MOHIT KUMAR

AUTOMATIC SPEAKER RECOGNITION USING LOW RESOURCES: EXPERIMENTS IN FEATURE REDUCTION AND LEARNING

AUTOMATIC SPEAKER RECOGNITION USING LOW RESOURCES: EXPERIMENTS IN FEATURE REDUCTION AND LEARNING

MOHIT KUMAR

Doctor of Philosohpy (Ph.D.)
Department of Computer Science and Engineering
Indian Institute of Technology Guwahati

August 2018

Dedicated to my family.

DECLARATION

I certify that

1. The work contained in this thesis is original, and has been done by myself under the general supervision of my supervisor.
2. The work has not been submitted to any other institute for any degree or diploma.
3. Whenever I have used materials (data, theoretical analysis, results) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.
4. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Guwahati, August 2018

Mohit Kumar

CERTIFICATE

This is to certify that this thesis entitled "**Automatic Speaker Recognition using low resources: Experiments in Feature Reduction and Learning**" being submitted by Mohit Kumar to the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, is a record of bona fide research work under my supervision and is worthy of consideration for the award of the degree of Doctor of Philosohpy (Ph.D.) of the Institute.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree or diploma.

Guwahati, August 2018

Prof. Pradip K. Das

ABSTRACT

The main objective of this thesis is to explore experiments about reduction of computations involved in the Automatic Speaker Recognition (ASR) task and about generating representations for speaker-related information from speech data automatically. ASR systems heavily depend on the features used for representation of speech information. Over the years, there has been a continuous effort to generate features that can represent speech as best as possible. This has led to the use of larger feature sets in speech and speaker recognition systems. However, with the increasing size of the feature set, it may not necessarily be true that all features are equally important for speech representation. We investigate the relevance of individual features in one of popular feature sets, MFCCs. The objective was to identify features which are more important from speech information representation perspective. Experiments were conducted for the task of speaker recognition. Both linear and non-linear dimension reduction techniques were used for reducing features. Results indicate that it is possible to reduce the feature set size by more than 60% without significant loss in accuracy, leading to improvement in the response of the recognition system due to reduction in computations.

Furthermore, it is also possible to reduce the computational resources required for ASR at the model level. Recent trends have indicated the use of very high computations for solving the problem of speaker recognition. However, there are cases when gains are not commensurate to the additional computations involved. We studied the effect of size of UBM and the total variability matrix, T, in i-vector modeling on the recognition performance. Results indicate that increasing T beyond 50 does not improve the performance significantly. For UBM size, 128 is observed as the optimal mixture count. We also investigate the effect of length of training and testing data on the performance. For performing the experiments, we have used the ALIZE toolkit and TED-LIUM database. We were able to reduce the time for extraction of i-vectors from 2 hours and 30 minutes to about 12 minutes using our reduced parameters. We then extend the experiments for a two-pass speaker recognition system where first, the broad accent category of the test utterance is determined before proceeding to speaker recognition.

We achieved significant reductions in the sizes of feature sets, sometimes more than 60% reduction in certain cases and improvements were also obtained using i-vector modeling. The scope of reduction in computation and the obtained experimental results indicate that it may possible to derive and create better features than those that

currently exist for speech representation. Hence, we investigated the issue of learning features directly from speech data, rather than using handcrafted features, for the task of automatic speaker recognition. This strategy can provide us with features that are able to capture and represent important characteristics of speech data that are needed for the speaker recognition task. From the results of the experiments, it may be concluded that learning features directly from data instead of using human expertise to derive features can be highly beneficial. Besides saving resources in terms of expertise, it is indicated by experimental results for ASR task that they outperform the traditional MFCC feature set on average. It may also be inferred that using a feature set of size 15 can be beneficial for ASR task. Further, using the GMM-UBM architecture to model the speakers using the learned features leads to good performance without incurring the heavy computations involved in deep architectures used in end-to-end recognition systems.

PUBLICATIONS

[1] A. Joshi, M. Kumar, and P. K. Das. "Speaker Diarization: A review." In: *International Conference on Signal Processing and Communication (ICSC)*. 2016, pp. 191–196.

[2] M. Kumar and P. K. Das. "Learning contextual speech features from data for Automatic Speaker Recognition." In: *IETE Technical Review* (2018). Under review.

[3] M. Kumar, D. Dutta, and P. K. Das. "Issues in i-Vector Modeling: An Analysis of Total Variability Space and UBM Size." In: *Speech and Language Processing for Human-Machine Communications*. 2018, pp. 163–172.

[4] M. Kumar, S. Katti, and P. K. Das. "Exploration of Feature Reduction of MFCC Spectral Features in Speaker Recognition." In: *Advanced Computing and Communication Technologies* (2016), p. 151.

[5] M. Sant, M. Kumar, and P. K. Das. "Accent Recognition of Speakers using I-vector Representation." In: *International Symposium on Acoustics for Engineering Applications (NSA-2016)*. 2016.

[6] S. Sharma, M. Kumar, and P. K. Das. "A technique for dimension reduction of MFCC spectral features for speech recognition." In: *International Conference on Industrial Instrumentation and Control (ICIC)*. 2015, pp. 99–104.

No one can whistle a symphony.
It takes a whole orchestra to play it.

— H. E. Luccock

ACKNOWLEDGMENTS

I believe that nothing of significant value is ever achieved alone. I have been fortunate enough to have met several people who have helped me and contributed to my success in various ways during my eventful journey at IIT Guwahati.

First and foremost, I would like to express my sincere gratitude to my PhD supervisor Prof. Pradip K. Das for his valuable guidance, inspiration and encouragement throughout the course of my research work. He has been supportive at every step right from the very beginning. I would like to thank him particularly for all of his help with the careful correction of my manuscripts. I have learned a lot from him in all aspects of life – research, professional and personal. I would also like to thank him for supporting me to visit and attend various conferences. It has been a memorable experience to work under his supervision.

It gives me immense pleasure to thank my Doctoral Committee members – Dr. Pinaki Mitra, Dr. Rashmi Dutta Baruah and Prof. Bhaba Kumar Sarma – for their insightful comments and critical questions. They provided me with a fresh perspective for my work. I thank Prof. Shivashankar B. Nair for his words of advice and encouragement. I would also like to thank the Department of CSE, IIT Guwahati for providing different facilities to carry out my research work.

I particularly thank Abhishek, Sonia, Tushar Semwal and P Bhagath for helping me out in my research work on countless occasions. I have had very stimulating discussions with all of them. I thank my senior scholars, especially Mayank Agrawal and Shashi Shekhar Jha, for their advice and encouragement during my initial months as a research scholar. I also express my gratitude to all my fellow research scholars who have helped me in ways both small and big. I thank Gaurav Pandey, Anand Nayak and Anand Kamal for lifting me up when my spirits were low. I am grateful for these friendships. I also thank Abhay Kumar Singh for having long philosophical discussions with me. He helped me discover perspectives about things and life that I could not have seen on my own.

Last but most importantly, I thank my family. They have been a source of constant support and motivation. Without them, I would not be writing these words. I especially pay sincere gratitude to my

parents for bearing with me in times of uncertainty, for their unconditional love and sacrifice.

Mohit Kumar

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 WHAT IS SPEECH?

Speech is the ability of humans (and some animals) that enables us to communicate with each other. It is based on a set of rules that ensures only the intended meaning is communicated to the recipient. Humans have a very high degree of tolerance with respect to the deterioration of the speech and can understand what is spoken and who is speaking even in a very high noise environment. From a machine's point of view, speech is a signal, a set of continuously sampled numbers. To enable a machine to understand speech and derive the message from these numbers is a challenging task for the research community. Similarly difficult is the task of determining who is the message's producer, that is, the speaker or talker. The investigation into how to determine the speaker of a speech signal is the topic of this thesis.

Speech Recognition is the process through which a spoken utterance, produced by a speaker, is understood by a listener. Understanding, in this context, means that the listener successfully recognizes the idea that the speaker wants to convey. The usefulness of the speech recognition is so obvious that we see it everywhere in our daily life. For example, searching about a product on Google app through voice is enabled by a speech recognition engine under the hood. The interaction with assistive technologies have speech recognition as one of the modalities. Amazon Alexa is a personal assistant that uses speech recognition as one of its subsystems. Similar assistants are also available for use while driving a vehicle.

Speaker Recognition, a related task, is the process of determining the identity of the speaker from an utterance. This task is not concerned with what has been spoken, only who has spoken the utterance. The difference between speech and speaker recognition systems is illustrated in figure 1.1. There are applications, including some mentioned above, that use speaker recognition, sometimes with speech recognition as well. For example, a personal assistant could have functionality to make sure that it provides search results to a query only when its owner (or someone from a list of verified or pre-specified speakers) is the one asking about the information. The access to certain areas of a premises could also be regulated using speaker identity. There are security safes that not only use passcodes but also owner authentication through voice. Several other creative

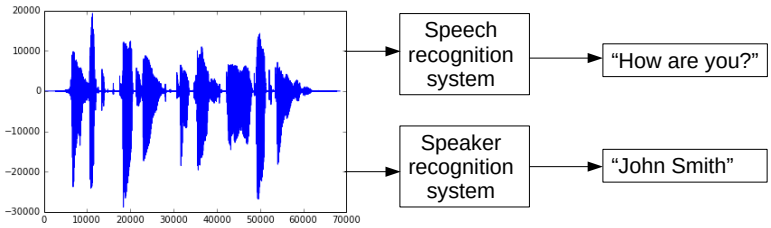

Figure 1.1: The difference between a speech recognition system and a speaker recognition system

and useful applications of speaker recognition task can also be found in other spheres of life.

1.2 MOTIVATION

Speech is a natural mode of communication for humans. The importance of this modality increases further when we consider people who are unable to access digital technology due to some unfortunate disability. It is imperative to pursue research that can enable everyone to have a quality life. Speech research is one such pursuit.

Besides, there are several specific areas that can be improved with the help of speech research, specifically speaker recognition. Such areas include access control and password reset facilities, speech data management and organisation as in diarization of meetings, transaction management and authentication through voice, forensic applications and personalization based on users. Speaker recognition directly serves to improve these application areas.

Over the years, speech and speaker recognition systems have become very complex and computationally intensive. They also need a lot of data to generalize well. While the generalization criterion is a good measure of the success of the system, there are cases where a specialized system can function well. In these cases, there are other constraints. For example, non-availability of intensive computational resources could be one. Another could be the availability of only low power and/or battery-powered devices. Here, resource could be computational power, battery, memory or internet connectivity. Some of these resources may be interdependent on each other such as computational power and battery. The mentioned constraints hold true in geographical regions that are not readily accessible from the mainland or are in deep rural pockets. A relevant example is the northeastern region of India where such resources can be hard to find.

To solve this problem, we need methods and systems that can work well with a relatively small dataset and are also fast. We can constrain these systems to work only in specific, critical applications to ensure

good performance. This kind of specific requirement of speech recognition systems goes exactly opposite to current research trends of big data and deeper neural networks. This thesis tries to address this niche area.

1.3 BRIEF LITERATURE SURVEY

Speech research has been an active area of research, especially since the 1950s. Even before people started working on recognizing sounds through machines, studies were already underway to know more about the process of hearing and understanding sounds by humans [6]. Soon the importance of automatic recognition of speech and speakers were realised and the field began to emerge out into the mainstream. Quickly, some of the theoretical aspects of automatic speech/speaker recognition and synthesis systems were formulated [19, 68] which were improved and adapted as was required in later systems.

One of the first speech recognition systems was developed using ideas (like formants determination) from acoustic-phonetics domain to recognize isolated sounds such as vowels [18]. Researchers at Bell Labs developed a simple, isolated digit recognition system. In this system, the main task was to locate spectral resonances in vowel regions of spoken digits. Phoneme recognition was the next milestone of researchers when Fry and Denes from University College in England tried to recognize four vowels and nine consonants [19]. This work used spectrum analysis and pattern matching techniques in background, along with a rudimentary language syntax in the form of allowable sequences of phonemes, to reach a recognition decision. It was discovered by researchers that using frequency analysis could be useful for recognition purposes and hence such a study was conducted with very promising results for recognizing spoken digits [14]. Similar experiments for recognizing speakers and voices also started to be conducted [64].

Earlier speaker recognition and verification approaches were focused on the acoustic-phonetic aspects of the speech and speaker characteristics. Figure 1.2 depicts a typical speech waveform along with some formants information. Many studies were conducted in the 1960s with such approaches to solve the speaker recognition problem [23, 73]. Some frequency domain based approaches were also proposed [27]. Other approaches such as cepstral measurements based [45], spectrum based [5], voice-prints based [38], etc. were also postulated for the problem. Several similarity measures for comparisons between sounds were soon devised to make efficient judgements [62].

As time progressed, technologies advanced and new concepts were formulated, more complicated approaches soon began to be devised for automatic speaker recognition and verification. Researchers proposed several fresh methods such as based on pitch contours [3],

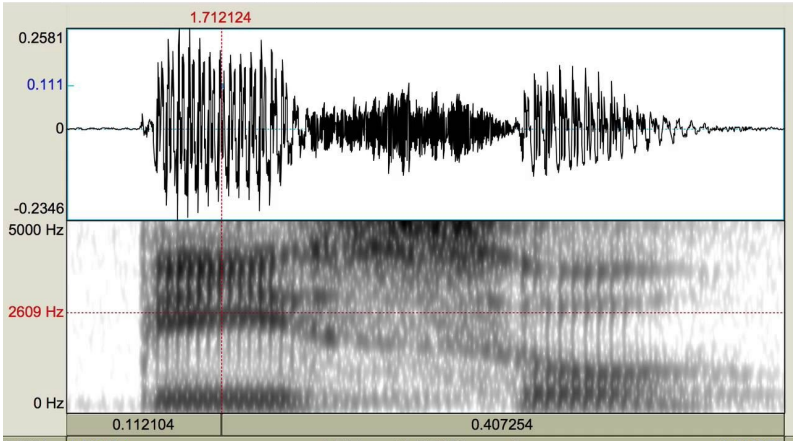

Figure 1.2: A typical speech waveform analysis window showing formants and resonances

efficient acoustic parameters [74], long term feature averaging techniques [48], etc. A lot of focus was given to developing good features at this stage of speech research. It was widely understood that spending time and resources on features can lead to a better recognition system. Hence, a lot of new features were proposed at this time. But the Mel Frequency Cepstral Coefficients (MFCC) [9], Linear Prediction Coeficients (LPC) [33] and the Perceptual Linear Prediction (PLP) [28] are recognized as most useful features that have stood the test of time. Figure 1.3 illustrates conversion of speech signal into features. Due to advancements in statistical processing, several such methods also came to the fore. Approaches using orthogonal linear prediction [60] and fuzzy sets decision making [52] were devised to improve the performance of the systems. With isolated and small vocabulary systems already in place, researcher also put their efforts in large and continuous recognition systems [47].

While some researchers were using vocal tract, pitch information [15, 51] and other acoustic parameters of speech signal to propose speaker recognition models, others started their work in areas such as features that would most describe a specific speaker [21], recognition with short utterances from the speakers [42], statistical methods such as vector quantization techniques for speaker recognition [67], etc. Finding speech segments that would enable better speaker discrimination [41], normalization techniques and comparisons of several statistical and dynamic features [20] were done to further improve the recognition techniques for speakers.

Further on, the field of speaker recognition came to be dominated by probabilistic methods such as Hidden Markov Models (HMMs) [55], Gaussian Mixture Models (GMMs) [57], Support Vector Machines (SVMs) [8] and Artficial Neural Networks (ANNs) [16]. GMMs were most successful among these methods [56]. I-vectors [11] method was

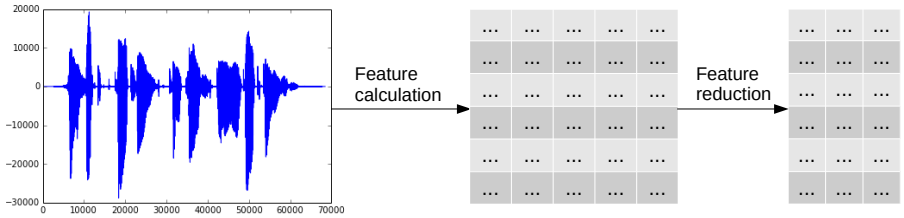

Figure 1.3: Conversion of speech signal to feature representation. Each row represents a block of signal and a column represents a dimension

then proposed based on GMMs to improve the performance also reduce the dimension of the speaker models. Currently, due to major advances in GPU technologies, deep neural networks have become state of art in speech recognition tasks [25].

1.4 RESEARCH ISSUES ADDRESSED IN THIS THESIS

The following research issues are addressed in this thesis:

1. The issue of speech representation is critical to the automatic processing of speech input. One of the main issues discussed in this thesis is how to represent the speech input most efficiently such that the recognition system performs quickly and without extensive computational load while not sacrificing on accuracy. To identify and rectify such problems, experiments for vowel and speaker recognition were performed and improvements were obtained. Based on the results of these experiments, significant reduction in the size of feature sets were obtained that lead to faster processing of speech inputs.

2. Reduction at model stage is also explored in this thesis through the method of i-vectors. The method is based on Gaussian Mixture Models and converts the high dimensional models to intermediate sized vectors. The parameters values that can optimize the model were found through speaker recognition experiments. These experiments provide the values that can lead to good recognition performance along with savings in computational time and resources. Experiments for recognizing speakers after recognizing accents are also discussed.

3. As discussed above, the issue of feature representation is very important for speech based systems. Learning of features from data automatically instead of using hand devised features for speaker recognition task was explored with promising results. When sufficient amount of data is available, it may be possible to learn the representations from the data itself. Experiments relating to feature learning were conducted and benefits of such

an approach are discussed along with their limitations and applications.

1.5 CONTRIBUTIONS OF THIS THESIS

The following details the major contributions of the thesis:

1. Experiments for reducing the dimension of features (such as LPCC, MFCC, PLP) were conducted for the tasks of vowel recognition. Following the promising results in this experiment, feature reduction experiments were also conducted for speaker recognition task. Features were reduced from 12 dimension down to 2 to observe the effects of such reduction on the performance of the recognition task. Several dimension reduction techniques such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), etc. were utilized to reduce the dimension of features.

2. I-vectors, an approach that can be seen as a way to reduce model size, was used for speaker recognition experiments. It has been used successfully to identify speakers but it still involves a high computational load. Experiments were performed in order to determine the best configuration for the method such that significant computations can be saved without sacrificing on accuracy. Based on this method, accent recognition experiments were also conducted as a form of two-step recognition process where accent is recognized first and then speaker recognition is done from a smaller set of speakers.

3. On experimenting with feature reduction approaches, it was hypothesized that there is a limit to which performance can be achieved using traditional handcrafted features for speech representation. Hence, a new representation or feature learning technique was proposed and implemented that has shown promising results for the task of speaker recognition.

1.6 ORGANISATION OF THE THESIS

This thesis is organised as follows:

1. Chapter 2 discusses the feature reduction experiments performed to reduce the feature set sizes for speaker recognition tasks. It also briefly relates the major concepts needed for the experiments, along with major works in the literature.

2. Chapter 3 deals with the reduction at the model level, by using the concept of i-vectors to represent speaker models. The experiments include determining key parameter values for the

speaker recognition system, followed by some accent recognition experiments to support speaker recognition task.

3. Chapter 4 presents the representation or feature learning experiments using Autoencoders with Recurrent Neural Networks. This chapter deals with the learning of low dimensional representation of features rather than reducing the hand-engineered features.

4. Chapter 5 summarizes the thesis and lays out the scope of further improvements in future works.

2

EXPLORING FEATURE REDUCTION IN AUTOMATIC SPEAKER RECOGNITION (ASR) EXPERIMENTS

Speech recognition, as described in the previous chapter, deals with the recognition of the content that is embedded in the speech signal. Speaker recognition deals with determining the identity of the person who generated the given speech signal. The task, therefore, is to ascertain the identity of the speaker, given a spoken utterance.

Speech signals cannot be directly compared to each other due to their time-dependent behaviour. Furthermore, two similar utterances can be spoken in such a way that they may appear different but would be semantically equivalent. For example, pronouncing the digit '7' (seven) by elongating the stress and time spent on vowels can lead to a longer utterance. On the other hand, a normally pronounced '7' would be comparatively shorter in duration. These kinds of practical issues render direct comparison useless for speech signals. Hence, we need some other representation that can help us compare different speech signals in order to determine their similarity or dissimilarity. This is achieved through what are known as features.

Features are required to represent a speech signal so that it can be processed by machines. A feature is a characteristic or property of an observed phenomenon that is individually measurable. In the domain of machine learning, it is recommended that the features be chosen in such a way that they are independent, informative as well as have high discriminating capacity. Choosing features appropriately can often be the difference between a robust system and an unreliable one. Features are often numeric in nature but can also be in other forms such as graphs and strings values, depending on the application. Features are usually represented conveniently as a vector, known as Feature vector. The selection and extraction of proper features falls under what is called as feature engineering. Automating the process of feature calculation is known as Feature learning, where not only the model but also the features themselves are learnt by the system itself.

In a lot of applications, the observed raw features are not directly useful. They may be redundant or can be too many in number to be applied for pattern recognition purposes. Therefore, a set of preprocessing techniques are usually applied to render the features more useful and relevant to the application at hand. These operations help to obtain a reduced set of features while improving the ability of the features to generalize well on unseen examples. This chapter explores the effect of some of the feature reduction and transformation

techniques applied on a feature set called Mel Frequency Cepstral Coefficients (MFCCs) for the task of automatic speaker recognition (ASR).

Mel Frequency Cepstrum (MFC) is obtained from short-term power spectrum of sounds. The MFCCs taken together make up the MFC. MFCCs are derived using a linear cosine transform of log power spectrum of a sound over the non-linear Mel scale of frequency. In MFC, the frequency bands are spaced equally on the mel scale rather than the ordinary frequency scale. This is believed to approximate the response of the human auditory system better than ordinary, linearly spaced frequency bands that are utilised in calculation of normal cepstrum. MFCC values are not considered very robust with regard to noise such as additive noise. As such, it is a useful practice to normalise the coefficient values. The power of log-mel-amplitudes can also be raised before applying DCT in order to make the features robust. Over the years, MFCCs have been considered as the most popular and useful feature set for speech representation. Some other popular features include Linear Predictive Cepstral Coefficients (LPCCs) and Perceptual Linear Prediction Cepstral Coefficients (PLPCCs).

It has also been observed that the size of feature sets has become larger over time. The goal behind the increasing size has been to represent as much information as possible with a large sized feature vector. But it may be the case that all the features in the feature set are not representing similar amount of information. Some of the features may be less important than others in conveying speech information. Therefore, it is worthwhile to determine which of the features can be more useful to the task of speaker recognition and which features are redundant or impact the recognition system negatively. Also useful is to find out if the features could be improved for speaker recognition task by applying some preprocessing methods. We explore these issues in the following experiments. The motivation behind this work is to understand the importance and utility of individual features in the feature sets. With this understanding, we can try to tailor our ASR systems such that they use fewer overall operations and can perform recognition tasks on low resource devices. One way to achieve this is to reduce the size of the features used to represent the audio sample in the system. This reduction process is the focus of this chapter. As we see in the results section, it is worthwhile to perform some preprocessing to reduce the feature vector size as we can get good performance even after significantly reducing the feature vector size. The details are presented in the Results and Discussion section of the chapter.

Several studies have been conducted exploring dimensionality reduction and speaker modeling. [31] has tried non-linear reduction techniques for HMM based phonetic recognition. [54] has used the Principal Component Analysis (PCA) reduction technique to develop

a speech emotion engine. [17] provides a lucid survey of major dimensionality reduction techniques that can be applied to various problems. [57] discussed the use of Gaussian Mixture Models to represent the speakers. This was a novel approach at the time. Further, [56] proposed modifications to existing GMM models for improving training time of the models. More advanced models were also introduced such as Joint Factor Analysis (JFA) [35], i-vector modeling [11], deep learning and neural networks based methods [26].

2.1 SPEAKER MODELING APPROACHES

2.1.1 *Gaussian Mixture Models*

Gaussian Mixture Models (GMMs) have been widely recognised as state-of-the-art method for speaker recognition and verification tasks [57]. The objective of a GMM is to find a set of finite, multi-dimensional Gaussian probability distributions that can best model the given input data. The input data, in case of speech domain, can be the features extracted from the raw signal.

More formally, a GMM is a parametric probability density function (PDF) that is represented as a weighted sum of component Gaussian densities. GMMs are employed as parametric models of the probability distribution of speech features. An example of such model can be the model for vocal-tract related spectral features (MFCCs, LPCCs, etc.) in a speaker recognition system.

A GMM model can be represented as a weighted sum of M component Gaussian densities as follows:

$$p(x|\lambda) = \sum_{i=1}^{M} w_i g(x|\mu_i, \Sigma_i) \tag{2.1}$$

where x is a D-dimensional vector, $w_i, i = 1, 2, \ldots, M$, are the mixture weights and $g(x|\mu_i, \Sigma_i), i = 1, 2, \ldots, M$, are the component densities of the Gaussian. Each individual component density is a D-variate Gaussian function as follows:

$$g(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu_i)'\Sigma_i^{-1}(x-\mu_i)\right\} \tag{2.2}$$

where μ_i is the mean vector for component i and Σ_i is the covariance matrix. The sum of all the weights must be equal to unity.

It is hypothesized that using a finite number of Gaussian components functions, we can represent a speaker's characteristics. That is, given enough Gaussian components, we should be able to capture the properties of a speaker. This characterisation needs to be such that it has sufficient discrimination capacity for different speakers in order to be successful as a speaker model.

The parameters for GMM can be estimated from training data using training algorithms such as the iterative algorithm Expectation Maximization (EM) [13] to obtain a Maximum Likelihood Estimate (MLE). Maximum A Posteriori (MAP) estimation can also be used in such cases when we have a well-trained prior model that can be adapted to the specific speaker models.

2.1.2 *Universal Background Models (UBMs)*

In the task of speaker *verification*, the log-likelihood ratio is calculated to determine if there is sufficient evidence to conclude that the given audio sample belongs to the claimed speaker. To calculate such a ratio, a 'background' model is needed against which the claimed speaker model is tested. There are many ways that this background model can be constructed. For example, a cohort of speakers can be selected for each speaker to act as background speakers for that speaker. This approach is expensive in terms of time, effort and expertise. Another approach can be to pool all the available data and estimate a 'neutral' or 'average' model and use this model as the background model for all the speakers. This is much more efficient than the former approach and does not require a huge amount of time and expertise. This model which uses all of the speakers' data is called as Universal Background Model (UBM). A UBM is, thus, a speaker independent Gaussian mixture model which is trained from a large set of speakers to represent the general speech characteristics.

It is apparent that the application of the UBM seems to be in the verification task rather than speaker identification or recognition task. However, there does exist an important application of UBMs in speaker identification task.

After obtaining the data (audio samples) for a particular speaker, we can build the speaker model by estimating the means and variances of the Gaussian components through MLE. Here we begin with a random initialization of the model and apply EM algorithm iteratively to refine that model. This approach had some issues. Due to random initialization it is possible that the time needed for the model to converge to stable values may become large. It is also possible that it may not converge to a good model unless initialized multiple times.

Another way to build speaker model is to use the UBM as the initial model and then improve on that. Since UBM may be considered to represent the general speech characteristics, it is a good idea to start with such a model as our initial model. The speaker model can, therefore, be obtained by using the UBM parameters as priors for each of the speakers. This approach to obtain speaker models by adapting the UBM model to speakers leads to Adapted GMMS. It has been verified through experiments that the adaptation approach to building

speaker models gives better models which leads to improved performance in the speaker recognition task.

2.2 FEATURE REDUCTION

As discussed above, speech signal is represented in the system through features. To make the feature less redundant, more expressive and manageable, some form of preprocessing can be applied. One of the desirable properties of preprocessing is that it can lead to huge reductions in the size of the feature vectors. This can be a significant advantage for low resource devices such as mobile phones and tablets. Another advantage is that by reducing the number of free variables, visualisation and interpretation becomes possible in many cases.

Feature reduction techniques [17] can be classified into two categories – *Feature selection* methods and *Feature re-extraction* methods.

Feature selection is the process of choosing a subset of features for building a model. We can use the individual merit of each feature in the feature vector and decide whether to retain it or not. The critical part of this approach is deciding the merit of the individual features.

It is also expected that in the presence of redundancy, we can eliminate some features without loosing much information in the process. Redundancy is said to occur in features if any two or more features are strongly correlated with each other. Feature selection can also help in model's generalization capability. Some of the feature selection methods are mentioned below.

2.2.1 *Feature Selection Methods*

During the course of this work, we investigated two methods to decide the merit of features and rank them – *Naïve dropping* method and *F-Ratio*.

2.2.1.1 *Naïve dropping*

In Naïve dropping method, we consider the order of the feature in the feature set as their rank. In case of MFCC feature vectors, the lower order coefficients, representing the lower bands of audio frequencies, were given higher preference than the higher frequencies. This is not a sophisticated method for the ranking of features as it does not use any analysis of the data to arrive at a decision. Rather it is based on the fact that for human audio samples, it is widely considered that lower frequencies contain more important information than the details present in the higher frequency region. We present the experiments performed using this approach in the experiments section.

2.2.1.2 *F-Ratio*

Another method for feature selection is called F-Ratio method [7, 53]. It uses inter-feature and intra-feature variance to arrive at the merit of the features. F-Ratio for i^{th} feature can be calculated as follows:

$$F_i = \frac{B_i}{W_i} \tag{2.3}$$

where B_i is the inter-feature variance and W_i is the intra-feature variance. The inter-feature variance is given by

$$B_i = \frac{1}{K}\sum_{k=1}^{K}(\mu_{ik} - \mu_i)^2 \tag{2.4}$$

and the intra-feature variance is given by

$$W_i = \frac{1}{K}\sum_{k=1}^{K} W_{ik} \tag{2.5}$$

where μ_{ik} and W_{ik} are the mean and variance of the i^{th} feature and the k^{th} group and μ_i is the overall mean of the i^{th} feature across groups. Here, a class is a particular feature within the feature vector. All the features vectors for a particular speaker are used to estimate the intra class and inter class variances. The above relation is then applied to reach an F-ratio, which gives the merit of each individual feature in the feature vector.

2.2.2 *Feature re-extraction*

Feature re-extraction is the process of transforming the features according to some predetermined criteria. This process is often used to obtain reduced features from existing features. The input to a feature re-extraction algorithm is an initial set of observations of a phenomenon. The algorithm then takes the input observations and transforms them into a set of derived values. These new derived values (features) are expected to remove redundancy and disentangle information contained in the observations. Moreover, this process can also lead to better interpretations of the model and system.

Feature re-extraction techniques can be classified as *linear* or *non-linear*. *Linear* dimension reduction techniques are applied first in the following experiments and two of the non-linear dimensionality reduction techniques are also investigated. Linear dimensionality reduction techniques are preferred since it is easy to analyze high dimensional data with linear methods due to their simple geometric interpretations. Usually the linear methods are also computationally less intensive than their non-linear counterparts. These methods focus on simple data characteristics such as covariance, eigen vectors,

eigen values, etc. Linear dimensionality reduction techniques can be used for exploring the geometric structure, compressing the data and extracting meaningful feature spaces.

On the other hand, *non-linear* dimensionality reduction techniques, in general, project data on to a higher dimensional space, instead of a same or lower dimensional space, where the geometry of the data is learnt. Many times, due to the intrinsic properties of data, it becomes nearly impossible to separate the data in lower dimensions for the purposes of classification and related tasks. In such cases, the data is higher dimensions where it becomes possible to linearly separate the data belonging to different classes. Examples of non-linear dimensionality reduction techniques include Kernel Principal Components Analysis, Isometric Mapping, Locally Linear Embedding, Non-Linear Principal Components Analysis, etc.

2.2.2.1 *Principal Components Analysis (PCA)*

One of the most popular linear methods for feature re-extraction is Principal Components Analysis (PCA). PCA [54] is a statistical technique. It uses orthogonal transformation to convert a set of measurements of possibly correlated variables into a set of linearly uncorrelated values. These values are called principal components. The number of principal components obtained through this technique is equal to the number of original variables. The formulation of PCA ensures that the first principal component has the largest variance. Each successive component has the next largest variance while being orthogonal to each of the previous components. Thus, it provides a new set of basis vectors.

The next part of the procedure comprises of actually obtaining the transformed representation using the basis vectors obtained through PCA. We can achieve this by projecting the original feature vector on the new basis. Furthermore, we can drop the vectors from the basis that have very low variance associated with them. In effect, we drop those columns from the transformation matrix that are associated with low variances. Using this transformation, the feature space can be projected onto a smaller subspace which best preserves most of the variance of the input data [53].

2.2.2.2 *Linear Discriminant Analysis (LDA)*

Linear Discriminant Analysis (LDA) [65] is another method used for dimensionality reduction that aims to retain as much class discriminatory information as possible. The main objective in LDA is to project the feature space onto a smaller subspace that contains most of the class variability. This is achieved by maximizing the F-Ratio of the training data in the transformed space. In this way, it can be considered as a generalization of the F-ratio method. LDA is different

from PCA in that LDA explicitly considers the class labels whereas PCA does not consider the class information. LDA has been successfully applied in many areas of research. Example applications include face recognition, where reduced representation of face images are obtained before proceeding to the classification stage. Another application is bankruptcy prediction using accounting and other financial data. LDA has also been successfully applied in biomedical and earth science domains.

2.2.2.3 *Factor Analysis (FA)*

Factor Analysis (FA) [69] is a linear generative model in which the latent variables are assumed to be Gaussian. The core idea of the technique is to model the observed variables in terms of a smaller number of 'factors' which are hypothesised to cause the observations. That is, factor analysis is a statistical tool that helps to explain the variability in the observed data, with the help of fewer, uncorrelated variables. For example, it is possible that in a given set of observations of 5 variables only 3 underlying factors are causing variations in all of those 5 variables. These kinds of problems can be tackled using factor analysis method.

The main objective is to try to model the observed measurements in terms of linear combinations of the hypothesized factors, along with some error term. With this kind of formulation, we can potentially represent a large sized observation in terms of smaller, unobserved causal variables. Not only is this beneficial for a faster processing of the signal, but this also tends to improve performance accuracy. With this kind of representation, we can then convert the original features into new and smaller sized features. The transformation of features into a smaller subspace is achieved by a loading matrix. The loading matrix is estimated using the formulation mentioned above.

2.2.2.4 *Kernel Principal Components Analysis (KPCA)*

The classical PCA approach works well only if the data is linearly separable. However, it is not guaranteed that data will always be of such type. Many natural phenomena produce data that can be highly non-linear. For data that is not directly linearly separable, non-linear reduction techniques have to be applied in order to make sense of the underlying structure of the data. The use of kernel functions or kernel trick is one of the solutions to deal with linearly inseparable data. A non-linear mapping function ϕ is defined so that the mapping of a data point can be written as $x \rightarrow \phi(x)$. This is called kernel function.

In case of classical PCA, covariance matrix is calculated for the data set but in case of KPCA the covariance matrix in the higher dimensional space is not computed explicitly, thus saving tedious computa-

tions. One such implementation is Radial Basis Function (RBF) kernel PCA.

2.2.2.5 *Isometric Mapping (ISOMAP)*

In [70] an approach to overcome the problem of finding intrinsic geometry by PCA or Multi Dimensional Scaling (MDS) is presented. Linear methods such as PCA and MDS fail to detect geometry of non-linear manifolds. This approach combines major features of PCA and MDS to learn non-linear manifolds. With all pairwise distance of data points, the approach tries to discover the intrinsic geometry of data. The crux of the algorithm lies in finding the geodesic distance between far away points. Far away distances are approximated by adding up sequence of short hops. These approximations are computed by finding shortest paths in a graph with edges connecting neighboring data points.

2.3 EXPERIMENTAL LAYOUT

2.3.1 *Data*

For the following experiments, we used two datasets. First, for a quick testing of the concept, we used our own recorded data. It consists of the ten English digits spoken by 10 male speakers in the age group from 23 to 27 years. Each digit was recorded 20 times at 16000 samples/sec with 16 bit representation using Cool Edit Pro 2.1 recording studio software. A total of 2000 utterances were used in the complete process. For the training part, 1500 utterances out of the 2000 were used. The rest 500 utterances were used as test data. The complete setup used for recording our own data is detailed in Table 2.1.

Table 2.1: Table showing details of recorded data

S. No	Item	Value
1	Utterances spoken	10 English digits (0-9)
2	No of speakers	10
3	Age	23-27 years
4	Repetitions per digit	20
5	Sampling Rate	16000
6	Representation	16 bit
7	Software used	Cool Edit Pro 2.1
8	No of training utterances	1500
9	No of testing utterances	500
10	Total utterances	2000

For a more complete and robust set of experiments, we used the TED-LIUM dataset [59]. It consists of recorded Ted Talks by different people with different backgrounds. For our experiments, the data consisted of 20 speakers. Each speaker's talk is for 4 minutes. Most of the talks contained some music, claps and silence. Since these components do not contribute toward speaker recognition, they were removed manually while retaining most of the speech. Each speaker had approximate 2 minutes of training data and 50 utterances of testing data, each of approximately 1 second. The details of TED-LIUM data are presented in Table 2.2.

Table 2.2: Table showing details of TED-LIUM data

S. No	Item	Value
1	Utterances spoken	TED Talks
2	No of speakers used	20
3	Training data	2 minutes
4	Testing data	1 sec
5	No of test cases per speaker	50

2.3.2 *Feature extraction*

For feature extraction in our experiments, we used the Hidden Markov Model Toolkit (HTK) [75]. The HTK toolkit is a toolkit for building Hidden Markov Models (HMMs). The toolkit is used widely to develop and deploy mainly speech recognition systems all over the world. It contains a suite of tools in C source code as well as binary executables that can be directly run. We used the HCOPY tool from HTK for the purpose of extracting Mel Frequency Cepstrum Coefficients (MFCCs) for each frame of the audio sample. The audio samples were first converted into frames of 320 samples each, along with sliding window of 80 samples. Finally, 12 dimensional feature vectors were obtained from HCOPY tool, along with the log energy component. We choose a relatively smaller sized feature vector to begin with because the focus of the current experiments is to reduce the size of the feature vector in order to achieve a faster processing and output time. Hence the size of feature vector was chosen as 12 and further reduced thereafter to achieve a minimal sized feature vector.

2.3.3 *Feature selection and transformation*

The following describes how the actual process of selection and transformation of features was carried out for the experiments. To perform feature selection, the feature vector was reduced by leaving out a par-

ticular feature from the feature vector. The decision as to what feature should be dropped was determined by the rank of the particular feature. Concretely, we leave out one feature that has the lowest merit as determined by the chosen criterion. From 12 features, we drop 1 and perform the speaker recognition experiments with 11 features. Next, we drop the lowest merit feature from the current features and perform recognition experiments with 10 features. This dropping of features is carried out until we are left with only 2 features. We observed the behaviour of the speaker recognition system during the course of this feature dropping process. The results and graphs are discussed in the following pages.

The process of feature re-extraction follows a slightly different path. First, we need to obtain a transformation matrix. The transformation matrix represents the crux of the process of feature re-extraction. The transformational capabilities of the matrix will depend on the method used to estimate the matrix. This matrix is a direct result of the underlying preprocessing method such as the already discussed PCA, LDA, Factor Analysis, etc. Thereafter obtaining this matrix, which represents a type of new basis, we project our existing data (raw features) on to this new projection space, thereby obtaining the new and possibly better representation. But this vanilla version of the feature re-extraction process doesn't yet lead to the reduction in the size of the feature vector. To actually reduce the feature vector size, we can leave out those basis vectors from the new basis obtained that account for relatively smaller amount of information. This corresponds to removing appropriate columns from the transformation matrix. Finally, we can just multiply the transformation matrix with the existing features to obtain a reduced feature vector. For the reduction part, we again follow similar procedure as in the the feature selection process. We begin with 12 features. We remove the lowest information containing vector from the new basis to obtain an 11 dimensional feature vector. We use this 11 dimensional vector to perform the speaker recognition experiment. Next we go on to reduce the feature vector to size 10 in a similar way and repeat the experiments with each new smaller dimension.

2.3.4 *Experiments*

We performed 5 experiments with our recorded dataset and similar experiments with the TED-LIUM dataset. Each of the five experiments correspond to one of techniques for feature reduction – either selection or re-extraction.

First, we used the Naïve dropping technique to reduce the features through selection. We started by reducing the data to 11 dimensions by leaving out the last feature as per the original order of the features, i.e. MFCCs. This process was continued until only 2 features were left

in the feature vector. Every time a feature was dropped from the feature vector, we performed a separate speaker recognition experiment to determine the accuracy of the reduced feature set for that particular dimension. The same strategy was followed for the F-Ratio method as well, only with the difference that the merit criterion was changed to F-Ratio of the features instead of the order of the MFCCs. The feature with the least F-Ratio was dropped first.

While working with feature re-extraction techniques, we obtained a transformation matrix that enables the transformation and reduction of feature vector. This matrix was obtained differently for each of the technique used. Next, we dropped the appropriate columns from each of the matrix, for each of the dimension (11 down to 2), to prepare the transformation matrix. Lastly, we multiplied the feature vector with the corresponding matrices to obtain reduced features, also from dimensions 11 down to 2.

For each of these reduced dimensions, we input the complete training data to the system for estimating the speaker models using the GMM-UBM approach. For training model, we used 32 mixtures of Gaussians to represent each speaker. The decision to use a lower count of mixtures was again motivated by our main objective which is to lower the computations.

2.4 RESULTS AND DISCUSSION

Figures 2.1 to 2.5 depict the trends of accuracy at speaker recognition experiments with reduced features for our recorded dataset. With Naïve Dropping method (Fig. 2.1), we observe that as we start leaving out coefficients from the feature vector, reducing the number of MFCC coefficients from 12 to 7, the accuracy hovers between 78.2% to 83.4%. But as we continue dropping the MFCCs further, the accuracy drops significantly – to 66.2% with 4 coefficients, to 47.2% with 3 coefficients and to 34% with only 2 coefficients. This trend of performance indicates that the first 5 MFC Coefficients are very critical to the speaker distinguishing and recognition process. The rest of the features, 6^{th} MFCC up to 12^{th} MFCC, help increase the accuracy and augment the performance of the speaker recognition system only slightly, especially when compared to the first 5 coefficients. If we select and use only first 5 coefficients for speaker recognition experiments, we can reduce the feature set size by more than 60% while still achieving a reasonable performance as shown in the results.

With the F-ratio method (Fig. 2.2) for feature selection, we again observe that dropping the last 5 MFC coefficients do not hamper the recognition performance significantly. With such a configuration, the accuracy ranges between 81.4% to 86.2%. However, on further drop of MFC coefficients, speaker recognition performance starts to degrade rapidly. This behaviour points toward the importance of first 7 fea-

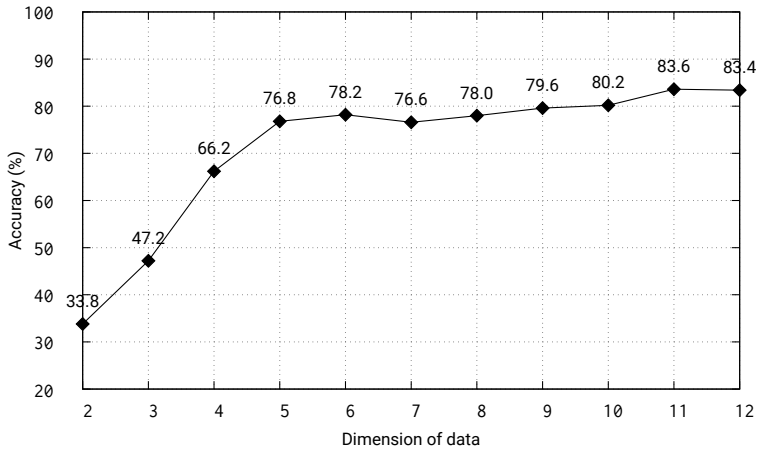

Figure 2.1: System performance as dimension is reduced with Naïve dropping for our own recorded data

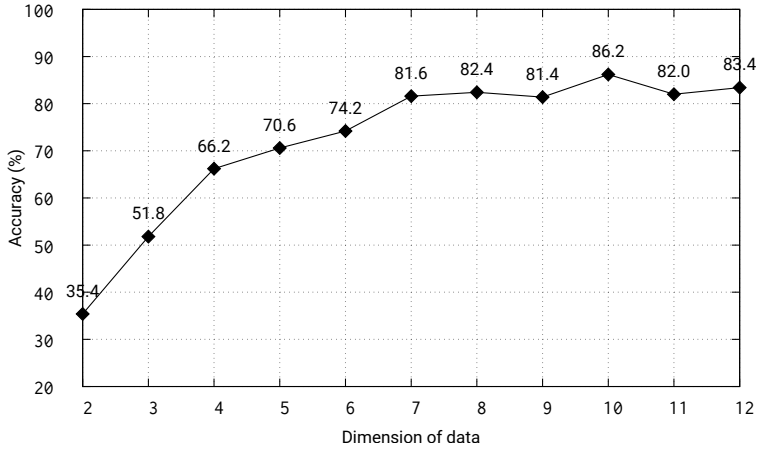

Figure 2.2: System performance as dimension is reduced with F-Ratio for our own recorded data

tures as determined by the F-ratio criterion. By retaining only the first 7 features as per the above criterion of F-Ratio, we are able to reduce the size of our feature vectors by more than 41% while maintaining reasonable accuracy in speaker recognition experiments.

We find similar trends in the experimental outcomes with feature re-extraction techniques. We observe that with PCA (Fig. 2.3), the contribution of the first 6 features in the feature vector is quite significant. The rest of the features, coefficient 7 to 12 in the transformed feature vectors, do not contribute as greatly to the recognition performance. The observed behaviour reaffirms our expectation from this experiment. Such a trend could be the result of features being ranked in order of decreasing variance, with lower order coefficients have larger variances. Thus, it could be concluded that the lower order coefficients are able to capture more information than their higher order counterparts.

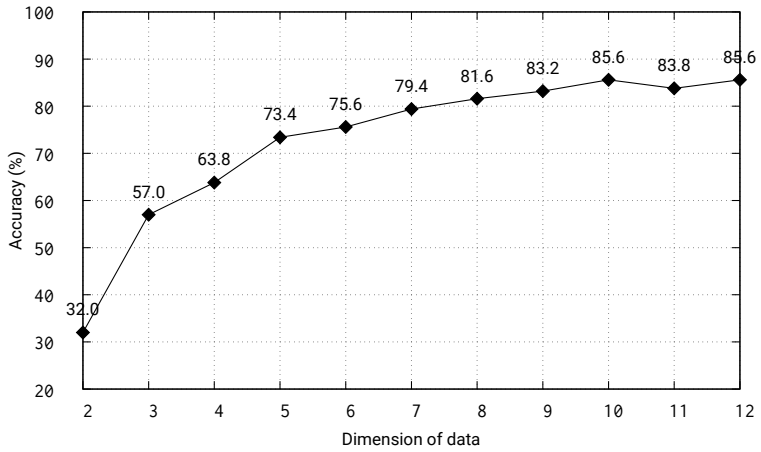

Figure 2.3: System performance as dimension is reduced with PCA for our own recorded data

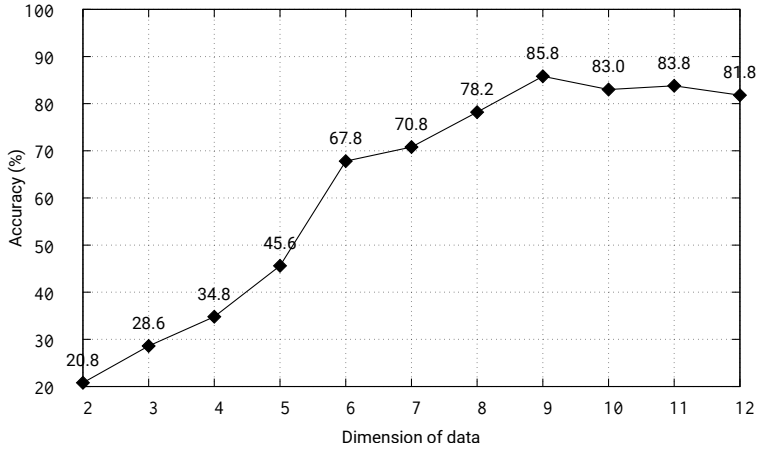

Figure 2.4: System performance as dimension is reduced with LDA for our own recorded data

In case of feature re-extraction with Linear Discriminant Analysis (LDA) (Fig. 2.4), we note that first 9 MFC coefficients in the feature vector contribute significant value to the performance of the speaker recognition experiments. We observe pointed increases in the accuracy with addition of each feature. However, notable increments are not observed when further features are added to the feature vector. On the contrary, the performance of the recognition system deteriorates to some extent. Therefore, it can be argued that with further addition of features, the system complexity seems to go up. It is also plausible that the addition of MFC coefficients after a certain point causes the system to confuse between speakers rather than contributing positively towards better recognition. Using the information from observed trends, we can reduce the feature vector size by 25% while delivering a superior performing system.

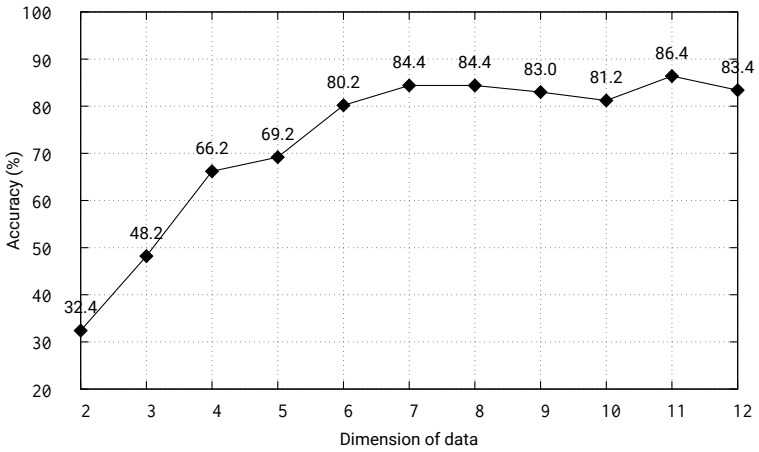

Figure 2.5: System performance as dimension is reduced with Factor Analysis for our own recorded data

Re-extracting features with Factor Analysis (Fig. 2.5), we observed that the speaker recognition performance vacillates between 84.4% to 86.4% while the feature vector size is varied from 12 down to 7. On further reduction of features from the feature vector, the performance seems to deteriorate precipitously. The observations lead us to conclude that it is possible to obtain more than 41% reduction in feature vector size while suffering a minute dip in performance. Table 2.3 summarizes the results of all the experiments with our recorded data concretely.

Table 2.3: Results showing accuracy v/s different dimension of data for different dimensionality reduction techniques for our recorded data

Dim.	Naïve	F-Ratio	PCA	LDA	Factor A
2	33.8	35.4	32.0	20.8	32.4
3	47.2	51.8	57.0	28.6	48.2
4	66.2	66.2	63.8	34.8	66.2
5	76.8	70.6	73.4	45.6	69.2
6	78.2	74.2	75.6	67.8	80.2
7	76.6	81.6	79.4	70.8	84.4
8	78.0	82.4	81.6	78.2	84.4
9	79.6	81.4	83.2	85.8	83.0
10	80.2	86.2	85.6	83.0	81.2
11	83.6	82.0	83.8	83.8	86.4
12	83.4	83.4	85.6	81.8	83.4

Figure 2.6 to 2.13 depict the performance of speaker recognition system when feature reduction was applied to a subset of the TED-LIUM dataset. Although there are close to 1500 speakers present

in the TED-LIUM dataset, these experiments were conducted for 20 speakers. Initially, as with the digits dataset, linear dimensionality reduction methods were applied to the speakers' data. However, some *non-linear* dimensionality reduction methods were also explored in these experiments.

It can be inferred from the graphs that the curve for accuracy of speaker recognition versus dimension of data follows a similar trend to that of the results of the experiments conducted with digits data. Following the same strategy for dropping coefficients, with Naïve dropping method (fig. 2.6), reducing the dimension of features from 12 to 7 causes the accuracy of system to slowly descend from 74.7% to 65.5%. Further we observe that dropping the six higher order (as per their natural order in Naïve dropping method) features from the feature vector affects the performance of system very minimally. But as we continue to drop the features further, the speaker recognition performance degrades quite rapidly. This response indicates that the 6 lower order coefficients carry more speaker related information than the higher order coefficients. This is very similar behaviour as observed with digits data.

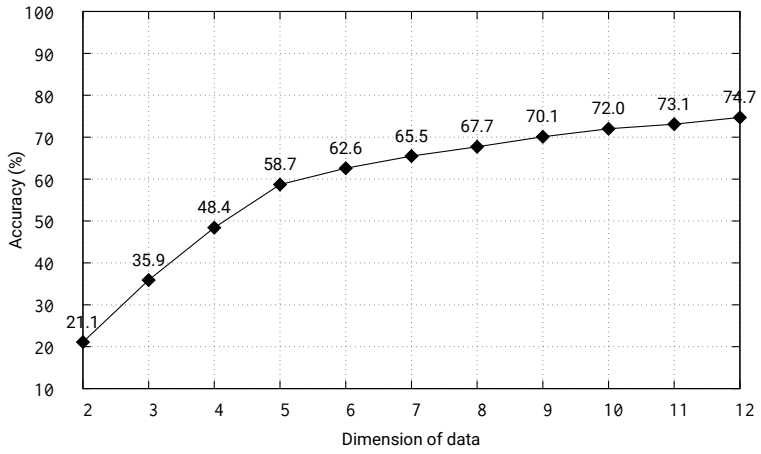

Figure 2.6: System performance as dimension is reduced with Naïve dropping for TED-LIUM data

As we can observe from fig. 2.7, in case of F-Ratio method, dropping the first 6 features again degrades the performance of the system significantly, whereas dropping the last 6 features doesn't hamper the performance of the system as much. This helps to further strengthen the importance of the lower order coefficients for the task of speaker recognition. Using the feature selection methods to reduce the size of feature vector, it is possible to drop about 40% of the features from the feature set and still be able to obtain reasonable performance for speaker recognition task.

Employing PCA (fig. 2.8) for the purpose of feature reduction, we observed that while reducing the feature vector size from 12 down

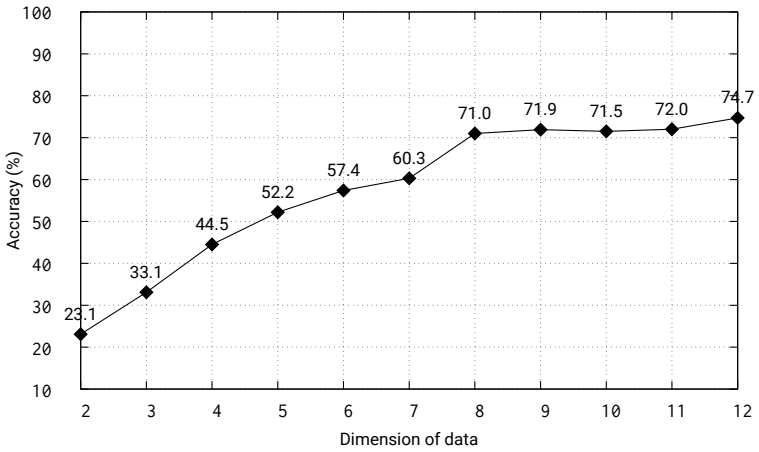

Figure 2.7: System performance as dimension is reduced with F-Ratio for TED-LIUM data

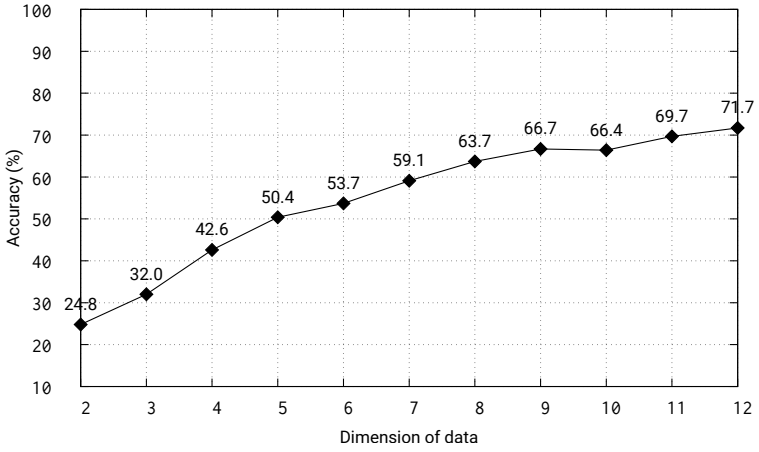

Figure 2.8: System performance as dimension is reduced with PCA for TED-LIUM data

to 6, the accuracy performance varies between 71.7% to 59.1%. We observe a similar curve as before between accuracy of speaker recognition with respect to the dimension of the feature vector.

Similar outcomes and trends of speaker recognition performance were observed in case of feature reduction with LDA (fig. 2.9) and Factor Analysis (fig. 2.10) on TED-LIUM speaker data of 20 speakers as that of experiments conducted with digits data of 10 speakers. These experiments indicate that first 9 features contribute heavily to the system's performance at speaker recognition task. We can see the matching trends in the graphs for both the techniques.

Unlike the experiments with digit data, some of the non-linear feature reduction techniques also were used with the TED-LIUM speaker data. Speaker recognition experiments after feature reduction with KPCA (fig. 2.11 and 2.13) demonstrated overall lower accuracy as compared to its linear counterpart PCA. In [30] it is mentioned that

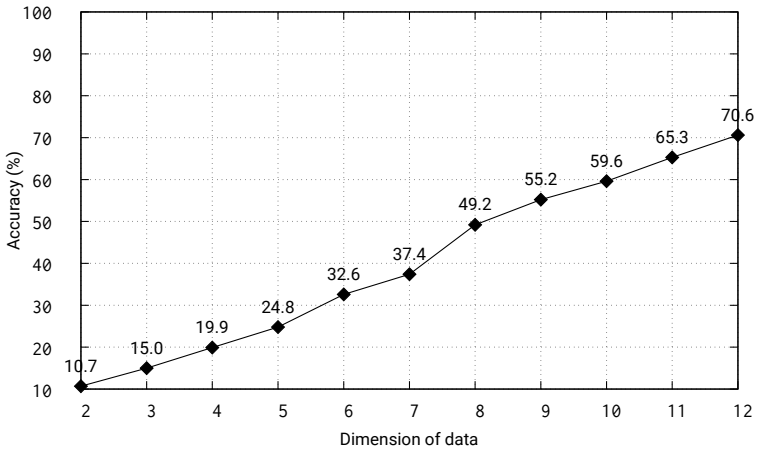

Figure 2.9: System performance as dimension is reduced with LDA for TED-LIUM data

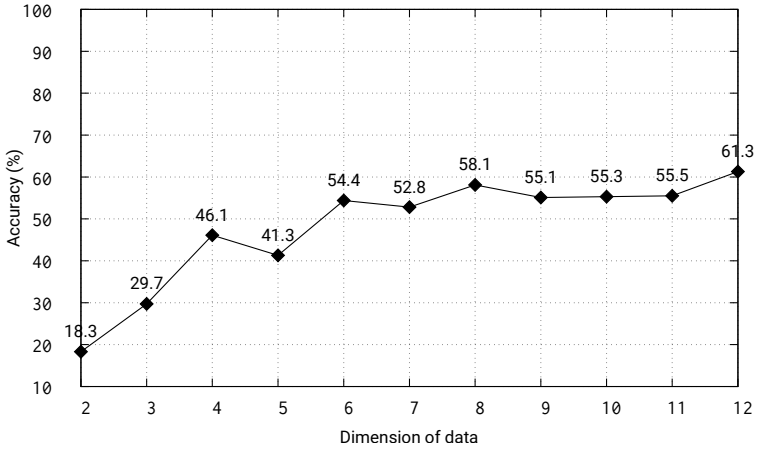

Figure 2.10: System performance as dimension is reduced with Factor Analysis for TED-LIUM data

the results can be sensitive to the choice of kernel used. It can be observed from fig. 2.11 that the accuracy of the recognition system doesn't vary significantly while features are reduced from 12 down to 5. However, the recognition system's performance deteriorate precipitously on further reduction in the feature vector size. Using KPCA it can be inferred that only 5 features are sufficient for system to provide performance level comparable to that obtained with the whole unaltered feature vector.

Intriguing results were obtained in case of feature reduction with the non-linear technique called ISOMAP (fig. 2.12). The performance of the speaker recognition system with the complete feature vector was increased by 32% on feature re-extraction with ISOMAP. The system demonstrated the highest accuracy of 94% at feature set size 10 and even after dropping more than 50% of the features from the feature vector the recognition system was able to perform at high ac-

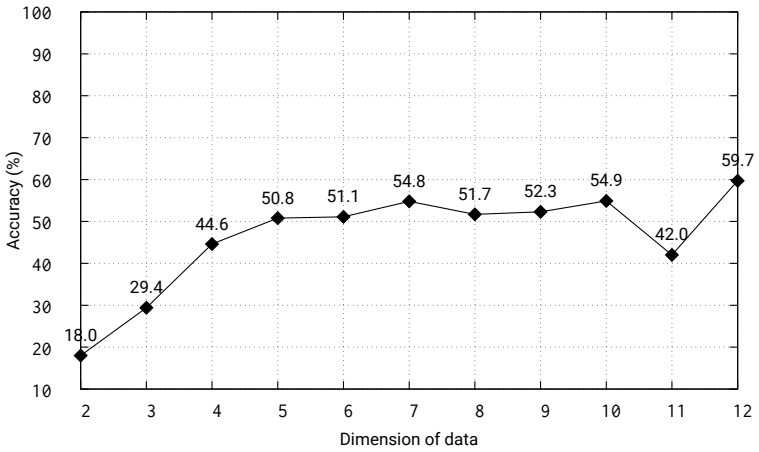

Figure 2.11: System performance as dimension is reduced with KPCA for TED-LIUM data

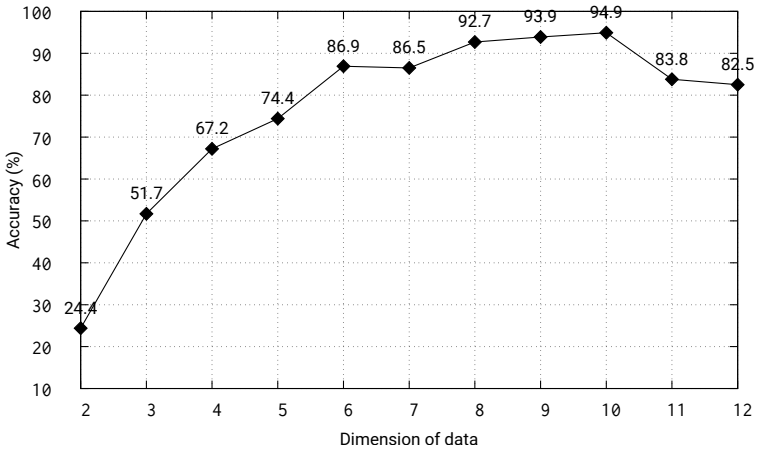

Figure 2.12: System performance as dimension is reduced with ISOMAP for TED-LIUM data

curacy levels. As observed with other feature reduction techniques earlier, the lower order features contribute critically toward the performance of the recognition system. Sudden drop in the accuracy is observed on further reducing the feature set size below 5. Table 2.4 lists the complete results obtained with the TED-LIUM dataset.

It can be inferred from the experimental observations that it is possible to achieve significant amounts of reduction in the size of the feature vector for the task of speaker recognition. The high gains in the feature vector size may require a minor sacrifice in the accuracy of the recognition system. However, with the reductions in the feature vector size, a significant speed up of the overall recognition can be achieved. Using a smaller feature vector can greatly reduce the amount of computations required for performing a recognition task. With reduced computations, we can expect the low resource devices to be able to perform the recognition tasks on the device itself.

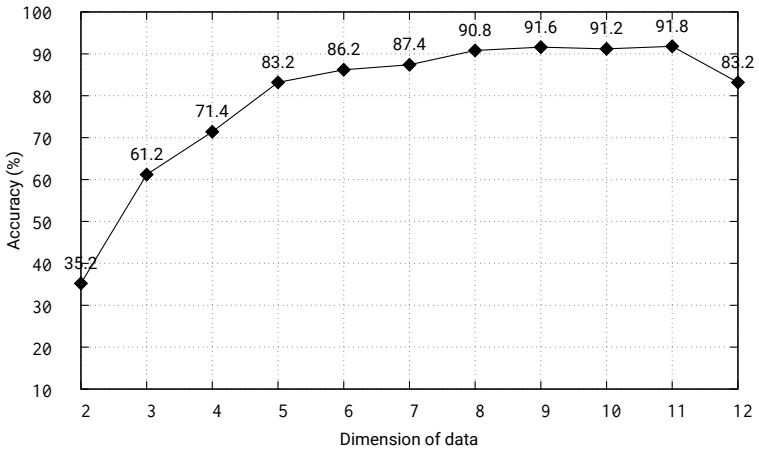

Figure 2.13: System performance as dimension is reduced with KPCA-RBF for TED-LIUM data

This approach of reducing the feature vector size can also be highly useful for scenarios where we require a two-pass recognition system or cases when we do not require extremely high accuracy but execution speed is critical. For such a two-pass system, this approach with reduced feature vector can serve as the first layer of the system for quickly pruning the search space of possible speakers. The second layer can take then search exhaustively in the pruned spaces of speakers to arrive at the final decision in the speaker recognition process.

2.5 CONCLUSION

In the preceding experiments, we investigated how the performance of speaker recognition system varies with respect to reduction in size of the feature set. These experiments were designed to help evaluate the performance at different sizes of the feature vector. With such trends available, it is possible to design systems that can provide similar performance levels but at a lesser cost in terms of time and computation since a smaller feature vector implies faster running times and lower computational costs. It is also possible to design systems specifically tailored for low resource devices that cannot afford to perform complex calculations needed for state-of-the-art methods.

We first used the Naïve dropping technique for reducing the feature set size. It provided promising results but there was no legitimate way to determine the rank of the features with this method as the order of the features itself is taken to be the rank of the feature in the feature vector. Therefore, the F-ratio method was tested to systematically decide on the merit of each feature. This methods provides a better approach to reduce the feature set size based on the variance property of individual features within the complete feature vector.

Table 2.4: Results showing accuracy v/s different dimension of data for different dimensionality reduction techniques for TED-LIUM data

Dim.	ND	FR	PCA	LDA	FA	KPCA	ISOMAP	RBF
2	21.1	23.1	24.8	10.7	18.3	18.0	24.4	35.2
3	35.9	33.1	32.0	15.0	29.7	29.4	51.7	61.2
4	48.4	44.5	42.6	19.9	46.1	44.6	67.2	71.4
5	58.7	52.2	50.4	24.8	41.3	50.8	74.4	83.2
6	62.6	57.4	53.7	32.6	54.4	51.1	86.9	86.2
7	65.5	60.3	59.1	37.4	52.8	54.8	86.5	87.4
8	67.7	71.0	63.7	49.2	58.1	51.7	92.7	90.8
9	70.1	71.9	66.7	55.2	55.1	52.3	93.9	91.6
10	72.0	71.5	66.4	59.6	55.3	54.9	94.9	91.2
11	73.1	72.0	69.7	65.3	55.5	42.0	83.8	91.8
12	74.7	74.7	71.7	70.6	61.3	59.7	82.5	83.2

Next, feature re-extraction methods were tested for feature reduction on the data. Feature re-extraction methods fall in two categories: Linear methods and Non-linear methods. Linear methods used in our experiments include PCA, LDA, Factor Analysis. KPCA and ISOMAP are examples of non-linear methods used.

First PCA was applied for feature reduction. PCA helps to transform the data using the variance property and then feature reduction is achieved by projecting original features on to a lower subspace. It was observed that approximately 10% performance was sacrificed while achieving about 50% reduction in the size of feature vector. With LDA as the feature re-extraction technique, similar trends of speaker recognition performance were observed. However, with the same amount of performance deterioration, the feature set size was reduced to a higher extent with LDA than what was achieved with PCA. Hence, in this view, LDA would be considered as the preferred method against PCA. In case of Factor Analysis, it was observed that it was possible to reduce the feature set size to more than 40% with minimal performance degradations. Nonlinear feature re-extraction methods viz. KPCA and ISOMAP, also demonstrated similar performance trends as the linear re-extraction methods for feature re-extraction. With KPCA, an overall lower accuracy of the speaker recognition system was observed but the trend of curve drawn with accuracy against dimension of features remained similar to the linear methods. It was possible to reduce the feature set size by approximately 50% with a minimal performance sacrifice, with performance even increasing at some lower dimensions. This can be concluded from the graphs presented above. With ISOMAP, it was again observed that it is possible to reduce the feature set by 50% while obtain-

ing an overall higher performance by approximately 30%. The best performance was observed with feature set of size 10 with ISOMAP technique. However, all re-extraction methods incur an overhead when the transformation of the test feature vector is applied during the testing phase as each test vector has to multiplied by a loading matrix.

Overall it can be concluded that using either re-extraction or selection techniques, it is possible to provide significant reductions in feature vector size which can directly affect the response time of the speaker recognition system. These reductions can lead to reduced processing time and thus enable us to provide speaker recognition services on low resources devices themselves. For example, let us consider the original dimension of MFCC feature set to be 12 and the reduced dimension to be 6. For calculation of component density of one Gaussian, for a single frame of speech, $O(n^{11})$ time steps are needed approximately where n is size of the feature. When original feature size of 12 is used, it comes to approximately $7.43x10^{11}$ time steps. For the reduced feature of size 6, the same evaluates to $6.5x10^{9}$ time steps. Hence we see a distinct reduction in the number of time steps needed for evaluating a single Gaussian density for one frame. As per the calculation, this will lead to a significant improvement for 32 mixture Gaussian with approximately 100 frames for a complete test utterance. We calculate this using standard running time complexity for matrix multiplication and inversion operations.

3

EXPERIMENTS WITH I-VECTORS

In the last chapter, we looked at reducing the size of feature vectors by processing the raw features. These processed features were then used to build speaker models that were used in the recognition step where we estimate the probability of the test utterance coming from one of the speakers. In this chapter, we discuss the experiments with i-vectors and the reductions obtained at the model level. The high dimensional speaker models are reduced to a lower dimension to make the speaker recognition faster and cheaper in terms of computation.

Numerous approaches have been applied to tackle the problem of automatic speaker recognition. For a long time, the GMM-UBM model was accepted as state-of-art approach for speaker recognition tasks. In 2005, the concept of Joint Factor Analysis (JFA) was introduced [34, 36]. The idea was to model the variabilities of speaker and channel effects separately as two distinct spaces. However, i-vector modeling was soon proposed as an improvement to JFA modeling.

The i-vector paradigm has been an active area in speaker recognition & verification research [71]. Dehak *et al.* [10] proposed a channel-blind approach for telephone as well as microphone data. In [2, 22], the authors have presented ways to calculate i-vectors in an efficient way. The use of PLDA for channel and speaker compensation has been discussed in [32]. At the same time [61] demonstrated that if training utterances are sufficiently long, shorter utterances for testing can also be used with a good recognition performance. Kenny *et al.* [37] showed that PLDA is useful even when the test utterances length is variable. In [4], the authors have tried to reduce the data requirement for training by using k-nearest neighbor (k-NN) algorithm. Mandasari *et al.* [46] has compared robustness of different approaches to speaker recognition. I-vectors have also been employed in language recognition tasks [12, 49]. In [43], the authors have used an extended feature vector, consisting of MFCCs with some other features in tandem, before the calculation of i-vectors. They have discovered the differences between features that are beneficial for speaker recognition and those suited to language recognition tasks.

Several studies have been conducted suggesting the use of i-vector based methods to solve the problem of speaker identification. Many studies have also experimented with different compensation techniques to be used along with the method. However, it is not explicitly investigated why a particular configuration of the method should be used. The motivation behind the following experiments is to determine the optimal size of the total variability matrix, T, which should

be used for generating i-vectors so that the computations involved can be reduced to a minimum, along with maintaining a reasonably high accuracy. It is also worth exploring how, for a particular size of T, the number of mixture in the UBM affect the performance of the recognition system. Other interesting variables are the size of the training and test utterances. Exploring these subtle but important issues is also one of the motivations behind this study.

3.1 SPEAKER MODELING WITH I-VECTORS

As described in the previous chapter, Gaussian Mixture Models - (GMMs) - were used as the standard modelling technique for speaker recognition tasks for almost a decade [57]. GMMs have the ability to form smooth approximations to arbitrary densities and hence, approximate complicated functions. The intuition behind the use of GMMs as speaker models is that the components of the Gaussian mixture may in some way represent speaker's broad phonetic events and therefore speaker's individual characteristics [56]. These properties of GMMs rendered them suitable for speaker recognition tasks. To handle data constraints and enable faster convergence, adapted GMMs were proposed. As per the concept of adapted GMMs, instead of building speaker models from scratch, UBM parameters were used to generate or 'adapt' speaker models from the generalized UBM parameters. The parameter estimation algorithm was correspondingly modified. Instead of the standard Expectation Maximization (EM), Maximum A Posteriori (MAP) algorithm was used for the adaptation of parameters for each individual speaker. The adaptation parameters control the balance of UBM characteristics and speaker utterance characteristics. These adaptation parameters control 'how much' of the UBM properties are copied into speakers' models.

Kenny *et al* [35] proposed a joint factor analysis model to explain most of the variance of speaker and channel through smaller number of factors. The model's aim was to make a decision that whether the difference in the test and reference utterances can be accounted for by inter-speaker variability or intra speaker variability. Intra speaker variability may arise due to channel and speaker's emotional and/or physical state. For example, if a speaker is ill with some disease, his utterances may differ from the way that he generally speaks. Similarly if a person is ill at ease or not feeling mentally well, it may affect the way she produces her utterances leading to marked differences in the utterances themselves. These types of situations can lead to intra speaker variabilities. Inter speaker variability, on the other hand, refers to the differences between distinct speakers' utterances. It is expected that the vocal tract configurations as well as speaking styles and characteristics like intonation, stress and accent are different for different speakers. This leads to differences in the utterances pro-

duced by them. This phenomenon is called Inter speaker variability. It was widely understood that speaker variability modeling was more important than accounting for channel variability. However, these assumptions were found to be misplaced when, in an analogous study [66] in face recognition, it was observed that modelling intra person variability, without modeling inter person variability, led to very high performance, sometimes eclipsing the performance of the system that modelled inter person variability. Such experiments forced researchers to reconsider their assumptions and hence, channel variability modelling was used in the form of Joint Factor Analysis (JFA) model.

Suppose, the number of components in the UBM are C and the dimensionality of the acoustic feature vectors (MFCC) is F, then by concatenating the F-dimensional speaker adapted GMM mean vectors, we get a supervector of size CF. In JFA, a speaker utterance is represented by such a supervector M. As per the JFA approach, a supervector for a speaker consists of and can be decomposed into its constituent components – speaker dependent, speaker independent, channel dependent and residual components. Further, each component can be represented by a low-dimensional set of factors which act along the principal dimensions of the corresponding component.

The joint factor analysis model can be represented as follows:

$$M = m + Vy + Ux + Dz \tag{3.1}$$

where,
M = The speaker dependent supervector
m = A speaker independent supervector (from UBM) of dimension $CF\times1$
V = The eigenvoice matrix (rectangular matrix of low rank)
y = A vector representing the speaker factors
U = The eigenchannel matrix (rectangular matrix of low rank)
x = A vector representing channel factors
D = The residual matrix of dimension CF×CF
z = The speaker specific residual components

The following is the procedure to use the JFA model in practice. First, using offline data for all the speakers, the UBM is constructed as discussed earlier. Then, applying 1^{st} and 2^{nd} order Baum-Welch statistics on the UBM mean vectors, the JFA matrices are trained in the following order–

1. Train V assuming U and D are zero.
2. Using V, train U, assuming D is zero.
3. Using V and U, train the residual matrix D.

Once the enrolment data is available, MFCC are extracted and supervectors are created by concatenation of the adapted GMM mean vectors. Then, the values of the factors y, x and z are computed. These factors are stored for comparison with test values. During testing phase, again factors y, x and z are computed in the same way for the test utterances. Scoring is done by computing the log likelihood of the test utterance feature vectors against speaker models. The highest score is obtained from the log-likelihood scores of all the speakers and the speaker corresponding to the highest score is declared to be the identified speaker.

Dehak *et al* [11] showed that even while modeling channel factors and speaker factors separately, there was still some information about speaker characteristics in the channel factors when modelling with the Joint Factor Analysis approach. With this in consideration, they proposed a single space for modelling both the speaker as well as channel characteristics. This new space was termed Total Variability Space since this space comprises both the sources of variabilities together. The total variability space is learned and used to extract i-vectors. In i-vector modeling, a GMM supervector is represented as follows:

$$M = m + Tw \tag{3.2}$$

where,
M = Speaker and channel dependent supervector
m = Speaker and channel independent supervector (UBM supervector)
T = A low rank matrix which represents the principal directions of the speaker and channel variability (total variability space)
w = A vector representing the total factors
The components of w are called Total Factors and they are referred to as i-vectors.

To implement the i-vector method, the following process is followed. Note that this process is very similar to the JFA training and testing procedure. There are some differences in key places that differentiates this model. Firstly, using training data, the UBM is constructed offline as described earlier. Then, the total variability matrix T is trained, the process for which is exactly similar to training of the eigenvoice matrix V of JFA. However, there is one important difference. On one hand, during the eigenvoice training, all recordings of a given speaker are considered to belong to the same speaker. On the other hand, during total variability matrix training, we pretend that the utterances from a particular speaker are also produced by different speakers. Once the enrolment data is available, MFCC are extracted and supervectors are created by concatenation with the

adapted GMM mean vectors. Then, we can compute the values of the total factors, w. These factors are stored for comparison with test values.

During testing phase, again the total factors w are computed similarly from the test utterances. For comparison of the i-vectors, cosine scoring is used, which is calculated as follows-

$$score(\omega_1, \omega_2) = \frac{\omega_1^t \omega_2}{\sqrt{\omega_1^t \omega_1}\sqrt{\omega_2^t \omega_2}} \tag{3.3}$$

where, ω_1 and ω_2 are the vectors whose distance is to be measured. The value can vary between -1 and 1. The highest score is obtained and the respective speaker is declared a match.

3.2 EXPERIMENTAL LAYOUT

3.2.1 *Data/corpus*

For our experiments, we used a part of the TED-LIUM [59] dataset/ database. The TED-LIUM dataset consists of 1495 TED talks recorded at a sample rate of 16000, 16 bit representation, with a bit rate of 256 K. The encoding used is 16-bit signed integer PCM. Out of the 1495 recordings, we used 300 of them in our experiments which amount to more than 60 hours of speech data. For preprocessing, we removed the initial music found in all the data files.

For our first set of experiments, 50 of the 300 speakers' files were taken and split into two parts: training & testing. For training, we used approximately 15 minutes of the recording for each speaker. The testing part was different for each of the experiments conducted. For the first experiments, we used utterances of 5 seconds in length as test cases. Several such tests were conducted for each speaker resulting in at least 40 test cases for each speaker. The total test cases used were 2819. For the second and third experiment, we used utterances of length 10 and 15 seconds respectively. The fourth test used the complete test files which were of length at least 3 minutes and at most 8 minutes. These tests were designed to understand how the length of the test utterance affects the accuracy of the recognition system.

Further, to understand the role of the length of the training data with respect to accuracy, we conducted a fourth experiment using all 300 files. The training data was split in utterances of length 2, 4, 6, 8, 10, 12, 14, 16, 18 and 20 mins respectively. The other parameters of the system were kept as constants, which were arrived at as optimal from the previous three experiments.

3.2.2 *Feature extraction*

Feature extraction is an important step in any recognition system. For our experiments, we used the SPro tool to extract the features. We used the 60 dimension feature vector for each frame. Here each frame refers to a segment of 10 ms from the utterance. The feature set used is Mel Frequency Cepstrum Coefficients [44, 63]. The 60 dimensional feature vectors consist of 19 static coefficients followed by log energy value, followed by the delta & acceleration coefficients.

3.2.3 *Experiments*

We experimented with different sizes of the total variability matrix for a particular mixture count in the UBM. We started with 16 mixtures in the UBM and increased the mixture count up to 512. We also tested that the performance of system by keeping the i-vector size fixed while varying the mixture count in the UBM. Further, we tested the system with varying length of test utterances. We also evaluated the performance of the system by varying the length of the training data. For performing our experiments we used the ALIZE toolkit [39]. For scoring, we have used the Cosine Distance Scoring method.

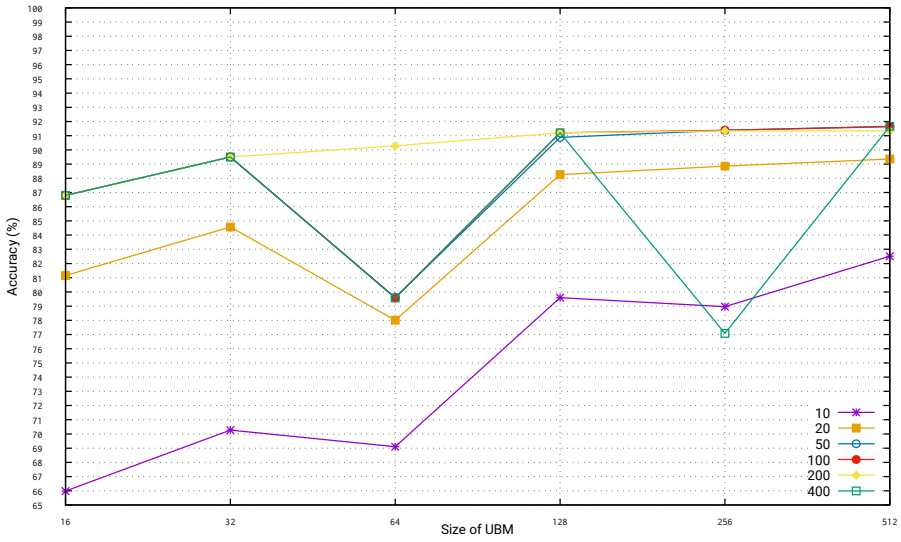

Figure 3.1: Performance with 5 sec test utterances for different T sizes

3.3 RESULTS AND DISCUSSION

In case of 5 sec length test utterances (Figures 3.1 and 3.2), we found that when we vary the number of mixtures in the UBM for different sizes of the total variability matrix, we see a distinct rise in accuracy when mixture count is increased from 16 to 32. However, there is

Table 3.1: Performance evaluation with 5, 10, 15 sec and full test files. Here **T Size** represents `Size of T matrix`, **Accuracy-5** represents `Accuracy for 5 sec case`, **Accuracy-10** represents `Accuracy for 10 sec case`, **Accuracy-15** represents `Accuracy for 15 sec case`, **Accuracy-f** represents `Accuracy for full file case`

Mixtures	T Size	Accuracy-5	Accuracy-10	Accuracy-15	Accuracy-f
512	10	82.51	88.76	88.83	98
512	20	89.36	93.08	92.66	100
512	50	91.66	94.16	92.77	100
512	100	91.63	94.16	92.77	100
512	200	91.34	94.16	92.77	100
512	400	91.66	94.16	92.77	100
256	10	78.96	87.39	88.40	96
256	20	88.86	93.01	91.91	100
256	50	91.38	94.02	93.30	100
256	100	91.38	94.02	93.30	100
256	200	91.34	94.02	93.30	100
256	400	77.08	94.02	93.30	100
128	10	79.60	87.25	87.66	98
128	20	88.26	92.44	91.60	100
128	50	90.88	94.02	92.77	100
128	100	91.20	94.02	92.87	100
128	200	91.20	94.02	92.87	100
128	400	91.20	94.02	92.77	100
64	10	69.10	86.74	87.13	98
64	20	78.01	91.79	90.64	100
64	50	79.60	93.37	92.55	100
64	100	79.60	93.37	92.55	100
64	200	90.28	93.37	92.55	100
64	400	79.57	93.37	92.55	100
32	10	70.27	80.69	84.04	94
32	20	84.57	90.06	89.79	100
32	50	89.50	93.01	92.23	100
32	100	89.50	93.01	92.23	100
32	200	89.50	93.01	92.23	100
32	400	89.50	93.01	92.23	100
16	10	65.98	77.45	81.28	92
16	20	81.16	88.33	88.40	100
16	50	86.80	91.71	91.28	100
16	100	86.80	91.71	91.28	100
16	200	86.80	91.71	91.28	100
16	400	86.80	91.71	91.28	100

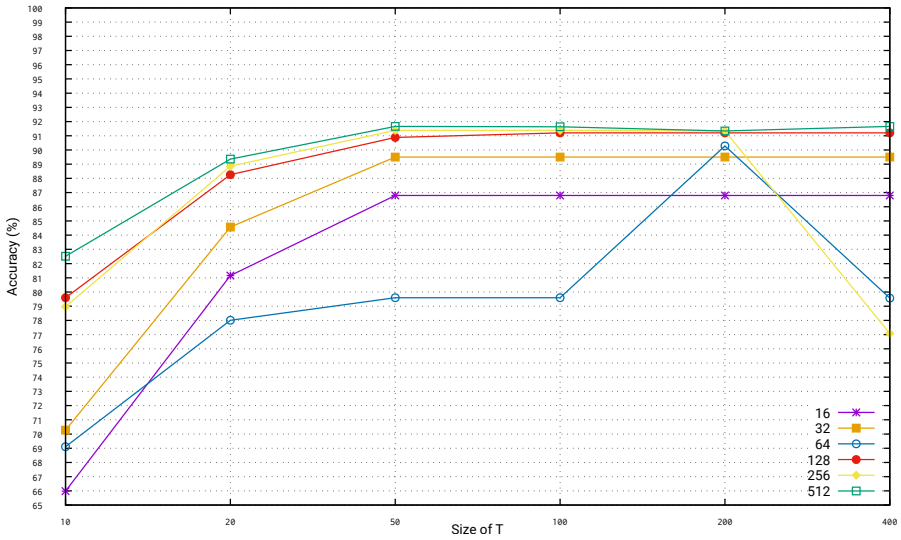

Figure 3.2: Performance with 5 sec test utterances for different UBM sizes

a significant dip at 64 mixtures except when the T matrix size is 200. This indicates that for representing the speaker characteristics, 64 may not be a good mixture count for most of the cases. On increasing the number of mixtures to 128, there is further increase in accuracy but the improvement from 32 to 128 mixtures is less than the improvement observed in going from 16 to 32 mixtures. Increasing the mixture count to 256 and 512 components also does not yield significant improvements. This indicates that after 128 mixtures, we are using a large amount of computation for very small improvements. Thus it would be useful to choose 128 mixtures for speaker representation. However, 32 mixture count can also serve well in cases where there are computational limitations.

We also observed that if we increase the size of the T matrix for different number of mixtures in the UBM, there are no significant gains after the size 50 as the accuracy curve becomes almost horizontal. There is, however, a marked improvement in accuracy of one case when the mixture count is 64. In that case, we see accuracy going from 79.60% at size 50 to 90.28% at size 200. Therefore, T size 50 may be proposed as an acceptable working configuration. Along with this, the mixture count may be chosen as 32 or 128, depending on the availability of computation resources.

Similar trends were seen for 10 sec, 15 sec and full length test files (Rest of the graphs are in Appendix A). Table 3.1 lists the complete set of results. Another interesting observation from the results is that when we increase the length of test utterances, performance increases going from 5 sec length to 10 sec length. However, there is either a dip or no improvement in performance while going from 10 sec to 15 sec utterances. We also see that the performance is 100%, except for the case of T size 10, for complete length test files. This is expected

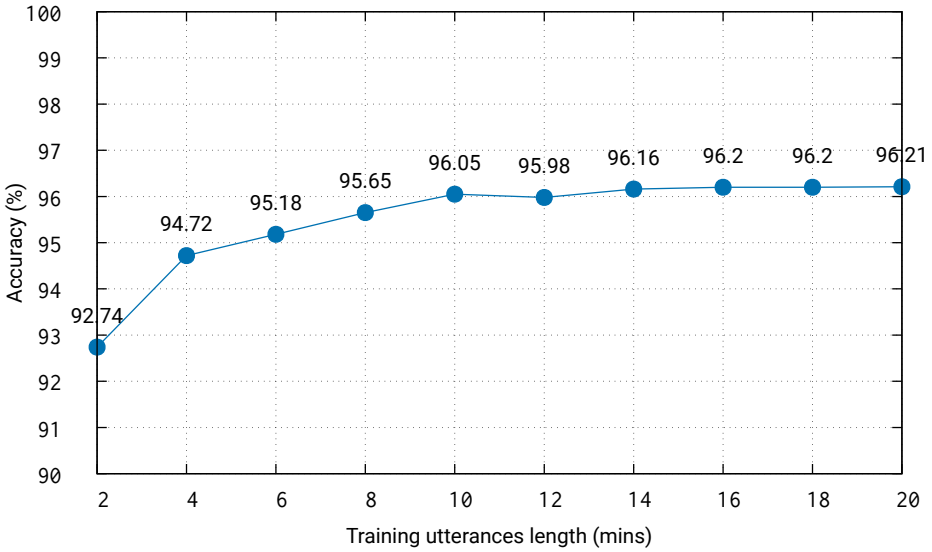

Figure 3.3: Performance evaluation with varying length of training files keeping UBM (128), T size (50) and test utterance length (10 sec) fixed

as the test utterances have sufficient amount of data for predicting accurately.

In the next experiment, we have taken all of the 300 speakers. Taking the optimal results from the above experiment, we have fixed the length of the testing files to be of 10 seconds each. Also, the parameters of the toolkit are similarly set from the output of the above experiments, viz. UBM size of 128 mixtures and a T matrix size of 50.

We find that (Figure 3.3) as we keep increasing the length of the training data from 2 minutes, the accuracy of the system keeps increasing rapidly till 10 minutes. However, there is a slight dip or no improvement in performance, when we go from 10 minutes to 12 minutes, where the performance dropped from 96.05% to 95.97%. Upon further increasing the training utterance length, there is a slow improvement in accuracy, while still not increasing much from the accuracy of 10 minutes. However, the computing time increases. Hence, it seems useful to take the training utterance length at 10 minutes. The detailed results are presented in Table 3.2.

3.3.1 *Additional Experiments: Accent recognition*

While performing the above experiments, we hypothesized a two-pass system where the geographical region or accent of a person could be determined in the first pass and then speaker recognition would be performed on the reduced search space of the particular geographical region or accent only, instead of the complete set of speakers. If there were a total of N accent groups and each accent

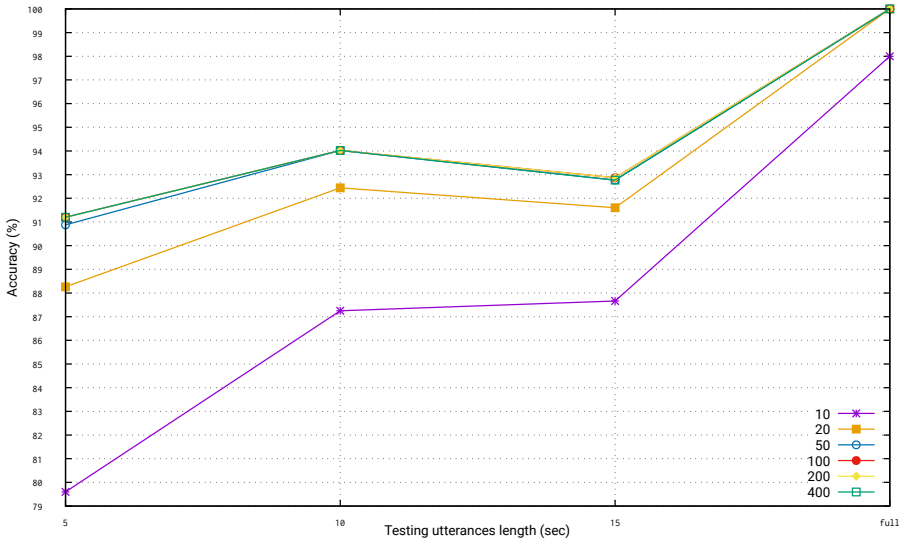

Figure 3.4: Performance with different length of testing files for UBM size 128

Table 3.2: Performance with different length training utterance keeping UBM (128), T Size (50) and Test Length (10 sec) constant

Training Data (Mins)	Accuracy
2	92.74
4	94.72
6	95.18
8	95.65
10	96.05
12	95.98
14	96.16
16	96.20
18	96.20
20	96.21

group has K speakers, then we could determine the speaker identity with the two-pass approach using $N + D$ models (by narrowing the search space to a particular accent group), instead of $N * D$ models (complete set of speakers). To verify this intuition, we performed experiments in this direction.

For the accent recognition experiments, we used a part of the TED-LIUM dataset. Out of the 1495 recordings, we used 40 of them in our experiments. As with the previous experiments, we manually removed the claps and other such portions from the recordings.

For the first set of experiments, speech from 40 speakers were used and split into two parts: training & testing. For training, we used 15 minutes of the recording from each speaker. For testing part for our baseline experiment, we used utterances of 5 seconds in length as test cases. Some of the test cases came from speakers who were not used

Table 3.3: Speaker recognition and accent recognition results

Size of T	Speaker Recognition (%)	Accent Recognition (%)
10	85.97	90.77
20	87.15	92.97
50	94.13	97.26
100	95.08	96.84
200	98.16	96.90
400	99.06	99.63

in training but another similar accented speaker was used to train the corresponding accent. Consequently, simple speaker recognition experiments are bound to fail with these conditions. However, accent recognition performs much better with these experimental conditions. Specifically, out of the 40 speakers, 36 were included in training while 4 were just used for testing purposes. Splitting the test files into 5 seconds lengths resulted in 3283 total test cases. The test cases where the corresponding speaker was included in training were 3263 and where the corresponding speaker was not included in training were 20. Manual classification of every speaker according to his/her accent or broad geographical region was performed. We came up with the following broad categories: African, Canadian, Caribbean, Ethiopian, Turkish, UK, US and Unidentified (if it was not possible to definitively identify the accent category).

3.3.2 *Tools used*

All experiments were performed using the ALIZE toolkit. The ALIZE toolkit is a free and open source toolkit that has been developed at the Laboratoire d'Informatique d'Avignon (LIA) since February 2003. It provides the necessary tools for development of speech and speaker applications and supports i-vector modeling.

3.3.3 *Results and discussion*

We performed the accent recognition experiments for different dimensions of the total variability matrix T. Table 3.3 lists the results. We observe that increasing the dimension of T leads to increase in the accuracy of the predicted accent.

Furthermore, as shown in Table 3.4 we observed that for some of test cases, the predicted speaker turned out to be incorrect. However the predicted accent was correct. This observations supports the intuition that we could use a two-tier recognition system to improve speaker recognition. The first stage could predict the accent and the

Table 3.4: Example where accent recognition performs better than speaker recognition

Size of T matrix	True Speaker	Predicted Speaker	Accent
10	Bill Davenhall	Aaron Koblin	US (both)
10	David Kelley	Dan Ariely	US (both)

Table 3.5: Example where speaker was not in training but accent determined correctly

Size of T matrix	True Speaker	Predicted Speaker	Accent
10	Ellen Gustafon	EdUlbrich	US (both)

speaker can then be determined from the search space of only the particular accent.

Another interesting observation was that accent recognition performed exceedingly well with speakers that were not included in training (Table 3.5). The corresponding accent was trained with one speaker and tested with another speaker of same accent. Out of 20 such cases, 19 were predicted correctly which is very encouraging for unseen data.

We investigated the average cosine distances, listed in Table 3.6 between the i-vectors of speakers of the same accent group. As expected we found that the similar accent speakers have i-vectors which are closer to each other in the total variability space. However, some of them were separated with bigger differences than others.

The same observation was obtained when we constructed a confusion matrix as shown in Table 3.7. It gives a clear idea about how many times the accent is incorrectly interpreted. We can also observe that i-vectors of which accents may be closer in total variability space. Maximum accuracy of 100% is observed for Turkish accent while minimum accuracy of 97.83% is obtained for the UK accent. Sometimes US and UK are observed to be getting confused with each other but African and Canadian are much better separated. This justifies the

Table 3.6: Average cosine distance between i-vectors of accent pairs

Accent-Accent	Average Cosine Distance
UK-UK	0.11
US-US	0.03
UK-US	-0.05
African-African	0.96
Canadian-Canadian	0.58
African-Canadian	0.08

Table 3.7: Confusion matrix for accent recognition experiments. Here, **Can.** represents Canadian, **Carib.** represents Caribbean, **Eth.** represents Ethiopian, **Un-id.** represents Unidentified

True ↓ /Predicted →	**African**	**Can.**	**Carib.**	**Eth.**	**Turkish**	**UK**	**US**	**Un-id.**
African	52	0	0	0	0	0	0	1
Can.	0	171	0	0	0	1	3	5
Carib.	0	0	103	0	0	0	1	0
Eth.	0	0	0	100	0	1	0	0
Turkish	0	0	0	0	100	0	0	0
UK	0	0	0	0	0	315	6	1
US	0	2	1	0	0	3	2173	2
Un-id.	0	0	0	0	0	0	0	242

difference between average cosine distances of UK-US (-0.05) and African-Canadian (0.08) accents.

3.4 CONCLUSION

In this work, we analyzed the effect of variation of size of the total variability matrix, T, for a given mixture count, on the accuracy of the i-vector system. We also investigated how the mixture count of UBM affects the accuracy while keeping the size of T matrix constant. We found that the T matrix size 50 along with mixture count of 32 or 128 shows good performance. This also helps to reduce computational costs. While studying the effect of length of test utterances on performance, we observed that the 10 sec length utterances produced the best results. However, 100% accuracy can be achieved with longer test utterances. While experimenting with the length of training utterances, it was observed that using 10 minutes seems the most useful option. Increasing the length of training utterances further does not provide much improvement in performance.

As in the previous problem, reducing the dimension of the total variability matrix and number of Gaussian mixtures needed for modeling directly influences the system time for speaker recognition due to involvement of numerous matrix operations. However, we measured the actual running time of experiments using state of art parameters (UBM size 2048, T 400) along with our own reduced parameters (UBM size 128, T 50). We were able to reduce the running time of i-vector extraction from 2 hours and 30 minutes to about 12 minutes which is quite significant.

We also performed accent recognition experiments for speakers from different geographical regions using i-vector approach. We obtained 99.63% accuracy for accent prediction task on 40 speakers, taken from the TED-LIUM database. We concluded that increasing the size of total variability space leads to increase in the accuracy of prediction. Further, accent recognition works better than simple

speaker recognition task, even when the test speaker was not present in the training data. And the i-vectors for speakers with similar accents were found closer to each other than those of speakers with differing accents. The results support the intuition for a two-pass system for speaker recognition using accent recognition as a first step. This will also reduce the number of models searched to predict the speaker identity provided that accent recognition performs correctly.

4

LEARNING CONTEXTUAL SPEAKER-RELATED REPRESENTATION FROM DATA

In the previous chapters, we looked at reducing the size of features and speaker models. After observing the outcomes of those experiments, we believe that there can be improvements at the feature stage in the speaker recognition system. As we have seen that it was possible to reduce the size of feature vectors without significantly compromising on the accuracy, it can be concluded that there was some information in the original feature vector that was not helping in the correct identification of speaker identities. Therefore, it is imperative that the feature set be improved such that it can capture important and discriminating information which help discern between different speakers.

However, designing a feature set to retain relevant information from the speech signal while discarding the unhelpful information has several issues associated with it. Foremost, the process requires hard work, expertise as well as domain and task-specific knowledge. These requirements can prove to be too constraining in many situations for speech and speaker recognition systems. Furthermore, a slight change in the task can force us to design a completely new feature set for the new problem. An alternate approach to generate new features automatically and attempt to learn the features from the speech data itself. Using this methodology has a number of advantages. It could alleviate the need for human expertise while also being comparatively better at speaker identity representation than their traditional counterpart. A lot of research in recent years have been focused toward achieving the goal of learning representations automatically from data [40, 76].

In this chapter, we present a Recurrent Neural Networks (RNNs) based Autoencoder (AE) architecture to learn features directly from data as an alternative to the traditional features such as the Mel Frequency Cepstral Coefficients (MFCCs).

Several studies in the literature have used both RNNs and AEs separately to good effect. Some of them have used AEs to derive intermediate representations of input data [50]. There have also been good results with different types of AEs in the past such as Denoising AE [72] for removing noise from images. AEs enable us to obtain a smaller representation of the input data such that we can regenerate the input from the smaller representation with minimal error, effectively acting as feature detectors.

RNNs [24], as described later in the chapter, are a type of neural nets that include the information from previous inputs with the current input. It is effectively a neural network that processes inputs through time steps and remembers what it has seen in previous time steps. These networks are effective when there is a time relation between the inputs. Time series, speech data, consecutive frames of videos are all such data that satisfy this property.

We use an RNN AE to leverage the benefits of both the architectures and combine their effectiveness to generate contextual representations of speaker-related data that may capture the context from before and after a particular frame of speech. This contextual information, from previous inputs, included with each frame provides us a better feature set.

There are studies that have used Deep Neural Networks including RNNs [25] directly to perform end-to-end speech recognition. In these cases, there is no separate calculation of features and modeling. Instead, the inputs (speech signals/utterances) are directly mapped to outputs (transcriptions). Our approach is different from such modeling methods in that we are using the RNN AE architecture to only generate a smaller representation of the input speech data. Without using the complete end-to-end pipeline of state-of-the-art methods, we can save computations involved in such speech recognition systems. Rather, we use a GMM-UBM system for modeling which is comparatively inexpensive and can provide good results for a speaker recognition task.

4.1 SPEAKER MODELING

Gaussian Mixture Models (GMMs) have been the mainstay of the speaker recognition and verification tasks for several years [57]. As introduced and discussed in chapter 2, the modeling approach involves using a finite number of Gaussian components that are hypothesized to represent speaker's characteristics. A GMM model can be represented as a weighted sum of M component Gaussian densities as follows:

$$p(x|\lambda) = \sum_{i=1}^{M} w_i g(x|\mu_i, \Sigma_i) \tag{4.1}$$

where x is a D-dimensional vector, w_i, $i = 1, 2, \ldots, M$ represent the mixture weights, and $g(x|\mu_i, \Sigma_i), i = 1, 2, \ldots, M$ define the component densities of the Gaussian mixture. Each individual component density is a D-variate Gaussian function which is defined as follows:

$$g(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu_i)'\Sigma_i^{-1}(x-\mu_i)\right\} \tag{4.2}$$

where μ_i is the mean vector for component i and Σ_i is the corresponding covariance matrix. The sum of all the weights must be equal to 1.

Universal Background Models (UBMs) [56] were proposed in efforts to improve the training time of models and the quality of the speaker models obtained. A UBM is speaker independent Gaussian mixture model which is trained from a large set of speakers to represent the general speech characteristics. The idea is to use the UBM for the adaptation of the individual speaker models with UBM parameters as priors. It was observed that this approach led to performance improvements in recognition experiments. The models obtained using UBMs are called Adapted GMMs.

Neural Networks are a brain-inspired idea of computation that have been used as a modeling technique for several difficult and often otherwise intractable problems. They have also been applied for modeling speech and speaker data and have had a fair amount of success. Neural networks consist of many interconnected individual units arranged in several layers. The first layer is called the input layer, the last layer is known as output layer and all the layers in between are termed as hidden layers. Each node computes an activation function and applies a non-linearity to squeeze the output into some particular range (usually -1 to 1 or 0 to 1) and passes the result to further layers for computation. A common example of a neural network architecture that has been successful in solving many problems is the Multi-layer Perceptron (MLP) [58]. Figure 4.1 shows a basic MLP architecture.

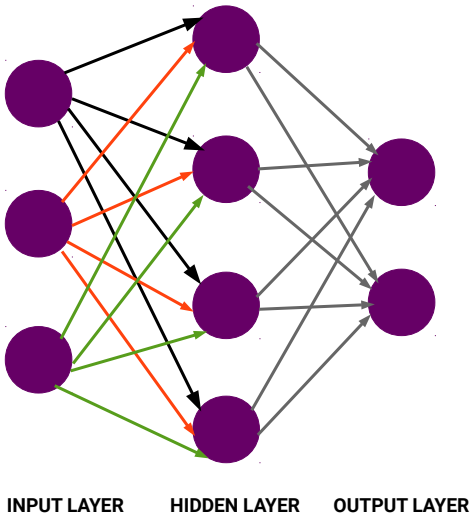

Figure 4.1: A typical Multi Layer Perceptron with 3 input nodes, 4 hidden nodes and 2 output nodes

Researchers then tried to extend the MLP architecture to more complicated and deeper networks using more number of units. But soon, it became infeasible to train these deeper networks quickly enough to be useful for important tasks. To solve this problem, new ways of training the models emerged and the field of deep learning came to the forefront of speech and speaker recognition research. By using clever techniques, it was possible to train several layers deep networks.

The traditional neural net architecture treated all the inputs given to it separately, with no relation between different inputs. This is problematic for data that is inherently time related. To resolve this issue and treat the different inputs as a sequence of data points, RNNs were proposed to capture time relationships among the input data. Figure 4.2 shows a typical RNN architecture and the internal state used to realize the time relationship between different inputs. The vanilla RNN being quite slow in practice, several clever modifications have now allowed researchers to practically utilize RNNs (and its faster variations) to model and predict time series type data. One such popular architecture is called the Long-Short Term Memory (LSTM) [29]. LSTM effectively uses its internal states to remember important things within the context of the input sequence. It maintains an internal state representation to understand and capture the time relation between inputs. We have used the LSTM model variant for our experiments to learn contextual features automatically from speech data.

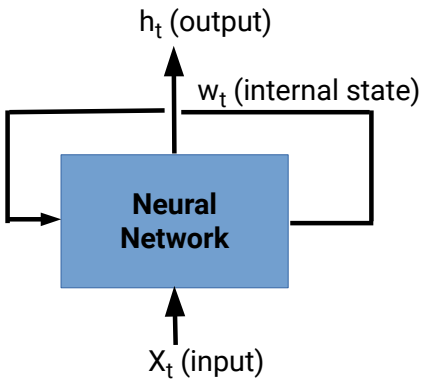

Figure 4.2: A typical representation of RNN model

AEs are a type of neural net architecture that tries to reconstruct the given input from its hidden layer representation. The hidden layer representation is generally quite smaller than the input size. This property allows them to effectively act as feature learners or detectors since the hidden representation which can generate the input back with minimal loss should be able to capture all the relevant information from the input. These architectures are one of the ways

that we can learn the features from the data itself without needing human expertise for explicitly devising task-specific features. This process of generating features implicitly is termed as feature learning or representation learning in the literature. Along similar directions, we propose an approach to learn features or representations for speaker-related information using RNNs (LSTMs) and AEs as an alternative approach to the very popular MFCCs feature set and other manually designed feature sets. Figure 4.3 shows our proposed architecture.

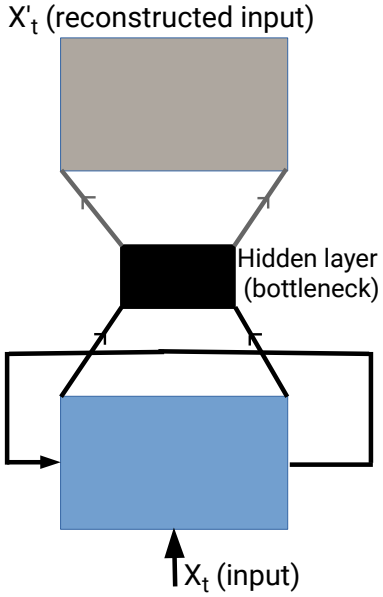

Figure 4.3: The proposed RNN (LSTM) AE model for feature learning

It is widely believed that it is vitally important to capture the spectral shape of the speech signal to effectively represent speech signals for processing to realize an accurate and robust recognition system. MFCCs [9] were devised in the 1980s as a way to capture and represent the spectral shape of the raw speech waveform. These representations were designed for short-term representation of the speech signal. Since their inception, they have slowly become the state-of-the-art features for almost every speech and speaker recognition system. Researchers have been trying to improve the MFCC feature set through various approaches, including some hybrid ones, in order to obtain features that can capture all the important aspects of speech signal while discarding away the noise or unhelpful portion. The search for such a feature set is still going on. Our method is another such effort to generate features that could be competitive or superior than the standard MFCCs for speech representation.

4.2 EXPERIMENTAL LAYOUT

4.2.1 *Data used*

For our experiments, we used a part of the TED-LIUM [59] dataset. The TED-LIUM dataset consists of 1495 TED talks recorded at a sample rate of 16000, 16 bit representation, with a bit rate of 256 K. The encoding used is 16-bit signed integer PCM. Out of the 1495 recordings, we used 50 of them in our experiments which amount to more than 40 hours of speech data. All files used were first manually pruned to remove all non-speech parts, which included music, advertisements and claps/cheering from the audience. Also, some files were musical performances. They were discarded and replaced with normal speech data.

For the experiments, the 50 files were taken and split into two parts: training & testing. For training, we used 5 minutes of the recording from each speaker. For the testing part, we used utterances of 0.5 seconds in length. For each speaker, 50 test cases were used.

4.2.2 *Feature learning*

We used an RNN (LSTM) AE setup, described in previous section, for learning features for the input speech data from each speaker. For implementation, we used the TensorFlow [1] library in Python. We feed the frames of speech as input. For capturing context information from the input signal, we use a context of 7 previous and 7 next frames. This context enables the RNN to use the neighborhood of a particular portion of the signal to better approximate and estimate the hidden layer. The output from the hidden layer is taken and reconstructed at the output layer. The square of the difference between the input and the reconstructed signal defines the loss function for the experiment. We train the model to minimize the loss value. When the network is trained, we obtain the hidden values such that the reconstruction of the signal leads to minimal loss. These hidden values can then be utilized as features or representations for speaker-related information for the input speaker data. Using the trained network, we then compute the features for all the speakers. One of the major advantages with this approach is better approximation of the hidden values that serve as features through the use of context information of neighborhood frames.

In order to determine the most appropriate size of feature set learned through the RNN (LSTM) AE system, we performed experiments with seven (7) different configurations. We experimented with 5, 10, 15, 20, 25, 30, 35 and 40 hidden units. Equivalently, these configurations represent the sizes of features or representations learned. With these experiments, we could come up with a reasonable configuration

for learning the most appropriate sized feature set. We list the results of these experiments in the next section.

4.2.3 *Speaker Modeling*

The computed features from the previous step are then fed into the GMM-UBM system to generate speaker models. First, a generalized UBM is trained on all data. Next, speaker models are adapted from the UBM to generate individual speaker models. The models consist of the means of Gaussian components, obtained by adapting UBM means using the speaker specific data for individual speakers. During the testing phase, we compute the probability of each of the utterances, given the speaker models and the speaker model giving the highest probability value is predicted to be the true speaker.

4.3 RESULTS AND DISCUSSION

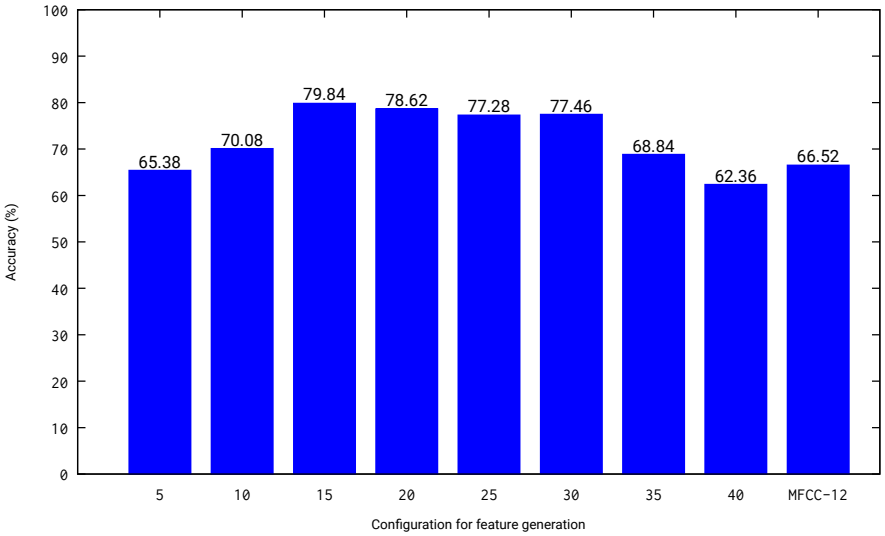

Figure 4.4: ASR performance with different feature sets. In the graph, **5, 10, . . . , 40** represents RNN (LSTM) AE architecture with 5, 10, . . . , 40 hidden units respectively. **MFCC-12** represents standard MFCC feature set of 12 dimension.

Figure 4.4 shows the results of the experiments. We have compared the proposed method of representation or feature learning with the traditional MFCC feature set of 12 dimension. Further, we have used various configurations of the RNN (LSTM) AE setup to learn representations of different sizes. The training and testing data used for all the experiments is kept same and is described in section 4.2.1. As we can see from the figure 4.4, the overall average accuracy of speaker recognition system using the proposed RNN (LSTM) AE method for feature learning outperforms the system using MFCC feature set in

Table 4.1: ASR performance with different RNN (LSTM) AE configurations for learning features and with standard MFCC feature set

Configuration	Feat. Size	Accuracy (%)
RNNAE-5	5	65.38
RNNAE-10	10	70.08
RNNAE-15	15	79.84
RNNAE-20	20	78.62
RNNAE-25	25	77.28
RNNAE-30	30	77.46
RNNAE-35	35	68.84
RNNAE-40	40	62.36
MFCC-12	12	66.52

5 of the 7 configurations. The average accuracy for the best configuration of the RNN (LSTM) AE system is 79.84% while it is 66.52% for the MFCC system. We also see that as we increase the size of the learned feature set, the accuracy only increases up to size 15. After that, it stays almost same or even deteriorates in some configurations. This indicates that choosing a bigger set of features may end up confusing the system rather than improving the accuracy proportionately. We can therefore infer that using a feature set of size 15 seems like a reasonable option for the given task of speaker recognition. The complete results of all experiments are listed in Table 4.1.

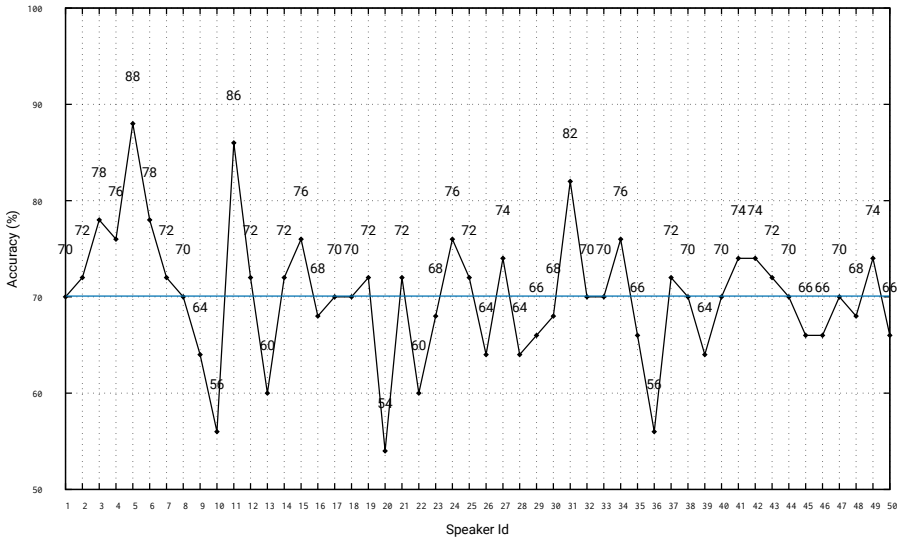

Figure 4.5: ASR performance with RNNAE-10 configuration for individual speakers

There are a few additional observations from the experimental results, however, that merit the reader's attention. We can observe from figure 4.5 that the accuracy for individual speakers varies quite a bit. On further analysis of some of the best and worst performing speakers, we found that the differences may be attributable to at least the accent of the speakers. The best performing speakers were found to be native speakers of English language. Since the pool of speakers used for the experiments contained a higher proportion of English speaking population, the model seems to have captured the properties of such speakers. This may have led to a bias toward the native English speakers and against non-native speakers as we found that most of the worst performing speakers were not from natively English speaking regions. In order to fix this problem, we need to have a database that has a balanced set of speakers from a wide range of regions. This would ensure that all the speakers' characteristics can be captured correctly by the GMM-UBM model and the biasing could be reduced to a minimum. From a technical perspective, we can imagine another solution for this problem. We can deliberately choose a base accent for constructing a model. From there onward, we can adapt the base model for other specific accents using data from respective speakers, from different regions. This work can form the extension of the current study.

Another interesting observation is the issue of incorrect test cases occurring in both high and low performing speakers. Since we did not manually extract the recordings for voiced regions, it is expected that there may be some test cases where there may not be sufficient voiced activity to detect the speakers correctly since the test cases were deliberately taken to be very short (0.5 s). To alleviate this problem, we can alter the way to construct our test cases in further experiments. Instead of choosing for fixed length test cases, we can specifically choose voiced regions using a robust criteria for voice activity detection (VAD). This type of use case is much more practical for usage of speaker recognition systems in practice as we cannot expect the users to speak at a predefined rate and for a specific amount of time. There are several VAD techniques present in literature that can give near perfect voiced regions.

4.4 CONCLUSION

From the results and discussions of the experiments, it is apparent that learning features or representations of speaker-related information directly from data instead of using human expertise to derive features can be highly beneficial. Besides saving resources in terms of expertise, it is indicated by experimental results for ASR task that they outperform the traditional MFCC feature set on average. As per the experimental results, it can be concluded that using a feature set of

size 15 can be beneficial for ASR experiments. Also, using the GMM-UBM architecture to model the speakers using the learned representation or features can lead to good performance without incurring the heavy computations involved in deep architectures used in end-to-end recognition systems.

It is further realized through the experiments that there is an urgent need for balanced databases that contain speakers from a wide range of geographical regions to ensure that sufficient variability of accents are accounted for in the model. The availability of such data can significantly impact the performance of ASR systems in practice. This also presents us with a technical challenge of adapting systems for different accent-specific conditions from a base set of speakers. Using such an approach, we can, to a large extent, compensate for an unbalanced database of speakers. However, such methods are not ubiquitous in the literature and need to be experimented and refined to suit ASR tasks.

5

CONCLUSIONS AND FUTURE WORK

In this chapter, we present the conclusions of the work carried out within the scope of this thesis. Experiments aimed at reducing computations involved in Automatic Speaker Recognition (ASR) systems through reductions in size of feature vectors and speaker models were performed. Significant improvements were observed with reductions at both feature and model level. Further, on observing scope of improvement in the MFCC feature set, a new method to learn features or representations for speaker-related information was proposed. It was observed that learning representations from the speech data can provide good accuracy for speaker recognition tasks. The next sections describe these experiments briefly and then list some of the possible future directions of research.

5.1 REDUCTION AT FEATURE LEVEL

In chapter 2, we looked at reducing the size of feature vectors by processing the raw features. These processed features were then used to build speaker models that were used in the recognition step where we estimate the probability of the test utterance coming from one of the speakers.

In the experiments, we investigated how the performance of speaker recognition system varies with respect to reduction in size of the feature set. These experiments were designed to help evaluate the performance at different sizes of the feature vector. With such trends available, it may be possible to design systems that can provide similar performance levels but at a lesser cost in terms of time and computation since a smaller feature vector implies faster running times and lower computational costs. It is also possible to design systems specifically tailored for low resource devices that cannot afford to perform complex calculations needed for state-of-the-art methods.

We first used the Naïve dropping technique for reducing the feature set size. It provided promising results but there was no legitimate way to determine the rank of the features with this method as the order of the features itself is taken to be the rank of the feature in the feature vector. Therefore, the F-ratio method was tested to systematically decide on the merit of each feature. Next, feature re-extraction methods were tested for feature reduction on the data.

First PCA was applied for feature reduction. It was observed that approximately 10% performance was sacrificed while achieving about 50% reduction in the size of feature vector. Using LDA as the feature

re-extraction technique, however, with the same amount of performance deterioration, the feature set size was reduced to a higher extent than what was achieved with PCA. Hence, in this view, LDA would be considered as the preferred method against PCA. In case of Factor Analysis, it was observed that it was possible to reduce the feature set size to more than 40% with minimal performance degradation. Nonlinear feature re-extraction methods viz. KPCA and ISOMAP, also demonstrated similar performance trends. With KPCA, it was possible to reduce the feature set size by approximately 50% with a minimal performance sacrifice, with performance even increasing at some lower dimensions. This can be concluded from the graphs presented above. With ISOMAP, it was again observed that it is possible to reduce the feature set by 50% while obtaining an overall higher performance by approximately 30%. The best performance was observed with feature set of size 10 with ISOMAP technique. However, all re-extraction methods incurred an overhead when the transformation of the test feature vector is applied during the testing phase as each test vector has to multiplied by a loading matrix.

Overall it can be concluded that using either re-extraction or selection techniques, it is possible to provide significant reductions in feature vector size which can directly affect the response time of the speaker recognition system. These reductions can lead to reduced processing time and thus enable us to provide speaker recognition services on low resources devices themselves. For example, if we consider the original dimension of MFCC feature set to be 12 and the reduced dimension to be 6. For calculation of component density of one Gaussian, for a single frame of speech, $O(n^{11})$ time steps are needed approximately where n is size of the feature. When original feature size of 12 is used, it comes to approximately 7.43×10^{11} time steps. For the reduced feature of size 6, the same evaluates to 6.5×10^{9} time steps. Hence we see a distinct reduction in the number of time steps needed for evaluating a single Gaussian density for one frame. As per the calculation, this will lead to a significant improvement for 32 mixture Gaussian with approximately 100 frames for a complete test utterance.

5.2 REDUCTION AT MODEL LEVEL

In chapter 3, we discussed the experiments with i-vectors and the reductions obtained at the model level. The high dimensional speaker models were reduced to a lower dimension to make the speaker recognition faster and cheaper in terms of computation. The configurations for model generation were tuned using the results of the experiments performed.

In these experiments, we analyzed the effect of variation of size of the total variability matrix, T, for a given mixture count, on the ac-

curacy of the i-vector system. We also investigated how the mixture count of UBM affects the accuracy while keeping the size of T matrix constant. We found that the T matrix size 50 along with mixture count of 32 or 128 shows good performance. This also helps to reduce computational costs. While studying the effect of length of test utterances on performance, we observed that the 10 sec length utterances produced the best results. However, 100% accuracy can be achieved with longer test utterances. While experimenting with the length of training utterances, it was observed that using 10 minutes seems the most useful option. Increasing the length of training utterances further does not provide much improvement in performance.

As in the case with reduction at feature level, reducing the dimension of the total variability matrix and number of Gaussian mixtures needed for speaker modeling directly influences the system time for speaker recognition due to involvement of numerous matrix operations. However, for these experiments, we measured the actual running time of experiments using state of art parameters (UBM size 2048, T 400) along with our own reduced parameters (UBM size 128, T 50). We were able to reduce the running time of i-vector extraction from 2 hours and 30 minutes to about 12 minutes which is quite significant. Similar improvements were observed at every stage of the experimental pipeline.

We also performed accent recognition experiments for speakers from different geographical regions using i-vector approach. We obtained 99.63% accuracy for accent prediction task on 40 speakers, taken from the TED-LIUM database. We concluded that increasing the size of total variability space leads to increase in the accuracy of prediction. Further, accent recognition works better than speaker recognition task, even when the test speaker was not present in the training data. And the i-vectors for speakers with similar accents were found closer to each other than those of speakers with differing accents. The results support the intuition for a two-pass system for speaker recognition using accent recognition as a first step. This will also reduce the number of models searched to predict the speaker identity, provided that accent recognition performs correctly.

5.3 LEARNING REPRESENTATIONS FOR SPEAKER-RELATED INFORMATION AUTOMATICALLY FROM DATA

After observing the outcomes of experiments in chapters 2 and 3, we hypothesized that there can be improvements at the feature stage in the automatic speaker recognition system. As we have seen that it was possible to reduce the size of feature vectors without significantly compromising on the accuracy, it can be concluded that there was some information in the original feature vector that was not helping in the correct identification of speaker identities. Therefore, it is im-

perative that the feature set be improved such that it can capture important and discriminating information which help discern between different speakers. This direction was the topic of investigation in chapter 4.

However, designing a feature set, from scratch, to retain important information from the speech signal while discarding the useless data has several problems associated with it. Foremost, the process requires hard work, expertise as well as domain and task-specific knowledge. These requirements can prove to be too constraining in many situations. Furthermore, a small change in the task can force us to completely re-design a new feature set for the modified problem. Alternatively, we can try to generate new features automatically and attempt to learn the features from the speech data itself. Using this methodology has a number of advantages. It could alleviate the need for human expertise while also being comparatively better at speaker identity representation than their traditional counterparts. A lot of research in recent years have been focused toward achieving the goal of learning representations automatically from data [40, 76].

In this chapter, we discussed a Recurrent Neural Networks (RNNs) based Autoencoder (AE) architecture to learn features directly from data as an alternative to the traditional features such as the Mel Frequency Cepstral Coefficients (MFCCs). We have used the LSTM model variant for our experiments to learn contextual features automatically from speech data.

From the results of the experiments, it is apparent that learning features or representations of speaker-related information directly from data instead of using human expertise to derive features can be highly beneficial. Besides saving resources in terms of expertise, it is indicated by experimental results for ASR task that they outperform the traditional MFCC feature set on average. It may also be concluded that using a feature set of size 15 can be beneficial for ASR experiments. Also, using the GMM-UBM architecture to model the speakers using the learned representation or features can lead to good performance without incurring the heavy computations involved in deep architectures used in end-to-end recognition systems.

It is further realized through the experiments that there is an urgent need for balanced databases that contain speakers from a wide range of geographical regions to ensure that sufficient variability of accents are accounted for in the model. The availability of such data can significantly impact the performance of ASR systems in practice. This also presents a challenge of adapting systems for different accent-specific conditions from a base set of speakers. Using such an approach, it may be possible, to a large extent, to compensate for an unbalanced database of speakers.

5.4 FUTURE RESEARCH DIRECTIONS

The following avenues could be explored as part of the future work of the thesis:

- Experiments performed for reducing the feature set size can be tested thoroughly by implementing them on actual low-resource devices for providing on-device speaker recognition services. The study can further be extended by including the device microphone itself for capturing the test utterances.
- Besides using practical configurations of the i-vector method to obtain smaller speaker models, research may be conducted to improve the method used to generate the model so that better speaker models can be derived.
- Other sophisticated methods may be employed for learning speaker-related representations that can enable more accurate speaker recognition systems.
- Speech data is needed for any and every experiment. It is important that several databases for speech are created for different speech tasks. The databases need to be created with care so as to be useful in their intended target experiments such as speech recognition, speaker recognition, accent recognition, speaker diarization, language recognition, etc.

5.5 DESIRABLE POINTS OR QUESTIONS RAISED BY THESIS EXAMINER

- **Question:** While the models for speaker identification based on speech extraction are well discussed, the features themselves needed to be mentioned in the automated method of feature extraction. Are they irrelevant for the present study?

  Response: While it is certainly possible to specify and emulate the extraction of a specific type of feature set, we did not work on this aspect intentionally. Our idea was to let the method automatically settle on to those representations that were best able to regenerate the original data points (speech data).

- **Question:** What are the relative advantages of the different extraction techniques with regard to speaker variation and noise variation?

  Response: While MFCCs have been found to be most useful feature extraction technique overall, different techniques may provide better performance for different speakers. Therefore, it may be possible to use different feature extraction techniques

for analysis of different speakers. Furthermore, this approach may be beneficial for speaker verification experiments where we claim a speaker identity and verify from the system.

During the course of the present study, we made extensive use of the TED LIUM database of speech data. This database consists of 1495 Ted talks recorded in very clear environments with minimal noise. Hence, we cannot comment about the advantages of different extraction techniques in regard to noise variation at the moment. However, this is an interesting direction that we can take up in future research. For this purpose, we can either manually introduce noise in clean speech database or we can work directly on existing noisy speech databases.

APPENDIX

In this appendix, we present the rest of the graphs for the results of experiments conducted in Chapter 3.

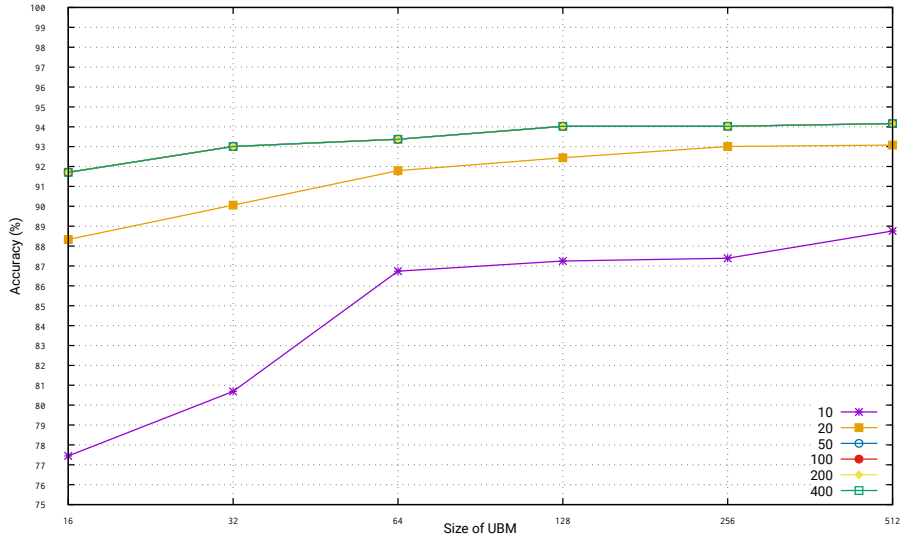

Figure A.1: Performance with 10 sec test utterances for different T sizes

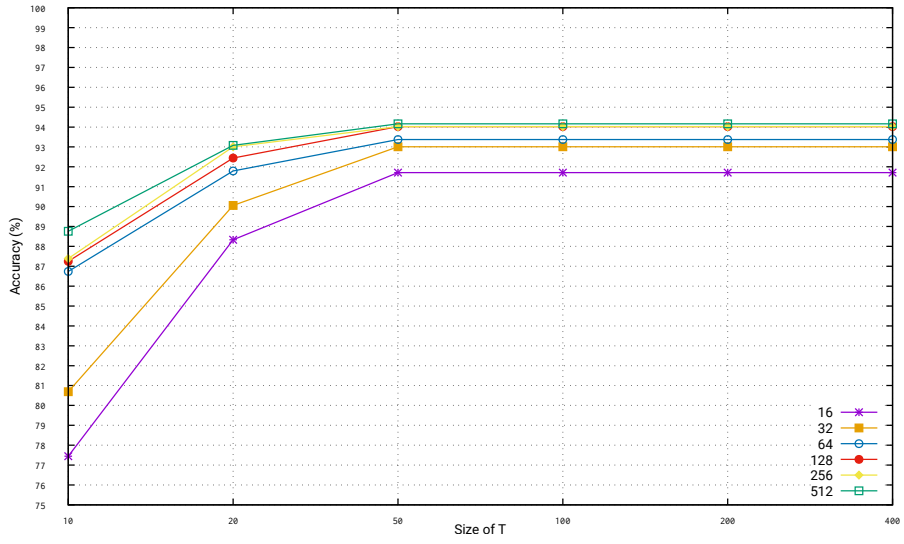

Figure A.2: Performance with 10 sec test utterances for different UBM sizes

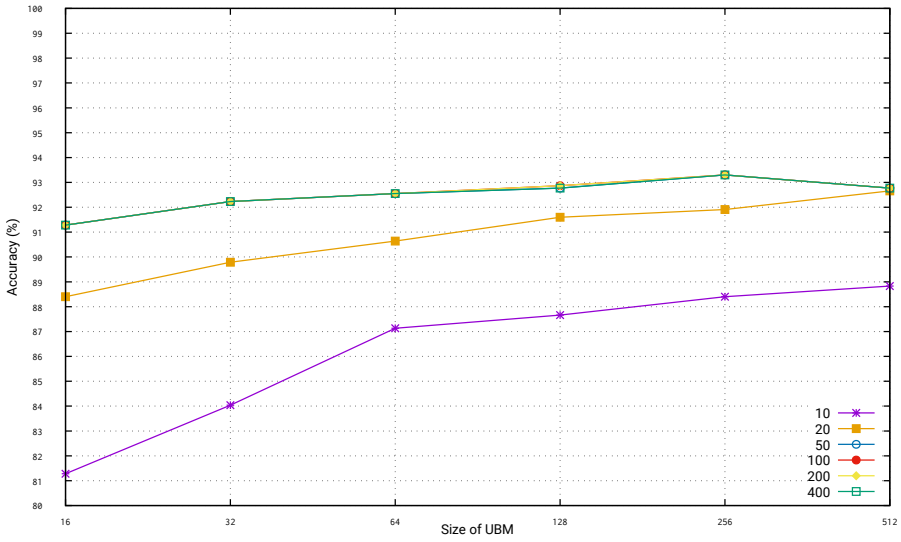

Figure A.3: Performance with 15 sec test utterances for different T sizes

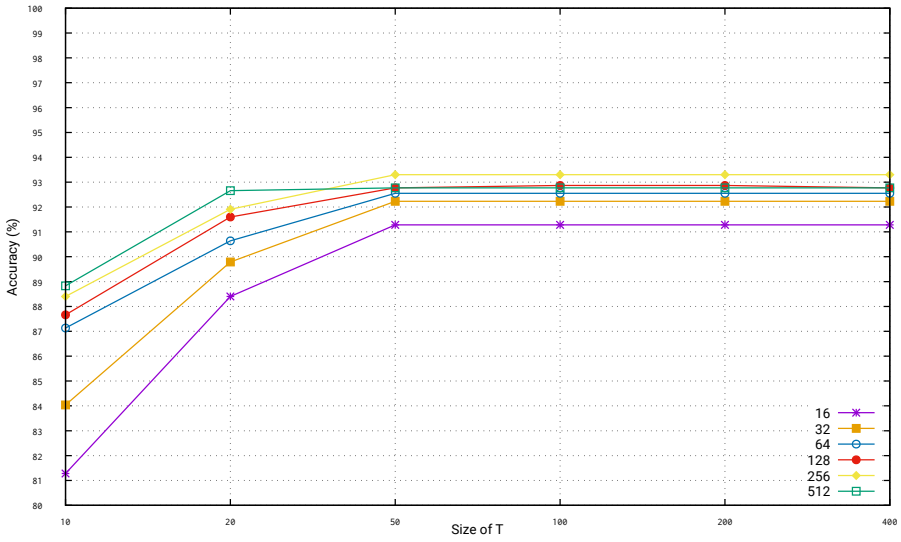

Figure A.4: Performance with 15 sec test utterances for different UBM sizes

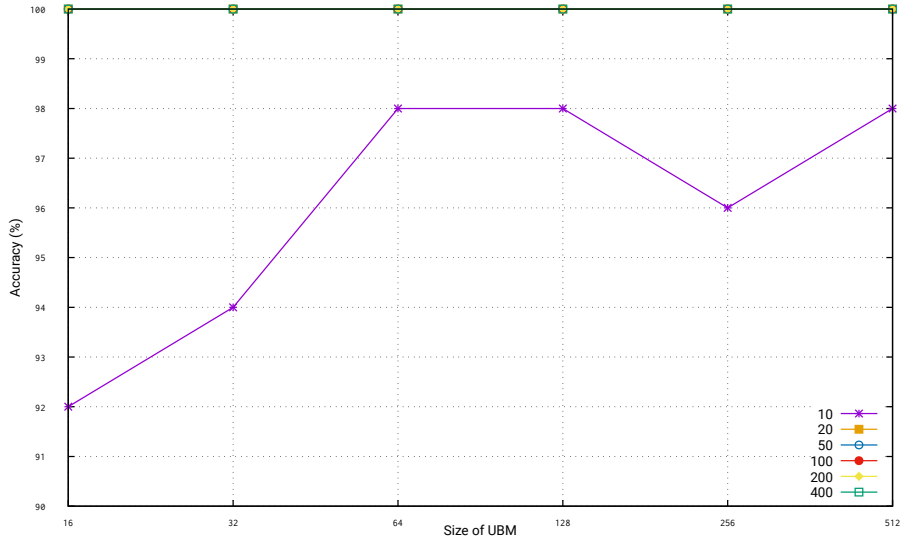

Figure A.5: Performance with full test files utterances for different T sizes

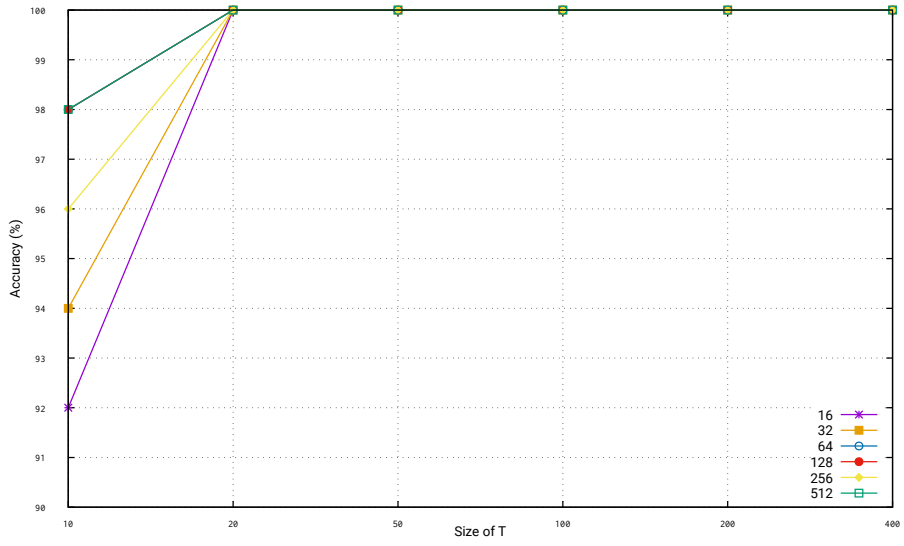

Figure A.6: Performance with full test files utterances for different UBM sizes

BIBLIOGRAPHY

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. "TensorFlow: A System for Large-Scale Machine Learning." In: *OSDI*. 2016, pp. 265–283.

[2] H. Aronowitz and O. Barkan. "Efficient approximated i-vector extraction." In: *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2012, pp. 4789–4792.

[3] B. S. Atal. "Automatic Speaker Recognition Based on Pitch Contours." In: *The Journal of the Acoustical Society of America* 52.6B (1972), pp. 1687–1697.

[4] S. Biswas, J. Rohdin, and K. Shinoda. "i-Vector Selection for Effective PLDA Modeling in Speaker Recognition." In: *Proc. Odyssey Workshop*. 2014, pp. 100–105.

[5] R. H. Bolt, F. S. Cooper, E. E. David, P. B. Denes, J. M. Pickett, and K. N. Stevens. "Speaker Identification by Speech Spectrograms: A Scientists' View of its Reliability for Legal Purposes." In: *The Journal of the Acoustical Society of America* 47.2B (1970), pp. 597–612.

[6] E. C. Cherry and W. K. Taylor. "Some Further Experiments upon the Recognition of Speech, with One and with Two Ears." In: *The Journal of the Acoustical Society of America* 26.4 (1954), pp. 554–559.

[7] D. Chow and W. H. Abdulla. "Speaker identification based on log area ratio and gaussian mixture models in narrow-band speech." In: *PRICAI 2004: Trends in Artificial Intelligence*. 2004, pp. 901–908.

[8] C. Cortes and V. Vapnik. "Support-vector networks." In: *Machine learning* 20.3 (1995), pp. 273–297.

[9] S. Davis and P. Mermelstein. "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences." In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28.4 (1980), pp. 357–366.

[10] N. Dehak, Z. N. Karam, D. A. Reynolds, R. Dehak, W. M. Campbell, J. R. Glass, et al. "A channel-blind system for speaker verification." In: *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2011, pp. 4536–4539.

[11] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. "Front-end factor analysis for speaker verification." In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.4 (2011), pp. 788–798.

[12] N. Dehak, P. A. Torres-Carrasquillo, D. A. Reynolds, and R. Dehak. "Language Recognition via i-vectors and Dimensionality Reduction." In: *INTERSPEECH*. 2011, pp. 857–860.

[13] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm." In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1977), pp. 1–38.

[14] P. Denes and M. V. Mathews. "Spoken Digit Recognition Using Time-Frequency Pattern Matching." In: *The Journal of the Acoustical Society of America* 32.11 (1960), pp. 1450–1455.

[15] G. R. Doddington. "Speaker recognition–Identifying people by their voices." In: *Proceedings of the IEEE* 73.11 (1985), pp. 1651–1664.

[16] K. R. Farrell, R. J. Mammone, and K. T. Assaleh. "Speaker recognition using neural networks and conventional classifiers." In: *IEEE Transactions on Speech and Audio Processing* 2.1 (1994), pp. 194–205.

[17] I. K. Fodor. *A survey of dimension reduction techniques*. 2002.

[18] J. W. Forgie and C. D. Forgie. "Results Obtained from a Vowel Recognition Computer Program." In: *The Journal of the Acoustical Society of America* 31.11 (1959), pp. 1480–1489.

[19] D. B. Fry. "Theoretical aspects of mechanical speech recognition." In: *Journal of the British Institution of Radio Engineers* 19.4 (1959), pp. 211–218.

[20] S. Furui. "Comparison of speaker recognition methods using statistical features and dynamic features." In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 29.3 (1981), pp. 342–350.

[21] S. Furui. "Research of individuality features in speech waves and automatic speaker recognition techniques." In: *Speech communication* 5.2 (1986), pp. 183–197.

[22] O. Glembek, L. Burget, P. Matě, M. Karafiát, and P. Kenny. "Simplification and optimization of i-vector extraction." In: *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2011, pp. 4516 –4519.

[23] J. W. Glenn and N. Kleiner. "Speaker Identification Based on Nasal Phonation." In: *The Journal of the Acoustical Society of America* 43.2 (1968), pp. 368–372.

[24] A. Graves. "Supervised Sequence Labelling." In: *Supervised Sequence Labelling with Recurrent Neural Networks* (2012).

[25] A. Graves, A. R. Mohamed, and G. Hinton. "Speech recognition with deep recurrent neural networks." In: *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. 2013, pp. 6645–6649.

[26] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, et al. "DeepSpeech: Scaling up end-to-end speech recognition." In: *arXiv preprint arXiv:1412.5567* (2014).

[27] W. A. Hargreaves and J. A. Starkweather. "Recognition of Speaker Identity." In: *Language and Speech* 6.2 (1963), pp. 63–67.

[28] H. Hermansky. "Perceptual linear predictive (PLP) analysis of speech." In: *The Journal of the Acoustical Society of America* 87.4 (1990), pp. 1738–1752.

[29] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory." In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[30] A. Honkela, S. Harmeling, L. Lundqvist, and H. Valpola. *Using kernel PCA for initialization of nonlinear factor analysis*. Tech. rep. 2003.

[31] H. Hu and S. A. Zahorian. "Dimensionality reduction methods for HMM phonetic recognition." In: *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. 2010, pp. 4854–4857.

[32] Y. Jiang, K. A. Lee, Z. Tang, B. Ma, A. Larcher, and H. Li. "PLDA Modeling in I-Vector and Supervector Space for Speaker Verification." In: *INTERSPEECH*. 2012.

[33] S. M. Kay and S. L. Marple. "Spectrum analysis–a modern perspective." In: *Proceedings of the IEEE* 69.11 (1981), pp. 1380–1419.

[34] P. Kenny, G. Boulianne, and P. Dumouchel. "Eigenvoice modeling with sparse training data." In: *IEEE Transactions on Speech and Audio Processing* 13.3 (2005), pp. 345–354.

[35] P. Kenny and P. Dumouchel. "Disentangling speaker and channel effects in speaker verification." In: *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Vol. 1. 2004, pp. I–37.

[36] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel. "Joint factor analysis versus eigenchannels in speaker recognition." In: *IEEE Transactions on Audio, Speech, and Language Processing* 15.4 (2007), pp. 1435–1447.

[37] P. Kenny, T. Stafylakis, P. Ouellet, M. J. Alam, and P. Dumouchel. "PLDA for speaker verification with utterances of arbitrary duration." In: *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2013, pp. 7649–7653.

[38] L. G. Kersta. "Speaker Recognition and Identification by Voiceprints." In: *Conn. BJ* 40 (1966), p. 586.

[39] A. Larcher, J. F. Bonastre, B. G. B. Fauve, K. A. Lee, C. Lévy, H. Li, J. S. D. Mason, and J. Y. Parfait. "ALIZE 3.0-open source toolkit for state-of-the-art speaker recognition." In: *INTERSPEECH*. 2013, pp. 2768–2772.

[40] H. Lee, P. Pham, Y. Largman, and A. Y. Ng. "Unsupervised feature learning for audio classification using convolutional deep belief networks." In: *Advances in Neural Information Processing Systems*. 2009, pp. 1096–1104.

[41] K. P. Li and J. E. Porter. "Normalizations and selection of speech segments for speaker recognition scoring." In: *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. 1988, pp. 595–598.

[42] K. Li and E. Wrench. "An approach to text-independent speaker recognition with short utterances." In: *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Vol. 8. 1983, pp. 555–558.

[43] M. Li and W. Liu. "Speaker verification and spoken language identification using a generalized i-vector framework with phonetic tokenizations and tandem features." In: *submitted to INTERSPEECH* (2014).

[44] Q. Liu, A. H. Sung, and M. Qiao. "Temporal derivative-based spectrum and mel-cepstrum audio steganalysis." In: *IEEE Transactions on Information Forensics and Security* 4.3 (2009), pp. 359–368.

[45] J. E. Luck. "Automatic Speaker Verification Using Cepstral Measurements." In: *The Journal of the Acoustical Society of America* 46.4B (1969), pp. 1026–1032.

[46] M. I. Mandasari, M. McLaren, D. Van Leeuwen, et al. "The effect of noise on modern automatic speaker recognition systems." In: *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2012, pp. 4249–4252.

[47] J. Markel and S. Davis. "Text-independent speaker recognition from a large linguistically unconstrained time-spaced database." In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 27.1 (1979), pp. 74–82.

[48] J. Markel, B. Oshika, and A. Gray. "Long-term feature averaging for speaker recognition." In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 25.4 (1977), pp. 330–337.

[49] D. Martınez, O. Plchot, L. Burget, O. Glembek, and P. Mate. "Language recognition in i-vectors space." In: *Proceedings of INTERSPEECH* (2011), pp. 861–864.

[50] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber. "Stacked convolutional auto-encoders for hierarchical feature extraction." In: *International Conference on Artificial Neural Networks*. 2011, pp. 52–59.

[51] T. Matsui and S. Furui. "Text-independent speaker recognition using vocal tract and pitch information." In: *First International Conference on Spoken Language Processing*. 1990.

[52] S. K. Pal and D. D. Majumder. "Fuzzy sets and decision making approaches in vowel and speaker recognition." In: *IEEE Transactions on Systems, Man, and Cybernetics* 7.8 (1977), pp. 625–629.

[53] K. K. Paliwal. "Dimensionality reduction of the enhanced feature set for the HMM-based speech recognizer." In: *Digital Signal Processing* 2.3 (1992), pp. 157–173.

[54] C. Quan, D. Wan, B. Zhang, and F. Ren. "Reduce the dimensions of emotional features by principal component analysis for speech emotion recognition." In: *IEEE/SICE International Symposium on System Integration (SII)*. 2013, pp. 222–226.

[55] L. R. Rabiner. "A tutorial on hidden Markov models and selected applications in speech recognition." In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286.

[56] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. "Speaker verification using adapted Gaussian mixture models." In: *Digital signal processing* 10.1-3 (2000), pp. 19–41.

[57] D. A. Reynolds and R. C. Rose. "Robust text-independent speaker identification using Gaussian mixture speaker models." In: *IEEE Transactions on Speech and Audio Processing* 3.1 (1995), pp. 72 –83.

[58] F. Rosenblatt. *Principles of neurodynamics. Perceptrons and the theory of brain mechanisms*. Tech. rep. Cornell Aeronautical Lab Inc Buffalo NY, 1961.

[59] A. Rousseau, P. Deléglise, and Y. Esteve. "TED-LIUM: an Automatic Speech Recognition dedicated corpus." In: *LREC*. 2012, pp. 125–129.

[60] M. Sambur. "Speaker recognition using orthogonal linear prediction." In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 24.4 (1976), pp. 283–289.

[61] A. K. Sarkar, D. Matrouf, P. M. Bousquet, and J. F. Bonastre. "Study of the Effect of I-vector Modeling on Short and Mismatch Utterance Duration for Speaker Verification." In: *INTERSPEECH*. 2012.

[62] M. R. Schroeder. "Similarity Measure for Automatic Speech and Speaker Recognition." In: *The Journal of the Acoustical Society of America* 43.2 (1968), pp. 375–377.

[63] S. Sharma, M. Kumar, and P. K. Das. "A technique for dimension reduction of MFCC spectral features for speech recognition." In: *International Conference on Industrial Instrumentation and Control (ICIC)*. 2015, pp. 99–104.

[64] J. N. Shearme and J. N. Holmes. "An Experiment Concerning the Recognition of Voices." In: *Language and Speech* 2.3 (1959), pp. 123–131.

[65] K. A. Sheela and K. S. Prasad. "Linear Discriminant Analysis F-Ratio for Optimization of TESPAR & MFCC Features for Speaker Recongnition." In: *Journal of Multimedia* 2.6 (2007), pp. 34 –43.

[66] S. Slomka, P. Castellano, P. Barger, S. Sridharan, and V. L. Narasimhan. "A comparison of Gaussian mixture and multiple binary classifier models for speaker verification." In: *Australian and New Zealand Conference on Intelligent Information Systems*. 1996, pp. 316–319.

[67] F. K. Soong, A. E. Rosenberg, B. H. Juang, and L. R. Rabiner. "Report: A vector quantization approach to speaker recognition." In: *Bell Labs Technical Journal* 66.2 (1987), pp. 14–26.

[68] K. N. Stevens. "Toward a Model for Speech Recognition." In: *The Journal of the Acoustical Society of America* 32.1 (1960), pp. 47–55.

[69] D. D. Suhr. "Principal component analysis vs. exploratory factor analysis." In: *SUGI 30 Proceedings* (2005), pp. 203–230.

[70] J. B. Tenenbaum, V. De Silva, and J. C. Langford. "A global geometric framework for nonlinear dimensionality reduction." In: *Science* 290.5500 (2000), pp. 2319–2323.

[71] P. Verma and P. K Das. "I-Vectors in speech processing applications: a survey." In: *International Journal of Speech Technology* 18.4 (2015), pp. 529–546.

[72] P. Vincent, H. Larochelle, Y. Bengio, and P. A. Manzagol. "Extracting and composing robust features with denoising autoencoders." In: *Proceedings of the 25th International Conference on Machine learning*. 2008, pp. 1096–1103.

[73] J. J. Wolf. "Acoustic Measurements for Speaker Recognition." In: *The Journal of the Acoustical Society of America* 46.1A (1969), pp. 89–90.

[74] J. J. Wolf. "Efficient Acoustic Parameters for Speaker Recognition." In: *The Journal of the Acoustical Society of America* 51.6B (1972), pp. 2044–2056.

[75] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. *The {HTK} Book Version 3.0*. Cambridge University Press, 2000.

[76] D. Yu, M. L. Seltzer, J. Li, J. T. Huang, and F. Seide. "Feature learning in deep neural networks-studies on speech recognition tasks." In: *arXiv preprint arXiv:1301.3605* (2013).