suitable to improve the performance of IVR systems in the area of accent classification, we have evaluated different features (i.e., MFCC, formants, pitch, and energy) and have proposed a new combination of features.

Thirdly, we outline our proposed method for speaker-independent accent classification based on a distance metric learning approach and evolution strategy. In addition, we also present a speaker-dependent accent classification system based on fuzzy canonical correlation analysis.

1.3 Main Contributions

The main contributions of this thesis can be described as:

- Improving the performance of k-means clustering algorithm based on evolution strategy
- Providing a systematic way by which one obtains an optimized Gaussian mixtures without repeating the experiments several times.
- Improving the performance of a speaker-independent IVR system based on a

GMM classifier by using a distance metric learning approach and NSES-based k-means clustering

- Improving the performance of a speaker-dependent IVR system based on a GMM classifier by using fuzzy canonical correlation analysis

1.4 Thesis outline

In Chapter 2, a detailed survey of current accent classification systems is provided. Section 2.1 provides details of speech production mechanism. This section is followed by feature extraction techniques in the area of accent classification. In this section, we discuss front-end signal processing techniques, such as pre-emphasis of a speech signal, framing and windowing, and spectral analysis. In Section 2.3, we explain different features for an accent classification system: phonetic features and prosodic features. Section 2.4 provides classification techniques, such as statistical and artificial neural networks. In Section 2.5, we provide a survey on support vector machines classifiers. This section is followed by a brief review of other classification techniques. Section 2.7 describes the problems of foreign accent databases. Finally, important conclusions are drawn in Section 2.8.

Chapter 3 reviews the problem addressed in this thesis in detail. In Section 3.2, we propose an architecture for a next generation IVR system that is a complete end-to-end speech-enabled system. Section 3.3 describes the core factors that really degrades the performance of accent classification systems. In the Section 3.4, we present accent-based ASR systems and explain the contribution of different modules, speech features, and a hybrid feature selection scheme to increase the performance of the system. Section 3.5 describes our proposed approach to improve the performance of next generation IVR system. Next, we provide introduction, theoretical foundations, and a graphical overall representation of the proposed method. This is followed by the distance metric learning module. In Section 3.5.5, we first present a brief overview of the evolution strategy, its forms, and pseudo code. This section is followed by a related work in the area of non-dominated sorting multi-objective evolutionary algorithms. In the Section 3.5.8, we provide a non-dominated sorting evolution strategy. This section is followed by an accent classification module and decision making and then a acoustic model switching module. Finally, we present a summary of this chapter.

In Chapter 4, we provide our proposed methodology for speaker-dependent accent classification for IVR systems. In Section 4.1, we describe a fuzzy clustering approach. Section 4.2 describes canonical correlation analysis in detail for speaker-dependent accent classification systems. In this section, first we provide introduction to canonical correlation analysis. This is followed by the theoretical foundations of the canonical correlation analysis and its derivation. Section 4.3 describes our proposed approach for a speaker-dependent next generation IVR system and fuzziness between phoneme boundary and phoneme classes. In this section, we first describe the fuzzy canonical correlation analysis-based accent clustering approach. In this section, we also explain a Gaussian mixture model-based accent classification system for speaker-dependent applications. Finally, important conclusions are drawn.

Chapter 5 provides assessment of the applications and analysis. In this chapter, we conducted experiments with our proposed approaches: speaker-independent and speaker-dependent accent classification systems. In Section 5.1, we present an evaluation of our proposed approach for a speaker-independent accent classification system using the TIMIT database. This is followed by evaluation of the proposed approach using the speech accent archive database. In this section, we show the experimental results for a speaker-independent accent classification system using English Arabic vs. American English, English Arabic vs. English Chinese, and English Arabic vs. English Russian. Section 5.1.3 shows experimental results of a speaker-independent accent classification using the foreign English accent database. In Section 5.2, we present assessment and analysis of a speaker-dependent accent classification application. We first present classification results of the proposed approach for a speaker-dependent application(fuzzy canonical correlation analysis) and a Gaussian mixture model using a standard k-means clustering approach with the TIMIT database. This is followed by evaluation of the proposed approach using the speech accent archive and foreign accent English databases. Finally, important conclusions are drawn.

In Chapter 6, conclusions are drawn and future directions for the work are provided. We also discuss further improvements to the non-dominated sorting evolution strategy.

Chapter 2

Background and Literature Review

The main purpose of this chapter is to give a comprehensive overview of techniques and systems in the field of accent classification. It is divided into eight sections. Section 2.1 gives an overview of a speech production system. Section 2.2 provides a detailed investigation of the current and previous approaches used to extract the features for accent classification systems. We discuss the features for speaker accent classification in Section 2.3. In Section 2.4, basic classification techniques in the context of accent identification are addressed. In this study, we organize them into two major categories: statistical based and artificial neural networks based. Under the umbrella of statistical classification techniques, we discuss the Hidden Markov model, the Gaussian mixture model, and vector codebooks. This section is followed by a discussion of two neural network classification techniques: radial basis neural network and multilayer perceptron. Support vector machines and other classification methods are briefly discussed in Section 5 and 6, respectively. Section 7 addresses the problems of foreign accent databases. Finally, important conclusions are drawn.

2.1 Speech Production

Speech is produced as a result of air pressure generated in the mouth and coordinated movements of the human auditory system. The most important human organ affecting the production of speech is the vocal tract. Its anatomy varies from speaker to speaker just as fingerprints vary from person to person [7]. Accent identification/classification is therefore a difficult problem. The process of speech communication between humans as well as automated agents involves the production of acoustic waves.

The main components of the human articulatory system are the lips, teeth and jaw, tongue, velum, nasal (nose) and oral (mouth) cavities, pharyngeal (throat) area, larynx, trachea, and lungs. In the production of speech, the role of the vocal and nasal tracts is very important for delivery of recognizable acoustic waves. The vocal tract is composed of the pharyngeal and oral cavities and the nasal tract. The nasal tract is composed of the nasal cavity. From the technical and signal processing point of view, the production of speech signals involves three steps: sound is first initiated, then filtered, and finally fine-tuned [9]. To initiate sound, lungs provide the source of energy for speech in a non-breathing pause. For speech production, during the non-breathing pause, air is expelled into the trachea. The resulting air pressure is excited when it passes through the larynx, leading to the periodic excitation of the speech signal to produce voiced sounds. Next, for filtering, the vocal and nasal tracts play an important role as the main acoustic filter. This filter shapes the sounds that are generated and excited by the larynx and lungs. Now the speech signal is ready for fine tuning and adjustments. For fine tuning, the tongue and lips are the main components. In addition, the teeth, jaw, and velum play a significant role. These components are known as the articulators as shown in Figure 2.1 (adapted from http://ispl.korea.ac.kr/ wikim/research/speech.html).

The resulting speech can be classified into voiced, unvoiced, plosive, mixed, silence, and whisper. A particular speech signal is then generated by the combination of these categories of sound. The produced speech is composed of phonemes.

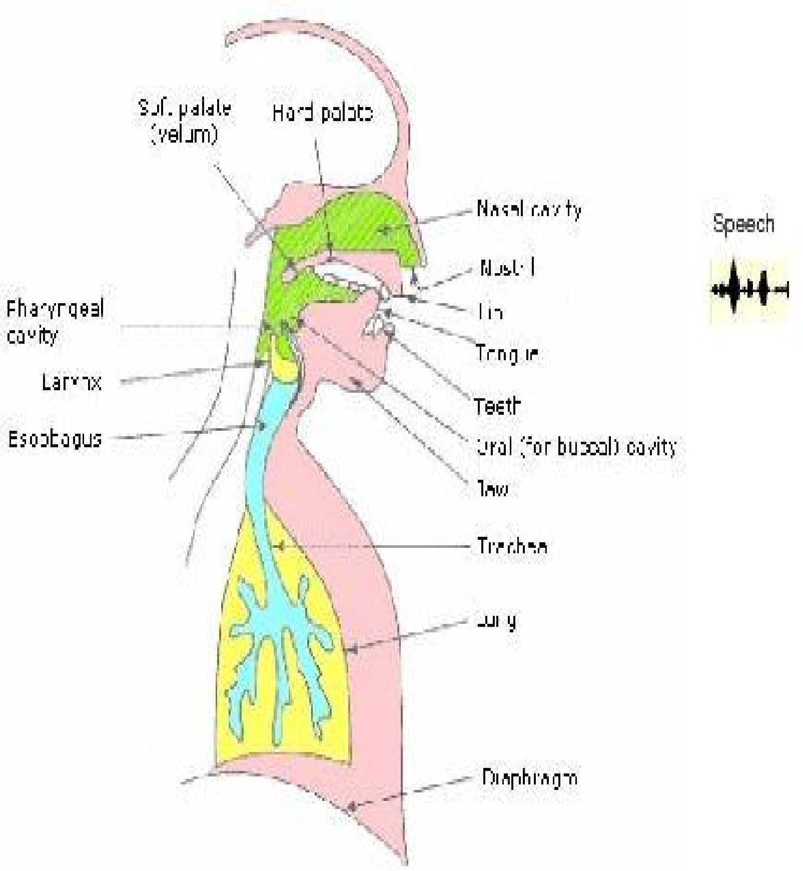


Figure 2.1: Speech production mechanism

The number of phonemes varies from language to language. For instance, standard American English consists of about 44 phonemes. These phonemes can be classified as vowels, semivowels, diphthongs, and consonants.

In most languages, phonemes can be classified as vowels and consonants. The periodic vibration of vocal cords in the larynx is the main means to produce vowels. This vibration of vocal cords results in a fundamental frequency or pitch of speech. Furthermore, in addition to fundamental frequency, the vocal tract generates resonant frequencies, which are called formants or formant frequencies. Normally, formant ranges from $f1 \sim f5$ have information that characterizes an accent, but good accent classification results can be achieved using the first three formants. One can easily distinguish between the formants in the spectrum of speech signals because most of the energy is concentrated in them. The concentration of energy in formants depends on the length and shape of the vocal tract. This explains the fact that women have higher formant frequencies than men, as their vocal tracts are shorter. Voiced phonemes are highly periodic, while unvoiced phonemes are rather stochastic.

2.2 Feature Extraction

The extraction of suitable features is a process of particular significance in a speech recognition application. Any ASR system, regardless of the task - be it speech recognition, speaker identification, speaker recognition, or accent classification - is composed of both training and evaluation modes. Both these modes include feature extraction. The feature extraction process in an ASR system is sometimes called front-end processing. There are many techniques by which we can extract speech features [24]. The process of feature extraction converts the digital speech into a sequence of numerical descriptors that are called feature vectors. The elements of feature vectors provide a robust, more compact and stable representation than does a raw speech input signal.

This section is organized as follows: it first considers the importance of front-end signal processing in an ASR system, followed by discussion of the pre-emphasis of a speech signal and how it is significant for feature extraction. Next, framing, windowing, and spectral analysis are examined from the front-end processing point of view.

2.2.1 Front-end Signal Processing in an ASR System

The performance of a speech recognition system based on accent depends mainly on its components, such as feature extraction and accent classification modules. The whole process is composed of accent classification, model selection, and speech recognition.

Feature extraction is the first step in any speech recognition system. It is believed that suitable features may enhance its performance. There are many variations in front-end processing to extract these features. The general procedure can be summarized in a block diagram, as shown in Figure 2.2.

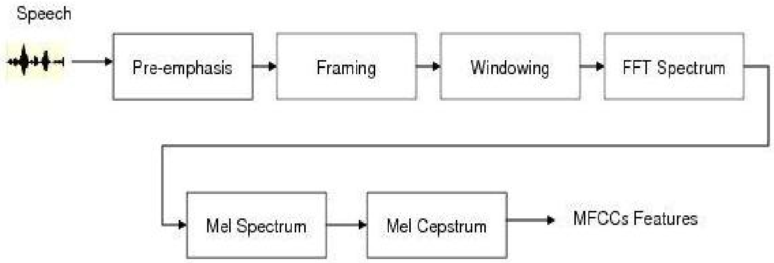


Figure 2.2: Mel-frequency features

The factors that affect the process of feature extraction are acoustic and transduction equipment, such as a microphone, preamplifier, filtering, and analog-to-digital converter. The analog speech signal is digitized before further processing. During the digitization step, the sampling frequency must be twice that of the highest fundamental frequency. This condition is imposed by Shannon theory in order to allow full recovery of the analog signal:

$$f_s = 2f,$$

where f_s is the sampling frequency, and f is the frequency of the input speech signal.

Speech signals that are easily understandable have a frequency range of 250 $\sim$ 8000 Hz. The sampling rate for this speech signal should be 16 KHz, a frequency domain normally used for microphone data. However, telephony speech has a frequency spectrum between 250 $\sim$ 4000 Hz. The band pass filter in Pulse Code Modulation cuts the frequency of the signal above 4 kHz. Therefore, in telephone speech, the sampling frequency should not be more than 8 kHz. This lower frequency causes telephone speech signals to be degraded in quality, and the performance of ASR systems is consequently affected by this band limitation of the speech signal.

2.2.2 Pre-emphasis of Speech Signal

Pre-emphasis is a process in which a filter is applied that increases the energy of the high frequency spectrum. The pre-emphasis filter is identical in form to lip radiation characteristics. From the speech production model, there is -6 dB/octave decay in voiced-speech signal as frequency increases. This phenomenon happens even though the strength of the speech emitted by the speaker is the same. Thus, the same power of speech signal is achieved at high and low frequencies.

The pre-emphasis filter can be represented in a z domain as

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

Similarly, it can be represented in a time domain as

$$x_n = x_n - \alpha x_{n-1},$$

where α is a constant and falls generally in the range of $0.9 \leq \alpha \leq 1$. The precise value of α seems to be problematic, because in the case of unvoiced speech signals the value of α is zero. Thus, the pre-emphasis filter is a high frequency filter that does not suppress the signal in the low frequency domain, but rather achieves or maintains a gain in the higher frequency domain. This effect can be clearly seen in Figure 2.3, with $\alpha = 0.95$.

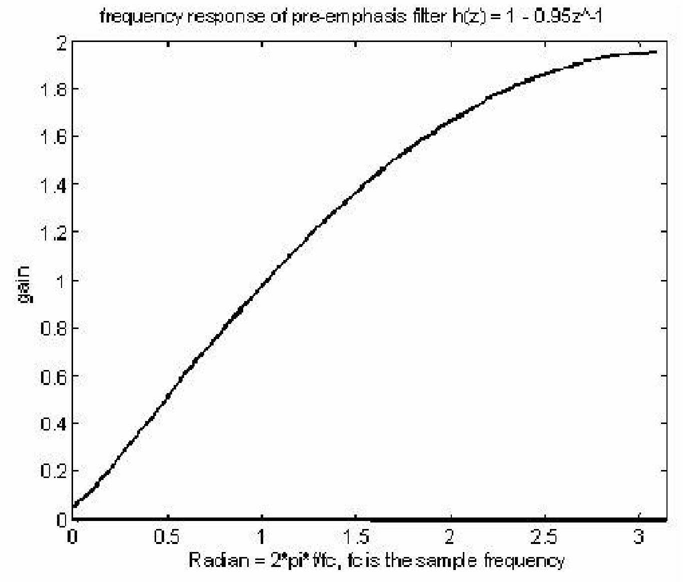


Figure 2.3: Pre-emphasis of speech signal

The main function of the pre-emphasis filter is to flatten the signal, giving a spectrum, which consists of formants of the same height. Doing so allows the Mel spectrum and Liner Predictive Coding to accurately model speech signals.

2.2.3 Framing and Windowing

Speech analysis and processing is a challenge because of variability in the speech signal which occurs for each sound and is non-stationary. Speech analysis techniques must therefore be performed on short windowed segments; the speech signal is segmented and grouped in a set of samples, called frames. The duration of each frame typically varies from 10 to 30 ms, making the speech signal appear almost stationary within the frame [9]. The variability characteristic of the speech signal must be maintained, and for a smoother feature set over time, each successive frame must overlap the next. This overlapping factor varies from application to application; normally it is considered in the range of 30 to 50 percent of the frame length. The next step is windowing, which involves multiplication of the set of frames by a finite-duration window.

The role of windowing is important when frames are transformed to the frequency domain by use of Discrete Fourier Transform (DFT). During the transformation, a frequency leakage occurs, which reduces the performance of the signal. One solution among many to deal with this leakage is windowing.

In speech processing, one of the most important windows often used in an ASR is the hamming window, which is defined so:

$$h(n) = \begin{cases} \alpha - (1-\alpha)\cos(2\pi\frac{n}{N-1}), & 0 \le n \le N-1 \\ 0 & otherwise \end{cases} \tag{2.1}$$

where $\alpha = 0.54$ and N is the window length.

To compare the performance of hamming and rectangular windows, Figure 2.4 shows a spectrum of sine waves with a frequency of 30KHz. We applied hamming and rectangular windows to the signal. It is clear that the hamming window has smaller side lobes than the rectangular window. Thus, the hamming window causes less frequency leakage than the rectangular one.

2.2.4 Spectral Analysis

Spectral analysis is the most important module in front-end processing and has a great impact on the performance of an ASR system. The quality of the features depend primarily on this module. In this module, features can be calculated by time domain methods or by frequency domain ones. In a time domain, extraction of features from each frame has the advantage of simplicity, easy physical interpretation, and quick calculation. The features of a speech signal that are calculated in a time domain are as follows:

- Pitch period

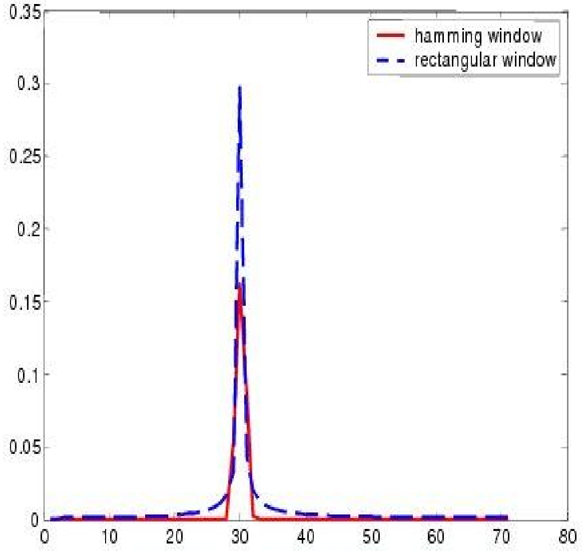


Figure 2.4: Rectangular window vs. hamming window

- Short-time average energy and amplitude
- Short-time autocorrelation
- Root-mean-square
- Maximum amplitude
- Difference between maximum and minimum values in the positive and the negative halves of the signal
- Autocorrelation peaks

The feature extraction process in a time domain has advantages over the feature extraction process in a frequency domain.

In general, time domain features are less accurate than frequency domain features [7]. Thus, the most useful of feature extraction methods are available in a frequency domain. These include Finite Impulse Response (FIR), filter-bank, and LPC. Due to their effectiveness, many variations of these techniques have been developed, such as the Mel-scale FFT filter-bank and Perceptual Linear Predictive.

2.3 Features for Speaker Accent Classification

An important issue in the design of speaker accent speech recognition systems is the proper extraction of acoustic features that efficiently characterize different accents. The proper choice of acoustic features strongly affects the performance of the speech recognition system. It is also believed that the overall performance of ASR systems is highly dependent on the quality and robustness of the speech features [3]. Acoustic features can be divided into two main categories: phonetic features and prosodic features. Prosodic features have proved to be the key factor in human perception. On the other hand, in the speaker identification task, phonetic features (i.e., MFCC features) are considered best for speaker recognition and identification. There is still a debate over which features are best for accent classification/identification. However, in [25], Waibel has provided evidence that a combination of prosodic and phonetic features may improve recognition performance.

The phonetic features of speech are further divided into non-parametric and parametric representation. The non-parametric includes MFCC and IMELDA. The parametric includes LPC, SMC, PLP, and RASTA. In this chapter, we discuss only MFCC features in the context of accent classification. The prosodic features can be further divided into formants, intonation, pitch, energy, lexical stress, and emotions. Under the category of prosodic features, we briefly discuss all these features because they are supposed to be stronger candidates for accent classification than phonetic ones.

2.3.1 Phonetic Features

A. Mel Spectrum

Mel is a unit of measure of the perceived pitch or frequency of a tone [7]. Mel-scale frequency coefficient components are based on a short-time spectrum. The basic principle on which MFCC features are based is the combination of linear and non-linear (i.e., logarithmic) frequency representation. The Mel-scale has a linear spacing below 100 Hz and logarithmic above 1000 Hz. Therefore, MFCC features are based on the human perception mechanism. The human ear is able to hear sound tones with a linear scale below 1 kHz, and a logarithmic one above 1 kHz. The relationship between the Mel-scale and linear scale is shown below:

$$mel(f) = 2595 * \log 10(1 + \frac{f}{700}). \tag{2.2}$$

The mel-scale is linear below 100 Hz and logarithmic above 1000 Hz can be easily seen in Figure 2.5.

To achieve an approximately equal resolution on a mel-scale, we usually use a filter-bank. The overlapping of triangular filters, as shown in Figure 2.6, plays an important role in this process. Further detailed derivations are shown in Appendix A.

2.3.2 Prosodic Features

Prosodic features have been proven to be a key factor in human perception, but generally, accent recognition is a new field, and there are no strong recommendations regarding which features are best. In [25], it is shown that the combination of prosodic and phonetic features could improve recognition performance [26][27][28].

In our proposed study, we investigate such combinations of phonetic and prosodic features. The prosodic features include formants, intonation, pitch, lexical stress, and rhythm.

In [29], Gajjic and Paliwal introduced an accent classification scheme based on HMM to analyze the performance of LPC, MFCC, and Formants. It is found that phonetic features (i.e., MFCC and LPC) are less robust in noise than prosodic

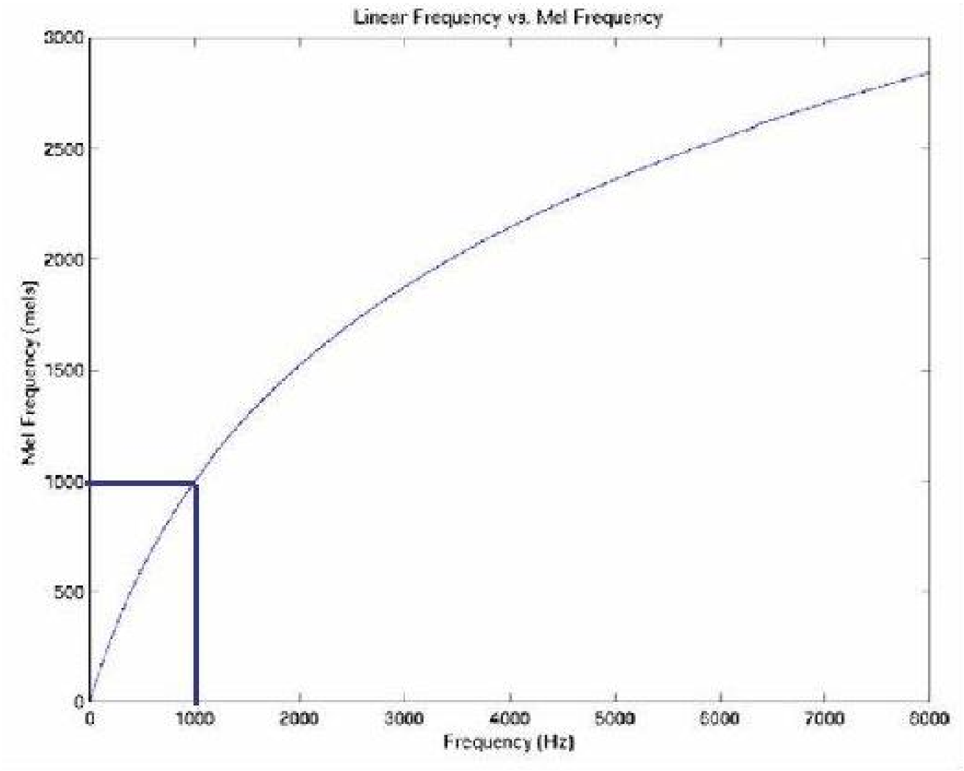


Figure 2.5: Mel-frequency vs. linear frequency

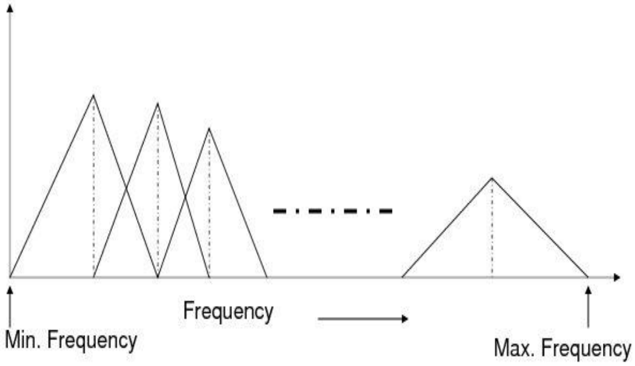


Figure 2.6: Mel-scale filterbank

features (i.e., Formants). Furthermore, in a noise-free environment, the proposed features exhibit only a slight decrease in performance. Here, no comparison is given with RASTA features, which are felt to be robust in a noisy environment [24]. In [23], Yan and Vaseghi analyzed the formants of three major English accents: British, American, and Australian, and concluded that the second formant is the most dynamic and influential formant for conveying accent information. In [30], Arslan and Hansen pointed out that the second formant (F_2) and the third formant (F_3) in native-speakers play important roles in conveying accent-related information because they cause problems with detailed tongue movements. In [31], Fujisaki *et al.* pointed out that the fundamental frequency contour (f_0) is known as a main acoustic feature to discriminate between the accents of speakers of Swedish and all other Scandinavian languages. In [32], Burnett and Pary pointed out that the first three formants give an indication of speaker vowel behavior. As mentioned in the literature, most accent-related information is in the vowel. Unfortunately, vowels are more often distorted than consonants in accented speech.

A. Intonation

Intonation is defined as the rise and fall of the pitch of the voice in speech. It may also be defined as the modulation of the voice doing the speaking. Thus, it is not difficult to distinguish whether individuals are speaking English or another language in a crowd. Generally, intonation is used to convey the emotions of a speaker speaking to humans or with an automated agent.

Many factors affect intonation and contribute to intonation style, the environment, topic of discussion, etc. Intonation often changes in response to who is being spoken to; for example, in the case of an adult talking to a high school student or an employee talking to his/her manager.

In [33], intonation permits speakers to imply the opposite of actual words spoken. For example, the answers to the question, “Did you enjoy last night’s party?” tell whether the speaker had a “great” time or he/she just enjoyed the party. The overall level of information depends on how the speaker says the word great. Furthermore, in Mandarin Chinese, intonation changes the complete meaning of words. This feature is not available in the English language. It is further added that intonation has a very important role in speech communication, but the difficulty is that it is context-dependent.

B. Lexical Stress

Lexical stress may be defined as the degree of vocal tract forces used to articulate a syllable, a word, or a phrase in a sentence. This results in applying stress in such a way as to change the meaning of a sentence [33].

Stress is a very important factor in the English language, and proper placement of stress is an essential part of the word shape. Stressed words in a sentence tend to be perceived as more prominent than non-stressed sounds. The fact is that lexical stress in English depends mainly on pitch, duration of sound, and possibly on vowel quality. In any sentence, stress is an important factor for making an allophone transcription. The allophone variation can depend on the place of lexical stress placed on the various allophones.

C. Lexicon Rhythm and Length

Rhythm in the English language is based on stressed syllables [33]. We may divide the speech utterance into groups of syllables. These syllables may be stressed or non-stressed syllables. Generally, each speech utterance contains one and only one stressed syllable. For example, in the terms *man hours* and *manpower hours*, man is a stressed syllable in both sentences. The */man/* syllable is shorter in length than manpower. This variable duration of phonemes or word can be used for linguistic purposes.

For example, the sound */n/* in "fantastic" can be lengthened to convey extra meaning. The rhythm may pose the variation in length such as */bend/* and */bent/*. The */n/* is longer in length in */bent/* as compared to */bend/*. This variation in phoneme length makes accent classification difficult.

2.4 Classification Schemes

Various classification techniques have been used in the literature for accent classification. Among the well-known ones are statistical classifiers, Support Vector Machine (SVM) based classifiers, and Artificial Neural Networks (ANNs) based classifiers. In truth, there is no strong recommendation of or agreement on which classifier is the most suitable for accent classification, perhaps because this is a new field in speech recognition, unlike speaker recognition or identification. The existing classifiers have their own merits and demerits for accent classification. To achieve strong classification results, it is necessary to combine the merits of more than one classifier or clustering techniques.

In previous studies, HMMs and Gaussian Mixture Models (GMM) have been the most popular approaches to accent classification, and most proposed accent classification systems have incorporated these structures.

The main objective of this section is to give an overview of different classifiers used in accent classification–HMM, GMM, SVMs and Radial Basis Functions (RBF)–and their advantages and disadvantages. These classifiers can be broadly divided into two main classes: Statistical-Based Classifiers and Connectionist-Based Classifiers.

2.4.1 Statistical-Based Classifiers

In the statistical-based classification approach, each class of data is modeled by the probability distribution determined during the training phase. The probability distribution depends mainly on the training data.

The statistical classification approach is the most popular in ASR applications and the reason that HMM is widely used for accent classification and in other speech recognition applications. On the other hand, the GMM is considered a state-of-the-art classifier for speech applications such as speaker verification and speaker identification.

Statistical classifiers have a solid mathematical basis and are simple to implement. There are many methods for training statistical classifiers, such as Maximum a Posterior (MP) and Maximum Likelihood (ML). In addition to these, statistical classifiers have very strong optimization algorithms during their training, such as the Expectation Maximization (EM) algorithm. However, statistical classifiers have some limitations, such as a lengthy training period. The other drawback during the training phase is that they need proper initialization for model parameters [34].

Here we discuss HMM and GMM in the context of accent classification.

Hidden Markov Model

HMMs are well known in the area of speech recognition and machine intelligence, and are extensively used in accent classification systems due to their ability to cope with speech variations by means of stochastic modeling [35][36]. HMM is, moreover, physically related to the speech production mechanism. As in the speech production mechanism, the speaker does not know the interaction of human organ states and the human neural network. We may say that states for the production of speech are hidden. This same principle applies in HMM modeling.

The most important design issue in accent-based speech recognition systems is the choice of a good classifier. The classifier should be able to efficiently characterize the underlying properties in different accents. Thus, HMM-based accent classification systems have been extensively studied and implemented in speech recognition tasks such as speaker verification, speaker identification, phoneme recognition, and language identification.

The most important property of HMM that makes it a strong candidate for classification and recognition tasks is the ability of its hidden states to capture the temporal structure of training data. According to the application domain, two well-known HMM topologies (i.e., ergodic and left-to-right) are generally used in speech applications [37].

Generally, HMM provides classification accuracy that is comparable to that of other classifiers. Due to the popularity of HMM in speech recognition applications, researchers often try to implement HMMs for accent identification. In [38], Teixeira *et al.* introduced an accent classification approach based on a parallel set of ergodic nets with context-independent HMM units. In this approach, the ergodic topology was also substituted by pronunciation transcriptions. Basically, a hierarchical classification approach is presented that includes three steps: speaker gender identification, classification of speaker accent, and finally, selection of an unknown speaker by the recognizer module. The experimental results shown are very good for gender identification (i.e. 94%) but less so for accent identification (i.e. 74%). In [21], Hansen and Arslan introduced a speaker accent classification scheme based on HMM codebooks generated after training. Prosodic features are analyzed, and it is claimed that they have an impact on accent classification. The classification results for unknown text and known text are 81.5% and 88.9% respectively. In [19], Kat and Fung introduced a phoneme class HMM-based method for fast accent identification. The accent classification results are no more accurate than those from other methods, but it is claimed that the proposed method is faster. In general, Energy, Formants, and fundamental frequency information are found to be the most discriminative features for identifying possible accents. In [39], Berklin *et al.* introduced a method to improve the performance of accent identification systems by incorporating English structure knowledge. Accent identification improved from 86% to 96% in the case of English native speakers vs. speakers whose mother tongue was Vietnamese and 78% to 84% for English native speakers vs. speakers whose mother tongue was Lebanese.

In [40], Angkititrakul and Hansen introduced an accent classification technique based on the Stochastic Trajectory Model (STM) for each phoneme to classify different foreign accents. The proposed method in this approach outperforms both HMM and GMM by a small margin. The classification rate with the proposed method is 67% and with GMM, 66%. In [41], Kumpf and King introduced an automatic accent classification scheme based on accent-specific HMMs and phoneme bi-gram language models. In this approach, first an HMM model is trained with specific accent phoneme classes and then is used to segment the accented speech to further train the recognizer. This approach has an edge in the automatic labeling of accent-specific pronunciation dictionaries. The best accent classification rates reported in this paper are 85.3% and 76.6% for the accent pair and the three classes of accent, respectively. In [42], Humphriest *et al.* used a pronunciation dictionary to adapt speaker accents. They first derive a pronunciation dictionary using a decision tree to model all pronunciation variations. Next, the HMM model is trained on London and South-East England speakers in the training phase. In the recognition phase, models are adapted with the pronunciation dictionary for the recognition of Lancaster and Yorkshire-accented speakers. It is claimed that the addition of an accent-specific dictionary can reduce error rates by almost 20%. In [43], Goronzy and Eisele also proposed a method to generate non-native pronunciations automatically and used them for accent adaptation to improve the performance of an accent-based ASR system. In [44], Yoshimura *et al.* introduced mora HMMs to identify the isolated accented words. In this study, mora is considered as a unit of accent information. It is indicated that mora HMMs using automatically extracted features are useful in identifying accented word patterns. In this study, it is pointed out that, for multi-speaker environments, larger size codebooks (i.e., 256) give better accent identification rates, and for speaker-independent environments, smaller codebooks (i.e., 128) give better results. The identification rates indicated are 84.9% and 74.1% for multi-speaker and speaker-independent, respectively. In [45], Arslan and Hansen analyzed the impact of selective training on the performance of an HMM accent identification systems. It is claimed that by applying selective training, a 9.4% improvement could be had in the accent classification error rate. In [46], Wang *et al.* introduced a multilingual speech recognition system for Mandarin, Cantonese, and English based on HMM and found that adapting the ASR system imporoves its performance. This proposed work is based on isolated words, and error reduction (i.e., 40%) is not the same as in a continuous speech recognition system. In [47], Yang *et al.* introduced an acoustic model adaptation method based on HMM and MLLR. It is shown that accent-dependent HMM and MLLR can reduce error rates and improve the performance of ASR systems.

Gaussian Mixture Model

Speech signals are stochastic in that a particular sound, such as any phoneme (e.g., /axr/), is never pronounced exactly the same by the same speaker. This variability by means of stochastic modeling is shown [48] by a multivariate Gaussian mixture probability density function (pdf).

A Gaussian mixture model can be considered a special form of continuous HMM which has only one state [48]. However, the training and testing of GMM is faster than that for HMM due to one state variations. Thus, GMMs are more suitable for accent classification and speaker recognition and identification, even though HMMs are also extensively studied and implemented for speech recognition applications.

Despite the successful implementation of GMMs in speech application, GMMs have some limitations. GMMs cannot model the temporal structure of the training data. The underlying assumption is that all training and the testing algorithms developed so far are based on all vectors being independent. Another problem with GMMs is the initialization of the Gaussian mixture components [49]. We did not know the exact number of Gaussian components necessary for the optimization of GMM.

In [50], a number of experiments are conducted and show that 32 Gaussian mixture components is best. It is also shown that 64 Gaussian components outperform the 32 Gaussian components. It is further suggested the 64 Gaussian components take more training time than the 32. Basically , a trade-off exists between training time and accuracy, the number of Gaussian components depending on the amount of data to be trained.

After HMMs, most other work on accent classification is based on GMMs. In [51], Lin and Simske introduced a phoneme-less hierarchical accent classification technique based on GMM. The classification technique in this paper is actually based on two stages: in the first stage, two models are trained, one for the accents of male speakers, and the other for the accents of female speakers. The main purpose of the first stage is to recognize the speaker's gender. In the second stage, the accent classification is invoked using a selected accent model based on GMM. It is claimed that a hierarchical classification scheme is better than a direct classification scheme (i.e., one without gender identification). The accent classification accuracies based on a GMM classifier provided in this paper are 81.5% and 83.8% for direct and hierarchical classification schemes, respectively. Furthermore, it is also proved by experimental results that accent classification is a more difficult problem than gender identification. It is also suggested that boundaries between different accents can be very fuzzy. One problem with the approach presented in this paper is that 64 Gaussian components were used. It is normally recommended that 32 Gaussian mixture components are better as a trade-off between accuracy and training time. Furthermore, sufficient training data are used to model the variations of different speakers and it is more or less the same as the classification scheme presented in [50].

In [50], Chen *et al.* introduced an accent classification scheme based on GMMs. The classification structure is the same as that discussed in [51], apart from the addition of some experimental results to explore the effect of Gaussian components in GMM on accent identification. It was determined that 3 to 5 utterances are sufficient to allow recognition of a speaker's accent. In practical applications, this is not the case if accents are very close to one another; one needs more utterances to model the variabilities among the speakers. In [52], Ghesquiere and Compernolle introduced a scheme based on a hierarchical accent classification approach with formant and duration features. The classification approach has two steps: to identify the gender and then to identify the accent. The accent classification accuracy reported in this paper is 52.6%. In [53], Yi and Fung introduced a scheme based on accent model reconstruction. The aim of this approach is to use Gaussian distributions from accented-speech models and to adjust the pre-trained model, thereby uncovering more variations in pronunciations. The word-error-rate reduction obtained in this work is 4.4% for accented speech.

2.4.2 Connectionist Modelling-Based Classifiers

Now that the statistical-based classification approaches such as HMMs and GMMs for solving the problem of accent classification have been introduced, this section next provides an overview of ANNs. This literature review is limited to RBF and Multilayer Perceptron (MLP).

ANNs are the most common classifiers used in pattern recognition applications. ANNs have merits and demerits just as other statistical classifiers do. The performance of ANN classifiers is better than that of their counterpart statistical classifiers, if the training examples belong to a small application domain. ANNs are considered better than statistical classifiers in problems of modeling non-linear mapping.

ANNs can be broadly categorized into three main domains: MLP, RBF, and recurrent neural networks. Recurrent neural networks are not used to solve the problem of accent classification. RBF is generally considered to be a statistical classifier; therefore, it is unlikely that RBF will outperform GMMs. The advantage of RBF is that its training is very fast; it is therefore most suitable for real-time application for small application domains.

On the other hand, MLP is used for accent classification, largely because it is simple to implement and possesses a well-defined training algorithm. The main drawback with MLP is that its training is a time-consuming process; thus, it is not suitable for real-time applications. In addition, ANNs have many other design parameters, including the number of hidden neurons, the activation function, the number of hidden layers, and the number of neurons in each layer, that cannot be optimized easily or accurately. The proper setting of the parameters by trial and error makes training difficult.

In [54], Chan *et al.* did a comparative study between comparative learning ANNs, Back Propagation (BP) ANNs, and counter propagation ANNs. BP ANNs demonstrated superior performance in accent classification tasks (i.e., 90%) with pitch period and the first three formants. The database used in this application is small, but in real word applications, this is not always the case.

2.5 Support Vector Machines

SVM is an important example of a linear discriminant classifier [55]. SVM classifiers are mainly used to map the ordinal feature vectors to a linearly separable space. SVM classifiers are used in speaker recognition, speaker identification, and emotion recognition tasks. SVM classifiers have one advantage over statistical classifiers in that there is only one global minimum. However, in SVM classifiers, there is no systematic way to choose kernel functions and, hence, there is no guarantee that the transformed space will be separable. In [56], the authors introduced an SVM-based accent classification system using English Arabic and English Indian. In this work, speakers read a single page of English text on each of three topics. In practical applications, we do not ask for a speaker to read one page for the training of an acoustic classifier. It is well-known in speech processing and speaker identification task that the longer the utterance, the higher the classification accuracy. In [57], the authors provided experimental results and came up with the conclusion that SVM classifiers found to be very similar to that of HMM classifier for binary accent classification problems.

2.6 Other Classifiers

Many other classifiers have been successfully applied to improve the performance of ASR systems. Among these are *k-nn*, decision trees, and Naive Bayes classifiers. However, the implementation of these techniques in the area of accent classification has not been significant as compared to other classifiers, such as HMM, GMM, and RBFs.

2.7 Foreign Accent Databases

An important issue in speech recognition research is that most accent-based speech databases are not available for public use. Therefore, the collection of speech databases in general has become a major priority for speech researchers [58][59][60][61]. Unfortunately, most research institutes have created their own private databases that are inaccessible to outside researchers. On the other hand, databases that are available for public use do not have sufficient utterances for specific accents.

In this study, we have used three different databases to train and evaluate the performance of the proposed methods: TIMIT, the speech accent archive, and the Foreign Accented English (FAE) databases. This is why we divided our experiments into three stages.

In the first stage, we used the TIMIT database for the training and evaluation of the proposed method. This database has recordings from eight dialect regions of American English and in general reflects high performance for speaker recognition tasks with microphone applications.

In the second stage, we used the speech accent archive database, which is a useful database of foreign accents. However, most of the accent categories do not have sufficient utterances to allow a comparative system performance evaluation with the results obtained using the TIMIT database.

In the third stage, we used the FAE database to train and test the performance of the system. This database is based on foreign accents but is recorded with a sampling rate of 8000 Hz. Performance of a system trained using this database is not comparable with a system trained on the TIMIT because the TIMIT database is recorded at 16000 Hz. FAE must be regarded as a degraded speech database [9] because the sampling frequency is low. The performance of the system will not be the same as databases that are recorded over 16000 Hz and 8000 Hz.

2.8 Summary

In this chapter, we reviewed existing methods and techniques for accent classification and described two main approaches: statistical classifiers and connectionist classifiers. We saw that connectionist classifiers are rarely used for accent classification, despite being better able to discriminate between accents than statistical classifiers. On the other hand, statistical classifiers can be trained very quickly, but each classifier is trained just on its own data and not on the data of other classes. Therefore, every classifier has merits and demerits.

In the next chapter, we propose an accent classification scheme based on a distance metric learning approach and evolution strategy to improve the performance of speaker-independent IVR systems. A detailed description of the methodology is provided in Chapter 3.

Chapter 3

Speaker Independent Accent Classification Systems

Speaker independent accent classification is a complicated task because accents vary widely, even within a single community or country. Accent variation reduces the performance of Natural Language Call Routing (NLCR) systems. Intuitively, it is very difficult to capture all accent variations among different groups of native and non-native speakers. Accent classification becomes a complicated task when non-native speakers start to learn a second language, the substitution of their own native language phonology is common. Thus, it is very difficult for an automated speech recognition system to understand a caller's accent and route calls to an appropriately trained acoustic model, one that has been trained on a well-matched accent database. Most of the techniques discussed in the literature are based on a speaker-dependent approach and use the distance between data points as a dissimilarity measure. However, one can argue that distances such as the Euclidian distance are not suitable to capture the intrinsic pattern of data points. Some dimensionality reduction techniques such as multidimensional scaling [62], Isomap [63], Laplacian eigenmaps method [64], and semidefinite embedding [65] address this issue by using common distances. Kernelized methods also attempt to develop a solution by mapping data points to a new space where the distances might be more meaningful [66]. Another approach could be to learn a new distance metric from training data points.

The approach proposed in this thesis exploits the class inequivalent side information that is based on dissimilar pairs of points between two classes of accent. In this way, we transfer each accent group to a new space where the Euclidian distances between similar and dissimilar points are at their minimum and maximum, respectively [67][68].

This chapter is organized as follows: Section 3.1 describes the problem. In Section 3.2, we provide our proposed architecture for a next generation interactive voice response system. Section 3.3 considers different factors that make accent identification a complicated task. Section 3.4 demonstrates the overall natural language call-routing system and explains the contribution of different modules to enhance the performance of NLCR systems. In Section 3.5, we explain our proposed speaker-independent accent classification system and we conclude with a wrap up and suggestions for further work.

3.1 Problem Definition

Speech recognition accuracy is the most desired feature of speech-enabled applications, and highly valuable for commercial enterprises. Since Automatic Speech Recognition (ASR) systems lag far behind their counterparts (i.e., human agents) in performance, developing speech recognition applications that can replace human agents is a complicated and a challenging task.

Possible solutions to improve the performance of ASR systems include speaker-dependent speech recognition, domain-specific speech applications, or isolated word speech recognition systems. However, the performance of these systems degrades when such techniques are implemented in a flexible environment (flexible in the sense that the system maintains its performance irrespective of any speaker). Designing such a system would be a very difficult task. One solution would be to design a speaker-independent speech recognition system. But designing such a system requires huge amounts of training data. This training constraint of speaker-independent systems makes them not well-suited for many of the speech applications.

The problem is, *how to develop an efficient speech recognition system that maintains its performance whatever accent a speaker uses.*

This leads to two sub questions:

- Is any existing acoustic model well enough trained to capture all inter- and intra-speaker variations?
- Is there any mechanism that can deal with all speaker variations and provide an efficient and robust speech recognition system?

The first question is fairly difficult to answer, because to capture all intra- and inter-speaker variations, one needs a huge training database. To collect such a database is not only difficult but may also be impractical in view of the almost infinite number of possible variations. In addition, a huge grammar model would be needed to improve the accuracy of speech recognition systems. Current state-of-the-art computers are really a rather slow means for processing such a huge grammar model.

To improve the performance of existing speech recognition systems, we propose an architecture that not only improves speech recognition performance but also provides a framework for next-generation automated interactive voice response systems. A detailed description of the architecture is provided in the next section.

3.2 Proposed Architecture for Next Generation Interactive Voice Response System

Our proposed personalized Next Generation Interactive Voice Response (NGIVR) system is a complete end-to-end speech-enabled solution. Our ultimate goal is to make this system work as an automated personal assistant. It will have a full capability to deal with incoming calls, play back messages for users, their family members, friends, and business clients. It will be able to process scheduling notes or access a calendar to answer different questions about schedules and appointments, and update schedules when necessary. It will be fully equipped to take care of messages, process them, and put them into a database.

In the chain of operations, first we describe the importance of this next generation IVR system and its implementation issues. The proposed system is totally different from a traditional telephone or IVR system. The concept is based on the current revolution in the telecommunications industry caused by the expansion of Internet technologies. The traditional telephone system, called the Public Switched Telephone Network (PSTN) system, has been successfully used in the telecommunications industry since its inception. But, unfortunately, PSTN innovation has failed to match the explosive growth of technological advancements in the Internet industry. The technological revolution in Internet is largely due to open-source development platforms, which have lowered barriers and allow anyone to contribute innovative ideas. In the early 90s, much of the innovation was led by enthusiasts who desired to develop more powerful ways to communicate.

Our proposed architecture, as shown in Figure 3.1, is a complete natural language-based IVR system (natural language in the sense that it has the capability to recognize an input query, and based on the concepts provided in the query, extract information from a database). The system should have a high accuracy rate because it can distinguish between incoming calls based on a Telephone Microphone (TM) identification system. It is a well-known problem of automated speech recognition systems that their accuracy degrades if they have been trained with microphone data and tested with telephone data. The TM module in our system will detect the originating mode of the caller and switch to the acoustic model that is best suited to the caller. Our system has two kinds of speech servers: one that is trained with speech utterances recorded at 16 kHz, and another that is trained with speech utterances recorded at 8 kHz.

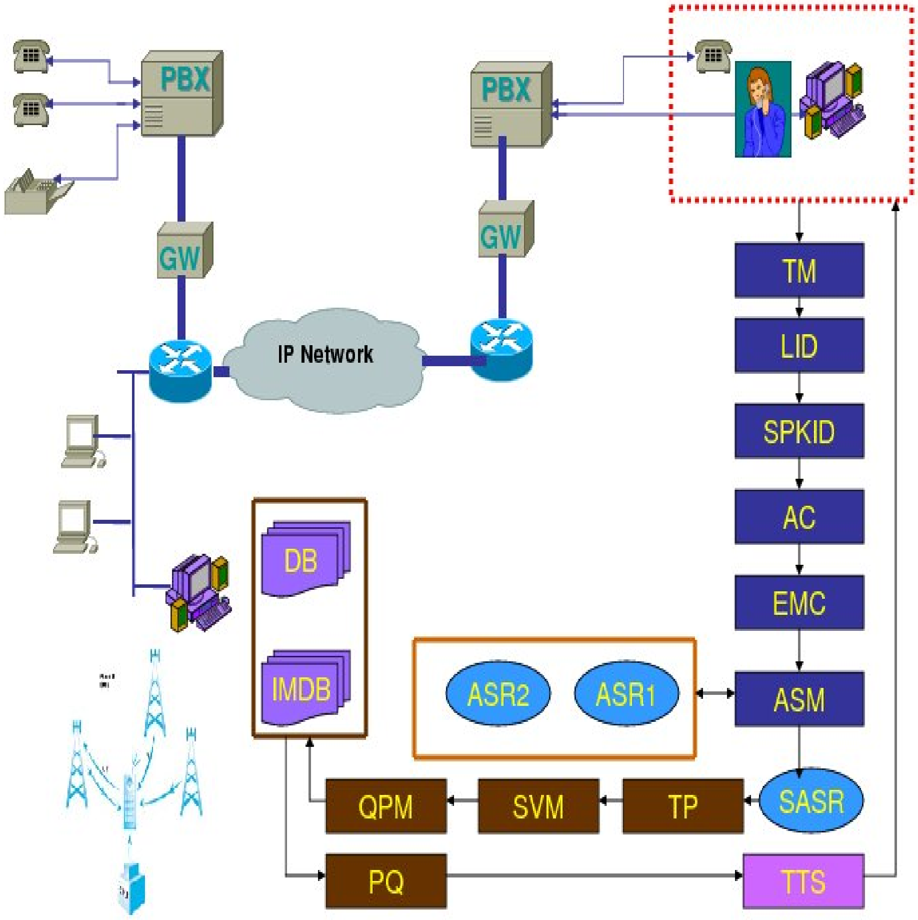


Figure 3.1: Next generation personalized IVR system

The second module in the chain of operations is the Language Identification (LID) module [69]. This module is basically responsible for identifying the caller's language. At this stage, the system is based on binary language identification, such as French or English, because all modern automated speech recognition systems give 90% priority to native speakers and 10% priority to non-native speakers. After implementing the LID module, we need to identify the caller, whether he/she is a family member, a friend, or an unknown caller. We will have different messages for each group in our voice-mail recordings box. For example, if in the instant notes supplied, the user writes a message for his son John as follows, “I am in a meeting and am dispatching a computer to you. Please stay at home,” whenever John calls the system, it will respond so:

- System: *This is an automated speech assistant system, How may I help you?*
- Caller: *I want to talk to my father (or John senior).*
- System: *Are you John?*
- Caller: *Yes.*
- System: *Okay, I have a message for you. Your father has purchased a computer for you, so stay at home and he will call you after his meeting.*
- Caller: *Okay.*
- System: *Is there something else you want to know?*
- Caller: *No thanks.*
- System: *Have a great day.*

If the system is not able to identify the speaker, it will ask the caller to record a message. The system will then store the message in its “unknown speaker” voice-mail box. However, this is not the end of the story: the system will also be able to recognize the message, provide a text log file in the incoming message folder, and tag it as from an unknown caller. In this way, the system will be able to provide text as well as audio transcriptions. This tagging allows users to search for an input query within the personalized IVR system. If the user says to the system, “Did John leave any messages on Tuesday,” the system will extract the information from the database and give textual as well as audio-form information about the message. This capability will help when users are busy and want to check for urgent messages only. In addition, the system will be able to indicate that a message is urgent and continuously prompt the user about any urgent messages if the name of the incoming caller matches the user's high priority message list.

The next module in the chain of operations, the accent classification system, is very important because the overall speech recognition accuracy of the proposed NGIVR system depends on it. The principal task of this module is to provide decisions about a caller's population group. For example, for native-American or native-Canadian speakers, the system will route the call to the acoustic model that has been thoroughly trained on a database of speech utterances recorded by such speakers. However, if the system classifies the input caller as a non-native speaker, then the call will be transferred to an acoustic model trained with a non-native speaker database. Switching of the acoustic models is a very important mechanism for overall performance improvement of the speech recognition system. This module is followed by a Emotion Classification (EMC) module and an Acoustic Model Switching (AMS) module. The task of the switching module is to select an acoustic model based on the classification scores provided by the Accent Classification (AC) module. The task of the emotion classification module is to determine the emotional state of the caller and add this information during dialogue initiation. We have not worked on this module, but will do so in the future.

The other modules, the selected automatic speech recognition module, Text Processing (TP) module, text classification module based on SVM, Query Processing (QP) module, and the Text-To-Speech (TTS) engine support the above-mentioned modules. All help in processing the automatic speech recognition results and select only key words instead of considering a whole query.

The following section discusses the core factors that really degrade the performance of accent classification systems. As mentioned above, in the proposed architecture, accent classification is the main module that improves the performance of IVR systems.

3.3 Factors Affecting Accent Classification

Accent recognition is a speech science problem and is most often referred to in automatic speech recognition systems as foreign accent classification or identification. Accent is defined as a pronunciation pattern used by a speaker reflecting his or her native language, but also tending to be reflected when the speaker is speaking another language. More generally, it is patterns of pronunciation designed or used by a social group or community to which a speaker belongs. Linguistically, accent variations lie in phonetic and prosodic characteristics [70]. In general, individuals who speak another language instead of their native language are referred to as non-native speakers. The ability of the non-native in any language varies from person to person and depends on the following factors:

- The age at which the individual started to learn the second language.
- Interaction of the individual with electronic and print media.
- How many years a speaker has spent learning and using the second language, etc.

However, the ability to reduce accent traits depends mainly on the length of time a speaker has spent to learn a second language. To learn a second language, one must develop a modified set of phonemes (i.e., the rise and fall of pitch, stress, and rhythm). The rise and fall of pitch in a speech segment is known as intonation. Intonation is important in the study of accent classification/identification and, thus, the role of intonation has been studied extensively in different languages [71][72][73]. Each language has a different structure of intonation, which depends on phonetic structure, semantic structure, and syntax. In [70][74], it was shown that German and English have different fundamental frequency contours. Generally, accent identification problems due to the inter-confusability of phoneme classes and similarities between different languages are the most challenging problems in ASR systems. Phoneme overlap degrades the performance of any speech recognition system when the system is tested using a speaker with a different accent.

3.3.1 Inter-language Confusability

The most important factor contributing to the increase of fuzziness between phoneme boundaries and phoneme classes is inter-language confusability. This confusability affects the performance of call-routing systems, especially in natural language call-routing applications. Inter-language confusability is a dominant factor even in the case of a single language. Geographical settlements can have a great impact on the variability of the phoneme set and can create some degree of inter-language confusability. Linguists estimate that approximately 4000 different languages are being spoken and understood by humans around the world [75]. It has been determined that the phoneme traits of a language can change due to geography. Even though the language may be the same, there is a change in pronunciation. This change leads to some languages being very close to each other in terms of pronunciation, while others are not close at all. For example, English and German are close to each other but very different from Chinese. Language grouping depends on the level of similarities and dissimilarities; languages are grouped into different language families on the basis of their phonetic, semantic, prosodic, and syntactic structure. In language families, speaker accent development includes phoneme production, tongue and lip movements, articulations, and other physiological processes related the vocal tract. When speakers start to learn a second language, the substitution of native language phoneme pronunciation is a common process. This substitution makes accent identification a difficult and complicated task. For example, in the case of the English word *cat*, the sound of */AE/* is not available in Arabic. Arabic native speakers therefore substitute */AA/* instead of */AE/*. It has also been pointed out that most of the non-native speakers of such languages as Turkish, Spanish, Hungarian, and Indian substitute */D/* and */T/* for */DH/* and */TH/* consistently.

3.4 Accent-Based NGIVR System

The proposed architecture for our accent-based ASR system is shown in Figure 3.2. In the chain of operations, the speech signal is first converted from analog to digital called digitized speech. This speech is then used to extract the speech feature vectors. Such vectors (i.e., acoustic representation) play an important role in the performance of an ASR system and provide the means to separate the classes of speech sound.

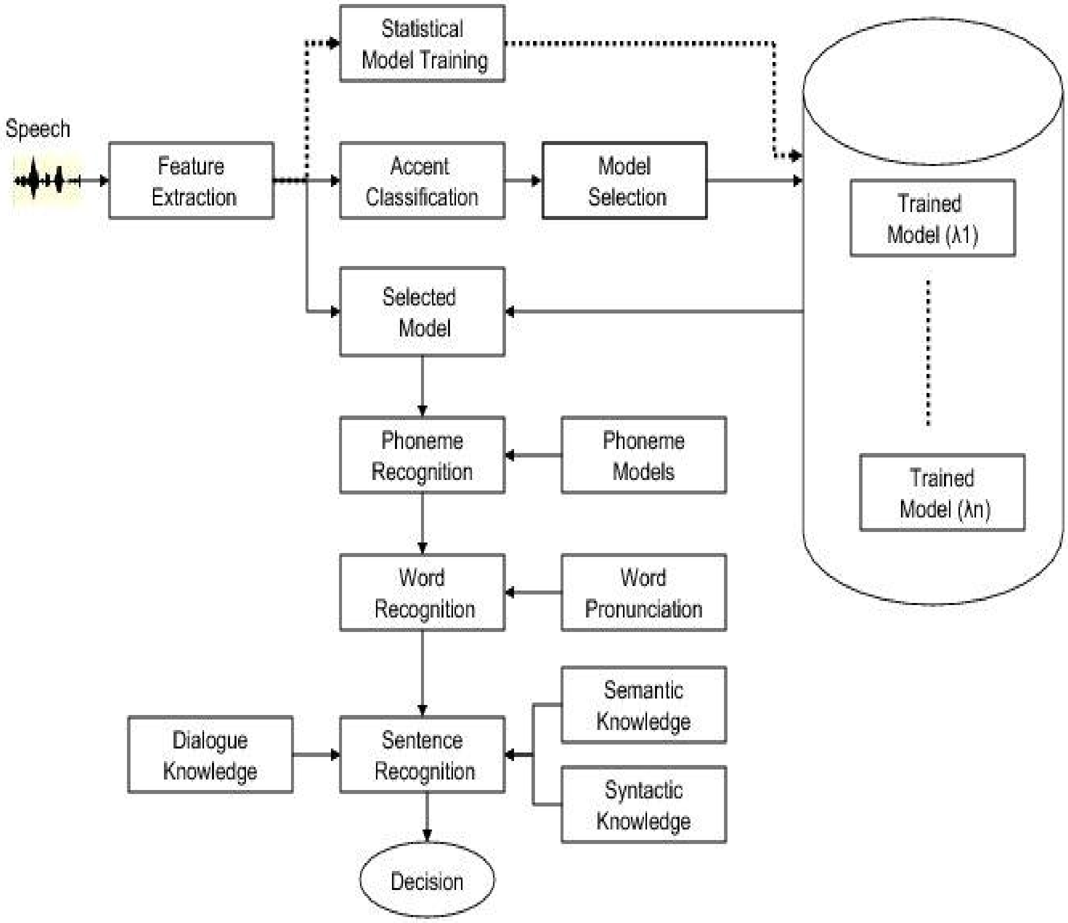


Figure 3.2: Accent-based NLCR system

Many variations of the speech features characterize the accents of different speakers. In our proposed approach, we have implemented Mel Frequency Cepstral Coefficient (MFCC) features, the first three formants, and energy. It is believed that MFCC features are considered the most effective for speaker identification tasks, and the first three formants capture most variations in the phoneme of any language. Hence, we selected only the first three formant features from among all other prosodic features and the MFCC features from a set of phonetic features. Our proposed structure of speech feature vectors is shown in Figure 3.3.

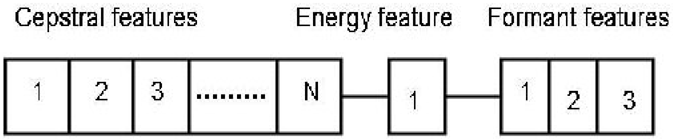


Figure 3.3: The structure of a combined feature vector

In this structure, we combine the MFCC, energy, and the first three formants as one vector. Thus, we have a 17-dimensional feature vector in the case of 13 MFCCs and a 29-dimensional feature vector in the case of 25 MFCCs. We directly concatenate to formulate a feature vector that is proposed to improve the performance of accent-based natural language call-routing systems. Next, during the training phase, the acoustic classifier is trained on these feature vectors for different speakers. The dotted line in Figure 3.2 shows the training path for the acoustic models. After their successful training of the models, the extracted features are used to test the model's ability to recognize phonemes by matching the extracted features from an unknown utterance. Each of these phoneme matches can be viewed as a local match, which leads to a global match through the integration of many local matches. The global match is a result of the best sequence of words that match the data.

During accent classification, feature vectors are used to classify the accent of an unknown speaker. Each input utterance is used by each accent model. The accent model that gives the maximum score to the input utterance is considered to be the best candidate to match the input of an unknown speaker. The highest score is used to select an appropriate model for speech recognition. Our proposed architecture is very flexible and may be used for any speech application, including a natural language understanding scenario, by implementing semantic and dialogue knowledge.

For speech recognition, we have employed HMM. During phoneme recognition, we employed phoneme models. In the case of American English, we trained 38 phoneme models. Next, for word and sentence recognition, we applied a dictionary and grammar model. Two approaches were used to generate a grammar: the SLM [76] and the GSLM. SLM is used primarily for large applications, while GSLM is used for small ones. In our proposed system, we used SLM.

If we integrate all the modules, as seen in Figure 3.2, it becomes very clear that the performance of the ASR system depends primarily on accent classification. Our goal is to use a distance metric learning technique and evolution strategy to enhance the performance of the accent-based ASR system for routing a call to its appropriate destination.

3.5 The Proposed Approach for Speaker Independent Accent classification Systems

3.5.1 Introduction

In this thesis, a new method is proposed based on a distance metric learning approach and evolution strategy. Distance metric learning methodology depends on side information from dissimilar pairs of accent groups to transfer data points to a new space where the Euclidian distances between similar and dissimilar points are at their minimum and maximum, respectively [67][68]. In this approach, we employ only dissimilar information from two groups of accents instead of both similar and dissimilar information. Intuitively, in the case of accent classification, we need to learn a distance metric that preserves the dissimilarity of accent groups. Thus, the Euclidian distances between two accent classes are maximized. We therefore employ a distance metric learning technique to transfer the data points to a new feature space where the distances between different accent classes are maximized. In addition, evolution strategy plays a very important role in obtaining the optimized Gaussian mixtures for improving the performance of the classification system.

To achieve optimized Gaussian mixtures, we employ a NSES-based k-means clustering algorithm on the training data set processed by the distance learning metric approach. The main objectives of NSES are to find the cluster centroids as well as the optimal number of clusters for a given data set. The principal advantage of k-means clustering is that it tends to converge quickly, but generally with less accurate clustering centroids. Therefore, this type of clustering approach usually yields locally optimal solutions because it is easily trapped into local minima/maxima, depending on the nature of the objective function.

To address this localizing issue, we propose an NSES-based k-means clustering algorithm for finding globally optimal clustering centroids and the optimum number of clusters. This NSES-based k-means clustering yields globally optimized Gaussian mixtures for an accent classification system. In the final accent classification stage, we implement a GMM to classify the accent of an unknown speaker.

The following subsections first present the theoretical foundations of the distance metric learning approach and then explain the graphical representation of the proposed methodology.

3.5.2 Theoretical Foundations

A Distance Metric Learning (DML) approach is based on learning a distance measure of training examples in the input space of data. Learning a good distance metric in feature space in an important task in many real world applications, such as classification, supervised kernel machines, k-means clustering, image processing, content-based image retrieval, and other computer vision tasks. It has an equal importance in supervised and unsupervised machine learning tasks. Due to its implementation in several machine learning applications, there has been considerable research interest on DML approaches to preserve the insight structure of training examples.

DML algorithms can be divided into two main categories: supervised and unsupervised. In a supervised DML approach, class information is extracted automatically by similar and dissimilar pairs of training classes. Similar pairs mean that the data points belong to the same class, and dissimilar pairs mean that the data points belong to a different class. This way of extracting class information is unlike the traditional way of class labeling; there is no need to label the data before processing. The supervised DML approach is further divided as follows:

- Global distance metric learning
- Local distance metric learning

Learning distance metrics in a global sense means learning all pairwise constraints simultaneously; in a local sense it means to satisfy local pairwise constraints.

Related Work

DML has played a vital role in clustering, classification, and information retrieval applications. The authors of [77][78] have shown the importance of learning an appropriate distance metric in classification tasks. It is proposed in the literature that learning a good distance metric can substantially increase the performance of a machine learning algorithm beyond what is possible with the standard Euclidean distance.

3.5.3 Description of the Proposed Framework

This section provides an overall representation of the proposed method as shown in Fig. 3.4. In the chain of operations, the DML module finds dissimilar pairs of accent classes and transfers the data points to a new space where the distances between the dissimilar points are at their maximum. Then, the eigenvector with the largest eigenvalue of each class of inequivalent side information is found and the data projected to a new dimensional space (i.e., less than the original feature space) for further processing. We are now able to apply any clustering technique to further process the data. In the proposed framework, the ultimate goal is to train a GMM for accent classification. Gaussian mixtures are needed to train the model for accent classification purposes. We therefore need a robust clustering technique that yields globally optimal Gaussian mixtures to efficiently train the classifier. We selected a k-means clustering algorithm for finding the Gaussian mixtures because k-means tends to converge quickly. The principle disadvantage of k-means is that, most of the time, its solutions are only locally optimal.

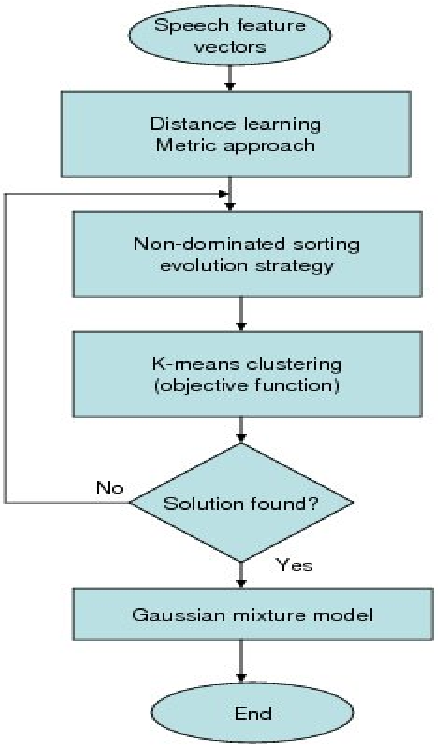

Speech feature vectors
Distance learning Metric approach
Non-dominated sorting evolution strategy
K-means clustering (objective function)
No
Solution found?
Yes
Gaussian mixture model
End

Figure 3.4: The structure of proposed framework

To address this issue, we propose a hybrid clustering approach based on NSES and k-means clustering algorithm. The principle task of NSES is to avoid the shortcomings of a standard k-means clustering approach. Our approach is used to avoid the situation in which the k-means clustering algorithm is trapped in a local minima. It also offers some freedom of choice and variability in Darwinian evolution. Thus, only the fittest cluster centroids survive and all others are eliminated. This process of elimination continues until we reach the maximum number of iterations or we arrive at an optimal solution. At the end of this process, we thereby get more suitable and robust clustering centroids for a Gaussian classifier. In the final stage of the proposed approach, we implement GMM as an accent classification module.

The following sections explain in detail each module of our proposed approach.

3.5.4 Distance Metric Learning Module

We employ a supervised distance metric learning approach for accent classification tasks in an interactive voice response system. Our main objective is to enhance the performance of the accent classification system. We employ a closed-form solution [79] to solve the distance metric learning approach. We use only class inequivalent side information, not both class equivalent and class inequivalent information as proposed in [67][68][79]. In our case, the distance metric learning is explicitly learned to maximize the distance between points that belong to different accent classes. Thus, our learning distance metric is based on the intuition of side information that, in turn, is based on information extracted from dissimilar pairs of data.

Let us consider that we have a data set, $X = \{x_t; t = 1, 2, ..., T\}$ be a set of T patterns, $\mathbf{x}_t \subseteq R^n$, $i = 1, 2, ..., N$, where N represents the dimension of the input feature space. We denote the dissimilar pairs of data by DP. Therefore, we define the set of dissimilar pairs of points as follows:

$$DP : (x_i, x_j) \in \quad \{\text{different accent classes}\},$$

where x_i and x_j are dissimilar pairs of the training examples. The Mahalanobis distance between the samples x_i and x_j is defined as

$$d(x_i, x_j) = \parallel x_i - x_j \parallel = \sqrt{(x_i - x_j)^T A (x_i - x_j)}. \tag{3.1}$$

We want to learn the metric, d_A, in which the distance between the dissimilar points DP enlarges. Thus, DP gives a metric that preserves the distances between dissimilar accent groups. Since A is a positive semidefinite matrix, it can be written as $A = WW^T$. If the dimension of d_A is $D\mathbf{x}d$, $d < D$ is equivalent to calculating the distance in the transformed subspace of A. Ideally, our distance metric A should induce a distance metric $D^{(A)}$ over points x_i and x_j as

$$D^{(A)}(x_i, x_j) = \parallel x_i - x_j \parallel = \sqrt{(x_i - x_j)^T A (x_i - x_j)}, \tag{3.2}$$

where A is a positive semi-definite matrix such that, by definition, $A \succeq 0$. The cost function is defined as follows:

$$L(A) = \sum_{(x_i, x_j) \in DP} \parallel x_i - x_j \parallel_A^2 \tag{3.3}$$

This cost function learns a distance metric d_A that maximizes the distance between dissimilar accent classes. Before solving the cost function, it can be written as

$$L(A) = \sum_{(x_i, x_j) \in DP} (x_i - x_j)^T A (x_i - x_j), \tag{3.4}$$

where A is semidefinite and can be written as $A = WW^T$. We thereby write the cost function as follows:

$$\begin{aligned} L(A) &= \sum_{(x_i, x_j) \in DP} (x_i - x_j)^T W W^T (x_i - x_j) \\ &= \sum_{(x_i, x_j) \in DP} Tr((x_i - x_j)^T W W^T (x_i - x_j)) \\ &= \sum_{(x_i, x_j) \in DP} Tr(W^T (x_i - x_j)(x_i - x_j)^T W) \end{aligned} \tag{3.5}$$

This objective function leads to an optimization problem:

$$max_{_A} L(A) \tag{3.6}$$

subject to the following constraints:

$$A \succeq 0 \tag{3.7}$$

$$Tr(A) = 1 \tag{3.8}$$

The first constraint ($A \succeq 0$) means positive semidefiniteness of the matrix and ensures a Euclidean metric. The second constraint prevents a trivial solution in which all distances are zero. We used a closed-form solution to optimize the objective function. The main advantage of such a solution is that it provides the best possible transformation solution in one step, unlike to off-the-shelf optimization and iterative methods [79]. However, the closed-form solution of Equation (3.6), by applying a Lagrangian function gives

$$\begin{aligned} \max_{(W,\lambda)} \phi(W,\lambda) = &\sum_{(x_i,x_j)\in DP} Tr(W^T(x_i - x_j)(x_i - x_j)^T W) \\ &- \lambda(Tr(WW^T) - 1). \end{aligned} \tag{3.9}$$

Taking a derivative and equating to zero, we have,

$$\left[\sum_{(x_i,x_j)\in DP} (x_i - x_j)(x_i - x_j)^T \right] W = \lambda W \tag{3.10}$$

where the Metric of Dissimilar Points (MDP) can be written as follows:

$$MDP = \sum_{(x_i,x_j)\in DP} (x_i - x_j)(x_i - x_j)^T \tag{3.11}$$

Hence, the above equation can be written in a standard eigenvector problem as

$$(MDP)W = \lambda W \tag{3.12}$$

Now, we can easily get optimal matrix W having eigenvector corresponding to the largest eigenvalue.

In our proposed approach, we intend to do clustering and then classify the accents of different speakers. In the clustering and classification task, we usually deal with more than one set of dissimilar side information. For instance, in the case of accent clustering, there could be several clusters, such as the ones that result from a slight variation in a phoneme pronounced by different speakers. These slight variations are due to non-native speakers accidentally introducing new phoneme boundaries based on their native languages. To extend the above metric learning method to exploit several sets of dissimilar pairs, first we construct a matrix for each set of dissimilar pairs. Then, the new learned matrices are combined to form a single matrix.

Suppose we have k sets of side information, where $k = 1, 2, ..., K$. We construct k matrices $A_1, ..., A_K$. We then take the largest eigenvector v_k, of each matrix A_k, and form a new matrix of column vectors v_k, $\overline{A} = \{v_1, ..., v_K\}$. $\overline{A}$ is a rank k matrix, which can project data to a k dimensional subspace. Since $\overline{A}$ is not an orthogonal matrix, one can construct an orthogonal matrix $\widehat{A} = UU'$, where U is the k largest eigenvectors of $\overline{A}$.

The motivation to implement our learning metric technique is to improve the clustering as well as classify the accents of different speakers. Since A is a positive semi-definite, $A = WW^T$, W can easily be calculated by applying singular value decomposition to A. Rewriting, we have

$$\begin{aligned}
D^A &= \sqrt{(x_i - x_i)^T A (x_i - x_j)} \\
&= \sqrt{(x_i - x_j)^T W W^T (x_i - x_j)} \\
&= \sqrt{(W^T x_i - W^T x_j)^T (W^T x_i - W^T x_j)} \\
&= \sqrt{(z_i - z_j)^T (z_i - z_j)} \\
&= \|z_i - z_j\|_2
\end{aligned}$$

where $z_i = W^T x_i$. It is clear that this transformation also reduces the dimension from N to k, where k is the rank of A. Any classifying or clustering algorithm can then be applied to the new data points. We employed k-means clustering to initialize Gaussian mixtures for a GMM classifier. In the k-means algorithm, clustering performance is very sensitive to the input selection of cluster centroids [80]. In the standard k-means algorithm, we are not able to select the cluster centroids that yield a globally optimum solution [81]. There are two main drawbacks with the standard k-means algorithm: insufficient ways to select the optimum number of clusters and clustering partitions that provide globally optimum solutions [82]. Thus, the initialization process that randomly generates the initial centroids might produce different clustering solutions for the same data.

To improve the performance of k-means clustering, several researchers have worked on hybrid k-means clustering approaches [83][84][85]. First, we discuss the related work aimed to improve the performance of k-means clustering and then we present our proposed methodology.

Related Work

k-means clustering is widely used because of its computational efficiency. However, its fast convergence does not mean that it will also yield cluster centroids that are the globally optimal points of the data. The k-means algorithm (Appendix-C shows further details) seeks the k cluster centroids that minimize the sum of squared Euclidean distances between the data items and their respective cluster centroid. However, the k-means clustering algorithm is inheritably very sensitive to the initial selection of the k centroids, and there is no mechanism by which we can select the optimum number of clusters for data in hand. The other main problem with the k-means clustering algorithm is that the algorithm is likely to converge to partitions that are not globally optimum. Thus, many researchers incorporate the global searching capabilities of algorithms based on natural evolution (Appendix-D) and the social behavior of birds.

In [86], a hybrid genetic algorithm is proposed that finds the optimal partition of any given data. In this paper, it is proposed that a genetic k-means algorithm converges to a global optimum. In [87], the authors employed a genetic algorithm to improve the performance of the k-means clustering algorithm. To improve the process time, three new crossover operators are introduced in this paper. The authors main objective in using a genetic algorithm was to partition a large data set. The genetic algorithm-based k-means algorithm yielded an improved performance for clustering large data sets. The paper addressed the well-known problem of the k-means clustering algorithm getting stuck in local minima.

In [88], the authors extended the work of [86] and identified a faster version of a genetic algorithm-based k-means clustering algorithm. Their work was also inspired by the deficiency of the k-means algorithm. The algorithms in [86][88] always converge to global optima, but the algorithm proposed in [88] is faster than the algorithm presented in [86].

Building on the above-mentioned work, in the next section, we propose a new algorithm based on non-dominated sorting evolution strategy.

3.5.5 Non-dominated Sorting Evolution Strategy Module

In this section, first we describe a brief overview of evolution strategy and its forms. Next, we give a concept of non-dominated evolution strategy. Finally, we present our proposed algorithm.

Evolution Strategy

Evolution Strategy (ES) is a stochastic search algorithm based on the principle of natural evolution (Appendix-E). It is a process of continuous reproduction, just like in biological species. Thus, the next generation keeps the traits of its predecessors with the highest fitness values and transfers its genetic characteristics to the next generation, resulting in a final generation more robust and efficient than the previous generations.

Evolution strategy was developed at the same time as Holland and his students were developing genetic algorithms in the late 1960s and early 1970s in the United States and I. Rechenberg and his team were working on evolution strategy in Germany. These two techniques are based on the process of natural evolution, but their way of encoding and implementation of evolutionary operators was different. Initially, the encoding mechanism of a genetic algorithm was based on binary strings, while evolution strategy was developed to tackle real-value optimization problems. Normally, the population size in evolution strategy is less than that for genetic algorithms, and instead of crossover, more emphasis is placed on mutation. These factors make evolution strategy faster and easier to implement than genetic algorithms. This computational benefit motivated us to employ evolution strategy instead of genetic algorithms.

The initial version of evolution strategy was very simple perhaps because of the speed of early simple computers. This early version was introduced with the name of $(1 + 1)$-ES. In this approach, the structure of parent and offspring was very simple. Each parent goes through the process of natural evolution and yields an offspring. The " $+$ " sign is used for the selection mechanism and means that the next generation will be selected from the current population of both parent and offspring. Thus, the best individual will be selected as a parent for the next generation. This approach can be viewed as a point-to-point random walk. During this random walk operation, the next movement is determined by the fitness value of the individual. If the next position is better than the current one, the control will be transferred to the next point. There are chances that the algorithm may become stuck in a local minimum or cause premature convergence due to the mutation operator. To address this issue, I. Rechenberg introduced a $1/5$ *success rule* to control the premature convergence and make the algorithm faster. Still there is no guarantee that the algorithm will converge to the global maxima of a numerical optimization problem.

To avoid this situation, two new variants of evolution strategy are provided, $(\mu + \lambda)$-ES and (μ, λ)-ES. In the $(\mu + \lambda)$-ES approach, the next generation is selected from a pool of current parents and offspring. The number of offspring, λ, is selected from more than μ in each generation. This principle is based on the natural reproduction mechanisms of some biological species who produce many offspring, of which only few with the highest fitness values survive. A selection mechanism is employed to prune back the current population to its original size. In this approach, there are still chances a robust individual with poor strategy variables can stay in the population. So, a (μ, λ)-ES selection mechanism is introduced where offspring replacing parents can sometimes be more effective than $(\mu + \lambda)$-ES. However, in this technique the next generation is selected from offspring only. Thus, the good solutions from the previous generations are lost. But in the case of $(\mu + \lambda)$-ES the best solutions discovered so far are preserved and the next generation is selected from a pool of good solutions. This effectiveness is the reason that we employ $(\mu + \lambda)$-ES to improve the performance of the k-means clustering algorithm. A general principle of evolution strategy is shown in Figure 3.8.

Evolution strategy differs from a genetic algorithm mainly in respect to the encoding of the solution and as the selection of the operators (Appendix-E). Evolution strategies have sophisticated ways of changing the genetic characteristics of an individual by using mutation operators and increased selection pressure. The selection of an individual is performed deterministically, unlike the case of a genetic algorithm, where a stochastic method is used.

3.5.6 Non-dominated Sorting Genetic Algorithm-I

The Non-dominated Sorting Genetic Algorithm (NSGA) was proposed by Srinvivas and Deb in 1994. NSGA is another variation of Goldberg's ranking procedure for multiobjective genetic algorithms (Appendix-E). The NSGA is based on the concept of population classification into several layers. Each layer is comprised of

```
Evolution Strategy
$t = 0$;
Initialize($P_t$);
Evaluate($P_t$);
WHILE $isNotTerminated()$ do
  $P_p(t)$ =selectBest $P(\mu, P(t))$;
  $P_c(t)$ = reproduce $(\lambda, P_p)$;
  mutate $P_c(t)$;
  evaluate $P_c(t)$;
  if $(usePlusStrategy)$ then $P(t+1) = P(t+1) \cup P(t)$
    else $P(t+1) = P_c(t)$;
  $t = t + 1$;
END_WHILE
```

Figure 3.5: Pseudo code of evolution strategy

a set of individuals based on non-dominance. A set of non-dominated individuals are classified into one category. Hence, each category of individuals is assigned a dummy fitness value that is calculated based on the number of individuals in that category. This way of assigning fitness values allows all individuals to take part in the reproduction of the next generation. However, to reduce the population pressure on one region, a dummy fitness value is shared by the individuals in that category. The pseudo code is provided in Figure 3.6.

The main drawback with this algorithm is that its layering classification makes it computationally expensive. The other factor that reduces its popularity is that the individuals in front (i.e., the first classification layer) always have more chance to take part in the reproduction mechanism for the next generation, again resulting in convergence of the population towards certain regions. A fitness sharing mechanism tries to maintain population diversity but does not appear to be effective enough.

To further improve this algorithm Deb *et al.*[89] proposed another algorithm, called the non-dominated sorting genetic algorithm-II.

```
NSGA-I
$t = 0$;
Initialize($P_t$);
Evaluate($P_t$);
Assign rank based on pareto dominance in each run;
Compute niche count;
Assign shared fitness;
WHILE $isNotTerminated()$ do
  Selection of the fittest individuals via stochastic universal sampling;
  Single point crossover;
  mutate $P_c(t)$;
  evaluate $P_c(t)$;
  Assign rank based on pareto dominance in each run;
  Compute niche count;
  Assign shared fitness;
  $t = t + 1$;
END_WHILE
```

Figure 3.6: Pseudo code of non-dominated genetic algorithm-I

3.5.7 Non-dominated Sorting Genetic Algorithm-II

Non-dominated Sorting Genetic Algorithm (NSGA)-II is based on the original design of NSGA. In this algorithm, population fronts are created on the basis of non-dominance. The algorithm is based on the idea of a single fitness value assigned to a particular front of individuals or a category. During this fitness assignment, the first front, F_1, is created of individuals that are not dominated by any other individual in the population. This front is then given the highest fitness value and is temporarily removed from the population. Similarly, the second highest front, F_2, of individuals is created and assigned the second highest fitness value. This process of assignment continues until the whole population is divided into different fronts. After this fitness assignment, each individual in a particular front is assigned a crowding distance (i.e., a normalized distance to the closest neighbor in the front). It uses this crowding distance as a measure to make sure that all members are a certain distance apart. Thus, the algorithm prevents premature convergence and makes the population diverse. The pseudo code is presented in Figure 3.7. During every evolutionary phase, a population of size $2N$ is generated. Thus, the N best individuals are selected for the next generation. The main drawback with this algorithm is that its crowding distance assignment mechanism. The problem with this approach is that it discards all members (elements) of the non-dominated set which have less crowding distance. Most of the solution members that belong to a crowded region are eliminated. Thus, to make the members more diverse, a new approach is needed that makes the elimination process set-by-step. To overcome this shortcoming and to make the process of non-dominance more robust, we proposed a new algorithm based on NSES. The detailed description of the proposed methodology is presented in the next section.

3.5.8 Non-dominated Sorting Evolution Strategy-based k-means Clustering

The main difference between genetic algorithms and our approach is the way each presents an individual? and the type of evolutionary operators used. For individual representation, ES uses real values instead of binary strings.

In our proposed approach the main objective is to do clustering and then classify a speaker's accent and route a call to an appropriate acoustic model that has been thoroughly trained on a database of speech utterances recorded by such speakers. For clustering, we used the k-means algorithm; because the principle advantage of k-means clustering is that it tends to converge quickly. However, it generally comes up with locally optimal solutions because it is easily trapped into local minima/maxima, depending on the objective function. There is no efficient automated way to find the optimal number of clusters for a given data set. To address these drawbacks, we formulate the clustering problem under two objectives:

```
NSGA-II
$t = 0$;
Set population size $N$;
Initialize($P_t$);
Evaluate($P_t$);
Assign rank based on Pareto dominance - $sort$;
Generate offspring population ($Q_t$);
Apply binary tournament selection;
Apply recombination and mutation;
WHILE $isNotTerminated()$ do
  FOR $P_c(t) \cup Q_c(t)$
    Assign rank based on Pareto -$sort$;
    Generate set of non-dominated vectors along $PF_{known}$;
    Determine crowding distance between points and each front;
  END_ FOR
    Select points on lower front and are outside a crowding distance ;
    Generate next generation $P_{t+1}$;
    Binary tournament selection;
    Recombination and mutation;
  $t = t + 1$;
END_WHILE
```

Figure 3.7: Pseudo code of non-dominated genetic algorithm-II

minimizing of the clustering errors and finding the optimum number of cluster centroids K. Instead of yielding one solution, a heuristic algorithm should find the set of clustering solutions called the Pareto optimal set for this bi-objective clustering problem. The solutions in the Pareto optimal set are non-dominated by one another. The Pareto optimal set exhibits the relationship between K and the associated clustering error. To solve the bi-objective clustering problem, we propose an NSES multi-objective algorithm. Figure 3.8 illustrates the pseudo code of a non-dominated sorting evolution strategy, where t represents a generation index,

P_t a parent population whose size is $|P|$, Q_t an offspring population, R_t a combined population, $F = \{F_1, \ldots, F_{2|P|}\}$ the ordered set of non-dominated sets, $\prec$ a crowded-comparison operator. The best non-dominated solutions compose F_1, the second best non-dominated solutions compose F_2, and so on.

The proposed algorithm, based on NSES and k-means clustering, first initializes a population of cluster centroids (line 1-2). After population initialization, the algorithm evaluates each individual, and assigns a fitness value, and then enters into the evolutionary process (line 5 - 20) to generate the next generation of individuals. For every generation, the non-dominated sorting algorithm is applied to the combined population of parents and offspring (Line 6-7) to select half of them as a new parent population according to non-dominance rank and crowding distance (Line 11). Using special ES operators and the k-means clustering algorithm, the hybrid NSES creates the offspring population from the parent population (Line 17). The generational loop continues until t becomes T. To explain the whole process of this hybrid NSES algorithm, we shall next explain, in details and step-by-step, the encoding mechanism, population initialization, non-dominated sorting and selection, offspring generation, recombination, mutation, and k-means clustering bi-objective problem.

Population Encoding Mechanism

The population encoding mechanism commonly used for clustering problems can be divided into two categories: Partitioning-Based (PB) and Centroid-Based (CB). In the partitioning-based clustering, an individual is represented as a string of N integers ranging from 1 to K for a fixed K. The i-th integer in the string represents which cluster the i-th pattern is subject to. In the case of centroid-based clustering, an individual is represented by as an array of D-dimensional centroids, where the size of the array indicates the number of clusters, K. Thus, an individual of the proposed hybrid NSES should encode a variable number of clusters. In our approach, we use the centroid-based encoding mechanism. An

```
FUNCTION NSES
$t = 0$;
Initialize($P_t$);
Evaluate($P_t$);
$Q_t = \emptyset$;
WHILE $t < T$
  $R_t = P_t \cup Q_t$;
  $F$ = Fast_nondominated_sort($R_t$);
  $P_{t+1} = \emptyset$;
  $r = 1$;
  WHILE $|P_{t+1}| + |F_r| \leq |P|$
    Crowd_distance_assignment($F_r$);
    $P_{t+1} = P_{t+1} \cup F_r$;
    $r = r + 1$;
  END_WHILE
  Sort($F_r$, $\prec$);
  $P_{t+1} = P_{t+1} \cup F_r[1 : (|P| - |P_{t+1}|)]$;
  $Q_{t+1}$ = Generate_offspring($P_{t+1}$);
  Evaluate($Q_{t+1}$);
  $t = t + 1$;
END_WHILE
```

Figure 3.8: Pseudo code of the NSES.

individual, I, is given as a pair of (KD)-dimensional object variable vectors $\overrightarrow{c}$ and the same-sized mutation, step size vectors $\overrightarrow{\sigma}$. The latter is used to implement a lognormal self-adaptive mutation operation, which is described under the mutation operator. Figure 3.9 illustrates the structure of the individual string.

From the bio-inspired algorithm's point of view, the difference between the two methods lies in the realization of the crossover and mutation mechanism. The main problem in the partitioning-based representation is that the clusters become non-convex if simple random crossover mechanism is applied. The convexity of the

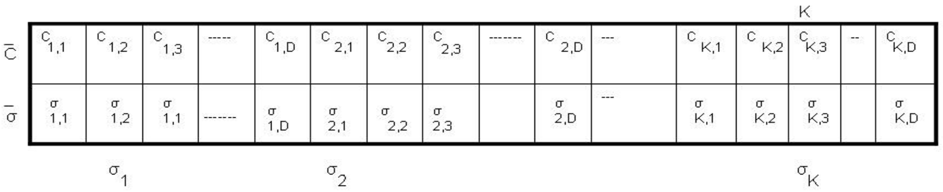


Figure 3.9: Encoding of solution candidates

solutions can be restored by applying the k-means algorithm, but then the resulting cluster centroids tend to move towards the centroids of the data set. This movement of clusters moves the solutions systematically in the same direction, which is the main reason to slows down the search. It is therefore more appropriate and effective to operate with the cluster centroids than with the partitioning table.

Population Initialization

During the hybrid NSES run, the size of P_t and Q_t is maintained as a predetermined value $\bar{P}$ for $t \geq 1$. At the initialization stage, however, the hybrid NSES fills P_0 with $2\bar{P}$ individuals representing variable-sized cluster centroid sets. When initializing each individual, the hybrid NSES first determines the number of centroids, K, as a uniform random number in $\{2, \ldots, \bar{K}\}$, where $\bar{K}$ is the maximally allowed value of K. It then initializes each centroid's location and the corresponding mutation step size one by one. Let the lower bound and upper bound of the d-th element of the patterns are be L_d and U_d, respectively, for $d \in \{1, \ldots, D\}$. The d-th element of each centroid is chosen as a uniform random number in $[L_d, U_d]$. The corresponding mutation step size is set to $\frac{U_d - L_d}{5}$, where U_d and L_d reprent the upper and lower bounds of the $d-th$ dimension, respectively and gives better results as compared other combinations.

Non-dominated Sorting and Selection

The non-dominated sorting is the key procedure of the proposed hybrid NSES that sorts the solutions in R_t based on their non-dominance rank. The sorting result is represented as $F = (F_1, F_2, \ldots)$. Each element of F represents a non-dominated front, and the subscript denotes its non-dominance rank. That is, F_1 denotes the front of non-dominated solutions in R_t, F_2 is the front for $R_t \backslash F_1$, F_3 is for $R_t \backslash (F_1 \cup F_2)$, and so on.

```
Procedure select
P = ∅;
i = 1;
WHILE |P| + |F_i| ≤ P̄ DO
  crowding-distance-assignment(F_i);
  P = P ∪ F_i;
  i = i + 1;
END_WHILE
crowding-distance-sort(F_i);
P = P ∪ F_i[1 : (P̄ − |P|)];
RETURN P;
```

Figure 3.10: Pseudo code of selection procedure.

The pseudo code of this selection procedure is outlined in Fig. 3.10, where P represents the parent population to be returned. The selection procedure starts from the best non-dominated front, F_1. If the size of F_1 is smaller than $\bar{P}$, the elements of F_1 are copied to P. The remaining members of P are chosen from subsequent non-dominated fronts. If $|P| + |F_l|\bar{P}$ for a certain $l \geq 1$, the solutions in F_l having the longest crowding distance are chosen as the last members of P. The crowding distance represents the distance to the neighborhood solutions in the same non-dominated front. The solution with a higher crowding distance is preferable since it can achieve a uniform spread of non-dominated solutions. The proposed NSES adopts the new non-dominated sorting algorithm instead of the original sorting algorithm of NSGA-II. Non-dominated sorting plays a very important role in keeping the population size equal to that of the initial population, because, at the end of each generation, the size of the new population doubles. We

```
FUNCTION Pruning non-dominated solution ($P_{t+1}$)
Calculate crowding distance of each member of set $F$;
Create a data structure for crowding distance of set $F$;
Create an ascending order list $H$ of $F$ based on crowding distance as key;
WHILE $|H| \leq |P|$
  Create a list of $H \leq N$ based on crowding distance;
  Remove member based on lowest crowding distance;
  Calculate new crowding distance;
  Update $H$;
END_WHILE
```

Figure 3.11: Pseudo code for non-dominated sorting assignment.

employ crowding distance for pruning the non-dominance, thus making the spread of extreme solutions as high as possible and avoiding the solutions being trapped in a specific region. The first task of non-dominant sorting is to select the number of non-dominated sets based on crowding distance. The crowding distance is calculated by measuring the distance between the nearest neighbors of the solution on each side. Assignment of a crowding distance to the particular element is based on normalization. This process is done by taking the difference of the maximum and minimum values of the distance of each neighbor and dividing these distances by the difference, adding them up, and assigning a crowding distance to the solution member under consideration. In addition, we also assign a maximum crowding distance value to the members of the non-dominated set that are at their minimum and maximum, respectively. This assignment ensures that both extreme solutions are retained. Finally, the members of the non-dominated set are sorted, and those with the highest crowding distance are selected for keeping the population to the original size.

The problem with this approach is that it discards all members (elements) of the non-dominated set which have less crowding distance. Most of the solution members that belong to a crowded region are eliminated. Thus, to make the members more diverse, a new approach is needed that makes the elimination process set-by-step. The proposed non-dominated sorting and crowding distance assignment is presented in the Figure 3.11.

The main advantage of this algorithm is that it does not eliminate the members of, F, blindly. In Figure 3.11, $|P|$ means the size of the initial population. The algorithm is forced to make the population size after each generation equal to the original population's size. The efficient implementation of this algorithm is based on a priority queue such as heap sort algorithm.

Offspring Generation

The procedure in Line 17 of Fig 3.8 creates $|P|$ offspring by applying evolutionary operators to P_{t+1} and fills Q_{t+1} with the offspring. The pseudo code of the offspring generating procedure is outlined in Fig. 3.12. It first selects I_1 from P_{t+1} with a tournament selection method whose tournament size is two. One individual is chosen from the tournament based on domination relationship. If one set of centroids dominates the other both in K and classification error, the set will be chosen as I_1. If they are non-dominated by each other, the individual with the higher crowding distance is selected. After selecting I_2 with the same method, the offspring-generating procedure applies recombination, mutation, and evaluation operators to I_1 and I_2. Suppose I_1 and I_2 represent k_1 and k_2 centroids, respectively. The recombination operator builds a pool of the $(k_1 + k_2)$ centroids and divides it randomly into two groups to obtain recombined I_1 and I_2. The standard Gaussian-distributed mutation operator with a constant step size is used for the mutation operator. This procedure is further explained in detail. In the following subsections, we explain each evolutionary operator in detail.

Recombination

Two parents are selected from P via tournament selection of size two and replicated into two offspring, I_1 and I_2. In this process of recombination, we selected

```
FUNCTION Generate_offspring($P_{t+1}$)
WHILE $|Q_{t+1}| < |P|$
  $I_1$, $I_2$ = Select($P_{t+1}$);
  Recombine($I_1$, $I_2$);
  Mutate($I_1$, $I_2$);
  k-means ($I_1$, $I_2$);
  Evaluate($I_1$, $I_2$);
  $Q_{t+1} = Q_{t+1} \cup \{I_1, I_2\}$;
END_WHILE
```

Figure 3.12: Pseudo code for offspring generation.

a recombination probability of $p_R \in [0, 1]$. In our algorithm, I_1 and I_2 represent K_1 and K_2 partition centroids, respectively. The recombination operator builds a pool of all the $K_1 + K_2$ centroids and divides them into two categories to obtain the recombined I_1 and I_2. The size of one category is chosen as a uniform random integer in $[\max(2, K_1 + K_2 - \bar{K}), \min(\bar{K}, K_1 + K_2 - 2)]$ so that the recombined I_1 and I_2 should represent centroids of more than one but no greater than $\bar{K}$. The mutation step size segment corresponding to a centroid moves together during the recombination procedure. Fig. 3.13 illustrates the this recombination operation.

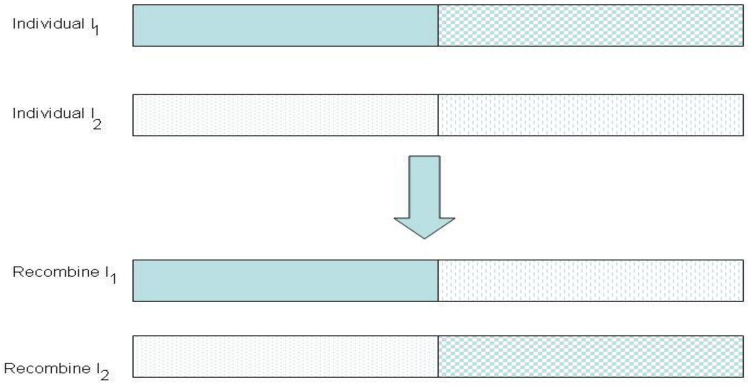


Figure 3.13: Recombination operation

Mutation

The standard lognormal self-adaptive mutation mechanism is applied to the recombined I_1 and I_2 [90]. For an offspring that denotes K centroids, two self-adaptation constants, τ_0 and τ, are set to $1/\sqrt{2\sqrt{KD}}$ and $1/\sqrt{2KD}$, respectively. The mutation step size $\sigma_{k,d}$ corresponding to $c_{k,d}$ for $k \in \{1, \ldots, K\}$ and $d \in \{1, \ldots, D\}$ is updated so:

$$\sigma_{k,d} \leftarrow \sigma_{k,d} \exp(z + z_{k,d}),$$

where $z := \tau_0 N(0,1)$ is a random constant commonly used for all the mutation step sizes in the offspring, $z_{k,d} := \tau N(0,1)$ is a constant randomly chosen for each mutation step size, and $N(0,1)$ is a normally distributed random number whose mean and variance are zero and one, respectively. The centroids element is then updated as:

$$c_{k,d} \leftarrow c_{k,d} + \sigma_{k,d} N(0,1).$$

The mutated $c_{k,d}$ is limited within the valid range $[L_d, U_d]$.

For a single-objective problem, we can prove the global convergence of the NSES once it incorporates the fixed mutation step sizes. However, this study adopts the self-adaptive mutation, which is known better in for local convergence speed.

k-means Operator

After the mutation operation, the k-means algorithm processes I_1 and I_2 one by one. The centroids represented by each individual work as the initial points of the k-means algorithm. At every iteration of the k-means, N patterns are associated with the nearest centroid, and then the centroids are moved to the center of the patterns they represent. The iteration terminates when either the centroids no longer move or the maximum iteration number, M, is reached. During the k-means run, the TWCVs of the two offspring are computed. The primary objective of the k-means clustering is to minimize the clustering error, which is generally given as Total Within Cluster Variation (TWCV) [86]

Let $\{x_i, i = 1, \ldots, N\}$ be the set of N patterns and $x_{i,d}$ the d-th element of x_i for $d \in \{1, \ldots, D\}$. A solution candidate of the clustering problem can be represented by a matrix $W := [w_{i,k}]$, where

$$w_{i,k} = \begin{cases} 1, & \text{if } i\text{-th pattern belongs to } k\text{-th cluster,} \\ 0, & \text{otherwise} \end{cases}$$

for $i \in \{1, \ldots, N\}$ and $k \in \{1, \ldots, K\}$, matrix $W := [w_{i,k}]$ that satisfying

$$\sum_{k=1}^{K} w_{i,k} = 1.$$

Let the centroid of the k-th cluster be $c_k = (c_{k,1}, \ldots, c_{k,D})$, then

$$c_{k,d} = \frac{\sum_{i=1}^{N} w_{i,k} x_{i,d}}{\sum_{i=1}^{N} w_{i,k}}.$$

The within-cluster variation of the k-th cluster is defined as

$$S^{(k)}(W) := \sum_{i=1}^{N} w_{i,k} \sum_{j=1}^{D} (x_{i,d} - c_{k,d})^2$$

and the TWCV is given as

$$S(W) := \sum_{k=1}^{K} \sum_{i=1}^{N} w_{i,k} \sum_{j=1}^{D} (x_{i,d} - c_{k,d})^2.$$

The objective of the clustering problem is to find a $W^* = [w^*_{i,k}]$ such that

$$S(W^*) = \min_{W} S(W).$$

It can be easily shown that for given patterns, the minimum TWCV is monotonically decreasing with respect to K. For the extreme case, $K = N$, we could achieve the minimum TWCV of zero by assigning one cluster to each pattern, which is meaningless in terms of data clustering. This is why most previous studies decided on a reasonable value of K first and sought for the best clusters accordingly. In this study, we consider the minimization of K as the second objective. A small K is preferable since it achieves a simple partition of given patterns. Moreover, incorporating this second objective enables us to secure the chance to find the better clustering results that can be achieved with a different K. Note that the k-means may end up with empty clusters. That is, a certain centroids may have no nearest patterns and no longer move during the k-means run. The proposed hybrid NSES avoids this situation by moving such a centroid to the pattern that is farthest from the nearest centroids. With this method, the k-means effectively minimizes the TWCV without generating empty clusters.

Thus, the NSES ends up with the optimal number of clusters and clustering centroids. After successful clustering of data, we applied GMM for a speaker's accent classification. For each accent group, we have one GMM model, λ_a. We trained GMM models by using Gaussian mixtures provided by the NSES algorithm. This automated method for calculating the Gaussian mixtures yields the best combination of Gaussian mixtures and optimum number of Gaussian components.

The following sections discuss the accent classification module and decision making and acoustic model switching modules for NGIVR systems.

3.5.9 Accent Classification Module

Consider a set of N feature vectors, $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, extracted from utterances belonging to a particular accent. The speech feature vectors are extracted by speech processing techniques, such as MFCCs, etc., each of which is a d-dimensional feature vector. The likelihood of GMM can be written as

$$P(X|\lambda) = \prod_{i=1}^{N} p(x_i|\lambda) \tag{3.13}$$

To calculate the parameters of the GMM model, the log-likelihood of a speaker belongs to a particular accent group a can be computed as follows:

$$\begin{aligned} L &= \log P(X|\lambda_a) \\ &= \log \prod_{i=1}^{N} p(x_i|\lambda_a) \\ &= \sum_{i=1}^{N} \log p(x_i|\lambda_a) \end{aligned} \tag{3.14}$$

where, $i = 1, 2, ..., N$ is the total number of speech feature vectors belong to a particular accent and $p(x_i|\lambda_a)$ is the Gaussian mixture density and can be computed as

$$p(x_i|\lambda) = \sum_{j=1}^{M} w_j \mathcal{N}(x_i; m_j, \Sigma_j), \tag{3.15}$$

where $\lambda = \{w_i, \bar{\mu}_i, \Sigma_i\}$ are the model parameters, $j = 1, 2, ...M$, are the mixture weights with mean vector m_j and covariances matrices Σ_j. The component densities are given by the multivariate Gaussian density so:

$$\mathcal{N}(x_i; m_j, \Sigma_j) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_j|^{\frac{1}{2}}} exp^{-\frac{1}{2}(x_i - \mu_j)^T \Sigma_j^{-1} (x_i - \mu_j)} \tag{3.16}$$

where T stands for transpose and $|\ |$ stands for the discriminant of a matrix.

The basic goal in training is to estimate the model parameters that maximize the above likelihood function. Since this function is very nonlinear in its model parameters, iterative techniques such as the Expectation Maximization (EM) algorithm must be employed. The above training procedure is repeated for each given class.

After a successful training of GMM models, we need to evaluate them. In the evaluation phase, an unknown speech utterance is represented by a sequence of feature vectors $X = (x_1, ..., x_N)$ and accent models by λ_a, where $a = 1, 2, ..., T$, is the total number of accent models. Now the main purpose of GMM classifier is to classify X utterances of a speaker into T accent models by computing the likelihood of an unknown speaker given each accent model, λ_a, and select accent $\hat{A}$ as

$$\hat{A} = \arg \max_{1 \le x \le T} \sum_{i=1}^{N} \log p(x_i|\lambda_a), \tag{3.17}$$

where X refers to the sequence of feature vectors extracted from the testing utterance, and $p(X|\lambda_a)$ is the likelihood function of X, given that it is generated by the a^{th} accent model.

3.5.10 Decision Making and Acoustic Model Switching Module

The final step in accent classification is the decision-making to determine in which accent group an unknown speaker belongs. The process of feature extraction and pattern matching is the same in most speech recognition applications; the decision depends on the type of application.

In our case, for a TIMIT database, we selected three speaker accent models. Basically, we have selected three different regional accents from the TIMIT database, each region having 100 utterances of different speakers. Let us say that A_a, is a particular accent model, and $a = \{1, 2, .., T\}$, where T is the total number accent models. We have three accent models in the case of the TIMIT database, each of which has 100 utterances. We have four accent group and each group has 16 speech utterances in the case of the speech accent database, and 100 in the case of the foreign accent database. In the decision-making process, an accent model is selected that has a maximum score (X, A_a) match between the unknown speaker's feature vectors, $X = \{x_1, x_2, x_3, ...x_N\}$, where each of which is a d-dimensional feature vector. We have implemented 12 MFCC, 3 formants $(f_0, f_1,$ and $f_3)$, and energy feature vectors. We also conducted experiments by selecting different combinations of the MFCC and prosody features to evaluate the performance of the proposed method, as shown in Chapter 5. The final decision regarding the speaker is made so:

$$Decision = \arg\max_{a} score(X, A_a) \tag{3.18}$$

In this way, we find the best match of a speaker to his/her particular accent group, as shown in Figure 3.11.

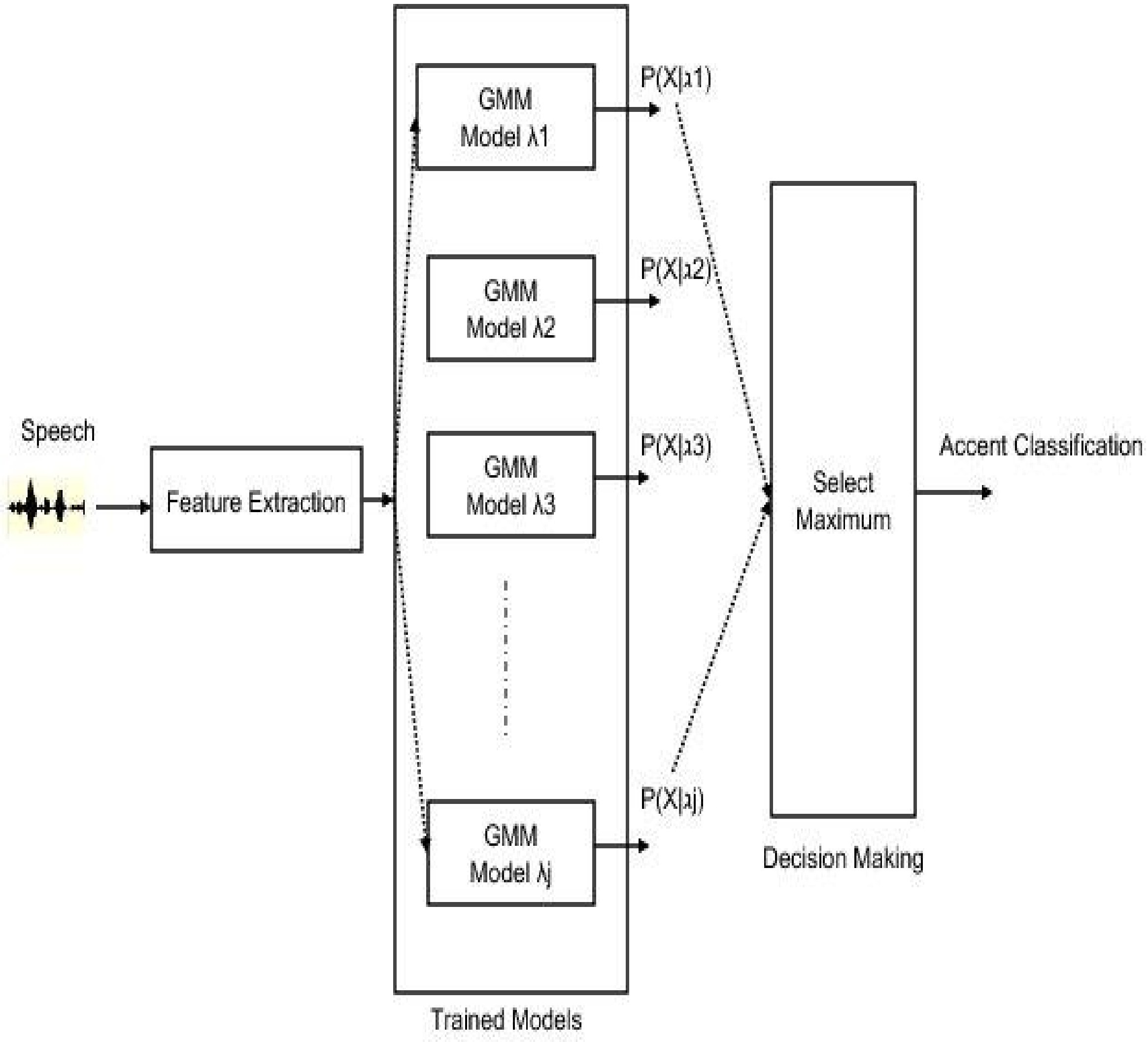


Figure 3.14: Accent classification system

The scoring mechanism provided by GMM classifiers with an unknown speech utterance is computed as follows:

$$\log p(\mathbf{X}|\lambda_a) = \sum_{i=1}^{N} \log p(\mathbf{x}_i|\lambda_a), \tag{3.19}$$

where i is the total number of feature vectors in an utterance and a is the total number of accent models. Finally, a maximum likelihood classifier hypothesis, $\mathbf{A}$, as the accent of the unknown utterance, is given as

$$\hat{\mathbf{A}} = \arg\max_{1 \leq a \leq 2} \log p(\mathbf{X}|\lambda_a) \tag{3.20}$$

3.6 Summary

Accent identification is a key factor in improving the performance of natural language call-routing systems. Accent identification is a complicated task because accents vary greatly. To enhance the performance of speaker-independent accent-based IVR system, we employ a GMM classifier. However, GMM performance depends on the initial partitions and number of Gaussian mixtures, both of which can reduce performance if poorly chosen. To overcome these shortcomings, we propose an accent classification system based on a DML approach and $(\mu + \lambda)$-ES. The DML approach depends on side information from dissimilar pairs of accent groups to transfer data points to a new feature space where the Euclidean distances between similar and dissimilar points are at their minimum and maximum, respectively. We use a closed-form solution that research shows to be simpler and faster than other solutions (i.e., off-the-shelf and iterative methods). Finally, a multiobjective NSES-based K-means clustering algorithm is employed on the training data set processed by the distance learning metric approach. The main objectives of NSES are to find the cluster centroids as well as the optimal number of clusters for a given data set. The principal advantage of K-means clustering is that it tends to converge faster, but generally with less accurate clustering centroids. Therefore, this type of clustering approach usually comes up with solutions that are only locally optimal because it is easily trapped into local minima/maxima, depending on the nature of the objective function. To address this localizing problem, we propose an NSES-based K-means clustering algorithm for finding globally optimal clustering centroids and numbers of clusters. This NSES-based K-means clustering yields globally optimized Gaussian components for an accent classification system.

In the next chapter, we provide speaker-dependent accent-based IVR system. In some real-world application, we need to deploy a speaker-dependent accent-based IVR system, such as personalized IVR systems. In this system, we need to identify the caller, whether he/she is a family member, a friend, or a business client. Then, the caller's query is transferred to a well-trained speech recognition system that is specific and best adapted for this caller's accent.

Chapter 4

Speaker Dependent Accent Classification System

As seen in the previous chapters, there are many problems affecting the performance of accent-based NLCR systems and it is very difficult to improve the performance of such systems. Accent classification usually depends on vowels and different combinations of vowels and consonants. This has led to fuzziness between phoneme boundaries and phoneme classes caused by co-articulation. The fuzziness between classes is caused by variation in speech organs, speaking style, and accent. The most important factors that have made foreign accent classification problem a challenging research issue are the anatomy of the vocal tract, fuzziness between phonemes, and inter-language confusability. The natural movements of speech organs lead to overlapping between consonants and vowels, resulting in fuzziness between phoneme boundaries, phoneme classes, and inter-language confusability. However, in some restricted speech applications, such as speaker recognition [91][92][93], speaker verification [94], and speaker identification [95][96], researchers have achieved satisfactory results. Our proposed method in the previous chapter outperforms well-known techniques in the literature for speaker-independent NLCR applications. Our proposed method for speaker-independent accent classification system provided in Chapter 3 is based on class inequivalent side information that maps the unknown testing data to the training feature space. However, for speaker-dependent applications, class inequivalent information is already available during the training and the testing phase. Thus, for a speaker-dependent application we propose a new approach based on fuzzy canonical correlation analysis.

The main objective of this chapter is to improve the performance of speaker-dependent natural language call-routing systems by applying a fuzzy canonical correlation-based clustering approach to find appropriate Gaussian mixtures for a GMM classifier. In our proposed method, we implement such a fuzzy clustering approach to minimize the within-group sum-of-square-error and canonical correlation analysis to maximize the correlation between the feature vectors and cluster centroids.

In this chapter we first describe a fuzzy clustering approach and then a canonical correlation analysis. Finally, we describe in detail our proposed methodology for a speaker-dependent accent classification system.

4.1 Fuzzy c-means Clustering

The grouping of speakers into different clusters can be broadly divided into hard clustering and soft clustering. In hard clustering, each datum is classified to one cluster only. This technique may not be suitable for accent recognition because it cannot accurately classify phonemes due to the fuzziness between phoneme boundaries and phoneme classes.

An alternative clustering technique is fuzzy c-means, by which each datum is assigned membership values indicating its degree of belonging to each of the given clusters. This value is called a fuzzy membership value and varies from 0 to 1. Unlike hard clustering, soft clustering has an inherent capability to fuzzily partition data. Fuzzy clustering is seen as a suitable technique to resolve the problem of fuzziness between phoneme boundaries and their classes. Fuzziness in this case results from variations in speech signals and coordinated movements of the speech organs.

The speech production mechanism results in an overlapping of phonemes. Therefore, it is very difficult or nearly impossible to exactly draw the boundaries to separate different combinations of phonemes, such as vowel-consonant-vowel. The only suitable solution is to use a clustering technique that is fuzzy in nature to characterize the underlying fuzziness in the phonemes. Intuitively, this has led to the implementation of fuzzy c-means clustering instead of other hard clustering techniques such as k-means.

Similar to the c-means algorithm, the main objective of the within-group fuzzy error function is to measure the sum-of-square-error between $\mathbf{x}_j$ and $\mathbf{c}_i$, where j denotes speech feature vectors and i denotes the total number of cluster centroids. The fuzzy c-means clustering algorithm in [97] is based on the basic principle of objective function minimization, the fuzzy within-group sum-of-square-error objective function in [97] can be written as

$$J_{wse}(U, c_1, .., C) = \sum_{j=1}^{N} \sum_{i=1}^{C} u_{ij}^m d_{ij}^2 \tag{4.1}$$

subject to

$$0 \leq u_{ij} \leq 1 \;\; \forall i, j$$

$$\sum_{i=1}^{C} u_{ij} = 1, \quad \forall j = 1, 2, ..., N,$$

where $m \in [1, \infty)$ is the weighting exponent. However, the Euclidean distance between the ith cluster centre and jth data point can be written as

$$d_{ij} = \parallel x_i^{(j)} - c_j \parallel . \tag{4.2}$$

In the above equations, the index, i, denotes the number of clusters for data set X, and j denotes the total number of data points in the set, X. The necessary conditions for (4.1) to reach its minimum are

$$c_j = \frac{\sum_{j=1}^{N} u_{ij}^m x_j}{\sum_{j=1}^{N} u_{ij}^m} \tag{4.3}$$

and

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} [d_{ij}/d_{ik}]^{\frac{2}{(m-1)}}}.$$

In fuzzy c-means clustering, the cluster centres are randomly initialized, and then the iterative procedure is carried out. There is no systematic or accurate rule for the initialization of cluster centres. Therefore, there is no guarantee that the c-means algorithm will converge to an optimum solution. The performance of the cluster centers thus depends primarily on the initial cluster centroids. Despite this, c-means has been successfully implemented in many applications, such as speaker recognition, speaker identification, image processing, robotics, weather forecasting, and many others [98],[99]. The fuzzy c-means pseudo code is presented in Algorithm 1.

4.2 Canonical Correlation Analysis

4.2.1 Introduction

Canonical correlation analysis is a well-developed multivariate statistical tool used to measure the linear relationship between sets of data. It was first proposed by H. Hoteling in 1936 [100]. It is extensively studied and implemented in a wide variety of applications, such as speaker adaptation [101], image and signal processing [102][103], biometrics [104], and neural networks [105].

4.2.2 Theoretical Foundations

Let $\mathbf{x} \in R^p$ be a set of the speech feature vectors and $\mathbf{y} \in R^q$ a set of cluster centriods, where $q \leq p$. Assume that the data have a zero mean. The main

Algorithm 4.1 Fuzzy c-means

```
1: Input: number of clusters $C$, degree of fuzziness $m \geq 1$, set of speech feature vectors, $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$
2: Initialize the elements of membership matrix $u_{ij}$ with random values between 0 and 1, such that
       $\sum_{i=1}^{C} u_{ij} = 1 \; \forall j = 1, ..., N$
   where
       $2 \leq c \leq n$
3: Calculate fuzzy cluster centres using as
       $c_j = \frac{\sum_{j=1}^{N} u_{ij}^m x_i}{\sum_{j=1}^{N} u_{ij}^m}$
4: Calculate the distances $d_{ij}$, $i$= 1,2,..., $C$ and j=1, 2,....,$N$ using Equation (4.2).
5: Update the fuzzy membership matrix
       $u_{ij} = \frac{1}{\sum_{k=1}^{C} [d_{ij}/d_{ik}]^{\frac{2}{(m-1)}}}$
6: Compute the objective function to minimize within-group sum-of-square-error
       $J_{wse}(U, c_1, .., C) = \sum_{i=1}^{C} \sum_{j=1}^{n} u_{ij}^m d_{ij}^2$
7: Stop the algorithm if there is no significant change in updating the cluster centroids or $i > I_{max}$, where $I_{max}$ is the maximum number of iterations.
8: i = i + 1
9: Go to step 3
```

objective of Canonical Correlation Analysis (CCA) is to find linear transformations for $\mathbf{x}$ and $\mathbf{y}$ such that projections are maximally correlated. More precisely, CCA is required to find the projection matrices $\mathbf{A}_x$ and $\mathbf{B}_y$, such that the correlation between $u = \mathbf{A}'_x\mathbf{x}$ and $v = \mathbf{B}'_y\mathbf{y}$ is maximized.

We are interested in calculating the maximum correlation. Since the correlation of multiple u and multiple v is the same as the correlation of u and v, the correlation coefficient between u and v is defined as

$$\rho = \frac{E[(u,v)]}{\sqrt{E[u^2]E[v^2]}}. \tag{4.4}$$

We therefore, want to calculate $\mathbf{A}_x$ and $\mathbf{B}_y$ to be such that u and v have zero mean $E[\mathbf{x}] = E[\mathbf{y}] = 0$ and unit variance, that is,

$$E[u^2] = 1 \tag{4.5}$$

and

$$E[v^2] = 1. \tag{4.6}$$

By solving Equation(4.5), we have

$$\mathbf{A}'_x\Sigma_{xx}\mathbf{A}_x = 1. \tag{4.7}$$

Similarly, by solving Equation (4.6), we have

$$\mathbf{B}'_y\Sigma_{yy}\mathbf{B}_y = 1. \tag{4.8}$$

After calculating the variance within-sets, we can also calculate the covariance between-sets as

$$E[uv] = E[\mathbf{A}'_x\mathbf{x}\mathbf{y}\mathbf{B}'_y]$$

$$E[uv] = \mathbf{A}'_\mathbf{x}\Sigma_{xy}\mathbf{B}_\mathbf{y}, \tag{4.9}$$

where Σ_{xx} and Σ_{yy} are called within-sets covariance matrices, and Σ_{xy} and Σ_{yx} are called the between-sets covariance matrices.

Substituting the values of u and v in Equation (4.4), we have

$$\begin{aligned} \rho &= \frac{E[\mathbf{A}'_x \mathbf{x}\mathbf{y}\mathbf{B}'_y]}{\sqrt{(\mathbf{A}'_x\mathbf{x})^2(\mathbf{B}'_y\mathbf{y})^2}} \\ &= \frac{\mathbf{A}'_x E[\mathbf{x}\mathbf{y}']\mathbf{B}_y}{\sqrt{\mathbf{A}'_x E[\mathbf{x}\mathbf{x}']\mathbf{A}_x \mathbf{B}'_y E[\mathbf{y}\mathbf{y}']\mathbf{B}_y}} \end{aligned} \tag{4.10}$$

Thus, by applying the constraints mentioned above, we can optimize the problem.

$$\rho = \max_{(\mathbf{A}_x,\mathbf{B}_y)} = \frac{\mathbf{A}'_x \Sigma_{xy}\mathbf{B}_y}{\sqrt{\mathbf{A}'_x\Sigma_{xx}\mathbf{A}_x B'_y \Sigma_{yy}\mathbf{B}_y}}, \tag{4.11}$$

where $\mathbf{A}'_x$ denotes the transpose of matrix A. From Equation (4.11), we can calculate the maximum canonical correlation co-efficient, ρ, with respect to $\mathbf{A}_x$ and $\mathbf{B}_y$.

Satisfying the conditions in Equations (4.5) and (4.6), the Lagrangian will take the form of

$$\psi(\lambda, \mathbf{A}_x, \mathbf{B}_y) = \mathbf{A}'_x\Sigma_{xy}\mathbf{B}_y - \frac{\lambda_x}{2}(\mathbf{A}'_x\Sigma_{xx}\mathbf{A}_x - 1) - \frac{\lambda_y}{2}(\mathbf{B}'_y\Sigma_{yy}\mathbf{B}_y - 1) \tag{4.12}$$

where λ_x and λ_y are Lagrange multipliers. Taking the derivative of Equation (4.12) with respect to $\mathbf{A_x}$ and $\mathbf{B_y}$, we obtain

$$\frac{\partial \psi}{\partial \mathbf{A}_x} = \Sigma_{xy}\mathbf{B}_y - \lambda_x \Sigma_{xx}\mathbf{A}_x = 0 \tag{4.13}$$

$$\frac{\partial \psi}{\partial \mathbf{B}_x} = \Sigma'_{xy}\mathbf{A}_x - \lambda_y \Sigma_{yy}\mathbf{B}_y = 0. \tag{4.14}$$

Multiplying the Equation (4.13) by $\mathbf{A'_x}$, we obtain

$$\mathbf{A}'_x\Sigma_{xy}\mathbf{B}_y - \lambda_x\mathbf{A}'_x\Sigma_{xx}\mathbf{A}_x = 0 \tag{4.15}$$

Similarly, multiplying the equation(4.14) by $\mathbf{B}'_{\mathbf{y}}$ so:

$$\mathbf{B}'_y\Sigma'_{xy}\mathbf{A}_x - \lambda_y\mathbf{B}'_y\Sigma_{yy}\mathbf{B}_y = 0 \tag{4.16}$$

Subtracting Equation (4.16) from equation (4.15), we obtain

$$\mathbf{A}'_{\mathbf{x}}\Sigma_{xy}\mathbf{B}_y - \lambda_x\mathbf{A}'_x\Sigma_{xx}\mathbf{A}_x - B'_y\Sigma'_{xy}\mathbf{A}_x + \lambda_y\mathbf{B}'_y\Sigma_{yy}\mathbf{B}_y.$$

$$= \lambda_y\mathbf{B}'_y\Sigma_{yy}\mathbf{B}_y - \lambda_x\mathbf{A}'_x\Sigma_{xx}\mathbf{A}_x$$

Since $\mathbf{B}'_y\Sigma_{yy}\mathbf{B}_y = 1$ and $\mathbf{A}'_x\Sigma_{xx}\mathbf{A}_x = 1$, this shows that $\lambda_y - \lambda_x = 0$, let $\lambda = \lambda_x = \lambda_y$. Assuming Σ_{yy} is invertible, we have

$$\mathbf{B}_y = \frac{\Sigma_{yy}^{-1}\Sigma'_{xy}\mathbf{A}_x}{\lambda}, \tag{4.17}$$

and substituting in Equation (4.13) gives

$$\Sigma_{xx}^{-1}\Sigma_{xy}\Sigma_{yy}^{-1}\Sigma'_{yx}\mathbf{A}_x = \lambda^2\mathbf{A}_x. \tag{4.18}$$

Similarly, substituting Equation (4.17) into Equation (4.14) gives

$$\Sigma_{yy}^{-1}\Sigma_{yx}\Sigma_{xx}^{-1}\Sigma'_{xy}\mathbf{B}_y = \lambda^2\mathbf{B}_y. \tag{4.19}$$

Equations (4.18) and (4.19) are generalized eigenvector problems of the form $Cx = \lambda Dx$. Now, we can therefore find $\mathbf{A}_x$ and $\mathbf{B}_y$ by solving Equation (4.18) and (4.19), respectively.

4.3 Proposed Method for Speaker Dependent Accent classification Systems

After introducing fuzzy clustering and canonical correlation analysis, we are now able to outline our proposed method for the problem of a speaker-dependent accent classification in ASR applications. In it, we have used fuzzy canonical correlation-based accent clustering and a GMM model for accent classification. In the acoustic model training phase, the data is first fuzzily partitioned using fuzzy clustering. In this way, thereafter, memberships to the cluster centres are determined by minimizing the distances from of feature vectors to cluster centers. We employ these fuzzy memberships associated with the speech feature vectors for class labels. This labeling process is based on a distance measuring approach for assigning fuzzy memberships to the speech feature vectors. A datum has a label value based on its higher membership to a particular cluster.

In the second level of clustering, the fuzzy membership values are again calculated by fuzzy clustering and updated by maximizing the correlation between the linear combinations of two groups of variables. In our proposed method, one group has fuzzy membership values and the other has the speech feature vectors of a particular accent group. Thus, this hybrid technique yields fuzzy clusters that are based not only on minimizing the distance between the cluster centroids and speech feature vectors but also on maximizing the canonical correlation coefficients to effectively deal with fuzziness between phoneme boundaries and phoneme classes. This fuzziness can be illustrated by taking as an example of the English word *BOY*. *BOY*, which is arranged in a consonant-vowel-consonant and depicts the fuzziness between phonemes. (Figure 4.1). In Figure 4.1, we can easily analyze the concept of fuzziness between consonants and vowels. This fuzziness, as explained above, is caused by natural movements of the speech organs. Natural overlapping makes the problem of accent classification difficult. Therefore, it is quite difficult for k-means to accurately characterize these combinations. The same problem arises with all distance-based clustering techniques. We therefore propose a clustering technique that is based not only on the distances between feature vectors, but also on maximizing the correlation between them.

To start with, we present our Fuzzy Canonical Correlation Analysis (FCCA)-based clustering algorithm. Next, we apply GMM for accent classification.

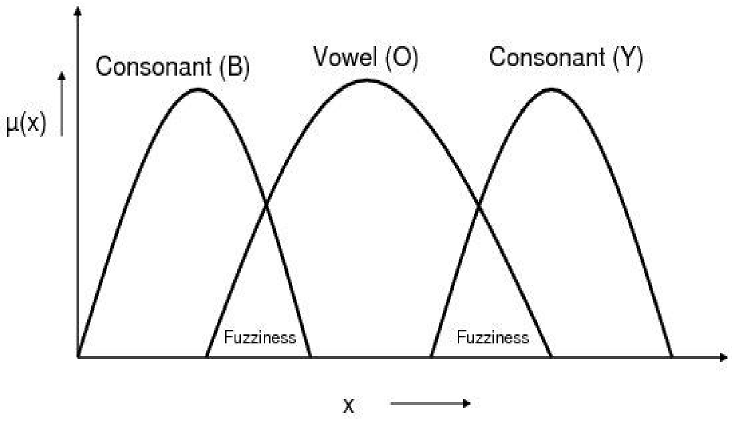


Figure 4.1: A concept of fuzziness between phonemes

4.3.1 Fuzzy Canonical Correlation Analysis (FCCA)-based Accent Clustering

As mentioned previously, in the proposed approach, we used fuzzy canonical correlation analysis to assign fuzzy memberships to the speech feature vectors. The overall structure of the system in the training phase is shown in Figure 4.2. The training procedure involves minimizing within-group sum-of-square-error by fuzzy c-means and maximizing the correlation between the linear combination of two groups of random variables by canonical correlation analysis.

Consider two sets of variables. One set of variables is the class indicators obtained by fuzzy clustering based on higher membership values, and another variable is the speech feature vectors of speech utterances recorded by speakers having a particular accent. Two vectors **x** and **y** consisting class labels and speech feature vectors can be represented as

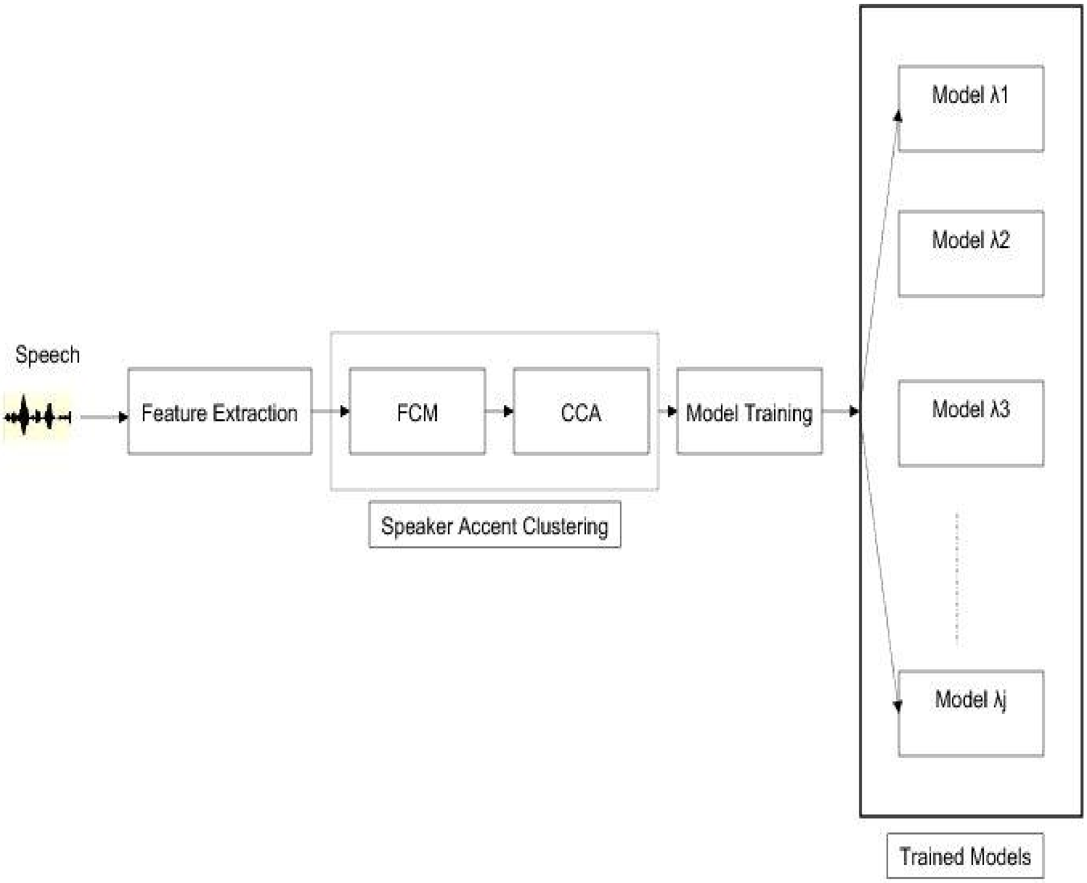


Figure 4.2: Training phase of the system

$$\mathbf{x_x} = (x_{ji}, x_{j2}, ..., x_{jP})$$

$$\mathbf{x_y} = (y_{jk}, y_{j2}, ..., x_{jP+Q})$$

where $i = \{1, 2, ..., P\}$ and $k = \{P+1, P+2, ..., P+Q\}$, such that $P \leq Q$. $\mathbf{x}_{ji}$ is a vector of class indicator based on fuzzy clustering takes a value between (0,1), j= {1,2,..,J}, is the data samples. If $\mathbf{x}_{ji} = 1$, it means that the data sample j belongs to cluster C_i otherwise $\mathbf{x}_{ji} = 0$.

Let us consider the linear combinations [106] with respect to $\mathbf{x}$ and $\mathbf{y}$ as

$$u = A_c^T(\mathbf{x} - \mathbf{v}_{c_x})$$
$$v = B_c^T(\mathbf{y} - \mathbf{v}_{c_y})$$

where $v_{c_x} = (v_{c1}, v_{c2}, ..., v_{cP})$ and $v_{c_y} = (v_{cP+1}, v_{cP+2}, ..., v_{cP+Q})$. Thus, $v_c = (v_{c_x}, v_{c_y})$ represents the cluster centre. The subscript $c = (c_1, c_2, ..., C)$ partitions J samples into C clusters. Now our main objective is to obtain A_c and B_c which represent coefficient vectors of $\mathbf{x}$ and $\mathbf{y}$. Our main objective is to employ fuzzy canonical correlation analysis for obtaining fuzzy clusters based on minimization of the within-group sum-of-squared error and maximization of canonical correlation coefficients by employing a hybrid objective function [107] as

$$\begin{aligned}\psi(\lambda, U, V) = &\sum_{c=1}^{C}[\alpha\{\mathbf{A_c}^T\Sigma_c^{xy}\mathbf{B_c} - \frac{\lambda_c^x}{2}(\mathbf{A_c}^T\Sigma_c^{xx}\mathbf{A_c} - 1) - \frac{\lambda_c^y}{2}(\mathbf{B_c}^T\Sigma_c^{yy}\mathbf{B_c} - 1)\} \\ &- (1-\alpha)\sum_{j=1}^{J}u_{cj}d_{cj}^2 - \beta\sum_{j=1}^{J}u_{cj}\log u_{cj}] \\ &- \sum_{j=1}^{J}\eta_j(\sum_{c=1}^{C}u_{cj} - 1),\end{aligned} \tag{4.20}$$

where

$$\Sigma_c^{xy} = \Sigma_{j=1}^J u_{cj}(x_{ji} - v_{ci})(x_{jk} - v_{ck}),$$

$i = 1, 2, ..., P$, and $k = P+1, ..., P+Q$. Σ_c^{xx} and Σ_c^{yy} are defines by $i = 1, 2, ..., P$, $k = 1, 2, ..., P$ and $i = P+1, P+2, ..., P+Q$, $k = P+1, P+2, ..., P+Q$, respectively. β is a weighting parameter that specify the degree of fuzziness of the clusters, and α is a tradeoff between canonical correlation and fuzzy clustering. There is no automated way by which we can determine the values of α and β. However, by repeating experiments several time we chose the values of α and β between 0.6 and 0.8, respectively. This suboptimal values of α and β varies from application to application. λ_c^x and λ_c^y, and η are the Lagrangian multipliers.

Now taking the derivative of the objective function with respect to u_{ci} and equating to zero, we have

$$
\begin{aligned}
\frac{d\psi}{dv_{ci}} &= -\alpha \sum_{k=P+1}^{P+Q} A_{ci}B_{ck} \sum_{j=1}^{J} u_{cj}(x_{jk} - v_{ck}) \\
&+ \frac{1}{2}\lambda_c^x \sum_{k=1}^{P} A_{ci}A_{ck} \sum_{j=1}^{J} u_{cj}(x_{jk} - v_{ck}) + 2(1+\alpha)\sum_{j=1}^{J} u_{cj}(x_{ji} - v_{ci}) \\
&= 0
\end{aligned}
\tag{4.21}
$$

where

$$v_{ci} = \frac{\Sigma_{j=1}^{J} u_{cj}x_{ji}}{\sum_{j=1}^{J} u_{cj}}$$

Now, similarly taking the derivative with respect to $\mathbf{A_c}$ and $\mathbf{B_c}$ and solving it, we have

$$\lambda_c^x = \lambda_c^y = \mathbf{A}_c^T \Sigma_c^{xy} \mathbf{B}_c$$

Thus, if Σ_c^{xx} and Σ_c^{yy} are non-singular, then the canonical form of u is obtained from the eigenvector A_c corresponding eigenvalue as

$$(\Sigma_c^{xx})^{-1}\Sigma_c^{xy}(\Sigma_c^{yy})^{-1}(\Sigma_c^{xy})^T \mathbf{A}_c = \lambda_c^2 \mathbf{A}_c$$

Hence, $\mathbf{A}_c$ can be written so:

$$\mathbf{A}_c = \frac{\mathbf{A}_c}{\sqrt{\mathbf{A}_c^T \Sigma_c^{xx} \mathbf{A}_c}}$$

Finally, taking the derivative of the objective function with respect to u_{cj} to calculate the fuzzy canonical correlation membership matrix u_{ij}, we have

$$u_{cj} = \frac{\exp(A_{cj})}{\sum_{m}^{C} \exp(A_{mj})} \tag{4.22}$$

where

$$\begin{aligned}
A_{mj} = {} & \frac{\alpha}{\beta}(\sum_{i=1}^{P}\sum_{k=P+1}^{P+Q}(x_{ji}-v_{mi})(x_{jk}-v_{mk})A_{mi}B_{mk} \\
& -\lambda_m\sum_{i=1}^{P}\sum_{k=1}^{P}(x_{ji}-v_{mi})(x_{jk}-v_{mk})A_{mi}A_{mk} \\
& -\lambda_m\sum_{i=P+1}^{P+Q}\sum_{k=P+1}^{P+Q}(x_{ji}-v_{mi})(x_{jk}-v_{mk})B_{mi}B_{mk} \\
& -\frac{1-\alpha}{\beta}\sum_{i=1}^{P+Q}(x_{ji}-v_{mi})^2
\end{aligned}$$

where $m = \{1, 2, 3, ..., C\}$ the total number of initial centroids provided by the user, j denotes the samples of feature vectors that are partitioned into C clusters. A_{mj} becomes large when when distance between the speech feature vectors and their cluster centroids is small. Similarly, A_{mj} becomes large when the correlation between the speech feature vectors and their cluster centroids is large. Thus, this hybrid technique yields fuzzy clusters that are based not only on minimizing the distance between the cluster centroids and the speech feature vectors but also on maximizing the correlation between them. In this way, we obtain the cluster centroids that are well representative of the data to be clustered. We therefore obtained robust cluster centroids for a GMM classifier.

The implementation of the GMM as a fuzzy canonical correlation-based classifier in our proposed architecture is same as we describe in Chapter 3. The main difference between both proposed approaches is the way for selecting Gaussian mixtures. For speaker-dependent accent-based IVR systems, we employed fuzzy canonical correlation analysis to improve the performance of the accent-based classification system.

4.4 Summary

In this chapter, we proposed fuzzy canonical correlation analysis for improving the performance of a speaker-dependent based next generation IVR system. This approach is based on two algorithms: fuzzy c-means clustering and canonical correlation analysis. In fuzzy c-means, each datum is assigned membership values indicating its degree of belonging to each of the given clusters. This value is called a fuzzy membership value and varies from 0 to 1. Unlike hard clustering, soft clustering has an inherent capability to fuzzily partition data. Fuzzy clustering is the most suitable technique to resolve the problem of fuzziness between phoneme boundaries and their classes. Fuzziness in this case results from variations in speech signals and coordinated movements of the speech organs. The main principle task of canonical correlation analysis is the find maximum correlation between the cluster centroids and speech feature vectors. Thus, this hybrid methodology yields fuzzy clusters that are based not only on minimizing the distance between the cluster centroids but also maximizing the correlation between the cluster centroids and the speech feature vectors. In this chapter, we provided a framework for speaker-dependent accent-based IVR. The implementation of a speaker-dependent classification module and switching between the acoustic models based on a particular accent group significantly improve the accuracy of call-routing systems. Speaker-dependent accent-based ASR system allows a computer to identify the words spoken by different speakers into a microphone or telephone and route the call to an appropriate acoustic model that has been thoroughly trained and best adapted on the speech utterances recorded by such a speaker.

In the next chapter, we provide assessment and analysis of the proposed methodologies for speaker-independent and speaker-dependent IVR systems.

Chapter 5

Assessment of the Methodologies Proposed

To evaluate our proposed approaches, we applied it to classify speaker accents by using three databases: TIMIT, the speech accent archive database, and the FAE. The TIMIT speech database was designed to train and evaluate the performance of automatic speech recognition systems. It consists of the utterances of speakers representing the eight major dialect regions of American English [108]. From the TIMIT database, we selected three examples of American accent to train and test our proposed method. These particular three varieties were chosen because the number of speakers in each is almost equal. Next, we used the speech accent database to evaluate the performance of the proposed method. From this database, we selected four different foreign accents: American, Arabic, Russian, and Chinese. This database contains a very small number of utterances for each specific accent. Due to this constraint, we selected sixteen utterances per accent. Furthermore, we evaluated the performance of the proposed method using a degraded speech database, FAE.

As for the features, we used MFCC, the first three formants, and energy features. Basically, we used a hybrid scheme of prosodic and phonetic features. During the training and evaluation of our proposed method, we implemented different numbers of features (i.e., 12 MFCC, 39 MFCC, etc.) with the first three formants and energy. In addition to suitable features, we also needed an appropriate number of Gaussian mixtures, which were analyzed by an experimental study, as the selection of suitable Gaussian mixtures has a significant impact on the performance of an accent-based IVR system.

This chapter is organized as follows: First, we present the experimental results of using our proposed approach for speaker-independent accent classification employing the TIMIT database. This section is followed by an evaluation of the system using the speech accent archive database. Next, we use a degraded speech database to further evaluate the performance of the system. Finally, we provide the classification performance of a speaker-dependent system using the TIMIT, speech accented archive, and FAE databases.

5.1 Speaker Independent Accent Classification

In this section, we describe a number of experiments we performed using the three speech databases, the TIMIT, speech accent, and foreign accented English databases. The TIMIT and speech accent databases are recorded at 16 kHz, and the foreign accented English database is recorded at 8 kHz.

In the following subsections, first we present an evaluation of a speaker-independent system by employing a distance metric learning and evolution strategy-based Gaussian classifier using the TIMIT database. This section is followed by an evaluation of the system using the speech accent database. Finally, we present experimental results using the foreign accented English database.

5.1.1 Evaluation of the Proposed Approach using the TIMIT database

To evaluate our proposed approach, we applied it to classify speaker accents by using the TIMIT database. We selected two dialect regions of American English (i.e., Northern Midland and Western) to train and test our proposed method. In our experiments, we consider each dialect region as an accent group. We experimented by selecting randomly 10 speakers from each accent group (i.e., 100 utterances for each accent). We divided each accent group data set into training data and testing data sets. For the training data set, we selected 80 utterances for training and 20 utterances for evaluation (i.e., 80% for training and 20% for testing). During the data set preparation phase, we carefully selected those speakers in the testing data set that were not available in the training data set. Thus, we avoided the condition of speaker overlap. In short, we conducted experiments in a speaker-independent environment.

For the training and evaluation of classifiers we used five-fold cross validation. We divided the whole training and testing data set into five classes. Each class has a set of 20 utterances. For the first round of training and testing of the accent classification system, the training data set includes classes 1, 2, 3, and 4. The testing data set includes class 5. In the second round of training and testing of classification model, class 1 is switched with class 5. Now, class 1 becomes a testing data set. Similarly we repeated the whole set of experiments five times to obtain average classification results for each set of Gaussian mixtures.

During the training and testing phase, we experimented with our proposed method to obtain Gaussian mixtures suitable for producing higher classification results. The experimental results are shown in Table 5.5. With this approach, we did not need to select the initial seeds for k-means clustering. Our evolutionary-based k-means clustering method solves this problem automatically. To compare the classification performance of our approach, we trained and tested it with the GMM classifier, employing a standard k-means algorithm with the same parameters and the same training data and testing data sets. The classification results obtained using different numbers of Gaussian parameters are as follows: 67% with 4 Gaussian mixtures, 65% with 8 Gaussian mixtures, 60% with 16 Gaussian mixture, 60% with 32 Gaussian mixtures, and 55% with 64 Gaussian mixtures (as shown in Table 5.1).

Table 5.1: Classification results using GMM

ID	No. Gaussian mixtures	Accuracy
1	4	65%
2	8	65%
3	16	60%
4	32	60%
5	64	55%

We also employed the same training and testing data sets for the HMM. We achieved accuracy rates using different numbers of HMM states and Gaussian mixtures as follows: 55% with five HMM states and six Gaussian mixtures, 60% with three HMM states and six Gaussian mixtures, 65% with three HMM states and four Gaussian mixtures, 67% with three HMM states and two Gaussian mixtures, and 57.50% with three HMM states and eight Gaussian mixtures (as shown in Table 5.2).

Table 5.2: Classification results using HMM

ID	No. hidden states	No. Gaussian mixtures	Accuracy
1	4	6	55%
2	3	6	65%
3	3	4	65%
4	3	2	67.50%
5	3	8	57.50%

Similarly, in the case of the vector quantization Gaussian mixture model, we used different numbers of codebook values (i.e., 32 to 256). The accent classification results were obtained using different numbers of codebook values and are as follows: 47% with 32 codebook vectors, 60% with 64 codebook vectors, 57.50% with 128 codebook vectors, and 52.50% with 256 codebook vectors (as shown in Table 5.3).

Table 5.3: Classification results using VQ-GMM

ID	No. codebook vectors	Accuracy
1	32	47.50%
2	64	60%
3	128	57.50%
4	256	52.50%

We also implemented RBF [109] to compare the performance of our proposed method. We trained and tested RBF with different numbers of hidden neurons. The classification results using different numbers of hidden neurons are as follows: 47.50% with thirty hidden neurons, 65% with sixty hidden neurons, 60% with 90 hidden neurons, and 52.50 with a hundred neurons (as shown in Table 5.4).

Table 5.4: Classification results using RBF

ID	No. of hidden Neurons	Accuracy
1	30	60%
2	60	52.50%
3	90	47.50%
4	100	65%

5.1.2 Evaluation of the Proposed Approach using the Speech Accent Database

We conducted a number of experiments using the speech accent archive database. We selected four accent groups, English Arabic, English Chinese, English Russian, and American English. The English Arabic accent group means that the speakers

Table 5.5: Comparison of different methods

ID	Accent classifier type	Accuracy
1	Proposed Method	75.0%
2	GMM	67.50%
3	HMM	67%
4	VQ-GMM	60%
5	RBF	65%

are native Arabic speakers, but recorded the utterances in English. These non-native speakers often substitute their native phoneme pronunciation when speaking English as a second language. This substitution makes accent identification a difficult and complicated task.

In each accent group, we made sure that there were at least four to five female speakers. For example, in the English Arabic accent group, we have sixteen non-native speakers five female speakers and eleven male speakers. The number of speaker utterances is less than in the TIMIT database, but the utterances themselves are also more difficult and lengthy then those in the TIMIT database. Each utterance is based on a paragraph. The English text of each utterance is

Please call Stella. Ask her to bring these things with her from the store: Six spoons of fresh snow peas, five thick slabs of blue cheese, and maybe a snack for her brother Bob. We also need a small plastic snake and a big toy frog for the kids. She can scoop these things into three red bags, and we will go meet her Wednesday at the train station.

The speech utterances in the TIMIT database are much shorter than those in the speech accent archive database:

She had your dark suit in greasy wash water all year.

Using the speech accent archive database, we experimented only with the GMM classifier instead of all the other techniques we had experimented on using the TIMIT database, because it is well-known and has already proved its superior classification performance as compared to HMM, VQGMM, and RBF. In this section, we compare the performance of our proposed approach with a GMM classifier that is based on a standard k-means clustering approach. To evaluate the performance of our proposed approach, we employed a four-fold cross validation technique. During the training and testing of our proposed approach, we divided the training and testing database into four classes. Each class consists of four speakers. In the first round of training our accent classifier, we selected classes 1 and 2 for training and classes 3 and 4 for testing. In the second round, we switched the class 1 training data with the class 3 utterances previously selected for testing the classifier. Thus, during this round, we had class 3 and 2 for the training of the classifier and class 1 and 4 for the testing. We repeated the experiments four times to obtain average classification results.

In the next section, we evaluate our proposed approach using different accent groups, English Arabic vs. American English, English Arabic vs. English Chinese, and English Arabic vs. English Russian.

English Arabic vs. American English

We first experimented using English Arabic and American English accent groups. Before providing a performance compression of our proposed approach, we provide accent classification results obtained with a GMM using standard k-means clustering algorithm as an accent classifier. The Gaussian mixtures were selected using a hit-and-trial method. We started training by selecting 8, 16, and 32 Gaussian mixtures. We manually selected these Gaussian mixtures as the initial seeds for the k-means clustering algorithm for finding mixtures using the GMM classifier. Finding mixtures in this way is very difficult and time consuming.

For each set of Gaussian mixtures, we employed a four-fold cross validation and obtained average classification results. For our 8, 16, and 32 Gaussian mixtures we achieved accent classification accuracy rates of 68.75%, 70.83%, and 70.00%, respectively. These results are also shown in Table 5.6.

Table 5.6: English Arabic vs. American English

Reference	Accent group	GMM Accuracy	Proposed Approach
1	8	68.75%	N/A
2	16	70.83%	75.00%
3	32	70.00%	N/A

In Table 5.6, we have used the notation N/A, which means that our proposed method only provides the optimal number of Gaussian mixtures. It checks all possible combinations of the Gaussian mixtures and results for the best combination of Gaussian mixtures. The term N/A means “not available” in our experiments.

English Arabic vs. English Chinese

In this section, we present the results obtained using English Arabic and English Chinese. To compare the performance of our proposed method, we trained and tested the method using cross validation. During the training and the evaluation phase, we used two accents groups: English Arabic speakers and English Chinese ones. Each group was divided into 4 classes, and each class included 4 non-native speakers. This grouping was employed for cross validation purposes. The cross validation procedure is same as that used in English Arabic vs. American English accent classification.

We conducted several experiments using cross validation with each set of Gaussian mixtures. First, we started experiments with the GMM using a standard k-means clustering algorithm in the model training process. Using conventional training, we were not sure which set of Gaussian mixtures would provide optimal classification results. Thus, we started training models initially with a smaller number of mixtures, 4 Gaussian mixtures. The classification result obtained using 4 Gaussian mixtures is 60.25%. Similarly, we trained and tested the models with 8, 16, and 32 mixtures to check all possible ways to get optimal classification results. For 8, 16, and 32 Gaussian mixtures, we got 58.13%, 56.50%, and 56.50%, respectively.

Similarly, we trained and tested our proposed approach and obtained an accuracy rate of 68.75%. The experimental results are shown in a tabular form for quick comparison in Table 5.7.

Table 5.7: English Arabic vs. English Chinese

Reference	Accent group	GMM Accuracy	Proposed Approach
1	4	62.25%	N/A
2	8	58.13%	68.75%
3	16	56.50%	N/A
4	32	56.50%	N/A

During experiments we noted that the best classification results using our proposed approach was 75%.

English Arabic vs. English Russian

We conducted a number of experiments using English Arabic vs. English Russian and experimented using the same criteria for cross validation. The results are shown in Table 5.8.

Table 5.8: English Arabic vs. English Russian

Reference	Accent group	GMM Accuracy	Proposed Approach
1	4	58.33%	N/A
2	8	58.33%	N/A
3	16	61.27%	67.50%
4	32	60.33%	N/A

5.1.3 Evaluation of the Proposed Approach using Foreign Accented Database

We also conducted a number of experiments with the FAE database. We selected two different accent groups, English Arabic and English Farsi (i.e., Persian). This database allowed us to evaluate the performance of our proposed approach using a noisy database. This database is noisy in the sense that it includes the telephone transmission medium and switching equipment noise during recording of the database. It also gives degraded speech utterances as compared to the TIMIT and the speech accent archive database. The FAE database is recorded at a sampling rate of 8 kHz, as compared to TIMIT, recorded at 16 kHz.

During the training and testing of the accent classifier, we used 100 different speakers in each accent group. We also experimented with FAE using cross validation. In each round, we divided the training and testing data set into four classes, each class including 15 speakers. During the first round, we selected classes 1 and 2 as our training database and classes 3 and 4 as the testing database. In the second round, we switched class 1 with class 3 and made this the training database and classes 1 and 4 our testing database. Similarly, we repeated our experiments four times to get average classification results for each set of Gaussian mixtures.

In the next section, we describe our experiments with English Arabic vs. English Farsi. The experimental results are provided in Table 5.9.

English Arabic vs. English Farsi

We conducted a number of experiments using the FAE database to evaluate our proposed approach for noisy data. We started experiments with 4, 8, 16, and 32 Gaussian mixtures as the initial seeds for the k-means clustering algorithm for finding appropriate mixtures for a GMM classifier. Experimental results using a standard k-means clustering are provided in Table 5.9. To compare the performance of our proposed approach, we use the FAE data base for training and testing our modified GMM-based accent classifier. Table 5.11 shows that our proposed approach is superior to GMM training based on a standard k-means clustering algorithm.

Table 5.9: English Arabic vs. English Farsi

Reference	Accent group	GMM Accuracy	Proposed Approach
1	4	58.00%	N/A
2	8	52.00%	N/A
3	16	50.00%	N/A
4	32	58.00%	60.00%

5.2 Speaker Dependent Accent Classification

As explained in Chapter 4, the proposed method uses FCCA to classify the accents of different speakers. The experimental results presented in this section show that the Fuzzy-based accent classification approach enjoys performance superior to that of the most extensively studied and implemented approach (i.e., k-means-based GMM). To evaluate our proposed system, we conducted experiments using the TIMIT database (i.e., three different dialect regions from the TIMIT database: Northern, North Midland, and Western). We used 300 utterances for each accent.

In our proposed method, the accent classification based on FCCA includes feature extraction from a raw speech signal and the training of a GMM model. We applied 12 MFCCs with the first three formants and energy. Before the speech feature extraction, we removed the silence portion of speech (the pauses). This removal increases the overall performance of the ASR system. Next, we applied a fuzzy clustering technique to fuzzily partition the feature vectors. In this step, the clustering algorithm assigns membership values to the feature vectors. As we explained in the proposed method section, there is an overlapping between vowels and consonants, resulting in a fuzziness between phoneme classes and phoneme boundaries. Hence, our approach uses the inherent property of fuzziness to deal with the overlapping phenomenon. Fuzzy clustering performs better than k-means. The overall procedure for training of the proposed method is shown in Figure 5.1. Finally, a GMM model is trained with this fuzzily partitioned data.

Using the TIMIT database, we trained three different GMM models without taking into account the impact of gender identification. Gaussian models are trained with a mixture of speakers (i.e., males and females). It is believed that performing the gender identification before the training of GMMs improves the overall performance of the system. To evaluate the performance of the proposed system, we experimented by selecting different numbers of Gaussian components.

5.2.1 Evaluation of the Proposed Approach and k-means GMM using TIMIT database

The k-means clustering approach is well known and extensively implemented for the training of GMMs for accent identification. The overall procedure for this training is same as that shown in Figure 5.1, with the exception of the clustering module. In this approach, k-means is applied instead of fuzzy clustering. We applied all the same parameters to this scheme and compared the performance of our proposed method. We used the same Gaussian mixtures, number of iterations, training database, and testing database (i.e., TIMIT). The overall performance of the accent classification system based on k-means is shown in Table 5.10. We conducted a number of experiments with different Gaussian mixtures to analyze the best performance of each model.

It is clear from Table 5.10 that our proposed method outperforms the approach most widely used for accent classification. As we discussed in the literature review, GMM with k-Means clustering is well-known and extensively implemented for accent classification tasks. All the results that are shown in Table 5.10 were obtained using the TIMIT database with 12 MFCCs, the first three formants, and energy features. In this section, we further evaluate the performance of the proposed approach using other available speech databases for accent classification.

In our proposed methodology, the main objective is to maximize the correlation between the feature vectors and their respective fuzzy membership values. In this approach, we first extract speech features based on MFCCs, formants, and energy from the raw speech signal. This is followed by silence removal from the speech signal. Next, we apply fuzzy clustering to assign fuzzy membership values to each feature vector. This results in two sets of vectors to help us identify the correlation between them based on CCA. The overall flow (procedure) for the training of the system is shown in Figure 5.1.

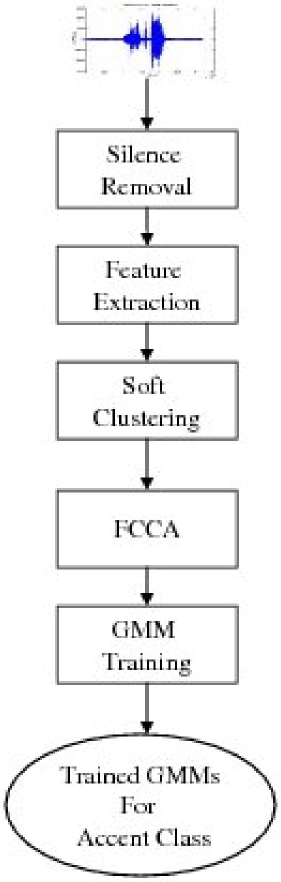


Figure 5.1: Training procedure for FCCA

To evaluate the performance of the proposed method with other benchmark schemes, we compared the experimental results using the TIMIT database, the speech accent, and the FAE databases. For comparison, we investigated the most widely used techniques in the area of accent classification: GMM, HMM, Vector codebooks, and RBF. We conducted experiments with these methods with the same databases and adjusted the model parameters in the same way.

To analyze the impact of our proposed method on the overall performance of accent-based classification, we selected randomly four Gaussian mixtures, as shown in Table 5.10. During experiment, we noticed that by using 39 MFCC, the first three formants, and energy features, we obtained 89% vs. 85.76% accent classification results using 32 Gaussian mixtures.

Table 5.10: Proposed method vs. k-means GMM

ID	Gaussian Mixtures	No. Itts.	Proposed method	k-Means GMM
1	8	100	71.11	70.56
2	16	100	77.44	75.56
3	17	100	78.50	75.56
4	32	100	85.83	83.33

The same experiments were conducted with 25 MFCCs, the first three formants, and energy features. The experimental results are shown in Table 5.11.

Table 5.11: Speech accent archive database using 25 MFCCs

ID	Gaussian Mixtures	No. Itts.	Proposed method	k-Means GMM
1	8	100	78.50	75.0
2	17	100	87.50	81.57
3	25	100	83.56	81.25

5.2.2 Evaluation of the Proposed Speaker Dependent Approach vs. Other Classifiers

We experimented with different classifiers that are those most used in the literature of accent classification, such as HMM, VQGMM, and RBF. For each techniques, we employed the cross validation criterion, and in the case of speaker-independent accent classification system.

For HMM, we also employed the same training and testing data sets as we experimented with for the FCCA-based accent classifier. We experimented using different numbers of HMM states and Gaussian mixtures, such as five HMM states and six Gaussian mixtures, three HMM states and six Gaussian mixtures, three HMM states and four Gaussian mixtures, three HMM states and two Gaussian mixtures, three HMM states and eight Gaussian mixtures. The highest accuracy rates are mentioned in Table 5.12.

Similarly, in the case of vector quantization GMM, we experimented using different numbers of codebook values, such as 32, 64, 128, and 256 codebook vectors. The highest classification results using this technique are shown in Table 5.12.

We also implemented RBF [109] to compare the performance of our proposed method. We trained and tested RBF with different numbers of hidden neurons, such as 30, 60, 90, and 100 hidden neurons. The highest accuracy score is shown in Table 5.12.

Table 5.12: Overall classification results

ID	Method	No. Itts.	Accuracy
1	Proposed	100	89.32
2	GMM	100	85.76
4	HMM	100	84.79
5	VC-HMM	100	70.00
6	RBF	100	65.00

5.2.3 Evaluation of the Proposed Method using the Speech Accent Database

The speech accent archive database was developed for the purpose of foreign accent identification. We selected four different accents from it: Arabic, English, Russian, and Chinese. Each accent database has sixteen utterances. The ratio of male and female speakers for each accent is not equal. To evaluate the performance of our proposed method in comparison to other classification techniques, we again conducted experiments with a different number of Gaussian mixtures, demonstrating thereby that our models achieve optimized results using identical configurations. In this section, we performed experiments with pairs of accents (i.e. Arabic and English). We did not compare the proposed method to other classification techniques because the k-means GMM model already outperforms them. The accent classification results using Arabic and English accents are shown in Table 5.13. For each GMM model training, we used 8 utterances for the training of the accent classifier. The remaining 8 are used to test the classifier.

Table 5.13: Speech accent archive database with 13 MFCCs

ID	Gaussian Mixtures	No. Itts.	Proposed method	k-Means GMM
1	8	100	68.75	67.50
2	10	100	81.25	75.00
4	17	100	75.00	68.72

5.2.4 Evaluation of the Proposed Method using FAE Database

In this section, we further analyze the performance of our proposed method with a degraded speech database (i.e., low sampling frequency). From, this database, we selected Arabic, Farsi, Russian, and English. We conducted experiments on a pair of accents to evaluate the accent classification performance of the proposed method with band-limited speech data. We compared the accent classification performance with a k-means GMM classifier, as the classification performance of the HMM model is very low (58%) with the FAE database.

In these experiments, we randomly selected 100 utterances per accent. We used 80 utterances for the training of each accent model and 20 utterances per accent to evaluate the classification of the accent classifier. The experimental results are shown in Table 5.14.

Table 5.14: Accent classification using FAE database

ID	Gaussian Mixtures	No. Itts.	Proposed method	k-Means GMM
1	4	100	67.50	65.00
2	6	100	63.00	60.00
3	8	100	77.50	65.00
4	10	100	70.50	67.50
5	14	100	72.50	65.12
6	17	100	72.50	70.33

5.3 Summary

This chapter has described our experiments using three databases: TIMIT, the speech accent archive, and the FAE database. In the case of the TIMIT database, we experimented using speakers from three dialect regions. To evaluate the performance of the proposed methodologies with the TIMIT database, we conducted a number of experiments using HMM, GMM, vector quantization GMM, and RBF. We found experimentally that our proposed approaches are more efficient than the GMM classifier based on k-means clustering approach, which is a widely used technique for accent classification.

To further evaluate our proposed approaches for speaker-independent and speaker-dependent accent classification systems, we conducted a number of experiments with different accent categories, such as English Arabic vs. American English, English Arabic vs. English Chinese, and English Arabic vs. English Russian. We found that our proposed approaches outperform the well-known techniques in the literature of accent classification. Finally, we employed the FAE speech database to evaluate the performance of our proposed methodologies and achieved higher classification results for both speaker-independent and speaker-dependent applications.

In the next chapter, we provide our conclusions and future directions for research.

Chapter 6

Conclusion and Future Research

This thesis has reviewed the problem of accent classification in ASR systems. Classification has a tremendous impact on their performance. Researchers have tried to tackle this problem by applying HMM models, GMM models, and accent-based pronunciation dictionary-based approaches. As accents vary widely from country to country and even vary between communities, this accounts for the fact that the accuracy of accent classification methods remains unsatisfactory. There are many factors that make accent identification a challenging research issue but, generally, the main factors are as follows: inter- and intra-speaker variations, background noise, resistance variations in a local loop, and noise due to telephony transmission and switching mechanisms.

We conducted a number of experiments using three different databases. We find that when the speech accent archive database is used, results are more accurate. However, the results are lower using the FAE database because it is a band-limited, or low quality, voice database. The FAE database is recorded at 8 kHz, while the speech accent and the TIMIT databases are recorded at 16 kHz. We also note that increasing the number of Gaussian components does not necessarily increase the accuracy of the Gaussian classifier. Accuracy truly depends on the size and type of the training data. It is generally suggested in the literature that 32 Gaussian components is the best choice, a tradeoff between the training time and the accuracy of a classifier.

The main important point for discussion is that the DML outperforms all proposed methods in a speaker-independent application, whereas the fuzzy canonical correlation-based GMM classifier outperforms all others in a speaker-dependent application. We also noticed that NSES-based GMM outperforms k-means-based GMM because a hybrid clustering approach avoids the situation in which the k-means clustering algorithm becomes trapped in a local minimum. The hybrid clustering approach suggests some freedom of choice and variability in Darwinian evolution. Thus, only the fittest survive and all others are eliminated. We thereby get more suitable and robust individuals as Gaussian components. The main insight advantage on the proposed approach is that we employ the benefits of both the global and local search capabilities of the NSES and EM algorithms, respectively. Consequently, we can confidently say that fuzzy canonical correlation-based GMM works well in the case of speaker-dependent applications, whereas the NSES-based GMM outperforms most of the existing techniques in the literature of accent classification for speaker-independent applications.

For speaker-dependent applications, we implemented a fuzzy canonical correlation-based Gaussian classifier for accent classification of three accent classes. The main contribution of canonical correlation is to fine tune the membership values and thereby maximize the correlation within classes and maximize out-of-class variations. A hybrid clustering approach for optimizing the Gaussian mixtures yields more accurate classification results than the other methods proposed in the accent classification area. The main idea behind employing the fuzzy canonical correlation clustering approach is to deal with the fuzziness between phoneme boundaries and classes.

The main contribution of the distance learning metric algorithm with a closed-form solution is a faster convergence than with all other off-the-shelf and iterative algorithms. The proposed approach exploits the side information that is based on dissimilar points between two classes of accent. We transfer each accent group to a new space where the Euclidian distances between similar and dissimilar points are at their minimum and maximum, respectively.

We also tested Linear Discriminant Analysis (LDA) for accent classification. The classification results are less satisfactory than those obtained with our proposed method, perhaps because of the reduction in dimensionality. Even though linear discriminant analysis is a very common classifier, due to the dimensionality reduction of the speech features, it loses accent classification accuracy. Our proposed method is also based on a dimensionality reduction technique, but the information extracted from the dissimilar pairs of accent groups as side information supplements the classification accuracy.

More research is needed in real-world situations, especially with regard to language models and natural language understanding. Nevertheless, we have achieved encouraging results in the current experiments. There remains, however, the need to further improve the non-dominated sorting mechanism to reduce the time complexity of the evolution strategy-based k-means clustering algorithm.

APPENDICES

Appendix A

MFCCs Feature Vectors

MFCCs features are the best ones for speaker recognition and identification and can be extracted as

- Pre-Emphasis:

  Pre-emphasis filtering is used to compensate the loss of -6 db/octave roll-off. This compensation is only needed for voiced speech signals. However, for simplicity, pre-emphasis is normally applied to unvoiced speech signals as well. The filtering action may be achieved as follows:

  $$y[n] = x[x] - \alpha x[n-1] \tag{A.1}$$

  where $y[n]$ is the current output of the pre-emphasis filter, $x[n]$ is the current input to the filter, $x[n-1]$ is the previous input sample to the filter, and α is a constant. The range of α normally lies between 0.9 and 1. If α is selected as 1, the pre-emphasis process is skipped.

  Taking *z-transform* of Equation (A.1) and the transfer function of the filter gives

  $$H(z) = 1 - \alpha z^{-1} \tag{A.2}$$

- Windowing

  To preserve the slow varying characteristic transfer function of the vocal tract, we need a windowing function. There are many such functions, but a very common one in speech applications is the hamming window, defined as

$$h(n) = \begin{cases} \alpha - (1-\alpha)\cos(2\pi\frac{n}{N-1}), & 0 \leq n \leq N-1 \\ 0 & otherwise \end{cases} \tag{A.3}$$

  where $\alpha = 0.54$ and N is the length of the window. The length of the window varies from application to application; a large sized window gives a good frequency-domain resolution but poor time-domain resolution. As a rule of thumb for speech applications, a window size of 20 ms is generally considered a good choice.

- Power Spectrum

  After characterizing the input speech signal in framing and widowing, we need to compute the power spectrum of the speech segment by performing DFT, and than compute its magnitude squared as

$$S[i] = (real(X[i]))^2 + (imag(X[i]))^2 \tag{A.4}$$

- Mel Spectrum

  After calculating the power spectrum, we need to calculate the mel-spectrum. It is calculated so:

$$\tilde{S}[k] = \sum_{i=0}^{N/2} S[i]M_k[i], \tag{A.5}$$

  where k = {0, 1, 2, K-1} and represents the total number of mel filters. The index, N, is the length of the DFT.

- Mel Cepstrum

After implementing the above steps, we are now able to obtain the Mel cepstrum as:

$$c[n] = \sum_{i=0}^{K-1} ln(\tilde{S}[i]) cos(\frac{\pi n}{2K}(2i+1), \tag{A.6}$$

where n = {0, 1, ...,C-1} and is called the number of cepstrum coefficients. The *Delta* and *Delta*^{2} mel cepstrum coefficients can be calculated as

$$\Delta c[n] = c[n+1] - c[n] \tag{A.7}$$

$$\Delta^2 c[n] = \Delta c[n+1] - \Delta c[n] \tag{A.8}$$

Appendix B

Syllable Structure

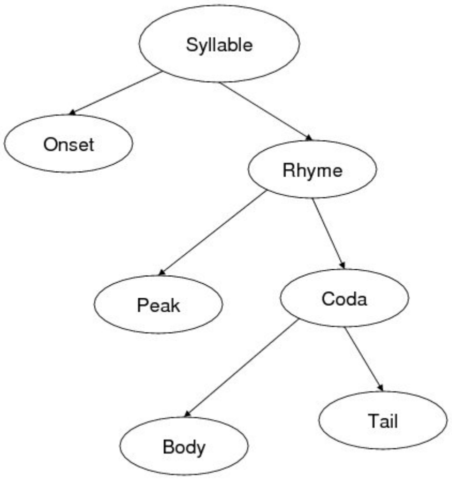


Figure B.1: Syllable structure

Appendix C

Impact of Clustering Techniques on Interactive Voice Response system

Clustering is one of the most widely used tools for understanding and exploring data structures and is widely used across all disciplines, from engineering to social sciences, from computer science to biological sciences. The main goal of clustering is to determine the intrinsic grouping of unlabeled data. The focus of this chapter is to describe how the performance of clustering algorithms' can be improved by incorporating biology inspired techniques, such as Evolutionary Strategy and genetic algorithm. In our proposed approach, we employ the k-means one to calculate the initial centroid vectors and then ES to refine these cluster centroids. When finding the cluster centroids, the k-means clustering algorithm tends to converge faster than ES, but generally with less accurate clustering. To address this issue, we propose a hybrid clustering scheme to cluster the data and then provide these cluster centroid vectors to the GMM as its components. In this section we discuss the k-means clustering algorithm in detail.

C.1 k-means Clustering

k-means clustering is one of the most widely used tools for exploring data structures [110][55][111, 112]. In this clustering method, data items belong to one and only one cluster, with a membership value of either {1, 0}. Data items are classified into groups based on the attributes or features. This grouping is based on measuring a distance between data objects. We used Euclidean distance for measuring the distance between the data objects. The k-means clustering algorithm was introduced by J. B. MacQueen in 1967. Initially, it selects K number of data points randomly from the data set to be partitioned. Each data object is assigned to a particular cluster centroid based on the similarity between the data objects. To make the cluster centroids the centre of gravity of each partition, at each iteration, the arithmetic means of each cluster is calculated and the old centroid is replaced by a new one. This process of assigning the data objects to the centroids and recalculating the centroids is repeated until stable clusters are formed, that is when there is no change in the arithmetic in the arithmetic mean of the clusters. The main objective of k-means is to identify clusters of similar objects in the feature space. This similarity or dissimilarity is quantified by the measure of distance between two objects. Let $\mathbf{x}_1 = (x_{1,1}, x_{1,2}, ..., x_{1,d})$ and $\mathbf{x}_2 = (x_{2,1}, x_{2,2}, ..., x_{2,d})$ be any two points in the feature space. The most common approach used to measure the distance between points is Euclidian distance and may be defined between $\mathbf{x_1}$ and $\mathbf{x_2}$ as follows:

$$
\begin{aligned}
h(\mathbf{x}_1, \mathbf{x}_2) &= \left[\sum_{j=1}^{d} (\mathbf{x}_{1,j} - \mathbf{x}_{2,j})^2 \right]^{1/2} \\
&= \|\mathbf{x}_1 - \mathbf{x}_2\|_2 \,,
\end{aligned}
$$

where d is the dimension of feature vector $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively. In the literature the Manhattan distance for measuring the distance between data objects is also used. It can be represented so:

$$\begin{aligned} h(\mathbf{x}_1, \mathbf{x}_2) &= \sum_{j=1}^{d} (\mathbf{x}_{1,j} - \mathbf{x}_{2,j}) \\ &= \left\| \mathbf{x}_1 - \mathbf{x}_2 \right\| . \end{aligned}$$

The Euclidian distance metric has an intuitive ability to measure the proximity between data objects. It works well for data sets that have compact and isolated clusters [113]. Several researchers have introduced different methods for measuring the distance between the data objects [114][115][116].

The k-means algorithm is a non-hierarchical, partition-clustering approach. It is simple and fast in its attempts to locally improve an arbitrary K initial centroids. In the k-means algorithm the value of K is supplied by the user. In practical applications, the k-means algorithm must be run many times to get good cluster centroids. However, k-means clustering is popular because it is easy to implement. Its time complexity is $O(n)$, where n is the total number of data patterns. The main problem with k-means clustering is the selection of initial centroids. The performance of k-means depends on these initial centroids, and due to this initial selection, the algorithm may converge to local minima. The partition performance of the algorithm depends on the initial selection of centroids and shown in Fig. C.1. If we start with selecting initial centroids A, B, and C as the initial partitions, then, we end up with final partitions of {(A), (B, C), (D, E)}. The final squared error is much larger for this partition than for the best partition {(A, B, C), (D), (E)}. The correct three-cluster solution is obtained by selecting A, B, and D as the initial cluster centerfolds.

The main objective of k-means is to optimize the sum of squared error. The optimization is over the assigned number of clusters. There is no automated way in a standard k-means algorithm to increase or decrease the the number of clusters for seeking a lowest error. Thus, k-means clustering, due to this constraint, might not be able to find global minima.

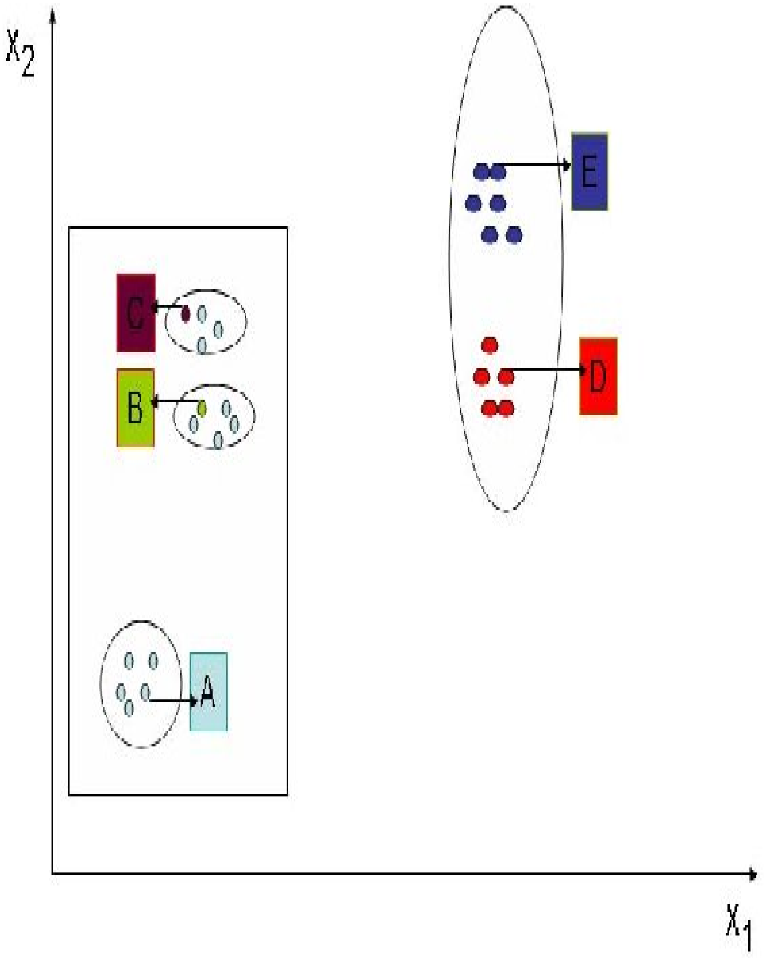


Figure C.1: The k-means clustering algorithm is sensitive to initialization

In the k-means algorithm, clustering performance is very sensitive to the input selection of cluster centroids [80]. In the standard k-means algorithm, we are not able to select the cluster centroids that yield a globally optimum solution [81]. There are two main drawbacks with the standard k-means algorithm: insufficient ways to select the optimum number of input number of clusters and no clustering solution that provides a globally optimum solution [82] . Thus, an initialization process that randomly generates the initial centroids might produce different clustering solutions on the same data.

Algorithm C.1 k-means clustering algorithm

```
1: Choose $K$ points, such as $\mathbf{m}^{(1)}, ..., m^{(K)}$ in a $d$-dimensional feature space. These points will serve as cluster centroids, such as, $\mathbf{C}^{(1)}, ..., C^{(K)}$. Normally, this initial choice of centroids is from a random sample of $\mathbf{x}$ feature vectors from the data set having $n$ objects .
2: Allocate $n$ objects, such that, $\mathbf{x} \in \{1, 2, ..., n\}$ to $k = \{1, 2, .., K\}$ clusters by assigning a feature vector $\mathbf{x}$ to a cluster $C^k$, provided that $\mathbf{x}$ is closer to $m^{(k)}$; otherwise, assign them to other cluster.
3: After allocating $n$ objects, recompute the $K$ centroids, $\mathbf{m}^{(1)}, ..., m^{(K)}$ by taking the mean value of the data objects within the cluster, such as:
       $\mathbf{m}^{(k)} = \frac{1}{N} \sum_{\mathbf{x} \in C^k} \mathbf{x},$
   where $k = \{1, ..., K\}$ and $N$ is the total number of data points in $C^{(k)}$.
4: Stop the algorithm if there is no significant change in K means centroids or after $i > I_{max}$, maximum number of iterations.
5: i = i + 1
6: Go to step 2
```

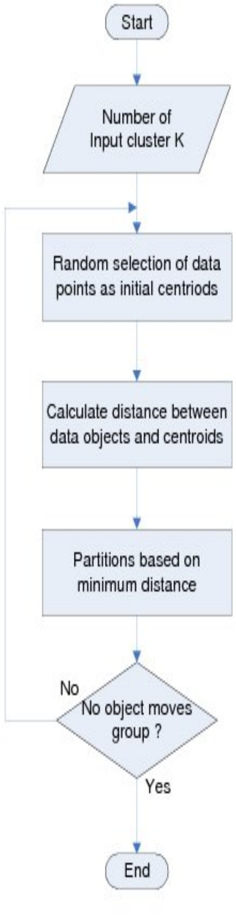

Start
Number of Input cluster K
Random selection of data points as initial centroids
Calculate distance between data objects and centroids
Partitions based on minimum distance
No
No object moves group ?
Yes
End

Figure C.2: k-means clustering algorithm

Appendix D

Overview of Optimization Techniques for k-means Clustering

The main goal of single objective optimization is to find the best solution that corresponds to the minimum value of a single objective function to fulfill all different objectives in one. This kind of optimization is a useful tool in our problem to get optimized cluster centroids. Hence, the problem with this objective function is that usually it cannot provide alternative solutions. Thus, there is no trade-off between different objective functions to achieve an optimized solution. However, in a multiobjective optimization, there is not a single optimal solution, so the interaction between different objective functions yields a set of solutions, known as trade-off non-dominated or Pareto-optimal solutions. Pareto optimum was originally proposed by F. Y. Edgeworth [117] and latter generalized by V. Pareto [118]. Thus, the most commonly term in the literature is known as Pareto Optimum. Regarding the definition of Pareto Optimality [119][120], it is defined as:

"$\mathbf{x}^*$ *is Pareto optimal if there exists no feasible vector* $\mathbf{x}$ *that would decrease some criterion without simultaneous increase in at least one other criterion in order to minimize the objective function."*

To further clarify the concept of multi-objective optimization, there are a few more definitions [119][120], such as, Pareto Dominance, Pareto Optimal Set and

Pareto Front. Generally, a multiobjective optimization problem consists of a number of objective functions associated with a number of constraints, such as equality and inequality. We can write a multiobjective problem as [121]:

$$\begin{aligned} \text{Minimize/Maximize} f_i(\mathbf{x}) & \qquad i = \{1, 2, .., N\} \\ \text{Subject to:} & \\ g_j(\mathbf{x}) \leq 0 & \qquad j = \{1, 2, .., J\} \\ h_k(\mathbf{x}) = 0 & \qquad k = \{1, 2, .., K\} \end{aligned}$$

where $\mathbf{x}$ is a N dimensional vector having N decision variables.

In the context of above object function, we explain the multiobject optimization concepts, such as:

- Pareto Dominance

  As, we are tackling the problem of minimization, so, if a vector x^1 is partially less than a vector $\mathbf{x}^2$ (i.e., $\mathbf{x}^1 < \mathbf{x}^2$), when all values of $\mathbf{x}^2$ are not less than $\mathbf{x}^1$. It means that the solution $\mathbf{x}^1$ is superior than solution $\mathbf{x}^2$. Thus, $\mathbf{x}^1$ is said to dominate $\mathbf{x}^2$.

- Pareto Optimal set.

  All solutions that dominate others are called sets of optimal solutions. These solutions are called non-inferior or efficient solutions, and their corresponding vectors are called non-dominated. Hence, a collection of these vectors is called a Pareto optimal set.

- Pareto Front

  When solutions are plotted in objective space, the non-dominated vectors are collectively shown as a collection of superior solutions. This collection of superior solutions is called a Pareto front.

D.1 Classical Optimization Techniques vs. Evolutionary Algorithms

In mathematics, the term optimization refers the study that determines the best solution of a problem among a set of solutions (alternatives) in hand in which one seeks to minimize or maximize a real function by selecting the values of real or integer variables from within an allowed set in a systematic way. Optimization techniques are now very common in everyday problems, such as industrial planning, resource allocation, scheduling, decision-making, cluster centroid optimization, etc. For example, how does a clustering algorithm decide a central gravity point for a group of data or a cluster that is the best representation of that cluster, or the number of clusters (i.e., partitions) that are suitable for this data set.

Optimization techniques, in general, can be traced to methods developed to optimize the logistic supplies to the armies during World War II. During war, any techniques that promised to improve the effectiveness of war efforts are desperately need by military commanders. The main objective was to improve the deployment of armies in different areas supported by logistic supplies, such as machines, weapons, and bombing patterns to achieve optimum safety from anti-submarine and anti-aircraft patrols.

The first method that was developed was the simplex method. The performance of the simplex methods was enhanced after the war when the first electronic computers were becoming available. Thus, we may say that the history of computing and optimization walk side-by-side. In the early age of electronic computers, a vast majority of all calculations was devoted to optimization using the simplex method. Starting from the simplex method to-date search and optimization methods can be divided into three categories: enumerative, deterministic, and stochastic (random). An overview of common optimization techniques is shown in Fig. D.1[119]. A brief overview of all techniques is provided in order to make clear why we selected evolutionary algorithms for selecting optimum cluster centroids and the number of

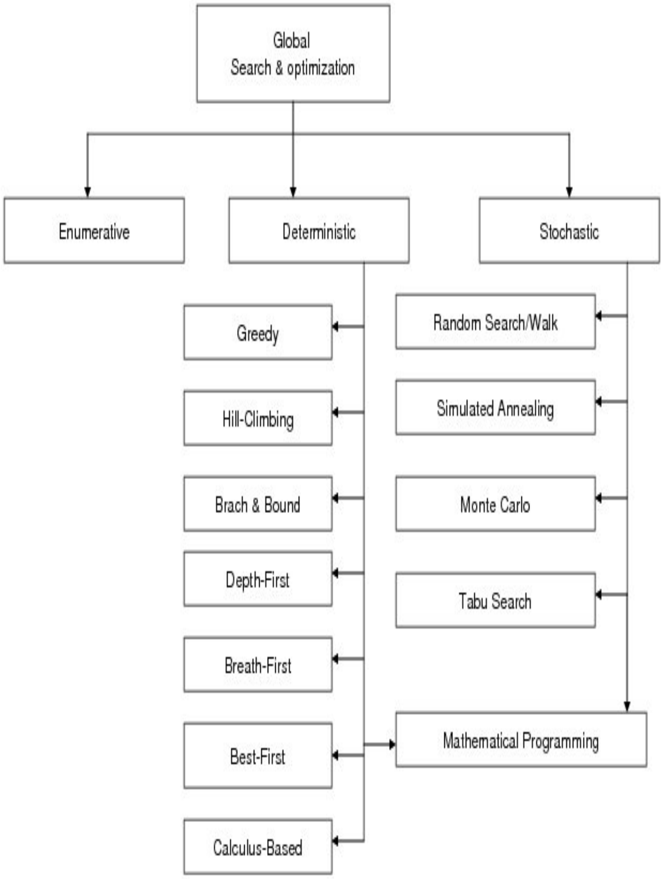


Figure D.1: Global optimization approaches

clusters for a given data set.

D.1.1 Enumerative Technique

Enumerative optimization techniques are the simplest optimization techniques that are used to evaluate every point in a search space in order. This non-randomness behavior makes the enumerative search strategy inefficient as the search space become large. Hence, most real world optimization problems are computationally intensive, and to effectively employ enumerative search techniques, one needs to implement a method that restricts the search space to reach an acceptable solution in an acceptable time [122].

D.1.2 Deterministic Algorithms

Deterministic algorithms are the most common and studied algorithms. Deterministic algorithms predict solutions by incorporating the problem domain knowledge. Generally, deterministic algorithms behave predictably and can be considered as graph/tree search algorithms. A brief overview of these is presented as follows.

Greedy Algorithm

A greedy algorithm is simple and straightforward. Normally, the greedy algorithm works in phases to build up solutions piece by piece, and always chooses the next piece that has the most advantages over the previous one, and calls these pieces locally optimum choices. Thus, when the algorithm terminates, the locally optima, solution might be considered globally optimum. Therefore, there are chances that one might get suboptimal solutions instead of global optimum, because the greedy algorithm always assumes the suboptimal solution as a part of global optimal solution [123][124]. Conclusively, if one does not need the best answer, then simple greedy algorithms are better than complex algorithms, because they are easy to implement.

Hill-Climbing Algorith

Hill-Climbing belongs to the family of local search algorithms. It always works in the direction of steepest ascent or descent depending on the nature of the objective function from the current position. This steepest ascent/descent property of the algorithm makes it a good choice for unimodal functions, as shown in Fig. D.2. But, its credibility is challenged when we employ it in a search space that has multiple local optima, plateaus, and ridges, as in Fig D.3. Hence, the presence of multiple local optima points in the search space reduce the effectiveness of the algorithm [125].

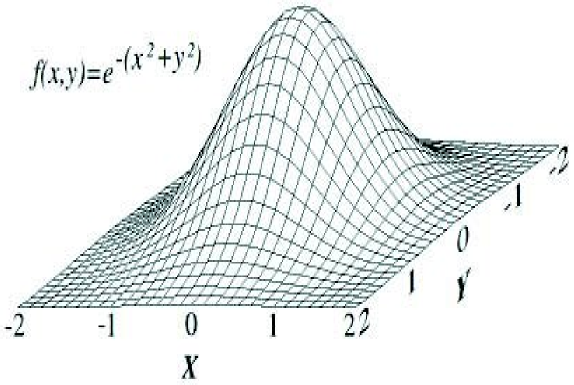


Figure D.2: Uni-modal search space

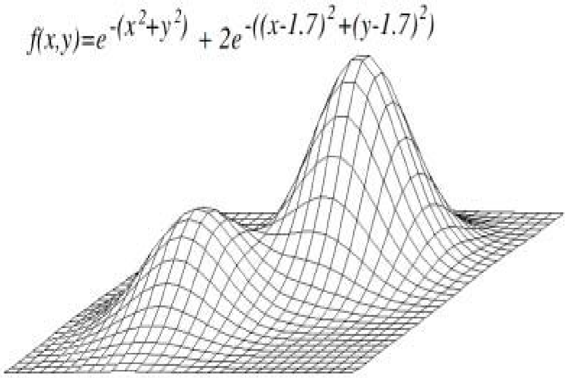


Figure D.3: Multi-modal search space

Branch and Bound Search Algorithm

Branch and Bound is a general purpose search algorithm for solving discrete and combinatorial optimization problems. It evaluates all potential solutions in an enumerative way. The key idea of BB algorithms is based on node (a set of solution) pruning. The pruning procedure maintains a global variable that records the minimum upper bound of the subregions visited so far. Any node whose lower bound value is greater than the value hold by the global variable can be discarded. Hence, if a lower bound for a node, x_{node}, is greater than the upper bound of another node, y_{node}, then x_{node} may be safely discarded from the search space. Nodes are discarded using a node estimator algorithm that determines whether the solution or node is promising [126] or not. Therefore, the effectiveness of BB algorithm fairly depends on node-splitting and upper and lower bound estimator algorithms.

Depth-First Algorithm

Depth-first is an mathematical tool used for tree, tree structure, or graph searching. The algorithm starts at a specific vertex (i.e., root node) making it a current node. The algorithm proceeds from the current node to the next sibling, and if the next node is not visited first, then this node becomes the next current node. However, if this node is already visited, then the algorithm backtrack to its previous current node. Each node will expand to its dead node, as shown in Fig. D.4. The process of finding tree structures terminates when backtracking leads back to the original root node.

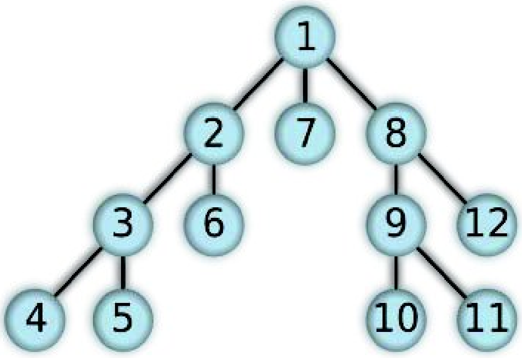


Figure D.4: Node expansion order

Breadth-First Algorithm

Breadth-First is similar to the depth-first algorithm, but its action of exploring nodes is different to that of DF algorithms. The difference between depth-first and breath-first at their start is shown in Fig. D.5 and Fig. D.6. At time $t = 0$, the blue colored lines show that depth-first algorithms only go to the next node in a sequence. The root node expands to its sub-nodes, but in breath-first the root visits all of its immediate nodes. The solid node indicates that it is visited at $t = 0$, and nodes that have "1" written inside indicate that they are visited at $t = 0$ in the case of breadth-first algorithms. There is another version of this family of search algorithms that starts to expand nodes from the high promising node. This way of searching graphs is called the Best-First search algorithm.

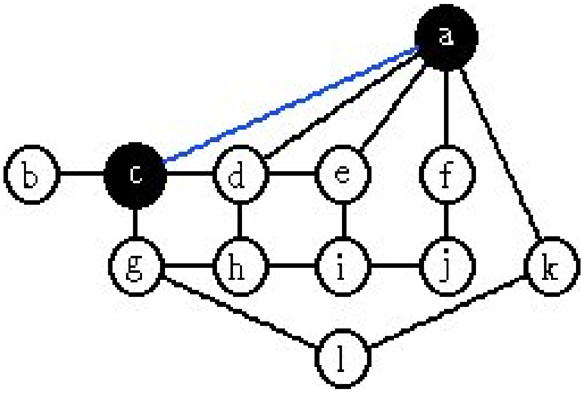


Figure D.5: BF Node expansion order

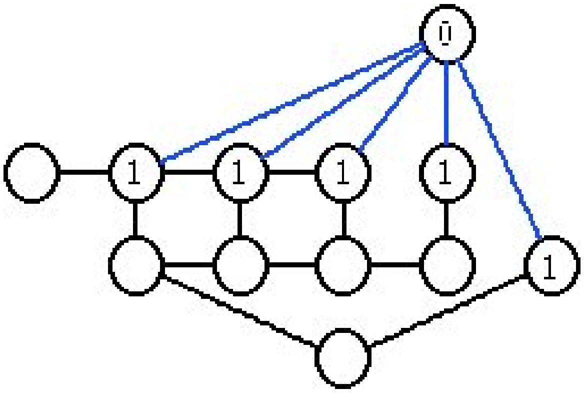


Figure D.6: Another node expansion order

D.1.3 Stochastic Search Algorithms

In real world applications, many optimization problems are high dimensional, discontinuous, multimodal, and NP-complete. Therefore, deterministic techniques are not suitable to solve them [127][128]. In many scientific and engineering applications, stochastic search algorithms outperform their counterparts. They are therefore becoming more popular for solving computationally hard combinatorial problems.

The following stochastic search and optimization techniques, such as random walk, Simulated Annealing (SA), Monte Carlo methods, Tabu search, and Evolutionary Computation, were introduced in the literature as alternative approaches.

Random Walk

Random walk a is mathematical tool to do optimization taking successive steps in random directions. Because of its randomness, it is widely applied to engineering applications, computer science, physics, economics, and ecology as a fundamental model for random processes in time. The behavior of the algorithm depends only on its previous position at some previous time. Hence, random walk is generally related to the diffusion processes.

Simulated Annealing

Simulated annealing SA is an algorithm that is based on an annealing analogy for global optimization problems in a large search space. It is generally used when the search space is discrete.

Evolutionary Computation

Evolutionary Computation is a general term for a number of stochastic algorithms that are based on natural evolution concepts, such as, reproduction, mutation, recombination, natural selection, and survival of the fittest. EC has been applied in a wide rage of engineering applications, from computer science to operational research. It is the most demanding research area for optimization problems, such as multi-modal and irregular search space, but still a young field. Under the umbrella of EC are a number of biological inspired search algorithms, such as, GA, ES, genetic programming, and evolutionary programming. All these algorithms are based on the concept of natural evolution and the Darwinian concept of “survival of the fittest." The fitness value associated with a solution (i.e., individual) determines whether this individual will take part in the next generation or not. All these algorithms under the umbrella of EC are generally referred to as evolutionary algorithms.

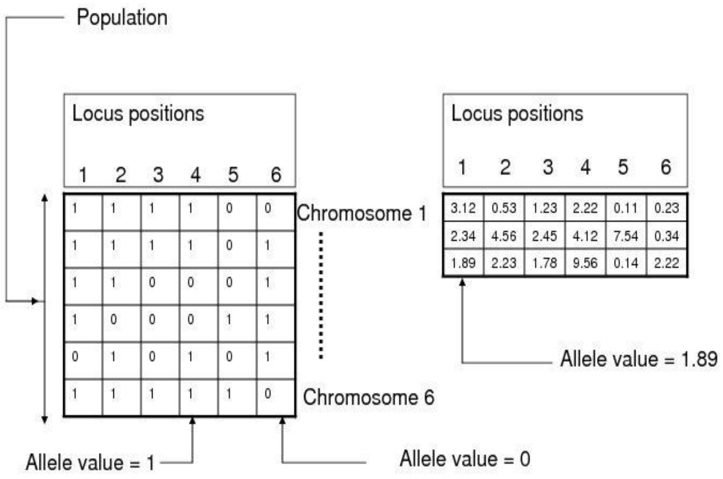
Evolutionary Algorithm

This section explains the basic structure of Evolutionary Algorithms, terms and concepts that will be used in the next chapters for the explanation of genetic algorithm and evolution strategy. In all EAs, the initial population represents the potential solutions of the problem. Hence, each individual is an encoded solution to a particular problem. Normally, an individual is represented by a string of chromosomes that is based on the principle of biologically genotypes. A genotype is composed of one or more than one chromosomes. However, each chromosome has its own set of genes, which are placed in a chromosome at a particular position called the locus, as shown in the Fig. D.7. The value held by a particular locus is called an allele that represents a genetic characteristic. A generalized representation of EAs is shown in the Fig. D.7 that represents both binary encoding and real-valued population of chromosomes.

EAs are based on the principle of natural evolution and evolutionary operators operate on the EAs population that is made up of chromosomes for generating a better solution with higher and higher fitness value. A unique characteristics of EAs that makes them a better choice for an optimization is due to three major evolutionary operators, such as mutation, crossover, and selection.

To further explain the concept behind the EAs, a step-by-step graphical representation is shown in Fig. D.8.

In the chain of operations, the initial population is randomly generated. Each individual in the population represents a potential solution. The basic idea is to represent each individual in an array of strings called *chromosomes*. Each *chromosome* is a combination of *genes*, and the position of a *gene* in a *chromosome*



1	2	3	4	5	6
1	1	1	1	0	0
1	1	1	1	0	1
1	1	0	0	0	1
1	0	0	0	1	1
0	1	0	1	0	1
1	1	1	1	1	0

1	2	3	4	5	6
3.12	0.53	1.23	2.22	0.11	0.23
2.34	4.56	2.45	4.12	7.54	0.34
1.89	2.23	1.78	9.56	0.14	2.22

Figure D.7: Generalized structure of evolutionary algorithms

is called a *locus*[109]. The initial population based on individuals is evolved according to processes based on natural evolution, such as selection, crossover, and mutation. In the selection phase, we select an individual that has superior fitness for reproduction. In the crossover, genetic information is combined from the parents and produces new offspring. In the process of evolution, one needs to perturb the genes to make them more adaptive to their environment. Mutation operators are used to alter some of the characteristic of the genes of a chromosomes. Mutation can be done for all genes or for any particular genes of an individual and makes the individual more robust, with a higher fitness value. In natural evolution, the individual that has the highest fitness value will transmit its genes to the next generation. In this way, individuals that do not have potential solutions will die or be removed from the population. Hence, this bio-inspired approach makes the EAs more suitable for optimization

To simulate this bio-inspired concept of natural evolution, we try to explain each evolutionary operator for its contribution for the success of EAs to solve real world problems. Initially, we selected a random population of chromosomes: a representation of potential solutions (i.e., individuals) to the problem. To generate

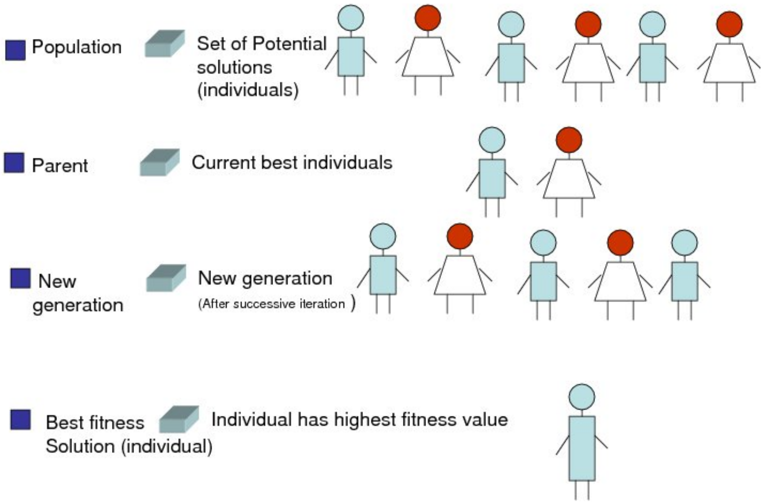

Population
Set of Potential solutions (individuals)
Parent
Current best individuals
New generation
New generation
(After successive iteration)
Best fitness Solution (individual)
Individual has highest fitness value

Figure D.8: Step-by-step EAs

a next generation of individuals, we need to select above-average individuals and must therefore select those individuals that have fitness values above the average of the whole population. To make the next generation more robust and able to cope with the environmental stress, we also need to reject below average individuals. Then, to get the next generation, we employ a crossover operator that will carry on the genetic characteristics of both parents. New offspring may have better fitness values than their successors. If the offspring have higher fitness values then the parents are kept for the next generation; otherwise they might be rejected depending on the selection criteria. The concept of crossover is explained in Fig.D.9.

To make the next generation more robust and diverse, we employ mutation evolutionary operators. For simplicity, we present a simple bitwise mutation on the individuals that will change the chromosome allele at a particular locus. One bit mutation is shown in Fig. D.10.

Due to these evolutionary operators, EAs are well suited for multi-modal and high dimensional or N-complete optimization problems. In order to have a comparison with deterministic algorithms, they are handicapped by their requirements for problem domain knowledge to direct or limit the search space in an exceptionally

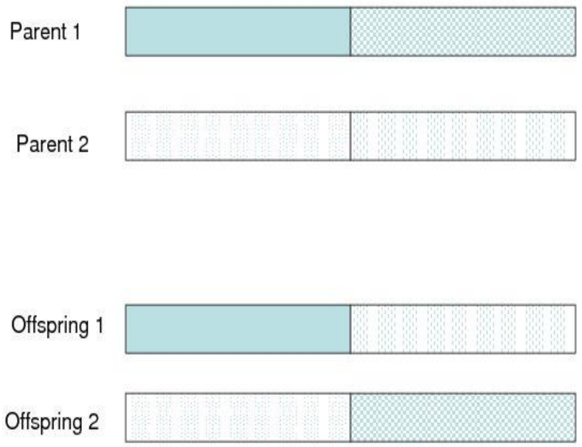


Figure D.9: Single-point crossover

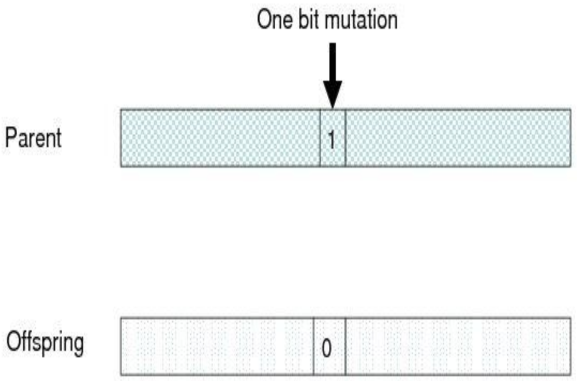


Figure D.10: Single-point mutation

large search space. However, EAs are more suitable for high dimensional, discontinuous, multi-modal, and N-complete optimization problems. We thus selected genetic algorithms and evolution strategy to optimize the cluster centroids and cluster numbers for k-means clustering algorithms and keep prevent them being trapped by local minima.

Appendix E

Evolutionary-based Approaches for IVR Systems

Evolutionary algorithms have the advantage of bio-inspired operators that make them superior to other optimization techniques. Evolutionary algorithms generally operate on a set of potential solutions called an initial population that have evolved based on survival of the fittest to produce better and better individuals that ultimately come up with more accurate solutions. In each generation, the individuals that have better approximation will actively take part in the process of breeding the next generation. Hence, EAs are modeled on natural processes that include selection, recombination, mutation, and improving behavior by sharing local and global information. These evolutionary operators fulfil the purpose of creating new, and redistributing existing, gene information.

In the following sections, we discuss evolutionary strategy, genetic algorithms and their impact on the performance of accent-based IVR systems.

E.1 Genetic Algorithm

The idea of genetic algorithms is based on the concept of natural selection and natural genetics and is derived from the evolution in nature. The universe is full of millions of species with different behaviors and characteristics. All these plants, animals, and other creatures have evolved, and continue evolving, over millions of years adapting to their specific environments to survive. Individuals that are robust enough will survive, and weaker ones tend to die out. These individuals tend to generate a next generation that is more suitable to environmental changes and which has better chances to survive. This process is dictated by the laws of natural selection and evolution. Thus, genetic algorithms exploit the idea of survival of the fittest and interbreeding to generate a more diverse and robust next generation.

E.1.1 Genetic Algorithm: In a Natural Perspective

Genetic algorithms are inspired by biological DNA evolution procedures. The DNA structure is a combination of chromosomes that are made up of sequences of genes chained together in chromosomes. In the human body (as in other living organisms) are rod-like structures called chromosomes. These chromosomes dictate the breeding characteristics of an individual (i.e., color of eyes, hair, and body structure). The actual value of a gene is encoded into an allele; the genes are encoded into alleles, and sequences of genes are grouped together in chromosomes. This grouping results in a DNA structure. In the natural evolution process, when two individuals mate, both parents pass their chromosomes into the next generations (offspring). During this process of evolution, the two chromosomes exchange genetic characteristics and form a new chromosomes with embedded characteristics of both parent's chromosome. Hence, the chromosomes undergo a crossover of genetic material that leads to a new individual. However, if the crossover of genetic material is not sufficient to produce a new good combination, then the genetic material undergos mutation to introduce more diversity in the genetic makeup of the resulting chromosome. Genetic algorithms are modeled on this process of evolution.

Genetic algorithms use string structures that are analogous to chromosomes. The idea of genetic characteristics is as mapped bits in a string and, hence, the values stored into these bits are based on an allele's formation. The combination of these values gives a chromosomes that undergos an evaluation process and is rated as per fitness values. For the next generation, the strings are selected based on their fitness value; the strings that have the highest fitness values are selected for the next generation and weaker ones die away. When strings are selected for a next generation, then the natural evolution operators work to make the next generation more robust. The whole process is depicted in Fig. E.1.

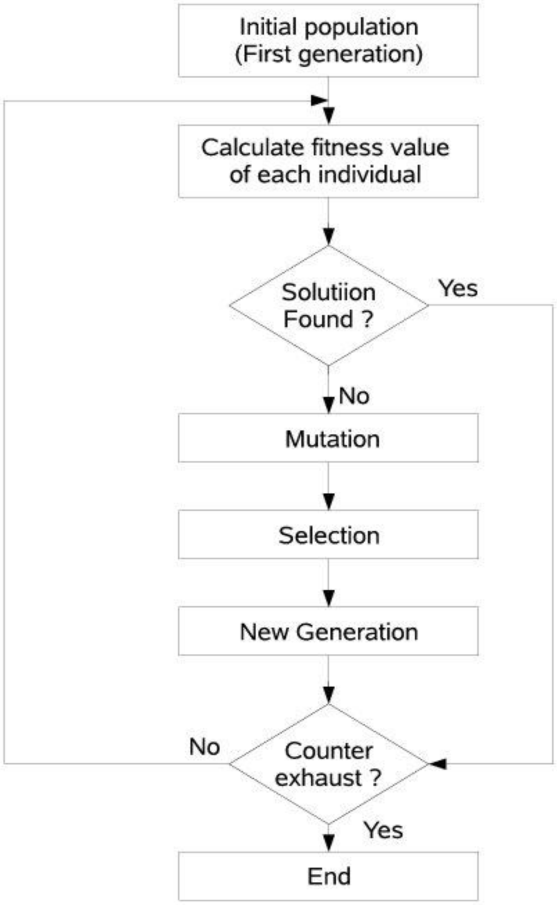


Figure E.1: Simple genetic algorithm

Our main objective in using the genetic algorithms is to improve the performance of interactive voice response system. Genetic algorithms are an attractive candidates for helping us achieve a globally optimum solution for clustering centroids. The reason for this whole research is to improve the performance of k-means clustering. We also employed the genetic algorithms as search algorithms to obtain globally optimal Gaussian components to improve the overall performance of the IVR system. We used genetic algorithms for the optimization of cluster centroids as well as the optimum number of clusters for a data set in hand. Most of the work in the literature is for using genetic algorithms to optimize cluster centroids. This optimization technique is called single objective optimization.

To improve the k-means clustering algorithm, we provide a NSGA for multiobjective optimization. To improve the performance of accent classification, we employed non-dominated sorting evolution strategy-based k-means clustering algorithm. The proposed algorithm gives better evolutionary operators than NSGA. We would like to discuss the NSGA and other methods in which genetic algorithms are used for multiobjective optimization.

E.2 Multiobjective Genetic Algorithm Techniques

In this section, we discuss the development of multiobjective genetic algorithms and associated techniques that help to improve the performance of genetic algorithm, such as different variations in the evolutionary operators. Genetic algorithms are considered to be metaheuristic problem solvers for feasible solutions in difficult search spaces. Famous variations of genetic algorithms include

- Vector Evaluated Genetic Algorithm
- Lexicographic Ordering Genetic Algorithm
- Weight-based Genetic Algorithm
- Multiple Objective Genetic Algorithm
- Niched Pareto Genetic Algorithm
- Niched Pareto Genetic Algorithm 2
- Non-dominated Sorting Genetic Algorithm

- Non-dominated Sorting Genetic Algorithm II
- Distance-based Pareto Genetic Algorithm
- Thermodynamical Genetic Algorithm

Before going into detail about the above-mentioned algorithms, it is necessary to give the origin of evolutionary algorithms for multiobjective optimization. The first multiobjective evolutionary algorithm was introduced in the mid-1980s by David Schaffer. However, there was a previous contribution by Ito *et al.* [129], in which a genetic algorithm was used to solve multiobjective optimization problems. At the same conference in which Schaffer's works was presented, Fourman [130] also presented an application of a multiobjective evolutionary algorithm.

The following section presents a generic multiobjective evolutionary algorithm that is base for all state-of-the-art multiobjective evolutionary algorithms.

E.2.1 A Generic Multiobjective Evolutionary Algorithm

As noted, an evolutionary algorithm is based on the principles of natural evolution. An effective multiobjective algorithm should incorporate the following steps:

- initialize the population, P, of N individuals, an encoding of the problem domain either as a binary, or a real value. Thus, this initial population is a set of potential solutions for a a problem in hand. Then assign a fitness value to each individual.
- Remove Pareto dominated individuals and assign Pareto ranking, as in $P- > P^i$.
- Keep the population at a reasonable computational number by using niching, sharing, and crowding distance techniques.

- Perform evolutionary operators to generate the next generation of population P^{ii}. For this next generation of individuals, employ different selection techniques, such as ranking, tournament selection etc. for recombination.

- After recombination, select individuals for next generation P^{iii} based on elitism. Elitism seems to be a good approach because individuals with better fitness values result in a better next generation.

- Remove pareto dominated individuals from P^{iii} population

- Keep non-dominated individuals by sorting P^{iii} in an archive of P^{iv}. Then merge the current population P^{iii} and P^{iv} . Hence, the P^{iv} archive now contains current populations, and a Pareto front due to this current population. The concept of Pareto front and non-dominated individuals is shown in Fig.E.2.

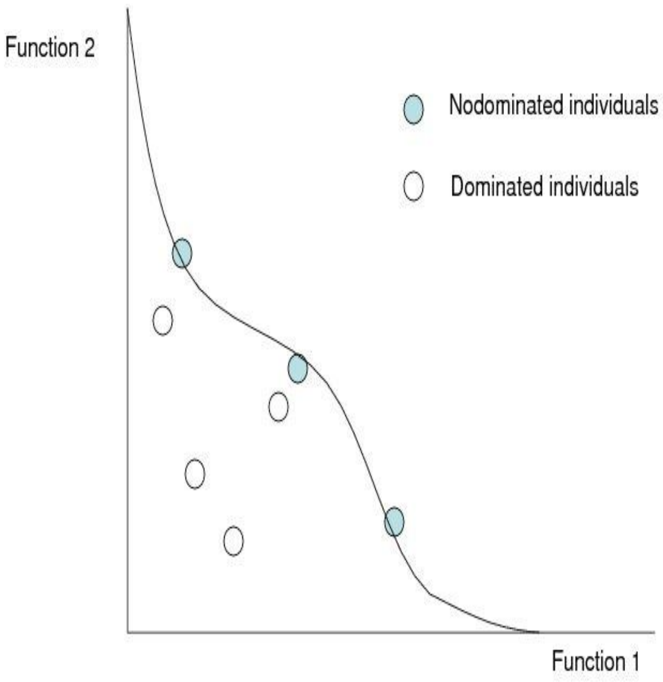


Figure E.2: Another simple genetic algorithm

To further elaborate the concept of multiobjective genetic algorithm, a generic pseudo code in presented in Algorithm 4.

Algorithm E.1 Generic multiobjective evolutionary algorithm

```
Population initialization
Evaluate objective function values over population
Assign rank based on pareto dominance
Compute Niche count
Assign shared fitness or crowding
While not reached a termination condition
Select good individual from $P- > P^i$
Apply evolutionary operators: recombination and mutation on current population
Generate next population $P^{ii}$
Evaluate fitness value of offsprings $P^{ii}$
Rank the individuals based on pareto dominanace (i.e., $P^i union P^{ii} - > P^{iii}$)
Compute niche count
Assiagn crowding or share fitness measure to the individuals
Limit $P^{iii} - > P$
Copy current population $P^{iii}$ to $P^{iv}$ based on pareto dominance.
End While
```

Genetic Algorithm for Multiobjective Optimization

E.2.2 Vector Evaluated Genetic Algorithm

A Vector Evaluated Genetic Algorithm was proposed by Schaffer in 1985 and is considered the first implementation of a genetic algorithm for multiobjective optimization problems. In this approach a vector of k objective functions is selected to solve a multiobjective optimization problem, and the selection criterion is based on a vector-valued fitness measure. The main principle in this approach is a proportional selection of the population of each objective function. Basically, it is an extension of the simple genetic algorithm. Generally, for a problem of k objectives, a k sub-population of size M/q is generated, where M is the total initial population. Then, all these sub-populations are shuffled to obtain a new population of size M, in order to further employ the evolutionary operators, such as, crossover and mutation.

Schaffer's work uses proportional fitness measures that are, in turn, proportional to the objective function itself. Thus, the VEGA is not able to generate concave portions of the Pareto front. However, VEGA is also not able to handle constraints, where it's biased is not a suitable choice to solve the multiobjective problems [131].

Another technique introduced in the literature to improve the selection procedure is the Goldberg technique, called multiobjective genetic algorithm.

E.2.3 Multiobjective Genetic Algorithm

This algorithm is a variation of Goldberg's algorithm of multiobjective genetic algorithm and was proposed by Carlos M. Fonseca and Peter J. Fleming [132]. The main idea in this algorithm is the ranking of an individual based on the number of individuals by which it is dominated, as shown in the Fig. E.3. The ranking mechanism of the dominated individuals x_i at a particular generation t is given as follows:

$rank(x_i, t) = 1 + p_i^{(t)}$ where x_i is the individual dominated by the individual $p_i^{(t)}$. The pseudo code of MOGA is presented in Algorithm 4.

Thus, this algorithm is a variation of Goldberg's multiobjective genetic algorithm regarding fitness assignment. The main difference is in the ranking mechanism for dominated individuals. For ranking the dominated individuals, first sort the individuals based on the objective value of each individual. Then assign the rank to each individual starting from the best rank (rank 1) to the worst rank ($rank\ n \leq P_{total}$). During the process of assigning rank to individuals, if more than

Algorithm E.2 MOGA

```
1: Set counter t = 0;
2: Generate initial population $P$
3: Evaluate the initial population
4: Assign ranking of individuals based on Pareto dominance
5: Compute niche count
6: Assign linearly scaled fitness
7: Share the fitness
8: while (t or solution found)
9: t = t + 1
10: Selection of the fittest individuals via stochastic universal sampling
11: Single point crossover
12: Mutation
13: Evaluate new generated population
14: Assign ranking to individuals based on Pareto dominance
15: count niche count
16: Assign linearly scaled fitness
17: Share the fitness
18: Go to step 8 until a satisfactory solution is achieved or the computation is
    exhausted.
```

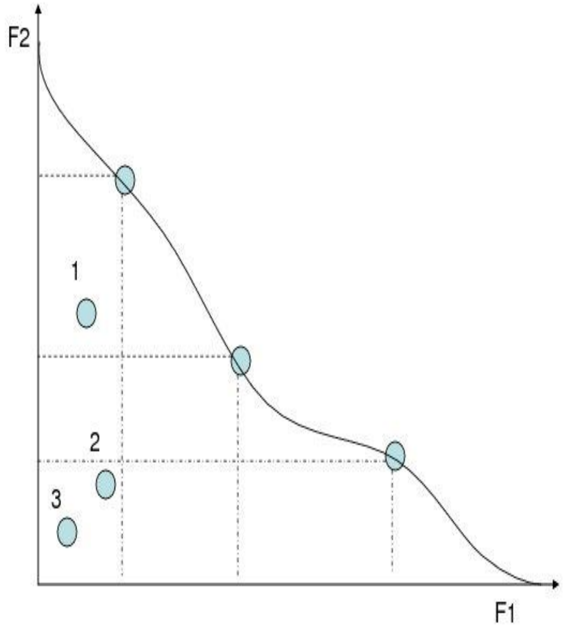


Figure E.3: Multiobjective genetic algorithm

one individual has the same rank, then average the fitness of each individual and sample at the same rate. This sampling technique allows an appropriate selective pressure and maintain the global population fitness constant.

In [133], Goldberg and Deb discuss different selection techniques for genetic algorithms and find that this the ranking selection approach is likely to produce a large pressure on the population in a specific direction that might force the algorithm to produce premature convergence. To avoid this selective pressure due to block fitness assignment, Fonseca and Fleming [132] introduced a sharing mechanism involving objective values to distribute the population over the Pareto optimal region.

References

[1] C. S. Yang and H. Kasuya. Speaker individualities of vocal tract shapes of japanese vowels measured by magnetic resonance images. In *Proceedings of the Fourth International Conference on Spoken Languages*, volume 2, pages 949 – 952, 1996. 1

[2] R. J. Moreno, B. Raj, E. Gouvea, and R.M. Stern. Multivariate-gaussian-based cepstral normalization for robust speech recognition. In *IEEE International Conference on Acoustic, Speech and Signal Srocessing*, volume 1, pages 137 – 140, 1995. 1

[3] A. J. Rubio Ayuso and J. M. Lopez Soler. *Speech recognition and coding new advances and trends.* Springer Press, 1995. 1, 2, 19

[4] C. Mokbel, L. Mauuary, D. Jouvet, J. Monne, C. Sorin, J. Simonin, and K. Bartkova. Towards improving asr robustness for psn and gsm telephone applications. In *Proceedings of the Third IEEE Workshop on Interactive Voice Technology for Telecommunications Applications*, pages 73 – 76, 1996. 1

[5] J. C. Junqua and J. P. Haton. *Robustness in automatic speech recognition: fundamentals and applications.* Kluwer Academic, 1993. 1

[6] S. Gorony. *Robust adaptation to non-native accents in automatic speech recognition.* Springer Press, 2002. 1

[7] X. Huang, A. Acero, and H. Hon. *Spoken language processing: a guide to theory, algorithm, and development.* Prentice Hall Press, 2001. 1, 11, 18, 20

[8] C. Becchetti and L. P. Ricotti. *Speech recognition: theory and C++ implementation*. John Weley and Sons, 1999. 1

[9] F. J. Owens. *Signal processing of speech*. McGraw-Hill, Inc., 1993. 1, 11, 16, 32

[10] Dave Burke. *Speech processing for IP networks*. John Weley, 2007. 2

[11] J. Gao, H. F. Wang, Li, and Lee. A unified approach to statistical language modeling for chinese. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1703 – 1706, 2000. 3

[12] C. Chelba. Portability of syntactic structure for language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 544a – 544d, 2001. 3

[13] D. R. R. Smith and R. D. Patterson. The interaction of glottal-pulse rate and vocal-tract length in judgements of speaker size, sex, and age. *Journal of the Acoustical Society of America*, 118(5):3177 – 3186, 2005. 4

[14] L. R. Rabinar. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceeding of the IEEE*, volume 77, pages 257 – 286, 1989. 4

[15] A. Cohen. Hidden markov models in biomedical signal processing. In *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 3, pages 1145 – 1150, 1998. 4

[16] S. J. Young, P. C. Woodland, and W. J. Byrne. Spontaneous speech recognition for the credit card corpus using the htk toolkit. *IEEE Transactions on Speech and Audio Processing*, 2(4):328 – 339, 1994. 4

[17] S. Ullah, F. Karray, A. Abghari, and S. Podder. Soft computing-based approach for natural language call routing systems. In *IEEE International*

Symposium on Signal Processing and its Applications, pages on 1 – 4, 2007. 5

[18] Q. Yan, S. Vaseghi, D. Rentzos, C-H. Ho, and E. Turajlic. Analysis of acoustic correlates of british, australian and american accents. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 345 – 350, 2003. 6

[19] L. W. Kat and P. Fung. Fast accent identification and accented speech recognition. In *Proceedings of the IEEE Internation Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 221 – 224, 1999. 6, 26

[20] S. Deshpande, S. Chikkerur, and V. Govindaraju. Accent classification in speech. In *the Fourth IEEE Workshop on Automatic Identification Advanced Technologies*, pages 139 – 224, 2005. 6

[21] J. H. L. Hansen and L. M. Arslan. Foreign accent classification using source generator based prosodic features. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 836 – 839, 1995. 6, 26

[22] Pongtep Angkititrakul and John H. L. Hansen. Advances in phone-based modeling for automatic accent classification. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 14(2):634 – 646, 2005. 6

[23] Q. Yan and S. Vaseghi. Analysis, modelling and synthesis of formants of british, american and australian accents. In *the IEEE International Conference on Acoustic, Speech, and Signal Processing*, volume 1, pages I–712 – I715, 2003. 6, 22

[24] H. Hermansky and N. Morgan. Rasta processing of speech. *IEEE Transactions on Speech and Audio Processing*, 2(4):578 – 589, 1994. 13, 22

[25] A. Waibel. *Prosody and speech recognition.* Morgan Kaufmann Inc. Press, 1988. 19, 20

[26] C. H Wu, Y. J. Chen, and G. L. Yan. Integration of phonetic and prosodic information for robust utterance verification. In *Proceedings of the IEEE Vison, Images and Signal Processing*, volume 147, pages 55 – 61, 2000. 20

[27] R. DeMori, M. Gilloux, G. Mercier, M. Simon, C. Tarridec, J. Vaissiere, D. Gillet, and M. Gerard. Integration of acoustic, phonetic, prosodic and lexical knowledge in an expert system for speech understanding. In *the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 9, pages 320 – 323, 1984. 20

[28] Y. Obuchi and N. Sato. Language identification using phonetic and prosodic hmms with feature normalization. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 569 – 572, 2005. 20

[29] B. Gajic and K. K. Paliwal. Robust feature extraction using subband spectral centroid histogram. In *Proceedings of the IEEE Acoustics, Speech, and Signal Processing*, volume 1, pages 85 – 88, 2001. 20

[30] L. M. Arslan and J. H. L. Hansen. Frequency characteristics of foreign accented speech. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 1123–1126, 1997. 22

[31] H. Fujisaki, M. Ljungqvist, and H. Murata. Analysis and modeling of word accent and sentence intonation in swedish. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 211 – 214, 1993. 22

[32] I. S. Burnett and J. J. Pary. On the effect of accent and language on low rate speech coders. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 291–294, 1996. 22

[33] E. J. Yannakoudakis and P. J. Hutton. *Speech synthesis and recognition systems.* John Weley and Sons, 1987. 22, 23

[34] C. M. Bishop. *Neural networks for pattern recognition.* Oxford University Press, 1995. 25

[35] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and L. Lang. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(3):328 – 339, 1989. 25

[36] J. B. Hampshire and A. H. Waibel. A novel objective function for improved phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 1(2):216 – 228, 1990. 25

[37] L. Rabiner and B. Juang. An introduction to hidden markov models. *IEEE Acoustics, Speech, and Signal Processing Magzine*, 1:4–16, 1986. 25

[38] C. Teixeira, I. Trancoso, and A. Serralheiro. Accent identification. In *Proceedings of the IEEE International Conference on Spoken Language*, volume 3, pages 1784 – 1787, 1996. 26

[39] K. Berkling, M. Zissman, J. Vonweller, and C. Cleirigh. Improving accent identification through knowledge of english syllables structure. In *the Fifth IEEE International Conference on Spkoen Language Processing*, 1998. 26

[40] P. Angkititrakul and J. H. L. Hansen. Stochastic trajectory model analysis for accent classification. In *International Conference on Spoken Language Processing*, volume 1, pages 493 – 496, 2002. 26

[41] K. Kumpf and R. W. King. Automatic accent classification of foreign accented australian english speech. In *Proceedings of the Fourth IEEE International Conference on Spoken Language*, volume 3, pages 1740 – 1743, 1996. 26

[42] J. J. Humphriest, P. C. Woodland, and D. Pearce. Using accent-specific pronunciation modeling for robust speech recognition. In *Proceedings of the Fourth IEEE International Conference on Spoken Language*, volume 4, pages 2324 – 2327, 1996. 27

[43] S. Goronzy and K. Eisele. Automatic pronunciation modeling for multiple non-native accents. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 123 – 128, 2003. 27

[44] T. Yoshimura, S. Hayamizu, and K. Tanaka. Word accent patterns modeling by concatenation of mora hidden markov models. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–69 – I–72, 1994. 27

[45] L. M. Arslan and J. H. Hansen. Improved hmm training and scoring strategies with application to accent classification. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 589 – 592, 1996. 27

[46] X. Wang, Y. Cao, F. Ding, and Y. Tang. An embedded multilingual speech recognition system for mandarin, cantonese, and english. In *Proceedings of the IEEE International Conference on Natural Language Processing and Knowledge Engineering*, pages 758 – 764, 2003. 27

[47] J. Yang, Y. Pu, H. Wei, and Z. Zhao. Acoustic model adaptation in large vocabulary continuous mandarin speech recognition for non-native speakers. In *Proceedings of the 7th IEEE International Conference on Signal Processing*, volume 1, pages 687 – 690, 2004. 27

[48] D. Reynold and C. Rose. Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing*, 3(1):72–83, 1995. 28

[49] L. Rabiner and B. Juang. *Fundamentals of speech recognition.* Prentice Hall, 1993. 28

[50] T. Chen, C. Huang, E. Chang, and J. Wang. Automatic accent identification using gaussian mixture models. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 343 – 346, 2001. 28, 29

[51] X. Lin and S. Simske. phoneme-less hierarchical accent classification. In *Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 1801 – 1804, 2004. 28, 29

[52] P. J. Ghesquiere and D. V. Compernolle. Flemish accent identification based on formants and duration features. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–749 – I–752, 2002. 29

[53] L. Yi and P. Fung. Partial change accent models for accented mandarin speech recognition. In *IEEE International Conference on Signals, Systems and Computers*, pages 111 – 116, 2003. 29

[54] M. V. Chan, X. Feng, J. A. Heinen, and R. J. Niederjohn. Classification of speech accents with neural networks. In *IEEE International Conference on Computational Intelligence*, volume 7, pages 4483 – 4486, 1994. 30

[55] R. O. Duda, P. E. Hart, , and D. G. Stork. *Pattern classification.* Wiley Interscience, New York, 2nd edition, 2001. 31, 118

[56] C. Pedersen and J. Diederich. Accent classification using support vector machines. In *IEEE International Conference on Computer and Information Science*, pages 444 – 449, 2007. 31

[57] H. Tang and Ali Ghorbani. Accent classification using support vector machine and hidden markov model. *Lecture notes in computer science*, pages 1 – 3, 2003. 31

[58] V. Zue and S. Seneff andJ.Glass. Speech database development at mit. In *TIMIT and beyond, Speech Communication*, 1990. 32

[59] J. B. Millar. The description of spoken language. In *Australian International Conference on Speech Science and Technology*, pages 80–90, 1992. 32

[60] R. Cole, M. Noe1, D. C. Burnett, M. Fanty, T. Lander, B. Oshika, and S. Sutton. Corpus development activities at the centre for spoken language understanding. In *ARPA Workshop in Human Language Technology*, 1994. 32

[61] Y. Zhang, M. Pijpers, R. Togneri, and M. Alder. Cdigits: A large isolated english digit database. In *Australian International Conference on Speech Science and Technology*, pages 820–825, 1994. 32

[62] T. Cox and M. Cox. *Multidimentional scaling*. Chapman and Hall, London, 1994. 34

[63] V. d. Silva J. B. Tenenbaum and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000. 34

[64] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2003. 34

[65] K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *International Journal of Computer Vision*, 70:77 – 90, 2006. 34

[66] B. Schoelkopf and A. J. Smola. *Learning with kernels*. Cambridge, MA: The MIT Press, 2002. 35

[67] M. Jordan E. Xing, A. Ng and S. Russell. Distance metric learning, with application to clustering with side-information. *Advances in Neural Information Processing Systems*, 2003. 35, 45, 49

[68] D. Wilkinson A. Ghodsi and F. Southey. Improving embeddings by flexible exploitation of side information. In *The 20th International Joint Conference on Artificial Intellegenc*, 2006. 35, 45, 49

[69] S. Ullah and F. Karray. Hybrid feature selection approach for natural language call routing systems. In *IEEE International Conference on Emerging Technologies*, pages 1 – 5, 2007. 38

[70] J. C. Wells. *Accent of English*. Cambridge university Press, 1982. 41

[71] E. Navas, I. Hernaez, and J. M. Sanchez. Subjective evaluation of synthetic intonation. In *IEEE Workshop on Speech Synthesis*, 2002. 41

[72] T. Ifukube, T. Hokimoto, J. Matsushima, and Y. Nejime. A digital hearing aid having a function of intonation emphasis. In *Proceedings of the IEEE International Conference on Engineering in Medicine and Biology Society*, pages 23 – 26, 1995. 41

[73] R. Kongkachandra, S. Pansang, T. Sripramong, and C. Kimpan. Thai intonation analysis in harmonic-frequency domain. In *IEEE Asia-Pacific International Conference on Circuits and Sysytems*, pages 156 – 168, 1998. 41

[74] L. M. Arslan. Foreign accent classification in american english. In *Ph.D Thesis, Duke University*, 1996. 41

[75] J. C. Wells. *Accent of English*. Cambridge University press, 1982. 42

[76] Jianfeng Gao, Hai-Feng Wang, Mingjing Li, and Kai-Fu Lee. A unified approach to statistical language modeling for chinese. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1703–1706, 2000. 45

[77] J. Goldberger, S. Roweis, G. Hinton, , and R. Salakhutdinov. Neighbourhood component analysis. In *IEEE International Conference on NIPS*, 2005. 47

[78] K. Weinberger, J. Blitzer, , and L. Saul. Distance metric learing for large margin nearest neighbour classification. In *IEEE International Conference on Neural Information Processing Systems*. 47

[79] B. Alipanahi, M. Biggs, and A. Ghodsi. Distance metric learning vs. fisher discriminant analysis. In *Twenty-Third National Conference on Artificial Intelligence*, 2008. 49, 51

[80] M. Laszlo and S. Mukherjee. A genetic algorithm using hyper-quadtrees for low-dimentional k-means clstering. *IEEE Transations on Pattern Analysis and Machine Intelligence*, 28(4):533–543, 2006. 53, 120

[81] F.-X. Wu. A genetic weighted k-means algorithm for clustering gene expression data. In *IEEE Second International Multisymposium on Computer and Computational Sciences*, pages 68 – 75, 2007. 53, 120

[82] S.Z. Selim and M. A. Ismail. K-means type algorithms: A generalized convergence theroem and characterization of local optimility. *IEEE Transations on Pattern Analysis and Machine Intelligence*, 6:81 – 87, 1984. 53, 120

[83] S.-S. Cheng, Y.-H. Chao, H.-M. Wang, and H.-C. Fu. A prototypes-embedded genetic k-means algorithm. In *In Proceedings of 18th International Conference on Pattern Recognition*, pages 724 – 727, 2006. 53

[84] C.-Y. Lee and E. K. Antonsson. Dynamic partitional clustering using evolution strategies. In *In Proceedings of 3rd Asia-Pacific Conference on Simulated Evolution and Learning*, volume 4, pages 2716 – 2721, 2000. 53

[85] U. Maulik and S. Bandyopadhyay. Genetic algorithm-based clustering technique. In *IEEE Conference of Pattern Recognition*, volume 33, pages 1455–1465, 2000. 53

[86] K. Krishna and M. N. Murty. Genetic k-menas algorithm. *IEEE Transations on Systems, Man, and Cybernetics*, 29(3):433 – 439, 1999. 54, 69

[87] P. Franti, J. Kivijarvi, T. Kaukoranta, , and O. Nevalainen. Genetic algorithms for large scale clustering problems. *The Computer Journal*, 40(9):547 – 554, 1997. 54

[88] Y. Lu, S. Lu, F. Fotouhi, Y. Deng, and S. J. Brown. Fgka: A fast genetic k-menas clustering algorithm. In *IEEE International Conference on*, 2004. 54

[89] K. Deb, A. Pratap, S. Agarwal, , and T. Meyarivan. Fast elitism multi-objective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002. 57

[90] H. P. Schwefel. *Numerical Optimization for Computer Models*. John Willey, 1981. 68

[91] Q. Huo and C. H. Lee. On-line adaptive learning of the continuous density hidden markov model based on approximate recursive bayes estimate. *IEEE Transactions on Speech and Audio Processing*, 5(2):161–172, 1997. 76

[92] X. Huang and K. F. Lee. On speaker-independent, speaker-dependent, and speaker-adaptive speech recognition. *IEEE Transactions on Speech and Audio Processing*, 1(2):877–880, 1993. 76

[93] B. Milner and S. Semnani. Robust speech recognition over ip networks. In *Proceedings of the IEEE International Conference on Acoustic, Speech, and Signal Processing*, volume 2, pages 403 – 406, 1993. 76

[94] H. Fenglei and W. Bingxi. Text-independent speaker verification using speaker clustering and support vector machines. In *Proceedings of the IEEE International Conference on Signal Processing*, volume 1, pages 456 – 459, 2000. 76

[95] Y. Zhong-Xuan, Y. Chong-Zhi, and F. Yuan. Text independent speaker identification using fuzzy mathematical algorithm. In *Proceedings of the IEEE International Conference on Acoustic, Speech, and Signal Processing*, volume 2, pages 403 – 406, 1993. 76

[96] H. A. Murthy, F. Beaufays, L. P. Heck, and M. Weintraub. Robust text-independent speaker identification over telephone channels. *IEEE Transactions on Speech and Audio Processing*, 7(5):554–568, 1999. 76

[97] J. C. Bezdek. *Pattren recognition with fuzzy objective function algorithms.* Plenum Press New York and London Press, 1981. 78

[98] J. Sun, F. Karray, O. Basir, and M. Kamel. Natural language understanding through fuzzy logic inference and its application to speech recognition. In *IEEE International Conference on Fuzzy Systems*, 2002. 79

[99] B. Chen and L. L. Hoberock. A fuzzy neural network architecture for fuzzy control and classification. In *IEEE International Conference on Neural Networks*, 1996. 79

[100] H. Hotelling. *Relations between two sets of variates.* Biometrika Press, 1936. 79

[101] K. Choukri, G. Chollet, and Y. Grenier. Spectral transformations through canonical correlation analysis for speaker adptation in asr. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 11, pages 2659 – 2662, 1986. 79

[102] Z. Bai, G. Huang, and L. Yang. A radar anti-jamming technology based on canonical correlation analysis. In *IEEE International Conference on Neural Networks and Brain*, volume 1, pages 9 – 12, 2005. 79

[103] S. Ando. Image field categorization and edge/corner detection from gradient covariance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(2):179 – 190, 2000. 79

[104] C. Davatzikos, D. Shen, A. Mohamed, and S. K. Kyriacou. A framework for predictive modeling of anatomical deformations. *IEEE Transactions on Medical Imaging*, 20(8):836–843, 2001. 79

[105] Z. Gou and C. Fyfe. A robust canonical correlation neural network. In *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*, pages 239–248, 2002. 79

[106] T. W. Anderson. *An introduction to multivariate statistical analysis.* John Weley, 1984. 87

[107] A. Yamakawa, H. Ichihashi, and T. Miyoshi. Fuzzy c-means with maximizing canonical correlation coefficients. In *Faji Shisutemu Shinpojiumu Koen Ronbunshu*, volume 15, pages 631 – 632, 1999. 87

[108] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue. Timit acoustic-phonetic continuous speech corpus. In *National Institute of Standards and Technology Disc 1-1.1, NTIS Order No. PB91-5050651996*, 1990. 91

[109] F. Karray and C. D. Silva. *Soft computing and intelligent system design: theory, tools and applications.* Addison Wesley Press, 2004. 95, 105, 133

[110] S. Llyod. Square quentization on pcm. *IEEE Transactions on Information Theory*, 28:129–137, 1982. 118

[111] A. Baraldi and P. Blonda. A survey of fuzzy clustering algorithms for pattern recognition. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 29(6):778–785, 1999. 118

[112] A. K. Jain and R. C. Dubes. *Algorithms for clustering.* Englewood Cliffs, NJ: Prentice-Hall, 1988. 118

[113] J. Mao and A.K. Jain. A self organizing network for hyperellipoidal clustering (hec). *IEEE Transations on Neural Networks*, 7:16–29, 1996. 119

[114] W. J. Rucklidge D. P. Huttenlocher, G. A. Klanderman. Comparing images using the hausdorff distance. *IEEE Transations on Pattern Analysis and Machine Intelligence*, 15(9):850 – 865, 1993. 119

[115] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz. *Robust face detection using Hausdorff distance*. Springer Berlin, 2001. 119

[116] M. P. Dubuisson and A. K. Jain. A modified hausdroff distance for object matching. In *IEEE International Conference of Pattern Recognition*, pages 566–568, 1994. 119

[117] F. Y. Edgeworth. *Mathematical physics*. P. Keagan, London, 1881. 123

[118] V. Pareto. *Cours D' economie politique, Vol. I and II.* F. Rouge, Lausanne, 1896. 123

[119] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen. *Evolutionary algorithms for solving multi-objective probles*. Springer, 2007. 123, 125

[120] P. van Larrhoven and E. Aarts. *Simulated annealing: theory and applications.* Kluwer Academic Publishers, Dordrecht, 1987. 123

[121] S. S. Rao. *Optimization theory and applications*. Wiley Eastern Limited, New Delhi, 1991. 124

[122] Z. Michalewicz and D. B. Fogel. *How to solve it: modern heuristics*. Springer, Berlin, 2004. 126

[123] P. Brately G. Brassard. *Algorithmics: theory and pratice.* Prentic-Hall, Englewood Cllif, New Jersey, 1988. 127

[124] P. Husbands. *Genetic algorithm in optimization and adaptation.* Halsted Press, New York, 1992. 127

[125] S. Russell and P. Norvig. *Artificial intelligence: a modern approach.* Prentice-Hall, New Jersey, 1995. 127

[126] R. Neapolitan and K. Naimopour. *Foundations of algorithms.* D. C. Health and Company, Lexington, Massachusetts, 1996. 128

[127] L. J. Fogel. *Artificial intelligence through simulated evolution.* John Wiley and Sons, Inc., New York, 1999. 130

[128] M. Garey and D. Johnson. *Computers and interactability: a guide to the theory of NP-Completeness.* Freeman, 1979. 130

[129] K. Ito, S. Akagi, and M. Nishikawa. A multiobjective optimization approach to a design problem of heat insulation for thermal distribution piping network systems. *ASME Trasanctions on General of Mechanisms, Transmissions and Automation in Design*, pages 206 – 213, 1983. 140

[130] M. P. Fourman. Compaction of sybolic layout using genetic algorithm. *First International Conference on Genetic algorithms*, pages 141 – 153, 1985. 140

[131] C. A. C. Coello and A. H. Aguirre. Design of combinational logic circuits through an evolutioinary multiobjective optimization approach. *Artificial Intellegence for Engineering, Design, Analysis, and Manufacture*, 16(1):39–53, 2002. 143

[132] C. M. Fonseca and P. J. Fleming. Genetic algorithm for multiobjective optimization: formation, discusion and generalization. In *Fifth International Conference on Genetic Algorithms*, pages 416–423, 1993. 143, 145

[133] D. E. Goldberg and K. Deb. A comparision of selection schemes using a genetic algorithm. In *Editor Foundations of Genetic algorithms*, pages 69 – 93, 1991. 145