Phonetic aware techniques for Speaker Verification

THÈSE N^{O} 8886 (2018)

PRÉSENTÉE LE 14 NOVEMBRE 2018

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
LABORATOIRE DE L'IDIAP
PROGRAMME DOCTORAL EN GÉNIE ÉLECTRIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Subhadeep DEY

acceptée sur proposition du jury:

Dr A. Schmid, président du jury
Prof. H. Bourlard, Dr P. Motlicek, directeurs de thèse
Dr H. Cernocky, rapporteur
Prof. J.-F. BONASTRE, rapporteur
Prof. J.-Ph. Thiran , rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2018

Abstract

The goal of this thesis is to improve current state-of-the-art techniques in speaker verification (SV), typically based on “identity-vectors” (*i-vectors*) and *deep neural network (DNN)*, by exploiting diverse (phonetic) information extracted using various techniques such as *automatic speech recognition (ASR)*. Different speakers span different subspaces within a universal acoustic space, usually modelled by “universal background model”. The speaker-specific subspace depends on the speaker’s voice characteristics, but also on the verbalised text of a speaker. In current state-of-the-art SV systems, *i-vectors* are extracted by applying a factor analysis technique to obtain low dimensional speaker-specific representation. Furthermore, DNN output is also employed in a conventional i-vector framework to model phonetic information embedded in the speech signal. This thesis proposes various techniques to exploit phonetic knowledge of speech to further enrich speaker characteristics.

More specifically, the techniques proposed in this thesis are applied to various SV tasks, namely, text-independent and text-dependent SV. For text-independent SV task, several ASR systems are developed and applied to compute phonetic posterior probabilities, subsequently exploited to enhance the speaker-specific information included in i-vectors. These approaches are then extended for text-dependent SV task, exploiting temporal information in a principled way, i.e., by using dynamic time warping applied on speaker informative vectors.

Finally, as opposed to training DNN with phonetic information, DNN is trained in an end-to-end fashion to directly discriminate speakers. The baseline end-to-end SV approach consists of mapping a variable length speech segment to a fixed dimensional speaker vector by estimating the mean of hidden representations in DNN structure. We improve upon this technique by computing a distance function between two utterances which takes into account common phonetic units. The whole network is optimized by employing a triplet-loss objective function.

The proposed approaches are evaluated on commonly used datasets such as NIST SRE 2010 and RSR2015. Significant improvements are observed over the baseline systems on both the text-dependent and text- independent SV tasks by applying phonetic knowledge.

Keywords: speaker verification, text-dependent speaker verification, i-vector, PLDA, deep neural network, phonetic information, speaker embedding, end-to-end, distance metric.

Résumé

L'objectif de cette thèse est d'améliorer les techniques actuelles en vérification du locuteur (Speaker Verification, SV), généralement basées sur l'utilisation de "identity-vectors" (i-vectors) et de réseaux de neurones profonds (Deep Neural Network, DNN), en exploitant plusieurs informations (phonétiques) issues d'un système de reconnaissance automatique de la parole (Automatic Speech Recognition, ASR). Des locuteurs différents couvrent différents sous-espaces à l'intérieur d'un espace acoustique universel, communément appelé "Universal Background Model". Le sous-espace propre à chaque locuteur dépend des caractéristiques vocales de celui-ci, ainsi que du texte énoncé par le locuteur. Dans l'état de l'art actuel, les systèmes de SV extraient les i-vectors en appliquant une technique d'analyse factorielle permettant d'obtenir une représentation à faible dimension spécifique au locuteur. De plus, les DNN sont aussi utilisés dans le cadre conventionnel des i-vectors, pour modéliser les caractéristiques phonétiques présentes dans le signal de parole. Dans cette thèse, nous proposons d'utiliser plusieurs techniques exploitant les connaissances phonétiques du signal de parole afin d'obtenir une représentation augmentée du locuteur.

Les techniques développées dans cette thèse sont appliquées à plusieurs tâches de SV, notamment dépendantes et indépendantes du texte. Pour la SV indépendante du texte, un système de reconnaissance vocale basé sur les DNN est utilisé pour estimer les probabilités phonétiques a posteriori, qui sont ensuite exploitées pour améliorer les informations propres au locuteur inclues dans les i-vectors. Pour la SV dépendant du texte, cette approche est étendue pour exploiter principalement l'information temporelle, c'est-à-dire en utilisant la déformation temporelle dynamique (dynamic time warping) sur les vecteurs d'informations des locuteurs.

Finalement, au lieu d'utiliser un DNN pour déduire les caractéristiques phonétiques, celui-ci est entraîné de bout-en-bout pour distinguer les locuteurs. La méthode de référence consiste à relier un segment de parole de longueur variable à un vecteur-locuteur de dimension fixe en estimant la moyenne des représentations internes au DNN. Nous améliorons cette technique en calculant la distance entre deux échantillons en utilisant leurs caractéristiques phonétiques communes. L'entièreté du réseau est optimisée grâce à une fonction objectif "triplet-loss".

Les approches proposées sont évaluées sur les bases de données RSR2015 et NIST SRE 2010. Une amélioration significative par rapport au système de référence a été mesurée en exploitant les caractéristiques phonétiques, à la fois pour les tâches dépendantes et indépendantes du texte.

Mots clefs: vérification du locuteur, vérification du locuteur dépendant du texte, i-vector, PLDA, deep neural network, information phonétique, speaker embedding, end-to-end, distance metric.

Acknowledgement

I would like to thank Herve Bourlard and Petr Motlicek for their supervision. In particular, I would like to express my gratitude to Herve Bourlard for providing good research environment at Idiap and for the guidance. I am indebted to Petr Motlicek for the patience, constructive and critical comments to my research.

I would like to thank my jury members: Alexandre Schmid, Jean-Philippe Thiran, Jean-Francois Bonastre and Honza Cernocky for reading my thesis and providing insightful comments.

I would like to thank European Union Project, Speaker Identification Integrated Project (SIIP) for funding my Phd research. My gratitude goes to my mentor at NEC Japan, Takafumi Koshinaka, for giving me an internship opportunity. In particular, the discussion with the researcher at NEC helped me to broaden my perspectives.

During these four years, I had the opportunity to work closely with Srikanth Madikeri and Marc Ferras. Their constructive comments towards my research work helped me immensely.

I would like to thank my friends at Idiap for making my stay at Martigny a good experience. In particular, the discussion during coffee breaks and various events organized by them helped me in various ways. I would also express my gratitude to the system administrator at Idiap for making the systems work seamlessly well. I would like to thank Idiap secretaries for helping me to arrange non research stuffs. Finally, I like to thank my family members for their support during these four years.

Contents

List of Figures

List of Tables

List of Acronyms

SV	speaker verification
GMM	Gaussian mixture model
UBM	universal background model
ASR	automatic speech recognition
SS	sufficient statistics
JFA	joint factor analysis
i-vector	identity-vector
HMM	hidden Markov model
DNN	deep neural network
AM	acoustic model
LM	language model
EM	expectation-maximization
KL	Kullback–Leibler
DTW	dynamic time warping
MAP	maximum a posteriori
LPCC	linear predictive coding coefficients
MFCC	mel-frequency cepstrum coefficients
SGMM	subspace Gaussian mixture model
NIST	national institute of standards and technology
SRE	speaker recognition evaluations
STG	short term Gaussianization
CDF	cumulative distribution function
LDA	linear discriminant analysis
PLDA	probabilistic linear discriminative analysis
DET	detection error tradeoff
EER	equal error rate

IB	information bottleneck
IPA	international phonetic alphabet
WER	word error rate
SIIP	speaker identification integrated project
Bi-LSTM	bi-directional long short term memory
ReLU	rectified linear unit
SVM	support vector machine
MLLR	maximum likelihood linear regression
DCF	decision cost function

1 Introduction

Contents

Due to widespread use of mobile phones, the speech processing applications are rising day-by-day. Speaker verification (SV) is a related speech processing technology that aims to authenticate the identity of a user from voice samples. SV systems are usually deployed in real-time scenarios such as for banking, etc. SV can be broadly categorized into text-dependent and text-independent tasks. In text-dependent SV, the user is constrained to utter a specific lexical content while no such constraints are applied for text-independent SV. Commercial companies like Apple, Google, Microsoft have released their text-dependent SV products with the lexical content of the voice sample to be “Hey Siri”, “Ok Google” and “Hey Cortana” respectively. Figure 1.1 shows a typical SV scenario in which a user gets authenticated to the system via voice. Building a SV system for these applications poses real challenges, as a process of user-authentication usually requires to operate over a few seconds of audio recordings. To achieve this, novel SV approaches are required, which extract speaker characteristics not only from acoustics of a speech signal but also other characteristics.

Typical SV approaches are built around a Gaussian mixture model (GMM) to cluster the acoustic space of the speaker feature vectors. The state-of-the-art SV technique employs factor analysis model on the GMM representation of the speakers in order to obtain a low dimension vector, referred to as *i-vector*. The i-vector approach consists of first computing the sufficient statistics (SS) and then obtaining the low dimensional speaker representation. SS extraction of an utterance aims to map a varying length speech utterance to a high dimensional vector. Typically, these SS are computed by scoring each frame of an utterance against a GMM. Recent research reveals that replacing GMM by deep neural network (DNN) outputs for extraction of SS results in significant improvement of SV performance (Lei et al., 2014). Unlike GMM employed to unsupervisely cluster the acoustic space, the DNN is usually trained to classify speech into phonetic classes in a supervised manner using text-transcripts. These findings suggest that the spoken text of the user is useful for SV in addition to acoustic (speaker-specific) characteristics.

In this thesis, we propose new approaches that exploit phonetic and speaker information for text-independent and text-dependent SV scenarios. For text-independent SV, we aim to incorporate phonetic information via automatic speech recognition (ASR) to compute SS as opposed to using directly DNN outputs. For text-dependent task, we present approaches exploiting context of phonetic units for building an SV system.

The rest of the chapter is organized as follows. Section 1.1 presents the motivation of the works presented in this thesis. In Section 1.2, we describe SV scenarios considered in this thesis to evaluate the developed techniques while in Section 1.3, we describe the different contributions made towards advancing the state-of-the-art SV techniques. Section 1.4 presents a chapter-wise outline that summarizes the contributions of the thesis.

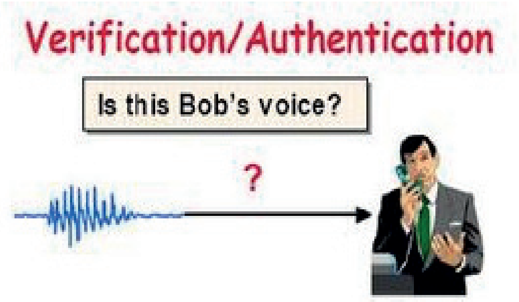

Figure 1.1: *Typical example of a speaker verification system that is deployed for practical scenarios.*

1.1 Motivations

In the past, it has been shown that certain sound units contain more speaker discriminating characteristics than others (Amino et al., 2006; Moez et al., 2016; Besacier et al., 2000). For example, Amino et al. (2006) have found that nasals and vowels are more effective in discriminating speakers than other phonemes. Furthermore, speakers are distinguishable in terms of the choice of the usage of words or the combination of words. This hypothesis was examined through the use of sequence of phone units by Campbell et al. (2003). Motivated by these evidences and the fact that the information carried in the sequence of phonetic units has not been studied after the emergence of the i-vector framework, we aim to employ the sequence information automatically extracted from voice recordings to improve SV.

The i-vector framework provides reasonable accuracies for various SV conditions, including short duration utterances. This approach usually employs a GMM that is trained in an unsupervised manner. This implies that the content information of the speech signal is ignored. Recent work suggests that phonetic information can be incorporated in the i-vector framework by the application of an ASR extracting complementary information. In Lei et al. (2014), this is achieved by first training DNN in an ASR fashion with outputs as the context-dependent phones (senones). The trained DNN produces senone posterior probabilities which can be directly used in extracting SS for i-vector framework, or the DNN outputs are further processed by an ASR decoder constrained by a lexicon and language model. Figure 1.2 shows the SV performance when different acoustic models are applied (Su and Wegmann, 2016). The results are presented in terms of equal error rate (EER), which correspond to the operating point at which the probability of false acceptance (i.e. impostor falsely authenticated) is equal to the probability of miss detection rate (i.e. correct speaker is rejected). The speaker dependent ASR system as shown in Figure 1.2 is trained by adapting the DNN acoustic model to each of the speakers. It can be observed from the figure that the EER of the SV decreases considerably when ASR is used in SV task. In other words, this result reveals that phonetic information of speech signal is useful for building an SV system. This thesis focuses on the application of

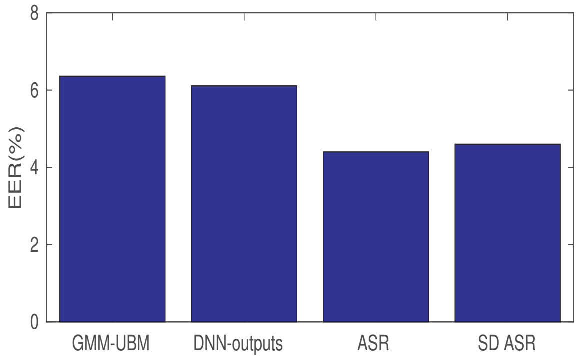

Figure 1.2: *Error rates on applying ASR outputs for speaker verification. SD refers to speaker dependent system. The SS from various ASR based systems are incorporated in the i-vector framework. The performances of these systems are evaluated on NIST SRE 2010 (Su and Wegmann, 2016).*

phonetic information to model the speaker representations.

1.2 Scenarios in the thesis

In this thesis, the application of phonetic information is investigated for two SV tasks, particularly, (i) text-dependent, and (ii) text-independent. For text-dependent SV, we are mainly interested in following two scenarios:

- **Fixed-phrase**: the speaker is constrained to utter a specific phrase for authentication. In this case, all speakers repeat the same phrases in different sessions.
- **Random-digit strings**: the user has to utter a prompted random permutation of digits for verification.

The error rates for **fixed-phrase** based SV is lower than for **random-digit strings** scenario. However, an disadvantage of **fixed-phrase** is that it is more susceptible to spoofing attacks than **random-digit strings**. We also evaluate our systems for text-independent scenario in which the user is not constrained to utter any specific phrase during enrollment and testing phase.

1.3 Summary of contributions

In this thesis, we aim at improving the state-of-the-art approaches to SV by exploiting phonetic information of the speech signal. As shown in Figure 1.3, we hypothesize that the knowledge from speech recognition system can be applied in order to better model the speaker characteristics. To confirm this hypothesis, we design techniques to tackle text-dependent and text-independent SV tasks. The contributions of the thesis are the following:

- The baseline i-vector based SV is implemented and evaluated on standard text-independent dataset. We provide an alternative approach to i-vector framework which applies subspace GMM (SGMM). The SGMM is typically trained in a supervised manner to capture phonetic and speaker variabilities. Similar to i-vector extraction, the SGMM framework is used to estimate low-dimensional speaker vectors and then used for training the back-end classifier.

- For text-independent scenario, we employ various ASR systems to compute SS which are subsequently applied for i-vector extraction. We then show that there is a direct correlation between the accuracy of the ASR system and the performance of SV systems built upon these models.

- The techniques developed for text-independent SV are further extended for **fixed-phrase** based SV. We experiment with exploiting context-dependent phone posterior probabilities applied in i-vector framework. The limitation of the baseline system is analyzed and we propose template matching approaches using speaker informative features (referred to as online i-vectors).

- The baseline i-vector framework is analyzed for operating on **random-digit strings** task. In contrast to phrase based SV, it is not straight-forward to incorporate content information. We propose to use SS computed from the ASR output, subsequently applied in i-vector extraction. Furthermore, we apply content matching to normalize the lexical-content of the enrollment to the test data using online i-vectors as features.

- Finally, unlike training SV components independently, we incorporate phonetic information in the DNN framework directly for text-dependent SV (**fixed-phrase** and **random-digit strings**). The DNN is trained to discriminate speakers in an end-to-end fashion. The conventional SV approach involves mapping a variable length speech segment to a fixed dimensional speaker vector by estimating the mean of hidden representations in DNN structure. This strategy may not use content information of speech signal efficiently which is essential for this task. We exploit phonetic information by computing a distance function with linguistic units common to both enrollment and test data. The whole network is optimized by employing a triplet-loss objective function in an end-to-end fashion to produce SV scores.

1.4 Thesis outline

This thesis is organized as follows.

Chapter 2 presents the relevant background literature on SV, as these approaches will be considered as baseline system in this thesis.

Chapter 3 explores various approaches developed for incorporating phonetic information of the speech signal for text-independent SV. In this context, we also explore the use of SGMMs

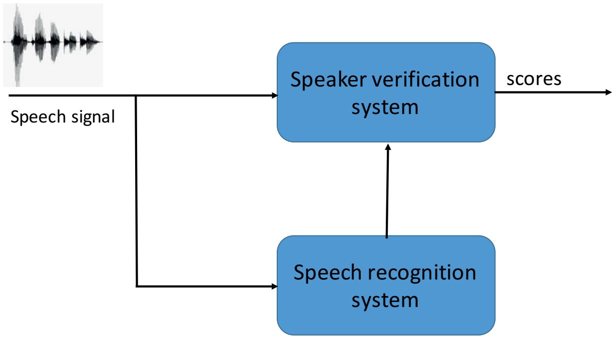

Figure 1.3: *Applying knowledge from speech recognition to improve speaker verification.*

Table 1.1: *Notations*

k	thousand
$\mathbb{R}$	The set of real numbers
$\mathbb{R}^D$	The set of D dimensional vectors over $\mathbb{R}$
$\mathbb{1}_{\text{condition}}$	is equal to 1 if the condition is true, 0 otherwise
Non bold capital letters indicate size or functions	
Non-bold small letters indicate scalars or functions	
Bold capital letters indicate matrices	
Bold small letters indicate column vectors	

and the application of acoustic model applied in ASR framework.

In Chapter 4, **fixed-phrase** based text-dependent SV task is explored. We carefully analyze the performance of the baseline system and highlight the limitation of these approaches. We propose new methods to incorporate of phonetic units sequence by applying template matching techniques.

In Chapter 5, we explore **random-digit strings** based text-dependent SV. We explore a new approach to use common set of phones or subword units to obtain SV scores.

Chapter 6 explores various DNN based speaker embedding approaches developed for text-dependent SV. We explore end-to-end approaches in this context. We propose a specific objective function in a DNN based framework that exploits phonetic information in an implicit manner.

1.5 Notations

Table 1.1 summarizes the general notations that are used in this thesis. This notation is consistent across chapters and when needed, a chapter specific notation is provided.

2 Background, Datasets and System Configurations for Speaker Verification

Contents

In this chapter we present background work on speaker verification. We also present datasets and system configurations for speaker verification that are used in this thesis.

2.1 Introduction

Automatically recognizing speakers is useful for various practical applications, such as in voice-based forensics, banking systems, etc. Speaker recognition tasks (i.e. diarisation) are often used to extract valuable input information to improve the accuracies of automatic speech recognition. In text-dependent SV, the user is constrained to utter a specific lexical content. Such systems are usually implemented using fixed-phrases or sequences of words/digits. In case of text-independent SV, no such constrain is imposed on the spoken-content of the client. Unconstrained spoken input makes it more challenging than the text-dependent task (Campbell Jr, 1997).

Building a SV for detecting speakers attempts to find distinguishing traits of the person from the voice samples. Past research shows that speakers sound differently due to the physical difference in the speech production mechanism, like vocal tract shape, larynx, etc (Kinnunen and Li, 2010). The language of the person and external environment also plays an important role in characterizing the voice characteristics. Conventional SV approaches rely on applying signal processing techniques to extract speaker invariant characteristics, which are followed by acoustic modelling, usually employing probabilistic models. The traditional SV approaches have shown to provide state-of-the-art performance in a variety of conditions, like telephone or microphone recordings, various-languages. However, they usually require a large collection of labelled speaker data in order to deliver good performance (Garcia-Romero, 2012).

This chapter aims at giving a concise introduction to SV. First, we describe an overview of speaker recognition. Then, we present the description of features that are applied for building a SV system. This is followed by a description of the state-of-the-art system for SV. Finally, the dataset and evaluation metrics are discussed.

2.2 Speaker Recognition

The speaker recognition task aims to infer the identity of the talker in an audio recording. The term speaker-recognition in itself can refer to speaker identification, verification or diarization. Speaker identification involves choosing the closest class of an input test utterance. This process involves comparing a test voice sample against ‘N’ speaker templates and assigning the label of the closest speaker. SV refers to the case in which two utterances are provided as input to the algorithm in order to decide whether the utterances share the same class identity or not. Speaker diarization is the task of partioning an audio recording into segments belonging to different speakers. We are primarily interested in SV task as, the goal of this thesis is closely aligned with the objective of the speaker identification integrated project (SIIP)[1]. The goal of the SIIP project is to identify unknown speakers from intercepted audio recordings. Since there is a strong correlation between speaker identification and SV, in this thesis, techniques to address SV task are explored.

[1] http://www.siip.eu/SIIP-Project

Table 2.1: *A valid enrollment-test phrase pair for text-dependent speaker verification systems for different tasks. The phrases in* ***Fixed-phrase*** *and* ***Seen*** *tasks are phonetically balanced.*

Tasks	Enrollment phrase	Test phrase
Fixed-phrase	“the redcoats ran like rabbits"	“the redcoats ran like rabbits"
Seen	{ “the redcoats ran like rabbits", “only lawyers love millionaires", $\cdots$ }	any of the enrollment phrases
Random-digit strings	{ “five", “four", $\cdots$, “ten" }	{ “two", “five", $\cdots$ }

There are various strategies to develop a text-dependent SV system (Larcher et al., 2014b). In **fixed-phrase** based text-dependent SV, the phrase of the test data is expected to be identical to the enrollment (as shown in line 1 of Table 2.1). In case it is not, the system is trained to detect the mismatch and reject the claim. In many text-dependent applications, we would like to impose lesser constraint on the speaker while maintaining the same level of accuracy of the **fixed-phrase** based systems (Larcher et al., 2014b, 2008; Stafylakis et al., 2016). In one of the scenarios, the words of the test phrase are subset of the content of the enrollment. A potential example is when speaker models are created by pooling all N phrases uttered by the speaker during enrollment, while during test phase, the speaker utters only one of the N phrases (Scheffer and Lei, 2014). We are also interested in these two text-dependent SV scenarios to better understand the effect of content information:

- **Seen**: The enrollment data is created by pooling all the phrases spoken by the speaker. The test data consists of a single phrase, as illustrated in Table 2.1 (Line 2), and
- **Random-digit strings**: the enrollment data consists of the speaker uttering permutations of ten digits. During testing, the speaker is prompted to utter five digits only as shown in Table 2.1 (Line 3).

For implementing text-independent SV, the user is not constrained to utter any system-defined lexical content. Thus, it makes the process less restrictive and more challenging to handle this task. For addressing SV, it involves extraction of speaker informative features and a classification algorithm. The feature extraction process is also referred to as the front-end while classification process is called as the back-end.

2.3 Feature Extraction

Figure 2.1 shows the pipeline for extracting feature vectors from the speech signal. The speech signal is first pre-processed by applying pre-emphasis. This step is done to remove constant shifts to the signal. Suppose the n^{th} speech sample is represented by $s(n)$, then the

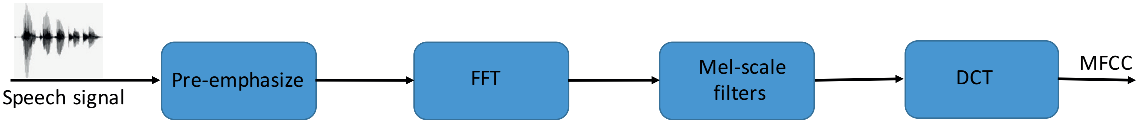

Figure 2.1: *Steps to extract features from speech signal.*

pre-emphasis is done by:

$$s(n) = s(n) - as(n-1),$$

where a is a pre-emphasis constant which is set to value of 0.97. This is then followed by a feature extraction module.

An ideal feature extractor aims to obtain a representation that captures speaker characteristics while filtering other variabilities of speech signal (Kinnunen and Li, 2010). These features should:

- be easy to compute,
- be robust to noise, and other environment conditions,
- be robust to spoofing attack,
- have large inter-speaker and less intra-speaker variability.

In literature, linear predictive cepstral coefficient (LPCC) and mel frequency cepstral coefficient (MFCC) have mostly been explored as features for SV (Kinnunen and Li, 2010). We describe MFCC feature extraction procedure as it is commonly applied in speech processing applications. MFCC features are computed by applying a sliding window of approximately 25 ms along the speech signal with a shift of 10 ms. This short segment of speech signal is assumed to be stationary and is referred to as a frame. Thus, an utterance is converted to a sequence of frames. The spectrogram of each speech frame is computed by applying fast Fourier transform (FFT). For a telephone speech with a sampling frequency of 8 kHz, the maximum frequency of the speech-frame in the spectral domain representation is expected to be 4 kHz. This is then followed by a filterbank analysis and a final compression. The MFCC computation applies a series of non evenly spaced triangular filters (usually 40). The centre frequencies of the filters are linearly spread in the mel domain. The mel-scale is chosen to mimic the human auditory perception. The filters are applied to accumulate frequency domain representation of the speech frame. A discrete cosine transform (DCT) is subsequently applied on the accumulated-outputs to obtain MFCC features. DCT is used to decorrelate the feature dimensions. It has been observed that a few co-efficients of MFCCs are sufficient for representing the short-time speech spectra. In particular, for speech processing application, only 13 coefficients are used, while 20 coefficients are usually used for speaker recognition (Povey et al., 2011b; Kinnunen and Li, 2010; Motlicek et al., 2015).

The state-of-the-art approaches append delta (referred to as Δ) and double-delta (referred to as $\Delta\Delta$) features to the MFCCs which aim to incorporate trajectory information (Dehak et al., 2011; Kinnunen and Li, 2010). For an utterance with sequence of MFCCs represented by $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T\}$, the delta-features are computed as the linear regression over a window of 'W' MFCCs as given by:

$$\mathbf{d}_t = \frac{\sum_{\rho=1}^{W} \rho(\mathbf{x}_{t+\rho} - \mathbf{x}_{t-\rho})}{2\sum_{\rho=1}^{W} \rho^2}, \tag{2.1}$$

where $\mathbf{d}_t$ are the delta features of an utterance for t^{th} frame. The double-delta is obtained by successive application of Equation 2.1 on the delta features. A short term Gaussianization (STG) is usually applied on MFCCs to remove unwanted variabilities, such as distortions due to channel, language, content, etc (Xiang et al., 2002; Pelecanos and Sridharan, 2001; Motlicek et al., 2015). STG aims to map feature components to a standard Gaussian distribution. STG can also be viewed as an approach to perform non-linear transformation of original features to warped features by using cumulative distribution function (CDF). STG is performed by using a sliding window of length L on each feature-dimension. The feature-values under the window are first sorted in ascending order and the rank of the current-frame (r, such that $N \geq r \geq 1$) is determined. Its corresponding CDF (c) will be given by:

$$\mathrm{c} = \frac{r - 1/2}{L}. \tag{2.2}$$

Then the warped features ($\hat{x}$) should satisfy:

$$c = \int_{-\infty}^{\hat{x}} \frac{1}{\sqrt{2\pi}} \exp(-\frac{y^2}{2}) dy. \tag{2.3}$$

The value of $\hat{x}$ can be obtained from the standard normal CDF. In most of the successful SV systems, the feature extraction is followed by probabilistic modelling. Typically in a SV framework, the verification process is divided into three phases: training, enrollment and the testing phase. During training, the parameters of the model are estimated from data of a large corpora, the enrollment and the test phases involve predicting the speaker label. We describe two successful statistical approaches to SV, namely, Gaussian mixture model-universal background model (GMM-UBM), and i-vector.

2.4 Gaussian Mixture Model - Universal Background Model

The GMM-UBM formulates the SV as a statistical hypothesis testing problem (Lee and Gauvain, 1993; Reynolds et al., 2000; Sturim et al., 2002). Mathematically, GMM-UBM seeks to obtain a ratio of two competing hypotheses (s_c) as given by:

$$s_c = (\frac{\mathrm{p}(\mathbf{X}|\mathbf{H}_0)}{\mathrm{p}(\mathbf{X}|\mathbf{H}_1)}) \geq \theta_\mathrm{t} \ (accept/reject), \tag{2.4}$$

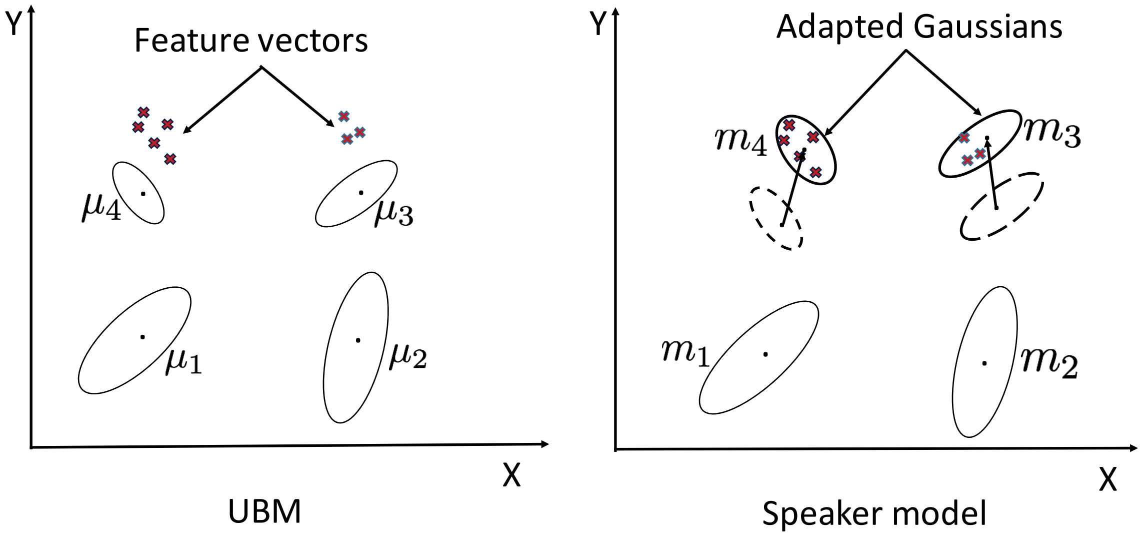

Figure 2.2: *Depiction of GMM-UBM approach to speaker verification. The mean vectors of the UBM are represented by {μ_1, μ_2, μ_3, μ_4} while the means of the speaker model are represented by {m_1, m_2, m_3, m_4}.*

where $\mathbf{X}$ is a speech utterance, $\mathbf{H}_0$ is the hypothesis that the utterance belongs to the claimed model (also referred to as null hypothesis) while $\mathbf{H}_1$ is the alternate hypothesis that the utterance is not spoken by the speaker. If the ratio (s_c) is greater than a threshold (θ_t), the claim is accepted otherwise, it is rejected. These two hypotheses are computed by applying a probability distribution function on the input speech features.

The GMM-UBM framework assumes the data to be generated from a GMM (Lee and Gauvain, 1993; Reynolds et al., 2000). Typically, thousands of hours of speaker data are used for building a large GMM (with 1 k mixture components) in the training phase, also referred to as UBM (Dehak et al., 2011). The training data is chosen so that it matches the evaluation condition. The probability density of a feature vector ($\mathbf{x}$) is given by:

$$p(\mathbf{x}|\lambda_{UBM}) = \sum_c \pi_c \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c), \tag{2.5}$$

where π_c is the weight of c^{th} Gaussian with mean $\boldsymbol{\mu}_c$ and covariance matrix $\boldsymbol{\Sigma}_c$, $\mathcal{N}$ is a multivariate normal distribution and λ_{UBM} refers to the parameters of the model ($\lambda_{UBM} = \{\pi_c$, $\boldsymbol{\mu}_c$, $\boldsymbol{\Sigma}_c\}_{c=1}^{K}$), such that $\sum_c \pi_c = 1$ assuming K is the mixture components of Gaussians. Assuming that an utterance is represented by $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \cdots, \mathbf{x}_T\}$ (as in Section 2.3), the likelihood (ℓ) is computed by assuming that each feature vector ($\mathbf{x}_i$) is independent and identically distributed (i.i.d.), as given by:

$$\ell(\mathbf{X}|\lambda_{UBM}) = \sum_{i=1}^{T} \log p(\mathbf{x}_i|\lambda_{UBM}). \tag{2.6}$$

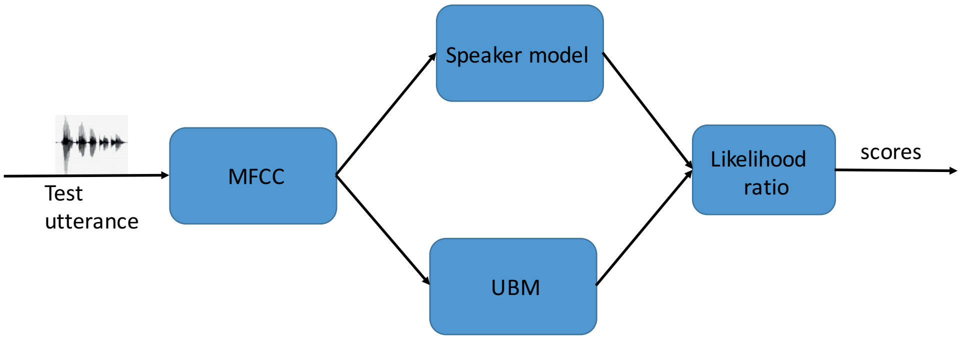

Figure 2.3: *Speaker verification system for obtaining likelihood of a test utterance.*

The total-data likelihood is computed by accumulating the likelihood of Equation 2.6 for each utterance. The parameters of the GMM model can be estimated by expectation maximization (EM) algorithm (Reynolds et al., 2000; Lee and Gauvain, 1993). EM involves the successive application of two steps, namely, E-step and M-step in an iterative manner in order to maximize the data-likelihood. In the E-step, the posterior probabilities of mixture components of Gaussians are computed with respect to the parameters of the model, while the M-step consists of re-estimating the parameters of the GMM. Diagonal covariance matrix of the GMM has shown to provide good results for SV.

The GMM-UBM approach is illustrated graphically by Figure 2.2. For creating the speaker model, the data of i^{th} speaker is taken to adapt the parameters of the GMM-UBM using maximum-a-posterior (MAP) principle (Reynolds et al., 2000; Lee and Gauvain, 1993). Thus, the new parameters of speaker model are given as follows:

$$\hat{\boldsymbol{\mu}}_i = \alpha_i^m \mathbf{E}(\mathbf{x}_i) + (1 - \alpha_i^m)\boldsymbol{\mu}_i, \tag{2.7}$$

$$\hat{\boldsymbol{\sigma}}_i^2 = \alpha_i^v \mathbf{E}(\mathbf{x}_i^2) + (1 - \alpha_i^v)(\boldsymbol{\sigma}_i^2 + \boldsymbol{\mu}_i^2) - \hat{\boldsymbol{\mu}}_i^2. \tag{2.8}$$

where α_i^v, α_i^m are the weights that balance the parameters of the GMM-UBM and new estimates, $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i^2$ are the means and variances of i^{th} mixture of UBM. The factors $\mathbf{E}(\mathbf{x}_i)$ and $\mathbf{E}(\mathbf{x}_i^2)$ are defined as the first and second order statistics of the data as defined by:

$$\mathbf{E}_i(\mathbf{x}) = \frac{1}{\eta_c} \sum_t p(i|\mathbf{x}_t)\mathbf{x}_t, \tag{2.9}$$

$$\mathbf{E}_i(\mathbf{x}^2) = \frac{1}{\eta_c} \sum_t p(i|\mathbf{x}_t)\mathbf{x}_t^2, \tag{2.10}$$

where η_c is the zeroth order statistics of the data. In practice, only the means of the GMM are adapted for obtaining the speaker models.

During evaluation, SV scores of an utterance are obtained by assuming that each of the observation is i.i.d. as given by Equation 2.6. The evaluation phase for GMM-UBM system is illustrated in Figure 2.3 which consists of obtaining likelihood score (ℓ^*) with respect to the speaker model (M) and GMM-UBM as given by:

$$\ell^*(\mathbf{X}, \lambda_{UBM}, \lambda_M) = \ell(\mathbf{X}|\lambda_\mathrm{M}) - \ell(\mathbf{X}|\lambda_{UBM}).$$

Furthermore, score normalization is applied on these GMM-UBM scores to compensate for mismatch in the enrollment and test data. We describe one of the successful score normalization techniques, referred to as T-norm (Auckenthaler et al., 2000; Hébert and Boies, 2005; Reynolds et al., 2000). It involves scoring of the test utterance against L impostor adapted models to obtain L likelihood scores. The mean ($\mu_\mathbf{X}$)and standard deviation ($\sigma_\mathbf{X}$) of the L scores are computed and used to obtain normalized scores as follows:

$$\ell^*(\mathbf{X}, \lambda_{UBM}, \lambda_M) = \frac{s(\mathbf{X}, \lambda_{UBM}, \lambda_M) - \mu_\mathbf{X}}{\sigma_\mathbf{X}}.$$

2.5 i-vector extraction

A GMM-UBM SV can be applied to compute zeroth-, first- and second-order statistics. These three statistics are also referred to as sufficient statistics (SS). Given an GMM with 'K' mixture components, the zeroth-order statistics (also referred to as soft-count) for c^{th} mixture (η_c) is computed as follows:

$$\eta_c = \sum_t p(c|\mathbf{x}_t, \lambda_{UBM}). \tag{2.11}$$

The zeroth-order statistics of the utterance is obtained by concatenating the soft-counts of all mixtures of GMM, i.e. $\boldsymbol{\eta} = [\eta_1, \eta_2, \eta_3, \cdots, \eta_K]^T$. The first order statistics of c^{th} mixture is computed as the weighted average of the features, as given by:

$$\mathbf{F}_c = \sum_t p(c|\mathbf{x}_t, \lambda_{UBM})\mathbf{x}_t. \tag{2.12}$$

The first-order statistics are obtained by the concatenating first order statistics of various clusters, $\mathbf{F} = [\mathbf{F}_1^T, \mathbf{F}_2^T, \mathbf{F}_3^T, \cdots, \mathbf{F}_K^T]^T$. The first-order statistics normalized by the soft-count per cluster are referred to as *mean super-vector*. Similarly, the second-order statistics are obtained by the concatenation of covariances of all mixtures of GMM, where the c^{th} cluster covariance is defined by:

$$\mathbf{S}_c = \sum_t p(c|\mathbf{x}_t, \lambda_{UBM})\mathbf{x}_t\mathbf{x}_t^T. \tag{2.13}$$

The matrix $\mathbf{S}_c$ is full covariance matrix. These SS are subsequently applied in the state-of-the-art i-vector technique (Garcia-Romero, 2012; Dehak et al., 2011). This approach is illustrated in Figure 2.4 and we describe it in this section. The i-vector framework aims to map variable length speech utterance into low-dimension vector, referred to as identity vector or i-vector. This representation comprises all the variabilities of speech signal, like language, content,

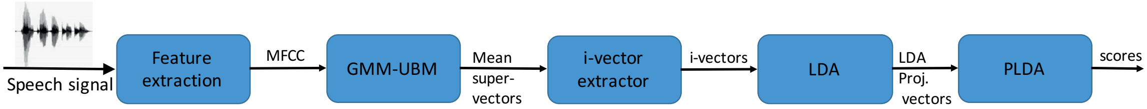

Figure 2.4: *The baseline i-vector system.*

speaker, etc. The state-of-the-art SV usually uses 1 k mixture components of GMM-UBM and 60 dimensional features, leading to 60 k dimensional first-order statistics, while the i-vector is chosen to be usually 400 dimensional. In the i-vector approach, the adapted mean super-vector (**m**) of an utterance can be decomposed as:

$$\mathbf{m} = \boldsymbol{\mu} + \boldsymbol{\Gamma}\mathbf{w} + \mathbf{e}_u, \tag{2.14}$$

where $\boldsymbol{\mu}$ is the mean super-vector of GMM-UBM, **w** is a random variable (also referred to as i-vector), which is assumed to have Gaussian distribution with zero mean and identity covariance matrix **I**, i.e. $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\boldsymbol{\Gamma}$ is referred to as the total variability matrix. The term $\mathbf{e}_u$ is the residual error. The parameters of the model are estimated by EM algorithm.

In the E-step, the posterior distribution of latent variable (**w**) is obtained using the sufficient statistics from the GMM-UBM. Considering an utterance represented by $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \cdots, \mathbf{x}_T\}$, the SS as defined by Equations 2.11, 2.12 and 2.13, are first computed. The i-vector of an utterance is obtained by:

$$\mathbf{w} = \boldsymbol{\Omega}^{-1}\boldsymbol{\Gamma}^T\boldsymbol{\psi}^{-1}\boldsymbol{\Gamma}\mathbf{F}_{\mathbf{w}}, \tag{2.15}$$

$$\boldsymbol{\Omega} = (\mathbf{I} + \boldsymbol{\Gamma}^T\boldsymbol{\psi}^{-1}\boldsymbol{\eta}\boldsymbol{\Gamma}), \tag{2.16}$$

where $\boldsymbol{\psi}$ is the covariance matrix of the error term, $\mathbf{e}_u$. Furthermore, these following accumulators are collected during the E-step:

$$\mathbf{C} = \sum_i \mathbf{F}_i\mathbf{w}_i, \tag{2.17}$$

$$\mathbf{A}^c = \sum_i \eta_c^i(\boldsymbol{\Omega}^{-1} + \mathbf{w}_i\mathbf{w}_i^T). \tag{2.18}$$

In the M-step, the parameter of the model is updated as given by:

$$\boldsymbol{\Gamma}^c = \mathbf{C}(\mathbf{A}^c)^{-1}, \tag{2.19}$$

where $\boldsymbol{\Gamma}^c$ is the c^{th} component of the total variability matrix ($\boldsymbol{\Gamma}$). The state-of-the-art system optionally applies linear discriminant analysis (LDA) on top of i-vectors to capture speaker variabilities (Dehak et al., 2011). In our experiments, we found that LDA benefits performance of text-independent SV while it degrades the performance of text-dependent SV.

2.6 Linear Discriminant Analysis

The LDA is widely applied in many pattern recognition task, such as image, speech, speaker recognition, etc (Dehak et al., 2011). As shown in Figure 2.5, LDA aims to find orthogonal basis vectors that can discriminate two or more classes. Furthermore, the vectors or axes are chosen in such a way that it maximizes inter-class variability and minimizes intra-class variance. The LDA optimization problem can be formulated by:

$$\mathrm{J}_b(\mathbf{v}) = \frac{\mathbf{v}^T \mathbf{S}_b \mathbf{v}}{\mathbf{v}^T \mathbf{S}_w \mathbf{v}},$$

where $\mathbf{v}$ is defined as the weight vector, $\mathrm{J}_b(\mathbf{v})$ is referred to as Rayleigh coefficient. The quantities $\mathbf{S}_b$ and $\mathbf{S}_w$ are referred to as inter-class and intra-class variance as defined by:

$$\mathbf{S}_b = \sum_{s=1}^{S} (\overline{\mathbf{w}}_s - \overline{\overline{\mathbf{w}}})(\overline{\mathbf{w}}_s - \overline{\overline{\mathbf{w}}})^T,$$

$$\mathbf{S}_b = \sum_{i=1}^{i=S} \frac{1}{n_i} \sum_{j=1}^{j=n_i} (\mathbf{w}_j^i - \overline{\mathbf{w}}_i)(\mathbf{w}_j^i - \overline{\mathbf{w}}_i)^T,$$

where $\overline{\mathbf{w}}_i$ is the mean of the i-vectors for i^{th} class, $\mathbf{W}_i = \{\mathbf{w}_1^i, \mathbf{w}_2^i, \cdots, \mathbf{w}_{n_i}^i\}$, $\overline{\overline{\mathbf{w}}}$ is the average of $\overline{\mathbf{w}}_i$ and n_i is the number of i-vectors of i^{th} class. The LDA formulation consists of maximizing the Rayleigh coefficient to obtain the following eigen-value equation:

$$\mathbf{S}_b \mathbf{v} = \lambda \mathbf{S}_w \mathbf{v},$$

where λ is the diagonal matrix of eigen-vectors.

In the state-of-the-art SV approaches, the i-vectors are first length normalized before applying LDA algorithm (Garcia-Romero, 2012; Garcia-Romero and Espy-Wilson, 2011). The LDA matrix is applied on i-vectors to obtain speaker discriminating vectors, referred to as LDA-projected features. probabilistic linear discriminative analysis (PLDA) model is further applied on these features to produce SV scores.

2.7 Probabilistic Linear Discriminative Analysis

The PLDA is applied with either i-vectors or LDA-projected i-vectors as input to the algorithm. (Prince and Elder, 2007). For convenience, we describe PLDA in this section assuming i-vector input-representation. In PLDA formulation, an i-vector ($\mathbf{w}$) can be decomposed into

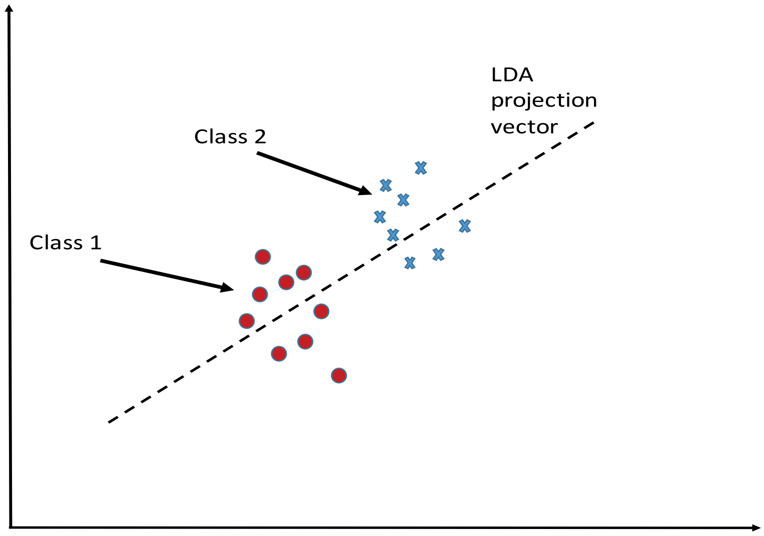

Figure 2.5: *Depiction of LDA for pattern recognition task.*

speaker factor as given by:

$$\mathbf{w} = \boldsymbol{\mu}_{\mathbf{w}} + \boldsymbol{\Pi}\boldsymbol{v} + \boldsymbol{\epsilon}_u, \tag{2.20}$$

where $\boldsymbol{\mu}_{\mathbf{w}}$ is the mean of i-vectors, $\boldsymbol{\Pi}$ is the speaker-variability matrix, $\boldsymbol{v}$ is the speaker-latent factor while $\boldsymbol{\epsilon}_u$ is the error term. Furthermore, it is assumed that the factor $\boldsymbol{v}$ is Gaussian distributed with zero mean and identity covariance matrix and the error term follows a Gaussian distribution with zero mean and full covariance matrix, $\mathbf{A}$. Intuitively, the parameter $\mathbf{A}$ represents the intra-speaker covariance matrix while the quantity $\boldsymbol{\Pi}\,\boldsymbol{\Pi}^T$ denotes the inter-class covariance. The parameters of the PLDA model (θ_{PLDA} = {$\boldsymbol{\mu}_{\mathbf{w}}$, $\boldsymbol{\Pi}$, $\mathbf{A}$}) are estimated from a large speaker labelled corpora in a maximum-likelihood fashion using EM algorithm.

The PLDA model can be applied to obtain log-likelihood scores. Assuming the i-vectors of the enrollment and test data represented by $\mathbf{w}_e$ and $\mathbf{w}_t$ respectively, the likelihood is defined as the ratio of the hypothesis that the vectors belong to the same class and the alternate hypothesis that the vectors do not share the same class identity. This is mathematically represented by:

$$s(\mathbf{w}_e, \mathbf{w}_t) = \log \frac{p(\mathbf{w}_e, \mathbf{w}_t | \mathbf{H}_0)}{p(\mathbf{w}_e, \mathbf{w}_t | \mathbf{H}_1)}. \tag{2.21}$$

For the PLDA model, assuming the probability of vectors ($\mathbf{w}_e$ and $\mathbf{w}_t$) are statistically independent, the above equation can be simplified to obtain:

$$s(\mathbf{w}_e, \mathbf{w}_t) = \log \frac{p(\mathbf{w}_e, \mathbf{w}_t | \theta_{PLDA})}{p(\mathbf{w}_e | \theta_{PLDA})\, p(\mathbf{w}_t | \theta_{PLDA})}. \tag{2.22}$$

The log-likelihood ratio can be simplified to obtain:

Table 2.2: *Classification of errors in statistical decision theory. Type I and II errors are of interest for SV task.*

Types of Errors		Null hypothesis ($\mathbf{H}_0$) is	
		True	False
Decision about $\mathbf{H}_0$	Fail to reject	True positive	Type II error (False Negative)
	Reject	Type I error (False positive)	True negative

$$s(\mathbf{w}_e, \mathbf{w}_t) = \log \mathcal{N}\left(\begin{bmatrix} \mathbf{w}_e \\ \mathbf{w}_t \end{bmatrix}; \begin{bmatrix} \boldsymbol{\mu}_{\mathbf{w}} \\ \boldsymbol{\mu}_{\mathbf{w}} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Upsilon} & \mathbf{B} \\ \mathbf{B} & \boldsymbol{\Upsilon} \end{bmatrix} \right) - \log \mathcal{N}\left(\begin{bmatrix} \mathbf{w}_e \\ \mathbf{w}_t \end{bmatrix}; \begin{bmatrix} \boldsymbol{\mu}_{\mathbf{w}} \\ \boldsymbol{\mu}_{\mathbf{w}} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Upsilon} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Upsilon} \end{bmatrix} \right), \tag{2.23}$$

where $\mathbf{B}= \boldsymbol{\Pi}\, \boldsymbol{\Pi}^T$, $\boldsymbol{\Upsilon} = \mathbf{B} + \mathbf{A}$. By setting $\boldsymbol{\mu}_{\boldsymbol{\nu}} = 0$, we obtain:

$$s(\mathbf{w}_e, \mathbf{w}_t) = \mathbf{w}_e^T \mathbf{Q} \mathbf{w}_e + \mathbf{w}_t^T \mathbf{Q} \mathbf{w}_t - 2 \mathbf{w}_e^T \mathbf{P} \mathbf{w}_t + constants, \tag{2.24}$$

where the matrices **P** and **Q** are defined by:

$$\mathbf{P} = \boldsymbol{\Upsilon}^{-1} - (\boldsymbol{\Upsilon} - \mathbf{B}\boldsymbol{\Upsilon}^{-1}\mathbf{B})^{-1}, \tag{2.25}$$

$$\mathbf{Q} = \boldsymbol{\Upsilon}^{-1}\mathbf{B}(\boldsymbol{\Upsilon} - \mathbf{B}\boldsymbol{\Upsilon}^{-1}\mathbf{B})^{-1}. \tag{2.26}$$

2.8 Joint Factor Analysis

Joint factor analysis (JFA) can be used as an alternative to the i-vector PLDA approach mentioned earlier for SV (Kenny et al., 2007). JFA has been successfully applied for text-dependent task in which the phonetic variability is explicitly modelled as a separate latent variable. In the JFA model, the mean super-vector of an utterance ($\mathbf{m}^{'}$) is factorized as follows:

$$\mathbf{m}^{'} = \boldsymbol{\mu} + \mathbf{D}\mathbf{z} + \mathbf{U}\mathbf{y}, \tag{2.27}$$

where **D** is a diagonal matrix capturing the speaker variabilities, $\boldsymbol{\mu}$ is the mean supervector of the UBM; **z** , **y** denote the speaker and channel factors respectively while **U** is the Eigenchannel matrix. The EM algorithm for i-vector approach is applied twice to obtain the parameters of the JFA model. In the first step, the parameter **D** is obtained from the equation $\mathbf{m}^{'} = \boldsymbol{\mu} + \mathbf{D}\mathbf{z}$, while in the second step, the **U** is estimated by re-normalizing the first order statistics. Given the parameters of JFA, we apply the Gauss-Seidel approach to obtain estimates of **z** and **y** for a speech recording. During evaluation, cosine distance between speaker factors (**z**) of the enrollment and test data are used to obtain SV scores.

2.9 Evaluation metrics

The performance of a SV system is usually measured using statistical classification theory (Bishop, 2016; Duda and Hart, 1973). For any classification task, four types of scenarios are encountered while analyzing the errors, as depicted in Table 2.2. The performance of a SV system is evaluated on these following two errors:

- **Type I error**: A type I error occurs when the null hypothesis ($\mathbf{H}_0$) is wrongly rejected. This is also referred to as miss detection, or false negative (P_{Miss}).
- **Type II error**: Type II errors occur when the null hypothesis ($\mathbf{H}_0$) is erroneously accepted. This error is also referred to as false alarm, or false positive (P_{FA}).

The Type I and II errors are computed based on decision-threshold (of Equation 2.4). Thus, if the threshold is set to a higher value, the system is expected to have less false positive errors while the lower threshold would lead to more false negative errors.

Detection error tradeoff (DET) curve is introduced to evaluate the SV on various detection thresholds (of Equation 2.4) (Martin et al., 1997). In the DET curve, the two errors are plotted on both axes, giving uniform treatment to both the errors. The Miss-detection rate is plotted along the Y-axis while the false positive rate is plotted along X-axis. It is to be noted that the scales along the X axis is a non-linear function of false positive rate (Martin et al., 1997). Figure 2.6 shows a typical DET curve of two systems for a SV task. From Figure 2.6, it can be observed that 'method 1' outperforms 'method 2' since the DET curve of the former approach is closer to the origin.

The NIST holds speaker recognition evaluations[2] on a regular basis and they define two metrics for evaluating the performance of the SV algorithms, namely, (i) equal error rate (EER), and (ii) decision cost function (DCF) (Martin et al., 1997; Doddington et al., 2000; Brümmer, 2007; Brümmer and de Villiers, 2013). EER is defined as the operating point of a system at which the miss-detection is equal to false-alarm rate. For example, the EER of method 1 of Figure 2.6 is approximately equal to 7%. In this thesis, the performance of all the systems is reported in terms of EER. DCF is defined as the weighted sum of false-alarm and miss-detection rates. These weights are obtained by using a cost function C_{FA} and C_{Miss} and prior probability of same-speaker (P_s) and different-speaker (P_d). The DCF can be expressed by:

$$DCF = P_d C_{FA} P_{FA} + P_s C_{Miss} P_{Miss}. \tag{2.28}$$

The DCF is computed for all possible detection-thresholds (of Equation 2.4) to obtain the minimum value, referred to as min-DCF. In this thesis, performance of selected systems is reported in minDCF in addition to EER and DET curve. The values of the costs (C_{FA} and C_{Miss}) depends on the particular applications. Typically for a text-independent system the C_{FA} is

[2] https://www.nist.gov/multimodal-information-group/speaker-recognition-evaluation-2012

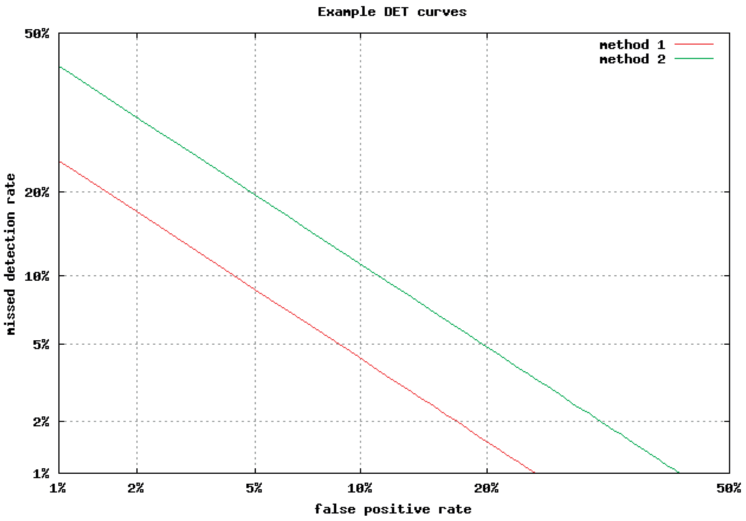

Figure 2.6: *The DET curves for two systems.*

set to 0.0001 while C_{Miss} is set to 0.01 (Martin and Greenberg, 2010; Greenberg et al., 2013; Brümmer, 2007; Brümmer and de Villiers, 2013).

2.10 Datasets

In this thesis, the experiments are performed for text-dependent and text-independent SV tasks. We evaluated the SV techniques primarily on female subsets since the results on female-evaluation data are usually significantly worse than for the male (Stafylakis et al., 2016). Below, we describe the datasets used in this thesis.

2.10.1 Text-independent SV

The Fisher (8 kHz) dataset is used as the training corpora (female)[3]. It consists of 13 k utterances with an average duration of an utterance of around 5 mins. The total duration of the training data is 1 k hours. A development data of about 100 utterances is used for evaluating the ASR performance (does not overlap with the training data). In all the experiments, the i-vectors are typically 400 dimensional (if not mentioned otherwise). The back-end classifier is trained using the NIST SRE 2004-2008 data (i.e. development data)[4,5,6]. It consists of 2.5 k speakers uttering 27 k audio recordings. LDA and PLDA models are trained on the development data using the speaker labels. The various SV systems are evaluated on NIST SRE 2010 evaluation set from conditions 1 to 5 (**Cond1** to **Cond5**) (Martin and Greenberg, 2010), where the evalution conditions are:

[3] https://catalog.ldc.upenn.edu/LDC2004S13
[4] https://catalog.ldc.upenn.edu/LDC2006S44
[5] https://www.nist.gov/itl/iad/mig/2008-nist-speaker-recognition-evaluation-results
[6] https://catalog.ldc.upenn.edu/LDC2011S10

- **Cond1**: Trials involving utterances from interview speech with matched microphones for enrollment and test. It contains a total of 33 k trials.
- **Cond2**: Trials that involve interview speech from different microphones for enrollment and test. It contains 118 k trials.
- **Cond3**: This condition involves trials that contain interview speech for enrollment and normal vocal effort conversational telephone test speech. It contains 31 k trials.
- **Cond4**: Trials involving interview speech as enrollment and normal vocal effort conversational telephone test speech recorded over a room microphone channel. It contains 45 k trials.
- **Cond5**: Trials involving normal vocal effort conversational telephone speech in enrollment and test speech. It contains 16 k trials.

2.10.2 Text-dependent SV

The SV are evaluated on these text-dependent SV tasks,

1. **Fixed-phrase**: The training data is drawn from Fisher English corpora (~120 h subset of female speakers). We used a subset of the Fisher data since we obtain similar performance regardless of training on whole dataset or subset. This subset of data contains 1.2 k utterances with an average duration of 5 mins per utterance. The choice of Fisher database as a training set was primarily motivated by the requirement of a well-transcribed and standardized data. The PLDA and JFA models are trained on a development set of RSR2015 (female).

   The Part1 (female) part of RSR2015 data contains 143 female speakers pronouncing 30 fixed passphrases spreading over nine sessions (Larcher et al., 2014b). Speakers are divided into three parts, background, development and evaluation portions. Data is collected from six different mobile devices with an average duration of 3 s. The development data contains 49 speakers with 12 k utterances. Evaluation data contains enrollment utterances which are recorded from a fixed mobile device while the test data comes from other devices. The number of speakers in the evaluation part is 47 with 8 k test utterances. All speech files are downsampled to 8 kHz for compatibility with other datasets used for system development.

   We also experimented with RedDots dataset on the fixed-phrase based text-dependent SV setup (Lee et al., 2015). The number of female speaker for RedDots is only 6 and the number of trials for female subset is very limited. The results of SV systems on female subset would not be statistically relevant due to the small number of trials, thus we perform experiments on the male subset only. The RedDots is more challenging than RSR2015 since it does not provide any development data from the same corpora. For the RedDots, the training data is drawn from the Fisher male (~120 h), similar to the above experimental setup. Since no development data was available for the experiments on RedDots, we choose the RSR2015, male data from Part1. The Part1 portion (male subset) of the RSR2015 dataset is used as the development data with 42 k utterances from 157 speakers. We evaluated our systems on the Part4 portion of RedDots database. The evaluation data of this dataset was distributed during the Interspeech 2016 Special session[7]. Compared to RSR2015, the RedDots contains more sessions of recording of speech data from each speaker. The dataset contains 52 sessions per speaker, with one session per week. Thus the challenge of the systems is to compensate for the long term intra-speaker variability (in addition to inter-speaker variability). We evaluated our system only on the male set of the database (Part4 text-dependent task only) (Lee et al., 2015). The Part 4 consists of 35 speakers pronouncing fixed-phrases (which are different from the phrases of the RSR2015 dataset). Similar to previous experimental setup, the speech files are downsampled to 8 kHz for compatibility with other datasets. It contains a total of 5 k target trials and 5229 k impostor trials.

2. **Seen**: We created the test set by following the protocol presented in Scheffer and Lei (2014) to evaluate our techniques. The data of each of the speakers involves 15 phrases with three sessions for each phrase, with a total of 45 utterances. The total duration of the enrollment of a speaker is 90 s. Test utterances consist of a speaker uttering phrases with a duration of 2 s. For this task, the evaluation trials consist of 4 k target and 211 k impostor trials. The Fisher female subset English is used as the training data since the evaluation is done on female data-set (as used for the fixed-phrase task). The Part1 of RSR2015 is used as the development data.

3. **Random-digit strings**: This subset contains 49 speakers pronouncing random sequence of digits. The standard protocol is adopted to perform text-dependent SV (Stafylakis et al., 2016; Larcher et al., 2014b). Three utterances (with an average duration of 12 s) are used for creating the enrollment model. The enrollment utterance consists of the speaker uttering a random sequence sequence of 10 digits. The test utterance consists of 5 digits with an average duration of 2 s. For this task, the evaluation trials consist of 5 k target and 253 k impostor trials. The Part 3 of RSR2015 *dev* portion was used as the development data. We used 3 k utterances consisting of 47 speakers pronouncing 10 digits.

The text-dependent SV systems are evaluated in three conditions (**Cond1** to **Cond3**). The conditions are:

- **Cond1**: The target speaker utters the wrong content,
- **Cond2**: The impostor utters the correct content, and

[7] https://sites.google.com/site/thereddotsproject/

- **Cond3**: The impostor pronounces the wrong content.

Finally, **Cond-all** combines the three text-dependent SV evaluation conditions. For **fixed-phrase** scenario, we evaluate the SV approaches in all the conditions. For **Seen** and **random-digit strings**, the techniques are evaluated only for **Cond2** since the other conditions require the system to perform utterance verification (which can only be done by an ASR) (Scheffer and Lei, 2014; Stafylakis et al., 2016).

2.11 System configuration

In this section, we describe the standard configurations of the features and various systems used in this thesis.

2.11.1 MFCC

MFCC features of 20 dimensions are extracted from 25 ms of frame of speech signal with 10 ms sliding window, appended with the delta and double delta features. STG is applied to the features using a 3 s sliding window (Motlicek et al., 2015). The VAD is based on a phone classifier (i.e. comparing the sum of posteriors over phone classes with the posterior of silence class to classify each frame as speech or non-speech). This is used to mark the start and end points of the speech region in the utterance.

2.11.2 i-vector and JFA configurations

Here, we describe the configuration of the baseline systems for text-independent and text-dependent SV. We use these configurations in all the chapters unless mentioned otherwise.

- **Text-independent SV**: A GMM-UBM with 2 k mixture components is trained on the Fisher data and i-vector extractor of 400 dimension is also trained on the same data. The i-vector dimension was reduced to 350 after LDA, followed by length normalization before being scored using PLDA.
- **Text-dependent SV**: For the **fixed-phrase** SV, we implemented gender-dependent GMM-UBMs (one male and one female) comprising 1 k mixture components trained using the Fisher subset (as described in Section 2.10.2). The parameters of i-vector extractors are estimated using the same training data as used for GMM-UBMs. The dimension of extractors is fixed to 400. The parameters of the JFA systems are estimated with speaker-phrase labels using the development data. The rank of the eigenchannel matrix **U** is fixed to 50. For the **random-digit strings** and **seen** tasks, the i-vector system on female data is used as the baseline.

3 Phonetic information for text-independent speaker verification

Contents

In this chapter we present approaches to incorporate phonetic information in the i-vector framework. This chapter is based on:

> Petr Motlicek, Subhadeep Dey, Srikanth Madikeri, and Lukas Burget. Employment of subspace Gaussian mixture models in speaker recognition. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4445–4449. IEEE, 2015

Srikanth Madikeri, Petr Motlicek, Marc Ferras, and Subhadeep Dey. Analysis of posterior estimation approaches to i-vector extraction for speaker recognition. Idiap-RR Idiap-Internal-RR-118-2016, Idiap, July 2016

3.1 Introduction

In the past, various approaches to incorporate phonetic information for SV have shown to improve the performance (Matsui and Furui, 1993, 1994; Ferras et al., 2007; Stolcke et al., 2005). In Matsui and Furui (1993, 1994), speaker-specific phoneme models are employed to authenticate the spoken text and speaker. In this approach, a hidden Markov model (HMM) based universal phoneme model is first trained by pooling data of all speakers. The parameters of the speaker-specific HMM models are re-estimated from the data of the class. During evaluation, the utterance is first decoded to obtain sequence of phonetic units. The speaker-specific models corresponding to the decoded-phonetic units are used to obtain SV scores. A similar approach is explored for GMM-UBM framework by Gutman and Bistritz (2002). In this approach, speaker-specific phonetic units are modelled by a GMM. Thus, each speaker model consists of a set of GMMs as opposed to a GMM. During evaluation, the test utterance is scored against all the phomene models to produce SV scores.

As opposed to using GMM-UBM, speech recognition based speaker adaptation techniques have been explored by Stolcke et al. (2005). In particular, they investigated the application of maximum likelihood linear regression (MLLR) transform as features for discriminating speakers. The MLLR transforms are estimated for the Gaussian mean vectors of the acoustic models using EM algorithm. The transformation matrix is converted to a high dimensional vector which is then used as input for the back-end for producing SV scores. The final classifier is a support vector machine (SVM), that aims to discriminate speakers using maximum-margin criteria.

The most successful application of phonetic information for SV is obtained in the i-vector framework (Lei et al., 2014). In the conventional i-vector approach, computing an i-vector for a given speech recording requires the sequence of short-term acoustic feature vectors, to be aligned with the Gaussian mixture components of a GMM-UBM. From the frame-to-mixture alignment, zeroth-, first- and second-order statistics are computed. The zero-th order statistics represent the effective number of feature vectors attributed to a particular mixture in the GMM-UBM. The first order statistics measure their deviation from the mixture mean while the second order statistics measure their variance around the mean. These statistics (so called sufficient statistics (SS)) are used to project the utterance onto a low dimensional subspace to obtain i-vector of an utterance. In Lei et al. (2014), the SS are computed using a DNN acoustic model that is trained in an ASR fashion. The results indicate that phonetic knowledge can be beneficial for performing SV. Motivated by the results, in this chapter we explore the application of ASR for SV. To this end, we investigate new approaches to compute SS directly from word-recognition lattices (used later for i-vector extraction). The application of SS from various ASR models, such as HMM/GMM, HMM/DNN, are investigated in this context as well. Furthermore, we investigate the use of subspace Gaussian mixture model (SGMM) employed to obtain speaker representations as opposed to using i-vectors (Motlicek et al., 2015). SGMM has been proposed in the context of ASR acoustic modeling approach based on GMM, where the parameters of the phonetic units are represented by a more compact set than

HMM/GMM (Povey et al., 2010). The speaker vectors computed from the SGMM framework can be applied directly as an input for subsequent PLDA modelling.

This chapter is organized as follows. In Section 3.2, the HMM/GMM based ASR is presented. This is then followed by a description of HMM/SGMM in Section 3.3. Section 3.5 describes the HMM/DNN framework. Section 3.6 describes the SV approaches built on top of ASR models. The experimental setup and results are presented in Sections 3.8 and 3.9 respectively. Finally, the chapter is concluded in Section 3.10.

3.2 HMM/GMM based ASR

Figure 3.1 shows the basic components of a typical ASR system. The task of an ASR is to produce a sequence of words ($\hat{\boldsymbol{\omega}}$) corresponding to an utterance ($\mathbf{X}$). We describe the basic elements of an ASR which are the following:

- **Acoustic model (AM)**: Each of the spoken words can be decomposed into smaller set of sound units, also known as phones. Each of the phone unit can be represented by a continuous density HMM. Typically, left and right context of every phone (tri-phone) units are employed as the basic unit of speech signal. The states of the tri-phone based HMMs are assumed to have Gaussian distribution and the states are tied to reduce the number of parameters. The context-dependent tied states (also referred to as *senones*) (Povey et al., 2011b) are obtained using a decision tree based on contextual and data-driven criteria.

- **Language model (LM)**: The language model is applied in an ASR system to generate a list of hypothesized words. Usually, a N-gram language model is used with the parameters are of the model are estimated on a large text-corpora.

- **Decoder**: The ASR decoder as shown in Figure 3.1 considers both AM and LM to generate most likely word sequence corresponding for each frame of the utterance. Mathematically, for an utterance with feature vectors $\mathbf{X}$, the decoder aims to produce a sequence of words $\boldsymbol{\omega}_{1:L} = \omega_1, \omega_2, \omega_3, \cdots, \omega_L$ as given by:

$$\hat{\boldsymbol{\omega}} = \arg\max_{\boldsymbol{\omega}}\{p(\mathbf{X}|\boldsymbol{\omega})\,p(\boldsymbol{\omega})\}. \tag{3.1}$$

  The quantity $p(\mathbf{X}|\boldsymbol{\omega})$ is referred to as likelihood and is computed using the AM, while $p(\boldsymbol{\omega})$ is referred to as prior probability of words and is computed using the LM. After decoding the utterance, usually word-recognition lattices are generated that compactly represent the most likely hypotheses of word sequences (Povey et al., 2011b).

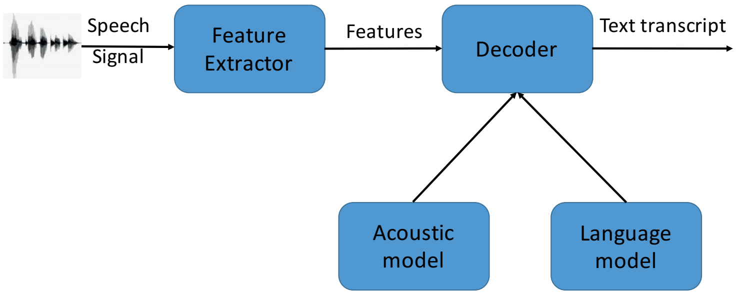

Figure 3.1: *Basic components of an ASR.*

3.3 Subspace Gaussian Mixture Model

In a different direction, SGMM has shown to achieve good ASR performance (Povey et al., 2010, 2011a; Povey, 2009). Figure 3.2 illustrates the SGMM technique for ASR. The SGMM method is an acoustic modeling approach in which a common GMM structure is shared across all the phonetic states. In this technique, the GMM mean supervector space is factorized into phonetic and speaker subspaces. While for ASR, the speaker subspace is constrained to have low dimensionality, the speaker vectors are set to have as many dimensions as used in the i-vector model. Each state is represented by a state vector that defines a mapping to the means and weights of the state's GMM. Let $\mathbf{x}$ be a F-dimensional feature vector, j represent a model state, $\mathbf{v}_j$ the S-dimensional state vector. The model of a state is defined by:

$$p\left(\mathbf{x}|j\right) = \sum_{i=1}^{i=I} w_{ji}\mathcal{N}\left(\mathbf{x};\boldsymbol{\mu}_{ji},\boldsymbol{\Sigma}_i\right), \tag{3.2}$$

$$\boldsymbol{\mu}_{ji} = \mathbf{M}_i\mathbf{v}_j + \mathbf{N}_i\mathbf{v}^s, \tag{3.3}$$

$$w_{ji} = \frac{\exp \mathbf{w}_i^t\mathbf{v}_j}{\sum_i^I \exp \mathbf{w}_i^t\mathbf{v}_j}, \tag{3.4}$$

where I is the number of Gaussians in the state, $\mathbf{M}_i$ and $\mathbf{w}_i$ are globally shared parameters. Typically, S is much less than $I(F+1)$ and hence the model is called "subspace" GMM. Each state j has M_j substates. The substates have their own mixture weights c_{jm} and vector $\mathbf{v}_{jm}$. The SGMM equations can be re-written as:

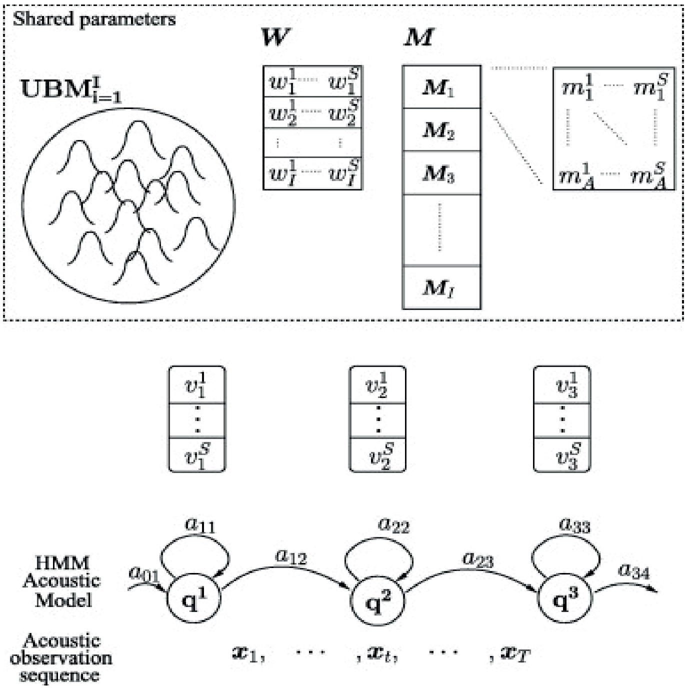

Figure 3.2: *SGMM – the emission probabilities of each context-dependent HMM-state $\boldsymbol{q}_j$ are modelled by GMM (where j is an index of HMM state) . Each HMM-state is parametrised by a vector $\boldsymbol{v}_j$. The parameters **M** and **W** are globally shared. The image has been taken from Imseng et al. (2014).*

$$p\left(\mathbf{x}|j\right) = \sum_{m=1}^{M_j} c_{jm} \sum_{i=1}^{I} w_{jmi} \mathcal{N}\left(\mathbf{x}; \boldsymbol{\mu}_{jmi}, \boldsymbol{\Sigma}_i\right), \tag{3.5}$$

$$\boldsymbol{\mu}_{jmi} = \mathbf{M}_i \mathbf{v}_{jm} + \mathbf{N}_i \mathbf{v}^s, \tag{3.6}$$

$$w_{jmi} = \frac{\exp \mathbf{w}_i^t \mathbf{v}_{jm}}{\sum_i^I \exp \mathbf{w}_i^t \mathbf{v}_{jm}}. \tag{3.7}$$

We refer to the speaker factor ($\mathbf{v}^s$) as sgmm-vector. In this chapter, we propose to apply sgmm-vectors to replace i-vectors in the i-vector PLDA framework.

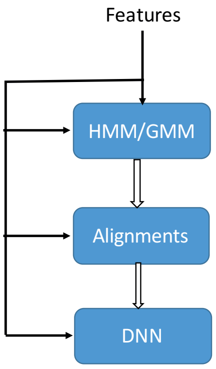

Figure 3.3: *DNN/HMM based ASR.*

3.4 HMM/SGMM based ASR

HMM/SGMM based ASR employs SGMM for acoustic modelling. Similar to HMM/GMM based ASR, HMM/SGMM modelling requires annotated data for training. Large amounts of annotated data help the underlying models to capture the phonetic and speaker variabilities in the data. To train the HMM/SGMM system for speaker recognition, the dimensionality of the speaker subspace is increased with respect to that of the phonetic subspace.

3.5 HMM/DNN based ASR

Acoustic models based on DNN have shown to significantly improved the ASR performance compared to the conventional HMM/GMM (Hinton et al., 2012). Figure 3.3 shows the training procedure of HMM/DNN based ASR. As implemented in the Kaldi recipe, the DNN training is usually done on top of the HMM/GMM, i.e. the decision tree and the senone alignments are obtained from the HMM/GMM based ASR (Povey et al., 2011b). The DNN is trained with senone units as target classes. The DNN takes a context of features as input and generate the senone posterior probabilities. We refer to this process as DNN forward pass (DNN FWD) (Povey et al., 2011b). For decoding an utterance, Equation 3.1 is applied, where the likelihood is obtained from the DNN FWD. The senone posteriors (or DNN outputs) are divided by the prior probabilities of senone units to compute likelihood. After decoding, word-recognition lattices are generated (similar to HMM/GMM) containing different word-hypotheses.

3.6 Senone posteriors for speaker verification

In Lei et al. (2014), it was shown that a DNN trained for ASR can replace the traditional GMM-UBM to estimate SS for i-vector extraction. The application of DNN FWD resulted in large performance gains for SV as better alignment is obtained with respect to the GMM-UBM components. The results showed that replacing unsupervised training of the GMM-

UBM components with well-defined acoustic classes can have a significant impact on SV performance.

3.6.1 Integration into i-vector framework

To integrate an ASR information into the i-vector framework, the parameters of the GMM-UBM are estimated from frame-level senone posterior probabilities (obtained by DNN FWD) (Lei et al., 2014). The parameters of the GMM-UBM are obtained as follows:

$$\pi_c = \frac{\sum_n \gamma_{n,c}}{\sum_c \sum_n \gamma_{n,c}} \tag{3.8}$$

$$\boldsymbol{\mu}_c = \frac{\sum_n \gamma_{n,c} \mathbf{x}_n}{\sum_n \gamma_{n,c}} \tag{3.9}$$

$$\boldsymbol{\Sigma}_c = \frac{\sum_n \gamma_{n,c} \left(\mathbf{x}_n - \boldsymbol{\mu}_c\right) \left(\mathbf{x}_n - \boldsymbol{\mu}_c\right)'}{\sum_n \gamma_{n,c}}, \tag{3.10}$$

where $\{\pi_c, \boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c\}_{c=1}^{c=K}$ are the parameters of the GMM-UBM, $\gamma_{n,c}$ is the posterior probability of the c^{th} senone unit generated by the DNN and $\mathbf{x}_n$ is the n^{th} feature vector. This GMM-UBM is then used to extract i-vectors.

3.7 Proposed analysis using ASR

The likelihoods converted from the senone posterior probabilities (of HMM/GMM, HMM/S-GMM and HMM/DNN models), along with word sequence probabilities from the LM, are passed to the decoder to obtain the ASR output. In Scheffer and Lei (2014), it was shown that a DNN trained for ASR can replace the traditional GMM-UBM to estimate SS for i-vector extraction. The posteriors obtained by DNN FWD process are directly used to compute SS. This technique resulted in large performance gains for SV systems, as better alignments are obtained with respect to the GMM-UBM components. The results showed that replacing unsupervised training of the GMM-UBM components with well-defined acoustic classes can have a significant impact on verification performance.

Although there has been sufficient evidence that phone-level classes possess speaker-discriminative information (Motlicek et al., 2015), successful integration into the state-of-the-art framework such as i-vector PLDA was not achieved until recently. The effectiveness of senone posteriors for i-vector extraction provides new research directions for SV.

Particularly, we seek to investigate whether we can take advantage of accurate senone alignments obtained by using the LM and the ASR decoder (Richardson et al., 2015; Su and Wegmann, 2016). The LM (together with a lexicon) not only offers more accurate alignments but may also help capture speaker-dependent characteristics closely related to the speech content, which we hypothesize is useful for better speaker discrimination.

We propose to study the estimation of SS from senone posteriors obtained at the output of the ASR decoder to take advantage of better senone alignments. Posterior vectors to estimate SS are obtained from the word recognition lattices (i.e., word LM is used in ASR engine to generate word recognition lattices). Eventually senone-level posteriors are extracted from these lattices similar to posterior vectors extracted with only acoustic models (e.g. DNN FWD). Even though the senone alignments are more accurate, they may need not result in better SV performance, because of their inherent sparsity. Such high sparsity arises as a result of smoothing the posterior vectors obtained from the DNN and smoothed by the ASR decoder based on word sequence probabilities from the LM. We show, through senone recognition rates, that this may not be favorable for SV systems given the nature of SS estimation. The contribution of senones is directly determined by not only their presence in the lattice generated by the ASR decoder, but also by the posterior values themselves. Extremely low values contribute little to the SS and may prove detrimental to the speaker recognition performance as they tend to have an effect similar to missing the senones altogether.

More specifically, although it can be expected that the SV should improve with better alignments, the posterior values per frame obtained from the lattices with the optimal AM and LM scaling parameters are extremely sparse. For instance, we observed that when the posteriors are thresholded, that is, posteriors less than a certain value (e.g. 10^{-5}) are floored to 0.0, the speech frame is no longer aligned to the true senone in $\approx 17\%$ of the frames (measured on Fisher dataset). Thus, even though the alignment obtained after decoding of HMM/DNN is more accurate compared to using only the posteriors after DNN FWD, such low scoring posteriors do not contribute to the SS. To deal with this problem, the likelihoods stored in word recognition lattices are first re-scaled prior to the forward-backward algorithm (Povey et al., 2011b). The best scaling was obtained when the AM was ~0.01 and the LM scale was 0.0. Other values for LM scale were also explored, but it proved beneficial to ignore the LM likelihoods once the recognition lattices are generated. The LM contribution is still available in the refined alignments provided by word recognition lattices. The proposed SV analysis using various ASR approaches (such as HMM/GMM, HMM/SGMM and HMM/DNN) is described in Figure 3.4.

3.8 Experimental Setup

SV experiments are conducted on the female data of NIST 2010 SRE in conditions 1 to 5. The Fisher corpora (female) is used as the training data while NIST SRE 2004 to 2008 are used as the development data. The details of the data are described in Section 2.10.1.

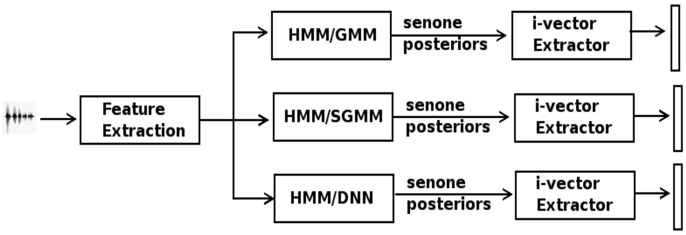

Figure 3.4: *Block diagram showing proposed SV techniques that use senone posteriors obtained from different ASR.*

3.8.1 Feature configuration and training data

The front-end uses 60 dimensional MFCC features along with delta and delta delta parameters as described in Section 2.11. All the ASRs employ a CMU dictionary with 42 k words and a 3 gram LM for decoding. The LM is trained on the Fisher data (female) with 1 k hours (Section 2.10.1).

3.8.2 HMM/GMM configuration

The HMM/GMM uses context-dependent triphone states with GMM observation probability density functions, and a total of 1’530 senones and 300 k Gaussians Gales and Young (2008). The number of senone units is automatically derived by the tree-clustering algorithm that is constrained to have around 2 k states in order to be comparable with the number of mixture components in GMM-UBM model. The HMM/GMM is used to generate senone posterior probabilities (as described in Section 3.5), which are then applied for i-vector extraction.

3.8.3 HMM/SGMM configurations

SGMM is trained with the same number of HMM states as HMM/GMM. Number of sub-states is roughly equal to the number of Gaussians in the HMM/GMM model. The phonetic subspace is constrained to a dimension of 40 (i.e. $S = 40$) while the speaker dimension is set to 400.

3.8.4 HMM/DNN configuration

The input to the DNN is 540 dimensional vector which is obtained by stacking 9 MFCC features. The DNN is trained to predict senone posterior probabilities. As mentioned in Section 3.5, HMM/DNN is usually trained with alignments from the HMM/GMM. We used the Kaldi toolkit to train a DNN, employing 6 hidden layers with 2 k sigmoid units per layer and softmax units at the output. The DNN parameters are initialized with stacked restricted Boltzmann machine that are pretrained in a greedy layer-wise fashion (Dahl et al., 2012). The baseline i-vector extractor is trained by extracting SS using DNN FWD (Lei et al., 2014).

Table 3.1: *ASR results on Fisher development set in WER (%) .*

System	WER (%)
HMM/GMM	42.3
HMM/DNN	26.0
HMM/SGMM	31.1

Table 3.2: *Frame based senone accuracies on Fisher development set for different acoustic models.*

ASR	SRA (%)
HMM/GMM	55.2
DNN forward pass	53.5
HMM/DNN (decoder)	73.4

3.8.5 i-vector approach

In this chapter, the i-vectors are computed by exploiting senone posteriors from HMM/GMM, HMM/SGMM, and HMM/DNN models (in addition to GMM-UBM and DNN FWD). The i-vector dimension is set to 400 in all the approaches. LDA and PLDA are applied on top of i-vectors as described in Section 2.11.

3.8.6 ASR results

The performances of the ASR approaches, namely the HMM/GMM, HMM/SGMM and the HMM/DNN, are compared in Table 3.1 in terms of word error rate (WER). The ASR systems are evaluated on a subset of the Fisher dataset (as described in Section 2.10.1). As expected, the WER is lower for the HMM/DNN.

3.8.7 Senone recognition accuracies

In this section, the frame based senone recognition accuracies (SRA) of various ASR approaches are analyzed. The performances are presented in Table 3.2. The SRA is the percentage of senones correctly identified according to the groundtruth (which is obtained by forced aligning the reference transcription using HMM/GMM). Typically Viterbi algorithm is applied for obtaining forced-alignment of an utterance Povey et al. (2011b). A speech frame is considered correctly identified if the highest senone posterior probability matches with the groundtruth. As expected, the SRA improves with better acoustic modelling and is the best when an ASR decoder is used with the word LM.

Table 3.3: *Comparison of speaker recognition performance in terms of EER (%) when using different senone posterior probabilities, namely UBM-GMM, DNN and SGMM.*

Systems	Cond1	Cond2	Cond3	Cond4	Cond5
$\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{GMM}}$	1.4	2.4	1.6	1.3	2.2
$\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{HMM\text{-}dec}}$	0.8	1.7	1.3	0.7	1.3
$\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{DNN}}$	**0.6**	**1.1**	**0.7**	**0.5**	1.0
$\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{DNN\text{-}dec}}$	0.8	1.4	**0.7**	**0.5**	**0.9**
$\mathbf{SGMM}_{\mathbf{PLDA}}$	1.3	2.4	2.1	1.2	2.0
$\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{SGMM\text{-}dec}}$	1.2	2.3	1.6	1.2	1.6

3.9 Results

The following SV approaches are explored in this section:

- $\mathbf{Ivec}_{\mathbf{PLDA}}$: The i-vector PLDA as described in Section 2.5. The i-vector techniques that use SS from GMM-UBM or DNN FWD are referred to as $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{GMM}}$ and $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{DNN}}$ respectively. The i-vector PLDA that employ SS from HMM/GMM, HMM/DNN or HMM/SGMM are referred to as $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{HMM\text{-}dec}}$, $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{DNN\text{-}dec}}$and $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{SGMM\text{-}dec}}$, respectively.

- $\mathbf{SGMM}_{\mathbf{PLDA}}$: SGMM is developed to obtain speaker vectors (sgmm-vectors) as opposed to using i-vectors. A PLDA is trained on these vectors.

The results on five conditions (Cond1 through Cond5) of NIST SRE 2010 dataset are presented in Table 3.3. Both EER and minDCF values are reported. The baseline approach is the conventional i-vector PLDA as described in Section 2.5. For matching microphone condition (Cond1), the EER of $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{GMM}}$ is already as low as 1.4%. For mismatched condition that have a large number of trials, such as Cond2, the EER is 2.4%.

It can be observed from Table 3.3 that the senone posteriors obtained from the HMM/GMM word-recognition lattices benefit the SV. Although the framework for integrating acoustic class-based posteriors from ASR already exist, these results have seldom been reported. For $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{HMM\text{-}dec}}$, significant improvements are observed for all conditions compared to $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{GMM}}$. Absolute improvements in EER for Cond5 of up to ~0.9% are obtained by $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{HMM\text{-}dec}}$ compared to $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{GMM}}$. This translates into an improvement of relative EER of ~41% (from 2.2% to 1.3% absolute). Thus, even with a less powerful ASR, it is possible to achieve considerable SV improvements. The results clearly demonstrate the significance of constraining the acoustic space using additional knowledge (provided by LM) through ASR although the availability of large amounts of manual annotated data has its cost.

The $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{DNN}}$ presented in Table 3.3 is, in principle, similar to the system presented in Lei et al. (2014). Compared to $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{GMM}}$, relative EER improvement of ≈**57**% (from 1.4% to 0.6% absolute) for Cond1. $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{DNN\text{-}dec}}$ provides the best performance in Cond5 where the EER is as low as 0.9%. A comparison between $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{DNN}}$ and $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{DNN\text{-}dec}}$ reveals that significant performance gain

Table 3.4: *Performance of the best performing SV techniques (from Table 3.3) in terms of EER (%)/minDCF (×100) for Cond5 of NIST SRE 2010.*

Systems	Cond5
$\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{GMM}}$	2.2/0.28
$\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{DNN}}$	1.0/0.16
$\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{DNN\text{-}dec}}$	**0.9/0.12**

can be achieved by exploiting the ASR decoder output. $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{DNN\text{-}dec}}$ improved over $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{DNN}}$ by relative EER of 10% for Cond5 (from 1.0% to 0.9% absolute). $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{DNN\text{-}dec}}$ clearly improved over $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{GMM}}$ with relative EER of 43% (from 1.4% to 0.8% absolute) for Cond1 and 56% (from 1.6% to 0.7%) for Cond3.

In the results discussed so far, a strong correlation between the SRA as presented in Table 3.2 and the EER, especially for the telephone condition (Cond5), can be seen. The EER decreases with the increase in SRA suggesting that better initial alignment can lead to better speaker modelling.

Next, the performances of $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{SGMM\text{-}dec}}$ and $\mathbf{SGMM}_{\mathbf{PLDA}}$ are presented. The performance of $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{SGMM\text{-}dec}}$ is consistently better than $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{GMM}}$ for all conditions except Cond3. In particular, absolute improvement in EER of 0.6% is obtained on Cond5 by $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{SGMM\text{-}dec}}$ over $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{GMM}}$. The $\mathbf{SGMM}_{\mathbf{PLDA}}$ outperforms the $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{GMM}}$with absolute EER of 0.1% and 0.2% for Cond1 and Cond5 respectively.

3.9.1 Summary of experiments on NIST SRE 2010

The minDCF and DET curve for three best performing SV approaches are presented in Table 3.4 and Figure 3.5 for Cond5. The systems include, (i) $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{GMM}}$, (ii) $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{DNN}}$, and (iii) $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{DNN\text{-}dec}}$. It can be observed from Table 3.4 that $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{DNN\text{-}dec}}$ is the best performing system in terms of minDCF on Cond5 with relative improvement of 57% over $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{GMM}}$.

3.10 Conclusions

In this chapter, we explore the application of phonetic information in the i-vector framework. We improved on the existing technique that uses DNN FWD by applying word-recognition lattices from ASR to compute SS. The SS is eventually used for extracting i-vectors. Our results indicate that computing SS from lattices can benefit the SV. We also showed that the performance gains are positively correlated to the senone recognition accuracy of the models. In particular, the $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{DNN\text{-}dec}}$ outperforms the $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{DNN}}$ in Condition 5 of NIST SRE 2010 by absolute EER of 10%.

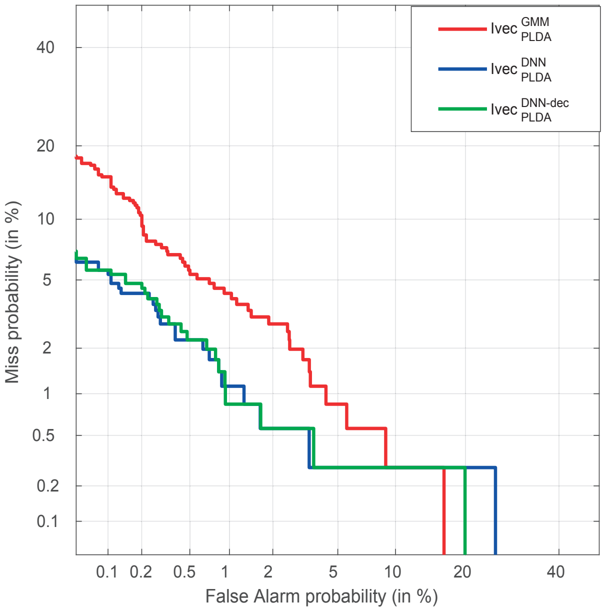

Figure 3.5: *DET curve of the systems presented in Table 3.4 for Cond5 of NIST SRE 2010.*

4 Template-matching for phrase based text-dependent system

Contents

In this chapter, we present approaches to exploit phonetic information using template matching algorithm for fixed-phrase based text-dependent speaker verification. This chapter is based on the following publications:

Subhadeep Dey, Srikanth Madikeri, Marc Ferras, and Petr Motlicek. Deep neural network based posteriors for text-dependent speaker verification. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5050–5054. IEEE, 2016a

Subhadeep Dey, Petr Motlicek, Srikanth Madikeri, and Marc Ferras. Exploiting sequence information for text-dependent speaker verification. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5370–5374. IEEE, 2017a

Subhadeep Dey, Petr Motlicek, Srikanth Madikeri, and Marc Ferras. Template-matching for text-dependent speaker verification. *Speech Communication*, 88: 96–105, 2017b

Table 4.1: *Types of trials in phrase based text-dependent speaker verification.*

	Correct Phrase	Wrong Phrase
Target Speaker	Target-Correct	Content-mismatch
Impostor Speaker	Speaker-mismatch	Content and speaker mismatch

4.1 Fundamental tasks

In the past few years, the state-of-the-art SV systems have shown to provide high performance for long duration speech recordings (Dehak et al., 2011; Garcia-Romero, 2012). In practical applications (forensics, biometrics, etc.), SV is often applied on short duration test utterances. However, results of the SV systems on short duration test set are yet to reach acceptable range of performance of any deployable system (Motlicek et al., 2015). Unlike unconstrained scenarios, application of SV systems on constrained content of the test utterances can bring reasonable performance. This is referred to as text-dependent task. Real applications have usually employed phrases, digits and short commands to constrain the content (Larcher et al., 2014b,a). In this chapter, we focus on text-dependent SV with phrases being shared across speakers. For example, in a text-dependent application, the user is expected to utter the phrase "My voice is my password" for authentication.

Phrase-based text-dependent SV involves the authentication of a claimed identity against a speaker speaking a known phrase. This phrase can be speaker-specific or common to all speakers and the phrase spoken by the speaker during enrollment phase may be different from the test phrase (Larcher et al., 2014b). In this thesis, we consider the scenario where the phrases chosen by the system during testing have already been uttered by the speaker during enrollment. As shown in Table 4.1, the system accepts a claim by recognizing both the speaker (based on its acoustic characteristics) and the phrase content of a speech utterance. In other words, impostor trials can be divided into three categories: (i) the content (phrase) does not match, (ii) the speaker does not match, and (iii) neither the speaker nor content matches.

State-of-the-art text-dependent SV systems are able to exploit text constraints to obtain high recognition accuracy (Kenny et al., 2014b,a). These systems are inspired by text-independent techniques such as i-vector and JFA being tailored to the text-dependent SV task. Besides intra-speaker and inter-session variabilities, text-dependent SV systems also need to deal with content variability.

Content or linguistic information is relevant to text-dependent SV based baseline systems as accept/reject decisions are directly linked to it. Content information has been introduced into conventional SV systems by computing SS from the DNN to obtain latent-vectors (Scheffer and Lei, 2014; Lei et al., 2014). Experiments on the standard database indicate superior performance of the baseline systems (Chen et al., 2015b; Larcher et al., 2014a, 2013). Even though conventional approaches explicitly model phonetic variability of content for text-dependent task, sequence information for the content variability is still ignored. Considering that content information can be decomposed into phonetic units (PU) and its sequence, i.e. the phone sequence information (PSI), standard i-vector and JFA systems obtain the same verification score for any permutation of the PSI. For the phrase “OK Google”, which comprises the sequence of phones /əʊˈkeɪ’ɡuːɡᵊl/, the permutation /ˈgu:gᵊləʊˈkeI/, in principle, would be expected to obtain the same score. This is due to the fact that SS depend only on the average feature characteristics in the i-vector and JFA frameworks. In this chapter, we present techniques that exploit both PU and PSI. To this end, we apply template matching technique, i.e. dynamic time warping (DTW), which has shown to perform well for text-dependent SV (Jelil et al., 2015). Compared to applying conventional spectral features in the DTW algorithm, posteriors extracted from DNN and GMM-UBM have been successfully used. It has been observed that DTW using DNN posterior features provides good performance in the content-mismatch conditions probably due to DNN posteriors are better at predicting phones (Dey et al., 2016a). However, this system performed poorly in the speaker-mismatch condition, probably due to content-discriminative features being computed using a DNN. In this chapter, we propose to incorporate speaker-informative features generated by an i-vector system to DTW algorithm.

This chapter is organized as follows. We first describe the baseline systems for phrase based text-dependent task in Section 4.2. The proposed template matching technique is described in Section 4.3. In Sections 4.4 and 4.5 describe the experimental setup and results respectively. Finally, the chapter is concluded in Section 4.6.

4.2 Baseline system

The GMM-UBM system as described in Chapter 2 has shown to be effective for phrase based text-dependent SV (Kenny et al., 2014a; Bhattacharya et al., 2016). The MAP technique on HMM-GMM system has also been shown to be powerful modelling technique for this task (Wang et al., 2016; Zeinali et al., 2017). We refer to this approach as MAP-GMM-HMM and it is decribed in this section. As shown in Figure 4.1, the MAP-GMM-HMM consists of creating a background model by a set of HMMs, where each HMM models a tri-phone units of the speech. The background HMM-GMM models is obtained by pooling data of all speakers in a supervised manner. Each of the HMM state represents context-dependent tied state (or senones), which are obtained by a data-driven process and a decision tree. This HMM-GMM system can be applied to obtain alignment of the training data. This model is also referred to as speaker independent (SI) model.

In literature, various speaker adaptation techniques have been investigated for ASR applications. The most common adaptation scheme, referred to as MAP adaptation of HMM-GMM, is considered in this chapter. In MAP adaptation, the background HMM-GMM is used to obtain adapted model as given by:

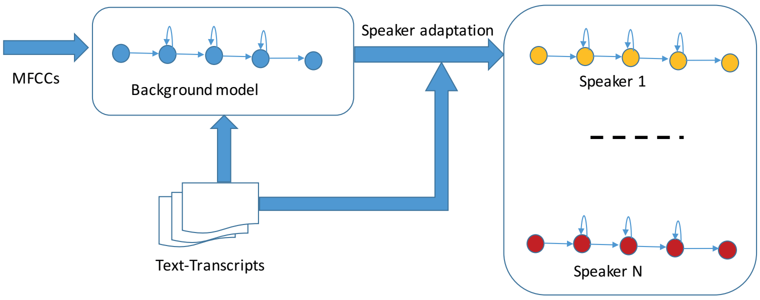

Figure 4.1: *Speaker adaptation in a HMM-GMM using MAP criteria.*

$$\lambda^{MAP} = \arg\max_{\lambda} \log(p(\lambda|\mathbf{X})) \propto \log p(\mathbf{X}|\lambda) + \log(p(\lambda)),$$

where λ, $\mathbf{X}$ refer to the parameters of the HMM-GMM models and the feature vectors respectively. In practice, the means of the HMM-GMM models are only adapted as given by:

$$\boldsymbol{\mu}_{j,m}^{HMM} = \frac{\tau \boldsymbol{\mu}_{j,m}^{HMM,0} + \sum_{t=1}^{T} \gamma_{j,t} \mathbf{x}_t}{\tau + \sum_t \gamma_{j,t}},$$

where $\boldsymbol{\mu}_{j,m}^{HMM}$ is the adapted mean of the m^{th} Gaussian of the j^{th} tri-phonetic unit, $\mu_{j,m}^{HMM,0}$ is the corresponding mean vector of the background model, τ is a constant factor and $\gamma_{j,t}$ is the posterior probability of mixture m of j^{th} HMM state.

During evaluation, the likelihood (ℓ_M)of the test utterance ($\mathbf{X}$) is computed against the speaker model (λ_M) and the background model as follows:

$$\ell_M(\mathbf{X}) = \log(p(\mathbf{X}|\lambda_M)) - \log(p(\mathbf{X}|\lambda_{UBM})). \tag{4.1}$$

Assuming text-transcript of the test data is available to us during evaluation, the likelihoods of Equation 4.1 ($\log(p(\mathbf{X}|\lambda_M))$ and $\log(p(\mathbf{X}|\lambda_{UBM}))$) can be computed against the acoustic models (λ_M and λ_{UBM}).

In addition to MAP-GMM-HMM, the i-vector and JFA have been shown to provide good performance for this task (Kenny et al., 2014b,a; Chen et al., 2015b). In the previous chapter, we described an approach to incorporate phonetic information in the i-vector framework by replacing the GMM-UBM by a DNN. This same technique can be extended for JFA model by computing posteriors of phonetic units to obtain the speaker factors. In this chapter, we

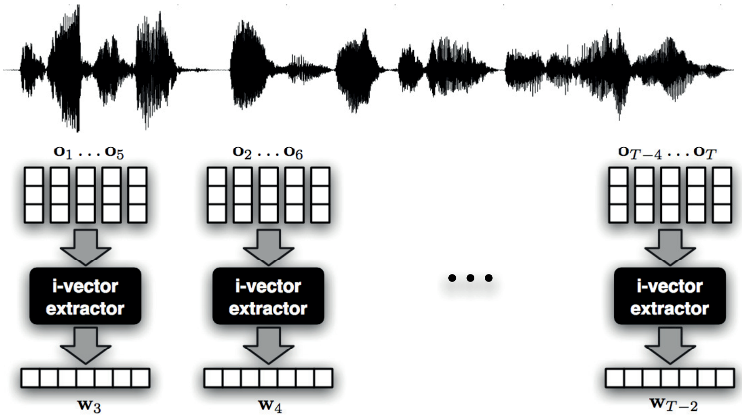

Figure 4.2: *Extraction of online i-vectors.*

consider the model-based approaches, such as i-vector, GMM-UBM, JFA and MAP-GMM-HMM, as the baseline systems.

4.3 Template matching

DNN-based approaches to i-vector/JFA modeling use PU information as target classes. However, the PSI of the phrase is ignored. We believe that exploiting the PSI in addition to PU will further improve performance, as text constraints for the task are being considered (Larcher et al., 2008). One approach to implicitly use PSI in i-vector system is by estimating senone posteriors obtained from after ASR decoding. These posteriors capture the long term context of speech signal as it is computed from decoded output (using LM and lexical model) (Su and Wegmann, 2016).

An alternative method to use the PSI is to model the idiosyncrasies of the speaker. A speaker not only has distinctive acoustic features but uses language in a characteristic manner, also called idiosyncrasies (Amino et al., 2006). These distinctive patterns of the speaker are usually expressed in terms of usage of words, phonemes (Shriberg, 2007; Campbell et al., 2003). In Campbell et al. (2003), PSI was used to estimate phone N-gram frequency. However, these approaches are mainly used as a source of high-level speaker-dependent features. As such, they have been used to enhance the performance of acoustic-based SV systems.

In a different direction, the spectral vectors of the speech signal, consisting of a specific phone sequence, have been used with DTW algorithm (Jelil et al., 2015; Dey et al., 2016a). This approach was shown to be effective for matching sequence of features and outperforms the model-based SV systems in content-mismatch conditions (Dey et al., 2016a), while in speaker mismatch condition, it provides reasonable accuracy. Motivated by the achieved results and the fact that DTW has not been investigated well enough after the emergence of subspace based techniques, we intend to further explore the DTW technique to address text-dependent SV problem.

4.3.1 Dynamic Time Warping

The DTW algorithm is a dynamic programming technique to compute the distance between two sequences of spectral vectors of arbitary length, and is commonly applied in query-by-example spoken term detection and other data mining tasks (Chen et al., 2015a; Keogh and Ratanamahatana, 2005). Being a non-parametric approach, it is well-suited for limited- or zero-resource tasks (Versteegh et al., 2015). The algorithm takes two sequences of features as input and finds the minimum cost mapping between them. The procedure involves computing all possible local distances between the two sequences (within a given range) and then back-tracking along the optimal path in terms of minimum distance (Brown and Rabiner, 1982). The DTW system performs well for the text-dependent SV task, especially for content-mismatch trials, due to the constraint in the spoken phrase.

In a conventional DTW system, MFCCs are used as input features to the DTW algorithm for performing text-dependent SV (Das et al., 2006; Bonastre et al., 2003). Besides MFCCs, senone posteriors have also been used as features to the algorithm (Dey et al., 2016a) by replacing Euclidean distance by the Kullback-Leibler (KL) divergence measure. Impressive gains were obtained with respect to a state-of-the-art i-vector system on content-mismatch conditions, while on speaker-mismatch trials, the system performs reasonably well (Dey et al., 2016a). As expected, the results indicate that these features might not contain enough speaker information to address a speaker recognition task. In the speaker-mismatch condition, the i-vector and JFA approaches performed considerably better than the DTW system. In view of these results, we propose to introduce speaker-informative features in the DTW algorithm. An i-vector system is used to extract these features. As opposed to the conventional approach of estimating i-vector for a whole utterance (2.5 mins for text-independent and 3 s for text-dependent systems), we propose to compute i-vectors on short segments of speech around 200ms. These features have also been referred to as online i-vectors (Peddinti et al., 2015; Madikeri et al., 2015).

4.3.2 Online i-vector features

The online i-vector features have been recently used for speech recognition and speaker diarization tasks, where they have shown promising results (Peddinti et al., 2015; Madikeri et al., 2015). In ASR, online i-vectors have been used for the purpose of adapting neural networks to speakers (Peddinti et al., 2015). In this case, online i-vectors are used as an input to the neural network, in addition to spectral features, to enhance speaker-specific information. The results obtained by this approach indicate that online i-vectors contain sufficient speaker information to improve ASR performance.

Online i-vectors have also been applied for the speaker diarization task within the Information Bottleneck (IB) framework for speaker clustering (Madikeri et al., 2015; Vijayasenan et al., 2011; Tishby et al., 2000). In this work, online i-vectors were appended to MFCC features to be fed into the speaker clustering algorithm. The additional gain in performance obtained by this approach compared to using only the spectral features suggests that the online i-vector representation carries speaker information as well. Motivated by the progress in content and speaker oriented tasks, we propose using online i-vectors as features for DTW systems. We now proceed to describe the method to apply online i-vectors.

Figure 4.2 illustrates the process of extracting online i-vectors from the speech signal. Let the speech utterance contains 'T' frames of speech given by $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_\mathrm{T}\}$, where $\mathbf{x}_t$ is the t^{th} speech frame. The online i-vector corresponding to t^{th} speech frame of an utterance is computed with a context size of L frames. The SS are computed on the sequence of speech frames, starting from t - L to t + L, for obtaining t^{th} feature vector. For a context size L = 10 frames, a sliding window of 21 frames is used with a shift step of 1 frame. Windows are centered at each frame in the utterance, which results in fewer frames being considered at the utterance boundaries. The corresponding sequence of online i-vectors is represented by $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_\mathrm{T}\}$ for an utterance. To compare two sequences of online i-vectors, the DTW algorithm is used with the cosine distance metric as given by:

$$d(\mathbf{w}_i, \mathbf{w}_j) = 1 - \frac{\mathbf{w}_i' \mathbf{w}_j}{||\mathbf{w}_i||\,||\mathbf{w}_j||},$$

where $\mathbf{w}_i$ and $\mathbf{w}_j$ are two i-vectors, $d(\mathbf{w}_i, \mathbf{w}_j)$ is the cosine distance between them and ||.|| represents the vector norm.

DTW scores computed on online i-vectors are expected to reflect both content and speaker similarities between enrollment and test templates. A window length of 200 ms, corresponding to average syllable duration, is able to capture both types of information.

4.3.3 PLDA projection features

A channel compensation model, such as PLDA, is usually applied on top of i-vectors in text-independent SV systems. The PLDA model produces verification scores by comparing two i-vectors. We apply the PLDA model on top of online i-vectors as we believe that it will help to factor out unnecessary channel information from the features. Training a PLDA model for the SV task uses speaker labels to define a set of classes to be discriminated. It is common to have multiple instances of speaker labelled i-vectors available for large text-independent datasets (Garcia-Romero and McCree, 2014; Lei et al., 2014). For a text-dependent scenario, the outcome of the task is linked to identifying content and speaker. This motivates the use of speaker-content classes for PLDA training (Dey et al., 2016a; Larcher et al., 2014a, 2013). Besides labelling content as whole phrases, phone classes can be obtained from a forced alignment of the data against given transcripts as well. Speaker labels are typically available as meta-data provided as part of the dataset. In this work, we experiment with both speaker-phrase and speaker-phone labels for training the PLDA hyperparameters on online i-vectors. PLDA is usually trained with speaker-phrase labels for text-dependent SV task (Dey et al.,

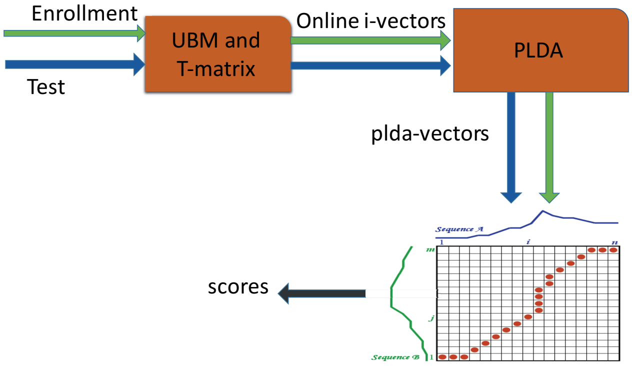

Figure 4.3: *The proposed system for **fixed-phrase** based text-dependent SV.*

2016a; Larcher et al., 2014a, 2013). We now describe the training procedure for PLDA with speaker-phone labels only.

The sequence of online i-vector features is extracted for q^{th} utterance of speaker s_k, which is represented by $\mathbf{W}_q^{s_k} = \{\mathbf{w}_{1,q}^{s_k}, \mathbf{w}_{2,q}^{s_k}, \cdots, \mathbf{w}_{\mathrm{M},q}^{s_k}\}$. The HMM/DNN based ASR system is used to align the speech signal with respect to the senone classes, which are then mapped to obtain the phone labels. We create a set of P phone classes for the speaker (s_k) $(\{D_1^{s_k}, D_2^{s_k}, D_3^{s_k}, \cdots, D_P^{s_k}\})$ for training the PLDA model, with the online i-vector $\mathbf{w}_t^{s_k} \in D_r^{s_k}$ if t^{th} MFCC feature of the utterance is aligned to r^{th} monophone. In a database with S speakers, we have $S \times P$ classes for training the PLDA model. In a phrase based SV, speaker-phonetic variability is useful for exploiting the text constraints of the task.

DTW uses online i-vectors after projection onto the inter-class PLDA subspace, also called PLDA projections. The cosine distance between enrollment and test templates is used for this purpose. In this process, PLDA compensates for variabilities other than speaker-content, such as channel variability.

The PLDA projections have been successfully used in related speech processing tasks such as speaker diarization and domain adaptation (Dey et al., 2016b; Madikeri et al., 2015). A reasonable gain in performance for speaker diarization is observed as compared to the system using only i-vector, which suggests that the PLDA model has enhanced the speaker representation of i-vectors (Madikeri et al., 2015).

The PLDA projection features are obtained as follows. From the PLDA model of Equation 2.20, the probability distribution of the speaker-phonetic factor is given by:

$$p(\boldsymbol{v}|\mathbf{w}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{v}}, \boldsymbol{\Sigma}_{\boldsymbol{v}}), \tag{4.2}$$

where the $\boldsymbol{\mu}_{\boldsymbol{v}}$ is the mean and $\boldsymbol{\Sigma}_{\boldsymbol{v}}$ is the covariance matrix of the Gaussian distribution. The mean is given by

Table 4.2: *Performance of the DNN and adapted-DNN (female) based ASR on RSR2015 and Fisher subset in terms of WER (%).*

Systems/Conditions	Fisher	RSR2015
DNN	24.5	85.0
adapted-DNN	28.2	17.1

$$\boldsymbol{\mu_v} = \boldsymbol{\Sigma_v}\boldsymbol{\Pi}\mathbf{A}^{-1}(\mathbf{w} - \boldsymbol{\mu}_{\mathbf{w}}), \tag{4.3}$$

where **A** is the covariance matrix of the error term of Equation 2.20 and **I** is the identity matrix. The covariance matrix ($\boldsymbol{\Sigma_v}$) is given by

$$\boldsymbol{\Sigma_v} = (\mathbf{I} + \boldsymbol{\Pi}^T\mathbf{A}^{-1}\Pi)^{-1}.$$

In this chapter, we refer the mean of the Gaussian distribution ($\boldsymbol{\mu_v}$) as the PLDA projection feature or plda-vectors of Figure 4.3 (the point estimate of the posterior distribution of the speaker-phonetic factor), which is subsequently applied in the DTW framework. The PLDA projection vector of a frame of speech is obtained by first computing the online i-vector and then projecting in the PLDA subspace as given by the Equation 4.3. Thus for an utterance, the number of PLDA-projection features is same as the speech frames. The proposed system is illustrated in Figure 4.3 where the final DTW score is applied for evaluating system.

4.4 Experimental Setup

Experiments are conducted on the RSR2015 (Part1, female) and RedDots (Part4, male) as described in Section 2.10.2. The details of the features, i-vector and GMM-UBM system are described in Section 2.11. The SV approaches are evaluated in three conditions, namely, (i) Cond1: content mismatch, (ii) Cond2: speaker mismatch, (iii) Cond3: speaker and content mismatch, and (iv) Cond-all: combining all conditions (Cond1 to 3), following the protocol in Larcher et al. (2014b).

4.4.1 HMM/GMM based MAP system configurations

Two separate phone based HMM/GMM acoustic models (male and female) are trained in a supervised manner with Fisher subset (~ 120 hours) as described in Section 2.10.2. Both the systems use 43 phones with a total of 2 k Gaussians.

4.4.2 HMM/DNN system configurations

The DNN, usually trained in ASR fashion, is employed to compute the posteriors of the senone units, which is then used in the DNN-based i-vector and JFA systems parameters estimation process. These posteriors are also used as feature streams in DTW systems. Two gender dependent ASR systems are trained for experiments, one male and another female, with their respective training data (as mentioned in the Section 2.10.2.

We now proceed to describe the ASR setup as used in Motlicek et al. (2015). Since the parameters of the two ASR systems are the same, we describe the configuration of one system (female) only. The HMM/GMM system (female) uses context-dependent tri-phone states and a total of 1.5 k senone states and 12 k Gaussians. This system is used to obtain senone alignments to train the DNN model. The DNN is trained with MFCC input features and a context size of 5 frames. It comprises 4 hidden layers with 1.2 k sigmoid units per layer. The output of the DNN is represented by softmax function. It is trained with stochastic gradient descent algorithm to minimize the cross-entropy function between the class labels (senone alignments) and the network output. After the convergence of the algorithm, the posterior probabilities of the senone units corresponding to an input speech frame are obtained at the output of the DNN.

4.4.3 ASR performance

The conventional hybrid ASR system uses DNN to estimate acoustic posterior probabilities plugged into the ASR decoder by employing LM. The performance of the female ASR system is evaluated on two batches of data, namely, (i) Fisher female subset with 200 utterances and, (ii) Part1, RSR2015 female subset consisting of 1 k utterances. The ASR system employs a CMU dictionary with 42 k words and a tri-gram LM for decoding with word LMs (Motlicek et al., 2015). The LM is trained on the transcript of Fisher subset (~ 120 hours). The WER on both the set are presented in Table 4.2. The WER of the female DNN is 24.5% on the Fisher subset. Poor performance on the RSR2015 subset is possibly due to acoustic mismatch between the RSR2015 and the training dataset (channel, accent mismatch).

In order to cope with large differences in performance of WER, we adapt the DNN with a small amount of data (~1 h) from RSR2015 database. In a DNN framework, it is usually done by adapting the weights of one of the layer keeping others layers fixed. The weights of the last layer of the DNN are adapted using a limited amount of transcribed in-domain data with the senone-discriminative backpropagation algorithm. The adapted-DNN provides better ASR results on the evaluation data than the DNN trained in resource rich domain. Thus we believe that the better ASR system will help in SV process. From Table 4.2, it can be observed that the adapted-DNN performs roughly equally well in both the databases (row 2 of Table 4.2) with absolute improvement of ~68% in terms of WER on the RSR2015 dataset. The DNN and the adapted-DNN (trained on the female portions) are then used for SV experiments on RSR2015 Part1, female evaluation set only.

The performance of the male-DNN is evaluated only on a Fisher male subset (200 utterances). The WER of this DNN is 30.5%. Since no development data is available from RedDots dataset, the adaptation of DNN could not be done.

4.4.4 Online i-vector configurations

Two online i-vector systems are developed (for male and female) using the training data as described in Section 2.10.2. Since the parameters of both the systems are similar, we describe the configurations of the female system only. The SS, required to estimate online i-vectors, are computed from short segments of speech signal of duration 200 ms. The i-vector extractor is 400 dimensional. To train the speaker-phone PLDA model, the ASR system developed in the previous subsection is used to obtain senone alignments. The senones are then mapped to one of 43 monophones to get the phone alignment. The PLDA is trained on the online i-vectors by assigning speaker-phone pair labels to each of the speech frames. The Part1 of RSR2015 dataset is used to train the PLDA. There are a total of 2 k classes (speaker-phone pairs) in the development set.

4.5 Results

In this section, we describe the results obtained with various systems described in Sections 4.2 and 4.3. We refer to the $\mathbf{MAP^{GMM}}$, $\mathbf{Ivec_{PLDA}}$ and **JFA** as the model-based systems. We first present the results on the RSR2015 dataset (Part1, female) and then proceed to RedDots (Part1, male). The conventional approaches include the DTW and model-based SV systems (MAP, i-vector and JFA) both employing GMM posteriors. Since it has been consistently reported in literature that MAP technique outperforms other approaches for text-dependent SV task (Kenny et al., 2014a,b), we consider the MAP system to act as the baseline system in both the experiments on RSR2015 and RedDots. In all the experiments involving PLDA, the input vectors to the model are length normalized. For the MAP, JFA and DTW systems, T-norm score normalization is applied (Barras and Gauvain, 2003; Dey et al., 2016a; Kenny et al., 2014b,a). In our experiments involving i-vectors, we observed that dimensionality reduction technique, like LDA, degraded the performance of the speaker recognition system. Thus, we do not report the performance of the systems using LDA transform. In all the experiments, the senone posterior probabilities are obtained using forward pass of DNN.

The various systems considered in this chapter are as follows:

- $\mathbf{MAP^{GMM}}$: the speaker models are obtained from GMM-UBM by MAP adaptation.
- $\mathbf{MAP^{HMM}}$: the speaker-models are obtained from HMM/GMM model as described in Section 4.2.
- $\mathbf{Ivec_{PLDA}}$: the conventional i-vector system for speaker recognition obtained using GMM or DNN SS, which are referred to as $\mathbf{Ivec_{PLDA}^{GMM}}$ or $\mathbf{Ivec_{PLDA}^{DNN}}$ respectively. The system with adapted-DNN SS is labelled as $\mathbf{Ivec_{PLDA}^{DNN\text{-}adp}}$.
- **JFA**: this system represents Joint Factor Analysis model. The JFA using GMM SS is referred to as $\mathbf{JFA^{GMM}}$ while the system using DNN and adapted-DNN SS are referred to as $\mathbf{JFA^{DNN}}$ and $\mathbf{JFA^{DNN\text{-}adp}}$ respectively.

Table 4.3: *Performance of the various GMM based baseline systems on RSR dataset in terms of EER (%). The* ***MAP***$^{\mathbf{GMM}}$ *outperforms other baseline systems in Cond-all.*

No.	Systems/Conditions	Cond1	Cond2	Cond3	Cond-all
1	**MAP**$^{\mathbf{GMM}}$	0.83	**2.15**	**0.21**	**0.69**
2	**MAP**$^{\mathbf{HMM}}$	**0.71**	4.42	0.42	1.32
3	**Ivec**$^{\mathbf{GMM}}_{\mathbf{PLDA}}$	1.24	2.82	0.32	0.91
4	**JFA**$^{\mathbf{GMM}}$	1.42	2.34	0.41	0.71

- **DTW**: raw speech features (MFCCs) and posteriograms obtained from the GMM or DNN are compared using the DTW algorithm in this system. The systems with MFCCs, GMM posteriors, DNN and adapted-DNN posteriors are referred to as **DTW-MFCC**, **DTW-post**$^{\mathbf{GMM}}$, **DTW-post**$^{\mathbf{DNN}}$ and **DTW-post**$^{\mathbf{DNN\text{-}adp}}$ respectively.
- **DTW-onIvec**: this system uses i-vector (estimated over short segments) as input to DTW algorithm. The i-vectors are computed using SS either from GMM or DNN, which are referred to as **DTW-onIvec**$^{\mathbf{GMM}}$ and **DTW-onIvec** $^{\mathbf{DNN}}$ respectively.
- **DTW-onIvec**$_{\mathbf{PLDA}}$: this system uses PLDA projection (as explained in Section 4.3.3) as input to the DTW algorithm. PLDA is trained either with speaker-phone or speaker-phrase as class definition. DTW system with PLDA (trained with speaker-phone labels) projection obtained using GMM posteriors (for online i-vector extraction) is referred to as **DTW-onIvec**$^{\mathbf{GMM}}_{\mathbf{PLDA,\ phn}}$ while with DNN is referred to as **DTW-onIvec**$^{\mathbf{DNN}}_{\mathbf{PLDA,\ phn}}$. The systems, with PLDA trained using speaker-phrase classes are referred to as **DTW-onIvec**$^{\mathbf{GMM}}_{\mathbf{PLDA,\ phr}}$ and **DTW-onIvec**$^{\mathbf{DNN}}_{\mathbf{PLDA,\ phr}}$.

4.5.1 Experiments on the RSR data (female)

The experiments are conducted with the training and evaluation data as detailed in Section 2.10.2. We first describe the model-based SV systems using GMM and DNN posteriors and then describe DTW systems.

Model-based SV systems with GMM posteriors

Table 4.3 compares the performance of various model-based SV systems exploiting GMM posteriors. It is to be noted that the results presented here are comparable or better than those published in Larcher et al. (2014b); Kenny et al. (2014b). The simple MAP technique, **MAP**$^{\mathbf{GMM}}$ (row 1) achieves the best results among the model-based SV systems, which is consistent with the results published in the literature. T-norm is applied on **MAP**$^{\mathbf{GMM}}$ scores with improvement of 24% relative EER (from 2.85% to 2.15% absolute) for condition 2. The **MAP**$^{\mathbf{HMM}}$ performs worse than the **MAP**$^{\mathbf{GMM}}$ in Cond-all, however in Cond1, the former system performs better than the latter system due to the ability of the HMM to capture sequential information.

Table 4.4: *Performance of the various DNN-based SV systems on RSR2015 dataset in terms of EER(%). The JFA system is the best performing system.*

No.	Systems/Conditions	Cond1	Cond2	Cond3	Cond-all
1	$\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{DNN}}$	0.71	2.52	0.21	0.73
2	$\mathbf{JFA}^{\mathbf{DNN}}$	**0.12**	**0.84**	**0.02**	**0.21**

In text-independent SV scenario, the $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{GMM}}$ system outperforms $\mathbf{MAP}^{\mathbf{GMM}}$ as evident by the success of the technique in past SV evaluations. However, in text-dependent scenario, the $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{GMM}}$ system performs worse, which may be due to the duration of the test utterances.

We explored JFA as well, as it has shown to be a dominating modeling technique for text-dependent SV scenario. The latent factor ($\mathbf{z}$) of the JFA model (Equation 2.27), which characterizes the speaker-phrase, is used to compute the cosine distance between the enrollment and test utterances. T-norm is applied to the scores produced by the JFA model. This system ($\mathbf{JFA}^{\mathbf{GMM}}$) performs better than the $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{GMM}}$ in condition 2, thus showing that the matrix $\mathbf{D}$ is able to model the speaker-phrase characteristics better than the matrix $\mathbf{\Pi}$ of the PLDA model as given by Equation 2.20. The JFA can be built with only the development data of RSR2015 dataset without the need of any Fisher database.

Model-based SV systems with DNN posteriors

As explained in Section 4.2, the $\mathbf{Ivec}_{\mathbf{PLDA}}$ and **JFA** systems benefit by incorporating linguistic information from HMM/DNN. The DNN acoustic model is employed to estimate the senone posteriors, which is then subsequently fed to i-vector extraction process. The 10 top scoring DNN posteriors are used to estimate the parameters of the i-vector and JFA models. The back-end classifier of the i-vector model (PLDA) is trained with multiple instances of speaker-phrase classes (from development data).

Table 4.4 shows the performance of the model-based SV systems with DNN posteriors. We observe that integrating DNN posteriors in the $\mathbf{Ivec}_{\mathbf{PLDA}}$ and **JFA** systems consistently improves the performance. In particular, $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{DNN}}$ improves upon $\mathbf{Ivec}_{\mathbf{PLDA}}^{\mathbf{GMM}}$ by 22% relative EER (from 0.91% to 0.73% absolute) for Cond-all condition. The $\mathbf{JFA}^{\mathbf{DNN}}$ achieves good results and clearly outperforms the $\mathbf{JFA}^{\mathbf{GMM}}$, this system performs better than the $\mathbf{MAP}^{\mathbf{GMM}}$ across all conditions by 66% relative EER (from 0.69% vs 0.21% absolute) for Cond-all. This validates the hypothesis that linguistic units of the speech signal are important for the i-vector and JFA based SV approaches.

Table 4.5: *Performance of the various DTW systems on RSR dataset in terms of EER(%). The DTW system using DNN posterior features performs better in content-mismatch conditions.*

No.	Systems/Conditions	Cond1	Cond2	Cond3	Cond-all
1	**DTW-MFCC**	0.38	4.52	0.11	1.23
2	**DTW-post$^{\text{GMM}}$**	0.13	**4.51**	0.11	1.22
3	**DTW-post$^{\text{DNN}}$**	**0.04**	4.61	**0.02**	**1.05**

Table 4.6: *Performance of the various adapted-DNN based systems on RSR dataset in terms of EER (%). The JFA is the best performing system.*

No.	Systems/Conditions	Cond1	Cond2	Cond3	Cond-all
1	**Ivec$^{\text{DNN-adp}}_{\text{PLDA}}$**	0.15	2.17	0.02	0.52
2	**JFA$^{\text{DNN-adp}}$**	0.11	**0.71**	0.02	**0.21**
3	**DTW-post$^{\text{DNN-adp}}$**	**0.02**	14.52	**0.01**	2.61

DTW based SV

The **DTW-MFCC** technique has been explored for text-dependent SV task in the past. It assumes that MFCCs contain speaker and content discriminating information, to be exploited by DTW algorithm. Furthermore, we experimented with GMM and (**DTW-post$^{\text{GMM}}$**), DNN posteriors (**DTW-post$^{\text{DNN}}$**) constituting input to DTW. It can be observed from Table 4.5 that all the DTW techniques achieve better results than the baseline model-based SV systems (**MAP$^{\text{GMM}}$**, **Ivec$^{\text{GMM}}_{\text{PLDA}}$** and **JFA$^{\text{GMM}}$** of Table 4.3) for content-mismatch conditions. However, for condition 2, the performance is significantly worse than the model-based SV systems with GMM posteriors (Table 4.3). It can be observed from Table 4.5 that **DTW-post$^{\text{DNN}}$** (row 3) outperforms the **MAP$^{\text{GMM}}$** for conditions 1 and 3 by 95% relative EER (from 0.83% vs 0.04% absolute) and 90% relative EER (from 0.21% vs 0.02% absolute) respectively.

SV using Adapted-DNN

Table 4.6 shows the performance of various systems (i-vector, JFA and DTW) exploiting posteriors obtained at the output of adapted-DNN. The main motivation of adaptation is to obtain better alignment of the evaluation data. The **Ivec$^{\text{DNN-adp}}_{\text{PLDA}}$**performs better than **Ivec$^{\text{DNN}}_{\text{PLDA}}$** across all conditions. This system performs better than the **MAP$^{\text{GMM}}$** by 26% relative EER (from 0.69% to 0.52% absolute) for Cond-all.

The senone posteriors of the adapted-DNN are used to estimate the parameters of the JFA model as given by Equation 2.27 (matrices **D** and **U**) and subsequently the latent variable **z** (during enrollment and testing phase). From Table 4.6 we observe that **JFA$^{\text{DNN-adp}}$** further improves upon **JFA$^{\text{DNN}}$**, particularly for Cond2, indicating that the DNN adaptation is useful in the i-vector and JFA.

Table 4.7: *Performance of the various DTW systems using online i-vector features on RSR database in terms of EER(%). The* $\textbf{DTW-onIvec}^{\textbf{DNN}}_{\textbf{PLDA, phn}}$ *is the best performing system.*

No.	Systems/Conditions	Cond1	Cond2	Cond3	Cond-all
1	$\textbf{DTW-onIvec}^{\textbf{GMM}}$	0.21	1.52	0.05	0.45
2	$\textbf{DTW-onIvec}^{\textbf{DNN}}$	0.03	0.75	0.02	0.23
3	$\textbf{onIvec}^{\textbf{GMM}}_{\textbf{PLDA}}$	4.41	6.49	1.03	1.93
4	$\textbf{onIvec}^{\textbf{DNN}}_{\textbf{PLDA}}$	1.62	4.42	0.39	1.06
5	$\textbf{DTW-onIvec}^{\textbf{GMM}}_{\textbf{PLDA, phn}}$	0.15	1.21	0.02	0.35
6	$\textbf{DTW-onIvec}^{\textbf{DNN}}_{\textbf{PLDA, phn}}$	**0.02**	**0.65**	**0.01**	**0.18**
7	$\textbf{DTW-onIvec}^{\textbf{DNN}}_{\textbf{PLDA, phr}}$	0.05	0.86	0.03	0.24

The senone posteriors from the adapted-DNN are used as features for the DTW algorithm. We observe that $\textbf{DTW-post}^{\textbf{DNN-adp}}$ performs better than $\textbf{Ivec}^{\textbf{DNN-adp}}_{\textbf{PLDA}}$ and $\textbf{JFA}^{\textbf{DNN-adp}}$ for content-mismatch conditions while significantly degrading performance for condition 2. This degradation in performance is due to the content-discriminating features. We attempt to solve this problem by extracting speaker-discriminating features for DTW algorithm.

DTW based SV with online i-vectors

The **DTW-onIvec** extracts i-vectors on short segments (online i-vectors), which are then used as input features to DTW algorithm. It can be observed from Table 4.7 that the $\textbf{DTW-onIvec}^{\textbf{GMM}}$ and $\textbf{DTW-onIvec}^{\textbf{DNN}}$ outperform the baseline $\textbf{MAP}^{\textbf{GMM}}$ by about 35% relative EER (from 0.69% to 0.45% absolute) and 67% relative EER (from 0.69% to 0.23% absolute) for Cond-all condition. This indicates that online i-vectors represent speakers sufficiently well. The DTW algorithm plays an important role in achieving good performance by the **DTW-onIvec** system. Therefore, without the sequence matching capability (of the DTW algorithm), the online i-vector system performing an averaging operation instead of preserving the sequential information is expected to provide worse results than **DTW-onIvec**. To test this hypothesis, we conducted an experiment by building a system (similar to $\textbf{Ivec}_{\textbf{PLDA}}$) as follows. A sequence of online i-vectors is extracted which is then averaged to obtain a representative i-vector of the utterance. The PLDA is trained using these averaged online i-vectors as features assuming speaker-phrase as classes. The distance between the enrollment and test speech signal is computed using the PLDA model with the averaged online i-vectors. We built two systems applying this strategy, one with GMM posteriors and another with DNN posteriors, which are referred to as $\textbf{onIvec}^{\textbf{GMM}}_{\textbf{PLDA}}$ and $\textbf{onIvec}^{\textbf{DNN}}_{\textbf{PLDA}}$ respectively in Table 4.7. We observe that $\textbf{onIvec}^{\textbf{GMM}}_{\textbf{PLDA}}$ and $\textbf{onIvec}^{\textbf{DNN}}_{\textbf{PLDA}}$ perform worse than **DTW-onIvec**. This result highlights the significance of DTW algorithm, in addition to the online i-vectors, in obtaining low error rates.

From Table 4.7, it can be observed that applying PLDA on top of the online i-vector fea-

Table 4.8: *Performance of the various systems on RSR2015 database in terms of EER(%)/minDCF(× 100) in Cond-all condition.*

No.	Systems/Conditions	Posteriors	Cond-all
1	**MAP**$^{\mathbf{GMM}}$ (Table 4.3)	GMM	0.69/0.329
2	**JFA**$^{\mathbf{DNN\text{-}adp}}$ (Table 4.6)	DNN	0.21/0.129
3	**Ivec**$^{\mathbf{DNN\text{-}adp}}_{\mathbf{PLDA}}$(Table 4.6)	DNN	0.51/0.339
4	**DTW-onIvec**$^{\mathbf{DNN}}_{\mathbf{PLDA,\ phn}}$ (Table 4.7)	DNN	**0.18/0.094**

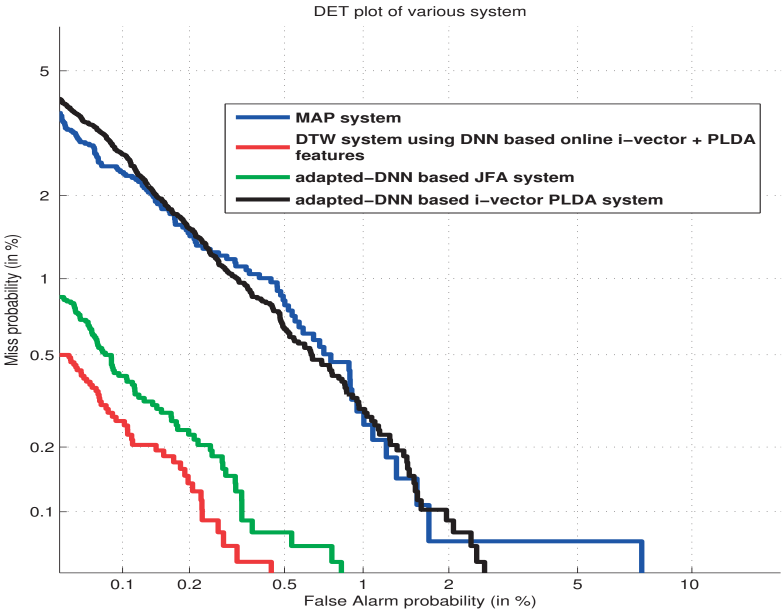

Figure 4.4: *DET curve of the systems presented in Table 4.8 on RSR2015 database.*

tures further improves the performance. The **DTW-onIvec**$^{\mathbf{DNN}}_{\mathbf{PLDA,\ phn}}$ improves over the **MAP**$^{\mathbf{GMM}}$ baseline system by 74% relative EER for Cond-all. In Section 4.3.3, we discussed the two possible methods of defining classes in the PLDA model with online i-vector features, which are speaker-phrase and speaker-phone. We observe that both the systems, **DTW-onIvec**$^{\mathbf{DNN}}_{\mathbf{PLDA,\ phn}}$ and **DTW-onIvec**$^{\mathbf{DNN}}_{\mathbf{PLDA,\ phr}}$, perform similar for all conditions. We did not obtain better results of **DTW-onIvec** using adapted-DNN than DNN and thus we are not presenting the results.

Summary of experiments on RSR2015 database

The minDCF and DET plot of some of the best performing systems are presented in Table 4.8 and Figure 4.4 respectively for Cond-all condition only. These systems include, (i) the **MAP**$^{\mathbf{GMM}}$ baseline, (ii) **Ivec**$^{\mathbf{DNN\text{-}adp}}_{\mathbf{PLDA}}$, (iii) **JFA**$^{\mathbf{DNN\text{-}adp}}$ and, (iv) **DTW-onIvec**$^{\mathbf{DNN}}_{\mathbf{PLDA,\ phn}}$. It is to be noted that **DTW-onIvec**$^{\mathbf{DNN}}_{\mathbf{PLDA,\ phn}}$ improves by 71% relative minDCF (from 0.329% to 0.094% absolute) compared to the baseline **MAP**$^{\mathbf{GMM}}$.

4.5.2 Experiments on the RedDots database (male)

Table 4.9 compares the performance of all systems on RedDots dataset across all the conditions. We consider the MAP system (**MAP**$^{\text{GMM}}$) using GMM posterior as the baseline since it has shown to provide good performance in Zeinali et al. (2016). The model-based SV systems perform worse on the RedDots database compared to RSR2015 database (Dey et al., 2016a). As it has been observed from the experiments on RSR2015 database, the model-based SV approaches with DNN acoustic model outperform those employing GMM. Thus, only the results of DNN based i-vector and JFA systems are reported on the RedDots database.

From Table 4.9, it can be observed that **MAP**$^{\text{GMM}}$ provides EER of 1.23% for Cond-all. The performance of the MAP system is worse on the RedDots than on the RSR2015 database across all conditions, possibly due to long-term intra-speaker variability. The **MAP**$^{\text{HMM}}$ outperforms **MAP**$^{\text{GMM}}$ on this part of the database by 26% relative EER (from 1.23% to 0.94% absolute) on Cond-all.

The **Ivec**$^{\text{DNN}}_{\text{PLDA}}$ and **JFA**$^{\text{DNN}}$ systems do not achieve good results as compared to **MAP**$^{\text{GMM}}$. The poor performance of i-vector and JFA systems can be possibly attributed to the fact that factoring out the content-variability with speaker-phrase data from RSR2015 is not a good choice.

The **DTW-post**$^{\text{DNN}}$ (row 5 of Table 4.9) performs better than model-based SV systems in content-mismatch trials (conditions 1 and 3) as it explicitly matches the content. In speaker-mismatch trials, even the **DTW-post**$^{\text{GMM}}$ (row 6) performs better than **DTW-post**$^{\text{DNN}}$.

The **DTW-onIvec**$^{\text{DNN}}$ performs better than **MAP**$^{\text{GMM}}$ by 55% relative EER (from 1.23% to 0.55% absolute) for Cond-all. Thus, on this database as well, the online i-vector representation with DTW algorithm achieves better results than **Ivec**$^{\text{DNN}}_{\text{PLDA}}$, **JFA**$^{\text{DNN}}$ and **MAP**$^{\text{GMM}}$. We experimented with using PLDA on top of online i-vectors. We observe that **DTW-onIvec**$^{\text{DNN}}_{\text{PLDA, phn}}$ further improves upon **DTW-onIvec**$^{\text{DNN}}$ with improvement of 3% relative EER (from 2.69% to 2.61% absolute) for Cond2. However, it can also be observed from Table 4.9 that training the PLDA with speaker-phrase labels degrades the performance. An explanation of the performance degradation is possibly due to training PLDA with speaker-phrase classes from RSR dataset (which do not match the evaluation phrases of RedDots).

4.6 Conclusions

In this chapter, we presented model- (MAP, i-vector and JFA) and DTW-based techniques for performing text-dependent SV with fixed phrases. We validated the techniques on two databases, female part of RSR and male part of RedDots. We experimented with model-based SV systems using GMM and DNN posteriors. From results, we observed that MAP technique performs the best among the model-based SV approaches exploiting GMM posteriors. Integrating DNN posteriors in the i-vector and JFA systems achieves good results across all the conditions, with JFA improves upon the MAP technique by 66% relative EER for Cond-all

Table 4.9: *Performance of all the systems on RedDots (Part4) database in terms of EER(%). The Cond-all refers to the system performance across all the 3 conditions.*

No.	Systems/Conditions	Cond1	Cond2	Cond3	Cond-all
1	$\mathbf{MAP}^{\mathbf{GMM}}$	5.62	4.04	0.90	1.23
2	$\mathbf{MAP}^{\mathbf{HMM}}$	2.63	3.72	0.73	0.94
3	$\mathbf{Ivec}^{\mathbf{DNN}}_{\mathbf{PLDA}}$	6.10	3.03	0.97	1.29
4	$\mathbf{JFA}^{\mathbf{DNN}}$	7.21	4.43	1.34	1.85
5	$\mathbf{DTW\text{-}post}^{\mathbf{DNN}}$	**0.62**	7.62	0.54	1.13
6	$\mathbf{DTW\text{-}post}^{\mathbf{GMM}}$	0.89	4.92	0.76	0.96
7	$\mathbf{DTW\text{-}onIvec}^{\mathbf{DNN}}$	0.99	2.69	0.44	**0.55**
8	$\mathbf{DTW\text{-}onIvec}^{\mathbf{DNN}}_{\mathbf{PLDA,\ phn}}$	0.81	**2.61**	**0.38**	**0.55**
9	$\mathbf{DTW\text{-}onIvec}^{\mathbf{DNN}}_{\mathbf{PLDA,\ phr}}$	1.24	2.85	0.51	0.62

in RSR dataset. This gain in performance is consistent with the results published for text-dependent and text-independent SV scenarios. Additional gain in performance is obtained with adapted-DNN, more particularly by the JFA technique. It clearly shows that obtaining better alignment for the evaluation data results in better performance.

The DTW algorithm offers an easy method to match the sequential patterns of the train and test templates. Being a non-parametric method, it does not require any training data for the development. We experimented with different input features for the DTW algorithm, namely MFCCs, GMM and DNN posteriors. In content-mismatch conditions, the DTW systems provide better results than the model-based SV systems. In particular, the DTW algorithm using DNN posteriors outperforms the MAP system in condition 1 by 95% relative EER in RSR dataset.

However, DTW system using DNN posteriors performs worse than MAP technique in speaker-mismatch condition. This degradation in performance is due to content-discriminating features. In this chapter, we address this problem by extracting speaker specific information by employing i-vector system. We extract online i-vectors (for short segments) using the i-vector extractor of the speech utterance resulting in sequences of online i-vectors extracted from enrollment and test utterances. The DTW algorithm is then used to match the train and test templates of online i-vectors. We found that this approach outperforms the MAP based system by 67% relative EER for Over-all condition in RSR database.

The PLDA is usually applied in state-of-the-art SV systems as a channel compensation model. In this chapter, we experimented with two different definition of class labels, namely, (i) speaker-phrase, and (ii) speaker-phone for training the PLDA. Although on RSR database, we obtained similar performance with both the strategies for defining classes, but on RedDots we obtained considerable performance benefit with speaker-phone labels.

5 Content-matching for random-digit based text-dependent speaker verification

Contents

In this chapter we present approaches to exploit phonetic information for addressing seen and random-digit strings tasks. This chapter is based on this publication:

> Subhadeep Dey, Petr Motlicek, Srikanth Madikeri, and Marc Ferras. Exploiting sequence information for text-dependent speaker verification. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5370–5374. IEEE, 2017a

In the last chapter, text-dependent SV using **fixed-phrases** has been explored where the user is constrained to utter a specific phrase. However in many practical applications, we would like to impose lesser constraint on the lexical content of the speaker. To this end, we are interested in employing random sequence of words or digits for implementing text-dependent SV (Larcher et al., 2014b). We refer to this task as content mismatch text-dependent SV. In this scenario, the spoken content of the enrollment utterance is not identical to the test data.

In this chapter, we are interested in two text-dependent scenarios, namely, **random-digit strings** and **seen** tasks as described in Section 2 (Larcher et al., 2014b; Scheffer and Lei, 2014). For **random-digit strings**, the enrollment data consist of a user uttering prompted digits randomly while in the test phase, the speaker pronounces a prompted random combination of a few unique digits. This leads to the creation of different co-articulation effects between the enrollment and test data. For the **seen** task, all the phrases spoken by the speaker are collected to obtain enrollment data while the test data consists of the speaker uttering one of the enrollment-phrases. In this chapter, we are interested in speaker mismatch condition only in these scenarios as it evaluates the system for SV. The content mismatch conditions in these scenarios have to be handled by an ASR. Evaluation of the baseline SV system on these tasks reveals severe degradation of performance as compared to the **fixed-phrase** case. However, an advantage of these scenarios is that they are more robust to replay attack (Stafylakis et al., 2016, 2015) than fixed-phrase.

The standard techniques, such as i-vector, JFA, have shown to provide reasonable SV performances for **random-digit strings** and **seen** tasks (Stafylakis et al., 2016, 2015; Scheffer and Lei, 2014). In literature, approaches that aim to match the lexical content (or phonetic units) of the enrollment and test data have shown to provide good results in these tasks (Chen et al., 2015b; Wang et al., 2016). Motivated by these results, we explore techniques to exploit the common phonetic units between enrollment and test data to provide SV scores in an unsupervised manner (i.e. without using text-transcript).

This chapter is organized as follows. Section 5.1 describes the baseline SV approaches considered in this chapter, while in Section 5.2, the proposed technique is presented. The experimental setup and results are described in Sections 5.3 and 5.4, and the chapter is concluded in Section 5.5.

5.1 Baseline Systems

The DNN based i-vector system (as described in the last chapter) is considered as one of the baseline systems. In Scheffer and Lei (2014), a posterior normalization technique (on top of DNN based i-vector approach) is proposed to scale the sufficient statistics (SS) of the enrollment data to match those of the test data. The posterior normalization technique is shown in Figure 5.1 and it aims to normalize the count of the senone units (of the enrollment data) before computing i-vectors. The technique is described as follows. Let N_e and N_t be the zero-th order statistics (as defined by Equation 2.11) of the enrollment and test utterances

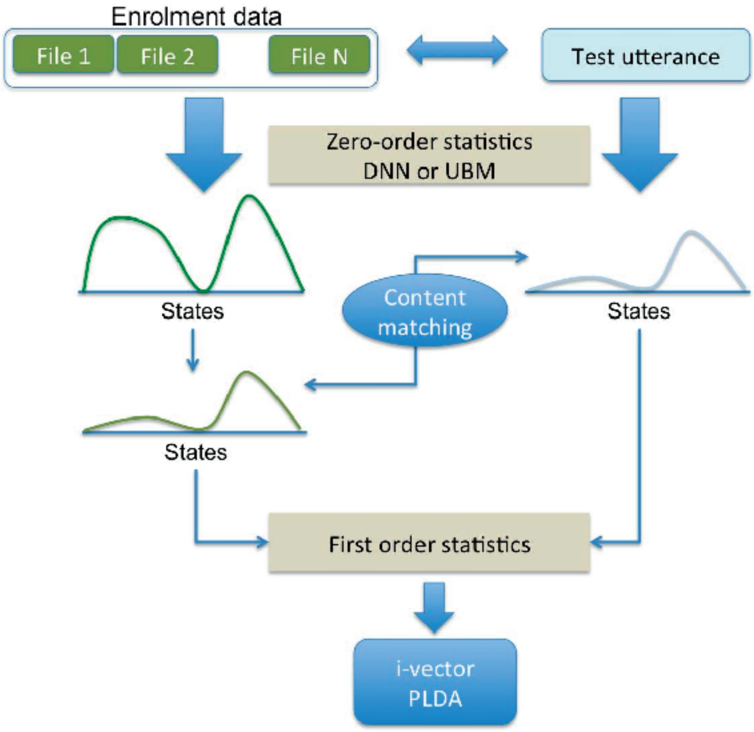

Figure 5.1: *Posterior normalization technique for text-dependent SV. The image has been taken from Scheffer and Lei (2014)*

Table 5.1: *Interpreting sufficient statistics for the posterior normalization approach. The N_e and N_t refer to the zero-th order statistics of the enrollment and test data.*

Conditions	Posterior normalization	Interpretation
$N_e \geq N_t$	From Eqn. 5.1, $\beta \leq 1$	Data selection
$N_e \leq N_t$	From Eqn. 5.1, $\beta \geq 1$	Reusing speech frames
$N_e \geq 0$, $N_t = 0$	From Eqn. 5.1, $\beta = 0$	Discard senone units
$N_e = 0$, $N_t \geq 0$	From Eqn. 5.1, $\beta = 0$	Data synthesis

respectively, and $\mathbf{F}_e$ and $\mathbf{F}_t$ be the first order statistics (as defined by Equation 2.12) of the enrollment and test utterances respectively. The new statistics for the enrollment are obtained as

$$\mathrm{N}_e^{'} = \beta \mathrm{N}_e$$

$$\mathbf{F}_e^{'} = \beta \mathbf{F}_e,$$

$$\beta = \frac{\mathrm{N}_t}{\mathrm{N}_e}, \tag{5.1}$$

where β is a normalization constant. When N_e or N_t is 0, β is set to zero as well. The details of the technique can be found in Scheffer and Lei (2014). The different scenarios for the normalization factor (β) are illustrated in Table 5.1. In addition to this posterior normalization technique, we consider GMM-UBM as the baseline system.

5.2 Posteriors and Content Matching

The techniques developed in the previous chapter (for **fixed-phrase**) cannot be applied for **seen** and **random-digit strings** tasks since the lexical content of the enrollment is not identical to that of the test data. We developed techniques that address the mismatch in the spoken content for both the tasks, by (a) one based on DNN posterior estimation, and (b) using online i-vectors. Both are described in the following section.

5.2.1 Senone posteriors from ASR decoder

The DNN based i-vector system involves computation of SS from DNN outputs. We propose to apply senone posteriors obtained from word-recognition lattices (from ASR) for the i-vector extraction since accurate estimation of phonetic units (compared to DNN outputs) can help to factor out the content variability (in the i-vector extraction). These lattices are obtained by decoding an utterance using acoustic, language and lexical models (of ASR) (Povey et al., 2011b). Furthermore, we use posterior normalization technique as proposed for the baseline system (Scheffer and Lei, 2014) on these senone posteriors.

5.2.2 Online i-vectors

In the past, strategies to exploit phonetic information have been successful for **seen** and **random-digit strings** (Wang et al., 2016). In Chen et al. (2015b), i-vectors are extracted for each of the senone units, which are then clustered to obtain speaker representation. In Scheffer and Lei (2014), they analyze the performance of i-vector system for **seen** task. Experiments using state-of-the-art techniques show that content mismatch has a strong impact on the SV performance (Scheffer and Lei, 2014) and normalizing posteriors reduces the error rate considerably. Past research shows that matching common linguistic units between enrollment and test data produces low error rate (Stolcke et al., 2007; Baker et al., 2005) for text-independent SV. We refer to the process of transforming the enrollment utterance to match the lexical content of test data as content matching. We present an approach to perform content matching by selecting regions explicitly in the enrollment data to match the test data.

In the last chapter, we used online i-vectors as features to DTW algorithm for fixed phrase based text-dependent SV. The achieved results indicate that online i-vectors contain speaker and content information. We use online i-vectors as features for performing content matching as well.

The strategy to perform content matching is as follows. Online i-vectors are estimated for each speech frame with a context of 10 frames (i.e. sufficient statistics are estimated with a window size of 21 frames). This leads to a sequence of online i-vectors corresponding to an utterance. Enrollment and test content are matched by computing the maximum similarity scores from each online i-vector in test to all instances in enrollment. As many scores as the number of speech frames in test utterance are obtained. Finally, these scores are averaged to obtain a global similarity score. The rationale behind this approach is to choose the closest frame in the enrollment data. The accumulated global score is obtained as follows

$$s(\mathbf{W}^e, \mathbf{W}^t) = \frac{1}{C} \sum_j min\{d(\mathbf{w}_i^e, \mathbf{w}_j^t), \forall i = \{1, 2, \cdots, R\}\}, \tag{5.2}$$

where $\mathbf{W}^e = \{\mathbf{w}_1^e, \mathbf{w}_2^e, \cdots, \mathbf{w}_R^e\}$ and $\mathbf{W}^t = \{\mathbf{w}_1^t, \mathbf{w}_2^t, \cdots, \mathbf{w}_C^t\}$ represent set of i-vectors for the enrollment and test data, the function $\mathrm{d}(\mathbf{w}_i^e, \mathbf{w}_j^t)$ computes the distance between the i-vectors $\mathbf{w}_i^e$ and $\mathbf{w}_j^t$. The score $s(\mathbf{W}^e, \mathbf{W}^t)$ represents the accumulated distance between the closest speech frames. We used cosine distance metric to compute the dissimilarity between two online i-vectors. A threshold on the cosine distance can be applied to detect if a test frame is not present in the enrollment data.

The content matching technique described above does not assume phonetic label of the speech frame. In a scenario, when phonetic alignments are obtained using the text-transcripts, the minimization of Equation 5.2 could be performed by iterating over the same phonetic category of the enrollment data.

5.2.3 PLDA as a feature extractor

The online i-vector representation contains other information in addition to the speaker content. In order to factor out the channel effects, a PLDA model is trained as the back-end classifier with online i-vectors as features. In the last chapter, PLDA trained with speaker-phone pairs is used for fixed phrase based text-dependent SV task. In this chapter, we explore speaker-word combination as classes definition for the training the PLDA. A speech recognizer is employed to align the development data with the word labels. Online i-vectors corresponding to within word boundaries are subsequently used as features for the PLDA model. The PLDA model is then used to project the online i-vectors using the parameters of the model to obtain channel compensated vectors as done in Section 4.3.3 (plda-vectors). The content matching algorithm can be applied on plda-vectors as well.

5.3 Experimental Setup

We used the same MFCC features as used in Chapter 2 (Section 2.11.1). The dimensionality of i-vector (also online i-vector) extractor is set to 400. For evaluation data, the Part 1 and 3 are used (as described in Section 2.10.2) for the **seen** and **random-digit strings**. The Fisher data is used as the training data (as described in Section 2.10.2). The performance of DNN based

Table 5.2: *Performance of the different baseline systems in terms of EER (%). The* $\textbf{MAP}^{\textbf{GMM}}$ *provides the best performance among the baseline systems in both tasks.*

Systems/Tasks	**seen**	**random-digit strings**
$\textbf{Ivec}_{\textbf{PLDA}}^{\textbf{GMM}}$	16.5	17.3
$\textbf{Ivec}_{\textbf{PLDA}}^{\textbf{DNN}}$	11.6	15.2
$\textbf{PN-Ivec}_{\textbf{PLDA}}^{\textbf{GMM}}$	12.3	15.8
$\textbf{PN-Ivec}_{\textbf{PLDA}}^{\textbf{DNN}}$	8.6	14.4
$\textbf{MAP}^{\textbf{GMM}}$	**4.5**	**8.6**

ASR is described in Section 4.4. We used the conventional ASR decoder parameters to obtain word recognition lattices (Povey et al., 2011b) (beam width of 13). The same type of lattices has been used previously for various tasks (Motlicek et al., 2012, 2013; Imseng et al., 2013). From these lattices, we obtain the senone posteriors. We observed that by fixing the acoustic scale parameter to 0.01, i-vectors are obtained that follow a Gaussian distribution. Furthermore, we observed that higher acoustic scale (> 0.01) leads to i-vectors with high kurtosis and thus making the PLDA model ineffective.

5.4 Experimental Results and Discussions

In this section, we describe the results obtained with the baseline and the proposed SV approaches. The various systems considered in this chapter are the following:

- $\textbf{PN-Ivec}_{\textbf{PLDA}}$: it uses posterior normalization technique as explained in Section 5.1. The SV approaches using GMM, DNN and decoded ASR lattice posteriors for i-vector extraction are referred to as $\textbf{PN-Ivec}_{\textbf{PLDA}}^{\textbf{GMM}}$, $\textbf{PN-Ivec}_{\textbf{PLDA}}^{\textbf{DNN}}$ and $\textbf{PN-Ivec}_{\textbf{PLDA}}^{\textbf{DNN-dec}}$ respectively.

- **CN-onIvec**: the SV techniques applying content matching technique using online i-vectors as explained in Section 5.2.2. The systems using GMM, DNN and decoded ASR lattice posteriors for online i-vector extraction are referred to as $\textbf{CN-onIvec}^{\textbf{GMM}}$, $\textbf{CN-onIvec}^{\textbf{DNN}}$ and $\textbf{CN-onIvec}^{\textbf{DNN-dec}}$ respectively.

- $\textbf{CN-onIvec}_{\textbf{PLDA}}^{\textbf{DNN}}$: a PLDA model is trained on top of the online i-vectors as the channel compensation model. We explore the use of speaker-phone and speaker-word pairs to train the PLDA. The SV approaches trained on plda-vectors (estimated using online i-vectors with DNN and decoded ASR posteriors) with speaker-phone pairs are referred to as $\textbf{CN-onIvec}_{\textbf{PLDA,p}}^{\textbf{DNN}}$ and $\textbf{CN-onIvec}_{\textbf{PLDA,p}}^{\textbf{DNN-dec}}$, while the systems trained on plda-vectors with speaker-word labels are referred to as $\textbf{CN-onIvec}_{\textbf{PLDA,w}}^{\textbf{DNN}}$ and $\textbf{CN-onIvec}_{\textbf{PLDA,w}}^{\textbf{DNN-dec}}$.

Table 5.3: *Performance of the different SV approaches (using senone posteriors extracted from decoded ASR lattices) in terms of EER (%). The* ***PN-Ivec***$^{\textbf{\textit{DNN-dec}}}_{\textbf{\textit{PLDA}}}$*performs the best among the other techniques for* ***seen*** *task.*

Systems/Tasks	**seen**	**random-digit strings**
Ivec$^{\textbf{DNN-dec}}_{\textbf{PLDA}}$	10.9	18.9
PN-Ivec$^{\textbf{DNN-dec}}_{\textbf{PLDA}}$	**5.6**	15.7

5.4.1 Baseline SV

Table 5.2 shows the performance of various i-vector and **MAP**$^{\textbf{GMM}}$ based SV for **seen** and **random-digits strings**. We observe that performance of the approaches on **seen** is significantly worse than the **fixed phrase** based text-dependent system (as described in previous chapter). Lower bound for **seen** task is 2.3% EER for the case when the phrases of the enrollment are identical to the test.

The posterior normalization technique is used to exploit the content of the enrollment data. We observe that this approach reduces the error rates by 26% relative EER (from 11.6% to 8.6% absolute) and 5% relative EER (from 15.2% to 14.4% absolute) for the **seen** and **random-digit strings**. Furthermore, we observe that incorporating the phonetic information (with DNN and decoded ASR posteriors) helps the SV. The **MAP**$^{\textbf{GMM}}$ provides the best performance among the baseline techniques considered in this chapter. The EER for this system is comparable to the results published in literature Stafylakis et al. (2015); Chen et al. (2015b). We applied T-norm on the scores produced by the **MAP**$^{\textbf{GMM}}$. T-norm improves **MAP**$^{\textbf{GMM}}$ by 2% absolute EER for the **random-digit strings**.

5.4.2 SV using ASR lattice posteriors

We explore the application of senone posteriors estimated from word recognition ASR lattices in an i-vector framework. Table 5.3 shows the performance of i-vector based SV using these posteriors. We observe that **Ivec**$^{\textbf{DNN-dec}}_{\textbf{PLDA}}$ outperforms **Ivec**$^{\textbf{DNN}}_{\textbf{PLDA}}$ for **seen** task by 0.7% absolute EER. Significant gain in performance is achieved by the **PN-Ivec**$^{\textbf{DNN-dec}}_{\textbf{PLDA}}$ compared to **PN-Ivec**$^{\textbf{DNN}}_{\textbf{PLDA}}$, with 35% relative EER (from 8.6% to 5.6% absolute) for **seen**. This indicates the importance of more accurate senone alignments in obtaining better SV performance for this task. However, performances of **Ivec**$^{\textbf{DNN-dec}}_{\textbf{PLDA}}$ and **PN-Ivec**$^{\textbf{DNN-dec}}_{\textbf{PLDA}}$ degrade for the **random-digit strings** compared to the **Ivec**$^{\textbf{DNN}}_{\textbf{PLDA}}$. One of the reasons could be that the performance of the ASR (unconstrained LM) is poor on the RSR2015 dataset (~ 80% WER).

5.4.3 SV using content matching

As opposed to using posterior normalization, we also explore content matching using online i-vectors, as described in Section 5.2.2. Table 5.4 shows the performance of the proposed SV

Table 5.4: *Performance of the different SV systems (using content matching technique) in terms of EER (%). The* ***CN-onIvec***$_{\mathbf{PLDA,w}}^{\mathbf{DNN}}$ *performs the best among the other systems in* ***random-digit strings*** *task. The * indicates the system using text-transcript.*

Systems/Tasks	**seen**	**random-digits strings**
CN-onIvec$^{\mathbf{GMM}}$	4.1	13.4
CN-onIvec$^{\mathbf{DNN}}$	2.8	12.2
CN-onIvec$^{\mathbf{DNN\text{-}dec}}$	4.3	15.5
CN-onIvec$_{\mathbf{PLDA,p}}^{\mathbf{DNN}}$	**2.7**	7.7
CN-onIvec$_{\mathbf{PLDA,w}}^{\mathbf{DNN}}$	**2.7**	**7.5**
CN*-onIvec$_{\mathbf{PLDA,w}}^{\mathbf{DNN}}$	**2.5**	7.6

using content matching. We observe that the proposed approaches outperform the posterior normalization based SV techniques for **seen**. In particular, the **CN-onIvec**$^{\mathbf{DNN}}$ performs better than **PN-Ivec**$_{\mathbf{PLDA}}^{\mathbf{DNN}}$ by relative EER of 67% (from 8.6% to 2.8% absolute) and 15% (from 14.4% to 12.2% absolute) for the **seen** and **random-digit strings** respectively. This indicates the importance of the content matching technique using online i-vectors. We observe that **CN-Ivec**$_{\mathbf{PLDA,p}}^{\mathbf{DNN}}$ performs better than the **MAP**$^{\mathbf{GMM}}$ by relative EER of 10% (8.6% to 7.7% absolute). The **CN-onIvec**$_{\mathbf{PLDA,w}}^{\mathbf{DNN}}$ further improves upon **CN-onIvec**$_{\mathbf{PLDA,p}}^{\mathbf{DNN}}$ by 0.2% absolute EER in **random-digit strings**. Thus, training the PLDA using speaker-word labels is more effective in the **random digits strings**.

We explore the scenario in which text-transcript of the utterance is provided to us (cheating experiment). In this case, the content-matching technique is used by performing the minimization operation of Equation 5.2 over the same phonetic units between enrollment and test data. An ASR is used to align the enrollment and test data with the ground truth. Scores from the closest frames between the enrollment and test data are accumulated by iterating over same phonetic classes. The EER for the **seen** task reduces by 0.2% absolute for the **CN-onIvec**$_{\mathbf{PLDA,w}}^{\mathbf{DNN}}$. However, for the **random-digit strings**, we did not get any improvement compared to **7.5**% EER.

5.4.4 Summary of experiments for seen and random-digit strings

The minDCF and DET plots of two best performing SV approaches are presented in Table 5.5 and Figures 5.2. The systems include, (i) **MAP**$^{\mathbf{GMM}}$, and (ii) **CN-Ivec**$_{\mathbf{PLDA,w}}^{\mathbf{DNN}}$. It can be observed from the Table 5.5 that **CN-onIvec**$_{\mathbf{PLDA,w}}^{\mathbf{DNN}}$ outperforms **MAP**$^{\mathbf{GMM}}$ (Table 5.2) by relative minDCF 48% (from 2.2 to 1.14) for **seen** task.

5.5 Conclusions

In this chapter, we address **seen** and **random-digit strings** based text-dependent SV. The posterior normalization technique shows significant gain in performance as compared to

Table 5.5: *Performance of the best performing SV techniques in terms of EER (%)/minDCF (× 100) for* ***seen*** *and* ***random-digit strings****.*

Systems	**seen**	**random-digit strings**
MAP$^{\mathbf{GMM}}$ (Table 5.2)	4.4/2.2	8.6/4.16
CN-Ivec$^{\mathbf{DNN}}_{\mathbf{PLDA,w}}$ (Table 6.4)	2.7/1.14	7.6/3.71

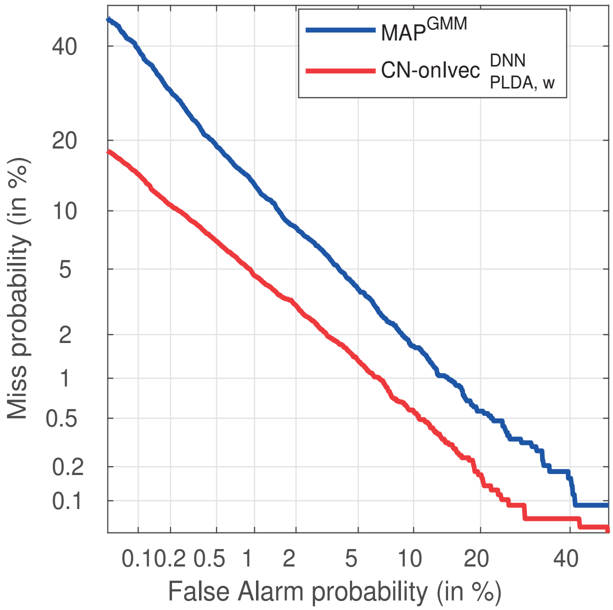

Figure 5.2: *DET curve of the SV approaches presented in Table 5.5 for* ***seen*** *task.*

conventional i-vector technique for **seen**. We proposed to further improve upon the posterior normalization by, (a) enhancing the senone prediction accuracy of the DNN posteriors, and (b) matching the lexical content of the enrollment to that of the test using online i-vectors. We explore the use of speaker-word pair to train the PLDA model on top of online i-vectors. The PLDA is used to obtain channel compensated vectors (plda-vectors). We observe that content matching using plda-vectors achieves the best results for **seen** and **random-digit strings** with 40% and 12% relative EER over **MAP$^{\mathbf{GMM}}$**.

6 DNN based speaker embedding for text-dependent speaker verification

Contents

This chapter presents DNN based speaker embedding exploiting phonetic information for text-dependent speaker verification. This chapter is based on the following publications:

Subhadeep Dey, Takafumi Koshinaka, Petr Motlicek, and Srikanth Madikeri. DNN based speaker embedding using content information for text-dependent speaker

verification. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018a

Subhadeep Dey, Srikanth Madikeri, and Petr Motlicek. End-to-end text-dependent speaker verification using novel distance measures. In *Proceedings of Interspeech*, 2018b

In the last few chapters, i-vector framework has been explored for **fixed-phrase** and **random-digit strings** based text-dependent SV. The i-vector approach assumes the data of the speaker to be generated by a GMM (Garcia-Romero and Espy-Wilson, 2011). In another direction, techniques that employ DNN for speaker discrimination are found to be beneficial for text-dependent SV (Variani et al., 2014; Heigold et al., 2016). In this approach, the activations of the last hidden layer of DNN capture the distinguishing characteristics of the speaker. However, phonetic knowledge of the speech signal is not used. In this chapter, we aim to exploit phonetic information for training speaker discriminative DNN as past research shows that lexical content of an utterance is beneficial for SV (Zeinali et al., 2016; Campbell et al., 2003). In the previous chapters, DNN is employed to predict phonetic labels to be subsequently applied for i-vector extraction, while in this chapter, the DNN is used for speaker classification.

Various approaches to DNN based speaker classification have been proposed in literature. In Variani et al. (2014), a DNN is employed to map feature vectors to speaker targets. The final layer of the DNN applies a soft-max function and the network is optimized using cross entropy as objective function. The outputs of last hidden layer are used to extract speaker representation (also referred to as speaker embedding) during evaluation phase. This approach is referred to as DNN based speaker embedding. A back-end classifier, such as PLDA, is applied on top of speaker embeddings to obtain SV scores.

As an alternative to DNN based speaker embedding approach, several studies have explored end-to-end SV (Heigold et al., 2016; Snyder et al., 2016; Chowdhury et al., 2017). End-to-end techniques involve directly optimizing SV based losses to train a neural network. The loss function is usually based on distance measure between a pair of audio recordings such that recordings from the same speaker will have a low distance-measure (Heigold et al., 2016; Nagrani et al., 2017). The baseline end-to-end approach consists of mapping a variable length speech segment to a fixed dimensional speaker vector by estimating the mean of hidden representations in DNN structure (Nagrani et al., 2017; Bredin, 2017). The distance between two utterances is obtained by computing Euclidean norm between the vectors. This approach performs worse than the conventional GMM-UBM based SV on a publicly available corpora (Bhattacharya et al., 2016; Snyder et al., 2016). We believe that the degraded performance is due to the employed averaging operation, which may not capture the phonetic information of an utterance. Recent studies indicate that techniques exploiting phonetic information in addition to speaker is beneficial for text-dependent SV (Chen et al., 2015b; Zeinali et al., 2016). In this chapter, we propose to incorporate phonetic information in the end-to-end SV by computing distance function with linguistic units co-occurring between enrollment and test data. The whole network is optimized in an end-to-end fashion to estimate SV scores.

The chapter is organized as follows. The DNN based speaker embedding approach is described in Section 6.1. This is then followed by description of end-to-end SV in Section 6.2. The proposed approaches are described in Sections 6.3 and 6.4. Experimental setup and results

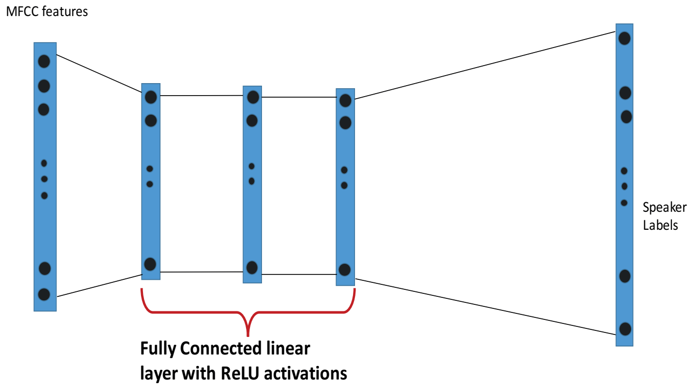

Figure 6.1: *The d-Vector approach for text-dependent SV.*

are described in Sections 6.5 and 6.6. Finally, the chapter is concluded in Section 6.7.

6.1 DNN based speaker embedding

In literature, DNN based speaker embedding approaches have shown to provide promising SV results (Variani et al., 2014; Heigold et al., 2016). In this section, we describe the following DNN based speaker embedding approaches, namely (i) d-Vector, (ii) utterance embedding, and (iii) speaker-phonetic embedding. The first two techniques use speaker labels for training the DNN, while the third approach requires phonetic information as well.

6.1.1 d-Vector

The d-Vector technique is proposed in Variani et al. (2014) for phrase based text-dependent SV. In this approach, a DNN is trained to predict speakers for each input speech frame (with context of frames appended to it). The network architecture, as shown in Figure 6.1, consists of a few fully connected (FC) layers and a final soft-max layer. The hidden layers of the DNN employ rectified linear unit (ReLU) activation function. The whole network is trained to minimize cross entropy objective function. During evaluation for an utterance, the final soft-max layer is discarded and the activations per frame of the last hidden layer are accumulated to obtain a speaker template ($\mathbf{h}^{'}$) as follows:

$$\mathbf{h}^{'} = \frac{1}{T}\sum_{t}\mathbf{h}_t, \tag{6.1}$$

where $\mathbf{h}_t$ is the hidden representation of the DNN for t^{th} frame of speech and T is the total number of speech frames. This representation ($\mathbf{h}^{'}$) is referred to as d-vector. The d-vectors of enrollment and test data are compared to obtain SV scores. It has been shown that training a PLDA (using d-vectors as features) is found to be helpful for SV.

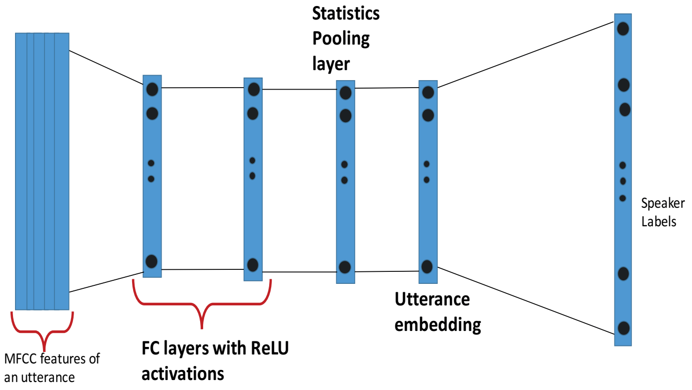

Figure 6.2: *Utterance embedding approach for text-dependent SV.*

6.1.2 Utterance Embedding

For the utterance embedding approach, a DNN is employed to map an utterance to a speaker label (Snyder et al., 2018). The network architecture for performing utterance embedding is shown in Figure 6.2. The network takes context-appended frames as input, which is then forwarded to a few FC layers. The output of the previous step is then passed to a statistics pooling layer. This layer computes the mean and standard deviation on the outputs of previous layer over entire audio recording. A FC layer is applied on the output of statistics pooling to obtain speaker embedding of an utterance (Snyder et al., 2017). A final soft-max layer is applied to compute posterior probability of speakers. During evaluation, the last layer is ignored and the speaker embedding of an utterance is employed. A PLDA is applied on top of the embeddings to provide SV scores.

6.1.3 Speaker-phonetic Embedding

The previous approaches to DNN based speaker embedding require speaker labels for training. However in literature, it has been found that training the DNN with phonetic information, in addition to speaker, is beneficial for text-dependent SV (Chen et al., 2015c). In this approach, the DNN is trained to optimize speaker and phonetic loss (cross entropy objective function). The activations from the last hidden layer of Figure 6.3 are used to represent speaker-phonetic embedding. Figure 6.3 illustrates the process of training a DNN to obtain speaker-phonetic embedding. A back-end classifier, such as PLDA is trained on these embeddings to obtain SV scores.

6.2 End-to-end SV

In the approaches described in the last section, the DNN is trained to classify speakers. During evaluation, the final layer is discarded and additional post-processing steps are required in order to perform SV. Recently Bredin (2017) introduced a end-to-end framework for training a DNN to output SV scores directly without the need of extra steps. The end-to-end network is

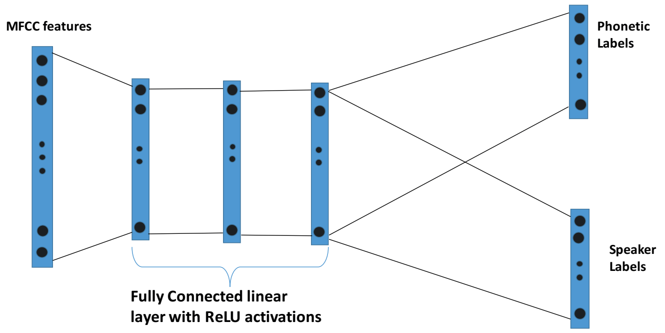

Figure 6.3: *Speaker-phonetic embedding approach for text-dependent SV.*

trained with Euclidean distance based loss function.

6.2.1 Triplet-loss

In this section, we describe a successful end-to-end approach, referred to as triplet-loss, which has shown to provide state-of-the-art results in object recognition applications (Schroff et al., 2015) and is described in the following section. Triplet-loss technique has shown to provide encouraging results in SV as well (Li et al., 2017). The triplet-loss approach comprises presenting three utterances (also referred to as the triplet, τ), as represented by the set $\{\mathbf{X}^a$, $\mathbf{X}^p$, $\mathbf{X}^n\}$, as input for training the network. In literature, these examples are popularly referred to as the anchor, positive and negative instances (Bredin, 2017; Li et al., 2017). These utterances of the triplet are selected in such a way that the anchor and positive utterances belong to the same class while the anchor and negative examples do not share the same speaker identity. Assuming the hidden representation of the utterance ($\mathbf{X}$) is represented by the function $\mathbf{f}(\mathbf{X})$, the triplet loss (E_{trip}) is given by

$$E_{trip}(\tau) = \mathrm{d}(\mathbf{f}(\mathbf{X}^a), \mathbf{f}(\mathbf{X}^p)) - \mathrm{d}(\mathbf{f}(\mathbf{X}^a), \mathbf{f}(\mathbf{X}^n)) + \alpha, \tag{6.2}$$

where d(.) is the function that computes the distance between two vectors, and α is a pre-defined constant (0.1 is used in our experiments). The threshold (α) represents the margin between the positive and negative examples. The most commonly used distance functions are Euclidean and cosine similarity. In this chapter, experiments are performed using the Euclidean distance. The network employing triplet-loss objective function is trained with triplets (τ) for which $E_{trip}(\tau) \geq 0$. In literature (Bredin, 2017; Li et al., 2017), a triplet (τ) can be categorized as:

- **Easy**: the network can classify the triplet correctly, i.e., for which E_{trip} - $\alpha \leq 0$,
- **Hard**: the network can not classify the triplet correctly with a margin α, i.e. $E_{trip} \geq 0$, and

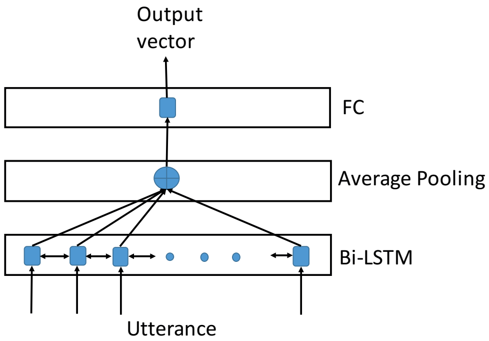

Figure 6.4: *The neural network architecture of triplet-loss approach for text-dependent SV.*

- **Semi-hard**: the network misclassifies the triplet, i.e., E_{trip} - $\alpha \geq 0$

In training a network using triplet-loss objective function, hard or semi-hard triplets are selected for each mini-batch. We apply the same network topology as used in speaker diarization and SV (Bredin, 2017) (as shown in Figure 6.4). The input is fed to a bi-directional Long Short Term Memory (bi-LSTM) or a FC layer with tanh activation function to produce speaker representation of a speech frame (Bredin, 2017; Heigold et al., 2016). This output is fed to Average Pooling layer that computes the mean of the activations to produce a vector. This vector is then forwarded to a FC layer to obtain speaker representation.

Let us assume that $\mathrm{d}_\tau^+ = \mathrm{d}(\mathbf{f}(\mathbf{X}^a), \mathbf{f}(\mathbf{X}^p))$ and $\mathrm{d}_\tau^- = \mathrm{d}(\mathbf{f}(\mathbf{X}^a), \mathbf{f}(\mathbf{X}^n))$ refer to the positive and negative distances respectively, then the triplet-loss for a mini-batch is defined by:

$$E_{minibatch} = \mu^+ - \mu^- + \alpha, \tag{6.3}$$

where μ^+ and μ^- are the averages of the positive (d_τ^+) and negative (d_τ^+) triplet distances in a mini-batch.

6.2.2 Triplet-loss with attention

In this chapter, we also explore an extension of the triplet-loss network by applying attention mechanism. This technique has also been used in the work to train a Siamese network (Chowdhury et al., 2017). The network architecture is shown in Figure 6.5. Unlike the conventional triplet network (as described above), the Average Pooling layer (in Figure 6.5) obtains speaker representation ($\mathbf{h}^{'}$) by linearly combining the hidden activations (denoted by $\{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_M\}$) after the first layer (bi-LSTM) with a weight vector, as given by:

$$\mathbf{h}^{'} = \sum_{i=0}^{M} \mathrm{w_i} \mathbf{h}_i,$$

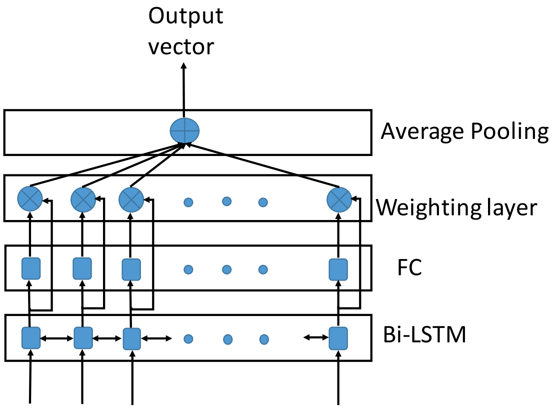

Figure 6.5: *The neural network architecture of triplet-loss approach with attention mechanism for text-dependent SV.*

where $\mathrm{w_i}$ is the weight of i^{th} speech frame. The weights are computed by using a FC (denoted by the function g, the first FC layer of Figure 6.5) and a tanh activation function as follows

$$\mathrm{w_i} = \tanh(\mathrm{g}(\mathbf{h}_i)), \tag{6.4}$$

and finally the weights are normalized over an utterance to obtain the attention vector as follows

$$\mathrm{w_i} = \frac{\mathrm{w_i}}{\sum_j \mathrm{w_j}}. \tag{6.5}$$

The attention based speaker representation ($\mathbf{h}^{'}$) is then used for training the triplet-loss as given by Equation 6.2.

6.3 Distance function for DNN

In the end-to-end approaches described in the last section, speaker representation is obtained by computing the mean or weighted mean of the hidden activations in DNN. The distance between two utterances is computed as the Euclidean distance between their respective speaker vectors. However, this approach has not shown to outperform the state-of-the-art i-vector system on a publicly available dataset (Bhattacharya et al., 2016). We hypothesize that the degraded performance is due to the averaging operation which may ignore the content information of the speech signal. In the past, it has been shown that performance of text-dependent SV can be substantially improved by exploiting phonetic information of an utterance (Chen et al., 2015b). In this section, we explore distance function that exploits phonetic information of the speech signal implicitly (i.e. without using text-transcript).

For the proposed loss function, the network architecture is similar to that of the triplet loss network (Figure 6.4). The main difference is that the Average Pooling layer has been removed from the network. Thus, an utterance produces as many hidden speaker representations as the number of speech frames. Let us assume the two utterances ($\mathbf{H}_e$ and $\mathbf{H}_t$) produce the following hidden representations {$\mathbf{h}_{e,1}$, $\mathbf{h}_{e,2}$, $\mathbf{h}_{e,3}$, $\cdots$, $\mathbf{h}_{e,i}$, $\cdots$, $\mathbf{h}_{e,R}$ } and {$\mathbf{h}_{t,1}$, $\mathbf{h}_{t,2}$, $\mathbf{h}_{t,3}$, $\cdots$, $\mathbf{h}_{t,j}$, $\cdots$, $\mathbf{h}_{t,C}$ }. We explore three distance functions as described below.

- **Average distance**: The average distance (D_{avg}) between two utterances $\mathbf{H}_e$ and $\mathbf{H}_t$ is given by:

  $$\mathrm{D}_{avg}(\mathbf{H}_e, \mathbf{H}_t) = \frac{1}{RC}\sum_{i,j} \mathrm{d}(\mathbf{h}_{e,i}, \mathbf{h}_{t,j}),$$

  where d is Euclidean distance between two vectors ($\mathbf{h}_{e,i}$ and $\mathbf{h}_{t,j}$). It is to be noted that if cosine-distance is used as d(.), then with some algebraic manipulation it can be seen that the average distance (D_{avg}) is same as the conventional triplet loss function of Equation 6.2.

- **Minimum distance**: The next loss function that we consider is based on scoring using the common set of phones between two utterances. Assuming that the hidden representation of frame of speech contains phonetic information as well, the minimum distance (D_{min}) is obtained as follows:

  $$\mathrm{D}_{min}(\mathbf{H}_e, \mathbf{H}_t) = \frac{1}{C}\sum_{j} min_i \mathrm{d}(\mathbf{h}_{e,i}, \mathbf{h}_{t,j}). \tag{6.6}$$

  This type of distance function has been used in the previous chapter (in the i-vector framework) but mainly as a post-processing step. It is to be noted that this proposed distance is not symmetric since $\mathrm{D}_{min}(\mathbf{H}_e, \mathbf{H}_t) \neq \mathrm{D}_{min}(\mathbf{H}_t, \mathbf{H}_e)$. The minimum function in Equation 6.6 aims to find the closest match of an utterance with hidden representation $\mathbf{h}_{t,j}$ against other features in $\mathbf{H}_e$. The minimum distance function assumes that the lexical content of $\mathbf{H}_t$ occurs in $\mathbf{H}_e$. Thus, the triplet-mining is performed in such a manner so as to preserve this condition during training.

- **Attention based distance function**: The previous loss function (minimum distance) does not take into account that some of the hidden representations in $\mathbf{H}_t$ contain more speaker discriminating information than the others. In order to incorporate this information in the loss function, we propose to apply the following attention based distance function (D_{attn}):

  $$\mathrm{D}_{attn}(\mathbf{H_e}, \mathbf{H_t}) = \sum_{j} \mathrm{w}_j min_i \mathrm{d}(\mathbf{h}_{e,i}, \mathbf{h}_{t,j}), \tag{6.7}$$

  where w_j is the weight of the j^{th} hidden representation and can be computed by using a FC layer as given by Equations 6.4 and 6.5. The network for performing this optimization is similar to one described in Section 6.2.2, the difference being in the distance function. We train the network using Equation 6.2 by replacing d(.) with the respective proposed distance functions D_{avg}, D_{min} and D_{attn}.

The triplet-networks described earlier can be used in an end-to-end training to produce SV scores directly by applying the appropriate distance function. It has been shown in literature, that applying a PLDA is beneficial for SV (Snyder et al., 2018). In this chapter, we apply a PLDA to compute the distance function d(.) as well. The PLDA is trained on these hidden representations $\mathbf{H}_e$ and $\mathbf{H}_t$ by using the speaker labels for training.

6.4 Triplet-loss using first order statistics

In this section, we describe an approach to use text-transcript for training triplet-loss networks. In literature, network such as Siamese network (similar to triplet-loss), is trained using phonetic information by employing first-order statistics of hidden representations (Zhang et al., 2016). Intuitively, first order statistics summarize the contribution of speakers per phonetic unit. First order statistics ($\mathbf{m}_c$) of an utterance with hidden representations, $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_T\}$ is computed as follows:

$$\mathbf{m}_c = \sum_i \mathbf{h}_i \mathbb{1}_\mathrm{i},$$

where $\mathbb{1}_\mathrm{i}$ is an indicator function that outputs one if i^{th} frame is assigned to c^{th} phonetic unit. We apply the same process for training the triplet-network and the technique for using first-order statistics is shown in Figure 6.6. To obtain the first order statistics, a state-of-the-art automatic speech recognizer is applied to align the development data with mono-phone units. The modified triplet loss function minimizes the hidden representation of anchor, positive and negative utterances based on the first order statistics as shown in Figure 6.6, (similar to the loss function in Zhang et al. (2016)) and is given by:

$$E_{trip,fos}(\tau) = \sum_c \mathrm{d}(\mathbf{m}_c^a, \mathbf{m}_c^p) - \mathrm{d}(\mathbf{m}_c^a, \mathbf{m}_c^n) + \alpha,$$

where $\mathbf{m}_c^a$, $\mathbf{m}_c^p$ and $\mathbf{m}_c^n$ are the first-order statistics of c^{th} cluster of the anchor, positive and negative instances respectively. The loss function ($E_{trip,fos}$) is fully differentiable and the gradients can be estimated efficiently with back propagation algorithm. Once the network has been trained, the outputs after the first layer (bi-LSTM) of Figure 6.6 are collected to obtain speaker vector. A PLDA is further trained on these vectors to produce SV scores.

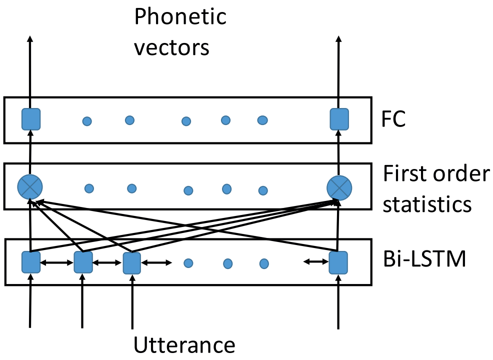

Figure 6.6: *First order statistics for training triplet-loss network.*

6.5 Experimental Setup

In this section, experimental setup of the baseline and the proposed systems are described.

6.5.1 Evaluation and Training Data

Experiments are performed on Part 1 and 3 portion of the RSR2015 dataset as described in Section 2.10.2. In this chapter, we are interested in evaluating the proposed techniques for speaker-mismatch trials only, for both the tasks. We used RSR2015 data (development and background) since using additional out-of-domain data (Fisher corpora) has not been found to be helpful for DNN based speaker embeddings. In-order to be consistent with the amount of training data, we used RSR2015 as the training data for the baseline and the proposed systems. Thus no out-of-domain data is used. The training data consists of 61 k utterances spoken by 94 speakers.

6.5.2 i-vector

We applied the standard MFCC features (with STG) as used in all the chapters. Due to the limited training data, we trained a smaller dimensional i-vector extractor. A 512 mixture GMM-UBM is trained on the training data and 200 dimensional i-vector extractor is trained subsequently. Finally, a PLDA is trained as part of the standard recipe of text-independent system with speaker labels of training data.

6.5.3 Speaker embeddings and end-to-end SV

For the d-Vector, we trained a single layer FC based system with the training data of RSR2015. We used only 940 utterances as the cross-validation data from the 94 speakers. We obtained 100% accuracy on the training and development data using the cross entropy loss function.

For the triplet-loss network, we use hard-triplets for training the network (Schroff et al., 2015). At any epoch, we generate triplets $(\mathbf{X}^a, \mathbf{X}^p, \mathbf{X}^n)$ such that the phonetic content of these utterances ($\mathbf{X}^a$, $\mathbf{X}^p$, $\mathbf{X}^n$) has maximal overlap. This leads to creating a total of 200 k triplets per epoch. We randomly choose a subset of these triplets to train the triplet-loss network. A learning rate of 0.001 was used throughout the experiments. A 1 k dimensional hidden layer is used in all the experiments. Pytorch was used for performing the experiments (Pytorch, 2017).

6.6 Experimental Results and Discussions

In this section, we describe the results obtained with the baseline and the proposed systems. We evaluated the performance of the following systems on **fixed-phrase** and **random-digit strings** tasks:

- **MAP$^{\text{GMM}}$**: This the baseline GMM-UBM as described in Section 2.4.

- **i-vector**: This is the conventional i-vector PLDA employing GMM-UBM. A PLDA is trained as the backend classifier.

- **d-Vector**: For **d-Vector**, a FC hidden layer is used as the network architecture for obtaining speaker representation. Section 6.1.1 describes the conventional technique to apply **d-Vector**. The **d-Vector** employs a PLDA model for scoring.

- **Spk-Phn**: This approach involves minimizing the speaker and phonetic losses as described in Section 6.1.3. We used only one hidden FC layer with ReLU activation function. The hidden activations from last layer of DNN are averaged to obtain speaker representation. A PLDA is further trained on these representations for obtaining SV scores.

- **Uttr-Embed**: This approach is described in Section 6.1.2 and consists of obtaining speaker embedding for an utterance. A back-end classifier, such as PLDA, is trained on top of speaker embeddings to produce SV scores.

- **Triplet**: This system optimizes the triplet-loss function on three utterances. The triplet-loss network is described in Section 6.2. This technique uses a bi-LSTM and a FC layer. Speaker representation of an utterance is obtained by collecting the activations after the Average Pooling layer (See Figure 6.4). Furthermore, a PLDA model is trained on these representations. The proposed triplet-loss network applying first order statistics (as described in Section 6.4) is referred to as **Triplet-Stats**. For this approach, the output activations after the bi-LSTM layer of Figure 6.6 are collected to obtain speaker vector. A PLDA is trained on these vectors for producing SV score. The approach applying attention based mechanism (as described in Section 6.2.2) is referred to as **Triplet-Attn** and described in Section 6.2.2.

- **Proposed systems**: The triplet-loss network applying the average, minimum and attention-based distance are referred to as **Avg-Dist**, **Min-Dist** and **Attn-Dist** (as described in Section 6.3) respectively. The proposed techniques are evaluated using

Table 6.1: *Performance of the various systems in terms of EER (%) on RSR2015* ***fixed-phrase*** *and* ***random-digit strings****. The* ***MAP***$^{\text{GMM}}$ *performs the best.*

Systems	**fixed-phrase** (%)	**random-digit strings** (%)
MAP$^{\textbf{GMM}}$	**2.3**	**7.8**
Ivec$^{\textbf{GMM}}_{\textbf{PLDA}}$	4.3	11.8
d-Vector	4.5	12.3
Uttr-Embed	4.5	12.4
Spk-Phn	4.3	12.7

Table 6.2: *Performance of the various triplet-loss network in terms of EER (%) on RSR2015* ***fixed-phrase*** *and* ***random-digit strings****. The* ***Triplet-Attn*** *performs the best.*

Systems	**fixed-phrase** (%)	**random-digit strings**
Triplet	6.9	15.2
Triplet-Attn	**4.4**	**11.7**
Triplet-Stats	-	12.4

end-to-end objective function. We also evaluate the performance of the proposed approaches on applying a PLDA as a post-processing step.

6.6.1 Baseline

We first describe the i-vector, GMM-UBM based SV. From Table 6.1, it can be observed that the **MAP**$^{\textbf{GMM}}$ significantly outperforms the **Ivec**$^{\textbf{GMM}}_{\textbf{PLDA}}$ for both the tasks. The peformance of **Ivec**$^{\textbf{GMM}}_{\textbf{PLDA}}$ is worse than the result reported in Chapter 4 for **fixed-phrase** task. This difference in performance could be due to that the **Ivec**$^{\textbf{GMM}}_{\textbf{PLDA}}$ in Chapter 4 is trained using Fisher data. However, the performance of **MAP**$^{\textbf{GMM}}$ is significantly better than the result of GMM-UBM in the last chapter for **random-digit strings**. We consider the **MAP**$^{\textbf{GMM}}$ as the baseline system for **fixed-phrase** and **random-digit strings**.

6.6.2 DNN based speaker embedding

Table 6.1 shows the performance of **d-Vector**, **Uttr-Embed**, **Spk-Phn** for SV. The SV results of these DNN based speaker embedding approaches are obtained by a PLDA model. From Table 6.1, it can be observed that SV performances of these approaches are very close to each other. The **Spk-Phn** provides good performance for the **fixed-phrase** task, while **d-Vector** provides good result for the **random-digit strings**. However, the performances of these approaches (**d-Vector**, **Uttr-Embed**, **Spk-Phn**) are worse than the **MAP**$^{\textbf{GMM}}$.

Table 6.3: *Performance of the various proposed systems in terms of EER (%) on RSR2015 for* ***fixed-phrase*** *and* ***random-digit strings****. The systems are evaluated using end-to-end objective function.*

Systems	**fixed-phrase** (%)	**random-digit strings** (%)
Avg-Dist	11.2	29.7
Min-Dist	**1.8**	**7.6**
Attn-Dist	9.4	29.1

6.6.3 Triplet-loss

From Table 6.2, we observe that the performance of **Triplet** is worse than **d-Vector** and **MAP**$^{\textbf{GMM}}$. It is to be noted that **Triplet** provides an EER of 23.2% using end-to-end loss (PLDA was not applied in this system) for **random-digit strings** task. An explanation of the poor performance of the triplet-loss approach could be that it requires large speaker population to provide results comparable to GMM-UBM. The **Triplet-Stats** performs better than **Triplet** for **random-digit strings** which indicate that text-transcription is beneficial for SV. The network employing **Triplet-Stats** could not be trained for the **fixed-phrase** task as we observed that the objective function did not converge during training.

We now describe the triplet-loss approach using attention mechanism as described in Section 6.2.2. The **Triplet-Attn** performs better than **Triplet** for both the tasks. Thus showing the importance of attention weights in producing speaker representation of an utterance. Furthermore, **Triplet-Attn** outperforms the baseline **Ivec**$^{\textbf{GMM}}_{\textbf{PLDA}}$ by 0.1% absolute EER for **random-digit strings**. However, **Triplet-Attn** perform worse than **MAP**$^{\textbf{GMM}}$.

6.6.4 Proposed distance based approaches

Table 6.3 shows the performance of the proposed SV approaches evaluated against their respective end-to-end objective function as described in Section 6.3. The results show that the **Min-Dist** performs the best among the proposed approaches and outperforms **MAP**$^{\textbf{GMM}}$ by relative EER of 21.7% (from 2.3% to 1.8% absolute) and 2.6% (from 7.8% to 7.6% absolute) for **fixed-phrase** and **random-digit strings** respectively. The performances of **Avg-Dist** and **Attn-Dist** are worse than the **Min-Dist**. Thus, showing the importance of selecting common phonetic regions between utterances for producing SV scores.

We also investigate the use of PLDA to compute end-to-end scores instead of Euclidean distance. Table 6.4 shows the performance of the proposed system on applying PLDA model. We observe that SV performances of all the systems improve on using the back-end classifier on top of the hidden DNN representations. The **Attn-Dist** benefits the most from applying PLDA with absolute improvement in EER of 23.7% (from 29.1% to 5.4%) and it outperforms the **MAP**$^{\textbf{GMM}}$ by 31% relative EER (from 7.8% to 5.4% absolute). The **Min-Dist** provides the

Table 6.4: *Performance of the various systems in terms of EER (%) on RSR2015* ***fixed-phrase*** *and* ***random-digit strings****. The systems are evaluated using end-to-end objective function using a back-end PLDA classifier.*

Systems	**fixed-phrase** (%)	**random-digit strings** (%)
Avg-Dist	3.4	15.7
Min-Dist	**1.2**	**5.0**
Attn-Dist	1.4	5.4

Table 6.5: *Performance of the best performing systems in terms of EER (%)/minDCF (× 100) for* ***fixed-phrase*** *and* ***random-digit strings****.*

Systems	**fixed-phrase**	**random-digit strings**
MAP$^{\text{GMM}}$ (Table 6.1)	2.3/1.03	7.8/3.71
Min-Dist (Table 6.4)	**1.2/0.63**	**5.0/2.56**
Attn-Dist (Table 6.4)	1.4/0.68	5.4/2.75

best performance with 1.2% and 5.0% for **fixed-phrase** and **random-digit strings** respectively. Furthermore, we investigated whether the attention weights in Equation 6.7 put more emphasize to vowels and nasals compared to other phoneme units. However, we did not find any correlation between the attention weights and phoneme units.

6.6.5 Summary of experiments on the RSR part 1 and 3

The minDCF and DET plots of some of the best performing systems on **fixed-phrase** and **random-digit strings** are presented in Table 6.5 and Figures 6.7 and 6.8 respectively. The systems include, (i) **MAP$^{\text{GMM}}$**, (ii) **Min-Dist**, and (ii) **Attn-Dist**. It can be observed from Table 6.5 that **Min-Dist** performs better than the baseline **MAP$^{\text{GMM}}$** by 41% relative minDCF (from 1.03 to 0.63 absolute) and 31% relative minDCF (from 3.71 to 2.56 absolute) for **fixed-phrase** and **random-digit strings**.

6.7 Conclusions

This chapter explores novel ideas in building end-to-end DNN based text-dependent SV system. The baseline approach consists of mapping a variable length speech segment to a fixed dimensional speaker vector by estimating the mean of hidden representations in DNN structure. The distance between two utterances is obtained by computing L2 norm between the vectors. This approach performs worse than the conventional GMM-UBM based SV on a publicly available corpora. We believe that a degraded performance is due to the employed averaging operation, which may not capture the phonetic information of an utterance. We

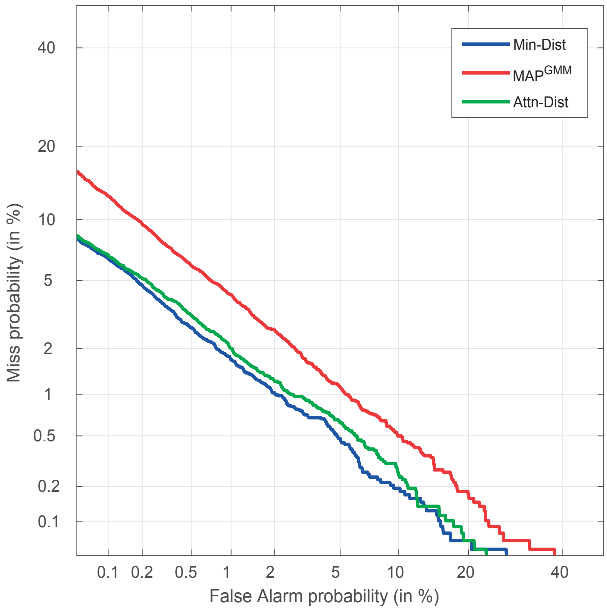

Figure 6.7: *DET curve of the systems presented in Table 6.5 for* ***fixed-phrase*** *task.*

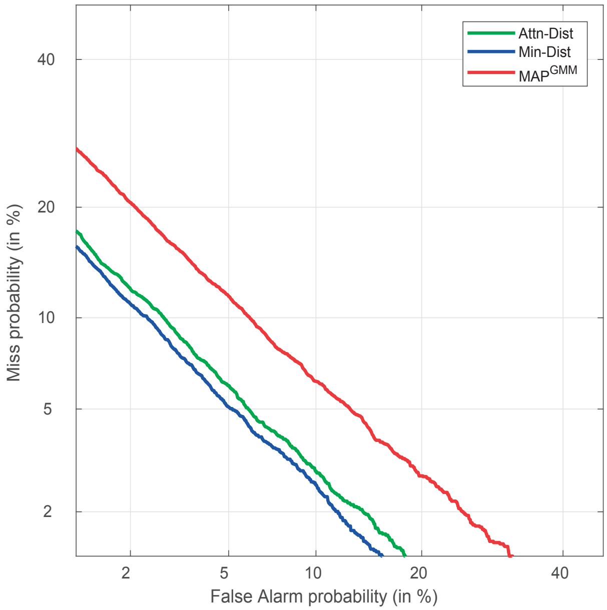

Figure 6.8: *DET curve of the systems presented in Table 6.5 for* ***random-digit strings****.*

propose to incorporate content information of the speech signal by computing distance function with linguistic units co-occurring between enrollment and test data. The whole network is optimized by employing a triplet-loss objective in an end-to-end fashion to estimate SV scores. Experiments on the RSR2015 dataset indicate that the proposed approach outperforms **MAP^{GMM}** by 48% and 36% relative EER for **fixed-phrase** and **random-digit** conditions respectively.

7 Conclusions and future work

7.1 Conclusions

In this thesis, we explored the application of phonetic knowledge (in addition to speaker characteristics) to address text-independent and text-dependent SV tasks. The phonetic knowledge is used in the i-vector PLDA framework by computing sufficient statistics computed from the senone posterior probabilities obtained at the output of DNN acoustic model. This technique was extended in several ways to address various text-independent and text-dependent SV scenarios.

For text-independent SV, SGMM model was proposed and employed to extract low dimensional speaker vectors that capture speaker characteristics in addition to phonetic knowledge. The performance was further improved, replacing SGMM by HMM/DNN allowing to extract complementary linguistic information using DNN based ASR. In addition to senone posterior probabilities estimated directly from the DNN output, the posteriors were also extracted from ASR word recognition lattices (i.e. smoothed by lexicon and language model), to be subsequently applied for i-vector extraction. The proposed approach performs better than the baseline i-vector system by 10% relative EER on Condition 5 of SRE10. We found a positive correlation between the phone and speaker recognition accuracies.

In this thesis, we explored two text-dependent SV scenarios, namely, (i) **fixed-phrase** and (ii) **random-digit strings**. For **fixed-phrase** case, the technique developed for text-independent SV was applied to extract i-vectors. Since this approach ignores information captured by a sequence of acoustic units, we developed new techniques to incorporate this information by using dynamic time warping combined with online i-vectors. The proposed approach outperforms the baseline approach by 95% and 70% relative EER on content and speaker mismatch conditions respectively. This result shows the importance of online i-vectors and DTW algorithm to capture the sequence and speaker information effectively. For **random digit strings**, we explored a technique that aims to match the lexical-content of the enrollment to the test data using online i-vectors as features. In particular, the proposed approach performs better than the baseline system by 12% relative EER which shows the importance of matching phonetic units between utterances.

We also explored the application of DNN based speaker embedding for text-dependent SV. Unlike building DNN to classify acoustic units, the DNN was trained (in an end-to-end fashion) to directly discriminate speakers. As opposed to training several SV components independently, we incorporated both speaker and phonetic information in the neural network framework for text-dependent SV (**fixed-phrase** and **random-digit strings**). We exploited phonetic information by computing a distance function with linguistic units common to both enrollment and test data. The whole network was then optimized by employing a triplet-loss objective function to produce SV scores. Experiments on the **fixed-phrase** and **random-digit strings** showed that the proposed approach improved upon the baseline system by 36% and 48% relative EER. This result indicate the importance of applying phonetic information for end-to-end SV.

7.2 Future work

In this work, the text-independent SVs were evaluated on NIST SRE 2010, which has English speakers only (Chapter 3) (Martin and Greenberg, 2010). Unfortunately, the dataset lacks linguistic variability as compared to more recent NIST evaluations (SRE 2012 and SRE 2016) (Sadjadi et al., 2017; Greenberg et al., 2013). These more recent NIST challenges evaluates SV approaches across multiple languages and acoustic conditions. Therefore, the techniques proposed in Chapter 3 need to be evaluated under the conditions available in these benchmark datasets. In this context, multi-lingual ASR might offer a good solution to replace mono-lingual engines in the DNN i-vector framework (Lei et al., 2014) to deal with under-resources languages.

The end-to-end approach (as proposed and explored in Chapter 6) can be extended for text-independent SV. In this case, the spoken content in the test data is not necessarily present in the enrollment utterance. Thus, the loss function for training the neural network (as presented in Chapter 6) can be modified to reflect this case. Therefore, exploring the loss function for triplet-loss approach in text-independent SV could be a potential research direction. Furthermore, the triplet-loss approach as presented in Chapter 6, requires selecting triplet instances for training. We choose triplets by exploiting phonetic information of utterances. In text-independent SV, efficient strategies for obtaining triplets need to be investigated that do not require task-related knowledge.

Bibliography

Kanae Amino, Tsutomu Sugawara, and Takayuki Arai. Idiosyncrasy of nasal sounds in human speaker identification and their acoustic properties. *Acoustical science and technology*, 27 (4):233–235, 2006.

Roland Auckenthaler, Michael Carey, and Harvey Lloyd-Thomas. Score normalization for text-independent speaker verification systems. *Digital Signal Processing*, 10(1-3):42–54, 2000.

Brendan J Baker, Robert J Vogt, and Sridha Sridharan. Gaussian mixture modelling of broad phonetic and syllabic events for text-independent speaker verification. In *European Conference on Speech Communication and Technology*, pages 2429–2432, 2005.

Claude Barras and J-L Gauvain. Feature and score normalization for speaker verification of cellular data. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 46–49, 2003.

Laurent Besacier, Jean-François Bonastre, and Corinne Fredouille. Localization and selection of speaker-specific information with statistical modeling. *Speech Communication*, 31(2-3): 89–106, 2000.

Gautam Bhattacharya, Jahangir Alam, Themos Stafylakis, and Patrick Kenny. Deep neural network based text-dependent speaker recognition: Preliminary results. *Odyssey 2016*, pages 9–15, 2016.

Christopher Bishop. *PATTERN RECOGNITION AND MACHINE LEARNING*. Springer-Verlag New York, 2016.

Jean-François Bonastre, Philippe Morin, and Jean-Claude Junqua. Gaussian dynamic warping (gdw) method applied to text-dependent speaker detection and verification. In *Eighth European Conference on Speech Communication and Technology*, 2003.

Hervé Bredin. Tristounet: triplet loss for speaker turn embedding. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5430–5434. IEEE, 2017.

Michael Brown and L Rabiner. An adaptive, ordered, graph search technique for dynamic time warping for isolated word recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 30(4):535–544, 1982.

Niko Brümmer. FoCal multi-class: Toolkit for evaluation, fusion and calibration of multi-class recognition scores—tutorial and user manual. *Software available at http://sites. google. com/site/nikobrummer/focalmulticlass*, 2007.

Niko Brümmer and Edwards de Villiers. The BOSARIS toolkit, 2013.

William M Campbell, Joseph P Campbell, Douglas A Reynolds, Douglas A Jones, and Timothy R Leek. Phonetic speaker recognition with support vector machines. In *Advances in neural information processing systems*, 2003.

Joseph P Campbell Jr. Speaker recognition: A tutorial. *Proceedings of the IEEE*, pages 1437–1462, 1997.

Guoguo Chen, Carolina Parada, and Tara N Sainath. Query-by-example keyword spotting using long short-term memory networks. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5236–5240. IEEE, 2015a.

Liping Chen, Kong Aik Lee, Bin Ma, Wu Guo, Haizhou Li, and Li-Rong Dai. Phone-centric local variability vector for text-constrained speaker verification. In *Proceedings of Interspeech*, 2015b.

Nanxin Chen, Yanmin Qian, and Kai Yu. Multi-task learning for text-dependent speaker verification. In *Proceedings of Interspeech*, 2015c.

FA Chowdhury, Quan Wang, Ignacio Lopez Moreno, and Li Wan. Attention-based models for text-dependent speaker verification. *arXiv preprint arXiv:1710.10470*, 2017.

George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):30–42, 2012.

A Das, VP Kumar, et al. Text-dependent speaker-recognition using one-pass dynamic programming algorithm. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages I–I. IEEE, 2006.

Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2011.

Subhadeep Dey, Srikanth Madikeri, Marc Ferras, and Petr Motlicek. Deep neural network based posteriors for text-dependent speaker verification. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5050–5054. IEEE, 2016a.

Subhadeep Dey, Srikanth Madikeri, and Petr Motlicek. Information theoretic clustering for unsupervised domain-adaptation. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5580–5584. IEEE, 2016b.

Subhadeep Dey, Petr Motlicek, Srikanth Madikeri, and Marc Ferras. Exploiting sequence information for text-dependent speaker verification. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5370–5374. IEEE, 2017a.

Subhadeep Dey, Petr Motlicek, Srikanth Madikeri, and Marc Ferras. Template-matching for text-dependent speaker verification. *Speech Communication*, 88:96–105, 2017b.

Subhadeep Dey, Takafumi Koshinaka, Petr Motlicek, and Srikanth Madikeri. DNN based speaker embedding using content information for text-dependent speaker verification. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018a.

Subhadeep Dey, Srikanth Madikeri, and Petr Motlicek. End-to-end text-dependent speaker verification using novel distance measures. In *Proceedings of Interspeech*, 2018b.

George R Doddington, Mark A Przybocki, Alvin F Martin, and Douglas A Reynolds. The NIST speaker recognition evaluation–overview, methodology, systems, results, perspective. *Speech Communication*, 31(2):225–254, 2000.

Richard O Duda and Peter E Hart. Pattern recognition and scene analysis, 1973.

Marc Ferras, Cheung Chi Leung, Claude Barras, and Jean-Luc Gauvain. Constrained MLLR for speaker recognition. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages 53–56. IEEE, 2007.

Mark Gales and Steve Young. The application of hidden Markov models in speech recognition. *Foundations and trends in signal processing*, 1(3):195–304, 2008.

Daniel Garcia-Romero. *Robust speaker recognition based on latent variable models*. PhD thesis, 2012.

Daniel Garcia-Romero and Carol Y Espy-Wilson. Analysis of i-vector length normalization in speaker recognition systems. In *Proceedings of Interspeech*, 2011.

Daniel Garcia-Romero and Alan McCree. Supervised domain adaptation for i-vector based speaker recognition. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4047–4051. IEEE, 2014.

Craig S Greenberg, Vincent M Stanford, Alvin F Martin, Meghana Yadagiri, George R Doddington, John J Godfrey, and Jaime Hernandez-Cordero. The 2012 NIST speaker recognition evaluation. In *Proceedings of Interspeech*, pages 1971–1975, 2013.

Dan Gutman and Yuval Bistritz. Speaker verification using phoneme-adapted Gaussian mixture models. In *Proceedings of European Signal Processing Conference*, pages 1–4. IEEE, 2002.

Matthieu Hébert and Daniel Boies. T-norm for text-dependent commercial speaker verification applications: Effect of lexical mismatch. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages I–729. IEEE, 2005.

Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer. End-to-end text-dependent speaker verification. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5115–5119. IEEE, 2016.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

David Imseng, Petr Motlicek, Philip N. Garner, and Hervé Bourlard. Impact of deep mlp architecture on different acoustic modeling techniques for under-resourced speech recognition. In *IEEE workshop on Automatic Speech Recognition and Understanding*, 2013.

David Imseng, Petr Motlicek, Hervé Bourlard, and Philip N Garner. Using out-of-language data to improve an under-resourced speech recognizer. *Speech communication*, 56:142–151, 2014.

Sarfaraz Jelil, Rohan Kumar Das, Rohit Sinha, and SR Mahadeva Prasanna. Speaker verification using Gaussian posteriorgrams on fixed phrase short utterances. In *Proceedings of Interspeech*, 2015.

Patrick Kenny, Gilles Boulianne, Pierre Ouellet, and Pierre Dumouchel. Joint factor analysis versus eigenchannels in speaker recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(4):1435–1447, 2007.

Patrick Kenny, Themos Stafylakis, J Alam, Pierre Ouellet, and Marcel Kockmann. Joint factor analysis for text-dependent speaker verification. In *Odyssey*, pages 1–8, 2014a.

Patrick Kenny, Themos Stafylakis, Pierre Ouellet, and Md Jahangir Alam. JFA-based front ends for speaker recognition. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1705–1709. IEEE, 2014b.

Eamonn Keogh and Chotirat Ann Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and information systems*, 7(3):358–386, 2005.

Tomi Kinnunen and Haizhou Li. An overview of text-independent speaker recognition: From features to supervectors. *Speech communication*, 52(1):12–40, 2010.

Anthony Larcher, Jean-François Bonastre, and John SD Mason. Reinforced temporal structure information for embedded utterance-based speaker recognition. In *Proceedings of Interspeech*, pages 371–374, 2008.

Anthony Larcher, Kong Aik Lee, Bin Ma, and Haizhou Li. Phonetically-constrained PLDA modeling for text-dependent speaker verification with multiple short utterances. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7673–7677. IEEE, 2013.

Anthony Larcher, Kong Aik Lee, Bin Ma, and Haizhou Li. Modelling the alternative hypothesis for text-dependent speaker verification. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 734–738. IEEE, 2014a.

Anthony Larcher, Kong Aik Lee, Bin Ma, and Haizhou Li. Text-dependent speaker verification: Classifiers, databases and RSR2015. *Speech Communication*, 60:56–77, 2014b.

C-H Lee and J-L Gauvain. Speaker adaptation based on MAP estimation of HMM parameters. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 558–561. IEEE, 1993.

Kong Aik Lee, Anthony Larcher, Guangsen Wang, Patrick Kenny, Niko Brümmer, David Van Leeuwen, Hagai Aronowitz, Marcel Kockmann, Carlos Vaquero, Bin Ma, et al. The RedDots data collection for speaker recognition. In *Proceedings of Interspeech*, pages 2996–3000, 2015.

Yun Lei, Nicolas Scheffer, Luciana Ferrer, and Mitchell McLaren. A novel scheme for speaker recognition using a phonetically-aware deep neural network. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1695–1699. IEEE, 2014.

Chao Li, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu. Deep speaker: an end-to-end neural speaker embedding system. *arXiv preprint arXiv:1705.02304*, 2017.

Srikanth Madikeri, Ivan Himawan, Petr Motlicek, and Marc Ferras. Integrating online i-vector extractor with information bottleneck based speaker diarization system. In *Proceedings of Interspeech*, pages 3105–3109, 2015.

Srikanth Madikeri, Petr Motlicek, Marc Ferras, and Subhadeep Dey. Analysis of posterior estimation approaches to i-vector extraction for speaker recognition. Idiap-RR Idiap-Internal-RR-118-2016, Idiap, July 2016.

Alvin Martin, George Doddington, Terri Kamm, Mark Ordowski, and Mark Przybocki. The DET curve in assessment of detection task performance. Technical report, National Inst of Standards and Technology Gaithersburg MD, 1997.

Alvin F Martin and Craig S Greenberg. The NIST 2010 speaker recognition evaluation. In *Proceedings of International Speech Communication Association*, 2010.

Tomoko Matsui and Sadaoki Furui. Concatenated phoneme models for text-variable speaker recognition. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 391–394. IEEE, 1993.

Tomoko Matsui and Sadaoki Furui. Speaker adaptation of tied-mixture-based phoneme models for text-prompted speaker recognition. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages I–125. IEEE, 1994.

Ajili Moez, Bonastre Jean-François, Ben Kheder Waad, Rossato Solange, and Kahn Juliette. Phonetic content impact on forensic voice comparison. In *Spoken Language Technology Workshop (SLT)*, pages 210–217. IEEE, 2016.

Petr Motlicek, Fabio Valente, and Igor Szoke. Improving acoustic based keyword spotting using LVCSR lattices. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4413–4416. IEEE, 2012.

Petr Motlicek, Philip N. Garner, Namhoon Kim, and Jeongmi Cho. Accent adaptation using subspace Gaussian mixture models. In *Proceedings of International Conference on International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2013.

Petr Motlicek, Subhadeep Dey, Srikanth Madikeri, and Lukas Burget. Employment of subspace Gaussian mixture models in speaker recognition. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4445–4449. IEEE, 2015.

Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. VoxCeleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612*, 2017.

Vijayaditya Peddinti, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Reverberation robust acoustic modeling using i-vectors with time delay neural networks. In *Proceedings of Interspeech*, 2015.

Jason Pelecanos and Sridha Sridharan. Feature warping for robust speaker verification. 2001.

Daniel Povey. A tutorial-style introduction to subspace Gaussian mixture models for speech recognition. *Microsoft Research, Redmond, WA*, 2009.

Daniel Povey, Lukas Burget, Mohit Agarwal, Pinar Akyazi, Kai Feng, Arnab Ghoshal, Ondrej Glembek, Nagendra K Goel, Martin Karafiát, Ariya Rastrow, et al. Subspace Gaussian mixture models for speech recognition. In *Proceedings of International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 4330–4333, 2010.

Daniel Povey, Lukáš Burget, Mohit Agarwal, Pinar Akyazi, Feng Kai, Arnab Ghoshal, Ondřej Glembek, Nagendra Goel, Martin Karafiát, Ariya Rastrow, et al. The subspace Gaussian mixture model—A structured model for speech recognition. *Computer Speech & Language*, 25(2):404–439, 2011a.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The Kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, 2011b.

Simon JD Prince and James H Elder. Probabilistic linear discriminant analysis for inferences about identity. In *International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

Pytorch. https://github.com/pytorch, 2017. URL https://github.com/pytorch.

Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. Speaker verification using adapted Gaussian mixture models. *Digital signal processing*, 10(1-3):19–41, 2000.

Fred Richardson, Douglas Reynolds, and Najim Dehak. Deep neural network approaches to speaker and language recognition. *IEEE Signal Processing Letters*, 22(10):1671–1675, 2015.

Seyed Omid Sadjadi, Timothée Kheyrkhah, Audrey Tong, Craig Greenberg, Elliot Singer Reynolds, Lisa Mason, and Jaime Hernandez-Cordero. The 2016 NIST speaker recognition evaluation. *Proceedings of Interspeech*, pages 1353–1357, 2017.

Nicolas Scheffer and Yun Lei. Content matching for short duration speaker recognition. In *Proceedings of Interspeech*, pages 1317–1321, 2014.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (ICPR)*, pages 815–823, 2015.

Elizabeth Shriberg. Higher-level features in speaker recognition. In *Speaker Classification I*, pages 241–259. Springer, 2007.

David Snyder, Pegah Ghahremani, Daniel Povey, Daniel Garcia-Romero, Yishay Carmiel, and Sanjeev Khudanpur. Deep neural network-based speaker embeddings for end-to-end speaker verification. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 165–170. IEEE, 2016.

David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur. Deep neural network embeddings for text-independent speaker verification. *Proceedings of Interspeech*, pages 999–1003, 2017.

David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust DNN embeddings for speaker recognition. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

Themos Stafylakis, Patrick Kenny, Md Jahangir Alam, and Marcel Kockmann. JFA for speaker recognition with random digit strings. In *Proceedings of Interspeech*, 2015.

Themos Stafylakis, Md Jahangir Alam, and Patrick Kenny. Text-dependent speaker recognition with random digit strings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(7):1194–1203, 2016.

A Stolcke, E Shriberg, L Ferrer, S Kajarekar, K Sonmez, and G Tur. Speech recognition as feature extraction for speaker recognition. In *IEEE workshop on Signal Processing Applications for Public Security and Forensics*, pages 1–5. IEEE, 2007.

Andreas Stolcke, Luciana Ferrer, Sachin Kajarekar, Elizabeth Shriberg, and Anand Venkataraman. MLLR transforms as features in speaker recognition. In *Proceedings of European Conference on Speech Communication and Technology*, 2005.

Douglas E Sturim, Douglas A Reynolds, Robert B Dunn, and Thomas F Quatieri. Speaker verification using text-constrained Gaussian mixture models. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 677–680, 2002.

Hang Su and Steven Wegmann. Factor analysis based speaker verification using ASR. In *Proceedings of Interspeech*, pages 2223–2227, 2016.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez. Deep neural networks for small footprint text-dependent speaker verification. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4052–4056. IEEE, 2014.

Maarten Versteegh, Roland Thiolliere, Thomas Schatz, Xuan Nga Cao, Xavier Anguera, Aren Jansen, and Emmanuel Dupoux. The zero resource speech challenge 2015. In *Proceedings of Interspeech*, 2015.

Deepu Vijayasenan, Fabio Valente, and Hervé Bourlard. An information theoretic combination of MFCC and TDOA features for speaker diarization. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(2):431–438, 2011.

Guangsen Wang, Kong-Aik Lee, Trung Hieu Nguyen, Hanwu Sun, and Bin Ma. Joint speaker and lexical modeling for short-term characterization of speaker. In *Proceedings of Interspeech*, pages 415–419, 2016.

Bing Xiang, Upendra V Chaudhari, Jiri Navratil, Ganesh N Ramaswamy, and Ramesh A Gopinath. Short-time Gaussianization for robust speaker verification. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 681–684. IEEE, 2002.

Hossein Zeinali, Hossein Sameti, Lukáš Burget, Jan Černockỳ, Nooshin Maghsoodi, and Pavel Matějka. i-vector/HMM based text-dependent speaker verification system for RedDots challenge. In *Proceedings of Interspeech*, pages 440–444, 2016.

Hossein Zeinali, Hossein Sameti, Lukáš Burget, et al. Text-dependent speaker verification based on i-vectors, neural networks and hidden Markov models. *Computer Speech & Language*, 46:53–71, 2017.

Shi-Xiong Zhang, Zhuo Chen, Yong Zhao, Jinyu Li, and Yifan Gong. End-to-end attention based text-dependent speaker verification. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 171–178. IEEE, 2016.

Subhadeep Dey

PhD Student, EPFL/Idiap

☎ *+41 (766) 40 93 45*
✉ *subhadeep.dey@idiap.ch, dey@epfl.ch*
http://www.idiap.ch/~sdey
https://www.linkedin.com/in/subhadeep-dey-269a7997

Summary

A computer science engineer working on application of signal processing and machine learning to speech research.

Education

2018(expected) **Doctoral Student in Electrical Engineering**, *École Polytechnique Fédérale de Lausanne(EPFL)*, Switzerland.

2013 **Master of Science (by Research) - Computer Science**, *Indian Institute of Technology (IIT) Madras*, India.

2007 **Bachelors in Technology - Computer Science**, *National Institute of Technology Silchar*, India.

Work Experience

Apr'14 – Present **Research in Speaker Verification**, *Idiap/École Polytechnique Fédérale de Lausanne(EPFL)*, Switzerland.
Supervisors: Prof. Hervé Bourlard, Dr. Petr Motlicek

- Exploiting sequence information of the speech signal using template matching techniques has shown to capture the speaker characteristics better than the conventional approaches
- Exploring techniques to use phonetic units of the speech signal for improving speaker recognition
- Analyzing and exploring various clustering algorithms on the domain adaptation task for speaker verification

Aug'17 – Oct'17 **Research internship in Speaker Verification**, *NEC Corporation*, Japan.
Supervisor: Dr. Takafumi Koshinaka

- Exploring various Deep Neural Network architectures for end-to-end Speaker Verification

Jun'13 – Apr'14 **Project Engineer**, *IIT Guwahati*, India.
Project Title: Development of speech based multi-level person authentication system
Supervisors: Prof. S. R. M. Prasanna, Dr. Rohit Sinha

- Development of a multi-modal person authentication system for speaker verification using text dependent, text independent and voice password methods
- Addressing practical issues involved with the speaker verification system like minimizing the authentication time of an user

Aug'08 – Jan'12 **Research in Language Identification**, *IIT Madras*, India.
Thesis Title: Universal syllable models for Language Identification
Supervisor: Prof. Hema A. Murthy

- Syllable based tokens have been explored for capturing the phonotactics
- Universal syllable models based on maximum a posteriori criteria have been explored to perform language identification

Jan'12 – Dec'12	**Project Engineer**, *IIT Madras*, India. **Project Title:** Security and resilience of Next Generation Networks **Supervisor:** Prof. Hema A. Murthy ○ Using machine learning techniques for modeling computer network traffic characteristics.
Jul'07 – Jun'08	**Project Engineer**, *Indian Institute of Science Bangalore*, India. **Project Title:** Recognition of printed text in Kannada and Tamil **Supervisor:** Prof A. G. Ramakrishnan ○ Extraction of features derived from spline and graph representation of Kannada characters.

Computer skills

Programming:	C, C++, Java, Python
Tools:	Kaldi, HTK, Bosaris, MATLAB, Pytorch

Journal Publications

- M. Ferras, S. Madikeri, P. Motlicek, **S. Dey** and H. Bourlard *"A Large-Scale Open-Source Acoustic Simulator for Speaker Recognition"*, in IEEE Signal Processing Letters, 2016.
- **S. Dey**, P. Motlicek, S. Madikeri and M. Ferras *"Template-matching for Text-dependent Speaker Verification"*, in **Speech Communication**, 2017.

Conference and Other Publications

- **S. Dey**, S. Madikeri and P. Motlicek *"End-to-end text-dependent speaker verification using novel distance measures,"* to appear in **InterSpeech 2018**.
- **S. Dey**, T. Koshinaka, P. Motlicek and S. Madikeri *"DNN based speaker embedding using content information for text-dependent speaker verification,"* in **ICASSP 2018**.
- **S. Dey**, M. Ferras, P. Motlicek and S. Madikeri *"Content normalization for text-dependent speaker verification,"* in **InterSpeech 2017**.
- **S. Dey**, P. Motlicek, S. Madikeri and M. Ferras *"Exploiting sequence information for text-dependent speaker verification"* in **ICASSP 2017**.
- S. Madikeri, M. Ferras, P. Motlicek and **S. Dey** *"Intra-class covariance adaptation in PLDA back-ends for speaker verification"* in **ICASSP 2017**.
- S. Madikeri, **S. Dey**, M. Ferras, P. Motlicek and I. Himawan *"IDIAP submission to the NIST SRE 2016 speaker recognition evaluation"* in **NIST Speaker Recognition Evaluation (SRE 2016) - Post Evaluation Workshop** (Sixth place in terms of EER).
- **S. Dey**, S. Madikeri, M. Ferras and P. Motlicek *"Deep neural network based posteriors for text-dependent speaker verification"* in **ICASSP 2016, Shanghai China**.
- **S. Dey**, S. Madikeri, M. Ferras and P. Motlicek *"Information theoretic clustering for unsupervised domain-adaptation"* in **ICASSP 2016, Shanghai China**.
- M. Ferras, S. Madikeri, **S. Dey** and P. Motlicek *"Inter-task System Fusion for Speaker Recognition"* in **InterSpeech 2016, San Francisco, USA**.
- P. Motlicek, **S. Dey**, S. Madikeri and L. Burget *"Employment of Subspace Gaussian Mixture Models in speaker recognition,"* **ICASSP 2015, Brisbane, Australia.**
- **S. Dey** and H. Murthy *"Unsupervised clustering of syllables for language identification,"* **Eusipco 2012, Bucharest, Romania.**

Achievement

- IEEE Ganesh N Ramaswamy memorial student grant sponsored by IBM for the best paper in the speaker/language recognition area in ICASSP 2017.