SABANCI UNIVERSITY

Local Representations and Random Sampling for Speaker Verification

by

Yusuf Ziya Işık

Submitted to
the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

SABANCI UNIVERSITY

August 2010

LOCAL REPRESENTATIONS AND RANDOM SAMPLING FOR SPEAKER VERIFICATION

APPROVED BY

Assist. Prof. Dr. Hakan ERDOĞAN ....
(Thesis Supervisor)

Assoc. Prof. Dr. Berrin YANIKOĞLU ....

Assist. Prof. Dr. Cenk DEMİROĞLU ....

Assist. Prof. Dr. İlker HAMZAOĞLU ....

Assist. Prof. Dr. Müjdat ÇETİN ....

DATE OF APPROVAL: ....

To my family…

Acknowledgements

I would like to express my gratitude to my thesis supervisor Hakan Erdoğan for his invaluable guidance, tolerance, positiveness, support and encouragement throughout my thesis.

I would like to thank my thesis jury members Berrin Yanıkoğlu, Cenk Demiroğlu, İlker Hamzaoğlu and Müjdat Çetin for their valuable ideas.

My thanks go to all my colleagues at TÜBİTAK UEKAE for their friendship, and for providing the right environment for me to focus on my work at speaker verification. I owe special thanks to my project manager, Mehmet Uğur Doğan, for his encouraging leadership and tolerance.

LOCAL REPRESENTATIONS AND RANDOM SAMPLING FOR SPEAKER VERIFICATION

YUSUF ZİYA IŞIK

EE, M.Sc. Thesis, 2010

Thesis Supervisor: Hakan Erdoğan

Keywords: speaker verification, Gaussian mixture models, within-session variability, session invariant

Abstract

In text-independent speaker verification, studies focused on compensating intra-speaker variabilities at the modeling stage through the last decade. Intra-speaker variabilities may be due to channel effects, phonetic content or the speaker himself in the form of speaking style, emotional state, health or other similar factors. Joint Factor Analysis, Total Variability Space compensation, Nuisance Attribute Projection are some of the most successful approaches for inter-session variability compensation in the literature.

In this thesis, we criticize the assumptions of low dimensionality of channel space in these methods and propose to partition the acoustic space into local regions. Intra-speaker variability compensation may be done in each local space separately. Two architectures are proposed depending on whether the subsequent modeling and scoring steps will also be done locally or globally.

We have also focused on a particular component of intra-speaker variability, namely within-session variability. The main source of within-session variability is the differences in the phonetic content of speech segments in a single utterance. The variabilities in phonetic content may be either due to across acoustic event variabilities or due to differences in the actual realizations of the acoustic events. We propose a method to combat these variabilities through random sampling of training utterance. The method is shown to be effective both in short and long test utterances.

KONUŞMACI DOĞRULAMA İÇİN YEREL BETİMLEMELER VE RASGELE ÖRNEKLEME

YUSUF ZİYA IŞIK

EE, Yüksek Lisans Tezi, 2010

Tez Danışmanı: Hakan Erdoğan

Anahtar Kelimeler: konuşmacı doğrulama, Gauss karışım modelleri, oturum içi değişkenlik, oturum bağımsız

Özet

Son on yılda, metin bağımsız konuşmacı tanıma alanında yapılan çalışmalar konuşmacı içi değişintileri modelleme esnasında giderme üzerine odaklanmıştır. Konuşmacı içi değişintiler kanal etkilerinden, fonetik içerikten, veya konuşma stili, duygusal durum, sağlık ve benzeri sebeplerle konuşmacının kendisinden kaynaklanabilir. Ortak Faktör Analizi, Toplam Değişkenlik Uzayı, Sıkıntı Öznitelik İzdüşümü literatürde oturumlar arası değişkenlikleri gidermede kullanılan yöntemlerin en başarılılarındandır.

Bu çalışmada, önerilen metodlardaki kanal uzayının düşük boyutlu olma varsayımını irdeledik ve akustik uzayı yerel bölgelere ayırmayı önerdik. Konuşmacı içi değişintiler her yerel bölgede bağımsız olarak bastırıldı. İleriki modelleme ve skorlama safhalarının yerel mi yoksa global mi yapılacağına bağlı olarak iki farklı yapı önerildi.

Konuşmacı içi değişintinin elemanlarından biri olan oturum içi değişkenlikler üzerinde de çalışıldı. Oturum içi değişkenliklerin ana kaynağı bir ses dosyasının farklı kısımları arasındaki fonetik içerik farklılıklarıdır. Fonetik içerik farklılıkları, akustik birimler arası değişintilerden kaynaklanabileceği gibi aynı akustik birimin farklı çıkarımlarından da kaynaklanabilir. Bu değişintileri giderme amaçlı olarak, eğitim verisinin rasgele örneklenmesine dayalı bir metod önerdik. Önerilen metodun hem kısa hem de uzun test verilerinde etkin olduğu gösterildi.

Contents

List of Figures

List of Tables

Abbreviations

DCF	**D**etection **C**ost **F**unction
DET	**D**ecision **E**rror **T**rade-off
EER	**E**qual **E**rror **R**ate
EM	**E**xpectation **M**aximization
GMM	**G**aussian **M**ixture **M**odel
JFA	**J**oint **F**actor **A**nalysis
LDA	**L**inear **D**iscriminant **A**nalysis
LDC	**L**inguistic **D**ata **C**onsortium
MFCC	**M**el **F**requency **C**epstral **C**oefficients
NAP	**N**uisance **A**ttribute **P**rojection
NIST	**N**ational **I**nstitute of **S**tandards and **T**echnology
SCV	**S**peaker **C**haracterization **V**ector
SRE	**S**peaker **R**ecognition **E**valuation
SVM	**S**upport **V**ector **M**achine
TVS	**T**otal **V**ariability **S**pace
T-norm	**T**est **norm**alization
UBM	**U**niversal **B**ackground **M**odel
VAD	**V**oice **A**ctivity **D**etector
WCCN	**W**ithin **C**lass **C**ovariance **N**ormalization
Z-norm	**Z**ero **norm**alization

Chapter 1

Introduction

Speaker recognition is a generic term for extracting information regarding the identity of the person speaking in a given speech utterance. This problem can be divided into two subproblems, namely speaker identification and speaker verification, according to the possible number of target speakers. Speaker identification concerns with determining the identity of the speaker of a given test utterance from a pre-built target speaker database. On the other hand, speaker verification is a binary classification problem where we are given a test utterance and a claimed speaker id, and required to decide whether the speakers match or not. Speaker recognition may be text-dependent if we are only required to identify the target speaker when uttering a specific text or may be text-independent when no restriction is put on the content of the speech utterance.

Speaker verification is commercially more attractive than identification since it finds a significant place of usage in telephone based access-control systems, forensic applications, and investigation systems used by police forces and intelligence organizations. The focus of this thesis is text-independent speaker verification.

1.1 General Structure of Speaker Verification Systems

Speaker verification may be stated as a binary hypothesis test where the two hypothesis are:

- H0: Test speech utterance X is from the claimed speaker S,

- H1: Test speech utterance X is **not** from the claimed speaker S.

The optimum test to decide between two hypothesis is a likelihood ratio test given as:

$$\frac{p(X|H0)}{p(X|H1)} = \begin{cases} > \theta & accept \;\; H0 \\ < \theta & accept \;\; H1, \end{cases} \tag{1.1}$$

where $p(X|H0)$ is the likelihood of the hypothesis H0 and $p(X|H1)$ is the likelihood of H1 given X. Most of the approaches in speaker verification may be viewed as a Likelihood Ratio Detector. The block diagram for such a system is given in Figure 1.1. The front-end processing block represents the components used for extracting features carrying speaker-specific information. The features extracted are then scored with the claimed speaker model to test H0 hypothesis and with an alternative model for H1 hypothesis. There is usually a last step where score normalization and/or calibration is performed.

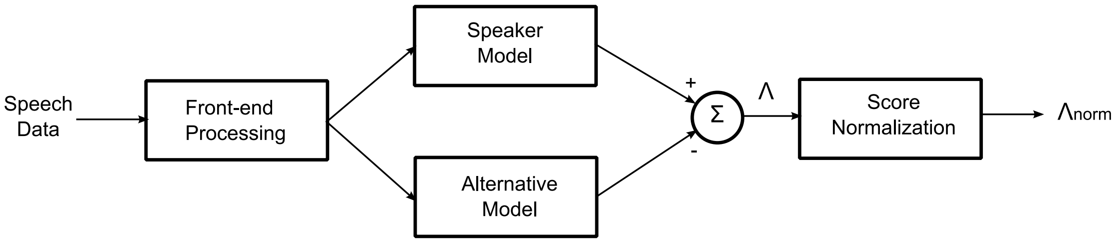

FIGURE 1.1: Likelihood ratio detection based speaker verification system.

Depending on the features used, speaker verification systems may be divided into two broad categories as: low-level systems and high-level systems. In low-level systems, features extracted from overlapping windows of 20-32 ms. of speech data are used. Common examples of these features are Mel Frequency Cepstral Coefficients (MFCC), Linear Frequency Cepstral Coefficients (LFCC), and Linear Predictive Cepstral Coefficients (LPCC). These features are typically augmented with their first and sometimes second derivatives. Silence frames are discarded using a voice activity detector and initial channel compensation is applied using one or more of the techiques like Cepstral Mean Normalization (CMN), Feature Warping [1], RASTA [2] and Feature Mapping [3]. These methods are based on the assumption that linear channel effects will shift the mean of cepstral coefficients and additive noise will modify their variance. In CMN, the mean of the data for each utterance is calculated and subtracted from all of the frames in the utterance. In Feature Warping, channel effects are eliminated by making the distribution of all the data the same. Since the channel distribution is unknown, Normal distribution is taken as the target distribution. The features are made short-time Gaussian by making the cumulative distribution of the cepstral coefficients approximately the same as that of the Normal distribution. In RASTA, a bandpass filter is used to remove spectral components that change more slowly or quickly than the typical range of change of speech. In Feature Mapping, the channel is modeled as a discrete variable and only finite number of possible channel values (like telephone,GSM,CDMA) are accepted. A training set with channels labeled, is needed for this purpose. First a general Gaussian Mixture Model (GMM) is trained using all of the data, and then the channel GMMs are obtained by MAP adaptation of this global GMM model using channel specific data. The global model and channel models are used to obtain transformation functions that map each channel data to global data. All of the frames in training and test are first mapped to the same channel (global GMM channel) and used afterwards. To find which transformation will be applied for mapping, top-1 scoring with channel GMMs are used and the channel giving the highest likelihood is accepted as the channel of the utterance.

High-level systems use features that represent long-span speaker characteristics. Tokens that represent speaking habits of speakers are extracted from speech utterances. Examples of high-level features are pitch and energy gestures [4], phone n-grams [5], phone binary trees [6], word n-grams and prosodic statistics. High level systems typically need more training data than low-level systems and depending on the token used may need a phone or speech recognizer working in parallel. Their classification accuracy is usually lower than that of low-level systems, but they fuse well with them at the score level [7]. In this thesis, we will only study on low-level speaker verification systems.

For speaker modeling both generative and discriminative approaches have been proposed. The most commonly used generative technique for speaker verification is GMM, while the most widely used discriminative technique is Support Vector Machines (SVM). In generative techniques, the alternative hypothesis may be represented with a set of speakers obtained for each subject (also known as cohort speakers) or with a single model obtained from data of a large speaker population. Although SVM does not fit into likelihood ratio detection based speaker verification systems, impostor utterances used in training may be seen as representatives of alternative hypothesis.

Methods compensating for intra-speaker variabilities that are due to effects like channel, spoken text, microphone used, speaking style, etc., are used at every level of speaker verification; front-end processing, modeling and scoring. Methods in the modeling stage typically works on very high dimensional *supervectors* obtained by concatenation of GMM mean vectors. Joint Factor Analysis (JFA) is one of the most successful generative technique that models both the inter-speaker and intra-speaker variabilities. Nuisance Attribute Projection (NAP) is used with Support Vector Machines (SVM) to remove the nuisance directions responsible for intra-speaker variabilities from supervectors. These methods all work on very high dimensional supervector space and typically assume that intra-speaker variabilities lie in a low dimensional subspace. These methods will be described in detail later in Chapter 2.

In scoring stage, after the likelihood ratio (or more generally score) is obtained, score normalization algorithms are applied to use a single threshold independent of nonspeaker factors for decision taking. The reason for this is based on two observations:

- The distribution of scores for a speaker changes with the channel (e.g. microphone used) and,
- Optimal thresholds for different speakers may differ.

Score normalization aims to compensate these variabilities by removing biases and scale factors estimated beforehand. Several approaches have been proposed for this objective, Zero normalization (Z-norm) and Test normalization (T-norm) being the most widely used ones [8]. Both Z-norm and T-norm, normalize the score $\Lambda(X)$ by:

$$\Lambda_{norm}(X) = \frac{\Lambda - \mu(X,S)}{\sigma(X,S)}, \tag{1.2}$$

where $\mu(X,S)$ is the mean and $\sigma(X,S)$ is the variance of impostor scores (scores coming from hypothesis H1) estimated for speaker S, while they differ in how these parameters are estimated. The normalization in Equation 1.2 shifts both impostor (hypothesis H1) and true (hypothesis H0) score distributions using mean and variance of impostor distribution. The impostor distribution of each speaker becomes normal with zero mean and unit variance, while true score distributions are shifted accordingly.

In Z-norm, $\mu(X, S)$ and $\sigma(X, S)$, are estimated at the enrollment stage by scoring the target speakers' model with a preselected database of impostor utterances. The log-likelihood scores are used to estimate speaker specific mean $\mu(X, S)$ and variance $\sigma(X, S)$ for the impostor (H1) distribution. The advantage of Z-norm is that the estimation of normalization parameters is performed off-line during training. In T-norm, they are estimated at scoring stage, by scoring the test utterance with a pretrained impostor models (T-norm models) in addition to claimed speaker. The mean $\mu(X, S)$ and variance $\sigma(X, S)$ of the log-likelihood scores of these impostor models for the test utterance are used as normalization parameters. The two methods may be used cascaded, ZT-norm (first Z-norm then T-norm) being the more used one. For the full success of these methods, Z-norm utterances must match the properties of test utterances while T-norm models must be close to the speaker model.

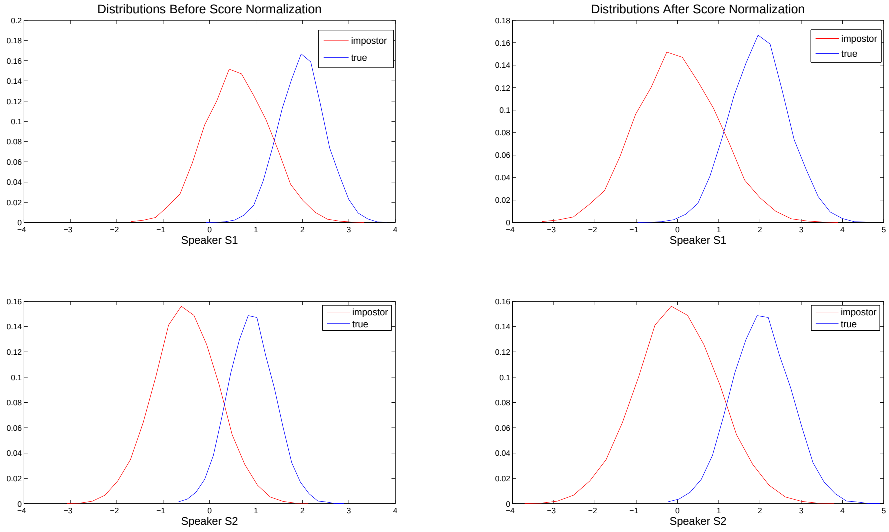

FIGURE 1.2: Impostor and true score distributions for two speakers before and after score normalization applied.

The effect of score normalization is shown for a hypothetical case in Figure 1.2. Here the impostor and true score distributions for two speakers, S1 and S2, are shown before and after score normalization. The plots in the first column show distributions before score normalization. Note that mean values of impostor and true distributions for the two speakers differ greatly, and it is impossible to select a single threshold giving certain false alarm and false reject rates for the two speakers. The plots in the second column show distributions after score normalization. Now the impostor distributions of both speakers become zero mean and unit variance, and true distributions shifted accordingly. In this case, it is possible to use a single threshold for both speakers.

1.2 Contributions of This Thesis

This work first presents a recipe to build a state of the art GMM supervector SVM system. Taking this system as a baseline, two methods as explained below are realized to improve performance.

- Two architectures to compensate inter-session variabilities separately at local regions in the acoustic space are proposed. Advantages of working at local subspaces and ways to incorporate existing methods in these architectures are discussed. A local version of GMM supervector SVM system with NAP channel compensation is realized.
- Ways to compensate within-session variabilities caused by across acoustic event variabilities and differences in actual realizations of acoustic events are investigated. A method using random sampling that targets mainly variabilities in actual realizations of acoustic events is proposed for GMM supervector SVM system. Depending on the size of short segments generated by random sampling, the method is also viable for across acoustic event variabilities. Performance improvements are shown both in short and long test utterances.

1.3 Outline of Thesis

This thesis consists of seven chapters. Chapter 1 defines the speaker verification problem and gives an overview of typical speaker verification systems. Chapter 2 describes the state of the art approaches to speaker verification. Chapter 3 describes the protocols and corpora used in NIST Speaker Recognition Evaluations (NIST SRE) which has a great impact on accelaration of the studies in this field. NIST SRE also provided the researchers with the necessary databases that includes most of the challenges of speaker verification in a controlled manner. Since we have participated in NIST SRE 2010 as part of the studies in this thesis, and we use mainly the NIST databases and protocols, it is important to have a brief knowledge of NIST SREs.

Chapter 4 describes the system we used for participating in NIST SRE 2010. This system also defines our baseline for the following chapters. The system description gives detailed recipes for building well performing GMM - UBM and GMM supervector SVM systems.

In Chapter 5, we propose our method for local implementations of intra-speaker variability compensation and speaker modeling. Chapter 6 describes our proposal of using random sampling for compensating within-session variabilities caused by differences in actual realizations of acoustic events and across phone variabilities. Finally, we conclude with Chapter 7.

Chapter 2

State of the Art Speaker Verification Methods

In this chapter, we give brief descriptions for state of the art low-level speaker verification methods. The section begins with Gaussian Mixture Model - Universal Background Model (GMM-UBM) method [9], which has been very popular after its proposal and become a starting point for more complicated systems. Joint Factor Analysis, which has shown great performance improvements over GMM-UBM method especially in case of significant session variabilities, is described next. Total Variability Analysis (TVS), which may be seen as the current best performing system, is a new improvement over JFA. Last, but not least, methods using Gaussian supervectors and Support Vector Machines are also mentioned briefly, including the channel compensation algorithms they utilize.

2.1 Gaussian Mixture Model - Universal Background Model Method

Gaussian Mixture Model is the most widely used generative approach for text-independent speaker verification. GMMs have the ability to approximate complicated probability density functions whose actual forms we do not know. For a D dimensional feature vector, $\mathbf{o}$, the GMM density is defined as:

$$p(\mathbf{o}|\lambda) = \sum_{i=1}^{M} w_i p_i(\mathbf{o}). \tag{2.1}$$

Each $p_i(\mathbf{o})$ is a Gaussian density, with $D \times 1$ dimensional mean vector μ_i and $D \times D$ dimensional covariance matrix Σ_i, given by

$$p_i(\mathbf{o}) = \frac{1}{(2\pi)^{D/2} |\Sigma_i|^{1/2}} \exp\left[-\frac{1}{2} (\mathbf{o} - \mu_i)^T \Sigma_i^{-1} (\mathbf{o} - \mu_i)\right]. \tag{2.2}$$

The mixture weights, w_i should sum up to one: $\sum_{i=1}^{M} w_i = 1$. So the parameters of the GMM are denoted by $\lambda = \{w_i, \mu_i, \Sigma_i\}$, where i= 1, ..., M. In general, diagonal covariance matrices are used in text-independent speaker verification. This avoids the need to invert $D \times D$ matrices, and a density represented by mixtures with full covariance matrices may equally well be represented by diagonal covariance mixtures if we increase the number of mixtures. Indeed, in [9], no significant performance gain is observed using full covariance Gaussian mixtures for text-independent speaker verification.

In [9], a likelihood-ratio detector based method using GMMs is proposed. In this method, a large GMM is trained from all the data of a pool of speakers. This GMM is called the Universal Background Model (UBM) or the world model. UBM is used for representing and calculating the likelihood of the alternative hypothesis. Separate UBMs may be trained for subpopulations. A classical example is having separate male and female UBMs.

UBM is also used to obtain reliable speaker models from small amount of training data. In [9], a new MAP adaptation algorithm called relevance MAP is proposed for this purpose. In the expectation step of relevance MAP, sufficient statistics for each mixture in the UBM are extracted from training data of the speaker. These sufficient statistics are the counts, first and second moments for each mixture. In speaker verification usually only the means of the UBM mixtures are adapted since adapting other parameters has not yielded any performance gain. In the second step of relevance MAP, these new sufficient statistics are combined with the previous statistics of the UBM using a data-dependent mixing coefficient. By using a data-dependent mixing-coefficient we try to update the mixtures with large amounts of speaker training data more, and the mixtures with few training data less. After adaptation, the mixtures who have no or few data will be nearly identical to the corresponding well-trained mixtures of the UBM. To achieve this goal, we first apply a soft alignment of speaker training vectors, $X = \{\mathbf{o}_1, \ldots, \mathbf{o}_T\}$, to the UBM mixture components. Let $p(i|\mathbf{o}_t)$ be the probability of the i^{th} mixture given $\mathbf{o}_t$:

$$p(i|\mathbf{o}_t) = \frac{w_i p_i(\mathbf{o}_t)}{\sum_{j=1}^{M} w_j p_j(\mathbf{o}_t)}. \tag{2.3}$$

The sufficient statistics for the mean parameter are the count, $n_i(X)$, and the first order moment $E_i(X)$. These are calculated as:

$$n_i(X) = \sum_{t=1}^{T} p(i|\mathbf{o}_t), \tag{2.4}$$

$$E_i(X) = \frac{1}{n_i(X)} \sum_{t=1}^{T} p(i|\mathbf{o}_t)\mathbf{o}_t. \tag{2.5}$$

These new statistics obtained from the training data are used to update the old UBM statistics for the mixture i to create the new mean vector, $\hat{\boldsymbol{\mu}}_i$,

$$\hat{\boldsymbol{\mu}}_i = \alpha_i E_i(\mathbf{o}) + (1 - \alpha_i)\,\boldsymbol{\mu}_i. \tag{2.6}$$

The data-dependent mixing coefficients α_i are given by:

$$\alpha_i = \frac{n_i(X)}{n_i(X) + r}. \tag{2.7}$$

Here r is a fixed quantity, called relevance factor, that controls the amount of adaptation from the UBM. The higher the relevance factor, the more training data is needed to adapt from the UBM.

Note that there is a tight coupling between the mixtures of the UBM and the mixtures of an adapted speaker model. This has an advantage in scoring since it makes a fast-scoring scheme called Top-N scoring possible. When a feature vector is scored against a large GMM, it is observed that only a few mixture components give high probability values, while the others contribute nearly nothing to the overall likelihood. Since the UBM and speaker mixtures are tightly coupled, it may be assumed that the same mixtures are active for a given frame in the UBM and the target speaker model. Using these two facts, in [9], a fast scoring method is proposed as:

- For each frame, first score only with the UBM and obtain the top performing N mixture components,
- Evaluate only with these N mixtures for the speaker and approximate the likelihood ratio by the top-N performing mixtures.

Using this method, for each frame we evaluate the likelihood of only M+N mixtures, where $N \ll M$, instead of $2 \times M$ mixtures. The effect is more dramatic if T-norm score normalization is also used.

2.2 Joint Factor Analysis

Despite the success of GMM - UBM approach in text-independent speaker verification, it has been observed that the performance degrades by mismatches in training and test data, known as inter-session variabilities. The source of these variabilities is commonly stated as channel effects (transmission environment, microphone used), but it is actually broader including variations due to phonetic content of the utterance and speakers' speaking style, emotional state, health, etc. These inter-session variabilities may also be called intra-speaker variabilities. During the last decade, studies in speaker verification focused on compensating these variabilities. The first attempts were at the feature and score levels, some examples of which are given in Chapter 1. Later, compensations at the modeling stage became more popular with the invention of Joint Factor Analysis by Patrick Kenny [10]. In [11] a model of session variability which is known as eigenchannel MAP is proposed. In [10, 12], eigenchannel MAP is integrated with standard models of inter-speaker variability, namely classical MAP [9] and eigenvoice MAP [13]. The resulting model of speaker and session variability is known as Joint Factor Analysis, outlined below:

Let C be the number of mixture components in the UBM and D be the dimension of the feature vectors. A $CD \times 1$ dimensional vector, known as supervector, is formed by concatenating D dimensional mean vectors of a GMM for each utterance. In JFA, it is assumed that, this speaker and channel dependent supervector $\mathbf{M}$, can be decomposed into a speaker supervector $\mathbf{s}$, and a channel supervector $\mathbf{c}$, where $\mathbf{s}$ and $\mathbf{c}$ are statistically independent and normally distributed. That is,

$$\mathbf{M} = \mathbf{s} + \mathbf{c}. \tag{2.8}$$

Furthermore the distribution of the speaker supervector is in the form:

$$\mathbf{s} = \mathbf{m} + \mathbf{vy} + \mathbf{dz}, \tag{2.9}$$

where $\mathbf{m}$ is a $CD \times 1$ speaker-independent mean vector, $\mathbf{v}$ is a $CD \times R_s$ rectangular matrix of low rank ($R_s \ll CD$), $\mathbf{y}$ is a normally distributed random vector, $\mathbf{d}$ is a $CD \times CD$ diagonal matrix and $\mathbf{z}$ is a normally distributed CD dimensional random vector. This is equivalent to saying that $\mathbf{s}$ is Gaussian distributed with mean $\mathbf{m}$ and covariance matrix $\mathbf{d^2} + \mathbf{vv}^*$. This is actually a combination of classical MAP where only $\mathbf{dz}$ component is used and eigenvoice MAP where only $\mathbf{vy}$ component is used. When large amount of training data exists, classical MAP seems to be the most appropriate choice. But it has the disadvantage that only the mixture components that have training data can be updated. Since $\mathbf{d}$ is diagonal it does not take the correlations between mixture components into account. In eigenvoice MAP we use a full but low-rank covariance matrix, and good adaptation can be achieved even with small amount of adaptation data. The disadvantage of eigenvoice MAP is that the speaker supervectors are assumed to lie in a linear manifold of low dimension known as the speaker space, and this space is spanned by the training speakers' supervectors. Even if the speakers' supervector stays elsewhere, eigenvoice MAP cannot estimate it no matter how much adaptation data is at hand. In JFA, these two components are used jointly to utilize the advantages of both methods. In first implementations of JFA, the speaker supervectors obtained are usually described by the $\mathbf{vy}$ component, and the $\mathbf{dz}$ component generally had no effect. In [12], the authors stated that this is an artifact of training procedure and proposed a recipe to decouple the estimation of $\mathbf{v}$ and $\mathbf{d}$ so that both terms will be beneficial.

The channel component of $\mathbf{M}$, $\mathbf{c}$, is assumed to be distributed according to:

$$\mathbf{c} = \mathbf{uf}, \tag{2.10}$$

where $\mathbf{u}$ is a $CD \times R_c$ dimensional rectangular matrix of low rank ($R_c \ll CD$), and $\mathbf{f}$ is a normally distributed random vector. That is $\mathbf{c}$ is normally distributed with zero mean and covariance matrix $\mathbf{uu}^*$. Furthermore, it is assumed that the speaker and channel subspaces do not overlap except at origin.

The JFA framework gives us the opportunity to obtain a speaker model immune to the channel effects in the training data, and to obtain the likelihood of a test utterance under a claimed speaker model without getting affected from the channel mismatches. During enrollment of a speaker, posterior distributions of $\mathbf{y}$ and $\mathbf{z}$ can be obtained. For training algorithms of JFA and their derivations, see the reference [10]. The likelihood of a test utterance X, can then be obtained by integrating over the posterior distributions of $\mathbf{y}$ and $\mathbf{z}$, and the prior distribution of $\mathbf{f}$, although MAP point estimates are usually used in practice. Scoring is done by computing the likelihood of the test utterance against session compensated speaker model ($\mathbf{M}-\mathbf{uf}$). For description and comparison of possible scoring methods in terms of performance and computational load, see reference [14].

2.3 Total Variability Space

In [15], Dehak et al. proposed an alternative to JFA where instead of the two separate speaker and channel spaces of JFA, only a single subspace is estimated. This space, called Total Variability Space (TVS), is both speaker and channel dependent. In TVS, each utterance is represented by a GMM supervector $\mathbf{M}$ given by;

$$\mathbf{M} = \mathbf{m} + \mathbf{Tw}, \tag{2.11}$$

where $\mathbf{m}$ is a $CD \times 1$ speaker and session independent mean vector, $\mathbf{T}$ is a $CD \times R$ rectangular matrix of low-rank ($R \ll CD$) and $\mathbf{w}$ is a normally distributed random vector. $\mathbf{M}$ is assumed to be normally distributed with mean $\mathbf{m}$ and covariance matrix $\mathbf{TT}^*$. Training procedure of $\mathbf{T}$ is the same as the training procedure of $\mathbf{v}$ in JFA, except in TVS each utterance is treated as coming from a different speaker. In this model factor analysis is used as a front-end to extract new features, $\mathbf{w}$ vectors, which are called total factors or identity vectors (i-vectors in short). Note that unlike JFA, TVS does not apply any inter-session variability compensation. Instead, compensations are applied after extraction of i-vectors. Since the total variability space is significantly smaller then the original supervector space, manipulations like modeling, compensation and scoring become more tractable and computationally efficient. In [16], three compensation algorithms are applied on i-vectors prior to scoring; linear discriminant analysis (LDA), nuisance attribute projection (NAP), and within class covariance normalization (WCCN). The best performance is achieved by sequential application of LDA and WCCN. Both SVMs and cosine distance scoring has been tried, cosine scoring giving the better performance.

To train the **T** matrix for the telephone case, enormous amount of data has been used in [16]. Such a large amount of data is not at hand for microphone case. In [17], Senoussaoui et al. proposed a method to obtain a **T** matrix that works well both for telephone and microphone data.

2.4 Support Vector Machine Based Methods

Support Vector Machine is a maximum-margin classifier that finds a hyperplane which separates two classes using sums of a kernel function $K(\cdot,\cdot)$;

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i t_i K(\mathbf{x}, \mathbf{x}_i) + d \tag{2.12}$$

where the t_i are the ideal outputs for support vectors $\mathbf{x}_i$, $\sum_{i=1}^{N} \alpha_i t_i = 0$, and $\alpha_i > 0$. The ideal outputs are -1 or 1 depending on the class of the corresponding support vectors. Support vectors are found by an optimization process which maximizes the margin, the minimal distance between the hyperplane seperating the two classes and the closest datapoints (support vectors) to the hyperplane. For classification, a decision is made depending on whether the value of $f(\mathbf{x})$ exceeds a threshold or not.

The kernel $K(\cdot,\cdot)$ is constrained to have certain properties known as Mercer conditions so that $K(\cdot,\cdot)$ can be expressed as:

$$K(\mathbf{x}, \mathbf{y}) = b(\mathbf{x})^T b(\mathbf{y}), \tag{2.13}$$

where $b(\mathbf{x})$ is a mapping from the input space to a possibly infinite dimensional SVM expansion space.

GMM supervector can be seen as a mapping between a speech utterance and a high dimensional space [18]. Note that speech utterances are in varying lengths, but we want to use fixed size vectors in kernel computation. Sequence kernels [19], solves this problem by first applying a mapping $b(\cdot)$ between an utterance and a high dimensional space and then applying the kernel at this new (fixed dimensional) space as in Equation (2.13). In case of GMM supervector representation, a GMM for each utterance is obtained by MAP adaptation of the UBM, and an approximation to KL divergence is applied as a distance metric since KL divergence itself does not satisfy Mercer conditions. The resulting kernel function for two mean only adapted GMMs for utterances X_a and X_b becomes:

$$K(X_a, X_b) = \sum_{i=1}^{M} w_i m_i^a \Sigma_i^{-1} m_i^b = \sum_{i=1}^{M} \left(\sqrt{w_i} \Sigma_i^{-\frac{1}{2}} m_i^a \right)^t \left(\sqrt{w_i} \Sigma_i^{-\frac{1}{2}} m_i^b \right), \qquad (2.14)$$

where Σ_i and w_i are the common covariance matrix and weight for the i^{th} mixture, and m_i^a, and m_i^b are the mean vectors of the i^{th} mixture for utterances X_a and X_b respectively. The kernel in Equation (2.14) is linear in the GMM supervector. The mapping $b(X)$ generating the supervector from the utterance for this kernel becomes:

$$b(X) = \begin{pmatrix} \sqrt{w_1} \Sigma_1^{-\frac{1}{2}} m_1 \\ \vdots \\ \sqrt{w_i} \Sigma_i^{-\frac{1}{2}} m_i \\ \vdots \\ \sqrt{w_M} \Sigma_M^{-\frac{1}{2}} m_M \end{pmatrix}. \qquad (2.15)$$

A useful aspect of this kernel is that we can use the model compaction technique from [19]. The SVM in Equation (2.12) can be summarized as:

$$f(\mathbf{x}) = \left(\sum_{i=1}^{N} \alpha_i t_i b(X_i) \right)^t b(X) + d = \mathbf{w^t} b(X) + d, \qquad (2.16)$$

where $\mathbf{w}$ is the quantity in the parantheses. This means that once we obtained the support vectors in training, we can compute $\mathbf{w}$ and get rid of the support vectors. Then we only have to compute a single inner product between the target model and the GMM supervector to obtain a score.

2.4.1 Nuisance Attribute Projection

The Nuisance Attribute Projection (NAP) method [20] works by removing subspaces that cause variability in the kernel. NAP constructs a new kernel,

$$K(X,Y) = [\mathbf{P}b(X)]^t\,[\mathbf{P}b(Y)] = b(X)^t\mathbf{P}b(Y) = b(X)^t\left(\mathbf{I} - \mathbf{v}\mathbf{v}^t\right)b(Y) \tag{2.17}$$

where $\mathbf{P}$ is a projection ($\mathbf{P}^2 = \mathbf{P}$), each column of $\mathbf{v}$ is a direction being removed from the SVM expansion space, and $b(\cdot)$ is the SVM expansion. The design criterion for $\mathbf{P}$ and correspondingly v is

$$\mathbf{v}^* = \underset{\mathbf{v}}{\operatorname{argmin}} \sum_{i,j} W_{i,j}\|\mathbf{P}b(X_i) - \mathbf{P}b(X_j)\|^2 \tag{2.18}$$

where X_i's are utterances in the training dataset. Typically we select $W_{i,j}$ as 1 if we want $b(X_i)$ and $b(X_j)$ to be closer in the expansion space, -1 if we want $b(X_i)$ and $b(X_j)$ to be distant in the expansion space and 0 otherwise. For example if session variability is the nuisance variable, then we can pick $W_{i,j} = 1$ if X_i and X_j belong to the same speaker, and $W_{i,j} = 0$ otherwise. In [18], it is shown that in this case NAP produces the same subspace as Factor Analysis.

2.4.2 Within Class Covariance Normalization

In [21], Hatch and Stolcke examined kernel selection for tasks involving one-versus-all classification problems. They worked on *generalized linear kernels* in the form:

$$K(\mathbf{x},\mathbf{y}) = \mathbf{x}^T\mathbf{R}\mathbf{y}, \tag{2.19}$$

where $\mathbf{x}$ and $\mathbf{y}$ are vectors in the input space, and $\mathbf{R}$ is a positive definite matrix. They first constructed a set of upper bounds on the rates of false positives and false negatives at a given score threshold. They showed that minimizing these bound leads to the closed form solution, $\mathbf{R} = \mathbf{W}^{-1}$, where $\mathbf{W}$ is the expected within-class covariance matrix of the data given by:

$$\mathbf{W} = \frac{1}{S}\sum_{i=1}^{S}\frac{1}{n_i}\sum_{j=1}^{n_i}\left(b(X_j^i) - \mu_i\right)\left(b(X_j^i) - \mu_i\right)^T. \tag{2.20}$$

Here S is the number of training speakers, n_i is the number of utterances for the i^{th} speaker, μ_i is the mean supervector for the i^{th} speaker, and $b(X_j^i)$ is the supervector obtained from the j^{th} utterance of the i^{th} speaker. In order to keep inner product nature of the kernel $K(\cdot,\cdot)$, a mapping function $\mathbf{H}(\cdot)$ can be defined as:

$$\mathbf{H}(b(X)) = \mathbf{A}^t b(X), \tag{2.21}$$

where $\mathbf{A}$ is obtained using a Cholesky decomposition of the matrix $\mathbf{R}$. While NAP achieves channel and session compensation through removing nuisance directions, WCCN optimally weights each of these directions to minimize a particular upper bound on error rate.

2.5 Evaluation Metrics for Speaker Verification

There are two types of trials in speaker verification; target trials where the claimed speaker is the actual speaker, and impostor trials where the claimed speaker is not speaking in the test utterance. A speaker verification system gives two outputs: a decision (either true or false) and a likelihood showing the systems level of confidence in the decision. Two types of errors may occur in this scenario; false rejects and false alarms. The false rejection rate ($P_{FR|target}$) is the percent of target trials labeled as impostor incorrectly. The false alarm rate is ($P_{FA|impostor}$) the percent of impostor trials labeled as target trials incorrectly. Several evaluation metrics have been proposed that represent these two trials. Here, we will describe those that are widely used in speaker verification community, namely Detection Cost Function (DCF), Equal Error Rate (EER), and Decision Error Tradeoff (DET) Curves.

DCF is the basic performance measure used in speaker recognition evaluations coordinated by National Institute of Standards and Technology (NIST).The C_{Det} cost is a weighted sum of the two error rates:

$$C_{Det} = C_{FR} * P_{FR|target} * P_{target} + C_{FA} * P_{FA|impostor} * P_{impostor}, \tag{2.22}$$

where C_{FR} is the cost of a false rejection, P_{target} is the prior probability of a target trial, C_{FA} is the cost of a false alarm, and $P_{impostor}$ is the prior probability of an impostor trial. This cost function is made more intuitive by normalizing it so that a system with no discriminative capability is assigned a cost of 1.0.

EER is the false reject (at the same time false alarm) rate at the operating point where the two error rates are equal. While EER and DCF being single number measures, DET plot is a graph plotting error rates for all the operating points of a system. An individual operating point corresonds to a threshold used to take decisions for trials as either true or false. By sweeping over all possible thresholds and calculating the two error rates, all of the operating points of a system are generated. DET plot is a variant of receiver operating characteristic (ROC) curve used by NIST, where the two axes are the two error rates.

Chapter 3

NIST Speaker Recognition Evaluations

The speech group at the National Institute of Standards and Technology (NIST) has been coordinating evaluations of text-independent speaker recognition technology since 1996 [22]. By providing explicit evaluation plans, common test sets, standard measurements of error, and platforms for participants to openly discuss algorithm success and failures, the NIST series of Speaker Recognition Evaluations (NIST SRE's) has provided a means for accelerating and recording the progress of text independent speaker recognition performance. The evaluations were conducted annually up to 2006 and every two years after then. The main task at NIST SRE's is speaker detection, although other tasks like speaker segmentation and tracking has also been investigated from period to period. The speaker detection task is defined as determining whether a specified speaker is speaking during a given segment of conversational speech. NIST evaluations focused primarily on conversational telephone speech but recent evaluations also considered cross channel data, where a telephone conversation is simultaneously recorded using several sensors of varying types, and interview data where face to face conversations with an interviewer have been recorded in a special room with several types of microphones.

There are certain evaluation rules that are applicable to nearly all recent NIST SRE's. Some of the important rules are listed below:

- Each decision is to be based only upon the specified test segment and target speaker model. Use of information about other test segments and/or other target speakers is not allowed.

- Knowledge of the telephone transmission channel type and of the telephone instrument type used in all segments is not allowed, except as determined by automatic means.

- Listening to the evaluation data, or any other experimental interaction with the data, is **not** allowed before all test results have been submitted.

- Knowledge of language used in all segments and of the gender of the target speaker **is** allowed. There are no cross-gender trials.

Various factors affecting performance of speaker verification has been explored during NIST evaluations. These may include gender, language used, microphone type (electret vs. carbon button), channel type (landline, cellular, different handsets), duration of training and testing utterances, speaking style and vocal effort of the speaker. Up to 2004, effect of duration has been investigated using two task conditions; limited data and extended data conditions. Limited data meant that the training and test segment data for each trial consisted of two minutes or less of concatenated segments of speech data, with silence intervals removed, while extended data meant that each of these consisted of an entire conversation side, or for training, multiple conversation sides. At NIST SRE 2004 and the following NIST SRE's, this distinction has been removed and instead, multiple testing conditions have been offered involving the amount and type of data available for both the training and the test segments. NIST selects one of the conditions as core condition and expects all the participants to complete this task while completing other tasks is optional. Sites participating in one or more of the speaker detection tests must report results for each test in its entirety. For each trial the decision (as true or false) and a likelihood score must be provided. NIST uses a detection cost function (DCF) for performance measurement. Decision Error Tradeoff (DET) curves are also produced. A log likelihood ratio based cost function is also performed on submissions whose scores are declared to represent log likelihood ratios.

In the next sessions of this chapter, the datasets used in NIST SRE's and the training and test conditions in recent NIST SRE's will be introduced. For further information see the relevant years NIST SRE Evaluation Plans [22].

3.1 Corpora used in NIST SRE's

The first NIST SRE's used Switchboard databases for speaker verification [23]. The challenges presented by this data include limited bandwidth, channel noise from various sources, the use of different microphones, recordings from different locations, and recordings collected over a period of time [24]. In several Switchboard phases, subjects were asked to complete calls within a variety of environments including quiet offices, public places and moving vehicles.

Mixer Corpus has been used for NIST SRE's since 2004. This corpus adds two dimensions to the traditional Switchboard collection: language and channels. Mixer is a corpus of multilingual and cross-channel speech data. Like previous speaker recognition corpora, the calls feature a multitude of speakers conversing on different topics and using a variety of handsets types. Unlike previous studies, a large subset of the subjects were bilingual. Further distinguishing Mixer from previous studies, some calls have also been recorded simultaneously via a multichannel recorder using a variety of microphones. There has been 5 phases up to now for Mixer corpus collection. Mixer studies include a number of separate tasks. All studies require the core collection of a small number (usually 10) of short calls (approximately 6 minutes) from a large number of subjects. In unique handset task, subjects are asked to make four calls from handsets that they use exactly once in the study. Once a handset reappears in the study, it is no longer considered unique. Extended Data task refers to collection of 20 or more calls per subject. In Transcript Reading task, subjects read samples from transcripts of their calls and calls from other subjects. Mixer 5 focuses on cross-channel recordings of face to face interviews where the goal is to elicit speech within a variety of situations. Table 3.1, taken from reference [25], summarizes the tasks completed on Mixer phases.

In Mixer 1 a large subset of the subjects were bilingual and conducted their conversations in Arabic, Mandarin, Russian and Spanish as well as in English. In both Mixer 1 and 2 subjects were allowed to initiate calls that were simultaneously recorded via

Tasks	SB	M1	M2	M3	M4	M5
Core Calls(8+)	x	x		x	x	x
Variable Environments	x					
Unique Handset(4+)	x	x	x	x	x	x
Extended Data(20+)		x	x	x	x	
Multilingual(4+)		x		x	x	
Cross Channel(4+)		x	x		x	
Transcript Reading(2+)		x				x
Interviews(6)						x

Table 3.1: Tasks within Switchboard and Mixer collection efforts

eight different microphones selected and placed to represent a variety of microphone and channel conditions. The multichannel sensors were side-address studio microphone, podium microphone, hanging microphone, PZM microphone, dictaphone, computer microphone, and two cellular phone headsets [26, 27]. Mixer 3 addresses needs for both language recognition and speaker recognition [25]. 3918 subjects completed 19951 calls. More than 2900 Mixer 3 subjects each made a call in one of 19 languages including Bengali, 4 dialects of Chinese, 3 dialects of English, Farsi, Hindi, Italian, Japanese, Korean, Russian, Spanish, Tagalog, Thai, Urdu, and Vietnamese. Mixer 4 consists of core telephone and cross channel data [25, 28]. All subjects are required to be native speakers of American English but there are dialect differences. The 8 microphone configuration built for Mixer 1 and 2 has been replaced with a system that can handle 16 channels though only 14 are used in Mixer studies.

Mixer 5 focused on cross-channel recording of face to face interview data. Each subject participates in 6 thirty minute interview sessions spread over at least three days, with at least 30 minutes rest between sessions that occur on the same day. In order to elicit multiple repetitions of a small amount of speech in which the same words appear, each of the six sessions begins with the subject answering the same questions. In the first session a warm-up part follows this with the kind of conversation characteristics of first meetings since the subject meets the interviewer the first time and enters to an unknown environment. The next part includes personal and family history of the subject. Informal conversations make up a large portion of the study and spans all of the interview sessions. The interviewer engages the subject in informal conversation exploring a variety of topics in search of those that the subject shows interest. In transcript reading the subject, using a natural speaking voice and style, reads individual utterances taken from transcripts of phone conversations collected in earlier studies at LDC. In story reading the subject reads stories containing phonetically balanced text. For sentence reading, the subject reads a subset of the TIMIT sentences in a natural reading voice and style. In phrase/word list reading, the subject reads from lists that are designed to produce speech that in which the vernacular is most easily heard. In the low vocal effort call the subject participates in a brief 5 minute telephone call characterized by low vocal effort as a natural result of a loud and clear telephone circuit in which the subject's voice is feed back through the headset. Since the subject hears his own voice through the headphone he/she automatically reduces his effort. In the high vocal effort call the subject participates in a brief telephone call where the subject's side tone and the remote caller's voice are weak and noisy. The addition of the noise causes the participant to increase his vocal effort. Table 3.2, taken from [28] shows the breakdown of speech act in Mixer 5. For further information on Mixer studies, please refer to references [25–28].

Session Number	1	2	3	4	5	6	Min
Repeating Questions	1	1	1	1	1	1	6
Warm-up	4						4
Family Personal	5						5
Informal Conversation	20	9	14	9	9	10	71
Transcript Reading		20	15	10	15	10	70
Story Reading				5			5
Sentence Reading					5		5
Phase/Word List Reading						5	5
Low Vocal Effort				5			5
High Vocal Effort						4	4
Total/Session	30	30	30	30	30	30	180

Table 3.2: Breakdown of Minutes/Speech Act/Session

3.2 Training and Test Conditions in Recent NIST SRE's

Each NIST SRE has several task conditions depending on the type and duration of the data used in training and testing. Since we use the recent NIST SRE's databases and task conditions throughout this thesis, these conditions are briefly described below.

3.2.1 NIST SRE 2006 Task Conditions

The five training conditions in NIST SRE 2006 are:

- A two-channel (4-wire) excerpt from a conversation estimated to contain approximately 10 seconds of speech of the target on its designated side.
- One two-channel (4-wire) conversation, of approximately five minutes total duration, with the target speaker channel designated.
- Three two-channel (4-wire) conversations involving the target speaker on their designated sides.
- Eight two-channel (4-wire) conversations involving the target speaker on their designated sides.
- Three summed-channel (2-wire) conversations, formed by sample-by-sample summing of their two sides.

The four test segment conditions are the following:

- A two-channel (4-wire) excerpt from a conversation estimated to contain approximately 10 seconds of speech of the putative target speaker on its designated side
- A two-channel (4-wire) conversation, of approximately five minutes total duration, with the putative target speaker channel designated.
- A summed-channel (2-wire) conversation formed by sample-by-sample summing of their two sides.
- A two-channel (4-wire) conversation, with the usual telephone speech replaced by auxiliary microphone data in the putative target speaker channel. This auxiliary microphone data is supplied on 8 kHz 8 bit μ-law form.

3.2.2 NIST SRE 2008 Task Conditions

The six training conditions in NIST SRE 2008 are:

- **10sec:** A two-channel excerpt from a telephone conversation estimated to contain approximately 10 seconds of speech of the target on its designated side.
- **short2:** One two-channel telephone conversational excerpt, of approximately five minutes total duration, with the target speaker channel designated *or* a microphone recorded conversational segment of approximately three minutes total duration involving the target speaker and an interviewer.
- **3conv:** Three two-channel telephone conversational excerpts involving the target speaker on their designated sides.
- **8conv:** Eight two-channel telephone conversation excerpts involving the target speaker on their designated sides.
- **long:** A single channel microphone recorded conversational segment of eight minutes or longer duration involving the target speaker and an interviewer.
- **3summed:** Three summed-channel telephone conversational excerpts, formed by sample-by-sample summing of their two sides.

The four test segment conditions are the following:

- **10sec:** A two-channel excerpt from a telephone conversation estimated to contain approximately 10 seconds of speech of the putative target speaker on its designated side.
- **short3:** A two-channel telephone conversational excerpt, of approximately five minutes total duration, with the putative target speaker channel designated *or* a similar such telephone conversation but with the putative target channel being a (simultaneously recorded) microphone channel *or* a microphone recorded conversational segment of approximately three minutes total duration involving the putative target speaker and an interviewer.
- **long:** A single channel microphone recorded conversational segment of eight minutes or longer duration involving the putative target speaker and an interviewer.
- **summed:** A summed-channel telephone conversation formed by sample-by-sample summing of its two sides.

3.2.3 NIST SRE 2010 Task Conditions

The four training conditions in NIST SRE 2010 are:

- **10sec:** A two-channel excerpt from a telephone conversation estimated to contain approximately 10 seconds of speech of the target on its designated side.
- **core:** One two-channel telephone conversational excerpt, of approximately five minutes total duration, with the target speaker channel designated *or* a microphone recorded conversational segment of three to fifteen minutes total duration involving the interviewee (target speaker) and an interviewer. In the former case the designated channel may either be a telephone channel or a room microphone channel; the other channel will always be a telephone one.
- **8conv:** Eight two-channel telephone conversation excerpts involving the target speaker on their designated sides.
- **8summed:** Eight summed-channel excerpts from telephone conversations of approximately five minutes total duration formed by sample-by-sample summing of their two sides.

The three test segment conditions are the following:

- **10sec:** A two-channel excerpt from a telephone conversation estimated to contain approximately 10 seconds of speech of the putative target speaker on its designated side.
- **core:** One two-channel telephone conversational excerpt, of approximately five minutes total duration, with the putative target speaker channel designated *or* a microphone recorded conversational segment of three to fifteen minutes total duration involving the interviewee (target speaker) and an interviewer. In the former case the designated channel may either be a telephone channel or a room microphone channel; the other channel will always be a telephone one.
- **summed:** A summed-channel telephone conversation of approximately five minutes total duration formed by sample-by-sample summing of its two sides.

In each evaluation, in addition to these task conditions, NIST has specified one or more common evaluation conditions, subsets of trials in the core test that satisfy additional constraints, in order to better foster technical interactions and technology comparisons among participating sites. The performance results on these trial subsets are treated as the basic official evaluation outcomes. Because of the multiple types of training and test conditions in the 2010 core test, and the likely disparity in the numbers of trials of different types, it is not appropriate to simply pool all trials as a primary indicator of overall performance. Instead the below common conditions have been considered as primary performance indicators by NIST:

1. All trials involving interview speech from the same microphone in training and test

2. All trials involving interview speech from different microphones in training and test

3. All trials involving interview training speech and normal vocal effort conversational telephone test speech

4. All trials involving interview training speech and normal vocal effort conversational telephone test speech recorded over a room microphone channel

5. All different number trials involving normal vocal effort conversational telephone speech in training and test

6. All telephone channel trials involving normal vocal effort conversational telephone speech in training and high vocal effort conversational telephone speech in test

7. All room microphone channel trials involving normal vocal effort conversational telephone speech in training and high vocal effort conversational telephone speech in test

8. All telephone channel trials involving normal vocal effort conversational telephone speech in training and low vocal effort conversational telephone speech in test

9. All room microphone channel trials involving normal vocal effort conversational telephone speech in training and low vocal effort conversational telephone speech in test

Throughout this thesis, when the test is performed on NIST SRE 2010 dataset, the performance will be measured separately for each of these common evaluation conditions.

Chapter 4

TUBITAK UEKAE - SABANCI University System for NIST SRE 2010

As a part of our study for this thesis, we have participated in NIST SRE 2010 with the GMM Supervector SVM baseline system we have developed. This was our first participation in NIST SRE's and it required a lot of effort for us to complete the necessary data preparation, text processing, algorithm implementation and server utilization tasks. This chapter describes the submitted system and it also describes the baseline system for the next chapters. We begin by describing the database utilization. The front-end including our voice activity detector comes next. Universal background model (UBM) is a key component nearly for all the low-level feature based systems. It is a large GMM trained over a huge database and this necessitates a carefully designed algorithm for Expectation Maximization based training. We will give the recipe we use for such a training algorithm. The next part describes our supervector extraction and SVM training phases. Finally the results in terms of DET plots are given for each of the 9 common evaluation conditions of NIST SRE 2010 and our achievements will be listed.

4.1 Database Organization

NIST SRE06 and SRE08 databases are used to build our system. From SRE06, 1 conversation 2 channel (4-wire) and 1 conversation auxiliary microphone data are used. From SRE08, data of short2, short3 and long conditions are used. Since NIST also wants a decision for each trial we need to select a threshold from a development set. For this purpose we split the SRE08 database into two parts. The first part used in training and the second part used in testing. The training part of SRE08 includes 300 unique male and 500 unique female speakers. The test part includes 182 unique male and 341 unique female speakers. We have used approximately 323 hours of speech for male, and 463 hours of speech for female UBMs. 1095 utterances used as impostors for male speakers whereas 1695 utterances used for female speakers. In Z-norm score normalization 1258 utterances used for male speakers and 1832 utterances used for female speakers. T-norm score normalization is not applied due to time constraints. In Table 4.1 the breakdown of speech data for world model generation is given. Table 4.2 and 4.3 shows breakdown for impostor and Z-norm utterances, respectively.

Speech Type	Male	Female
SRE06 phone	2538	3164
SRE06 microphone	1256	1424
SRE08 phone	2600	4700
SRE08 microphone	396	504
SRE08 interview	975	1375
Total	7765	11167

Table 4.1: Breakdown of Speech Data for World Model Training

Speech Type	Male	Female
phone	652	1116
microphone	213	288
interview	230	291
Total	1095	1695

Table 4.2: Breakdown of impostor utterances

Speech Type	Male	Female
phone	663	1130
microphone	238	262
interview	375	440
Total	1258	1832

Table 4.3: Breakdown of Z-norm utterances

4.2 Front-End

MFCC feature vectors are used for acoustic vector representation. 19 dimensional static vectors are extracted for every frame of 20 ms duration with 10 ms overlap using only the 300-3400 Hz bandwidth. Delta and delta-energy components are appended producing a 39 dimensional feature vector for each frame. Feature warping with a sliding window of 3 seconds is applied to these MFCC features. Voice activity detection algorithm explained below has been used to remove non-speech frames.

4.2.1 Voice Activity Detection

Our voice activity detector (VAD) uses a bi-Gaussian model trained for each utterance using log-energies of the frames. The log energy for a frame is calculated as :

$$E(i) = \log(1 + \sum_{j=1}^{K} R_i(j)). \tag{4.1}$$

where $R_i(j)$ is the j^{th} component of the magnitude of the Fourier transform of the i^{th} frame. A bi-Gaussian model is trained using the log-energies of all frames. The frames with a log-energy greater than a threshold are accepted as containing speech. The threshold is calculated using the parameters of the Gaussian with lower mean and is equal to:

$$th = \mu + 1.5 * \sigma. \tag{4.2}$$

where μ is the mean of the lower energy Gaussian and σ is its variance. The resulting labels are postprocessed to remove speech and silence segments whose lengths are less than 180 ms.

For the interview data in NIST SRE08 database, NIST provides the estimated intervals where the target speaker is speaking, as determined by an energy based segmenter utilizing the audio signals from lavalier microphones worn by each of the two speakers. We take the portion of the utterance that is labeled as belonging to the target speaker by NIST and at the same time labeled as speech by our VAD. For the interview segments in NIST SRE10, the provision of the interviewer's head mounted close-talking microphone signal in a time aligned second channel, with speech spectrum noise added to mask any residual speech of the interviewee, is provided by NIST. So we processed both channels using our VAD and labeled the frames that contain speech in the first channel and nonspeech in the second channel as speech belonging to the target speaker.

4.3 Universal Background Modeling Recipe

Universal Background Model (UBM) is one of the key components of most of the successful speaker verification methods. In GMM-UBM framework, it is both used to obtain a well trained model from small amounts of target speaker data and to model the alternative speaker hypothesis. In JFA, it is used to extract necessary statistics from the utterance. UBM is generally trained from large amount of data collected from corporas like Switchboard and NIST SREs. In [9], it is mentioned that same performance is achieved from a UBM trained from 6 hours of speech and a UBM trained with 1 hour of speech from the same speaker set of the first UBM. It seems that the number of distinct speakers is a more important aspect than the amount of data. The data for UBM training must be representative of the target speaker population, the channel and microphone types that the system may encounter. Since UBM is a huge GMM trained from a very large database (typically several hundreds of hours of speech data), the EM training algorithm must be carefully implemented possibly using some tricks to enforce the model to better generalize the data. Below is our recipe for UBM training. This recipe is followed to obtain gender-dependent UBMs for NIST SRE10 evaluation as well as for all other experiments in this thesis.

- The UBM is initialized by a single Gaussian with its mean equal to the mean of the overall data and its variance equal to the variance of the overall data.

- At each following step the number of mixtures is incremented by a factor of two and a maximum of 25 EM iterations are applied to obtain a GMM with the current number of mixtures. Variance flooring is applied to guarantee that variance values do not go beyond some percent of the overall variance of the data.

- Random sampling is used at each EM iteration to select 2 percent of each of the utterance. So at each EM iteration only 2 percent of the data is used and different portions of the data is seen by the algorithm at each iteration. Note that since different data is used at each EM iteration, it is no more guaranteed that the log-likelihood will increase at each iteration and the algorithm may terminate before it reaches the maximum number of iterations which is 25. This random sampling procedure greatly speeds up the training procedure which is a very time consuming task. It may also help to obtain a better generalizing model.

- To increment the number of mixtures by a factor of two, we first evaluate a condition number for each mixture. This condition number is a function of the variance of the mixture and, together with the mixture weights, is used to determine the mixtures suitable to splitting. The condition number for mixture i is given by:

$$c_i = D \log (2\pi) + \log |\Sigma_i| \tag{4.3}$$

where D is the dimension of the feature vectors and Σ_i is the diagonal covariance matrix of mixture i. The mean μ_c and standart deviation σ_c of the condition numbers are evaluated and a threshold is evaluated as $th_c = \mu_c - 3 * \sigma_c$. The mixtures whose condition numbers are less than this threshold or whose weights are less than a predefined threshold are labeled as unsuitable for splitting and preserved in the new GMM. The remaining mixtures are split into two until a new GMM with the desired number of mixtures is obtained. When a mixture is split into two, the new mixtures have the same variance vector as their parent, but the mean vector is 1.2 times the parents' mean vector for one child mixture and 0.8 times the parents' mean for the other mixture. Their weights are equal and half of their parents' weight. For each mixture we keep the number of previous splits which is 0 for all mixtures at the beginning. After a splitting, the child mixtures' splitting number will be 1 more than that of the parents'. The next mixture to be splitted is the one for which

$$\log(w) - n_s \tag{4.4}$$

is maximum where w is the weight of the mixture and n_s is the number of previous splits. The n_s term will make previously splitted mixtures less probable to be splitted again.

Two gender-dependent UBMs, each with 2048 mixtures, are trained using the above recipe for NIST SRE10 evaluations. 323 hours of speech for male UBM,and 463 hours of speech for female UBM is used from NIST SRE06 and NIST SRE08 databases.

4.4 GMM Supervector - SVM Training

Target speaker GMM models are obtained using mean-only relevance MAP algorithm with a relevance factor of 8. To obtain the GMM supervector, for each mixture we evaluate

$$\sqrt{w_i}\Sigma_i^{-1/2}(\mu_i^s - \mu_i^{UBM}). \tag{4.5}$$

and concatenate these into a single GMM Supervector (GSV). This GSV is then normalized to be unit norm. For the core condition of NIST SRE10 we have a single training utterance for each target. 1095 male and 1695 female utterances from NIST SRE06 and NIST SRE08 databases are used as impostor utterances for SVM training. Z-norm score normalization used with 1258 male and 1832 female utterances.

4.5 Results

We will give DET plots of our system for each of the common evaluation conditions. In Figure 4.1 DET plots for condition 1 (interview-same microphones in train and test) and condition 2 (interview- different microphones in train and test) is shown. It seem that there is a large performance degradation when we use different microphones in interview data. This is something we have expected due to two reasons: first; our system does not utilize any channel compensation method except feature warping and secondly we have few interview data in Z-norm and impostor utterances. Besides we do not compute separate Z-norm parameters for each data type (phone , microphone and interview) which most likely will improve performance.

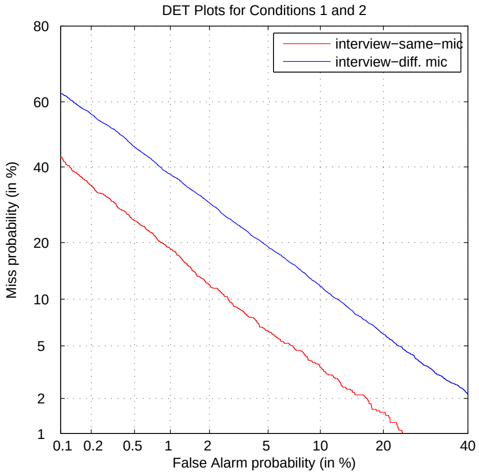

Figure 4.1: DET Plots for Condition 1 and 2.

Figure 4.2 shows DET plots for condition 3 (interview-normal vocal effort telephone) and 4 (interview - normal vocal effort microphone). We have performance degradation when simultaneously recorded room microphone data is used in testing. Due to the same reasons mentioned above this is something expected.

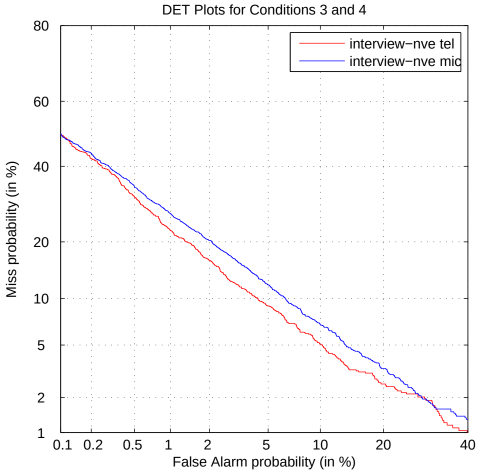

Figure 4.2: DET Plots for Condition 3 and 4.

Figure 4.3 shows DET plot for condition 5 (normal vocal effort - normal vocal effort different telephone). This case was the main target of NIST evaluations for years. It seem that our system does pretty well in this condition even though it does not utilize channel compensation.

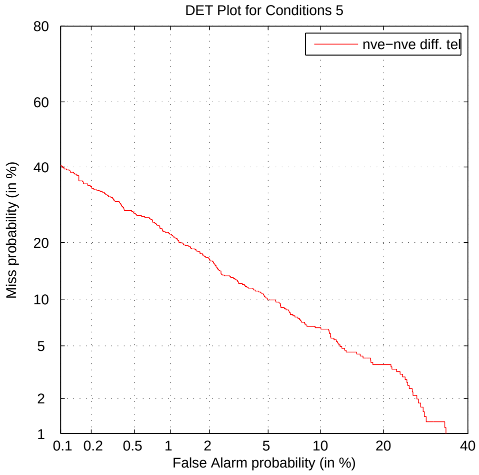

Figure 4.3: DET Plots for Condition 5.

Figure 4.4 shows DET plots for condition 6 (normal vocal effort - high vocal effort telephone) and 7 (normal vocal effort - high vocal effort microphone). Comparing this with 4.3 shows that mismatch in vocal effort degrades performance. Note that low and high vocal effort data is unique to NIST SRE10 database and our system has never utilized this kind of data.

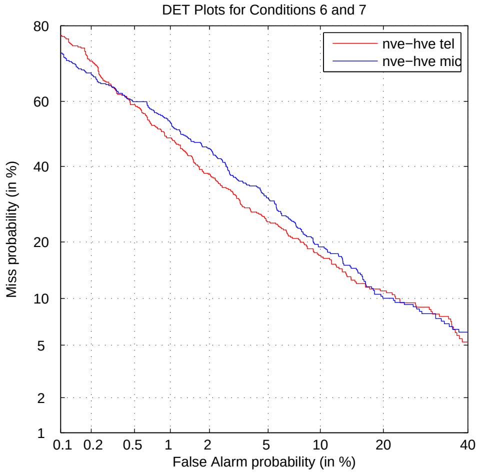

Figure 4.4: DET Plots for Condition 6 and 7.

Figure 4.5 shows DET plots for condition 8 (normal vocal effort - low vocal effort telephone) and 9 (normal vocal effort - low vocal effort microphone). Comparing this with 4.3 shows that actually our system has better performance with mismatch in vocal effort in the form of normal versus low case. This was the case for nearly all of the systems submitted to NIST SRE10 and it was quite surprising. Some debate on this surprising result has been made in NIST SRE workshop but the reason is not still clear.

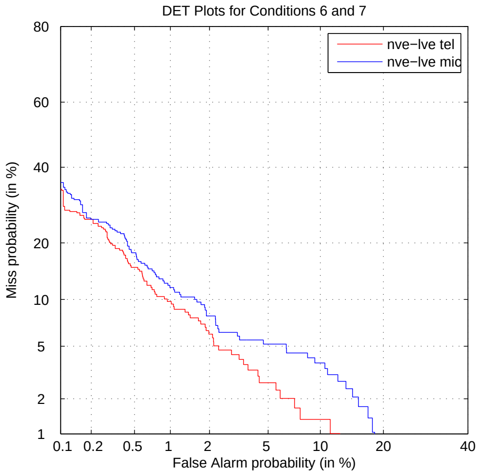

FIGURE 4.5: DET Plots for Condition 8 and 9.

Table 4.4 gives EER values for our system and average EER values for systems showing best performances in NIST SRE 2010. Note that these systems usually employ fusion of several systems, typically implemented by several cooperating sides, at the score level. The worst performing systems obtained EER values in the range 20-40 nearly for all conditions.

	C1	C2	C3	C4	C5	C6	C7	C8	C9
Our System	5.67	10.94	6.98	8.03	7.63	13.57	14.76	3.69	5.17
Best Systems	1.2-3	2-4	1.5-2	2-4	2-4	2-4	3-6	1	1-3

TABLE 4.4: EER values for our system and average EER values of best performing systems.

4.6 Conclusion

NIST SRE is the most compelling and the highest-ranking evaluation for speaker verification. Our participation in this evaluation gave us the opportunity to work on databases that contain most of the practical problems of speaker verification. We developed a baseline system for this thesis and during the evaluation timeline obtained great expertise in training GMM and SVM based speaker verification systems as well as working with large databases. It became apparent that we should spend more effort on channel compensation algorithms such as Nuisance Attribute Projection and Within Class

Covariance Normalization. It also showed us that choosing a right threshold for a system is a very important and a very difficult task. Mismatches in development and test databases can result in uncalibrated systems. Calibrating our classifiers is at least as important as developing better classifiers.

Chapter 5

Local Representations for Channel Robust Speaker Verification

The main speaker verification methods combatting with channel effects have two main assumptions: speaker and channel spaces are low dimensional, and these spaces do not overlap. These assumptions are more practical than theoretical. Joint Factor Analysis (JFA), Nuisance Attribute Projection (NAP), and Within-Class Covariance Normalization (WCCN), all work on very high dimensional supervectors produced by concatenation of Gaussian mixture means. They need an eigendecomposition in this high dimensional space and the maximum possible rank is determined by the number of individual utterances which is much smaller than the dimension of the supervectors. As noted by Patrick Kenny in [29], there is no theoretical proof for low dimensionality of channel and speaker spaces, instead we are forced to use these assumptions since sufficiently high rank covariance matrices are impossible to estimate and calculate with. Working in such a high dimensional space has also the drawback that the compensations and modeling become less tractable. Some methods like fully Bayesian ones and nonlinear manifold learning algorithms can not be applied. Due to the problems working in such a high dimensional space, Total Variability Space method [15], uses JFA as a front-end to reduce dimensionality and the supervectors are projected to a speaker and channel dependent low-dimensional space. Channel compensation, modeling and scoring methods are then used in this low-dimensional space.

Most of the main speaker verification algorithms can also be seen from a geometric point of view. GMM Supervector - SVM NAP method uses kernel values which may be seen as distances between speaker supervectors to learn discriminating hyperplanes. JFA algorithm, though usually implemented in a probabilistic sense, tries to learn eigenspeakers and eigenchannels in the supervector space. The speaker and channel factors may be seen as projections to these eigenvectors. Another method called Anchor Modelling describes each target speaker with a Speaker Characterization Vector (SCV) which is a vector of distances between the target speaker and a set of pre-selected speakers named anchor models. This method is also based on calculating distances between speaker models. In all of these methods we calculate a single value for the distance between speaker models. This is questionable because distance between two speakers in different portions of the high dimensional space we are working on may be significantly different. As an example take two hypothetical speakers using different dialects of the same language. The supervectors of these speakers will be closer to each other on the phonemes common to the two dialects, whereas they will be more apart for the phonemes the dialects differ.

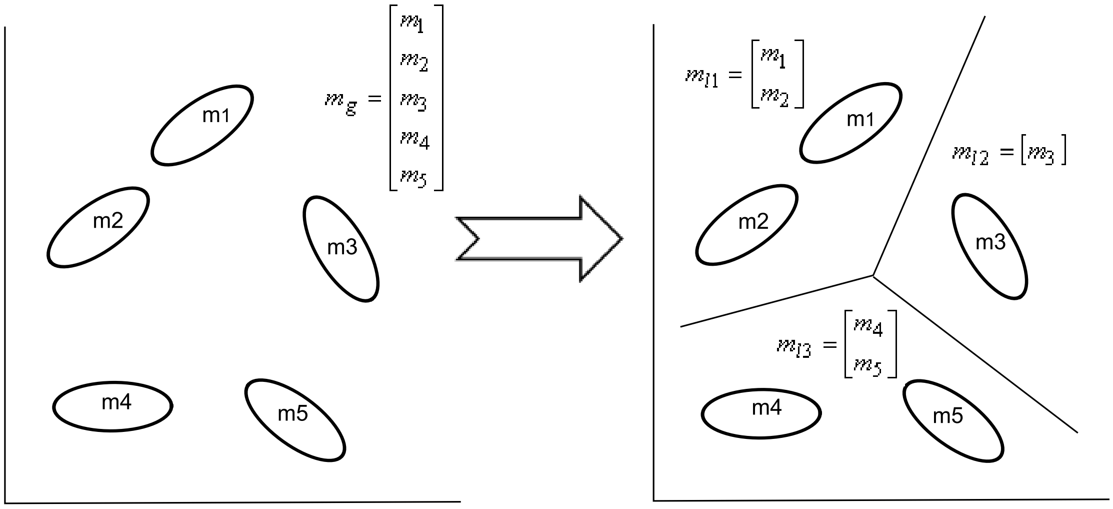
$m_g = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \end{bmatrix}$
m1
m2
m3
m4
m5
$m_{l1} = \begin{bmatrix} m_1 \\ m_2 \end{bmatrix}$
$m_{l2} = [m_3]$
$m_{l3} = \begin{bmatrix} m_4 \\ m_5 \end{bmatrix}$
m1
m2
m3
m4
m5

FIGURE 5.1: Global versus Local Speaker Representation.

These arguments motivated us to work on local speaker and channel modeling methods. If we can partition the acoustic space in a meaningful manner, we can work on each partition separately. This is demonstrated in Figure 5.1. In global speaker representation we obtain the speaker supervector by concatenating all the mean vectors of the target speakers' GMM, whereas in local speaker representation we partition the space into subregions and local speaker vectors are obtained for each partition by concatenating the mean vectors in the subregion. Subsequent channel compensation and/or speaker modeling and scoring steps may be applied in these low-dimensional local spaces. Depending on how local systems are combined, ways to several alternative methods will be opened. Working on local spaces will make it possible to estimate full rank covariance matrices whose maximum rank is not determined by the number of data at hand, but by the intrinsic dimensionality of the space we are modeling. It will also make it possible to implement algorithms for learning the manifold of the data.

Most of the current state of the art algorithms may be implemented in a local fashion. In the next section, we will discuss several scenarios for redesigning the architectures of current high-ranking speaker verification algorithms to support local speaker modeling and channel compensation. Next, we will propose a partitioning method for the acoustic space. A GMM Supervector SVM system with local channel compensation will be presented and experimental results will be given on NIST SRE10 dataset. Finally, this chapter concludes with a discussion.

5.1 Architectures for Local Speaker Representation and Channel Compensation

In this section, we discuss architectures to implement local versions of the current speaker verification algorithms. Figure 5.2 shows one possible architecture (will be called Architecture 1 from now on) for this purpose. This architecture proposes to concatenate channel compensated local vectors to obtain a new supervector for the target speaker where we apply speaker modeling and scoring tehniques. Now, we describe the steps to implement one of the state of the art speaker verification systems, namely GMM Supervector SVM system, using this architecture.

In GMM Supervector SVM, GMM local vectors will be concatenated after local channel compensation and scored with a previously trained SVM. The “Local Channel Compensation” blocks may include a variety of possible techniques used for SVM based system. This may include local versions of NAP and WCCN algorithms. In WCCN, and also in NAP when session variability is selected as the nuisance variable, we calculate the

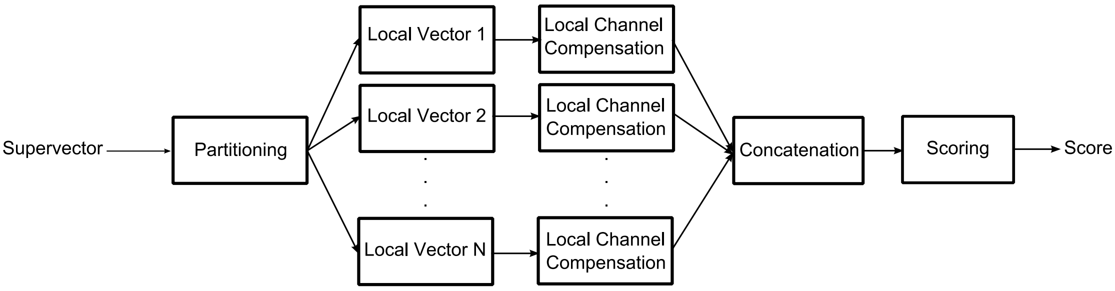

FIGURE 5.2: Local System Architecture 1.

within speaker covariance matrix C given as:

$$C = \frac{1}{N} X X^T, \tag{5.1}$$

where N is the number of training utterances, and X is a matrix whose columns contain mean removed training utterances obtained by:

$$x_i^j - \mu_i, \tag{5.2}$$

μ_i being the mean supervector for the i^{th} speaker, and x_i^j is the supervector for one of his training utterances. In NAP we calculate the top M eigenvectors of C as our nuisance space and remove the projections on this nuisance space from the supervectors to perform channel compensation. Let V be the matrix whose columns are the top M eigenvectors. The NAP projection P becomes

$$P = I - VV^T, \tag{5.3}$$

and the NAP kernel becomes

$$K(x, y) = (Px)^T Py = x^T P^T Py = x^T Py. \tag{5.4}$$

Note that the dimension of C is $F \times F$ where F is the dimension of the supervector which is quite large. It is not possible to directly calculate the eigenvectors of C, and instead we calculate the eigenvectors of D given as;

$$D = \frac{1}{N} X^T X. \tag{5.5}$$

Note that the dimension of D is $N \times N$. The eigenvalues of D and C are equal, and the relation between v_i, the i^{th} eigenvector of C, and u_i, the i^{th} eigenvector of D, is given by:

$$v_i = \frac{1}{(N\lambda_i)^{1/2}} X^T u_i, \tag{5.6}$$

where λ_i is the i^{th} eigenvalue.

In WCCN, we calculate the inverse of C, namely $R = C^{-1}$ and use this in kernel calculations as:

$$K(x, y) = x^T R y. \tag{5.7}$$

The rank of C and the number of eigenvectors is limited with the number of utterances in the dataset. Since WCCN needs the inverse of C which is impossible to calculate, kernel principal component analysis is usually used first to reduce dimension and WCCN is carried on this much smaller space [30]. PCA-complement of the supervector is appended after WCCN to prevent the removal of valuable speaker information by such a sharp dimension reduction.

For the local versions of these algorithms none of these is necessary since the dimension of each of the local spaces is small and the rank is not determined by the number of training utterances. Inverse of C can hopefully be calculated and used for channel compensation. Note that working on local spaces leads to block-diagonal versions of the original algorithms. Let for each supervector x, we extract K local vectors $\{x_1, x_2, ..., x_K\}$ according to our partitioning method, where K is the number of local partitions. In this case, the within speaker covariance matrix C becomes;

$$C = \begin{pmatrix} C_1 & & & & \\ & \ddots & & & \\ & & C_i & & \\ & & & \ddots & \\ & & & & C_K \end{pmatrix}, \tag{5.8}$$

where C_i is the within speaker covariance matrix for the i^{th} local partition. The NAP projection P also becomes block-diagonal,

$$P = \begin{pmatrix} I - V_1 V_1^T & & \\ & \ddots & \\ & & I - V_K V_K^T \end{pmatrix}, \tag{5.9}$$

where V_i is the top-M eigenvector matrix of the i^{th} local partition. Each block may be solved and applied separately. Similarly in WCCN inverse matrix R becomes block-diagonal and each local system can be solved on its own.

Working on a low-dimensional space also gives us the opportunity to investigate other channel compensation schemes not used in GMM Supervector SVM systems up to now. One example may be the LDA + WCCN method used in TVS systems after reducing dimensionality with Factor Analysis. Note that there may be speaker-specific data loss in this dimensionality reduction step of TVS, this is not the case for local channel compensation schemes.

Figure 5.3 shows the other architecture (will be called Architecture 2 from now on) proposed. In this architecture speaker modeling and scoring is performed on each of the channel compensated local vectors separately and the scores are fused to obtain the overall score. Logistic Regression algorithm typically used in speaker and language recognition for system combination may be a suitable choice for the fusion step.

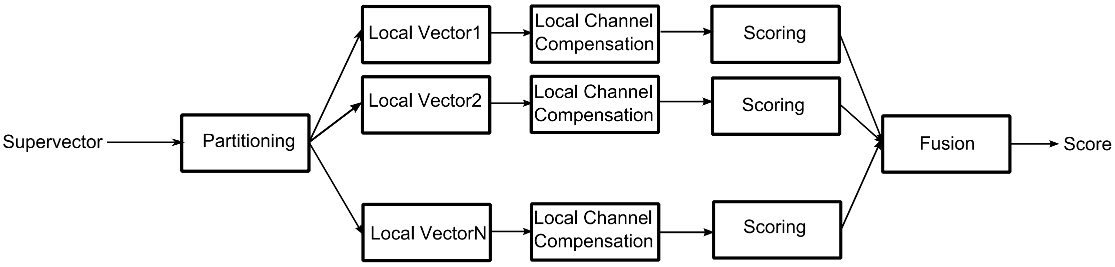

FIGURE 5.3: Local System Architecture 2. Systems using this architecture are not implemented in this thesis due to time constraints.

We can accept the speaker modeling and scoring blocks in both architecture as separate classifiers. The inputs to these blocks are the channel compensated speaker-dependent local vectors. In Architecture 1, we do an "early fusion" by concatenating these local vectors (features for the scoring system) and training a single classifier for the whole. In Architecture 2, "late fusion" is performed by training a separate classifier for each local space and combining the scores to obtain the final score and decision. In fact there is one more alternative, at least for the SVM classifiers, which is named Multiple Kernel Learning (MKL) and considered as "intermediate fusion" [31, 32]. In MKL, instead of using a single kernel or a fixed combination of kernels, we use a parameterized (by the weights of the kernels) combination of multiple kernels and the weights are also learned during training. We can use different kernel types on the same data, or we can use the same type of kernel with different parameters each focused on separate portions of the feature space. In our case there may be as many kernels as the local spaces and we can learn the weights of the kernels during training. These weights are learnt for SVM of each target speaker, hence they are speaker specific. This will give us the chance to learn whether the discriminating capability of each local space is speaker dependent or not.

In this thesis, we have only built GMM Supervector SVM systems using Architecture 1. Local NAP has been investigated. See the Experiments section for the details of the system.

5.2 Partitioning the Acoustic Space

When we decide to work on local spaces separately, the question of how to partition the acoustic space arises. One obvious choice is to use Gaussian mixtures of the UBM as building blocks. If we continue with this choice new questions appear like:

- What will be the merging criterion for obtaining local spaces from UBM mixtures?
- Will the number of mixtures in a local group be fixed or estimated for each group, and if fixed how will it be determined?
- Will we give permission to local groups to contain overlapping mixtures (same mixture in several local groups)?

Although these questions seem worthy to work on (especially trying to find a merging criterion suitable for our later goal : channel compensation), we will follow a basic solution for this task. We know from GMM-UBM systems that only several mixtures are "active" for a given speech frame which gives rise to Top-N scoring where N is typically 5 or 10. We, therefore, choose to group UBM mixtures according to distances between them. For each UBM mixture we evaluate m_i as:

$$m_i = \sqrt{w_i}\Sigma_i^{-1/2}\mu_i^s. \tag{5.10}$$

and the distance between mixtures i and j is the cosine distance between m_i and m_j given by

$$\frac{m_i \cdot m_j}{\parallel m_i \parallel \parallel m_j \parallel}. \tag{5.11}$$

After calculating the distances between mixtures we apply a greedy algorithm to group the mixtures and obtain the local spaces. The algorithm takes a fixed number for the mixtures in a local space and begins with the closest two mixtures. We select one of these mixtures as the base mixture and add to the group the next closest mixture to this base mixture. This process continues until the given fixed number of mixtures are added to the group. We do not accept overlap between local groups so these mixtures are marked as used and will not be added to the subsequent local groups. The process lasts when all the mixtures are put in a local group.

5.3 Experiments

We have built a GMM Supervector SVM system using Architecture 1 shown in Figure 5.2. The system uses Local NAP for channel compensation. The front-end, voice-activity detector and the 2048 mixture gender dependent UBM's are the same as those in Chapter 4. For the training of NAP transform we used NIST SRE06 and NIST SRE08 databases. The breakdown of the data used is shown in Table 5.1.

Speech Type	Male	Female
phone	2469	3744
microphone	984	1195
interview	1344	1827
Total	4797	6766

Table 5.1: Breakdown of Speech Data for Local NAP Training.

We applied Z-norm score normalization for both systems. The breakdown of Z-norm and impostor utterances are listed in Tables 5.2 and 5.3, respectively.

We have used 64 mixtures in each local group and since each GMM mean is 39 dimensional, the local vectors' dimension is 2496. Plot of the eigenvalues of the covariance matrix C in Equation 5.1, is given for a local group of male UBM in Figure 5.4. The percentage of energy versus the percentage of eigenvectors kept is given in Figure 5.5

Speech Type	Male	Female
phone	1053	1267
microphone	303	398
interview	336	588
Total	1692	2253

Table 5.2: Breakdown of impostor utterances for local systems.

Speech Type	Male	Female
phone	663	1130
microphone	238	262
interview	375	440
Total	1258	1832

Table 5.3: Breakdown of Z-norm utterances for local systems.

for the same local group. We can see that none of the eigenvalues go to zero and the dimension of the channel space is equal to the total dimension of the local space, although the first eigenvalues are significantly larger than the subsequent ones.

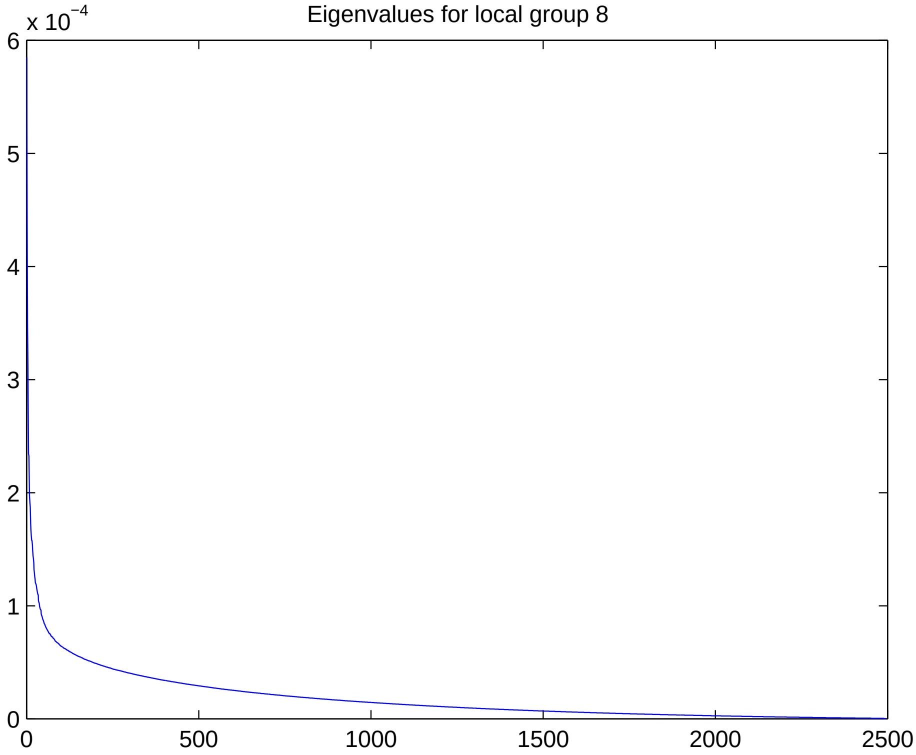

Figure 5.4: Plot of eigenvalues for local group 8 of the male UBM.

We have tested our systems on core condition of NIST SRE10 dataset. When applying Local NAP, the dimension of the removed nuisance space is selected as 40 for

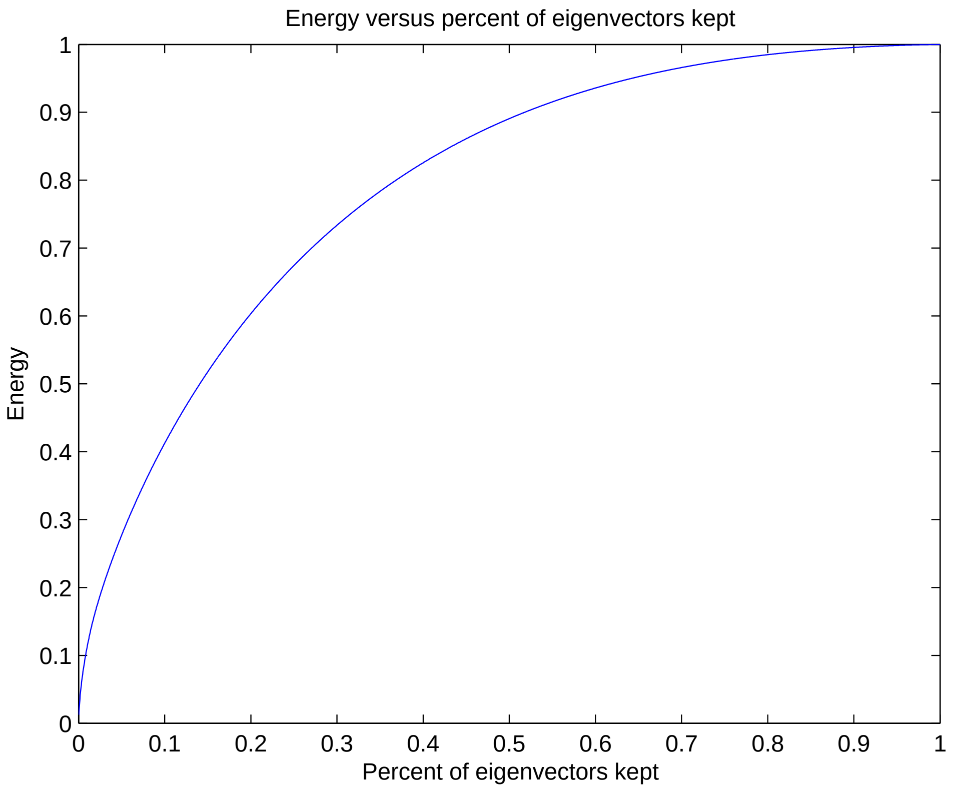

Figure 5.5: Plot of energy versus eigenvectors kept.

all local regions. In Figure 5.6, the DET plots for conditions 1 (interview-interview same microphone) and 2 (interview-interview different microphone) are given. In Figure 5.7, the DET plots for conditions 3 (interview-normal vocal effort telephone) and 4 (interview-normal vocal effort microphone) are given. Local NAP only slightly improves performance for the same microphone case (condition 1). In the other three conditions, where significant channel variations between training and test utterances exist, Local NAP greatly improves the performance.

In Figure 5.8, the DET plot for condition 5 (normal vocal effort-normal vocal effort different telephone) is given. In Figure 5.9, the DET plots for conditions 6 (normal vocal effort-high vocal effort telephone) and 7 (normal vocal effort-high vocal effort microphone) are given. In Figure 5.10, the DET plots for conditions 8 (normal vocal effort-low vocal effort telephone) and 9 (normal vocal effort-low vocal effort microphone) are given. Local NAP improves performance on all of the cases and at every operating point on the DET plots. This shows the power of Local NAP for intra-speaker variability compensation especially when significant channel variations exist between train and test utterances.

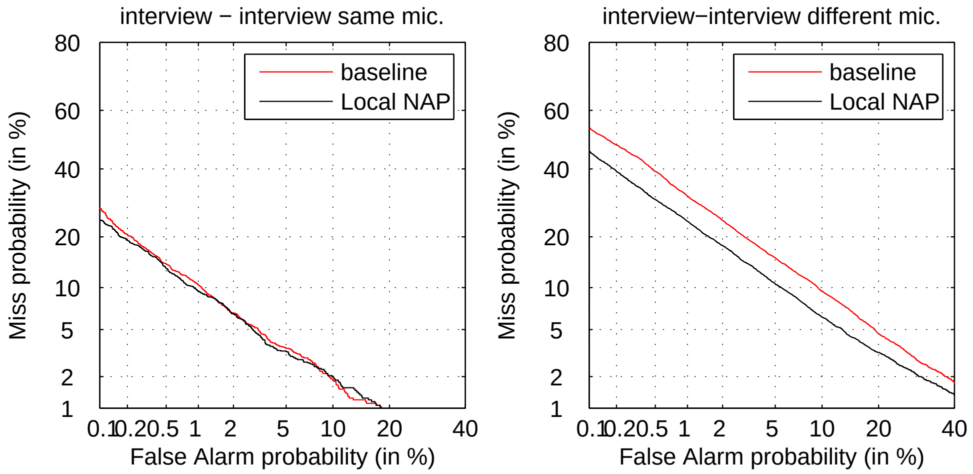

Figure 5.6: DET Plots of Local NAP vs baseline for Conditions 1 and 2.

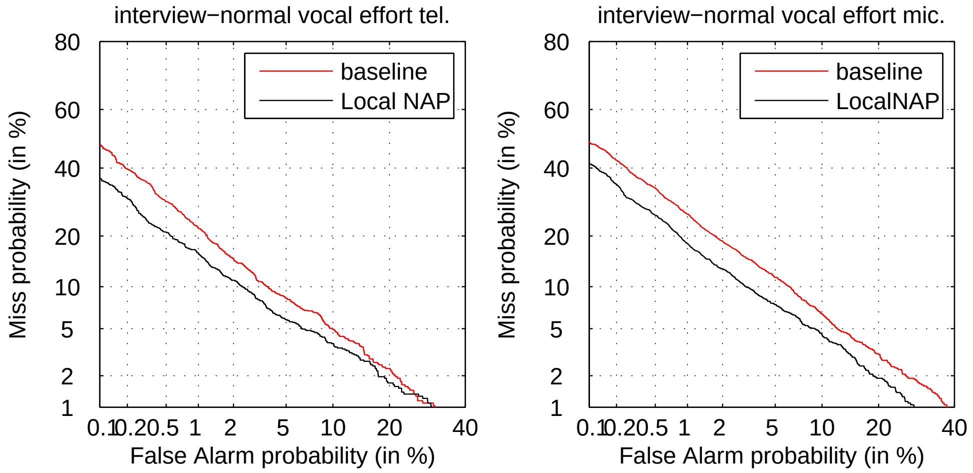

Figure 5.7: DET Plots of Local NAP vs baseline for Conditions 3 and 4.

In Tables 5.4 and 5.5, equal error rate (EER) and minimum DCF (minDCF) values are given for all conditions. Relative improvements between 15% to 30% are achieved in EER except condition 1. The results are similar in the minDCF case where we are evaluating the performance for very low false alarm rates.

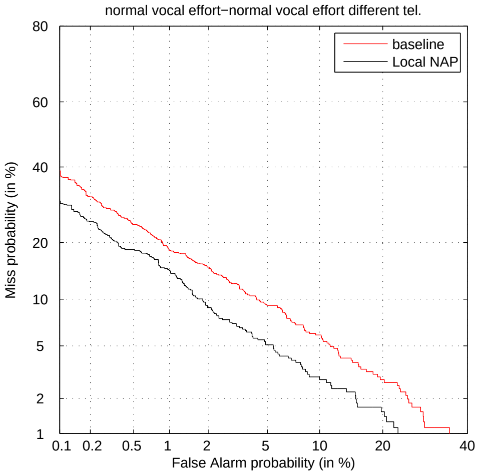

FIGURE 5.8: DET Plots of Local NAP vs baseline for Condition 5.

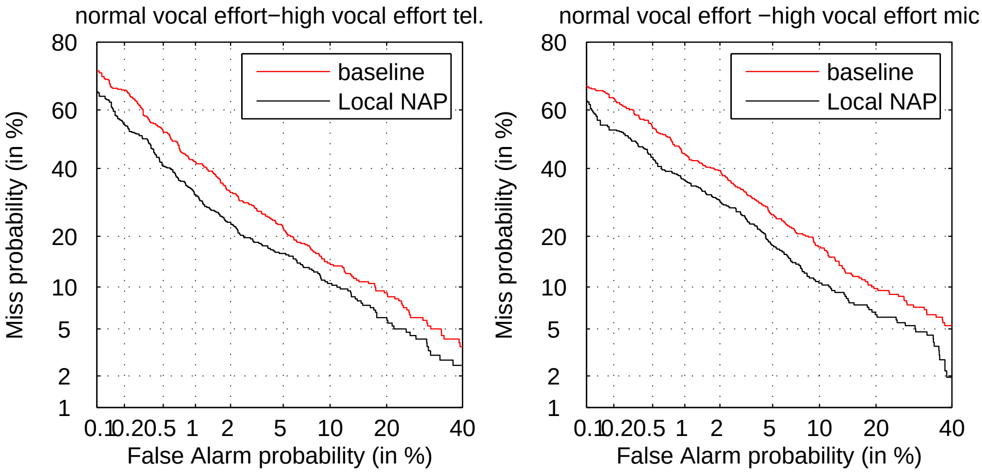

FIGURE 5.9: DET Plots of Local NAP vs baseline for Conditions 6 and 7.

5.4 Discussion

In this chapter, we have proposed to partition the acoustic space into local regions to better compensate channel effects and model speakers. We have discussed two possible architectures for local systems where current state-of the art speaker verification systems can easily be adapted. In this thesis, we have only built a system within the proposed Architecture 1 which uses NAP for channel compensation in local regions. Significant

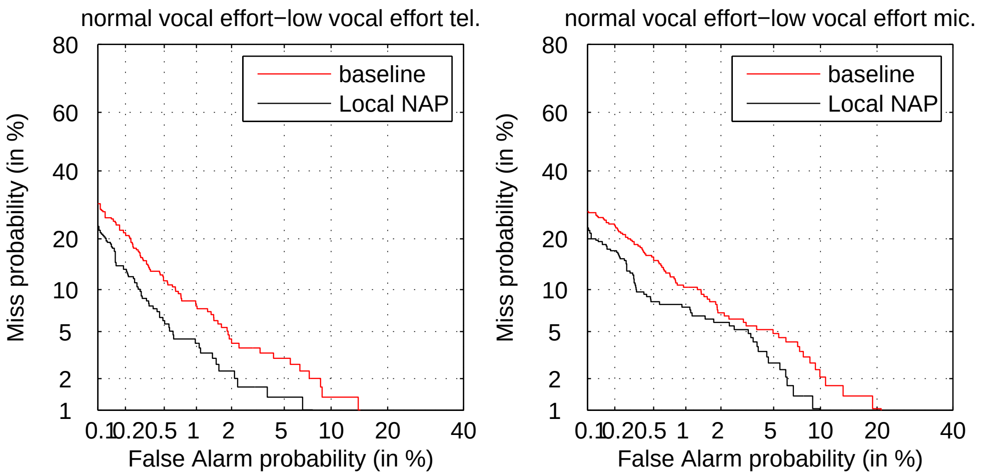

Figure 5.10: DET Plots of Local NAP vs baseline for Conditions 8 and 9.

	C1	C2	C3	C4	C5	C6	C7	C8	C9
Baseline	3.95	9.69	6.86	7.99	7.20	12.24	13.42	3.36	4.89
Local NAP	3.76	7.76	5.57	6.42	5.08	10.25	10.33	2.06	3.80

Table 5.4: EER values for the baseline and Local NAP systems for all conditions.

	C1	C2	C3	C4	C5	C6	C7	C8	C9
Baseline	5.849	8.338	8.595	7.622	6.097	9.466	8.858	6.109	5.940
Local NAP	5.109	7.729	7.038	6.983	5.812	9.272	8.217	5.202	3.586

Table 5.5: minDCF values ($\times 10^{-4}$) for the baseline and Local NAP systems for all conditions.

performance improvements are achieved for all conditions where substantial channel variations exist.

Working on local regions of low-dimensionality gives the ability to build more tractable systems and parameters like channel eigenvectors may more correctly be calculated. Moreover, it gives the opportunity to apply techniques not used before like LDA without previous dimensionality reduction or nonlinear manifold learning type algorithms. Partitioning the acoustic space is also an issue desiring further research. In this sense, the subject is not over yet and there seems to be much way to go in this direction. We plan to increase our efforts to apply other algorithms in a local sense and to find better ways of partitioning the acoustic space.

Chapter 6

Random Sampling for Acoustic Event Variability Compensation

For years, speaker verification studies focused on removing unwanted variabilities (channel, session, etc.) at the modeling stage. Joint Factor Analysis has given great improvement on this purpose. Despite its success for conditions typical in NIST SRE core trials with utterance lengths of 2-3 minutes in average, it is shown in [33] that the same performance could not be obtained for utterances of smaller lengths in training and/or testing like the ones in 10sec conditions of NIST SREs. For small sized utterances speaker factors still showed positive impact, whereas performance degradation has been observed with the addition of channel factors. The authors, in a subsequent work [34], observed that obtaining the session variability matrix **U** of JFA from training utterances with lengths matched with evaluation conditions improves the performance. This means that session variabilities depend on the utterance length. Since it is assumed that the session subspace carries mostly variabilities due to nuisances like transmission channel, handset and microphone used, this situation seems to be contradictive.

Actually, this perception of contradiction is due to the habit of using channel and session interchangeably in the speaker verification literature. In [35], the authors suspected that another source of session variability is becoming dominant for short utterances which could not be modeled by the **U** matrix typically learned from the NIST SRE utterances of longer length. They suggested that phonetic variability is the source of this performance degradation and propose to divide the session variabilities into two subsets as intersession variabilities and within-session variabilities. They divide each utterance into a sequence of N short segments and obtained a separate GMM supervector from them. For the short segment GMM supervectors the intersession characteristics are the same while they vary in a low dimensional within-session subspace. Under this proposal, JFA model for a short segment s_n becomes:

$$s_n = m + Vy + dz + U_I x + U_W w_n. \tag{6.1}$$

While $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ will be the same for all short segments of an utterance, they will have different within-session factors w_n. The authors used a phone recognizer to obtain the short segments. They have been able to obtain a JFA system flexible to be used for all utterance lengths. That is, their JFA system with within-session variability compensation gave better or comparable results with JFA systems of matched utterance length. However, significant performance improvements compared to the baseline could not be achieved.

We will use the more general term “acoustic event variability” instead of phonetic variability. The acoustic events may be phones, broad phonetic classes or Gaussian mixtures. The acoustic variability may be divided into two groups; the across acoustic event variability and the variabilities coming from the actual realization of the acoustic events. The works in [35, 36] are mostly related with the across acoustic event variabilities where they try to compensate the effects observing different acoustic events in train and test. For long utterances in train and test, we will observe every acoustic event sufficiently and hopefully across acoustic event variability will not be a significant problem, while for short utterances this is not the case. The variabilities coming from the actual realization of the acoustic event is part of the intra-speaker variability. Intra-speaker variability does not come only from environmental effects like transmission channel or microphone but also comes from the speaker himself. It is well known that factors like speaking style and emotional state affects the speakers performance of the acoustic events.

In [37], a method to model intra-speaker variability for the Anchor Model based speaker verification is proposed. In Anchor Modeling, each target speaker is represented relative to a pre-trained speaker set called anchor models. The speaker is characterized by a vector of log-likelihood ratios obtained by scoring target speaker training data with each of the anchor models’ GMM and expressing the likelihood relative to UBMs’ likelihood. This vector is called Speaker Characterization Vector (SCV) and speaker verification is performed by calculating the distance between clamimed speakers' SCV and SCV of the test data. This method lacks intra-speaker variability modeling and in [37], the authors proposed to describe each speaker as a distribution on the anchor space rather than a single point to remove this deficiency. Inspired by this work, in this chapter we propose a simple method to compensate the acoustic event variabilities at the modeling stage of a GMM Supervector SVM system. The next section will describe our method and we will conclude with experimental results on the NIST SRE 2006 dataset.

6.1 Acoustic Event Variability Compensation for GMM Supervector SVM System

In GMM Supervector SVM systems, when the target speaker has a single training utterance, the SVM algorithm learns a boundary that separates the single positive supervector from many negative supervectors of impostor utterances. That is we are representing the speaker as a single point like the traditional Anchor Modeling. Inspired by [37], instead of a single point in the space we want to represent possible locations where the speaker might be. To achieve this goal, we propose to extract short segments from the utterance and use them in SVM training. More specifically, for each short segment extracted, we randomly select frames from the utterance until we reach a previously selected fixed size. Let $X = \{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T\}$ be a training utterance of T frames. We obtain the supervector $\mathbf{x}$ by concatenating the variance normalized mean vectors of the GMM obtained by MAP adaptation of the UBM using frames in X. Let $\mathbf{x_i}$ be the supervector for the i^{th} short segment $S_i = \left\{\mathbf{o}_{\sigma_i(1)}, \mathbf{o}_{\sigma_i(2)}, \ldots, \mathbf{o}_{\sigma_i(D)}\right\}$, where $\{\sigma_i(1), \sigma_i(2), \ldots, \sigma_i(D)\}$ are the randomly selected indices and D is the selected fixed size for the short segments. We obtain GMM supervectors $\mathbf{x_i}$ from each short segment as well as from the original utterance itself. We use all of these GMM supervectors as positive examples in SVM training. This will put within-session variability compensation in the modeling stage.

The motivation to use random sampling of the frames is to guarantee that each short segment supervector contains information from most parts of the acoustic space. This is especially important if we use relevance MAP adaptation rather than Eigenvoice MAP adaptation for GMM training since relevance MAP will update only the mixture components that it sees in the adaptation data. Using this procedure we expect to model the within-session variability coming from the variations in the actual realization of the acoustic events. In our case acoustic events are Gaussian mixtures.

The fixed size chosen for the short segments is an important parameter. We do not want it too long making the short segment GMM supervectors saturating to the overall utterance supervector. The size of the supervector may also be a compromise between modeling within-session variabilities coming from the actual realizations of the mixtures or modeling across mixture variabilities. If we make the fixed size so small that only few acoustic events can be represented, then across mixture variabilities will dominate. Since in this thesis EigenVoice MAP has not been implemented, we prefer to choose the size so that variabilities coming from actual realizations dominate and across mixture variabilities are represented to some extent.

6.2 Experiments

To understand the effect of random sampling within an utterance we have run experiments on the male portion of 1conv4w-10sec4w and 1conv4w-1conv4w conditions of the NIST SRE06 dataset. We have trained a 1024 mixture gender-dependent Universal Background Model from NIST SRE04 and Switchboard corpora using approximately 127 hours of speech. Same front-end is used as the one in Chapter 4. We use 500 utterances selected from NIST SRE04 and Switchboard corpora for impostor utterances and for Z-norm score normalization when applied. We use 274 speakers from the NIST SRE05 dataset as T-norm speakers when T-norm score normalization is applied.

We have generated 10 short segments for each training utterance. Together with the supervector obtained from the whole utterance, we have 11 true examples for each target speaker in SVM training. We have tried two values for the fixed size; 10 sec and 24 sec. For the 1conv4w-10sec4w condition, we first tried a baseline system where no score normalization is applied. In Figure 6.1 the DET plots for this case is shown. It seems that the system with 24 sec short utterances performed better than the baseline except low false alarm region whereas 10sec fixed size system is generally worse than the baseline and comparable to it on high false alarm rates. The EER and minDCF values are given in Table 6.1.

	EER	minDCF($\times 10^{-2}$)
Baseline	14.14	5.43
24sec	**13.27**	**5.39**
10sec	14.00	5.71

Table 6.1: EER and minDCF values for 1conv4w-10sec4w condition

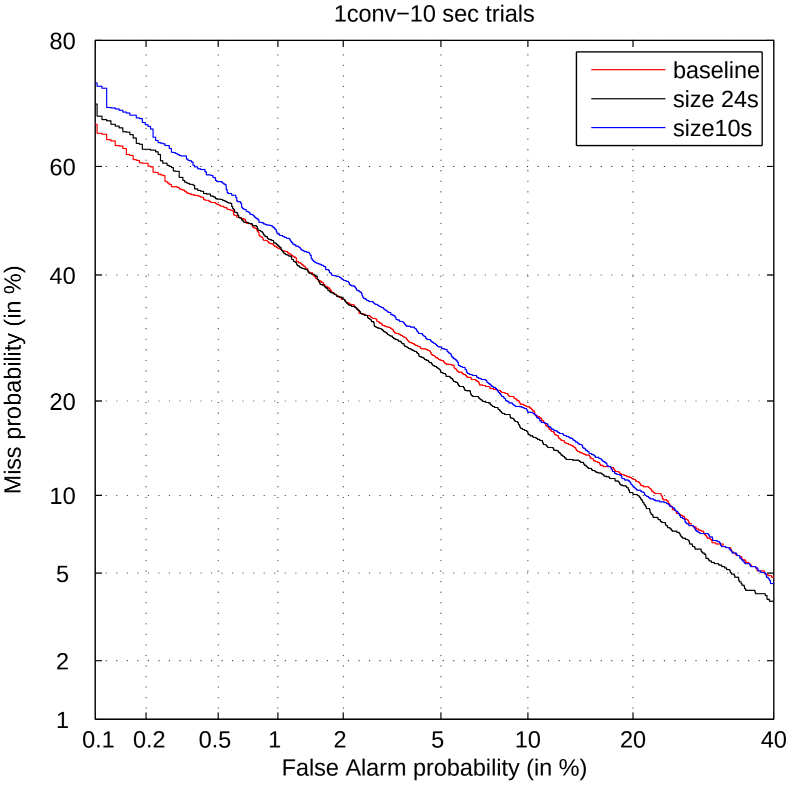

Figure 6.1: DET Plots for 1con4w-10sec4w condition.

We have applied T-norm score normalization to see how things change in case of score normalization. In Figure 6.2 the results for this case is shown. Table 6.2 shows the EER and minDCF values. T-norm score normalization improved the performance of all the systems slightly, leaving the relative performances unchanged. The higher miss rates in 10 sec seem to be the effect of using relevance MAP in adaptation. Since we could not update the mixtures we do not see in adaptation, using very short utterances causes the rejection of 10 sec positive test trials distant to the 10 sec short segments of the adaptation.

We also wanted to see how modeling within-session variabilities using random sampling affects performance in long test utterances. For this purpose we have run test for 1conv4w-1conv4w case. The baseline system has Z-norm score normalization in this

	EER	minDCF($\times 10^{-2}$)
Baseline	14.00	5.16
24sec	**12.97**	5.17
10sec	13.85	5.48

Table 6.2: EER and minDCF values for 1conv4w-10sec4w condition after T-norm score normalization.

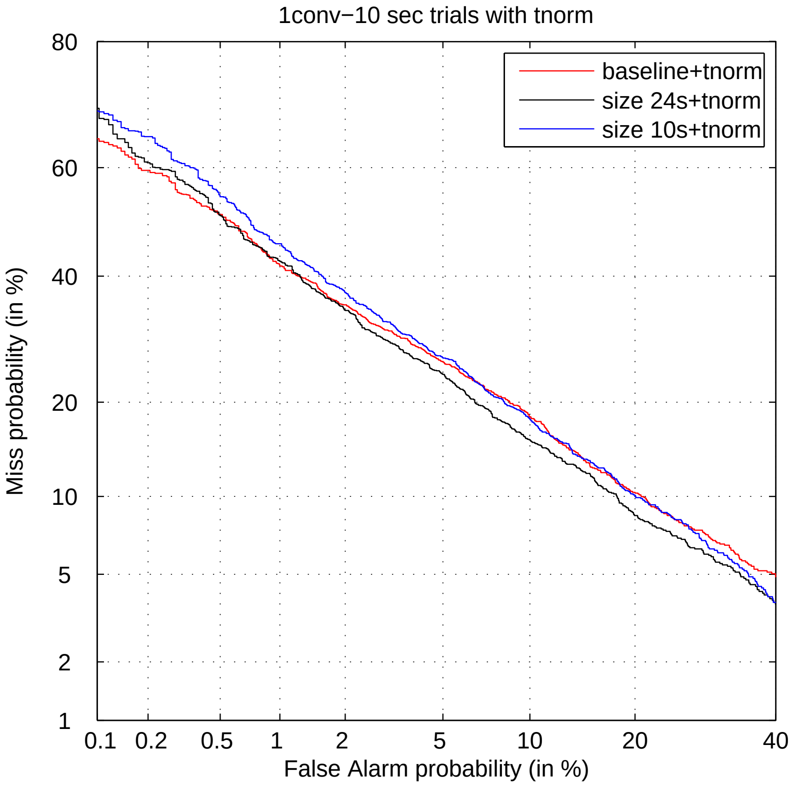

Figure 6.2: DET Plots for 1con4w-10sec4w condition with T-norm score normalization.

case. Figure 6.3 shows the DET plot comparing performances of baseline, 10 sec. fixed size and 24 sec. fixed size systems and Table 6.3 shows the EER and minDCF values obtained. Figure 6.4 and Table 6.4 show the case when T-norm is also added to all systems. In this case both of the random sampling systems outperform the baseline system. T-norm score normalization increases performance for all systems and do not change the relative performances much. Random sampling seems to be effective even for long utterance test segments. The system performances seem to saturate for high false-alarm rates. We can conclude that adding short segments through random sampling makes our system cope better with intra-speaker variabilities.

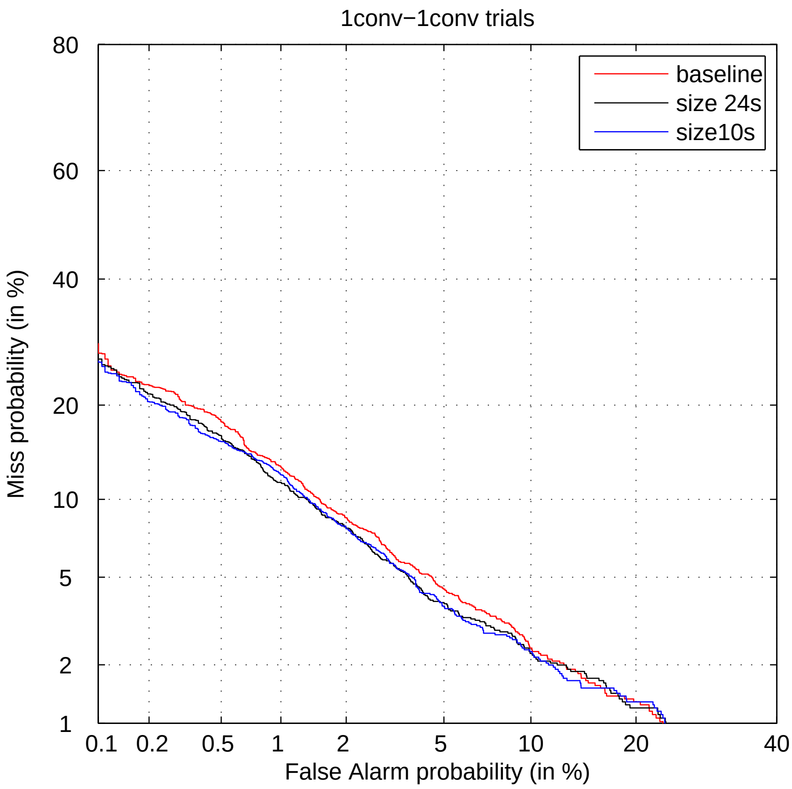

FIGURE 6.3: DET Plots for 1con4w-1con4w condition with Z-norm score normalization.

	EER	minDCF($\times 10^{-2}$)
Baseline	4.69	2.15
24sec	**4.22**	2.06
10sec	4.27	**2.03**

TABLE 6.3: EER and minDCF values for 1conv4w-1conv4w condition with Z-norm score normalization.

	EER	minDCF($\times 10^{-2}$)
Baseline	4.36	1.96
24sec	3.94	1.86
10sec	**3.90**	**1.85**

TABLE 6.4: EER and minDCF values for 1conv4w-1conv4w condition with ZT-norm score normalization.

6.3 Conclusion

In this chapter we have tried to deal with one component of intra-speaker variability through adding short utterances with random sampling in the modeling stage. We tried to model the differences in actual realizations of the acoustic events and instead of

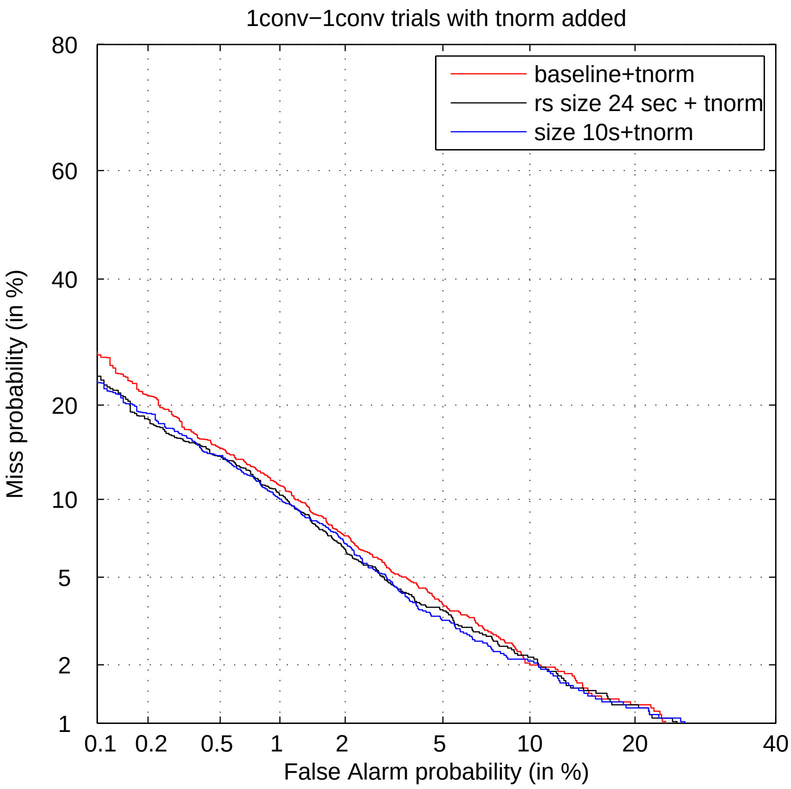

Figure 6.4: DET Plots for 1con4w-1con4w condition with ZT-norm score normalization.

representing the speaker as a single point in the space we give possible more locations where the speaker may be found. The method may also be used to model across acoustice event variabilities if a suitable adaptation algorithm like eigenvoice MAP is used. We have obtained performance improvements both for short (10sec) and long (1conv: 2-3 minutes) test utterances. One option that desires to be investigated is using short utterances in training. In the future, we plan to incorporate eigenvoice MAP adaptation to our system and work on this condition also. The method we proposed is a simple and easy to implement one, yet seems to be effective for a wide range of test conditions.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, we have discussed the assumption of low-dimensionality of channel space typical in current state-of-the art speaker verification systems. We have proposed to partition the acoustic space into local regions where we can apply channel compensation techniques separately. This would lead to more tractable solutions and gives a better view of the channel and speaker spaces. We argued that obtaining a single number as the distance between two speakers is questionable and using local regions may also support better modeling of inter-speaker variabilities. We have proposed two architectures for speaker verification in local regions in which current approaches can easily be used. The two architectures both apply channel compensation locally and differ in where the subsequent speaker modeling and scoring steps are realized. We have implemented a GMM supervector SVM system with local NAP based on one of these architectures.

We have also worked on a special case of intra-speaker variability, namely within-session variability. We have proposed a method to improve performance by compensating variabilities mainly coming from actual realizations of phones as well as variabilities across phones. We have shown that our technique not only improves performance for short test utterances but also for long test utterances.

7.2 Future Work

For future improvements on our proposals, we will focus on the below directions.

- Further studies on local region based speaker verification must be explored. We want to apply other approaches like TVS, LDA + WCCN, and JFA in the local architectures we proposed. In this thesis we only used one of the architectures (Architecture 1) where only channel compensation is applied locally. It may be interesting to model speakers and inter-speaker variabilities locally. We also want to explore how well local systems may be fused in the score level. Since the dimension of local regions are small enough we can leave the linearity assumption and try to learn the manifold of speaker space.
- Better ways of partitioning the acoustic space is also worth working on. It will be most beneficial if we can find a way to partition the space that will help in subsequent steps of channel compensation.
- In this thesis we have used random sampling to generate short segments from the training utterance. We tried to learn a better hyperplane by SVM taking phonetic intra-speaker variabilities into account. As a future work, we plan to investigate ways to remove these variabilities by feature transforms applied to the training supervectors.

Bibliography

[1] J. Pelecanos and S. Sridharan. Feature warping for robust speaker verification. In *2001: A Speaker Odyssey-The Speaker Recognition Workshop*, 2001.

[2] H. Hermansky and N. Morgan. RASTA processing of speech. *IEEE transactions on speech and audio processing*, 2(4):578–589, 1994.

[3] D.A. Reynolds. Channel robust speaker verification via feature mapping. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03)*, volume 2, 2003.

[4] A. Adami, R. Mihaescu, D.A. Reynolds, and J.J. Godfrey. Modeling prosodic dynamics for speaker recognition. In *Proc. ICASSP*, volume 4, pages 788–791, 2003.

[5] W.D. Andrews, M.A. Kohler, J.P. Campbell, J.J. Godfrey, and J. Hernández-Cordero. Gender-dependent phonetic refraction for speaker recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2002. Proceedings.(ICASSP'02)*, 2002.

[6] J. Navrátil, Q. Jin, W. Andrews, and J. Campbell. Phonetic speaker recognition using maximum likelihood binary decision tree models. In *Proc. of ICASSP*, 2003.

[7] D. Reynolds, W. Andrews, J. Campbell, J. Navratil, B. Peskin, A. Adami, Q. Jin, D. Klusacek, J. Abramson, R. Mihaescu, et al. The SuperSID project: Exploiting high-level information for high-accuracy speaker recognition. In *Proc. ICASSP*, volume 4, pages 784–787, 2003.

[8] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas. Score normalization for text-independent speaker verification systems. *Digital Signal Processing*, 10(1-3):42–54, 2000.

[9] D.A. Reynolds, T.F. Quatieri, and R.B. Dunn. Speaker verification using adapted Gaussian mixture models. *Digital signal processing*, 10(1-3):19–41, 2000.

[10] P. Kenny. Joint factor analysis of speaker and session variability: theory and algorithms. *The Centre de Recherche Informatique de Montreal, Technical Report CRIM-06/08-13*, 2005. URL http://www.crim.ca/perso/patrick.kenny/FAtheory.pdf.

[11] P. Kenny, M. Mihoubi, and P. Dumouchel. New MAP estimators for speaker recognition. In *Eighth European Conference on Speech Communication and Technology*, 2003.

[12] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel. Speaker and session variability in GMM-based speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(4):1448–1460, 2007.

[13] P. Kenny, G. Boulianne, and P. Dumouchel. Eigenvoice modeling with sparse training data. *IEEE Transactions on Speech and Audio Processing*, 13(3):345–354, 2005.

[14] O. Glembek, L. Burget, N. Dehak, N. Brummer, and P. Kenny. Comparison of scoring methods used in speaker recognition with Joint Factor Analysis (PDF). In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2009. Proceedings.(ICASSP'09)*, pages 4057–4060, 2009.

[15] N. Dehak, P. Kenny, R. Dehak, P. Ouellet, and P. Dumouchel. Front-end factor analysis for speaker verification. *submitted to IEEE Transaction on Audio, Speech and Language Processing.* URL http://groups.csail.mit.edu/sls/publications/2010/.

[16] N. Dehak, R. Dehak, P. Kenny, N. Brummer, P. Ouellet, and P. Dumouchel. Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification. In *Proceedings of Interspeech*, 2009.

[17] M. Senoussaoui, P. Kenny, N. Dehak, and P. Dumouchel. An i-vector Extractor Suitable for Speaker Recognition with both Microphone and Telephone Speech. In *Odyssey: The Speaker and Language Recognition Workshop, Brno, Czech Republic*, 2010.

[18] WM Campbell, DE Sturim, DA Reynolds, and A. Solomonoff. Svm Based Speaker Verification Using a GMM Supervector Kernel and NAP Variability Compensation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2006. Proceedings.(ICASSP'06)*.

[19] W. Campbell. Generalized linear discriminant sequence kernels for speaker recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, 2002.

[20] A. Solomonoff, W.M. Campbell, and I. Boardman. Advances in channel compensation for SVM speaker recognition. In *Proc. ICASSP*, pages 629–632, 2005.

[21] A.O. Hatch and A. Stolcke. Generalized linear kernels for one-versus-all classification: application to speaker recognition. In *2006 IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings*, 2006.

[22] Alvin Martin and Mark Przybocki. The nist speaker recognition evaluation series. *National Institute of Standard and Technology's web-site.* URL http://www.nist.gov/speech/test/spk.

[23] URL http://www.ldc.upenn.edu/.

[24] M.A. Przybocki and A.F. Martin. Odyssey Text Independent Evaluation Data. In *2001: A Speaker Odyssey-The Speaker Recognition Workshop*. ISCA, 2001.

[25] C. Cieri, L. Corson, D. Graff, and K. Walker. Resources for new research directions in speaker recognition: The Mixer 3, 4 and 5 corpora. In *Proc. Interspeech*, pages 950–954, 2007.

[26] C. Cieri, J.P. Campbell, H. Nakasone, D. Miller, and K. Walker. The mixer corpus of multilingual, multichannel speaker recognition data. In *Proc. 4th International Conference on Language Resources and Evaluation*, pages 26–28, 2004.

[27] C. Cieri, W. Andrews, J.P. Campbell, G. Doddington, J. Godfrey, S. Huang, M. Liberman, A. Martin, H. Nakasone, M. Przybocki, et al. The mixer and transcript reading corpora: Resources for multilingual, crosschannel speaker recognition research. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*, pages 117–120, 2006.

[28] L. Brandschain, C. Cieri, D. Graff, A. Neely, and K. Walker. Speaker recognition: building the Mixer 4 and 5 corpora. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation*, 2008.

[29] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel. A study of inter-speaker variability in speaker verification. *IEEE Trans. Audio, Speech and Language Processing*, 16(5):980–988, July 2008.

[30] A.O. Hatch, S. Kajarekar, and A. Stolcke. Within-class covariance normalization for SVM-based speaker recognition. In *Ninth International Conference on Spoken Language Processing*, pages 1471–1474, 2006.

[31] G.R.G. Lanckriet, N. Cristianini, P. Bartlett, L.E. Ghaoui, and M.I. Jordan. Learning the kernel matrix with semidefinite programming. *The Journal of Machine Learning Research*, 5:27–72, 2004.

[32] M. Gonen and E. Alpaydin. Localized multiple kernel learning. In *Proceedings of the 25th international conference on Machine learning*, pages 352–359. ACM, 2008.

[33] R. Vogt, C. Lustri, and S. Sridharan. Factor analysis modelling for speaker verification with short utterances. In *Odyssey: The Speaker and Language Recognition Workshop*, 2008.

[34] R. Vogt, B. Baker, and S. Sridharan. Factor analysis subspace estimation for speaker verification with short utterances. In *Proc. Interspeech*, pages 853–856, 2008.

[35] R. Vogt, J. Pelecanos, N. Scheffer, S. Kajarekar, and S. Sridharan. Within-session variability modelling for factor analysis speaker verification. In *Proc. Interspeech*, 2009.

[36] S.S. Kajarekar. Across-phone variability and diagonal term in joint factor analysis for speaker recognition. In *In: ICASSP, 14th-19th of March, 2010., Dallas, Texas, USA.*

[37] M. Collet, Y. Mami, D. Charlet, and F. Bimbot. Probabilistic anchor models approach for speaker verification. In *Ninth European Conference on Speech Communication and Technology*, 2005.