Universidad
Zaragoza
1542

Instituto de Investigación en Ingeniería de Aragón

Ph.D. Thesis

Advances on Speaker Recognition in non Collaborative Environments

Avances en Reconocimiento de Locutor en Entornos no Colaborativos

Jesús Antonio Villalba López

Thesis Advisor
Prof. Eduardo Lleida Solano

October 21, 2014

A mis padres

If of two subsequent events the probability of the first be $\frac{a}{N}$, and the probability of both together be $\frac{P}{N}$, then the probability of the second on supposition the first happens is $\frac{P}{a}$.

Thomas Bayes, *An Essay towards solving a Problem in the Doctrine of Chances*

It is not knowledge, but the act of learning, not possession but the act of getting there, which grants the greatest enjoyment.

Carl Friedrich Gauss

In other studies you go as far as others have gone before you, and there is nothing more to know; but in a scientific pursuit there is continual food for discovery and wonder.

Mary Shelley, *Frankenstein*

Acknowledgments

First and foremost, I would like to thank my advisor Prof. Eduardo Lleida for his guidance. When I was an undergraduate student, in 2003, Eduardo offered me my first grant to work in his group where I started to learn about speech recognition. After I graduated, he offered me a position in the Intelligent Environments lab. at Walqa technology park, where I continued working on speech, speaker and face recognition. Finally, he brought me back to the faculty where I developed most of the work included in this thesis. Through all this years, I had always counted with his trust and endless support.

I also want to thank all my colleagues of the ViVoLab group at University of Zaragoza, among them, Alfonso Ortega, David Martínez, Diego Castán, Oscar Saz, Jose Enrique García, Dayana Ribas, Julia Olcoz, Jorge Llombart and Paola García; for his friendship, collaboration, and fruitful discussions. Special thanks to Juan Diego Rosas that started with me in Walqa when there was nothing there; Carlos Vaquero that collaborated with me in the intense NIST evaluations; and Antonio Miguel for discussions about factor analysis and many other math related stuff.

During these years, I had the opportunity of doing two internships, which greatly benefited my research. I want to thank Jan Černocký for hosting my first stay at his Speech@FIT group in Brno. I also want to thank Ondřej Glembek and Pavel Matějka for their help during my stay with them, where I learned a big deal about factor analysis and how to set up a competitive speaker verification system.

I want to thank Niko Brümmer for allowing me to do my second internship with him, at Agnitio Research in Cape Town. With Niko I learned a big deal about Bayesian statistics, graphical models and variational Bayes, which were a great inspiration for many of the chapters of this thesis. I also want to thank Edward De Villiers and his sister Margaret for his hospitality during my stay in South Africa. I do not want to forget the Agnitio team in Madrid, Marta García, Luis Buera and Carlos Vaquero, that encouraged me to do this internship and that had provided invaluable feedback about many of the topics of this thesis.

I also want to thank all the participants of the Bosaris 2012 and JHU 2013 speaker recognition workshops for fruitful discussions and exchange of ideas, among them, Patrick Kenny, Najim Dehak, Lukaš Burget, Daniel García-Romero, Mireia Diez, Oldřich Plchot, Sandro Cumani, Albert Swart, Tomi Kinnunen, Yun Lei, Ignacio López-Moreno, Stephen Shum and Themos Stafylakis.

Finally, I would like to thank the entities that provided financial support for this research: the Spanish Government through national projects TIN2005-08660-C04-01, TIN2008-06856-C0504 and TIN2011-28169-C05-02; and the European Union (FEDER) through project INNPACTO IPT-2011-1696-390000.

Abstract

Speaker recognition performance is usually measured in ideal scenarios where speech is relatively clean, users are collaborative, and enough data to train speaker and channel models is available. However, when applying speaker verification (SV) in real environments, we face some challenges that deserve further research. This thesis dealt with some of them.

Speaker verification performance can decrease due to multiple causes: noise, reverberation, languages and channels different from those in the development data, etc. Thus, knowing that we can trust the speaker verification decisions is fundamental. This motivated us to study how to estimate the reliability of the decisions. We worked on solutions based on Bayesian networks. The networks model the causal relationships between the decision reliability, the speaker verification score and a set of quality measures computed from the enrollment and test utterances of the trial. The most important contribution of the thesis, in this matter, was a Bayesian network that describes how SV score distributions change when the trial segments are distorted. This network hypothesizes the existence of two scores: one observed and another hidden. The observed score or *noisy score* is the one provided by the SV system while the hidden score or *clean score* is the ideal score that we would obtain from high quality speech. From the posterior of the *clean score*, we can infer the trial reliability. This approach outperformed previous methods.

Currently, the i-vector paradigm is the state-of-the-art for SV, outperforming, in most task, JFA or SVM. For this reason, one part of the thesis focused on some issues that arise when modeling i-vector distributions. We considered the problem of simultaneously having i-vectors recorded in different conditions like multiple channel types, noise types or noise levels. We introduced a PLDA variant that approaches this problem in a principled way by allowing different channel distributions for each condition. Other issue that we addressed was how to consider the uncertainty about the model parameters that exists when the amount of development data is small compared with the i-vector dimension. We proposed to compute the posterior distribution of the model given the development data and, then, use it to integrate out the model parameters when evaluating likelihood ratios. With this method, we obtained a significant improvement on non-length-normalized i-vectors. Finally, we treated the problem of training PLDA in domains with scarce development data. We proposed to adapt a model trained on another domain, with enough data, to the target domain by using *Maximum a posteriori*.

In the last part of the thesis, we were interested in attacks to SV systems. We worked on detecting spoofing and tampering. These attacks have opposite intentions. While spoofing intends an impostor to impersonate a target speaker, tampering aims to conceal the speaker's identity. We focused on low effort attacks that criminals could perpetrate without needing any speech processing knowledge. Regarding spoofing, we studied replay attacks. For text-dependent systems the attack also involved creating the pass-phrase by cutting and pasting

excerpts from different victim's recordings. Regarding tampering, we studied covering the mouth with the hand or a handkerchief; and denasalization by pinching the nostrils. The attack detectors were based on two paradigms: acoustic features with GMM and SVM classifiers; and tracking of MFCC and pitch contours. Robustness improved by fusing the speaker and attack detectors.

Addressing the issues above is crucial to apply SV technology in the real world. In a wide range of applications–from forensic to banking–, we need to assess whether SV decisions are reliable since mistakes can imply large losses. To improve reliability, the statistical models have to be optimized to the task at hand. Novel modeling techniques should train models as much general as possible or be able to adapt models between domains needing minimal resources. Finally, countermeasures against attacks would improve acceptance of voice biometrics. Common resources and evaluation protocols are necessary to advance in this field and foster collaboration between institutions.

Resumen y Conclusiones

Generalmente, el rendimiento de los sistemas de reconocimiento de locutor se mide en condiciones ideales, donde la voz es relativamente limpia, los usuarios son colaborativos, y hay suficientes datos disponibles para entrenar modelos de locutor y de canal. No obstante, cuando aplicamos verificación de locutor en entornos reales nos enfrentamos a retos que merecen investigación en profundidad. Esta tesis aborda algunos de ellos.

El rendimiento de los sistemas de verificación puede empeorar debido a múltiple causas: ruido, reverberación, idiomas o canales distintos de los utilizados en la fase de desarrollo, etc. De este modo, saber que podemos confiar en las decisiones del sistema de verificación es fundamental. Esto nos motivó a estudiar formas de estimar la fiabilidad de las decisiones. Trabajamos en soluciones basadas en redes bayesianas. Las redes modelan las relaciones de causalidad entre la fiabilidad de la decisión, el *score* del sistema de verificación, y un conjunto de medidas de calidad calculadas sobre las señales de inscripción y test del *trial*. La contribución más importante de esta tesis, en este aspecto, es una red bayesiana que describe cómo varían las distribuciones de *scores* cuando los audios del *trial* están distorsionados. Esta red supone que existen dos *scores*: uno observado y otro oculto. El *score* observado o *score ruidoso* es el que se obtiene del sistema de verificación, mientras que el *score* oculto o *score limpio* es un score ideal que se obtendría si se tuviese voz de alta calidad. A partir de la distribución a *posteriori* del *score limpio*, se puede inferir la fiabilidad del *trial*. Esta aproximación consiguió mejores resultados que métodos previos.

Actualmente, el paradigma de los i-vectors es el estado del arte en verificación de locutor superando, en la mayoría de tareas, al *joint factor analysis* o los *support vector machines* (SVM). Por esta razón, otra de las partes de la tesis se enfocó en problemas que aparecen cuando se modelan las distribuciones de i-vectors. Se consideró el problema de tener simultáneamente i-vectors grabados en diferentes condiciones como múltiples tipos de canal, tipos o niveles de ruido. Introdujimos una variante de *probabilistic discriminant analysis* (PLDA) que intenta aproximar este problema de manera teóricamente correcta permitiendo que existan diferente distribuciones de canal para cada condición. Otro problema que se abordó fue como tener en cuenta la incertidumbre acerca de los parámetros del modelo que existe cuando la cantidad de datos de desarrollo es pequeña en comparación con la dimensión de los i-vectors. Se propuso calcular la distribución a *posteriori* del modelo dados los datos de desarrollo, y después usar esta distribución para evaluar los ratios de verosimilitud integrando los parámetros del modelo. Con este método se obtuvo una mejora significativa con i-vectors sin normalización en longitud. Finalmente, se abordó el problema de entrenar PLDA en dominios con escasos datos de desarrollo. Se propuso adaptar un modelo entrenado para otro dominio, con datos suficientes, al dominio de interés usando *Maximum a posteriori.*

En la última parte de la tesis, nos interesamos por los ataques a sistemas de verificación de locutor. Se trabajó en detectar ataques de *spoofing* y *tampering*. Ambos ataques tienen intenciones opuestas. Mientras que el *spoofing* intenta que un impostor suplante la identidad del usuario bajo test, el *tampering* intenta ocultar la identidad del locutor para no ser detectado. Nos enfocamos en ataques de baja tecnología, los cuales pueden ser llevados a cabo por cualquier criminal sin necesidad de tener conocimientos de procesado de voz. En el caso del *spoofing*, se estudiaron ataques basados en grabar la voz del usuario y reproducirla sobre el sistema. En sistemas dependientes del texto, el ataque también implica crear la contraseña cortando y pegando extractos de varias grabaciones de la víctima. En cuanto al *tampering*, se estudiaron ataques basados en cubrir la boca con la mano o con un pañuelo; y denasalización pellizcando las fosas nasales. Los sistemas para detectar los ataques estuvieron basados en características acústicas y clasificadores *Gaussian mixture models* y SVM; y seguimiento de contornos de *Mel filtered cepstral coefficients* y *pitch*. La fusión del verificador de locutor con los detectores de ataques mejoró la robustez del sistema.

Abordar los asuntos arriba descritos es crucial para poder aplicar verificación de locutor en el mundo real. En un amplio rango de aplicaciones –desde forenses a banca–, se necesita evaluar si las decisiones del verificador de locutor son fiables dado que los errores pueden acarrear grandes pérdidas. Para mejorar la fiabilidad, los modelos estadísticos deben estar optimizados para la tarea en cuestión. Los nuevos métodos de modelado deberían entrenar modelos lo más generales posible o ser capaces de adaptar modelos entre dominios necesitando para ello los mínimos recursos. Finalmente, las medidas contra ataques mejorarían la aceptación de los sistemas de biometría de voz. Se necesitan recursos y protocolos de evaluación comunes para poder avanzar en este campo y fomentar la colaboración entre instituciones.

Contents

List of Figures

List of Tables

List of Acronyms

BN Bayesian Network

CP Cut and Paste Spoofing Attack

CM Confidence Measure

CMS Cepstral Mean Subtraction

CMVN Cepstral Mean and Variance Normalization

DCF Detection Cost Function

DCT Discrete Cosine Transform

DET Detection Error Trade-off

DTW Dynamic Time Warping

EDC Energy Decay Curve

EER Equal Error Rate

EM Expectation Maximization

ETSI European Telecommunications Standards Institute

FFT Fast Fourier Transform

GD Gender Dependent

GI Gender Independent

GMM Gaussian Mixture Model

GSV Gaussian Supervector

H Entropy

HMM Hidden Markov Model

HP High Pass

I3A Instituto de Investigación en Ingeniería de Aragón (Aragon Institute for Engineering Research)

ITU International Telecommunication Union

JFA Joint Factor Analysis

LDA Linear Discriminant Analysis

LFR Low Frequency Energy Ratio

LHS Left Hand Side

LLK Log-likelihood

LLR Log-likelihood Ratio

LP Low Pass

LPC Linear Prediction Coefficients

MAP Maximum A Posteriori

MCPLDA Multi-channel PLDA

MD Minimum Divergence

MFCC Mel Frequency Cepstral Coefficients

MI Modulation Index

ML Maximum Likelihood

MSE Mean Square Error

NAP Nuissance Attribute Projection

NIST National Institute of Standards and Technology

PLDA Probabilistic Linear Discriminant Analysis

PLP Perceptual Linear Prediction

RAPT Robust Algorithm for Pitch Tracking

RASTA Relative Spectral

RHS Right Hand Side

RIR Room Impulse Response

SI Speaker Identification

SNR Signal-to-Noise Ratio

SR Spectral Ratio

SRE Speaker Recognition Evaluation

SV Speaker Verification

SVM Support Vector Machine

TEO Teager Energy Operator

UBM Universal Background Model

VAD Voice Activity Detector

VB Variational Bayes

VTS Vector Taylor Series

WCCN Within-class Covariance Normalization

Chapter 1

Introduction

1.1 Introduction

The field of biometrics intends to identify living subjects based on their physiological or behavioral patterns. There are many types of biometric traits like fingerprint, iris, DNA, gait, signature, face, hand geometry and voice. Some of them, like DNA, depend on the physiological characteristics of the individual; others, like signature, depend on behavioral characteristics; they may also depend on both. The interest in biometric applications has considerably grown in the last few years because many fields require higher levels of security. The threat of terrorism is one of the reasons but not the only one. Because of the recent advances made in information technology, a wide range of new services has emerged (electronic banking, email, social networks). All these services need reliable ways to authenticate their users. Currently, users must remember dozens of passwords and personal identification numbers (PIN). Evidently, using the same password for every service where we sign up is risky. Besides, passwords can be forgotten or stolen by different means. Biometrics could solve these problems if we make them reliable enough.

This thesis deals with the biometric modality known as speaker recognition. Speaker recognition is the ability of recognizing people by the characteristics of their voices. Both, the anatomy of the individuals and their behavioral patterns influence the properties of speech. On the one hand, people's voices depend on the shape of their vocal tract, larynx size and other voice production organs. On the other hand, each speaker has his *manner of speaking* that includes the use of a particular accent, rhythm, intonation style, pronunciation pattern and vocabulary [Kinnunen and Li, 2010].

Speaker recognition can fundamentally refer to two tasks [Reynolds, 2002]: verification and identification. Speaker verification (SV) determines whether a person is who he or she claims to be. In this case, possible impostors are unknown to the system, and it is called an *open-set* task. Speaker identification (SI) decides who is talking among a known group of speakers. In *closed-set identification*, the speaker under test must be one of the known speakers. On the contrary, in *open-set identification*, the speaker in the test segment may be none of the known speakers. Then, the task includes identification and verification together.

Speech is a natural means of communication so customers do not consider it intrusive and they accept it better than other biometric modalities. This fact favors the existence of many applications for speaker recognition. Moreover, nowadays the ubiquity of cell phones creates a perfect scenario for telephone based voice biometrics.

Forensics is one of the main application areas for speaker recognition. Speech can be an evidence in court to convict or discharge a defendant of a crime. The interest for applying speaker recognition as aid for forensic experts has increased in the last years [Gonzalez-Rodriguez et al., 2003, Pfister and Beutler, 2003, Alexander et al., 2004, Niemi-Laitinen et al., 2005, Gonzalez-Rodriguez et al., 2007, Campbell et al., 2009]. *Surveillance* is a related area. Law enforcement agencies search information by eavesdropping of a large number of telephone conversations. To deal with this huge volume of data, they need automatic means of analysis. Speaker recognition allows us to look for a specific criminal in this ocean of data and find out what he is planning [Marchetto et al., 2009].

Other areas of application are *identity authentication* and *access control*. Speaker recognition can control the access to physical facilities [Gupta et al., 2005], computer networks, web services or telephone resetting of passwords [Roberts, 2002]. It is especially important to increase safety of telephone banking transactions, electronic banking and e-commerce; fields that have experienced an important growth in recent years.

Currently, there is an increasing production of documents with spoken information (TV broadcast, conference meetings, Internet videos, etc.). To access all these documents in an efficient manner, we need to classify them and index them. Thus, we need automatic means of extracting meta-data like topics of discussion, participant names and so on. Speaker recognition technology can determine the turns of the speakers in conversations, this is known as *speaker diarization* [Reynolds et al., 2009].

Finally, another application of speaker recognition is *personalization*. By knowing the users' identity a computer application can learn their personal preferences for a service or device (radio settings, preferred TV channel, air-conditioning temperature, etc.). Then, when the application identifies the user it can load automatically the optimum settings. For example, ambient speech is used to know who is sitting where inside a car and personalize the experience of driver and passengers [Feld et al., 2010].

Speaker recognition systems can be classified into text-dependent or text-independent depending on the application they are intended for. In a text-dependent application, the user must utter a specific sentence requested by the system. This sentence can be fixed or change for every access. In a text-independent application, the recognizer has no knowledge of the content of the spoken utterance. Text-dependent systems constrain the speech used in the enrollment and test phases to be the same phrase, phrases or a small vocabulary set (digits). This has the advantage of achieving better performance with smaller amounts of enrollment and test data. The users of text-dependent systems have to be cooperative so they are appropriate for authentication applications. On the other hand, text-independent systems are unconstrained so they accept a wider range of applications. They are especially interesting for applications where the user is not cooperative (forensics, diarization) or where the user is doing other speech based interactions like using a speech recognition system.

Following, we give a brief review of the evolution of speaker recognition technology.

1.2 Brief Historical Evolution

The first works on speaker recognition date from the decade of the seventies. The first approaches [Atal, 1974] were text-dependent and used cepstral or linear predictor (LP) features as input for nearest neighbors (NN) classifiers. Later, dynamic time warping (DTW) starts to be employed for aligning the test and reference patterns [Furui, 1981]. In the eighties, there were some initial attempts to develop text-independent speaker recognition, but yet with high error rates. In [Schwartz et al., 1982], short-term spectral features are modeled with several density distributions. In [Higgins and Wohlford, 1986], the authors divide the enrollment utterances into short segments and derive a set of templates by k-means clustering, and then the test utterance is matched with the templates.

In the nineties, HMM-based systems started to replace those based on DTW [Reynolds and Carlson, 1995] for text-dependent applications. For text-independent recognizers, the distribution of short-term spectral features started to be modeled using Gaussian mixture models (GMM) [Reynolds, 1995, Reynolds and Rose, 1995]. This approach, which is still in use, meant a significant improvement. Since 1996, under the National Security Agency (NSA) founding, the National Institute of Standards and Technology (NIST) has been conducting periodic evaluations of speaker recognition systems with the goal of determining the state-of-the-art of the technology [Martin and Przybocki, 2001]. NIST provides a common protocol and performance measure to evaluate text-independent SV systems [Doddington, 2000]. Mostly, NIST evaluations have driven the efforts of the speaker recognition community to improve performance in situations with large inter-session variability. Thus, evaluations include trials with matched and unmatched conditions between enrollment and test segments. The main sources of inter-session variability in NIST datasets are channel effects (different telephone handsets, transmission channels, far-field microphones) or language [Przybocki et al., 2007].

In 2000, the GMM-UBM approach [Reynolds et al., 2000], in which speaker models are adapted from a generic world model, became popular. This approach has been the basis for most speaker recognition systems in the last decade. Reynolds' approach was still sensitive to the effects of inter-session variability. Since then, researchers have made a great effort to make the GMM-UBM approach robust to these effects. Some methods like feature mapping [Reynolds, 2003] and feature warping [Pelecanos and Sridharan, 2001] do feature-level session compensation. Other methods try to compensate at the model level, the most successful techniques had been SVM-GMM with *nuisance attribute projection* (NAP) [Campbell et al., 2006b] and *joint factor analysis* (JFA) [Kenny et al., 2007b]. Lately, a new approach known as *i-vectors* (identity vectors) has attracted a great attention [Dehak et al., 2011b]. This approach extracts a fixed length vector from the speech segment; and this vector is taken as feature for advanced pattern recognition algorithms [Brummer and De Villiers, 2010, Kenny, 2010].

1.3 Current Challenges

NIST evaluations have motivated researchers to cope with session mismatch effects. As a consequence, other issues have received less attention. This thesis focuses on some of those issues. First of all, NIST databases are rather clean. In real applications, we find signals with background noise, reverberation and artifacts (telephone tones, laughs, saturation). There are some works that utilize quality measures to improve speaker recognition. In [Solewicz and Koppel, 2005] and [Garcia-Romero et al., 2006], quality measures improve the fusion of SV systems. In [Harriero et al., 2009], several quality measures are used to predict the reliability of a speaker recognition system. The work in [Richiardi et al., 2006b] presents a method to estimate the reliability of speaker verification decisions with a Bayesian framework. However, more research needs to be done to know which factors affect speaker recognition performance and how to use quality measures to improve it.

Another issue to consider is the security attacks to SV systems. Attacks can be classified into two categories: spoofing or forgery and tampering. Spoofing is the fact of impersonating another person by using different techniques like voice transformation or playing of a recording of the victim. On the other hand, tampering consists in altering the voice to prevent being detected by SV. There are multiple techniques for voice disguise. In [Perrot et al., 2007], the authors do a study of voice disguise methods and classify them into electronic transformation or conversion; imitation; and mechanical and prosodic alteration. In [Figueiredo and Britto, 1996], the effects of speaking while grasping a pencil in the teeth were studied. In [Perrot et al., 2005], an impostor voice was transformed into the target speaker voice by a voice encoder and a decoder. More recently, in [De Leon et al., 2010b] a HMM based speech synthesizer with models adapted from the target speaker deceived a SV system.

Finally, it is well known that current approaches like JFA or NAP greatly depend on the type of the development data [Vaquero et al., 2009]. For instance, a JFA system developed with telephone data will be able to compensate inter-session variability due to different telephone handsets or transmission channels but it will perform poorly on speech from far-field microphones. The same will happen with other types of inter-session variability not considered during the development process. Making the system less sensitive to the development data is something that needs extra attention.

1.4 Thesis Organization

This thesis is organized into five parts related to different problems. The first part provides an overview of the evolution of text-independent speaker recognition technology during the last decade. We evaluate the systems implemented by our group for the NIST SRE 2006-2010 [Villalba et al., 2008, Villalba et al., 2010] on a common dataset (NIST SRE 2010). Thus, we evidence the great leap that the technology has given in the last years.

In the second part, we address the issue of estimating the reliability of the speaker verification decisions. A probabilistic reliability measure is computed from a group of quality measures, extracted from the speech segments involved in the trial. This measure allows us to discard unreliable trials in applications that require very accurate decisions but that do not need a decision for all the trials. In Chapter 3, we introduce the state-of-the-art and describe our experimental setup. In Chapter 4, we explain the quality measures that we extracted from the speech segments and show the correlation between this measures and SV performance. In Chapter 5, we revisit a state-of-the-art approach based on Bayesian networks to estimate the SV reliability. We propose some modifications of the BN and prove that performance improves by rejecting unreliable trials. In Chapter 6, we introduce a novel approach that clearly outperforms previous works. It is also based on a Bayesian network but with a different philosophy. The network models how speech distortions, such as noise and reverberation, modify the SV score distributions. The network can be applied to reject unreliable trials but also to obtain an improved SV likelihood ratio. The improved ratio reduces error rates without discarding trials. Datasets are affected by different distortions so a BN trained on one dataset may not perform well on another. In Chapter 7, we show how to adapt the BN presented in the previous chapter from one dataset to another by *maximum a posteriori* (MAP) estimation.

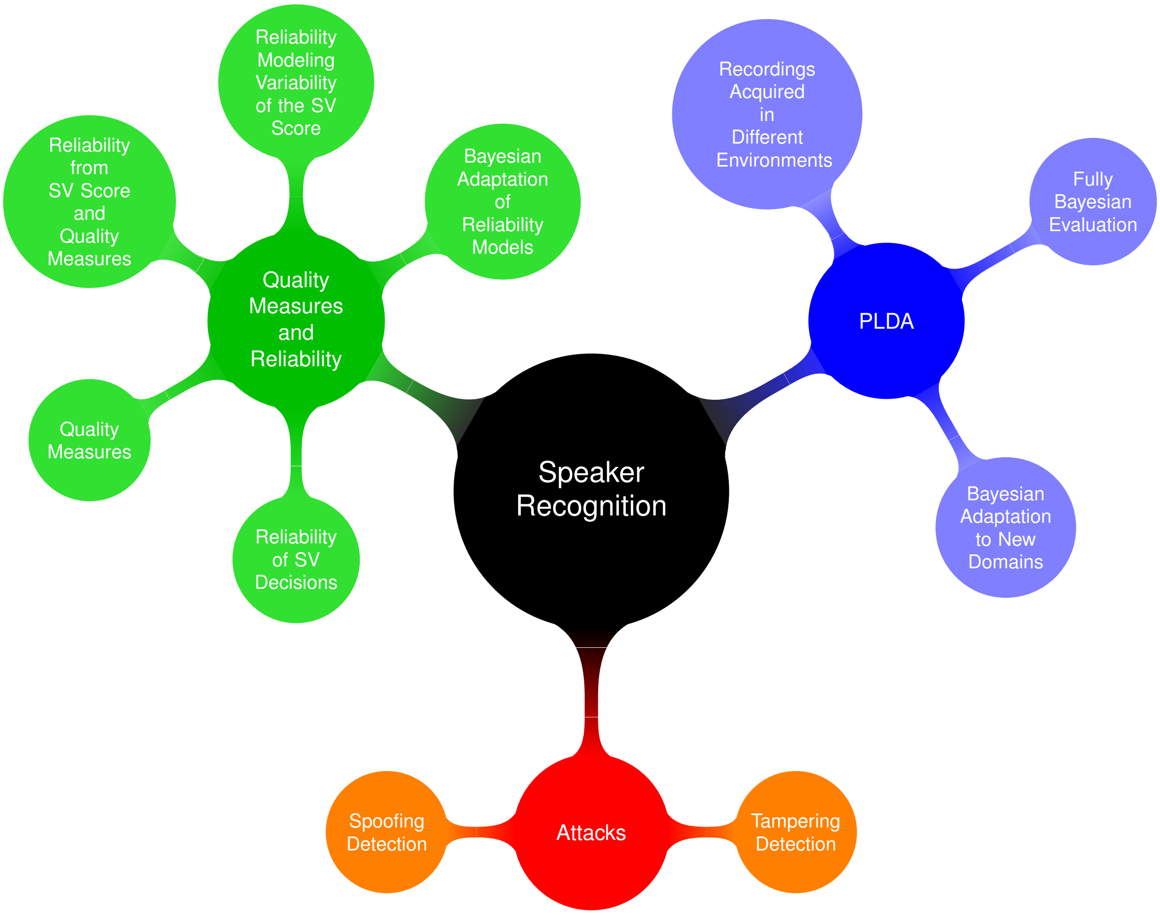

Reliability Modeling Variability of the SV Score
Reliability from SV Score and Quality Measures
Bayesian Adaptation of Reliability Models
Quality Measures and Reliability
Quality Measures
Reliability of SV Decisions
Recordings Acquired in Different Environments
Fully Bayesian Evaluation
PLDA
Bayesian Adaptation to New Domains
Speaker Recognition
Spoofing Detection
Attacks
Tampering Detection

Figure 1.1: Mind map of the thesis.

In the third part, we adopt the paradigm of speaker recognition based on i-vectors and a probabilistic linear discriminant analysis (PLDA) back-end. We propose several modifications of the standard PLDA to address different issues. In Chapter 8, we introduce Multi-channel PLDA, a PLDA variant where the prior distribution of the inter-session variability term depends on the channel type in which the speech was recorded. In Chapter 9, we implement fully Bayesian evaluation of PLDA. The Bayesian approach, instead of taking a point estimate of the model parameters, computes their posterior distribution given the development data. Hence, when we evaluate the likelihoods given the target and non-target hypothesis, the model posterior is employed to integrate out the model parameters. When the amount of development data is large compared with the size of the model, the posterior is sharply peaked and the standard and Bayesian evaluations provide similar results. On the contrary, if the amount of training data is small, the posterior is flatter; i.e., meaning that the uncertainty about the value of the model parameters is large. By integrating out the model, we take into account that uncertainty and, as a consequence, standard and Bayesian evaluations diverge. If the posterior of the model is wide, it has the side effect of preventing database mismatch. In Chapter 10, we propose to alleviate database mismatch by MAP adaptation of the PLDA parameters from one dataset with large amount of development data to another with scarce data.

In the fourth Part, we treat the problem of security attacks to SV systems. We focus on low effort attacks, which are the ones available to average criminals not counting with advanced knowledge of speaker recognition technology. Chapter 11 deals with spoofing attacks. Spoofing consists in impersonating a legitimate user. We experimented with two types of spoofing: replay attack and cut and paste. Reply attack consists in playing a recording of the victim on the microphone by using a loudspeaker. This attack affects text independent speaker recognition. Cut and paste attacks are for text-dependent systems. The phrase requested by the recognizer is built by concatenating words from several recordings. In Chapter 12, we deal with tampering that consists in altering one's voice for not being detected by SV. We worked on two types of alterations: covering the speaker mouth with the hand or a handkerchief; and nasalization. Nasalization refers to pinching the speaker's nose to alter the resonances of the vocal tract. We will show how SV performance of state-of-the-art speaker verification degrades with these attacks and propose methods to detect them.

Finally, in the fifth part, we offer some global conclusions of the thesis.

1.5 Notation

1.5.1 Mathematical Notation

Here, we introduce the mathematical notation used in this thesis. We distinguish different types of variables:

- Scalars are denoted by italic lowercase letters, e.g., x.
- Column vectors are denoted by roman boldface lower case letters, e.g., $\mathbf{x}$.
- Matrices are denoted by roman boldface upper case letters, e.g., $\mathbf{X}$.
- The superscript T denotes the transpose operator so $\mathbf{X}^T$ is the transpose of $\mathbf{X}$ and $\mathbf{x}^T$ is a row vector.
- Sets are denoted by italic uppercase letters, e.g., X.

Regarding probabilities, we use P to denote both probability mass functions and probability density functions. We use q to denote approximate posterior distributions like those used in variational Bayes algorithms. $P\left(\mathbf{X}|\mathbf{Y}\right)$ denotes the conditional distribution of the variable $\mathbf{X}$ given $\mathbf{Y}$. $P\left(\mathbf{X},\mathbf{Y}\right)$ denotes the joint distribution of the variables $\mathbf{X}$ and $\mathbf{Y}$. Taking expectations is very much needed for EM and variational Bayes algorithms. $\mathrm{E}_{\mathbf{Y}}\left[\mathrm{f}(\mathbf{X},\mathbf{Y})\right]$ denotes the expectation of $\mathrm{f}(\mathbf{X},\mathbf{Y})$ given the distribution of $\mathbf{Y}$:

$$\mathrm{E}_{\mathbf{Y}}\left[\mathrm{f}(\mathbf{X},\mathbf{Y})\right] = \int \mathrm{f}(\mathbf{X},\mathbf{Y}) P\left(\mathbf{Y}\right)\, \mathrm{d}\mathbf{Y} \tag{1.1}$$

where f is a generic function.

1.5.2 Graphical Models

Along this thesis, we extensively use diagrammatic representations of probability distributions called *probabilistic graphical models* [Bishop, 2006]. These provide some useful advantages:

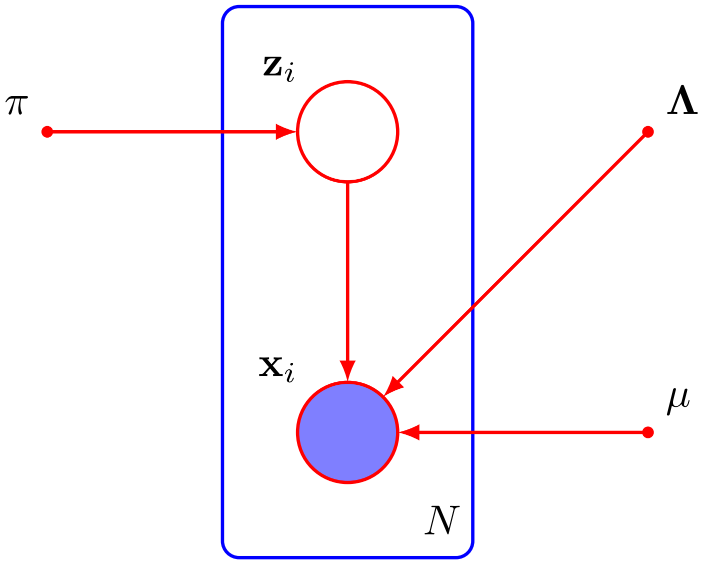


Figure 1.2: BN for a mixture of Gaussians.

- They allow us to visualize the structure of the probabilistic model and can be used to design and motivate new models.

- By inspecting the graph, we can infer the properties of the model, including the conditional dependencies between variables.

- Complex computations, required to perform inference and learning in sophisticated models, can be expressed in terms of graphical manipulations, in which underlying mathematical expressions are carried along implicitly.

The graph comprises nodes and links or arcs. Each node represents a variable or group of variables and the links express probabilistic relationships between variables. The graph facilitates to decompose the joint distribution over all the random variables into a product of factors each depending only on a subset of the variables.

We restrict ourselves to employ a particular case of graphical models called Bayesian networks (BN) or *directed graphical models*. In directed models the links have a particular direction indicated by arrows and they are useful to express causal relationships between the random variables. Looking at the graph, we can easily determine if two variables are or not independent given another set of variables. The rules to do it are summarized as follows [Brummer, 2010a]:

- Two variables, a and b, are conditionally independent given another set of variables C, if all paths on the graph between a and b are blocked.

- A path is blocked if any node on the path is blocked.

- A node is blocked if either:

  - Arrows on the path meet head-to-tail, or *tail-to-tail* at the node and the variable at the node is in C.

  - Arrows on the path meet *head-to-head* at the node and neither the node, nor any of its descendants–a descendant of a node c is any node that can be reached from c by following the arrows–are in C.

Thus, if a and b are independent given C, we can write $P(a, b|C) = P(a|C)\,P(b|C)$ and $P(a|C, b) = P(a|C)$.

Figure 1.2 shows an example of a Bayesian network representing a mixture of Gaussians. Empty nodes denote *hidden variables*, shaded nodes denote *observed variables* and small solid nodes denote *deterministic parameters*. A node or group of nodes surrounded by a box, called a *plate*, and labeled with N indicates that there are N nodes of that kind. In the example, $\mathbf{x}_i$ are the observed data samples and $\mathbf{z}_i$ are the hidden variables indicating which Gaussian generates each data sample. The plate indicates that we observe N data samples. The deterministic variables μ and $\mathbf{\Lambda}$ are the means and precisions of the Gaussians, which can be given or estimated by maximum likelihood.

Part I

Speaker Recognition

Chapter 2

Speaker Recognition Technology

2.1 Introduction

Speaker recognition is a biometric modality based on recognizing people by their voices. Human voice is influenced by the physiological features of the speech production organs (vocal tract, larynx mouth, nasal cavity, etc.) and behavioral patterns (accent, rhythm, intonation style, choice of vocabulary). Speaker recognition systems can perform two kinds of tasks: *speaker verification* (SV) and *speaker identification* (SI). Speaker verification consists in determining whether a person is the one that he or she claims to be. In this task, the system needs to be able to cope with unknown impostors (open-set task). On the contrary, speaker identification decides who is talking among a known group of speakers (closed-set task). In this thesis, we focus on the verification task.

This chapter presents a review of the main milestones of text-independent speaker verification technology along the last decade, from MAP adapted Gaussian mixture models to i-vectors. Some of the paradigms reviewed here were part of the I3A submissions to NIST SRE 2006–2010 [Villalba et al., 2008, Villalba et al., 2010]. At the end of the chapter, we compare their performance on the core condition of NIST SRE 2010.

2.2 Speaker Verification Systems

Figure 2.1 illustrates the components of most speaker verification systems. They consist of five blocks: feature extraction, statistical modeling, score normalization, calibration and decision taking. The feature extraction module transforms the raw speech samples into feature vectors appropriate for classification. Employing statistics, we study the feature distributions and create speaker and inter-session variability models that we use later to enroll new speakers and evaluate the SV trials. The scores generated by the evaluator are normalized and calibrated to obtain meaningful likelihood ratios. Finally, we apply a threshold to the likelihood ratios to make a decision.

As in other biometric modalities, a speaker recognition system has two phases of operation: *enrollment* phase and *test* phase. During the *enrollment* phase, the voice of the target speaker is recorded and employed to create a statistical model. During the *test* phase, a new speech segment is compared to the enrollment models to make a decision about the speaker's identity. Besides these two phases, we can consider a third phase, known as *development* phase. In this phase, carried out during the implementation of the system,

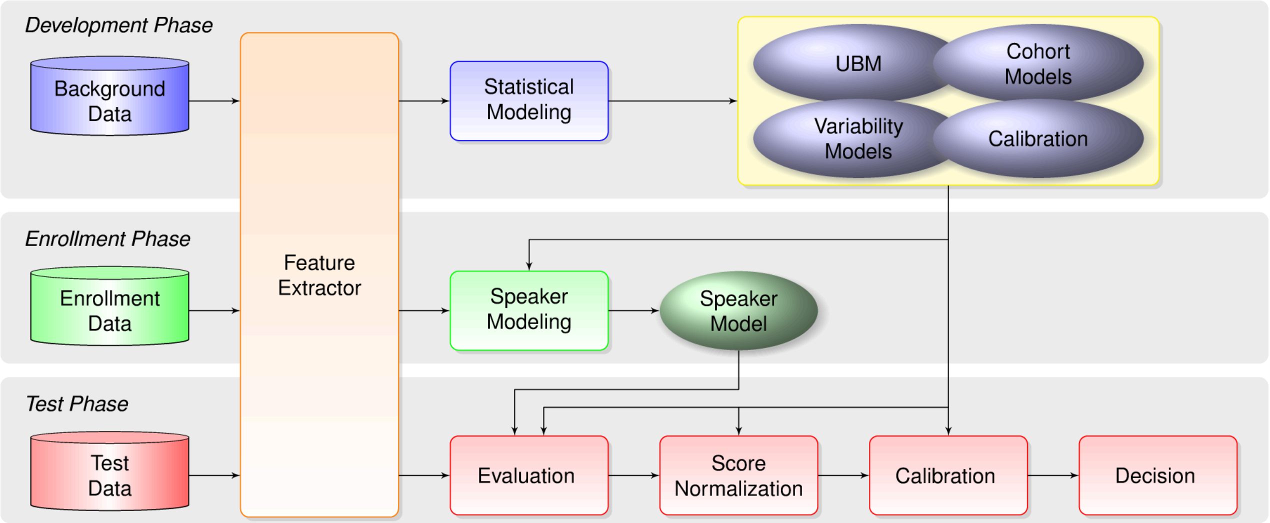


Figure 2.1: Components of a typical automatic speaker recognition system.

several tasks are accomplished: creating Universal Background Models (UBM), groups of impostor models, estimating models of inter-session variability, training of the score calibration function, setting up the optimum decision threshold for the intended application and so on.

The following sections describe each of the components of Figure 2.1 in detail.

2.3 Feature Extraction

The feature extraction module transforms the raw speech samples into feature vectors suitable for classification algorithms. Features for speaker recognition must retain the information about the speaker identity while they remove redundancy. There are different types of features containing identity information. According to their physical meaning we divide them into [Kinnunen and Li, 2010]: short-term spectral features, voice source features, spectro-temporal features, prosodic features and high-level features. The spectral features characterize the resonances of the vocal tract. The voice source features describe the glottal flow. Spectro-temporal and prosodic features capture intonation and rhythm. Finally, high level features capture particular word usage (idiolect), related to learned habits, dialect and style [Reynolds et al., 2003]. Features related to the short-term spectrum of the speech signal are the ones primarily used for speaker recognition. Besides being easy to calculate, they produce the best performance. However, systems based on other types of features, can improve the performance of spectral system by fusion [Campbell et al., 2007].

2.3.1 Short-term spectral features

There are several variants of short-term spectral features. Most of them are inspired in the way humans perceive sound. Mel Filtered Cepstral Coefficients (MFCC) [Davis and Mermelstein, 1980] are the most widespread. Figure 2.2 shows the block diagram to compute them.

The speech signal is divided into 25 msec. short frames with 10 msec. overlap. Within that interval, the signal is assumed stationary. One by one, the frames are pre-emphasized

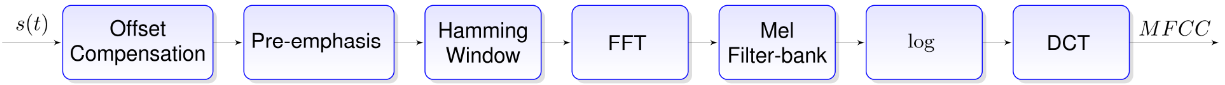


Figure 2.2: Block diagram of MFCC feature extraction.

and multiplied by a window function before spectral analysis. The pre-emphasis amplifies the high frequencies, whose intensity is much lower, and thus, it equilibrates the influence of high and low frequencies. Windowing mitigates the effects of the finite duration of the frames on the Fourier transform [Oppenheim et al., 1999]. Usually, the Hamming window is chosen. Following, the magnitude spectrum is computed with the fast Fourier transform (FFT). This representation is better than the raw speech samples to observe the resonances of the vocal tract.

The filter-bank integrates the energy present in several bands (usually 20–40). The triangular Mel filters are placed following a logarithmic frequency scale motivated by the behavior of the human auditory system [Davis and Mermelstein, 1980]. The dynamic range of the filter output is compressed by a logarithmic function. Finally, a discrete cosine transform (DCT) is applied and the first 12–20 coefficients are retained. The DCT helps to decorrelate the coefficients of the feature simplifying the modeling. Other popular features are usually modifications of these basic system, e.g., the advanced ETSI Front-End [ETSI, 2007].

Alternative features with similar properties are Linear Predictor Cepstral Coefficients [Gheorghe and Anton, 2008] and Perceptual Linear Prediction (PLP) [Hermansky, 1990]. The main different with respect to MFCC is that they use linear prediction, instead of the Fourier transform, to estimate the spectrum.

The modeling techniques currently used for speaker recognition assume that the frames are statistically independent. It is well known that this is not actually true given that there is a strong correlation between close frames. To include the temporal information, first and second order time-derivatives are appended to the feature vector. These derivatives are usually referred as deltas and double deltas or dynamic features [Furui, 1986].

2.3.2 Alternative features

There are features can add complementary information to the spectrum: voice source features, spectro-temporal features, prosody, high-level features and so on. Voice source features characterize the glottal pulse. We cannot measure them directly because of the filtering effect of the vocal tract. However, assuming that the glottal source and the vocal tract are mutually independent, vocal tract parameters are estimated by linear prediction and, then, the glottal pulse is recovered by inverse filtering of the original waveform [Kinnunen and Alku, 2009, Murty and Yegnanarayana, 2006, Prasanna et al., 2006, Zheng et al., 2007]. From the glottal pulse signal, we can extract several features: cepstral coefficients [Kinnunen and Alku, 2009], wavelet analysis [Zheng et al., 2007], residual phase [Murty and Yegnanarayana, 2006], features based on neural networks [Prasanna et al., 2006], higher-order statistics [Chetouani et al., 2009], etc.

Spectro-temporal features capture the variations of the vocal tract resonances along time. Modulation frequency was proposed for this purpose [Kinnunen, 2006,Kinnunen et al., 2008].

It represents the frequency content of the sub-band amplitude envelopes, and, in theory, it contains information about the speaking rate and other stylistic attributes. In [Kinnunen et al., 2006], a temporal DCT was applied on the trajectories of cepstral coefficients rather than on the spectral envelopes. The improvement achieved by fusing spectro-temporal and short-term spectral features was rather modest.

Prosody refers to aspects of speech like intonation patterns, speaking rate, rhythm and syllable stress. Prosody is a characteristic that, contrary to short-term features, spans over long segments like syllables, words or sentences. Common prosodic parameters are the pitch and energy contours of the segment. In [Dehak et al., 2007], they are successfully modeled by Legendre polynomials and factor analysis. More recently, in [Kockmann et al., 2011], speech segments are automatically split into syllables and for each syllable and its nucleus, they extract minimum, maximum, mean and slope of pitch and energy trajectories, as well as the durations of onset, nucleus and coda. From this set of features, they extract i-vectors by multinomial subspace models and classification is implemented by PLDA outperforming previous works. In these works, the fundamental frequency is commonly estimated by variants of the RAPT algorithm [Talkin, 1995].

Finally, high-level features refer to the speaker's lexicon. High-level features are sequences of discrete tokens that can correspond to words [Doddington, 2001], phones [Campbell et al., 2004], and prosodic events [Shriberg et al., 2005], among others. Sometimes, several phone recognizers trained on different languages are used to produce parallel token sequences [Ma et al., 2006]. The token sequences are classified by N-grams models.

2.3.3 Voice activity detection

A fundamental part of the feature extraction is *voice activity detection* (VAD). Removing non-speech frames is critical to achieve optimum performance. In the I3A submissions to NIST SRE 2008–2012 [Villalba et al., 2008, Villalba et al., 2010, Villalba et al., 2012] a VAD based on the *long-term spectral divergence* was used [Ramirez et al., 2004]. That solution is robust to noise types and levels. Other approaches are presented in [Mak and Yu, 2010] and [Sadjadi and Hansen, 2013].

2.3.4 Feature normalization

Intending to mitigate the channel distortions, a large amount of feature normalization techniques have been adopted. The simplest is *cepstral mean subtraction* (CMS). It takes advantage of the fact that MFCC is approximately homomorphic for filters that have a smooth transfer function. Convolutional effects in time domain become additive in cepstral domain. The mean value of the features over the entire speech segment is considered an approximation of the channel component. CMS subtracts the mean from each frame [Bimbot et al., 2004] to reduce the channel mismatch between utterances. *cepstral mean and variance normalization* (CMVN) extends CMS by normalizing the features by their standard deviation [Alam et al., 2011]. *Feature warping* [Pelecanos and Sridharan, 2001, Xiang et al., 2002] equalizes the feature distribution to match a Gaussian of zero mean and unit variance. Each feature is warped based on the histogram in a window of a few seconds around it. Previous methods are applied after silence removal to include only speech in the calculus of the distributions. *Relative Spectral* (RASTA) filtering [Hermansky and Morgan, 1994] applies a band pass filter along the temporal trajectory of each cepstral coefficient. The filter eliminates modulation frequencies unusual in speech signals. For example, low varying channels appear as low frequencies in the modulation spectrum.

The methods above do not need any channel models. On the contrary, *feature mapping* maps feature vectors into a channel independent space [Reynolds, 2003] by applying transformations learned on channel labeled data. Channel dependent GMMs are MAP adapted from a channel independent model. For each feature, the most likely channel is detected and we apply the mapping given by the relation between the channel dependent and independent GMM.

2.4 Statistical Modeling

In this section, we present the most popular models for speaker recognition in the last decade. They are mostly intended to model short-term spectral features such as MFCC, in which the frames are assumed to be independent and identically distributed (i.i.d.). However, some of them have been successfully applied on other types of features [Dehak et al., 2007].

2.4.1 Gaussian mixture models

In 1995, Reynolds introduced the use of Gaussian mixture models (GMM) for speaker recognition [Reynolds, 1995, Reynolds and Rose, 1995]. Since then, it has become the reference method for this task and it is the basis for the most successful approaches that have emerged in the last years. A GMM is a probability density function that, for a feature vector $\mathbf{x}$, is defined as:

$$P\left(\mathbf{x}|\lambda\right) = \sum_{k=1}^{K} w_k \mathcal{N}\left(\mathbf{x}|\mu_k, \Sigma_k\right) \tag{2.1}$$

where K is the number of components of the mixture; the weights w_k satisfy the constraint $\sum_{k=1}^{K} w_k = 1$ and

$$\mathcal{N}\left(\mathbf{x}|\mu_k, \Sigma_k\right) = \frac{1}{\left|2\pi\Sigma_k\right|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_k)^T \Sigma_k^{-1}(\mathbf{x}-\mu_k)\right) \tag{2.2}$$

is the d-dimensional Gaussian distribution with mean μ_k and covariance matrix Σ_k.

Training a GMM means to estimate the parameters $\lambda = \{w_k, \mu_k, \Sigma_k\}_{k=1}^{K}$ given a collection of feature vectors $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$. These parameters can be estimated following a maximum likelihood criterion (ML) using the expectation-maximization (EM) algorithm [Bishop, 2006].

In the GMM-UBM approach [Reynolds et al., 2000] a *universal background model* (UBM) is trained by EM iterations with several hours of speech from a large number of speakers. The UBM represents the speaker independent distribution of features. Then, target speakers' GMMs are obtained by adapting the means μ from the UBM with a *maximum a posteriori* (MAP) criterion.

In the verification phase, the average log-likelihood ratio (LLR) between the probabilities

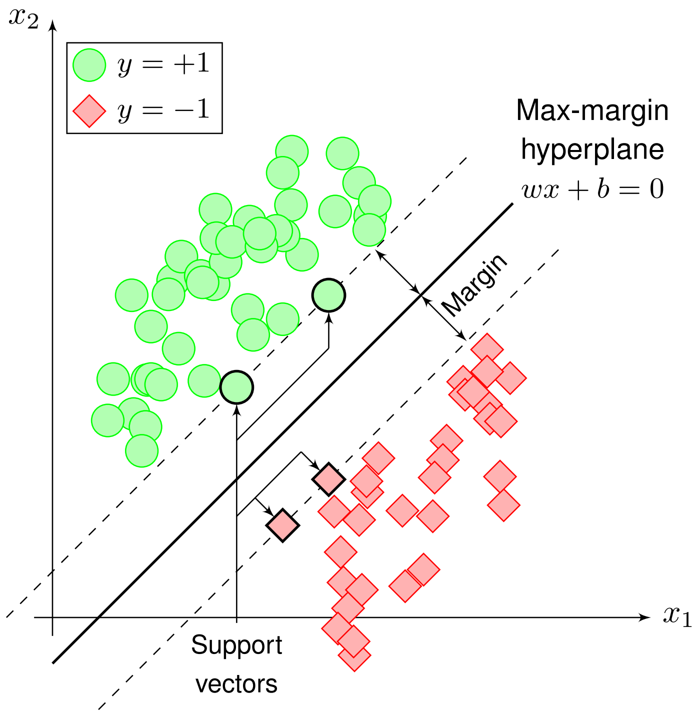


Figure 2.3: SVM classifier. The maximum margin hyperplane separates positive and negative examples.

of the test frames given the target and UBM models is computed:

$$\text{LLR} = \frac{1}{T}\sum_{t=1}^{T} \ln P\left(\mathbf{x}_t|\lambda_{\text{target}}\right) - \ln P\left(\mathbf{x}_t|\lambda_{\text{UBM}}\right) \tag{2.3}$$

The fact that all the speakers models are adapted from a common UBM makes the score ranges of different speakers comparable.

2.4.2 SVM-GMM

A *support vector machine* (SVM) is a binary classifier that separates two classes by a hyperplane in a high dimensional feature space [Cristianini and Shawe-Taylor, 2000]. In speaker recognition the classes are target speaker (+1) and impostor (-1). The hyperplane decision function can be written as,

$$f(\mathbf{x}) = \sum_{i=1}^{M} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b \tag{2.4}$$

where $f(\mathbf{x})$ represents the distance of the feature to the hyperplane. The $\mathbf{x}_i$ are the support vectors, α_i are the support vector weights and y_i are their corresponding labels such as $\sum_{i=1}^{M} y_i \alpha_i = 0$ and $\alpha_i > 0$. The support vectors together with their weights are obtained by a discriminative optimization algorithm from training data. The kernel function $K(\cdot,\cdot)$ must satisfy certain conditions (Mercer Theorem) [Cristianini and Shawe-Taylor, 2000] so that $K(\cdot,\cdot)$ can be expressed as an inner product $K(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})^T \phi(\mathbf{y})$ where $\phi(\mathbf{x})$ is a mapping from the input space to a higher dimensional space where the classes can be better separated by a hyperplane.

The optimization algorithm searches for the hyperplane that maximize the margin between both classes in the high dimensional space. The training points laying on the boundaries between classes are the support vectors as depicted in Figure 2.3.

To apply SVM on speaker recognition we need to define a kernel function that implicitly maps a speech utterance into a high dimensional feature of fixed length. One of the most successful approaches was presented in [Campbell et al., 2006a]. In this work, GMMs are adapted to each speaker utterance with the GMM-UBM paradigm. A kernel is derived by bounding the Kullback-Leibler (KL) divergence between two mixtures. Given two utterances utt_a and utt_b, the KL divergence kernel is defined as

$$K(\text{utt}_a, \text{utt}_b) = \sum_{k=1}^{K} \left(\sqrt{w_k} \Sigma_k^{-1/2} \mu_k^a \right)^T \left(\sqrt{w_k} \Sigma_k^{-1/2} \mu_k^b \right) \tag{2.5}$$

where μ_k^a and μ_k^b are the adapted means; and w_k and Σ_k are the UBM weights and variances. This technique is framed in the group known as super-vector methods. A Gaussian super-vector (GSV) is built by concatenating the means of the components of the GMM. A closer look reveals that this kernel is just an inner product between two GSV where the means are normalized by their corresponding weights and variances ($\sqrt{w_k}\Sigma_k^{-1/2}$).

To enroll a given speaker, the GSVs of his enrollment utterances are used as positive examples and a group of impostor utterances as negative examples. Thus, we train a SVM for each target speaker. Fast evaluation of the SVM can be implemented by taking advantage of the GSV kernel linearity. Equation (2.4) can be simplified as,

$$f(\mathbf{x}) = \left(\sum_{i=1}^{M} y_i \alpha_i \phi(\mathbf{x}_i) \right)^T \phi(\mathbf{x}) + b = \mathbf{w}^T \phi(\mathbf{x}) + b \tag{2.6}$$

where $\phi(\mathbf{x})$ is the GSV of the test segment, $\phi(\mathbf{x}_i)$ are the support vectors obtained during training and α_i their corresponding weights.

There are techniques to compensate the variability between sessions of the same speaker in the super-vector space. The *nuisance attribute projection* (NAP) approach was introduced in [Campbell et al., 2006b] and [Solomonoff et al., 2005]. NAP estimates a low rank matrix $\mathbf{U}$ which contains the directions of intra-speaker variability of the super-vectors. This matrix is usually called eigen-channels matrix. Then, the projection matrix defined as

$$\mathbf{P} = (\mathbf{I} - \mathbf{U}\mathbf{U}^T) \tag{2.7}$$

is multiplied by the super-vector to remove the undesired variability.

The criterion to estimate $\mathbf{U}$ consists in minimizing the objective function:

$$\mathbf{U}^* = \operatorname*{arg\,min}_{\mathbf{U}, \|\mathbf{U}\|=1} \sum_{i,j} W_{ij} \left\| \mathbf{P}(\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)) \right\|^2 \tag{2.8}$$

where $\mathbf{x}$ are utterances from a development dataset, which counts with a fairly large number of speakers with several sessions by speaker. To make same speaker super-vector of different session closer to one another, we set $W_{ij} = 1$ if utterances i and j belong to the same speaker and $W_{ij} = 0$ otherwise. In practice [Campbell et al., 2006b], that means to find the principal components of the within-class covariance matrix,

$$\mathbf{S} = \sum_{i=1}^{M} \sum_{j=1}^{N_i} \left(\phi(\mathbf{x}_{ij}) - \overline{\phi(\mathbf{x}_i)} \right) \left(\phi(\mathbf{x}_{ij}) - \overline{\phi(\mathbf{x}_i)} \right)^T \tag{2.9}$$

where M is the number of speakers, N_i is the number of sessions of speaker i, $\phi(\mathbf{x}_{ij})$ is the super-vector of the j^{th} session of speaker i and $\overline{\phi(\mathbf{x}_i)}$ is the average super-vector of i.

2.4.3 Joint factor analysis

Joint factor analysis (JFA) is a generative model that allows to estimate the speaker's GMM taking into account the different sources of variability (speaker and channel) separately. As in the classical MAP by Reynolds [Reynolds et al., 2000], JFA adapts the means of the UBM to the speaker while the weights and variances are shared among all the speakers. Therefore, the speaker model can be represented by a super-vector formed by concatenating the means of the corresponding GMM.

Joint factor analysis (JFA) for speaker recognition [Kenny et al., 2007b] assumes that the mean super-vector $\mathbf{M}$ for a given speech utterance can be decomposed into a speaker component $\mathbf{s}$ and a channel component $\mathbf{c}$:

$$\mathbf{M} = \mathbf{s} + \mathbf{c} \tag{2.10}$$

where $\mathbf{s}$ and $\mathbf{c}$ are *a priori* statistically independent and normally distributed. What we usually call channel component can include other inter-session variability effects like phonetic content or language.

The speaker component $\mathbf{s}$ is a hidden variable with the form

$$\mathbf{s} = \mathbf{m} + \mathbf{V}\mathbf{y} + \mathbf{D}\mathbf{z} \tag{2.11}$$

where $\mathbf{m}$ is the UBM mean super-vector, $\mathbf{V}$ is a low-rank matrix, $\mathbf{D}$ is a diagonal matrix and, $\mathbf{y}$ and $\mathbf{z}$ are standard normal distributed vectors. $\mathbf{V}$ is referred as the eigen-voices matrix and $\mathbf{y}$ as the speaker factors vector. The columns of $\mathbf{V}$ span the subspace of speaker variability. $\mathbf{D}$ is referred as the remaining variability matrix and $\mathbf{z}$ the remaining variability factors. $\mathbf{D}$ accounts for the speaker variability not included in $\mathbf{V}$.

The channel component $\mathbf{c}$ has the form

$$\mathbf{c} = \mathbf{U}\mathbf{x} \tag{2.12}$$

where $\mathbf{U}$ is a low-rank matrix and $\mathbf{x}$ is a standard normal distributed vector. $\mathbf{U}$ is referred as the eigen-channels matrix and $\mathbf{x}$ as the channel factors vector. The columns of $\mathbf{U}$ span the subspace of inter-session variability.

The matrices $\mathbf{V}$, $\mathbf{U}$ and $\mathbf{D}$ are called the hyperparameters of the JFA model. They are estimated from a development database composed of a large number of speakers recorded over several sessions. In [Kenny et al., 2008], the hyperparameter estimation is discussed in detail. In [Kenny, 2005], the author describes the mathematical formulation of JFA including the derivations of the equations involved.

In previous approaches, where enrollment was done by classical MAP, we needed to estimate all the values of the mean super-vector ($\sim$ 122000). When the amount of enrollment data is limited, we may not have enough samples for every Gaussian, and some of them are badly estimated. In JFA, the eigen-voices term accounts for the main part of the speaker deviation of the mean. To estimate its contribution we only need to estimate the ~ 300 coefficients of $\mathbf{y}$. Thus, the eigen-voice term prevents over-fitting. Besides, when the enrollment data is scarce the contribution of the remaining variability term is negligible.

Several scoring approaches have been proposed for the JFA model, the most representative are discussed in [Glembek et al., 2009]. The linear scoring is computationally efficient and produces good performance. The speaker model is obtained by computing the MAP point estimates of the factors $\mathbf{y}$ and $\mathbf{z}$ given the enrollment data. Then, the log-likelihood of the test data given the speaker model is approximated by its first order Taylor expansion about the UBM mean super-vector. After some algebra, we obtain,

$$\mathrm{LLR} \approx \left(\mathbf{V}\mathbf{y}_{\mathrm{enroll}} + \mathbf{D}\mathbf{z}_{\mathrm{enroll}}\right)^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{F}_{\mathrm{tst}} - \mathbf{N}_{\mathrm{tst}}\mathbf{m} - \mathbf{N}_{\mathrm{tst}}\mathbf{U}\mathbf{x}_{\mathrm{tst}}\right) \tag{2.13}$$

where $\mathbf{y}_{\mathrm{enroll}}$ and $\mathbf{z}_{\mathrm{enroll}}$ are the enrollment factors; $\mathbf{N}_{\mathrm{tst}}$ and $\mathbf{F}_{\mathrm{tst}}$ are the sufficient statistics of the test segment given the UBM; and $\mathbf{x}_{\mathrm{tst}}$ is MAP point estimate of the channel factor given the test segment. $\mathbf{F}_{\mathrm{tst}}$ is created by concatenating the first order cumulants of the UBM components. $\mathbf{N}_{\mathrm{tst}}$ is a diagonal matrix whose diagonal blocks are $N_k\mathbf{I}_d$, where N_k are the occupation cumulants of the k^{th} Gaussian and d the feature dimension.

In [Dehak et al., 2008] a comparison between JFA and GMM-SVM shows that JFA clearly outperforms SVM thanks to the use of speaker factors.

2.4.4 Identity vectors (i-vectors)

Lately, a new approach derived from factor analysis has received great attention among the speaker recognition community. Dehak [Dehak et al., 2011b] proposes to use factor analysis as a feature extractor. This approach, instead of defining a speaker and a channel variability space as JFA, defines a single space that is referred as *total variability space* and contains both sources of variability simultaneously. The GMM super-vector for a given utterance is

$$\mathbf{M} = \mathbf{m} + \mathbf{T}\phi \tag{2.14}$$

where $\mathbf{m}$ is the UBM mean, $\mathbf{T}$ is a low-rank matrix defining the total variability space and ϕ is a standard normal distributed vector. ϕ is referred as total variability factors or *identity vectors (i-vectors)* and they are used as features in a posterior classification stage.

Several classification algorithms have embraced this new feature. In [Dehak et al., 2011b], linear discriminant analysis (LDA) and within-class covariance normalization (WCCN) compensate the channel distortion in the i-vectors. Then, scoring is produced by cosine distance or SVM.

In [Matejka et al., 2011,Kenny, 2010], i-vectors are modeled by PLDA (a single Gaussian simplification of JFA) [Prince and Elder, 2007]. An i-vector ϕ_{ij} from the session j of the speaker i is written as

$$\phi_{ij} = \mu + \mathbf{V}\mathbf{y}_i + \mathbf{U}\mathbf{x}_{ij} + \epsilon_{ij} \tag{2.15}$$

where μ is a speaker independent term, $\mathbf{V}$ is a low-rank matrix of eigen-voices, $\mathbf{y}_i$ is the speaker factor vector, $\mathbf{U}$ is a low-rank matrix of eigen-channels, $\mathbf{x}_{ij}$ is the channel factor vector and ϵ_{ij} is an offset that accounts for the rest of channel variability not included in

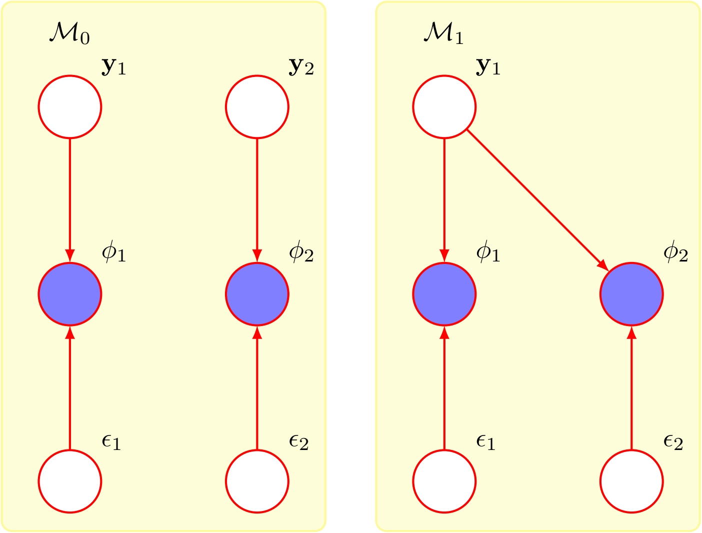


Figure 2.4: Speaker verification with PLDA. We compute the likelihood ratio of the i-vectors given two generative models: $\mathcal{M}_0$, which represents the non-target hypothesis, and $\mathcal{M}_1$, which represents the target hypothesis.

$\mathbf{Ux}_{ij}$. The factors $\mathbf{y}$ and $\mathbf{x}$ can have standard normal or heavy-tailed priors. The prior distribution of ϵ is a diagonal Gaussian.

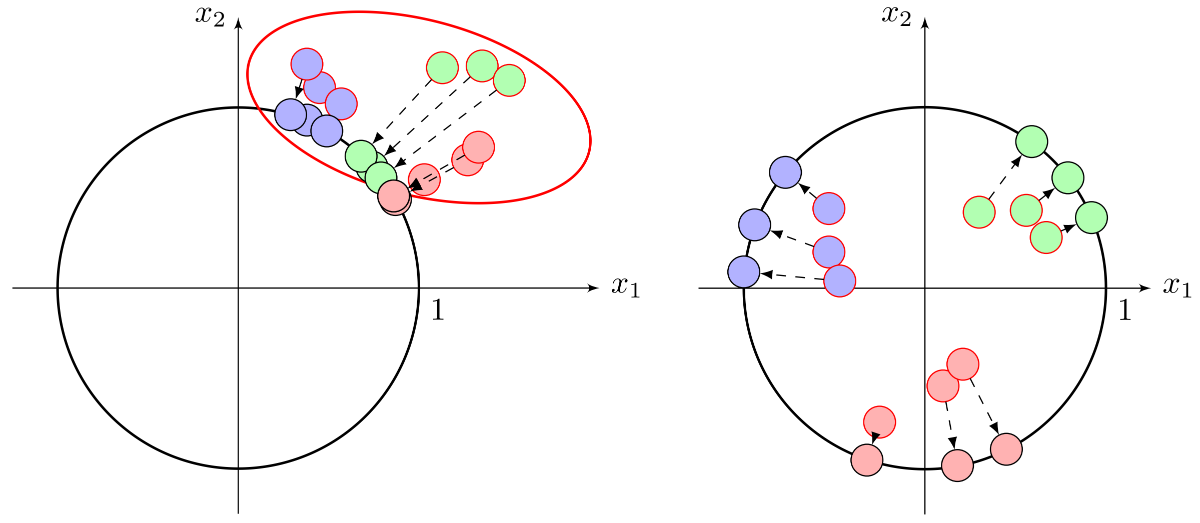
The simplified PLDA model (SPLDA) [Villalba, 2011] puts aside the eigen-channels term and assumes a full covariance Gaussian for the ϵ prior. This is equivalent to PLDA with full-rank $\mathbf{U}$. Another simplification is the two-covariance model [Brummer and De Villiers, 2010] where the i-vector is decomposed as $\phi_{ij} = \mathbf{y}_i + \epsilon_{ij}$ and the priors for $\mathbf{y}$ and ϵ are full covariance Gaussians. This model is equivalent to a PLDA with full-rank $\mathbf{V}$ and $\mathbf{U}$; thus it is also called full-rank PLDA.

PLDA is scored by computing the ratio between the likelihood of the trial i-vectors given the target hypothesis and the corresponding likelihood given the non-target hypothesis. Both hypothesis are illustrated by the graphical models in Figure 2.4. If the speaker in the enrollment and test i-vectors is the same ($\mathcal{M}_1$), both i-vectors share the same speaker factor $\mathbf{y}$ but have different channel offsets. On the other hand, if they belong to different speakers they also have different speaker factors. Thus, the ratio is computed as

$$R(\phi_1, \phi_2) = \frac{P(\phi_1, \phi_2|\mathcal{M}_1)}{P(\phi_1, \phi_2|\mathcal{M}_0)} = \frac{\int P(\phi_1, \phi_2|\mathbf{y}_1) P(\mathbf{y}_1)\ \mathrm{d}\mathbf{y}_1}{\int P(\phi_1|\mathbf{y}_1) P(\mathbf{y}_1)\ \mathrm{d}\mathbf{y}_1 \int P(\phi_2|\mathbf{y}_2) P(\mathbf{y}_2)\ \mathrm{d}\mathbf{y}_2} \ . \tag{2.16}$$

Note that the speaker identity variables are integrated out. Instead of computing point estimates for $\mathbf{y}$ and comparing the enrollment and test identity variables, we compute the likelihood that both are generated by the same $\mathbf{y}$ regardless of what is the value of $\mathbf{y}$. This method takes into account the uncertainty about the value of $\mathbf{y}$.

As for JFA, the parameters of the PLDA model can be trained by maximum likelihood and minimum divergence iterations [Brummer, 2010b] from a development dataset. Mathematical derivations of the EM algorithm for different PLDA flavors can be found in Appendix C. Discriminative training has also been proposed [Burget et al., 2011, Cumani et al., 2011, Cumani et al., 2012].



(a) Without centering and whitening.

(b) With centering and whitening.

Figure 2.5: Length normalization examples. Length normalized samples have borders in black and non-normalized have borders in red. The fill colors indicate different speakers.

The analysis presented in [Garcia-Romero and Espy-Wilson, 2011] proves that by normalizing the i-vectors by their magnitude–also called length normalization–boosts the performance of PLDA. That is

$$\hat{\phi} = \frac{\phi}{\|\phi\|} \; . \tag{2.17}$$

Before applying length normalization, i-vectors need to be centered and whitened. Figure 2.5 illustrates the length normalization procedure. If i-vectors are not centered and whitened, they are projected into a small region of the hypersphere which makes them less discriminative. On the other hand, centered and whitened i-vectors are evenly distributed around the unit hypersphere. For high dimensional vectors, length normalization makes heavy-tailed distributions to become closer to a Gaussian distribution, which makes possible to use efficient Gaussian models obtaining good performance. For some datasets like NIST SRE, length normalization eliminates the need of score normalization.

2.5 Score Normalization

The speaker verification decision is taken by comparing the score provided by the classifier with a threshold. If the score is higher than the threshold the target speaker is accepted, otherwise it is rejected. The choice of the optimum threshold is troublesome in speaker verification due to the score variability between trials. The Score variability may be caused by different phenomena. They include phonetic content of the utterances, length, channel type, noise, emotion or any other type of inter-session variability. Approaches like SVM-NAP or JFA need score normalization to reach optimum performance despite including inter-session variability compensation.

Score normalization techniques were introduced in [Li and Porter, 1988]. In this work, the authors observed large variances in the impostor and target score distributions. To reduce that variability, they proposed methods based on normalizing the impostor score distributions to be zero mean and unit variance. Thus, the normalized score s' is

$$s' = \frac{s - \mu}{\sigma} \tag{2.18}$$

where s is the raw score of the classifier, and μ and σ are the mean and standard deviation of the impostor scores distribution. μ and σ are computed from a cohort of impostors.

The most popular normalization techniques are Z-Norm and T-Norm [Auckenthaler et al., 2000]. In Z-Norm (zero normalization), μ and σ depend on the enrollment segments. We have to score a cohort of non-target test segments against the enrollment model; obtain the mean and standard deviation of those scores; and plug them in (2.18). On the other side, in T-Norm (test normalization), we must score the test segment against a cohort non-target models. Applying both, one behind the other, we have ZT-Norm or TZ-Norm. With the advent of i-vectors where scores are symmetric–the score is the same if we interchange the enrollment and test segments–, the S-Norm (symmetric norm) popularized [Brummer and Strasheim, 2009]. It is defined as

$$s' = \frac{s - \mu_{\text{enroll}}}{\sigma_{\text{enroll}}} + \frac{s - \mu_{\text{tst}}}{\sigma_{\text{tst}}} \tag{2.19}$$

where μ_{enroll} and σ_{enroll} are computed by scoring the enrollment segment against the cohort, and μ_{tst} and σ_{tst} by scoring the test segment against the cohort. Less popular types of normalization are H-Norm (handset normalization) [Heck and Weintraub, 1997], HT-Norm [Dunn et al., 2001] or C-Norm [Bimbot et al., 2004]. Selecting the cohort speakers similar to the target model can improve performance [Ramos et al., 2005, Sturim and Reynolds, 2005].

2.6 Calibration

The last step before decision making is calibration. For a binary classification problem calibration can simply refer to selecting the optimum decision threshold for the intended application. Applications are mainly defined by the prior probability for the target trials $P_{\mathcal{T}}$. For example, an application that searches for a criminal in thousands of phonecalls has a low $P_{\mathcal{T}}$. Meanwhile, a system that employs an electronic card plus speaker verification to access a bank account will have a high target prior because there will be very few impostors (they would need to steal the card before attempting access). The prior $P_{\mathcal{T}}$ is also called the *operating point* of the system.

Calibration is discussed in a wider sense in [Brummer and Preez, 2006]. Here, calibration consists in transforming the scores given by the classifier into meaningful log-likelihood ratios. Then, the optimum decision threshold is obtained by applying Bayes rule:

$$P\left(\mathcal{T}|\mathcal{D}\right) = \frac{P_{\mathcal{T}} P\left(\mathcal{D}|\mathcal{T}\right)}{P_{\mathcal{T}} P\left(\mathcal{D}|\mathcal{T}\right) + (1 - P_{\mathcal{T}}) P\left(\mathcal{D}|\mathcal{N}\right)} > 0.5 \implies \text{LLR} > -\log\frac{P_{\mathcal{T}}}{1 - P_{\mathcal{T}}} = -\text{logit} P_{\mathcal{T}} \tag{2.20}$$

where $\mathcal{D}$ refers to the trial enrollment and test data. A system that produces well-calibrated likelihood ratios can be used for any application. It may happen that the likelihood ratios are only valid for a range of operating points because extreme values of LLR do not have enough examples to properly train the calibration function.

Scores are usually calibrated by a monotonically increasing function. Linear calibration produces good results because it needs few parameters and, thus, there is low risk of over-fitting. Training calibration via linear logistic regression is the *de facto* standard in SV thanks to Niko Brummer's open source toolkits Focal [Brummer, 2006] and Bosaris [Brummer and De Villiers, 2011]. Logistic regression is also used to jointly fuse and calibrate the scores of multiple classifiers [Brummer et al., 2006]. Different flavors of logistic regression have been researched. The work in [Hautamaki et al., 2011] explores the effect of different regularizers. In [Hautamaki et al., 2012], a variational Bayes procedure integrates out the parameter that controls the weight of the regularization term. In [Ferrer et al., 2008], the authors improve the fusion by incorporating auxiliary information.

The reader can refer to Brummer's thesis [Brummer, 2010c] for further reading about calibration and evaluation of speaker recognizers.

2.7 Performance Evaluation

There are two types of errors in speaker verification systems: false rejections or misses and false acceptances or false alarms. A false rejection occurs when a valid target speaker is rejected. A false acceptance happens when an impostor is accepted. Both types of errors depend on the decision threshold ξ. A low decision threshold will produce low miss rates (P_{Miss}) and high false alarm rates (P_{FA}). On the contrary, a high threshold will reject many target speakers and accept very few impostors. The choice of the decision threshold is a trade-off between both types of errors and it depends on the kind of application for which the SV system is intended. The pair (P_{Miss},P_{FA}) defines the operating point of the system. The *equal error rate* (EER) is defined as the error rate at the operating point where $P_{\text{Miss}} = P_{\text{FA}}$. The EER is the most popular metric in the speaker recognition literature to compare different approaches.

The curve representing the trade-off between true and false acceptance rates is call *receiver operating characteristic* (ROC) and is monotonic increasing. If we plot miss rate against false alarm rate and the axes use a normal deviate scale, the curve is called *detection error trade-off* (DET). The non-linear probability scale makes the plots visually more intuitive. If the score distributions are Gaussians, the DET curves are straight lines and the distances between curves illustrate performance differences more clearly. The DET curve has become the standard performance representation since it was introduced by Martin [Martin et al., 1997] in the NIST evaluations. Examples of DET curves can be found in Figure 2.6.

Another performance metric is the detection cost function (DCF) C_{Det} [Doddington, 2000]. This is a weighted sum of the miss and false alarm rates:

$$C_{\text{Det}} = C_{\text{Miss}} P_{\mathcal{T}} P_{\text{Miss}} + C_{\text{FA}} (1 - P_{\mathcal{T}}) P_{\text{FA}} \tag{2.21}$$

where C_{Miss} and C_{FA} are the costs of having a miss or a false alarm, respectively, and $P_{\mathcal{T}}$ is the target prior probability. The parameters C_{Miss}, C_{FA} and $P_{\mathcal{T}}$ depend on the intended application. In this way, the cost function produces a measure that is meaningful for that application. The optimum operating point of the system is the pair (P_{Miss},P_{FA}) at which C_{Det} is minimum.

The primary performance measure of NIST evaluation is a normalized version of the DCF [NIST Speech Group, 2010]:

$$C_{\mathrm{Norm}} = C_{\mathrm{Det}} / \min(C_{\mathrm{Miss}} P_{\mathcal{T}}, C_{\mathrm{FA}}(1 - P_{\mathcal{T}})) \ . \tag{2.22}$$

Along this thesis, when we refer to DCF, we mean this normalized DCF. C_{Det} and C_{Norm} depend on the decision threshold ξ. It is common in the literature to evaluate systems with respect to their actual and minimum DCF. For actual DCF, we understand to compute the cost for a fixed threshold. If the scores a well-calibrated likelihood ratios, the threshold is selected to minimize Bayes risk as shown in (2.20). However, if we select the threshold that minimizes the cost on the test dataset, we obtain the minimum DCF. The latter allows us to compare systems regardless of the calibration. An actual $C_{\mathrm{Norm}} > 1$ means that our system is inappropriate for the intended application.

The P_{Miss} and P_{FA} are measured experimentally on a test corpus by counting the errors of each type. This means that large datasets are needed to measure error rates accurately. According to the Doddington's "rule of 30" [Doddington, 2000], to be 90% confident that the true error rate is ±30% of the measured error rate there must be at least 30 errors. Therefore, the lower the error rates the system provides, the larger the test set needed to measure those error rates precisely.

Since 1996 the National Institute of Standards and Technology (NIST) has been conducting periodic evaluations of the state-of-the-art of the speaker recognition technology (NIST SRE) [Martin and Przybocki, 2001, Przybocki et al., 2007]. NIST provides a common evaluation framework to compare approaches implemented by different research groups. NIST datasets include telephone phonecalls and interviews recorded over far-field microphones with same and cross-channel trials. It also has 10 second conditions, low and high vocal effort, cross-language trials, among others.

2.8 Experimental Comparison

In this section, we show the progress of speaker verification systems during the last years. We focus on short-time spectral features and the most representative statistical models described in Section 2.4. These systems have been part of the I3A submissions to NIST SRE from 2006 to 2010 [Villalba et al., 2008, Villalba et al., 2010].

2.8.1 Experimental setup

2.8.1.1 Evaluation dataset

We evaluate multiple approaches on a common dataset, the NIST SRE 2010 core extended condition [NIST Speech Group, 2010]. This dataset includes phonecalls recorded over telephone channel or far-field microphones, and interviews recorded over far-field microphones. Besides, conditions with high and low vocal effort were evaluated for the first time. The recording setup includes 14 microphones of different types placed in different locations around the subject [Cieri et al., 2007]. Segment durations are 5 minutes for phonecalls and 3 or 8 minutes for interviews.

The primary performance metric of NIST SRE 2010 is the normalized detection cost function given in (2.22) with the parameters $P_{\mathcal{T}} = 0.001$, $C_{\mathrm{Miss}} = 1$ and $C_{\mathrm{FA}} = 1$. These parameters represent an operating point with a very low number of false alarms. Taking into account Doddington's rule of 30, to measure the false alarm rate in this specific operating point we need a large number of non-target trials. For this reason, NIST decided to create the extended condition by including more than 2.8 millions of male trials and more than 3.6 millions of female trials.

Since, in the evaluation, there are multiple enrollment and test conditions with different number of trials of each type, it is inadequate to pool all the trials to measure the overall performance. Instead, nine common conditions were defined including the following subsets of trials:

- Det1: Trials involving interview speech from the same microphone in enrollment and test.
- Det2: Trials involving interview speech from different microphones in enrollment and test.
- Det3: Trials involving interview enrollment speech and normal vocal effort conversational telephone test speech.
- Det4: Trials involving interview enrollment speech and normal vocal effort conversational telephone test speech recorded over a room microphone channel.
- Det5: Different number trials involving normal vocal effort conversational telephone speech in enrollment and test.
- Det6: Telephone channel trials involving normal effort in enrollment and high vocal effort in test.
- Det7: Room microphone recorded phone call trials with normal vocal effort in enrollment and high vocal effort in test.
- Det8: Telephone channel trials involving normal effort in enrollment and low vocal effort in test.
- Det9: Room microphone recorded phone call trials with normal vocal effort in enrollment and low vocal effort in test.

2.8.1.2 Feature extraction

Our front-end extracts feature vectors with 20 MFCC including C0 (C0-C19) over a 25 ms hamming window with 10 ms frame rate (15 ms overlap). First and second order derivatives are computed over the feature vector sequence and appended.

Voice activity detection (VAD) was performed by computing the long-term spectral divergence (LTSD) of the signal every 10 ms, and comparing it against a threshold as in [Ramirez et al., 2004]. For phone calls, where two channels are available, namely channel of interest and reference channel, the reference channel was used for crosstalk removal. For interview segments, the NIST provided ASR labels were employed for removing the interviewer.

After silence removal selection, features were short-time Gaussianized using a 3 seconds window as in [Pelecanos and Sridharan, 2001].

2.8.1.3 GMM-UBM MAP classifier

Gender Dependent (GD) Universal Background Models (UBM) of 2048 diagonal covariance Gaussians were trained by EM iterations. For this purpose, we used all the telephone signals in NIST SRE 2004 to 2006 databases (649 male speakers with 7412 signals and 801 female speakers with 9889 signals).

Speakers were enrolled by MAP adaptation of the means of the UBM. Trials were evaluated by computing the average log-likelihood ratio of the test frames given the target speaker model and the UBM as in (2.3).

2.8.1.4 GSV-SVM-NAP classifier

Gaussian super-vectors were obtained from the MAP adapted GMM of each speech segment as described in Section 2.4.2. We used the enrollment segment as unique positive example to train the SVM. A cohort of impostors from NIST SRE 2004–2006 was used as negative examples. The cohort was composed of 2363 and 3314 segments for male and female speakers respectively.

Nuisance attribute projection was applied to super-vectors to remove inter-session variability. We trained 100 eigen-channels $\mathbf{U}_{\mathrm{phn}}$ on the telephone segments from the speakers with more 8 conversations in NIST SRE 2004–2006 (530 male speakers with 7398 signals and 731 female speakers with 9938 signals). Another 100 eigen-channels $\mathbf{U}_{\mathrm{mic}}$ were trained on all the segments from the speakers with microphone speech in NIST SRE 2005 and 2006 and 50 speakers of 2008 (106 male speakers with 6244 signals and 119 female speakers with 6919 signals). It has been observed that training $\mathbf{U}_{\mathrm{mic}}$ not only with microphone speech but pooling the telephone and microphone segments produces better performance. To train $\mathbf{U}_{\mathrm{mic}}$, we first applied $\mathbf{U}_{\mathrm{phn}}$ to the GSV to remove the variability common to telephone and microphone segments. Then, we trained $\mathbf{U}_{mic}$ on the compensated super-vectors. Thus, in theory, $\mathbf{U}_{\mathrm{mic}}$ only includes microphone variability and cross-channel variability. Finally, we stacked both eigen-channel matrices $\mathbf{U} = [\mathbf{U}_{\mathrm{phn}}\mathbf{U}_{\mathrm{mic}}]$.

2.8.1.5 JFA classifier

JFA hyperparameters were trained by maximum likelihood and minimum divergence iterations. 300 eigen-voices ($\mathbf{V}$) and 100 telephone eigen-channels ($\mathbf{U}_{\mathrm{phn}}$) were trained on telephone data from all the speakers of SRE2004, SRE2005 and SRE2006 databases having, at least, 8 recordings by speaker (530 male speakers with 7398 signals and 731 female speakers with 9938 signals). To speed up the training, some approximations were taken. First, we trained $\mathbf{V}$ by assuming that, for speakers with many recordings, channel effects cancel when we accumulate the sufficient statistics of all their sessions, and we can consider that the average channel factor $\mathbf{x}$ is zero. We also considered that the contribution of the residual variability term $\mathbf{Dz}$ to the speaker model is small compared to the rest of terms so $\mathbf{z}$ was also set to zero. Thus, $\mathbf{V}$ was trained by a simplified model with speaker factors only. Once we had $\mathbf{V}$, we computed MAP point estimates for the speaker factors $\hat{\mathbf{y}}$ of each speaker. The effect of the speaker can be removed of the first-order sufficient statistics as

$$\mathbf{F}' = \mathbf{F} - \mathbf{N}\mathbf{V}\hat{\mathbf{y}} \; . \tag{2.23}$$

Training $\mathbf{U}_{\mathrm{phn}}$ by fixing the sufficient statistics to $\mathbf{F}'$ during the EM algorithm is straightforward.

As for the GSV-SVM, another 100 eigen-channels ($\mathbf{U}_{\text{mic}}$) were trained on all the signals from speakers having far field microphone data in SRE2005 and SRE2006 and 50 speakers (kept out speakers) with interview data from SRE2008 (106 male speakers with 6244 signals and 119 female speakers with 6919 signals). Similarly to what we did for $\mathbf{U}_{\text{phn}}$, we computed MAP point estimates for $\mathbf{y}$ and $\mathbf{x}_{\text{phn}}$, and we used them to remove the speaker and telephone channel variability from $\mathbf{F}$

$$\mathbf{F}' = \mathbf{F} - \mathbf{N}\mathbf{V}\hat{\mathbf{y}} - \mathbf{N}\mathbf{U}_{\text{phn}}\hat{\mathbf{x}}_{\text{phn}} \; . \tag{2.24}$$

and then, we estimated $\mathbf{U}_{\text{mic}}$. Both eigen-channel matrices were stacked together.

Finally, the remaining speaker variability matrix ($\mathbf{D}$) was trained on the speakers of NIST SRE 2004–2006 with least than 8 recordings (201 male speakers with 547 signals and 152 female speakers with 668 signals). $\mathbf{D}$ is not trained on the same speakers as $\mathbf{V}$ because that drives to underestimate the amount of speaker variability not included in $\mathbf{V}$. Again, we computed MAP point estimates of $\mathbf{y}$ and $\mathbf{x}$, compensate $\mathbf{F}$ and estimate $\mathbf{D}$.

Speakers were enrolled by computing MAP point estimates of $\mathbf{y}$ and $\mathbf{z}$. $\mathbf{y}$ and $\mathbf{x}$ were jointly estimated by fixing $\mathbf{z} = 0$. Then, the contributions of $\mathbf{y}$ and $\mathbf{x}$ were removed from the statistics to compute $\mathbf{z}$. The linear scoring was used to evaluate the trials.

2.8.1.6 i-Vectors and PLDA classifier

We trained the total variability space by maximum likelihood and minimum divergence iterations on the same data as for the JFA eigen-voices matrix. We present results with the *raw* i-vectors and length normalized i-vectors. We used simplified PLDA for classification with 200 eigen-voices. The PLDA log-likelihood ratio was evaluated as shown in [Brummer and De Villiers, 2010].

The means and covariances needed to center and whiten the i-vectors before length normalization and the PLDA parameters were estimated on NIST SRE 2004–2006 and 50 speakers from 2008 pooling telephone and microphone segments. Mathematical derivations of the EM algorithm for PLDA can be found in Appendix C. All our models were gender dependent.

2.8.1.7 Score normalization

Score normalization was gender dependent. The GMM-UBM, GSV-SVM and JFA systems used ZT-Norm. As PLDA scoring is symmetric, the i-vector system without length normalization uses S-Norm. The system with length normalization does not need score normalization. The cohorts included 2300 male and 3300 female telephone segments from NIST SRE 2004–2006.

2.8.1.8 Calibration

Calibration was trained by linear logistic regression with the Bosaris toolkit [Brummer and De Villiers, 2011]. The calibration was gender and channel dependent. We trained different calibration functions for microphone–microphone, microphone–telephone and telephone–telephone trials.

To train calibration, we built a trial list with NIST SRE 2008 data including all the channel conditions in SRE 2010. This list included all trials that can be done from all training short, long and follow-up versus all testing short, long and follow-up English 2008 segments. We kept out the 50 speakers that we used to train NAP, JFA and PLDA. In total, we had around 4 millions of male trials and 10 millions of female trials.

2.8.2 Results

2.8.2.1 Classifiers analysis

In Table 2.1, we show results comparing the five classifiers previously described for the nine common conditions of NIST SRE10. Performance is measured in terms of EER, minimum and actual DCF. Besides, results are presented for male, female and the pool of both. We also display DET curves for the pool of male and female trials. Figure 2.6 shows results for the normal vocal effort common conditions and Figure 2.7 for the high and low vocal effort conditions.

The table and figures evidence a dramatic improvement between the basic GMM-UBM and the GSV-SVM thanks to the NAP channel compensation included in the latter. For example, for the det2 common condition (interview-interview different channel), we achieved relative improvements of 49% and 18% in terms of EER and minimum DCF respectively. And for det5 (phn-phn), we obtained improvements of 62% and 23%. The DET curves consistently improved along all the operating points. However, actual DCF did not improve some much. The GSV-SVM calibration is not very good and there is still a noticeable gap between minimum and actual DCF.

There is another jump of performance when we evolve from GSV-SVM to JFA thanks to the inclusion of eigen-voices. Compared to the GSV-SVM system, in det2, we attained relative improvements in EER and minimum DCF of about 53% and 25%. In det5, they improved by around 40% and 23%. Besides, in det5, calibration was almost perfect. Again, we ameliorated DET curves along all the operating points.

If we compare JFA and i-vectors based systems, we do not find a performance enhancement as evident as between JFA and previous approaches. In some common conditions JFA was still better. In terms of minimum DCF, the system with PLDA and raw i-vectors outperformed JFA only in conditions det1 (interview-interview same channel), det4 (interview-phonecall in microphone channel) and det7 (microphonic phonecalls with high vocal effort in test). However, the PLDA calibration was better and, in terms of actual DCF, it outperformed JFA in five conditions (det1, det2, det4, det7 and det9).

Regarding the system with PLDA and length normalized i-vectors, we improved in all the conditions that involve microphone speech in enrollment and test. We improved as much for minimum DCF as for actual DCF. For example, in det2, we attained improvements relative to JFA of about 22% in terms of EER and minimum DCF and 47% in terms of actual DCF. Looking at the DET curves, the improvement is larger in the low false alarm operating points. In telephone conditions, JFA had slightly better costs. However, we confirmed that by training PLDA with only telephone data we can reach the same performance in telephone as JFA, but, at the cost of deteriorating the microphone conditions. Definitely, PLDA was much better than JFA in microphonic conditions (det1, det2, det3, det4, det7 and det9) and JFA was slightly better in telephone conditions (det5, det6 and det8). We hypothesize that PLDA is better calibrated than JFA because of the approximations taken to evaluate the likelihood ratios. For PLDA, we compute the likelihood ratio by (2.16), which strictly complies with the rules of probability. On the contrary, for JFA, to speed-up the ratio

Table 2.1: EER(%)/MinDCF/ActDCF NIST SRE10 core extended common conditions for different classifiers.

CC	System	male			female			male + female		
		EER	MinDCF	ActDCF	EER	MinDCF	ActDCF	EER	MinDCF	ActDCF
1	GMM-UBM	5.15	0.742	0.953	6.74	0.852	0.987	6.24	0.807	0.971
	GSV-SVM	4.04	0.660	0.796	5.67	0.770	0.796	4.99	0.726	0.792
	JFA	1.35	0.352	0.587	2.96	0.705	0.744	2.25	0.597	0.673
	PLDA	**1.24**	0.395	0.499	2.97	0.589	**0.640**	2.21	0.578	0.580
	Lnorm+PLDA	1.25	**0.206**	**0.235**	**1.98**	**0.344**	0.644	**1.67**	**0.293**	**0.464**
2	GMM-UBM	11.18	0.951	1.000	15.83	0.990	1.000	13.86	0.972	1.000
	GSV-SVM	5.04	0.705	0.942	8.63	0.868	0.980	7.03	0.797	0.962
	JFA	1.78	0.402	0.902	4.45	0.731	0.950	3.28	0.594	0.928
	PLDA	1.85	0.567	0.767	4.55	0.761	0.787	3.40	0.718	0.778
	Lnorm+PLDA	**1.76**	**0.318**	**0.376**	**3.26**	**0.572**	**0.584**	**2.57**	**0.465**	**0.489**
3	GMM-UBM	9.64	0.984	1.052	11.15	0.998	1.092	10.51	0.993	1.071
	GSV-SVM	3.89	0.525	0.601	6.07	0.679	1.098	4.99	0.616	0.860
	JFA	2.61	**0.414**	**0.493**	3.42	0.576	**0.607**	3.05	0.546	**0.553**
	PLDA	3.27	0.625	0.713	3.75	0.725	0.751	3.52	0.696	0.732
	Lnorm+PLDA	**2.07**	0.448	0.494	**3.00**	**0.531**	0.612	**2.55**	**0.501**	**0.553**
4	GMM-UBM	10.36	0.867	0.985	12.70	0.930	0.998	11.72	0.903	0.991
	GSV-SVM	6.00	0.754	0.773	7.99	0.794	0.820	7.04	0.780	0.792
	JFA	1.99	0.416	0.770	3.69	0.684	0.835	2.83	0.590	0.802
	PLDA	1.74	0.440	0.608	3.33	0.707	0.724	2.57	0.657	0.669
	Lnorm+PLDA	**1.30**	**0.276**	**0.291**	**2.16**	**0.434**	**0.486**	**1.76**	**0.365**	**0.389**
5	GMM-UBM	9.84	0.867	0.877	12.19	0.862	0.895	11.10	0.882	0.886
	GSV-SVM	3.20	0.535	0.573	4.99	0.704	1.202	4.15	0.676	0.929
	JFA	**1.86**	**0.356**	**0.456**	**3.07**	**0.550**	**0.586**	**2.51**	**0.518**	**0.534**
	PLDA	2.72	0.534	0.806	3.97	0.781	0.791	3.41	0.781	0.802
	Lnorm+PLDA	2.28	0.386	0.486	3.67	0.589	0.645	3.03	0.578	0.581
6	GMM-UBM	17.26	0.994	1.383	21.29	0.998	1.129	19.58	0.997	1.233
	GSV-SVM	6.57	0.998	1.668	8.53	1.000	1.372	7.69	0.999	1.493
	JFA	**4.28**	**0.834**	**0.848**	**5.74**	**0.862**	**0.886**	**5.12**	**0.852**	**0.870**
	PLDA	5.50	0.887	0.941	7.31	0.931	0.947	6.52	0.917	0.945
	Lnorm+PLDA	5.81	0.905	0.919	7.59	0.892	0.900	6.97	0.900	0.908
7	GMM-UBM	15.41	0.927	1.000	28.81	0.944	1.000	23.06	0.961	1.000
	GSV-SVM	10.02	0.972	1.000	16.59	0.962	1.000	13.60	0.986	0.998
	JFA	**4.25**	0.955	0.958	9.39	0.935	0.989	6.92	0.970	0.973
	PLDA	5.13	0.721	0.771	9.37	0.902	0.921	7.41	0.842	0.848
	Lnorm+PLDA	4.82	**0.748**	**0.821**	**7.91**	**0.814**	0.875	**6.64**	**0.814**	**0.846**
8	GMM-UBM	9.88	0.942	1.026	11.55	0.947	1.052	10.96	0.949	1.043
	GSV-SVM	2.91	0.760	0.923	4.06	0.689	0.776	3.67	0.728	0.828
	JFA	**1.61**	0.447	**0.555**	**2.43**	**0.496**	**0.519**	**2.15**	**0.499**	**0.534**
	PLDA	2.02	0.661	0.806	3.22	0.758	0.773	2.79	0.757	0.786
	Lnorm+PLDA	1.64	**0.444**	0.599	2.72	0.553	0.575	2.33	0.537	0.586
9	GMM-UBM	4.70	0.734	1.000	6.58	0.708	1.000	5.95	0.736	1.000
	GSV-SVM	2.79	0.290	0.641	3.82	0.529	0.763	3.16	0.473	0.714
	JFA	1.25	0.068	0.692	1.30	0.337	0.827	1.44	0.257	0.772
	PLDA	1.76	0.323	0.427	1.31	0.639	0.819	1.48	0.624	0.654
	Lnorm+PLDA	**0.83**	**0.111**	**0.162**	**0.70**	**0.221**	**0.327**	**0.87**	**0.215**	**0.258**

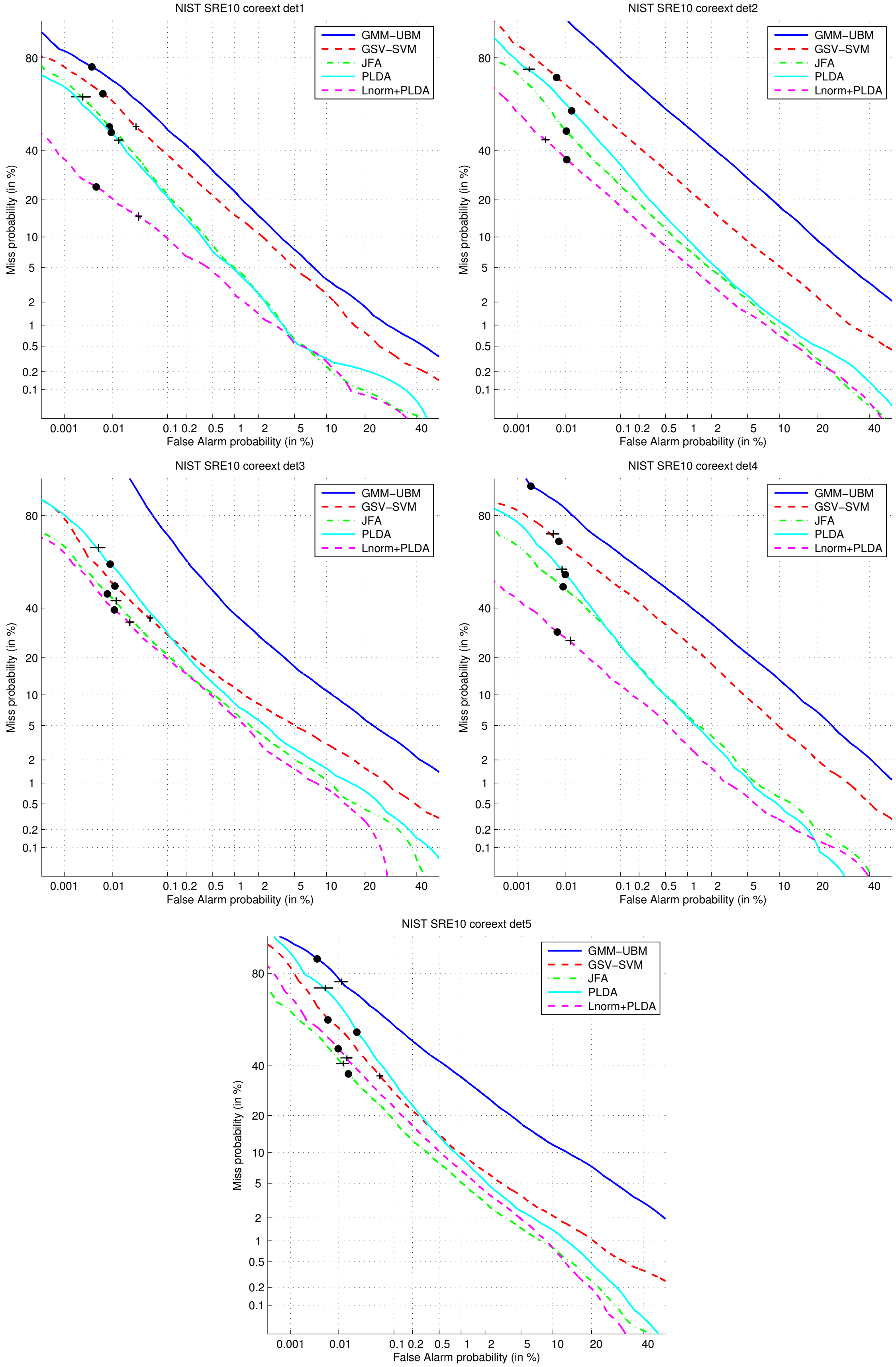


Figure 2.6: DET curves for core extended normal vocal effort common conditions.

evaluation we need to do several approximations (exact evaluation has large computational cost). First, we took MAP point estimates of the latent factors while in PLDA they are integrated out. Second we took the linear approximation of the ratio in (2.13).

If we compare male and female results, we see that the usual tendency of males presenting better performance than females was also observed here. That happens for all the systems and common conditions evaluated.

From GMM-UBM to PLDA, we witness and improvement of the calibration reaching actual DCF very near of the minimum DCF in most of the common conditions. The worse calibration happened in det1 (same microphone trials). That is because we trained a generic calibration for all the microphone against microphone conditions. Most of the trials used to train that calibration belong to condition det2 (different microphones trials) so the calibration is biased towards that condition. Besides in det2, we found two kind of trials with different score distributions: lavalier microphone and far-field microphones trials. To attain a better calibration, we would need to know the type of microphone (given or automatically extracted) and apply different calibration functions for each case.

In NIST SRE10, new conditions emulating high and low vocal effort phonecalls were added. During these calls the subject wore headphones which provide isolation from ambient room noise. White noise was introduced through the headphones to cause the participant to increase their vocal effort (det6, det7). For the low vocal effort (det8, det9) subjects heard their own voice through the headphone which automatically makes them to reduce their effort. DET curves for these conditions are shown in Figure 2.7. Results proved that high vocal effort strongly degrades performance compared to normal vocal effort. On the other side, low vocal effort condition presented very good results. We conclude that either low vocal effort does not affect performance or the procedure used to provoke the speaker to lower his vocal effort was unsuccessful.

Summing up, results prove that the mayor advances in speaker recognition came from the inclusion of inter-session compensation and speaker models based on eigen-voices. The system based on length normalized i-vectors and PLDA achieved the best overall performance across conditions. Besides, PLDA provided scores easier to calibrate what allowed us to obtain actual costs close to the minimum costs. That was especially significant for the interview conditions.

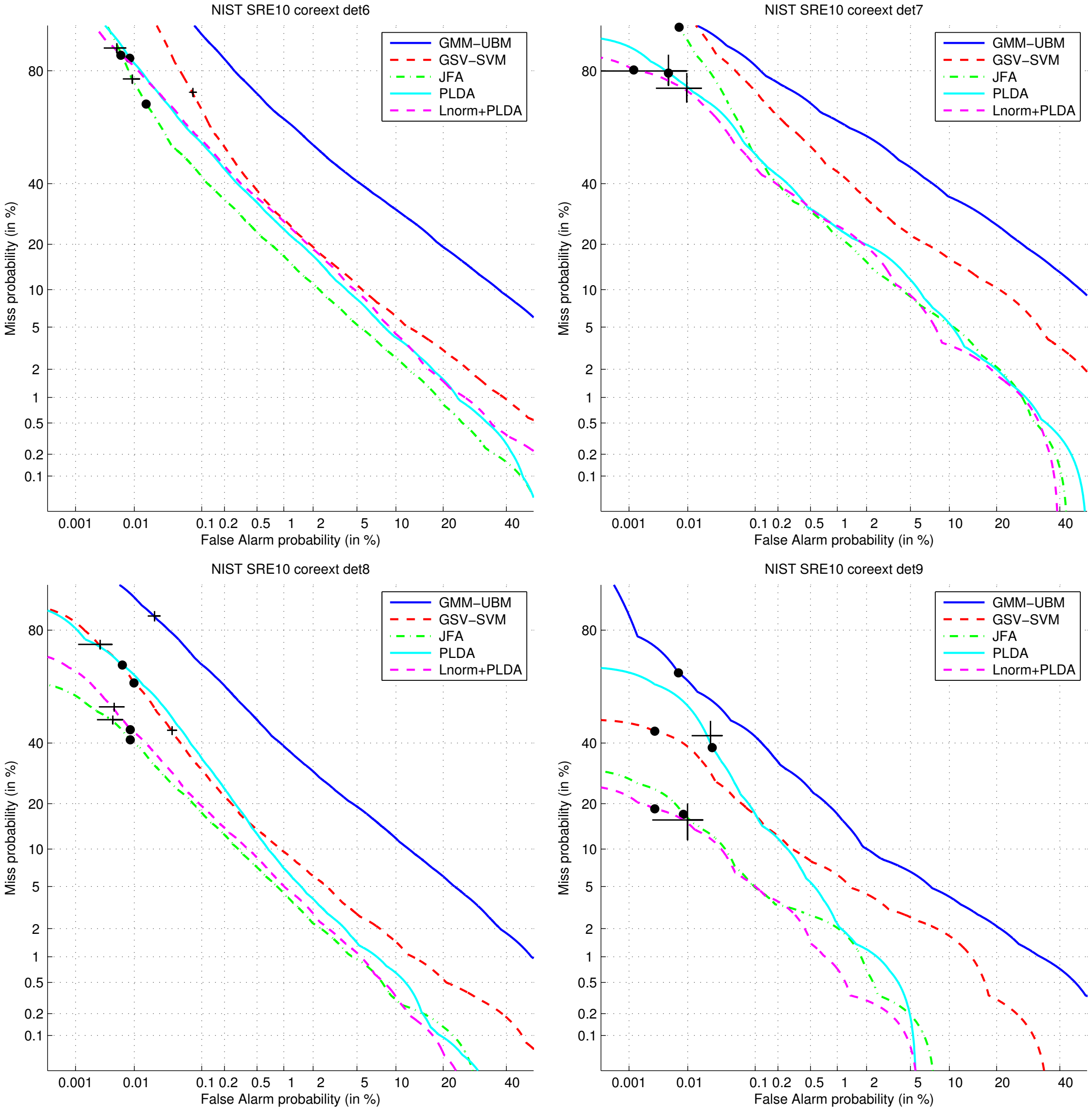


Figure 2.7: DET curves for core extended high (det6–7) and low (det8–9) vocal effort common conditions.

2.8.2.2 Score normalization analysis

Score normalization has been an essential part of speaker recognition systems from the beginnings of this technology. Recent advances like length normalization of i-vectors [Garcia-Romero and Espy-Wilson, 2011] or the heavy-tailed PLDA model [Kenny, 2010] have shown to produce well behaved scores and sparing the score normalization step. Table 2.2 compares JFA and PLDA systems with and without score normalization on common conditions det2 and det5. The table proves that JFA definitely need score normalization to obtain good performance. The PLDA system with raw i-vectors benefits from score normalization in the EER operating point but not so much in terms of DCF. Finally, the PLDA with length normalized i-vectors presents better results without score normalization in terms of both EER and DCF. The possibility of eliminating score normalization is a great advantage. On the one hand, we no longer need a cohort of impostors to compute the normalization parameters. And on the other hand, trial evaluation is faster because we save to score the trial segments against the cohorts.

Table 2.2: EER(%)/MinDCF/ActDCF NIST SRE10 core extended det5 with and without score normalization.

CC	System	without score norm.			with score norm.		
		EER	MinDCF	ActDCF	EER	MinDCF	ActDCF
2	JFA	6.82	0.868	0.996	3.28	0.594	0.928
	PLDA	4.21	0.578	0.778	3.40	0.718	0.778
	Lnorm+PLDA	**2.57**	**0.465**	**0.489**	**2.74**	**0.588**	**0.592**
5	JFA	4.90	0.732	0.985	**2.51**	**0.518**	**0.534**
	PLDA	5.54	0.627	0.806	3.41	0.781	0.802
	Lnorm+PLDA	**3.03**	**0.578**	**0.581**	3.10	0.656	0.659

2.9 Summary

In this chapter, we reviewed the evolution of the state of the art of text-independent speaker verification technology during the last years with special focus on systems based on short-term spectral features. We described all the blocks of a speaker verification system such as feature extraction, frame selection, feature normalization, statistic modeling, score normalization and calibration. Regarding the statistic modeling we focused on approaches that go from GMM-UBM with classical MAP speaker enrollment to the more advanced i-vector systems with PLDA based classification passing through Gaussian super-vector SVM and joint factor analysis.

We presented a comparative of the performance of different modeling approaches on the NIST SRE10 core extended condition (GMM-UBM, GSV-SVM, JFA and PLDA). Results proved the importance of techniques, such as JFA or i-vectors, that take into account inter-session variability compensation; at the same time that allow speaker enrollment with limited amount of data. We showed that the combination of i-vectors, length normalization and PLDA is, in general, the best approach given that it provides a very high performance as much in telephone as in microphone channels.

Speaker verification systems similar to those presented here will be used as baselines in the following chapters of this thesis.

Part II

Quality Measures and Reliability

Chapter 3

Reliability of the Speaker Verification Decisions

3.1 Introduction

In some situations, the quality of the signals involved in the speaker verification process is not as good as needed to take a reliable decision. Speaker verification performance is influenced by multiple factors: additive noise, reverberation, speaker's health, age and emotional state, type of microphone, transmission channel of the audio signal, language, amount of speech, etc. These factors can alter the scores produced by the speaker verification system in such manner that impostors are able to obtain higher scores and target speakers obtain lower scores than in optimum conditions. Thus, false alarm and misses rates increase.

It is well known that additive and convolutional noises greatly affect the distribution of cepstral features and thus, speaker verification [Ferrer et al., 2011]. Examples of additive noise are the sound of the air conditioning, the noise of a car engine, speech from a speaker different than the target speaker, babble noise from a crow, etc. Convolutional noise or reverberation depends on the physical characteristics of the room where the voice is recorded as well as the frequency responses of the elements present in the transmission channel like microphones or signal processors. Noise can be stationary if its characteristics do not change over time, or non-stationary if they are time dependent.

Emotion mismatch between enrollment and test segments also causes a drop of performance. There are some works treating this topic like [Li et al., 2005] where neutral enrollment speech is transformed according to statistical prosodic patterns of emotion utterances and several speaker models are trained on the converted speech. More recently, in [Yang and Chen, 2012], authors propose to compensate the deformations introduced by emotion at feature, model and score levels. In Chapter 2, we found a clear example of how emotional state affects performance in the results the high vocal effort conditions of NIST SRE10 [NIST Speech Group, 2010]. Table 2.1 evidences that performance degrades by 100% in terms of EER and by 60% in terms of DCF between normal (det5) and high vocal effort (det6) telephone speech.

The effect of age on speaker recognition is also analyzed on several works. In [Lei and Hansen, 2009], the authors add a term to JFA to account for age variability achieving some improvement on NIST SRE08. In [Kelly and Harte, 2011] the effect of age on speaker recognition is measured by using recordings of celebrities in a time span of 30 years. Their conclusions are that the SV score of the target trials starts to degrade when the time between enrollment and test exceeds 5 years. Besides, they observed an acceleration in score drop-off when the subject is above 60 years. The work in [Doddington, 2012] addresses the issue of age difference between target and non-target speaker populations. It manifests that, at a fixed miss probability ($P_{\text{Miss}} = 1\%$), the false alarm probability reduces substantially as the age difference increases. False alarm probability significantly decreases for age differences of as little as five years, with an order of magnitude reduction for age differences of forty years or more.

The effect of language on performance can be tested on NIST SRE06 and SRE08 evaluations [NIST Speech Group, 2006,NIST Speech Group, 2008] where we find evaluation conditions with English only trials and mixed language trials. In [Villalba et al., 2008], EER degraded by 77% between NIST SRE08 English and mixed trials. In [Lu et al., 2009], language factors are added to JFA model to compensate for language variability.

Speaker verification is not only affected by the mismatch between enrollment and test segments but also by the mismatch between development and evaluation data. That means that the speech that we employ to train UBM, JFA, i-vector extractors, PLDA, score calibration needs to be similar to the speech of the enrollment and test recordings. We saw an example of this in Section 2.8 where we applied different calibrations depending on the kind of trial (phn-phn, mic-phn and mic-mic) under test. We find another example in [Lei et al., 2012], there the problem of noise is addressed by including signals with artificially added noise in the i-vector extractor and PLDA training. For this reason, having a measure of the similarity between development and evaluation data can provide information about the reliability of the SV score.

In this part of the thesis, we investigate methods to compute a probabilistic measure of the reliability of the SV decisions in scenarios similar to the ones above described. We infer the trial reliability by combining the SV score and a group of quality measures extracted from the trial segments. The quality measures selected are related with the type of scenarios where the system is going to work. For example, we would use the signal-to-noise ratio if the system has to work in noisy environments. We mainly focus on degradations derived from the recording channel or device like additive noise and reverberation, although the methods presented here could be extended to other types of issues like age and language mismatch by adding quality measures related to them.

We intend to use the reliability measure to discard unreliable trials, that is, instead of classifying them as target or non-target, we say that the speaker verification decisions cannot be trusted. The motivation for this work came from companies dedicated to commercialize speaker verification technology whose customers demand this possibility. This approach has utility for applications that must provide very accurate decisions but that do not need to provide a decision for all the trials. An example would be a forensic application where we have several recordings that can prove the guilt of a criminal. The verdict of the court should be only based on the ones that provide a reliable evidence. Another application can be telephonic access to bank accounts where, in case of determining that the utterance is unreliable, we can ask the client to repeat the sentence.

In this chapter, we review the previous works and describe the experimental setup that is common to the rest of chapters of this part. Section 3.2 describes confidence measures used in the literature, which are derived from the classifier score or from quality measures of the speech signal or from combining both. Section 3.3 describes different criteria to compare reliability estimators. These include computing costs and EER on trials with a given reliability level or on trials with reliability larger than a threshold. We also define a extended cost function that penalizes to classify correct SV decisions as unreliable. In Section 3.4, we describe our SV system, based on i-vectors, and our databases.

Regarding the rest of this part, Chapter 4 describes the quality measures that we used in our experiments. In Chapter 5, we revisit the work in [Richiardi et al., 2006a] where Bayesian networks are used to infer the trial reliability. We tried new quality measures and studied how modifications of the BN structure affected the results. In Chapter 6 we present a totally new approach, also based on a Bayesian network, and prove that it outperform previous works. Finally, in Chapter 7, we take a Bayesian network trained for an environment with a large amount of development data and adapt it to domains with scarce development data by Bayesian *Maximum a posteriori* estimation.

3.2 Previous Works

In the last years, several works have proposed methods to decide the reliability of speaker verification trials. We can divide these methods into three groups. First, we find approaches based on deriving some confidence from the SV score. The SV score is itself a reliability measure. The higher it is the more reliable the target decision and the lower the more reliable the non-target decision. In most systems, the score is a log-likelihood ratio and if it is well-calibrated, we can say that scores near zero mean that the trial is non-reliable.

Other works rely on measures extracted from the enrollment and test segments. These measures carry information about the acoustical conditions of the recordings like noise and reverberation; segment duration, etc. Finally, there are procedures that combine the SV score and quality measures to provide a global reliability estimation.

3.2.1 Confidence from the classifier output

Numerous works compute confidence measures from the classifier output. The confidences are used for multiple purposes such us rejecting models, rejecting trials or fusing systems based on different features and biometric modalities. A Bayesian measure of confidence can be defined as the posterior probability that the verification decision is correct given the score [Gish and Schmidt, 1994]:

$$P\left(\text{correct}|s\right)=\frac{P_{\text{correct}}P\left(s|\text{correct}\right)}{P_{\text{correct}}P\left(s|\text{correct}\right)+\left(1-P_{\text{correct}}\right)P\left(s|\text{wrong}\right)} \tag{3.1}$$

where P_{correct} is the prior probability for correct classification, $P\left(s|\text{correct}\right)$ is the score distribution for correct classifications and $P\left(s|\text{correct}\right)$ for wrong classification. These distributions are estimated from a development set. First, an operating point (threshold) must be chosen to determine the correct and the wrong trials. The prior P_{correct} can be determined from the percentage of errors in the development set. However, if the test set has worst conditions than the development set, the confidence will be biased too high by the prior. We could compensate that by choosing non-informative priors or subjective priors based on what we expect from the test dataset. The score distributions can be modeled by mixtures of Gaussians.

Scores were used to determine the reliability of the speaker model in [Koolwaaij et al., 2000]. Model confidence is based on the distance between the scores of that model evaluated on the own speaker training material and on a cohort of impostors. That is

$$d = \frac{\max(0, \mu_{\mathcal{T}} - \mu_{\mathcal{N}})}{\sigma_{\mathcal{N}}} \tag{3.2}$$

where $\mu_{\mathcal{T}}$, $\mu_{\mathcal{N}}$ and $\sigma_{\mathcal{N}}$ are the means and standard deviation of the target and non-target scores for the given model. If the distance is too low the model is considered unreliable and the speaker needs to be re-enrolled.

Another Bayesian confidence measure can be just the posterior of the target hypothesis given the SV score s [Nakasone and Beck, 2001]. They fit a logistic function to the posterior probability:

$$P\left(\mathcal{T}|s\right) = \frac{1}{1 + \exp\left(-(\alpha s + \beta)\right)} \tag{3.3}$$

where the parameters α and β are computed by least squares regression. A similar idea was used in [Brummer and Preez, 2006], where scores are calibrated to become meaningful log-likelihood ratios by linear logistic regression. Then, the posterior probability can be computed by applying Bayes rule:

$$P\left(\mathcal{T}|\mathcal{D}\right) = \frac{P_{\mathcal{T}} P\left(\mathcal{D}|\mathcal{T}\right)}{P_{\mathcal{T}} P\left(\mathcal{D}|\mathcal{T}\right) + (1 - P_{\mathcal{T}}) P\left(\mathcal{D}|\mathcal{N}\right)} = \frac{1}{1 + \exp\left(-s + \log\left(\frac{1-P_{\mathcal{T}}}{P_{\mathcal{T}}}\right)\right)} \tag{3.4}$$

where $\mathcal{D}$ is the trial data and $P_{\mathcal{T}}$ is the target prior. Well calibrated log-likelihood ratios are application independent while the SV application is associated to the prior $P_{\mathcal{T}}$. This makes the latter method more general because just by changing the value of $P_{\mathcal{T}}$ in (3.4) we obtain a posterior adapted to a new application. On the contrary, Nakasone's method needs to fit a new logistic function each time that we change the priors. Brummer's approach has become the standard method to calibrate SV scores as was explained in Chapter 2.

In [Bengio et al., 2002], if Gaussian score distributions are assumed, authors propose to compute the difference between the probabilities of the score given the target and non-target distributions:

$$\mathrm{CM} = \left|\mathcal{N}\left(s|\mu_{\mathcal{T}}, \sigma^2_{\mathcal{T}}\right) - \mathcal{N}\left(s|\mu_{\mathcal{N}}, \sigma^2_{\mathcal{N}}\right)\right| \ . \tag{3.5}$$

In case that the scores does not fit to Gaussian distributions, they propose a non parametric model. The score space is quantized in a way that each level has the same number of training samples. The error rate (sum of misses and false rejections) for the scores in each quantization level is computed. That is a simple quality measure of the scores in each level. Target and impostor distributions are trained on a development set. Authors apply this confidence measure to improve the fusion of speaker and face recognition modalities. These measures are substituted in [Poh and Bengio, 2005] by another one that is defined as the difference between the miss rate and false acceptance rate for a certain score (taken as threshold):

$$\mathrm{CM} = \left|P_{\mathrm{Miss}}(s) - P_{\mathrm{FA}}(s)\right| \ . \tag{3.6}$$

Thus, the closer the score is to EER operating point, the lower the confidence.

The correlation coefficients between the score and the target and non-target distributions were used in [Mengusoglu, 2004]. A confidence measure is defined as the difference between both coefficients

$$\mathrm{CM} = r_{\mathcal{T}} - r_{\mathcal{N}} \tag{3.7}$$

where r is computed by applying the inverse Fisher z-transform [Hotelling, 1953] to the scores normalized by the mean and standard deviation of the corresponding distribution:

$$r_{\mathcal{T}} = \frac{\exp(2z_{\mathcal{T}}) - 1}{\exp(2z_{\mathcal{T}}) + 1} \qquad \text{with} \qquad z_{\mathcal{T}} = \frac{s - \mu_{\mathcal{T}}}{\sigma_{\mathcal{T}}} \tag{3.8}$$

$$r_{\mathcal{N}} = \frac{\exp(2z_{\mathcal{N}}) - 1}{\exp(2z_{\mathcal{N}}) + 1} \qquad \text{with} \qquad z_{\mathcal{N}} = \frac{s - \mu_{\mathcal{N}}}{\sigma_{\mathcal{N}}}\,. \tag{3.9}$$

A confidence value near zero means that we cannot take a reliable decision while values near 2 or -2 means that the trial is target or non-target with a high reliability.

In [Zheng et al., 2007], authors define a quantity called speaker discrimination power P for a SV trial as

$$P = \frac{\log P\left(\mathcal{D}|\mathcal{T}\right) - \log P\left(\mathcal{D}|\mathcal{N}\right)}{\left|\log P\left(\mathcal{D}|\mathcal{N}\right)\right|} \tag{3.10}$$

that is the log-likelihood ratio normalized by the log-likelihood of the UBM. The normalization term $|\log P\left(\mathcal{D}|\mathcal{N}\right)|$ aims to equalize the value of P for different SV systems. Then, P is used to fuse two systems given more importance to the one with higher discrimination power.

3.2.1.1 Confidence from quality measures of the speech signal

Some works base the confidence on the SV decisions on auxiliary information that can be computed from the speech utterances. This information is usually referred as quality measures in the literature. Examples of quality measures are utterance durations and signal-to-noise ratio given that it is well known that short utterances and noisy environments reduce the speaker recognition accuracy. In [Garcia-Romero et al., 2004], authors describe a framework to take advantage of quality measures at different levels of the SV process: model, score computation and fusion. They propose a frame-level quality measure based on the deviation of the fundamental frequency from the mean. The likelihood ratio for the GMM-UBM approach is computed by weighting each frame differently according to the quality measure. That work was extended in [Garcia-Romero et al., 2006] to include measures like SNR and the ITU P.563 objective speech quality assessment [ITU-T, 2004]. Besides, they implement a quality based score fusion scheme where there are two SVM based fusers , one for low quality and another for high quality trials, and they are weighted differently depending on the quality measures.

In [Solewicz and Koppel, 2005], three types of degradations were measured: communication channel, speaker style and speaker stress. Channel characteristics are measured by the mean and variance of the long-term spectrum of the conversations. They add the likelihood of the utterance frames given the UBM. Low likelihoods are expected for unseen channels pointing that the test segment is not well modeled by the UBM. Stylistic attributes are measured by means, ranges and symmetry of pitch, energy distributions and speaking rate. Finally, the Teager Energy Operator (TEO) [Zhou et al., 2001] is applied as indicator of speaker stress. They computed mean and variances of TEO coefficients in six critical bands. All these measures are applied to do selective fusion of speaker recognizers based on different features (spectral, phonetic, prosodic and idiolectal). Measures are clusterized by k-means and a different SVM fusers are trained for each cluster.

As well as SNR, high-order statistic of speech such us skewness and kurtosis were used in [Richiardi and Drygajlo, 2008]. Authors evaluate the correlation and mutual information between the SV score and each quality measure. Measures are evaluated in two databases: BANCA and XM2VTS. Results show different correlations values for target and non-target trials proving that degradations affect more to targets than to impostors. The comparative between quality measures is not conclusive. While, for BANCA, we find higher correlations for SNR than for high-order statistics, for XM2VTS, we find the opposite.

In [Harriero et al., 2009], SNR, ITU P.563, UBM log-likelihood and kurtosis of the LPC coefficients are analyzed. Authors observe a clear correlation between EER and the quality measures on NIST SRE 2006 and 2008 datasets.

3.2.1.2 Confidence from classifier score and quality measures

Seeing that both previous groups of confidence measures help to discriminate between right and wrong classified trials, the logical step forward is combining them into a unique value. In [Campbell et al., 2005], the SV score, numerator and denominator of the likelihood ratio, SNR, utterance durations and channel labels feed a multilayer perceptron to obtain a confidence for each score. In this case, the confidence represents the posterior probability for the target hypothesis.

Bayesian networks (BN) applied to obtain a probabilistic reliability measure were introduced in [Richiardi et al., 2005]. The BN establishes the causal relationships between random variables intervening in the SV process such as the SV score, quality measures, trial label, trial decision and reliability. This relationships allow us to compute the posterior probability for the trial reliability. This model will be discussed more thoroughly in the next chapter. In this work, the signal-to-noise ratio is used as quality measure. In case of low reliability, the system asks the user to utter a new sentence and chooses the one with higher reliability to provide the decision. In [Richiardi et al., 2006b, Richiardi et al., 2006a], the BN based approach is compared with previous works [Nakasone and Beck, 2001, Bengio et al., 2002, Poh and Bengio, 2005]. Authors conclude that Bayesian networks outperform previous approaches given the possibility of integrating multiple sources of information. The reliability estimation from the BN can also be applied to fuse speaker and facial biometric modalities [Kryszczuk et al., 2007].

3.3 Assessment of the Performance of the Reliability Estimator

This part of the thesis aims to obtain a reliability measure for speaker verification decisions that we can apply to discard unreliable trials. We have seen that there are multiple real world applications like forensics that can benefit of this possibility. In the following chapters, we will present several approaches to estimate the trial reliability based on Bayesian networks.

To compare them, we need performance measures able to assess which one is better. In this section, we attend that matter.

3.3.1 General definitions

Let $\theta \in \{\mathcal{T},\mathcal{N}\}$ denote the labeling of a given speaker verification trial where $\mathcal{T}$ represents the hypothesis that the trial is target and $\mathcal{N}$ that it is non-target. We denote by $\hat{\theta}$ the hard decision taken by applying a threshold ξ_θ to the SV score s:

$$\hat{\theta}(s,\xi_\theta) = \begin{cases} \mathcal{T} & \text{if } s \geq \xi_\theta \\ \mathcal{N} & \text{if } s < \xi_\theta \end{cases} \quad . \tag{3.11}$$

If the decision is right ($\theta = \hat{\theta}$), we say that the trial is reliable given the SV system, and we say that it is unreliable otherwise. We denote by $R \in \{\mathcal{R},\mathcal{U}\}$ the trial reliability where $\mathcal{R}$ means that the trial is reliable and $\mathcal{U}$ that it is unreliable. Moreover, $\hat{R}$ denotes the hard decision of the reliability estimator. The reliability detection systems proposed in the following chapters provide a score that is the log-likelihood ratio

$$\mathrm{LLR}_R = \mathrm{logit} P\left(R = \mathcal{R}|s,\mathbf{Q}\right) = \log \frac{P\left(R = \mathcal{R}|s,\mathbf{Q}\right)}{1 - P\left(R = \mathcal{R}|s,\mathbf{Q}\right)} \tag{3.12}$$

where $P\left(R = \mathcal{R}|s,\mathbf{Q}\right)$ is the posterior probability for reliable trial given the SV score and the quality measures. $\hat{R}$ is obtained by thresholding LLR_R:

$$\hat{R}(\mathrm{LLR}_R,\xi_R) = \begin{cases} \mathcal{R} & \text{if } \mathrm{LLR}_R \geq \xi_R \\ \mathcal{U} & \text{if } \mathrm{LLR}_R < \xi_R \end{cases} \quad . \tag{3.13}$$

3.3.2 Error rates and DCF on reliable trials

The first method that we use to compare reliability detectors consist of computing error rates by only counting the trials classified as reliable. This method was already proposed in [Grother and Tabassi, 2007].

Let $N_{\hat{\mathcal{R}}\mathcal{T}}(\xi_R)$ and $N_{\hat{\mathcal{R}}\mathcal{N}}(\xi_R)$ be the number of target and non-target trials classified as reliable by the reliability detector. Let $N_{\mathrm{Miss}\hat{\mathcal{R}}}(\xi_\theta,\xi_R)$ and $N_{\mathrm{FA}\hat{\mathcal{R}}}(\xi_\theta,\xi_R)$ the number of misses and false alarms that are incorrectly classified as reliable. Then, miss and false alarm rates on reliable trials are defined as:

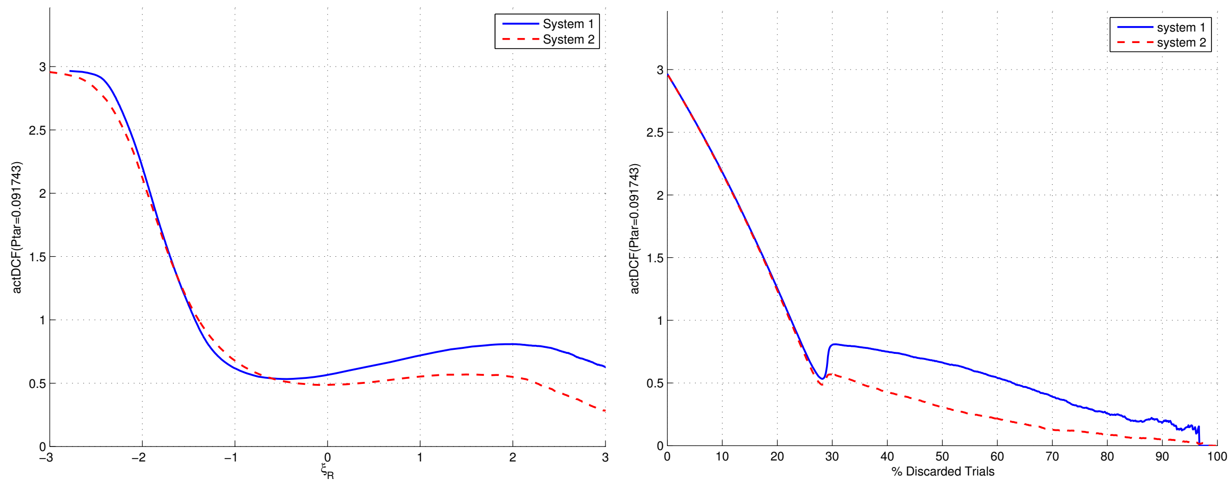
$$P_{\mathrm{Miss}\hat{\mathcal{R}}}(\xi_\theta,\xi_R) = \frac{N_{\mathrm{Miss}\hat{\mathcal{R}}}(\xi_\theta,\xi_R)}{N_{\hat{\mathcal{R}}\mathcal{T}}(\xi_R)} \qquad P_{\mathrm{FA}\hat{\mathcal{R}}}(\xi_\theta,\xi_R) = \frac{N_{\mathrm{FA}\hat{\mathcal{R}}}(\xi_\theta,\xi_R)}{N_{\hat{\mathcal{R}}\mathcal{N}}(\xi_R)} \; . \tag{3.14}$$

Now, we can redefine the EER and DCF on $P_{\mathrm{Miss}\hat{\mathcal{R}}}$ and $P_{\mathrm{FA}\hat{\mathcal{R}}}$. The EER on reliable trials for a fixed reliability threshold ξ_R is

$$\mathrm{EER}_{\hat{\mathcal{R}}}(\xi_R) = P_{\mathrm{Miss}\hat{\mathcal{R}}}(\xi_\theta,\xi_R) \qquad \text{if} P_{\mathrm{Miss}\hat{\mathcal{R}}}(\xi_\theta,\xi_R) = P_{\mathrm{FA}\hat{\mathcal{R}}}(\xi_\theta,\xi_R) \; . \tag{3.15}$$

The detection cost function is defined as:

$$C_{\hat{\mathcal{R}}}(\xi_\theta,\xi_R) = C_{\mathrm{Miss}} P_{\mathcal{T}} P_{\mathrm{Miss}\hat{\mathcal{R}}}(\xi_\theta,\xi_R) + C_{\mathrm{FA}}(1 - P_{\mathcal{T}}) P_{\mathrm{FA}\hat{\mathcal{R}}}(\xi_\theta,\xi_R) \tag{3.16}$$



(a)

(b)

Figure 3.1: % Example curves depicting actual DCF against reliability threshold ξ_R (a) and % discarded trials (b).

where C_{Miss} and C_{FA} are the miss and false alarm costs and $P_{\mathcal{T}}$ the target prior. We can also employ the normalized version:

$$C_{\text{Norm}\hat{\mathcal{R}}}(\xi_\theta, \xi_R) = C_{\hat{\mathcal{R}}}(\xi_\theta, \xi_R) / \min(C_{\text{Miss}} P_{\mathcal{T}}, C_{\text{FA}}(1 - P_{\mathcal{T}})) \; . \tag{3.17}$$

Note that the classical DCF in (2.21) is equivalent to this new DCF if all the trials are considered reliable, that is $C(\xi_\theta) = C_{\hat{\mathcal{R}}}(\xi_\theta, -\infty)$.

With these error rates and costs, we can create several graphical representations of the reliability detector performance, for example, DET curves. Fixing a SV system, we could compare the DET curves that we obtain from a reliability detector for different thresholds ξ_R. Other option is to compare the DET curves of different detectors for a fixed threshold ξ_R.

A drawback of DET curves is that, being parametric in threshold ξ_θ, they do not show the dependence of the error rates with ξ_R at a fixed ξ_θ. That is important because, in a real application, the expert will fix the threshold ξ_θ during the development phase by calibrating the scores on a database of controlled quality. Probably, the system operator will not be able to change the threshold and, if he does, he may not know how to do it effectively. For this reason, we want to evaluate if we can contain error rates by applying the reliability detector while maintaining the SV threshold ξ_θ set during the system development. The best way to measure this is through the actual normalized DCF $C_{\text{Norm}\hat{\mathcal{R}}}$ so it will be our primary performance indicator.

We propose to evaluate reliability detectors by comparing curves of $C_{\text{Norm}\hat{\mathcal{R}}}$ against the reliability threshold ξ_R and against the percentage of discarded trials. We chose the value of the SV threshold ξ_θ to minimize Bayes risk by assuming calibrated scores ($\xi_\theta = -\text{logit} P_{\mathcal{T}}$), and we kept it constant for all our experiments. The criterion to identify the best reliability detector on this curves is that the best detector should reduce the actual DCF as much as possible at the same time that it discards the lowest number of trials. Thus, we desire curves as much near of the origin as possible. The example in Figure 3.1 compares two systems exposing that system 2 outperforms system 1.

Observing the figure, we can think that the behavior of the system 1 curves is counter-intuitive. We refer to the fact that we expect that as we discard trials the DCF should always descend. However, the curves do not always decrease as we discard trials. An ideal reliability detector would provide monotonically decreasing curves. However, our reliability detector is just another pattern classifier that makes errors, the same as the speaker recognizer. When the reliability detector rejects correct trials the denominators in (3.14) decrease while the numerator do not and error rates increase.

We can verify what conditions need to hold to reduce the error rate. Let $P_{\mathrm{E}\hat{\mathcal{R}}}$ denote indistinctly the error rates $P_{\mathrm{Miss}\hat{\mathcal{R}}}$ or $P_{\mathrm{FA}\hat{\mathcal{R}}}$. It is computed as

$$P_{\mathrm{E}\hat{\mathcal{R}}}(\xi_\theta,\xi_R)=\frac{N_{E\hat{\mathcal{R}}}(\xi_\theta,\xi_R)}{N_{C\hat{\mathcal{R}}}(\xi_\theta,\xi_R)+N_{E\hat{\mathcal{R}}}(\xi_\theta,\xi_R)} \tag{3.18}$$

where $N_{C\hat{\mathcal{R}}}$ and $N_{E\hat{\mathcal{R}}}$ are, respectively, the number of correct and wrong trials classified as reliable. We can also write $P_{\mathrm{E}\hat{\mathcal{R}}}$ as

$$P_{\mathrm{E}\hat{\mathcal{R}}}(\xi_\theta,\xi_R)=\frac{N_E(\xi_\theta)P_{\mathrm{FR}}(\xi_\theta,\xi_R)}{N_C(\xi_\theta)(1-P_{\mathrm{FU}}(\xi_\theta,\xi_R))+N_E(\xi_\theta)P_{\mathrm{FR}}(\xi_\theta,\xi_R)} \tag{3.19}$$

where N_C and N_E are the total number of right and wrong classified trials. P_{FU} is the false unreliable rate– the probability of classifying a correct trial as unreliable–; and P_{FR} is the false reliable rate–the probability of classifying a wrong trial as reliable. P_{FU} and P_{FR} depend on the SV decision and reliability thresholds. A high ξ_R produces high P_{FU} and low P_{FR}, and vice versa. From (3.19), we derive that, given two thresholds ξ_{R_1} and ξ_{R_2},

$$P_{\mathrm{E}\hat{\mathcal{R}}}(\xi_\theta,\xi_{R_1})>P_{\mathrm{E}\hat{\mathcal{R}}}(\xi_\theta,\xi_{R_2}) \iff \frac{P_{\mathrm{FR}}(\xi_\theta,\xi_{R_1})}{1-P_{\mathrm{FU}}(\xi_\theta,\xi_{R_1})}>\frac{P_{\mathrm{FR}}(\xi_\theta,\xi_{R_2})}{1-P_{\mathrm{FU}}(\xi_\theta,\xi_{R_2})}\,. \tag{3.20}$$

For the particular case of comparing the error rates discarding and not discarding trials ($\xi_R=-\infty$), to obtain an improvement from the reliability detector we need that

$$P_{\mathrm{E}\hat{\mathcal{R}}}(\xi_\theta,\xi_R)<P_E(\xi_\theta) \iff P_{\mathrm{FU}}(\xi_\theta,\xi_R)+P_{\mathrm{FR}}(\xi_\theta,\xi_R)<1\,. \tag{3.21}$$

These equations prove that if, as we increase ξ_R, P_{FU} increases faster than P_{FR} decreases, we will obtain higher and higher $P_{\mathrm{Miss}\hat{\mathcal{R}}}$ and $P_{\mathrm{FA}\hat{\mathcal{R}}}$; and therefore higher costs. That is what we see in the figure.

The fact that the cost can increase as we discard more trials poses a problem to select the operating point of the reliability detector. For example, if our application allows us to reject 80% of the trials and we set the threshold ξ_R according to that, we could get worse performance than discarding just 50% of the trials. Therefore, we should be careful when setting ξ_R. A possible solution is to choose the ξ_R that minimizes the cost restricted to rejecting less than 80% of trials.

Another graphical representation could be DET curves for the reliability detector, instead of DET curves for the speaker detector. That is plotting $P_{\mathrm{FU}}(\xi_\theta,\xi_R)$ against $P_{\mathrm{FR}}(\xi_\theta,\xi_R)$ by sweeping the value of ξ_R and fixing ξ_θ. However, these curves do not provide direct information about how much the SV accuracy improves, or even, if it really does. For that reason, we decided not to use this kind of representation.

An added drawback to this way of computing error rates is that it can drive to indeterminations. For example, once all the target trials are discarded P_{Miss} is undefined and we cannot continue computing costs even when there are still non-target trials available.

3.3.3 Error rates and DCF on trials with a given reliability level

Another option, also proposed in [Grother and Tabassi, 2007], consist of quantizing the score of the reliability detector and compute the error rates for the trials lying in each quantization level.

Quantization levels can be chosen in different manners. For example, in [Harriero et al., 2009], quantization levels are taken non-uniforms and overlapped. Each level includes 20% of the trials with shift of 1% of trials between levels.

Let r denote the quantized value of the reliability detector score LLR_R. We define the error rates for reliability level k as:

$$P_{\mathrm{Miss}}(\xi_\theta, r=k) = \frac{N_{\mathrm{Miss}}(\xi_\theta, r=k)}{N_{\mathcal{T}}(r=k)} \qquad P_{\mathrm{FA}}(\xi_\theta, r=k) = \frac{N_{\mathrm{FA}}(\xi_\theta, r=k)}{N_{\mathcal{N}}(r=k)} \tag{3.22}$$

where $N_{\mathcal{T}}(r=k)$ and $N_{\mathcal{N}}(r=k)$ are the number of target and non-target trials in level k; and, $N_{\mathrm{Miss}}(r=k)$ and $N_{\mathrm{FA}}(r=k)$ are the number of misses and false alarms.

Equivalently, we can compute the EER and DCF for level k:

$$\mathrm{EER}(r=k) = P_{\mathrm{Miss}}(\xi_\theta, r=k) \qquad \mathrm{if}\, P_{\mathrm{Miss}}(\xi_\theta, r=k) = P_{\mathrm{FA}}(\xi_\theta, r=k) \tag{3.23}$$

$$C(\xi_\theta, r=k) = C_{\mathrm{Miss}} P_{\mathcal{T}} P_{\mathrm{Miss}}(\xi_\theta, r=k) + C_{\mathrm{FA}}(1-P_{\mathcal{T}}) P_{\mathrm{FA}}(\xi_\theta, r=k)\ . \tag{3.24}$$

Useful graphical representations can be DET curves per reliability level or; EER, minimum and actual normalized DCF against the reliability level.

3.3.4 Extended detection cost function (C_Q)

We propose a novel performance measure specific for speaker verification systems that includes the possibility of discarding trials. This measure is based on the classical DCF where we add two new terms that account for the cost of the errors made by the reliability detector. We define the extended detection cost function C_Q as:

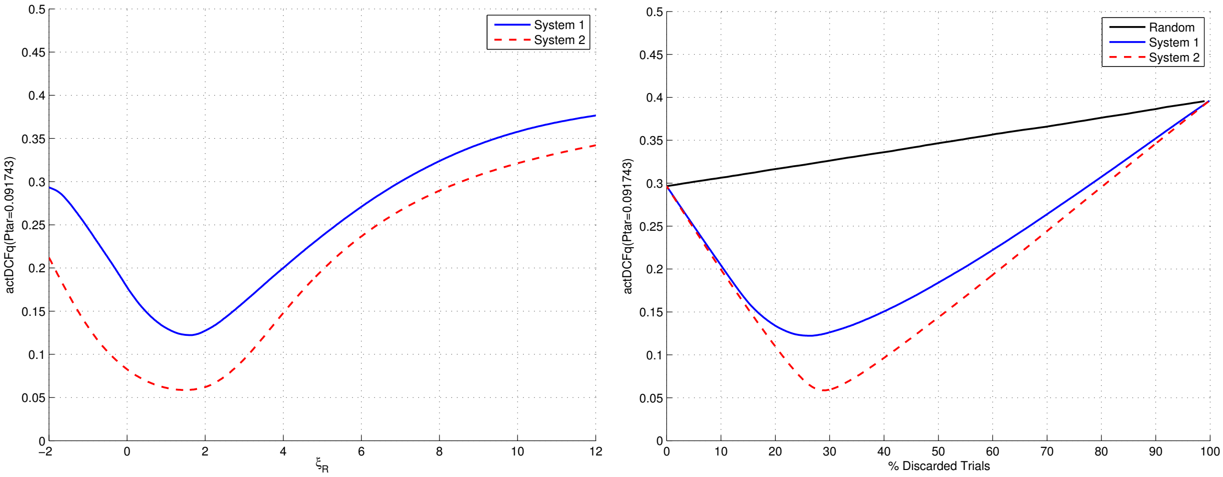
$$\begin{aligned} C_Q(\xi_\theta, \xi_R) =& C_{\mathrm{Miss}} P_{\mathcal{T}} P_{\mathrm{Miss}Q}(\xi_\theta, \xi_R) + C_{\mathrm{FA}}(1-P_{\mathcal{T}}) P_{\mathrm{FA}Q}(\xi_\theta, \xi_R) \\ &+ C_{\mathrm{FU}\mathcal{T}} P_{\mathcal{T}} P_{\mathrm{FU}\mathcal{T}Q}(\xi_\theta, \xi_R) + C_{\mathrm{FU}\mathcal{N}}(1-P_{\mathcal{T}}) P_{\mathrm{FU}\mathcal{N}Q}(\xi_\theta, \xi_R) \end{aligned} \tag{3.25}$$

where

$$\begin{aligned} P_{\mathrm{Miss}Q}(\xi_\theta, \xi_R) &= \frac{N_{\mathrm{Miss}\hat{\mathcal{R}}}(\xi_\theta, \xi_R)}{N_{\mathcal{T}}} & P_{\mathrm{FA}Q}(\xi_\theta, \xi_R) &= \frac{N_{\mathrm{FA}\hat{\mathcal{R}}}(\xi_\theta, \xi_R)}{N_{\mathcal{N}}} \\ P_{\mathrm{FU}\mathcal{T}Q}(\xi_\theta, \xi_R) &= \frac{N_{\mathrm{FU}\mathcal{T}}(\xi_\theta, \xi_R)}{N_{\mathcal{T}}} & P_{\mathrm{FU}\mathcal{N}Q}(\xi_\theta, \xi_R) &= \frac{N_{\mathrm{FU}\mathcal{N}}(\xi_\theta, \xi_R)}{N_{\mathcal{N}}}\ . \end{aligned} \tag{3.26}$$

$N_{\mathcal{T}}$ and $N_{\mathcal{N}}$ are total number of target and non-target trials. $N_{\mathrm{Miss}\hat{\mathcal{R}}}$ and $N_{\mathrm{FA}\hat{\mathcal{R}}}$ (already defined in Section 3.3.2) are the number of misses and false alarms bad classified as reliable. $N_{\mathrm{FU}\mathcal{T}}$ and $N_{\mathrm{FU}\mathcal{N}}$ are the number of false unreliable target and non-targets trials, that is, the number of discarded trials with correct SV decisions. The new costs $C_{\mathrm{FU}\mathcal{T}}$ and $C_{\mathrm{FU}\mathcal{N}}$ are the costs of false unreliable classification.

As for C_R, we will plot C_Q against the reliability detector threshold or against the percentage of discarded trials for a fixed value of ξ_θ. Figure 3.2 shows an example comparing two reliability detectors. System 2 is evidently better than system 1. The black curve represents the cost obtained by randomly discarding trials.



(a)

(b)

Figure 3.2: % Example curves depicting C_Q against reliability threshold ξ_R (a) and % discarded trials (b).

One of the advantages of this measure over the method of computing costs on reliable trials presented in Section 3.3.2 is that indeterminations are not possible. That is because the denominators in the error rates calculus are constant. Other advantage is that the operating point of the reliability detector can be established in a principled way. According to the specifications of the given application, we choose $P_{\mathcal{T}}$, C_{Miss}, C_{FA}, $C_{\text{FU}\mathcal{T}}$ and $C_{\text{FU}\mathcal{N}}$ and select the threshold

$$\xi_R^* = \arg\min_{\xi_R} C_Q(\xi_\theta, \xi_R)\ . \tag{3.27}$$

Given the ξ_R, we can determine the percentage of trials that we discard. So that C_Q makes sense, the cost of discarding a correct trial has to be lower than the cost of taking a bad SV decision. That means that $C_{\text{FU}\mathcal{T}} < C_{\text{Miss}}$ and $C_{\text{FU}\mathcal{N}} < C_{\text{FA}}$. We need to choose the costs carefully because if $C_{\text{FU}\mathcal{T}}$ and $C_{\text{FU}\mathcal{N}}$ are too low, we will obtain that C_Q is minimized by rejecting all the trials.

We can generalize the C_Q definition further. The expected number of SV errors on a dataset can be expressed by prior probabilities for reliable targets and non-targets trials: $P_{\mathcal{RT}}$ and $P_{\mathcal{RN}}$. We can manipulate (3.25) to make explicit the dependency on the reliability priors:

$$\begin{aligned}
C_Q(\xi_\theta,\xi_R) = & C_{\text{Miss}} P_{\mathcal{T}} \left(P_{\mathcal{RT}}(1-P_{\text{FU}\mathcal{T}}(\xi_\theta,\xi_R)) + (1-P_{\mathcal{RT}})P_{\text{FR}\mathcal{T}}(\xi_\theta,\xi_R)\right) P_{\text{Miss}\hat{\mathcal{R}}}(\xi_\theta,\xi_R) \\
& + C_{\text{FA}}(1-P_{\mathcal{T}}) \left(P_{\mathcal{RN}}(1-P_{\text{FU}\mathcal{N}}(\xi_\theta,\xi_R)) + (1-P_{\mathcal{RN}})P_{\text{FR}\mathcal{N}}(\xi_\theta,\xi_R)\right) P_{\text{FA}}(\xi_\theta,\xi_R) \\
& + C_{\text{FU}\mathcal{T}} P_{\mathcal{T}} P_{\mathcal{RT}} P_{\text{FU}\mathcal{T}}(\xi_\theta,\xi_R) + C_{\text{FU}\mathcal{N}}(1-P_{\mathcal{T}}) P_{\mathcal{RN}} P_{\text{FU}\mathcal{N}}(\xi_\theta,\xi_R)
\end{aligned} \tag{3.28}$$

where

$$\begin{aligned}
P_{\text{Miss}\hat{\mathcal{R}}}(\xi_\theta,\xi_R) &= \frac{N_{\text{Miss}\hat{\mathcal{R}}}(\xi_\theta,\xi_R)}{N_{\hat{\mathcal{R}}\mathcal{T}}(\xi_R)} & P_{\text{FA}\hat{\mathcal{R}}}(\xi_\theta,\xi_R) &= \frac{N_{\text{FA}\hat{\mathcal{R}}}(\xi_\theta,\xi_R)}{N_{\hat{\mathcal{R}}\mathcal{N}}(\xi_R)} \\
P_{\text{FR}\mathcal{T}}(\xi_\theta,\xi_R) &= \frac{N_{\text{FR}\mathcal{T}}(\xi_\theta,\xi_R)}{N_{\mathcal{UT}}(\xi_\theta)} & P_{\text{FR}\mathcal{N}}(\xi_\theta,\xi_R) &= \frac{N_{\text{FR}\mathcal{N}}(\xi_\theta,\xi_R)}{N_{\mathcal{UN}}(\xi_\theta)} \\
P_{\text{FU}\mathcal{T}}(\xi_\theta,\xi_R) &= \frac{N_{\text{FU}\mathcal{T}}(\xi_\theta,\xi_R)}{N_{\mathcal{RT}}(\xi_\theta)} & P_{\text{FU}\mathcal{N}}(\xi_\theta,\xi_R) &= \frac{N_{\text{FU}\mathcal{N}}(\xi_\theta,\xi_R)}{N_{\mathcal{RN}}(\xi_\theta)}\ .
\end{aligned} \tag{3.29}$$

Note that $P_{\text{Miss}\hat{\mathcal{R}}}$ and $P_{\text{FA}\hat{\mathcal{R}}}$ are same defined in Section 3.3.2. $N_{\text{FR}\mathcal{T}}$ and $N_{\text{FR}\mathcal{N}}$ are the number of target and non-targets bad classified as reliable; and, $N_{\text{FR}\mathcal{T}}$ and $N_{\text{FR}\mathcal{N}}$ are the bad classified as unreliable. $N_{\mathcal{U}\mathcal{T}}$ and $N_{\mathcal{U}\mathcal{N}}$ are the number of unreliable trials; and, $N_{\mathcal{R}\mathcal{T}}$ and $N_{\mathcal{R}\mathcal{N}}$ are the number of reliable trials. Equation (3.28) allows to evaluate C_Q for different values of the priors while (3.25) evaluates the cost for the actual values of $P_{\mathcal{R}\mathcal{T}}$ and $P_{\mathcal{R}\mathcal{N}}$ of the test dataset.

From now on, we will omit the dependencies of ξ_θ and ξ_R in error rates and cost to keep the notation uncluttered.

3.4 Experimental Setup

3.4.1 Speaker verification system

The SV baseline system was based on i-vectors with two-covariance model. We used 400 dimensional i-vectors. They were extracted from 20 short-time Gaussianized MFCC plus deltas and double deltas and a 2048 component diagonal covariance UBM. UBM, i-vector extractor and two-covariance model were gender independent and they were trained on telephone data from SRE04, SRE05 and SRE06. The i-vectors preprocessing included centering, whitening and length normalization.

SV scores were calibrated with the *Bosaris Toolkit* to optimize the *old NIST operating point* ($C_{\text{Miss}} = 10$, $C_{\text{FA}} = 1$, $P_{\mathcal{T}} = 0.01$). Calibration was trained on NIST SRE08 without noise added. Then, we applied this calibration function to all our datasets. We chose the Bayes decision threshold (2.29). On the clean part of SRE10, this system achieved an of EER=2.2%, minimum DCF=0.14 and actual DCF=0.17.

3.4.2 Databases

Here, we describe the databases that employed in the experiments that we will present in the following chapters. We have a synthetic database with noise and reverberation that we used to train and test our approaches. We also have real databases to verify that the models trained on the synthetic dataset can be applied to real scenarios.

3.4.2.1 NIST SRE with additive noise

We took the telephone part of NIST SRE08 and SRE10 databases and assumed that both are approximately clean. Then, we created a synthetic database by degrading NIST with different noise levels. We followed a protocol similar to the Aurora2 dataset [Hirsch and Pearce, 2000]. We added different Aurora2 noises to enrollment and test:

- Enrollment: suburban train, babble, car and exhibition hall.
- Test: restaurant, street, airport and train station.

Thus, we avoided the optimistic case in which the same type of noise is observed in enrollment and test samples. Noises were previously filtered by the ITU MIR telephone frequency response to simulate that they had pass through a telephone channel. The type of noise for each file was selected randomly. We created a list *file-noise* to be able to regenerate the dataset from scratch if needed.

Table 3.1: EER(%) dataset NIST SRE10 + noise.

EER(%)		Test SNR (dB)					
		CLEAN	20	15	10	5	0
Enrol. SNR (dB)	CLEAN	2.23	4.02	5.43	7.25	11.23	20.31
	20	3.75	8.89	9.96	11.65	14.83	21.83
	15	4.99	10.59	12.17	14.00	16.63	22.91
	10	7.70	13.46	15.35	17.58	20.56	25.49
	5	12.26	18.92	20.49	22.93	25.57	30.34
	0	22.53	27.18	28.69	30.91	32.85	35.93

We used the open source FaNT Tool [Hirsch, 2005] for adding noise to the signals. We have signal-to-noise ratios of 20dB, 15dB, 10dB, 5dB and 0dB.

For each file, we only added noise on the interest channel and kept the other channel clean. We used the reference channel to remove cross-talk from the interest channel.

With this data we created two trial lists: one list from NIST SRE08 data to train the reliability models, and another from NIST SRE10 for evaluation. The development list scored all the SRE08 enrollment segments against all the SRE08 test segments for all the signal-to-noise ratios added. The evaluation list is the official NIST SRE10 core (non-extended) det5 replicated for all possible enrollment and test signal-to-noise ratios. In Table 3.1, we show the EER that our SV system obtained on this dataset for each enroll–test noise pair. Error rates rapidly grew as we increase the noise power. If we pool all the conditions, we obtain EER=22.88%, minimum DCF=0.99 and actual DCF=2.96.

3.4.2.2 NIST SRE with reverberation

In order to create the reverberant dataset we also took NIST SRE08 and SRE10. We used a free Matlab ®package based on [McGovern, 2004]. This package includes two tools:

- RIR: calculates the impulse response of a rectangular room given the room dimensions, the reflection coefficients of the walls and the speaker and microphone locations.
- FCONV: used to convolve the room impulse response (RIR) with the clean signal.

We created random room impulse responses with the following criteria:

- 8 sizes of room, from small room to basketball court.
- Add a random number to the room size to change it ±50%.
- 8 different materials for the walls: rubber, granite, clay, concrete, steel, aluminum, brick and glass.
- Random speaker position inside the room.
- Random microphone position inside a square of 4 meters of side around the speaker.

Table 3.2: EER(%) dataset NIST SRE10 + reverberation.

EER(%)		Test reverb. time (sec)								
		CLEAN	0.025	0.05	0.075	0.1	0.25	0.5	0.75	1
Enrol reverb. time (sec)	CLEAN	2.23	3.37	5.20	7.71	18.99	19.50	24.66	22.10	20.38
	0.025	3.37	5.63	8.55	10.15	18.73	19.35	23.32	21.67	19.14
	0.050	5.40	7.72	10.50	14.17	20.35	21.39	25.09	22.22	20.34
	0.075	7.26	10.87	13.17	16.22	21.16	21.63	24.47	22.99	22.17
	0.100	17.88	18.18	20.69	21.29	28.32	29.25	32.92	32.90	32.68
	0.250	19.11	19.01	20.97	21.43	29.72	30.53	31.94	32.79	32.73
	0.500	22.57	22.07	23.98	24.41	31.53	32.19	30.34	31.20	32.74
	0.750	22.45	22.42	23.12	23.43	32.41	31.50	31.47	31.54	32.03
	1.000	19.57	19.72	21.48	21.71	31.02	31.21	31.01	30.99	29.73

For each RIR we computed the reverberation time (RT) as the time that the filter energy takes to fall 60dB. It is calculated from the energy decay curve (EDC) as

$$\mathrm{EDC}(t) = \int_t^\infty h(x)\,\mathrm{d}x \ , \tag{3.30}$$

$$T_{60} = \{t \ni \mathrm{EDC}(t) = \mathrm{EDC}(0) - 60\mathrm{dB}\} \ . \tag{3.31}$$

We assigned each RIR to one of 8 groups by the nearest reverberation time among 0.025, 0.05, 0.075, 0.1, 0.25, 0.5, 0.75 and 1 second. Each RIR is used only to degrade one file. The RIR for each file is selected randomly.

Each RIR is only used to degrade one file. The RIR for each file was selected randomly and we created a list *file-RIR* to be able to regenerate the database if needed.

For each file, we added reverberation on the interest channel and kept the other channel clean. We used the reference channel to remove cross-talk from the interest channel.

In the same manner as for the dataset with additive noise, we created one trial list from NIST SRE08 data to train the reliability models; and another form NIST SRE10 for evaluation. In Table 3.1, we show the EER that our SV system obtain on this dataset for each enroll–test reverberation time pair. By pooling all the conditions we obtained EER=33.52%, minimum DCF=0.99 and actual DCF=4.5. Furthermore, we created lists by pooling together the trials with noise and reverberation. There, we obtained EER=30.25%, min. DCF=0.99 and actual DCF=4.06.

3.4.2.3 NIST SRE with saturation

We added saturation to the test part of NIST SRE10 to experiment with our saturation detector. The test set was replicated to have signals with 0, 1, 2, 5, 10, 25, 50, and 75% of speech frames with saturation. The saturation level was different for each file and it was selected to provide the desired percentage of saturated frames.

3.4.2.4 Agnitio benchmark

Agnitio benchmark was provided by Agnitio S.L. It contains audios from many different sources to take as much speaker and channel variability as possible. The performance of our SV system on this datasets was EER=5.46%, minimum DCF=0.26 and actual DCF=1.49.

3.4.2.5 Ahumada IV

Ahumada IV is a database of good quality acquired by the Spanish Guardia Civil [Ramos et al., 2008]. Recordings were done by SITEL, a nationwide digital interception system. This system records digital wiretaps directly connected to all mobile telephone operators. On this dataset, our SV system provided EER=2.85%, minimum DCF=0.14 and actual DCF=2.96.

3.4.2.6 MOBIO

The MOBIO database [McCool et al., 2012] is a bi-modal (face/speaker) database collected from August 2008 to July 2010 in six different sites from five different countries. This led to a diverse database with both native and non-native English speakers. It consists of 152 people with a female-male ratio of nearly 1:2 (100 males and 52 females). In total 12 sessions were captured for each individual.

The database was recorded using two types of mobile devices: mobile phones (NOKIA N93i) and laptop computers (standard 2008 MacBook). The laptop computer was used to capture only one session (the very first session) while all the other data was captured on the mobile phone, including the first session.

The data collection was conducted in two phases: Phase I and Phase II. Each session of Phase I consisted of 5 short response questions, 5 short free speech questions, 1 pre-defined text, and 10 free speech questions. Phase II was made shorter and consisted of 5 short response questions,1 pre-defined text, and 5 free speech questions. We experimented with the evaluation protocol given in Phase II.

The speakers of the database are split up into three different sets: training, development and evaluation set:

- Training set: The data of this set are used to learn the background parameters of the algorithm (UBM, JFA, etc.). It can also be employed for score normalization (cohort, etc.). We did not use this data in our experiments; our models were trained on NIST SRE data only.

- Development set: This data is intended to choose meta-parameters or train calibration. For the enrollment of a target model, 5 audio files of the target speaker are provided, and it is forbidden to use the information of other speakers of the development set. The remaining audio files serve as test files, and scores are computed between all the test files and all the models. We use this data in Chapter 7 to train a Bayesian network that predicts the reliability of SV decisions.

- Evaluation set: This data is used for the final evaluation of performance. We employed this part to test reliability estimation models.

On the test set, our SV system performance was EER=12.37%, minimum DCF=0.57 and actual DCF=8.72.

3.5 Summary

In this part of the thesis, we intend to develop algorithms to estimate the reliability of the decisions taken by speaker verification systems. The multiples causes that can degrade

Table 3.3: Number of segments and trials of the reliability databases.

Database	#models	#tests	#trials	#tar trials	#nontar trials
SRE08 clean	1206	1173	767874	1269	766605
SRE08 with noise	7236	7038	27643464	45684	27597780
SRE08 with reverb.	10854	10557	62197794	102789	62095005
SRE10 clean	580	712	30373	708	29665
SRE10 with noise	3480	4272	1093428	25488	1067940
SRE10 with reverb.	5220	6408	2460213	57348	2402865
AGN Bench.	116	523	60668	1046	59622
Ahumada IV	91	442	40222	442	39780
MOBIO Dev.	42	4410	94500	4410	90090
MOBIO Eval.	58	6090	193620	6090	187530

SV performance motivated this work. We commented previous approaches to the problem. Some of those approaches are based on confidence measures computed from the distributions of the SV scores; others use measures capturing information about the acoustic quality of speech like SNR; and others try to combine both. The approaches that we investigated are included in the third group.

We discussed some methods to compare the performance of different reliability detectors. The first method consisted of computing misses, false alarm rates and DCF only with the trials that are classified as reliable. We can plot curves of actual DCF against the reliability detector threshold or against the number of trials discarded to compare systems. For the second method, we quantized the score of the reliability detector and computed error rates on the trials corresponding to each quantization level. Finally, we also proposed an extended DCF that included the cost of rejecting trials whose SV decisions are correct. This cost is interesting to select to operating of the reliability detector for a given application.

We also described the SV system whose reliability we are going to evaluate. It is a state-of-the-art system based on i-vectors and PLDA. Finally, we explained the databases that we used in our experiments: an artificial database with added noise and reverberation, and three more realistic databases.

Chapter 4

Quality Measures of Speech

4.1 Introduction

In this chapter we describe the quality measures that we extracted from the trials segments. These measures, on their own, do not provide information about whether the trial is target or non target but they can help to estimate the reliability of the SV decision.

Sections 4.2 to 4.12 enumerate our measures giving a description of the algorithm to compute them and showing proof of their correlation to SV performance. Our measure set included modulation index, signal-to-noise ratio, number of speech frames, jitter, shimmer, saturation detection, likelihood of the speech frames given the UBM model and given the factor analysis model used to compute the i-vector, and likelihood of the i-vector given the PLDA model. Besides, we present novel features obtained from the parameters needed to adapt a clean GMM to a noisy signal by applying the vector Taylor series paradigm [Li et al., 2009]. The correlation between measures and performance was studied based on graphs depicting the SV score or the DCF against the value of the measure. In Section 4.13, we present a method to combine quality measures by linear discriminant analysis for the purpose of detecting noise and reverberation. In Section 4.14, we rank the quality measures from the point of view of their relation to speaker verification performance. The most promising measures were the VTS parameters, modulation index, SNR and UBM log-likelihood. Finally, Section 4.15 summarizes the chapter.

4.2 Modulation Index

4.2.1 Description

The modulation index measures the amplitude variations of the speech signal. This measure has been typically used in AM radio communications to indicate how much the modulation varies around its unmodulated level. Figure 4.1 shows an example of an amplitude modulated signal.

The modulation index at time t is calculated as

$$\mathrm{Indx}(t) = \frac{v_{\max}(t) - v_{\min}(t)}{v_{\max}(t) + v_{\min}(t)} \; . \tag{4.1}$$

where $v(t)$ is the envelope of the signal and $v_{\max}(t)$ and $v_{\min}(t)$ are the local maximum and

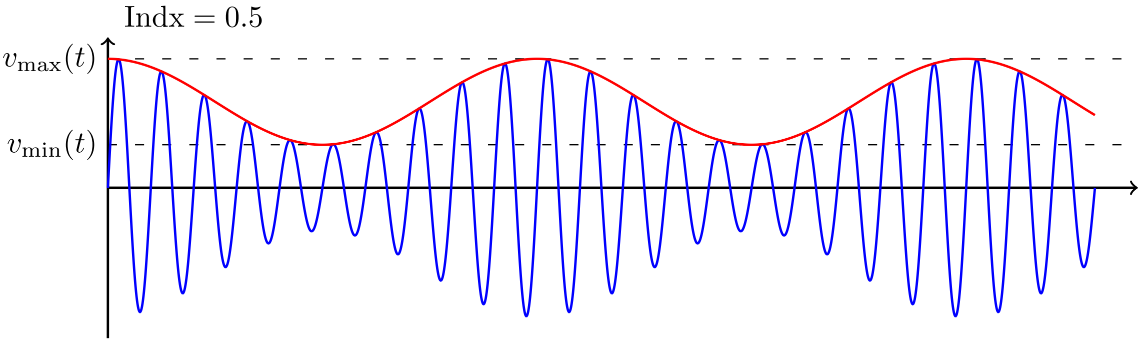


Figure 4.1: Illustration of the envelope of a signal.

minimum of the envelope in the region close to time t. The envelope of a recording with noise or reverberation has higher local minima and, therefore, lower modulation index.

Figure 4.2 depicts a block diagram of the algorithm that we used to compute the modulation index of speech signals. The envelope was approximated by the absolute value of the signal $s(t)$ down-sampled to 60 Hz. The down-sampling was performed in two steps, from 8 kHz to 200 Hz and then, to 60 Hz. Thus, we could use anti-aliasing filters of lower order and assure their stability. They are second order IIR filters. Then, we searched the peaks and valleys of the envelope and compute a value of modulation index per speech frame. Finally, the index is averaged over all speech frames.

4.2.2 Correlation with SV performance

We experimented on NIST SRE10 with added noise and reverberation to quantify the correlation between the modulation index and the SV performance. We restricted ourselves to the case where the enrollment signal is clean and the test is degraded. Figure 4.3 shows the relation between the target and non-target scores and the modulation index of the test segment. To create this figure, we uniformly quantified the modulation index into 20 levels and assigned each trials to its corresponding level according to the index of the test segment. Then for each level, we computed the mean and standard deviation of the trial scores. The filled dots indicate the score mean while the error bars are twice the standard deviation long. The black lines show the linear regression between the scores and the modulation index. The target scores were very correlated with the modulation index. They strongly dropped for low index values. For additive noise, the relation between scores and modulation index was approximately linear. Non-target scores for lower indexes were larger than for lower indexes. However, the effect was not as pronounced as for the targets.

As the modulation index decreases the target and non-target score distributions become closer, which directly deteriorates recognition performance as we can see in Figure 4.4. This figure displays EER and minimum DCF against modulation index. This representation

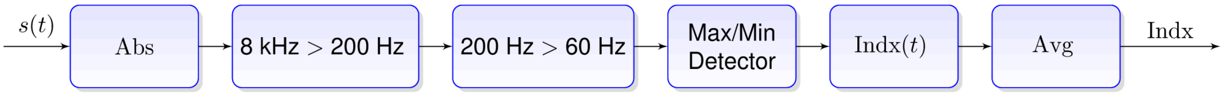
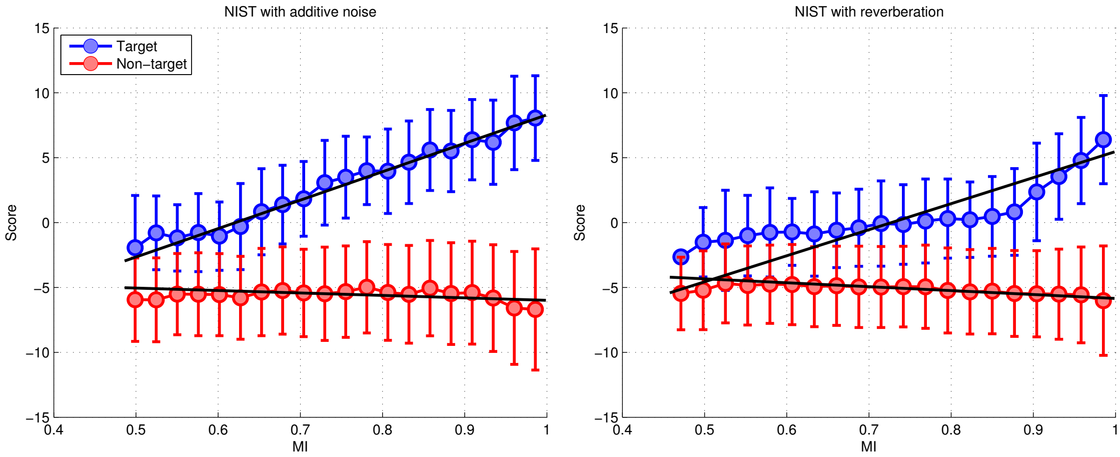


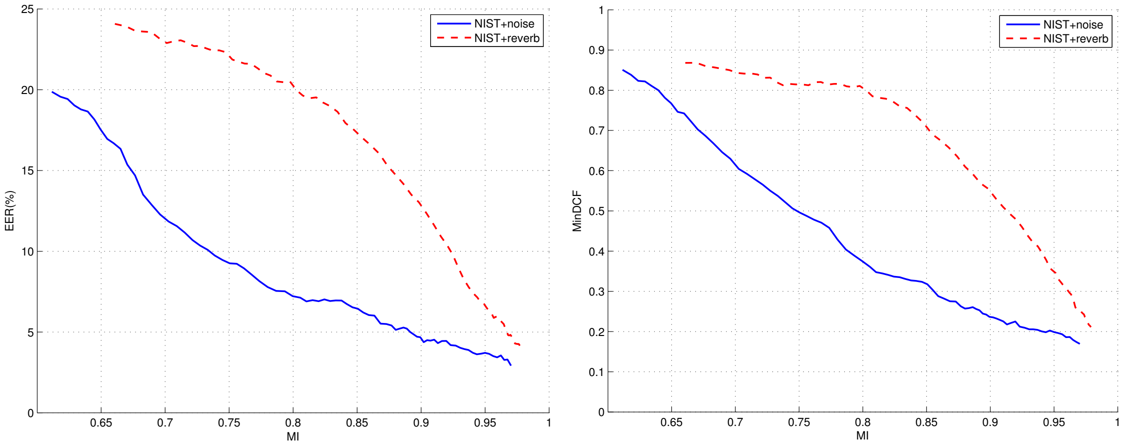
Figure 4.2: Block diagram to compute the modulation index.



(a) (b)

Figure 4.3: Score range against modulation index for NIST SRE10 + additive noise (a) and reverberation (b). Black lines plot the linear regression between score and modulation index.

is based on the method to compare reliability detectors proposed in Section 3.3.3. We quantized the modulation index and computed error rates for the trials lying in each quantization level. In this case, we took non-uniform overlapped quantization levels. Each level included 20% of the trials and there were a shift of 1% of the trials between levels. Figures like this one allow us to analyze the goodness of the measure for speaker verification. We conjectured that, to discriminate between reliable and unreliable trials, for certain values of the measure we should observe very low error rate, and for others high error rate. With this in mind, we can compare measures according to the minimum EER and according to the difference between the maximum and minimum EER in the figure (ΔEER). In the same manner, we could compare DCF curves. Regarding the modulation index, ΔEER is around 20 for both noise and reverberation and the EER is lower than 4% for indexes close to 1. For additive noise, error rates grow gradually as the index decreases. For reverberation,



(a) EER (%) (b) Minimum DCF

Figure 4.4: SV performance against modulation index.

error rates grow rapidly at the beginning and then stabilize. We concluded that additive noise affects similarly to modulation index and error rates while reverberation can affect error rates without producing a significant reduction of the modulation index.

4.3 Signal-to-Noise Ratio

4.3.1 Description

The signal-to-noise ratio (SNR) is the ratio between the power of the speech signal and the power of the background noise. Most methods to compute the SNR of speech are based on measuring noise during silence intervals. This techniques rely on the assumption that noise is stationary or varies slowly between silence intervals. This is a drawback because, in real situations, we find signals where silence intervals are too few to follow the noise evolution or they are too short for a good noise estimation. Instead, we proposed a method based on the properties of the voiced speech intervals. The most part of the energy of voiced speech is concentrated in multiples of its pitch frequency while the frequency distribution of additive noise is more uniform. We estimated the clean signal and noise powers separately by filters with comb like frequency responses. The proportion of voiced segments is large enough to track the noise progress in a wide range of real applications and provide a frame by frame quality measure.

Figure 4.5 depicts the block diagram of our algorithm. Two comb filters were employed, H_s to estimate the clean signal and H_n to estimate the noise. Before H_n, if the modulation index of the segment is over a threshold, the signal was filtered by a plainer version of its

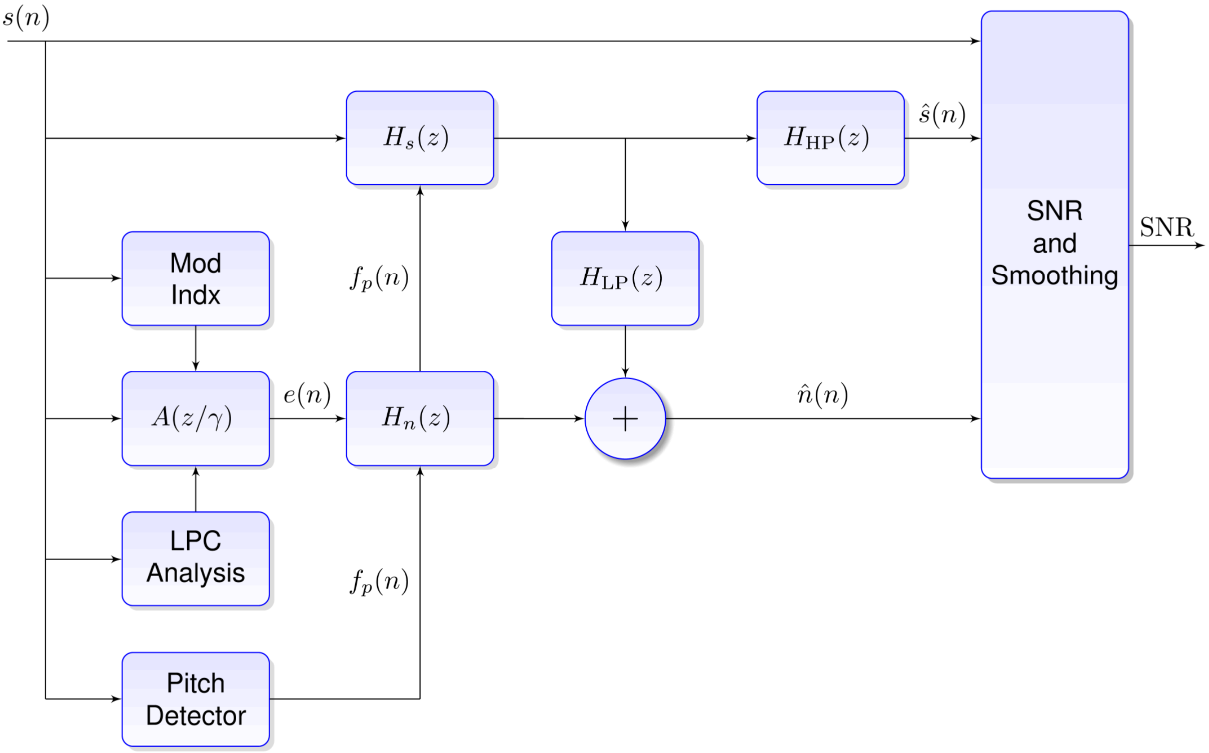


Figure 4.5: Block diagram to compute the SNR.

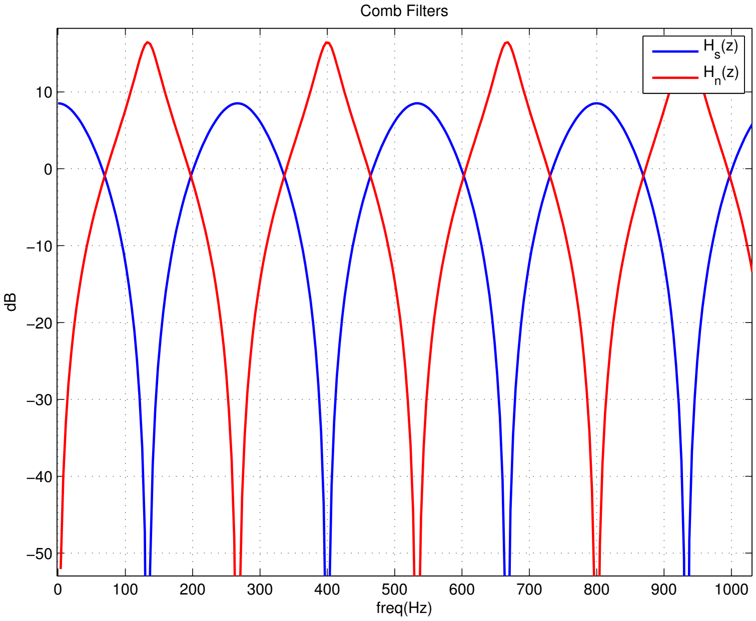


Figure 4.6: Comb filters frequency response.

LPC (Linear Prediction Coefficients) inverse filter. Thus, we further reduced the presence of speech signal in the noise estimation. Due to the comb filters design, low frequency noise (< 60 Hz) remains in the speech estimation while it is eliminated from the noise. We solved this issue by a pair of complementary low and high pass filters (H_{LP} and H_{HP}. We filtered the clean speech by H_{LP} to measure the low frequency noise and we added it to the noise estimation to obtain $\hat{n}$. The high pass filter removes the low frequency noise from the speech providing the final estimate $\hat{s}$. We computed SNR from $\hat{s}$ and $\hat{n}$. Finally, we applied a smoothing procedure to obtain the SNR of the unvoiced segments and voiced segments where the pitch measure was considered inaccurate. In the next sections, we explain the blocks of the figure in more detail.

4.3.1.1 Comb filters

Comb filters have a periodic frequency response consisting of alternate pass and stop bands. The filter $H_s(z,t)$, that estimates the clean the speech, has pass bands in the multiples of the pitch frequency nf_p where the speech power is maximum and stop bands in $(n+1/2)f_p$ where there is only noise. To estimate the noise, the filter $H_n(z,t)$ has its bands inverted with respect to $H_s(z,t)$. To provide this kind of response, the filters add or subtract a guess of the speech signal in the current instant inferred from the samples of adjacent pitch periods. Thus, it produces constructive or destructive interference. We implemented non causal IIR filters with the following transfer functions:

$$H_s(z,t) = \frac{0.5z^{T_p(t)} + 1 + 0.5z^{-T_p(t)}}{1-\alpha_s z^{-T_p(t)}} \quad H_n(z,t) = \frac{-0.5z^{T_p(t)} + 1 - 0.5z^{-T_p(t)}}{1+\alpha_n z^{-T_p(t)}} \tag{4.2}$$

where $T_p(t)$ is the pitch period at time t and, α_s and α_n are coefficients that modify the bandwidth of the filter. We empirically selected $\alpha_s = 0.25$ and $\alpha_n = 0.7$. As the pitch period changes along the speech segment these are time varying filters. Figure 4.6 shows their frequency responses.

For this method to work, we needed a fine pitch estimation, an error of only one sample in the pitch period degrades the performance. Our pitch estimator was based on the RAPT algorithm [Talkin, 1995]. The maxima of the normalized autocorrelation of the LPC prediction error are initial candidates for the pitch period. We feed them into a dynamic programming algorithm that selects the best one.

4.3.1.2 LPC inverse filter

We must take into account that pitch estimators always produce some precision errors as well as speech signals are not completely periodic. For these reasons, some residual speech always remained present in the noise estimation from H_n leading to a sub-estimation of the SNR. This effect was more significant in cases with high SNR (larger than 15 dB) when the amount of noise in the noise estimation is smaller than the residue. The inverse LPC filter $A(z)$ matched to the input signal was included before H_n to reduce that residue. The output of $A(z)$ is the LPC prediction error where the speech formants are eliminated.

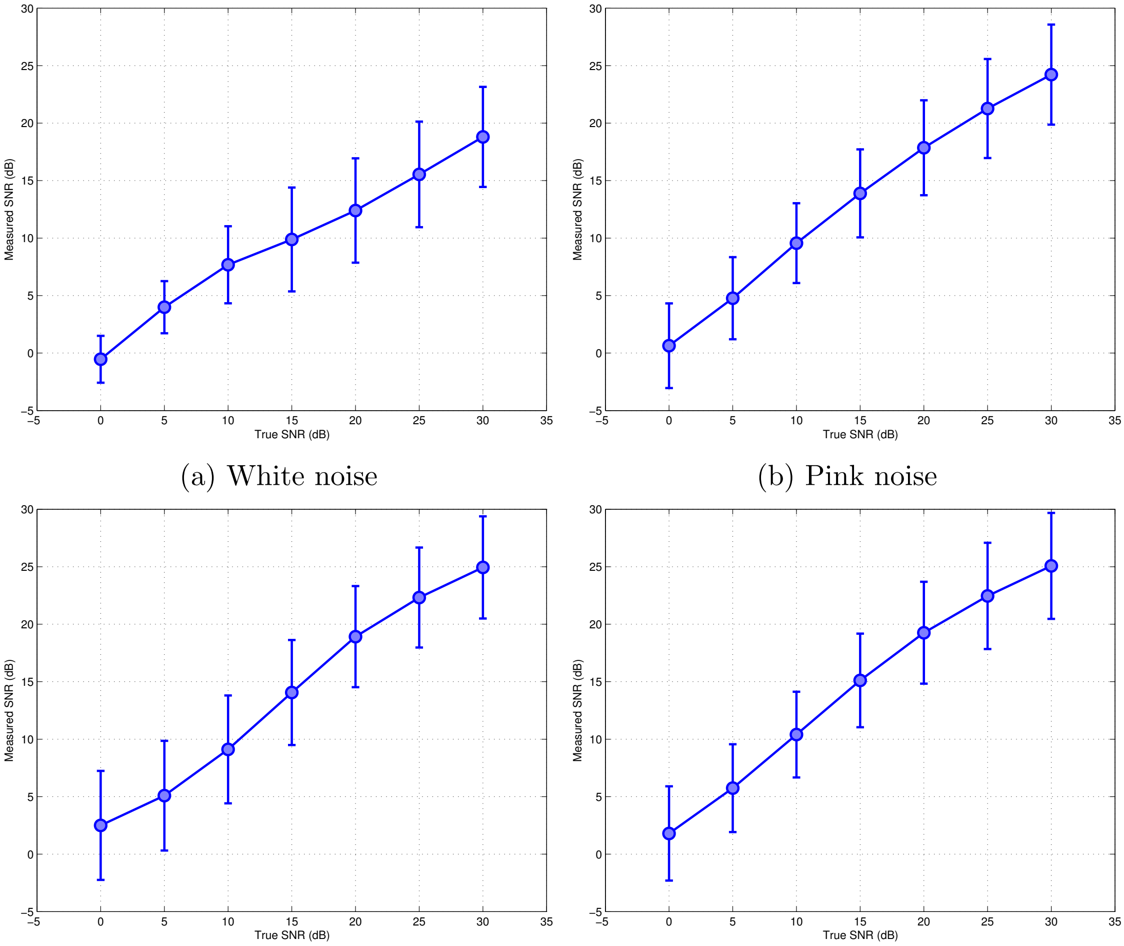
We noted that, sometimes, after adding inverse filter, the system over-estimated the SNR. We concluded that it was because we estimated the LP coefficients from the noisy signal, for very colored noises, the coefficients included also the spectral distribution of the noise. Then, the inverse filter also reduced the amount of noise in the noise estimation. To solve this issue, we decided not to apply $A(z)$ to very noisy signals and to apply a flatter version of it to cleaner signals. We made an *a priori* decision about the noise level based on the modulation index. If the modulation index was larger than 0.7, we assumed that the noise level was low and applied the filter $A(z/\gamma)$ where

$$\gamma = (\mathrm{Indx} - 0.7)/0.3 \ . \tag{4.3}$$

The parameter γ flattens the filter when the modulation index is lower than 1. This methods improved the SNR estimation for high SNR.

4.3.1.3 Low frequency noise compensation

As shown in Figure 4.6, the clean speech filter H_s has a maximum at 0 Hz. At that frequency there is no speech ever, so, if there is low frequency noise like in cars or air conditioning, it will be present in the clean signal estimation. Contrarily, the filter H_n has a minimum at 0 Hz and the low frequency noise will not be included in the noise estimation. Thus, we obtained an over-estimation of the power of the clean speech and sub-estimation of the noise and, therefore, over-estimation of the SNR. To compensate that, we combined of a low pass and a high pass filters. We passed the output of H_s through a high-pass filter H_{HP} to eliminate the low frequency noise from the clean speech estimation. Besides, the low-pass filter H_{LP} estimated the low frequency noise and we added it to the output of the filter H_n. Both were second order IIR Butterworth filters designed to have a cut frequency of 60 Hz. After these steps, we obtained the final estimations of the clean speech $\hat{s}(n)$ and the noise $\hat{n}(s)$.



(a) White noise

(b) Pink noise

(c) Car noise

(d) Factory noise

Figure 4.7: True SNR against Measured SNR.

4.3.1.4 SNR estimation and smoothing

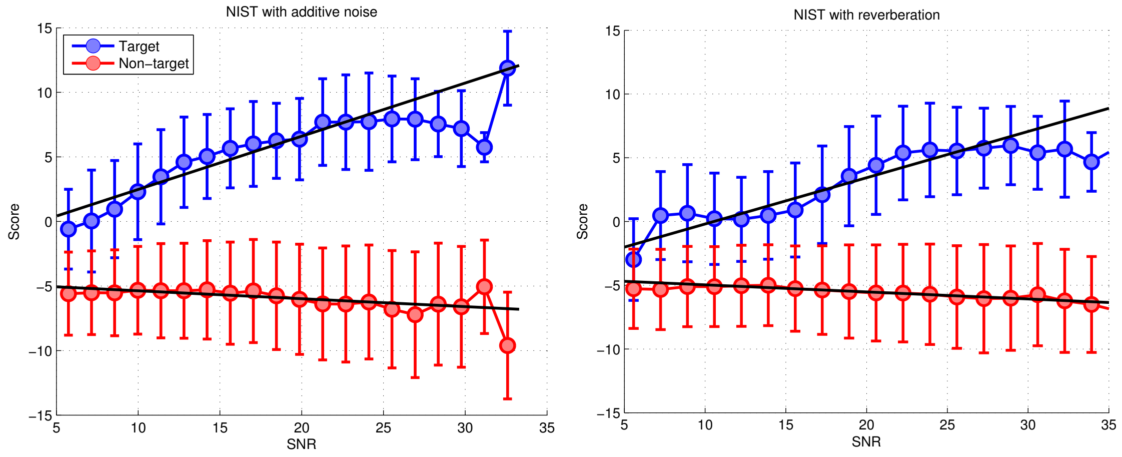
As first approximation to the SNR, we computed the ratio between the powers of $\hat{s}(n)$ and $\hat{n}(s)$:

$$\mathrm{CR} = 10 \log_{10} \left(\frac{P_{\hat{s}}}{P_{\hat{n}}} \right) \text{ (dB)} \tag{4.4}$$

where CR stands for *comb filters ratio*. This ratio is affected by two factors. On the one hand, the gains of the filters introduce a bias in the SNR. On the other hand, as previously commented, there was a residual amount of speech in the noise estimation due to pitch errors that leaded to over-estimate the noise power. In our experiments, we observed that we can calibrate the CR by linear regression to obtain a good approximation of the SNR:

$$\mathrm{SNR} = -1.68 + 1.26 \text{ CR (dB)} . \tag{4.5}$$

This algorithm allows us to compute the SNR for voiced segments. To obtain the SNR for unvoiced segments we considered that they are short enough to suppose that noise does not abruptly change along them. Thus, we can apply a smoothing procedure on the noise



(a) (b)

Figure 4.8: Score range against SNR for NIST SRE10 + additive noise (a) and reverberation (b). Black lines plot the linear regression between score and SNR.

power. First, by combining the calibrated SNR and the power of the original signal P_s we obtained the noise power for voiced segments:

$$\hat{P}_n = P_s - \text{SNR} - 10\log_{10}\left(1 + 10^{-\frac{\text{SNR}}{10}}\right) \text{ (dB)} \, . \tag{4.6}$$

Then, $\hat{P}_n$ is interpolated in the unvoiced segments and smoothed by using a one pole IIR filter:

$$\hat{P}'_n = (1-\alpha)\hat{P}_n + \alpha\hat{P}'_n \, . \tag{4.7}$$

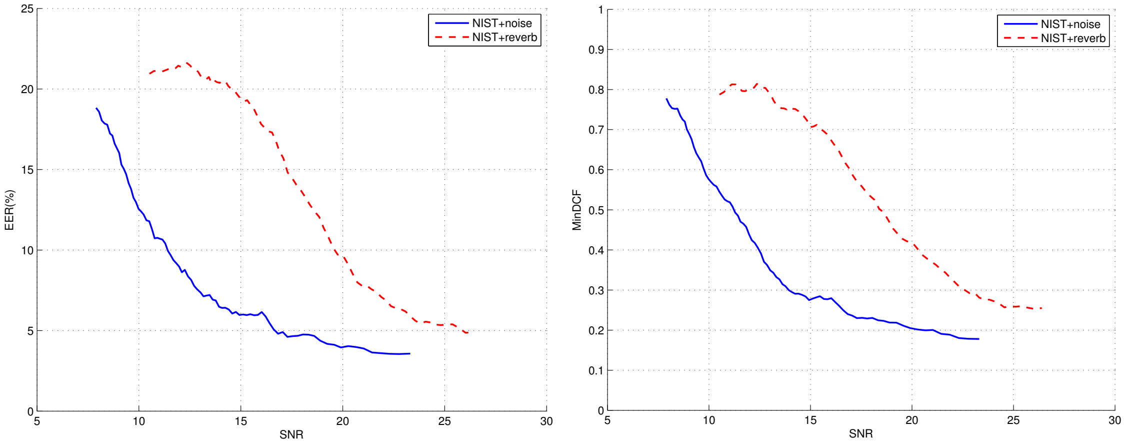
We chose the filter coefficient to compensate the effect of pitch detection errors and yet, to be able to track noise power changes ($\alpha = 0.95$). Finally, the SNR was re-computed as:

$$\text{SNR}' = P_s - \hat{P}_n + 10\log_{10}\left(1 - 10^{-\frac{P_s - \hat{P}'_n}{10}}\right) \text{ (dB)} \, . \tag{4.8}$$

4.3.2 SNR estimation performance

We tuned the SNR estimator on the Albayzin database [Moreno et al., 1993]. Albayzin is a Spanish spoken database designed for speech recognition. We used a sub-corpus of the database composed by utterances from a set of 200 phonetically balanced sentences. 4 speakers utter the overall set and 160 speakers a subset of 25 sentences, which makes 4100 sentences. This is a clean database so we added noise to obtain average SNR of 0, 3, 5, 10, 20 and 30 dB. We tested 4 types of noises: white, pink, car and factory.

Figure 4.7 displays the relation between frame level measured and true SNR. We quantized true SNR into levels separated by 5dB. Dots indicate the average SNR for each quantization level and bars indicate the standard deviation. For white noise, we appreciate a sub-estimation of the SNR from values starting at 10 dB, for pink noise at 20 dB, and for car and factory noises at 25 dB. Nevertheless, car and factory noise over-estimated the SNR for values around 0 dB. We tuned the algorithm to perform better for SNR lower than 10 dB because there was where the speaker verification accuracy dramatically dropped.



(a) EER (%)

(b) Minimum DCF

Figure 4.9: SV performance against SNR.

4.3.3 Correlation with SV performance

Figure 4.8 relates target and non-target scores with the measured SNR of the test segment. Similarly to what we saw for the modulation index, at low SNR target scores severely decay while non-targets slightly ascend. Figure 4.9 shows EER and minimum DCF against SNR. For NIST with additive noise, EER starts at 3.59% for 25 dB of SNR and increases rapidly from 15 dB down. This indicates an evident correlation between SNR and performance. For NIST with reverberation, EER starts at 4.90%, a higher value, and begins to rise at 23 dB. That means that reverberation can affect performance without having a great impact in the SNR estimation. Compared to the modulation index, SNR presents higher minimum EER and DCF and smaller difference between their maximum and minimum values. Thus in theory, a classifier based on the modulation index should discriminate better between reliable and unreliable trials.

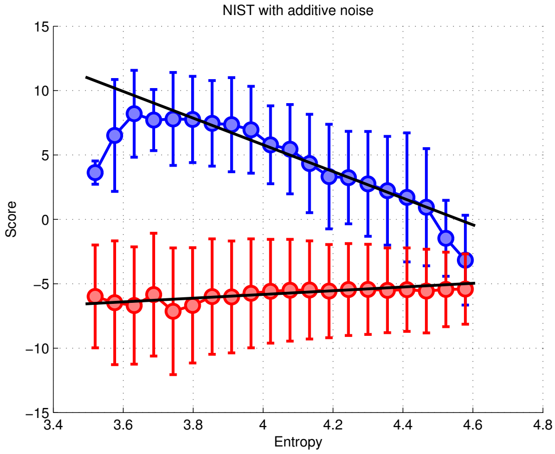
4.4 Spectral Entropy

4.4.1 Description

Entropy is a measure related to the peakiness or flatness of a probability distribution. The spectrum of a signal can be seen as the probability distribution that indicates which frequencies have a larger probability of appearance. To make the spectrum to look like a probability distribution we just need to normalize it to sum 1. Thus, the spectral entropy for a frame t is computed as

$$H(t) = -\sum_{\omega} \frac{|X(\omega,t)|^2}{\sum_{\omega'} |X(\omega',t)|^2} \log \frac{|X(\omega,t)|^2}{\sum_{\omega'} |X(\omega',t)|^2} \tag{4.9}$$

where $|X(\omega,t)|^2$ is the short term power spectrum of the signal. For our implementation, we divided the frequency axis into 32 bins so the maximum entropy is 5. The entropy of the segment was computed as the average over all speech frames. The idea of using the entropy as quality measure relies in the assumption that a clean signal should have a more organized spectrum, while a noisy signal should have a flatter spectrum.



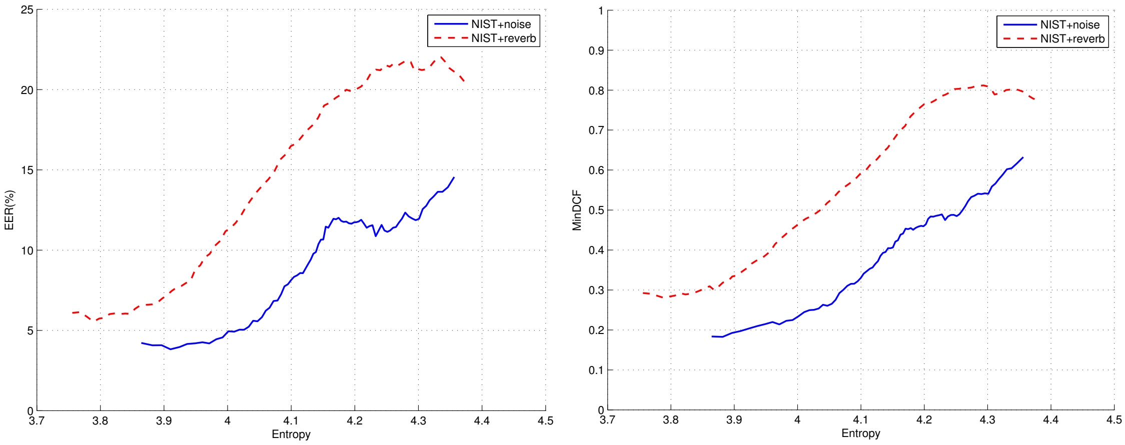
(a)

(b)

Figure 4.10: Score range against spectral entropy for NIST SRE10 + additive noise (a) and reverberation (b). Black lines plot the linear regression between score and spectral entropy.

4.4.2 Correlation with SV performance

Figure 4.10 depicts the relation between the scores and the spectral entropy of the segment. Figure 4.11 shows EER and minimum DCF against entropy. As expected, for higher entropy the target non-target score distributions are closer and performance decreases. For additive noise, target scores decrease linearly with entropy. Compared to SNR and modulation index, the difference between minimum and maximum EER is much smaller (ΔEER=11) so we expect this feature to discriminate worse. For reverberation ΔEER is around 16 which leads us to think that entropy will be more helpful for datasets with convolutional noise.



(a) EER (%)

(b) Minimum DCF

Figure 4.11: SV performance against spectral entropy.

4.5 Number of Speech Frames

4.5.1 Description

Having longer speech segments allows us to estimate better speaker models and therefore to reach lower error rates. For example, in NIST SRE the core condition (5 min. vs. 5 min.) has an EER around 2% while for the 10sec. vs. 10sec. condition it is around 12%. Thus, we used the number of speech frames detected by our VAD [Ramirez et al., 2004] as quality measure to predict performance.

4.5.2 Correlation with SV performance

Figure 4.12 presents target and non-target score distributions against the number of speech frames of the test segment. Figure 4.13 shows EER and minimum DCF against the number of frames. The core condition of NIST SRE10 is composed of 5 minutes segments, most of them having more than 2000 speech frames. For reverberation, the curves are more or less flat meaning that scores and performance do not depend on the number of frames–This is true for this dataset because 2000 frames is usually enough to produce good performance. On the contrary, for additive noise, target scores decays as the number of frames decreases (Figure 4.12a). The decay was linear from 15000 frames down. EER rapidly increased from 10000 frames down (Figure 4.13a). We observed ΔEER=13 which indicates that the number of frames will discriminate between reliable and unreliable trials worse than SNR and modulation index. This was a case where correlation did not mean causality. The error rates did not increase because there were less speech frames. It was noise what, at the same time, reduced the detected speech frames and worsened the performance.

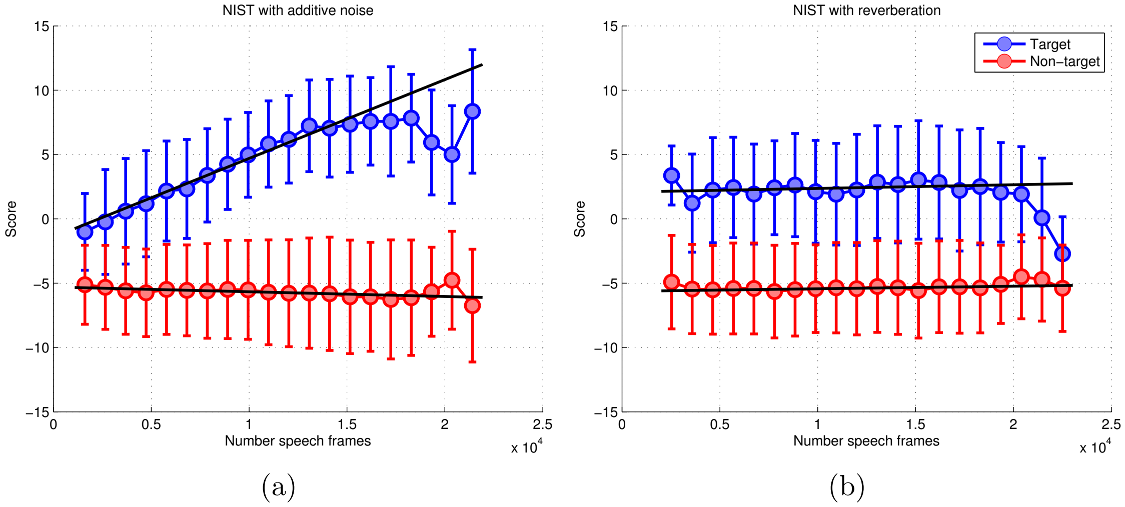
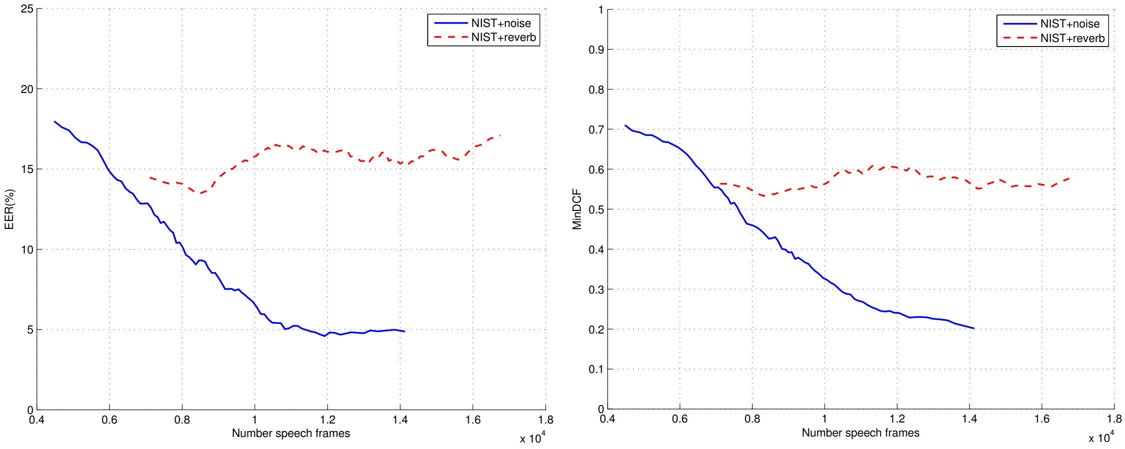


Figure 4.12: Score range against number of speech frames for NIST SRE10 + additive noise (a) and reverberation (b). Black lines plot the linear regression between score and number of speech frames.



(a) EER (%) (b) Minimum DCF

Figure 4.13: SV performance against number of speech frames.

4.6 UBM Log-Likelihood

4.6.1 Description

The Universal Background Model (UBM) is a GMM that represents the probability distribution of the speech features of the development database. Speaker models are adapted from this UBM. The UBM is usually trained on good quality databases so it can be considered as the distribution of normal speech. The feature distribution of the degraded signals will likely differ from the UBM what will lead to worse speaker models. For example, if the distribution of the trial segments does not match the UBM, we will compute inaccurate occupation probabilities for the UBM Gaussians. When we use those occupation probabilities to compute the JFA sufficient statics we will obtain i-vectors whose distribution do not match the PLDA model. For this reason, the likelihood of the utterance given the UBM is a measure of speech degradation. This measure was first used in [Harriero et al., 2009] with good results.

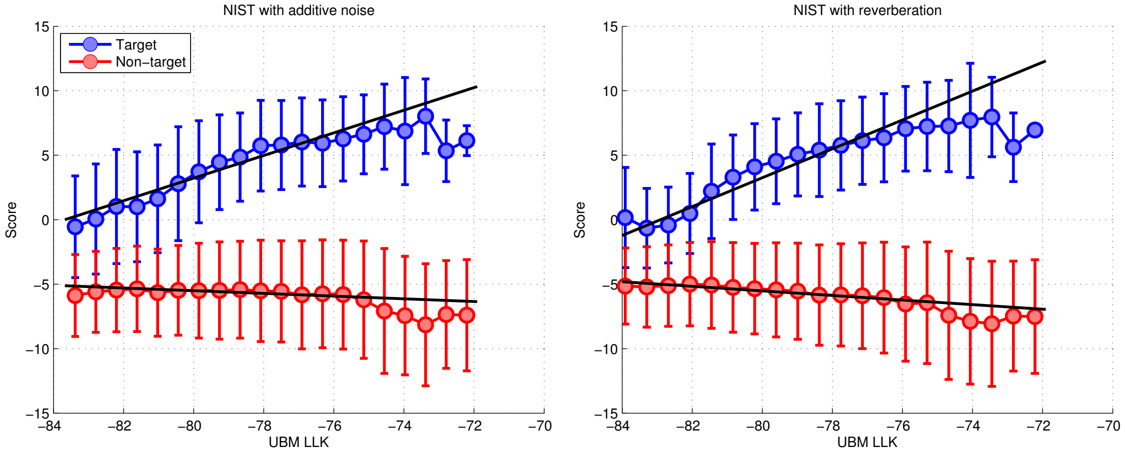
The log-likelihood of the utterance given the UBM normalized by the number of frames is

$$\ln P\left(\mathcal{X}|\mathbf{m},\mathbf{\Sigma}\right)=\frac{1}{T}\sum_{t=1}^{T}\ln\sum_{k=1}^{K}\mathcal{N}\left(X_t|\mathbf{m}_k,\mathbf{\Sigma}_k\right) \tag{4.10}$$

where $\mathcal{X}=\{X_1,X_2,\dots,X_T\}$ is a sequence of features; $\mathcal{N}$ is the Gaussian distribution; $\mathbf{m}_k$ and $\mathbf{\Sigma}_k$ are the mean and covariance of the k^{th} Gaussian; and $\mathbf{m}$ and $\mathbf{\Sigma}$ the set of means and covariances of all the Gaussians.

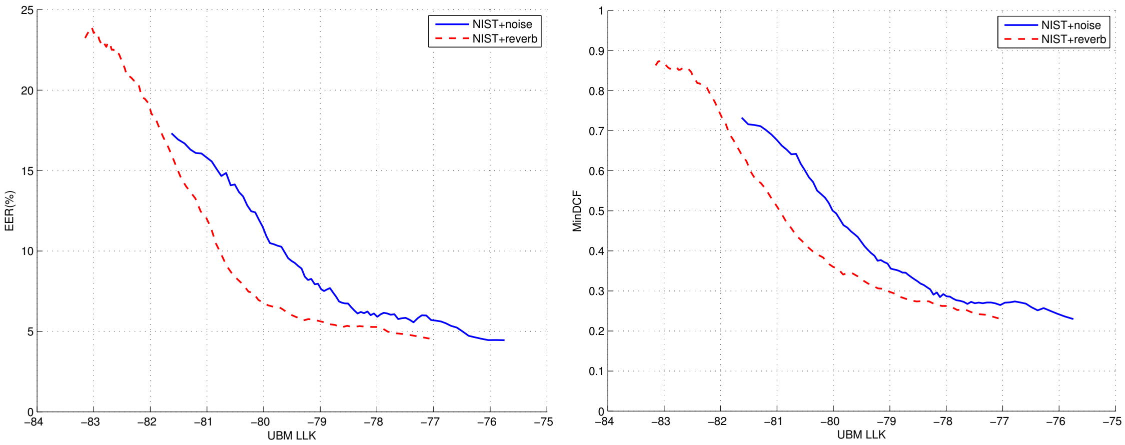
4.6.2 Correlation with SV performance

Figure 4.14 shows the relation between the target and non-target scores and the UBM log-likelihood of the segment. Figure 4.15 shows EER and minimum DCF against the UBM log-likelihood. The figures for noise and reverberation are much closer than for the previous quality measures. That could mean that the UBM may help to detect unreliable trials independently of what is the type of degradation of the speech signal.



(a) (b)

Figure 4.14: Score range against the UBM LLk for NIST SRE10 + additive noise (a) and reverberation (b). Black lines plot the linear regression between score and the UBM LLk.



(a) EER (%) (b) Minimum DCF

Figure 4.15: SV performance against the UBM LLk.

4.7 i-Vector Extractor Likelihood

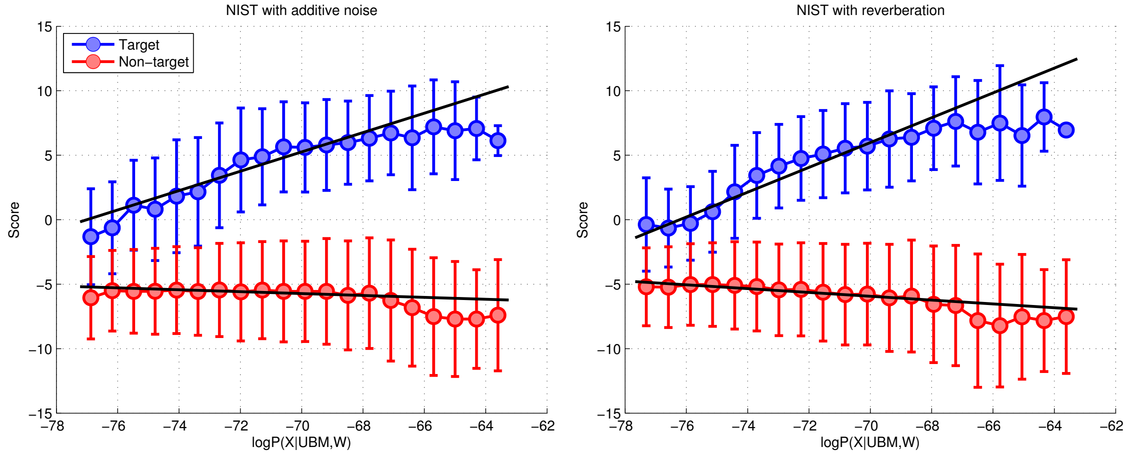
4.7.1 Description

As explained in Section 2.4.4, the i-vector approach for speaker recognition [Dehak et al., 2011b] assumes that the GMM super-vector mean $\mathbf{M}$ corresponding to a given utterance can be written as

$$\mathbf{M} = \mathbf{m} + \mathbf{T}\phi \tag{4.11}$$

where $\mathbf{m}$ is the UBM means super-vector, $\mathbf{T}$ is a low-rank matrix and ϕ is a standard normal distributed vector. ϕ is referred as i-vector in the literature. $\mathbf{T}$ defines the total variability space, i.e. the directions in which we can move the UBM to adapt it to the utterance.

Using the same reasoning as for the likelihood of the UBM, degraded signals should fit into this model worse than the signals that are similar to those of the development set. If



(a) (b)

Figure 4.16: Score range against the i-vector extractor LLk for NIST SRE10 + additive noise (a) and reverberation (b). Black lines plot the linear regression between score and the i-vector extractor LLk.

the variability directions given by $\mathbf{T}$ are not valid for the degraded signals we will obtain less discriminative i-vectors. Thus, we proposed the log-likelihood of the utterance given the i-vector extraction model as quality measure. This log-likelihood is given by

$$\ln P\left(\mathcal{X}|\mathbf{Z},\mathbf{m},\boldsymbol{\Sigma},\mathbf{T}\right) = \ln\int P\left(\mathcal{X}|\phi,\mathbf{Z},\mathbf{m},\boldsymbol{\Sigma},\mathbf{T}\right)\mathcal{N}\left(\phi|\mathbf{0},\mathbf{I}\right)\,\mathrm{d}\phi \tag{4.12}$$

$$= \ln P\left(\mathcal{X}|\mathbf{Z},\mathbf{m},\boldsymbol{\Sigma}\right) - \frac{1}{2}\ln|\mathbf{L}| + \frac{1}{2}\mathrm{E}\left[\phi\right]^{T}\mathbf{T}^{T}\boldsymbol{\Sigma}^{-1}\overline{F} \tag{4.13}$$

where

$$\ln P\left(\mathcal{X}|\mathbf{Z},\mathbf{m},\boldsymbol{\Sigma}\right) = -\frac{1}{2}\sum_{k=1}^{K} N_k \ln|2\pi\boldsymbol{\Sigma}_k| - \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\overline{\mathbf{S}}\right)\ ; \tag{4.14}$$

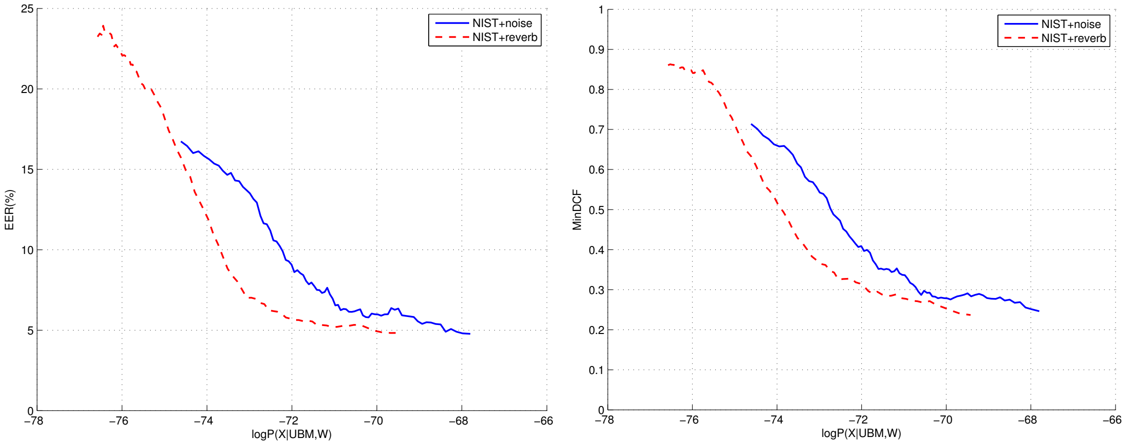
$\mathbf{Z}$ are the Gaussian assignments of the feature vectors; $\boldsymbol{\Sigma}$ is a matrix with the covariances of each Gaussian $\boldsymbol{\Sigma}_k$ in the diagonal; $\mathbf{N}$, $\overline{\mathbf{F}}$ and $\overline{\mathbf{S}}$ are the sufficient statistics of the utterance centered in the UBM. For each Gaussian, they are defined by

$$\mathbf{N}_k = \sum_{t=1}^{T} P\left(k|X_t\right) \tag{4.15}$$

$$\overline{\mathbf{F}}_k = \sum_{t=1}^{T} P\left(k|X_t\right)\left(X_t - \mathbf{m}_k\right) \tag{4.16}$$

$$\overline{\mathbf{S}}_k = \sum_{t=1}^{T} P\left(k|X_t\right)\left(X_t - \mathbf{m}_k\right)(X_t - \mathbf{m}_k)^T\ . \tag{4.17}$$

Thus, $\overline{\mathbf{F}}$ is a super-vector resulting of the concatenation of the $\overline{\mathbf{F}}_k$ and $\overline{\mathbf{S}}$ is a block diagonal matrix whose diagonal is composed by concatenating $\overline{\mathbf{S}}_k$. $\mathrm{E}\left[\phi\right]$ and $\mathbf{L}$ are the mean and



(a) EER (%)

(b) Minimum DCF

Figure 4.17: SV performance against the i-vector extractor LLk.

precision of the posterior distribution of ϕ and are given by

$$\mathbf{L} = \mathbf{I} + \sum_{k=1}^{K} N_k \mathbf{T}_k^T \mathbf{\Sigma}_k^{-1} \mathbf{T}_k \tag{4.18}$$

$$\mathrm{E}\left[\phi\right] = \mathbf{L}^{-1}\mathbf{T}^T\mathbf{\Sigma}^{-1}\overline{\mathbf{F}} \tag{4.19}$$

where $\mathbf{T}_k$ are the rows of $\mathbf{T}$ corresponding to the k^{th} Gaussian.

We need to normalize the log-likelihood by the number of frames T to fairly compare between different segments.

4.7.2 Correlation with SV performance

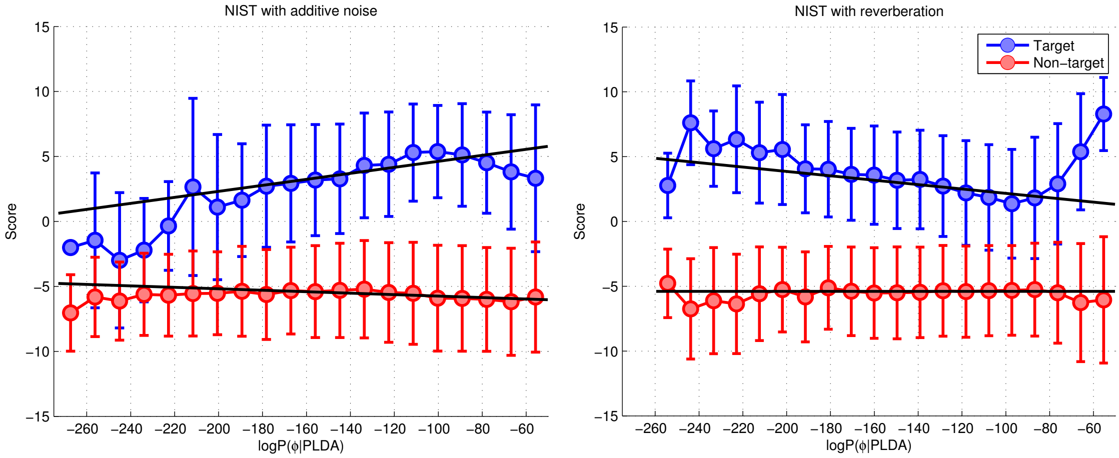
Figure 4.16 shows the relation between the target and non-target scores and the i-vector extractor log-likelihood of the segment. Figure 4.17 shows EER and minimum DCF against the log-likelihood. Comparing these figures with the ones corresponding to the UBM log-likelihood (Figures 4.14 and 4.15), we appreciate that the figures are quite similar.

Note that the term $\ln P\left(\mathcal{X}|\mathbf{Z},\mathbf{m},\mathbf{\Sigma}\right)$ in Equation (4.13) is very close to the log-likelihood of the UBM. Thus, this measure is approximately the log-likelihood of the UBM plus a second term that accounts for the fact that the utterances GMM can only move in the directions contained in $\mathbf{T}$. According to the figures, we can say that this second term does not have a great influence on the total likelihood. This measure does not add information about the SV performance complementary to the UBM likelihood.

4.8 i-Vector Likelihood Given the PLDA Model

4.8.1 Description

The i-vectors distribution is generally modeled by a linear Gaussian generative model called Probabilistic Linear Discriminant Analysis (PLDA). There are different flavor of PLDA that are described in detail in Appendix C. The SV system used in our experiments was based on



(a) (b)

Figure 4.18: Score range against the i-vector LLk for NIST SRE10 + additive noise (a) and reverberation (b). Black lines plot the linear regression between score and the i-vector LLk.

the simplified version of PLDA called *two-covariance model* or *full-rank* PLDA that writes an i-vector ϕ_{ij} of speaker i as:

$$\phi_{ij} = \mathbf{y}_i + \epsilon_{ij}. \tag{4.20}$$

where $\mathbf{y}_s$ is denoted as the *speaker identity variable* and ϵ is the *channel offset*. Between different observations of the speaker, the identity variable remains constant while the channel offset varies. The PLDA model $\mathcal{M}$ is defined by the two following probability distributions:

$$P(\mathbf{y}|\mathcal{M}) = \mathcal{N}\left(\mathbf{y}|\mu, \mathbf{B}^{-1}\right) \tag{4.21}$$

$$P(\phi|\mathbf{y}, \mathcal{M}) = \mathcal{N}\left(\phi|\mathbf{y}, \mathbf{W}^{-1}\right) \tag{4.22}$$

where $\mathcal{N}$ denotes a Gaussian distribution; μ is a speaker-independent mean; $\mathbf{B}^{-1}$ is the between speaker covariance matrix and $\mathbf{W}^{-1}$ is the within speaker covariance matrix.

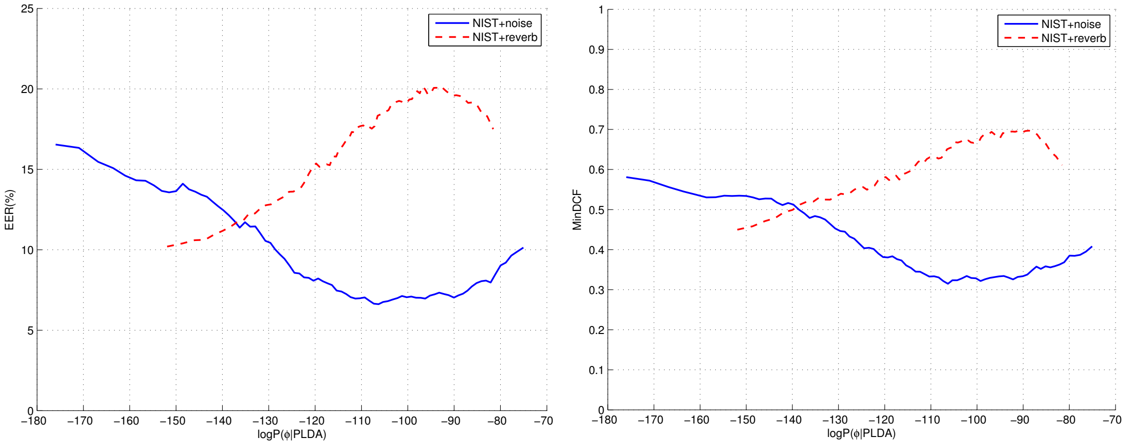
We proposed to use likelihood of the i-vector given the PLDA model as another quality measure. For the two-covariance model, that is given by

$$P(\phi|\mathcal{M}) = \mathcal{N}\left(\phi|\mu, \mathbf{B}^{-1} + \mathbf{W}^{-1}\right) \ . \tag{4.23}$$

As for the likelihood of the UBM, we expected i-vectors of degraded signals to produce lower likelihoods than the i-vectors of clean signals.

4.8.2 Correlation with SV performance

Figure 4.18 plots target and non-target scores against the i-vector log-likelihood of the test segment. Figure 4.19 shows EER and minimum DCF against the log-likelihood. The curves indicate that the effect of additive noise on the likelihood is radically different to the one of reverberation. While, for additive noise worse performance corresponds to lower values of likelihood, for reverberation the opposite happens. That means that reverberation moves the i-vector closer to the PLDA mean μ while additive noise moves it away. This behavior makes difficult to apply this measure as performance predictor unless we know the distortion type of the segments. By comparing with previous measures like SNR or modulation index,



(a) EER (%)

(b) Minimum DCF

Figure 4.19: SV performance against the i-vector LLk.

we observe that minimum EER and DCF are noticeably higher so this measure cannot indicate which trials are the best. Besides, the difference between maximum and minimum error rates is smaller. In consequence, the PLDA likelihood will not be a good feature to discriminate between reliable and unreliable trials by itself but, joined to others, it could add some complementary information.

4.9 VTS Parameters

4.9.1 Description

Figure 4.20 depicts the general model for an acoustic environment with additive and convolutive noise. The observed distorted speech signal $y(m)$ is generated from the clean speech signal $x(m)$ according to

$$y(m) = x(m) * h(m) + n(m) \tag{4.24}$$

where $n(m)$ is the noise signal and $h(m)$ is the impulsional response of the channel.

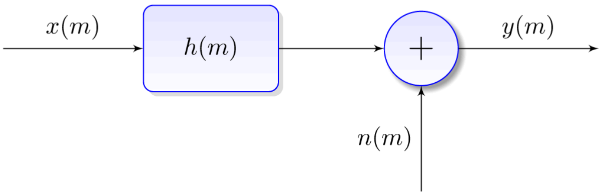


Figure 4.20: Model for environment distortion.

The equivalent relation can be established in the spectral domain by applying the Fourier transform,

$$|Y(f)| = |X(f)||H(f)| + |N(f)| \,. \tag{4.25}$$

Mapping (4.25) to the cepstral domain we obtain this well established non-linear distortion model:

$$\mathbf{y} = \mathbf{x} + \mathbf{h} + \mathrm{g}(\mathbf{x}, \mathbf{h}, \mathbf{n}) \tag{4.26}$$

$$\mathrm{g}(\mathbf{x}, \mathbf{h}, \mathbf{n}) = \mathbf{C} \ln\left(1 + \exp\left(\mathbf{C}^{-1}\left(\mathbf{n} - \mathbf{x} - \mathbf{h}\right)\right)\right) \tag{4.27}$$

where $\mathbf{C}$ is the non-square discrete cosine transform (DCT) matrix used to compute the MFCC and $\mathbf{C}^{-1}$ is its pseudo-inverse and; $\mathbf{y}$, $\mathbf{x}$, $\mathbf{h}$ and $\mathbf{n}$ are the vector valued distorted speech, clean speech, channel and noise in the MFCC domain.

Several methods had been proposed to jointly compensate additive and convolutive (JAC) distortions in speech recognition systems. One of the most successful approaches was introduced in [Moreno et al., 1996], and is based on using vector Taylor series (VTS) to approximate the non-linearity in (4.26) with a linear function. VTS makes analytically tractable the problem of estimating the noise and channel distributions of the segment. In that first work, the noise and channel means were used to compensate the MFCC features. The work in [Acero et al., 2000], instead of trying to estimate the clean features, proposes to adapt the GMM of the clean speech to the distorted space. This line of work was continued in [Li et al., 2007,Li et al., 2009,Kalinli et al., 2009] reaching considerably better performance than adapting the features. The approaches based on adapting the model succeed because they modify the GMM variances the take into account the uncertainty about the noise estimation while the approaches based on feature compensation just make point estimates of the noise values.

The first order VTS expansion of (4.26) with respect to $\mathbf{x}$, $\mathbf{n}$ and $\mathbf{h}$ around their mean values is given by

$$\mathbf{y} \approx \mu_x + \mu_h + \mathrm{g}(\mu_\mathbf{x}, \mu_\mathbf{h}, \mu_\mathbf{n}) + \mathbf{G}(\mathbf{x} - \mu_\mathbf{x}) + \mathbf{G}(\mathbf{h} - \mu_\mathbf{h}) + (\mathbf{I} - \mathbf{G})(\mathbf{n} - \mu_\mathbf{n}) \tag{4.28}$$

where

$$\left.\frac{\partial \mathbf{y}}{\partial \mathbf{x}}\right|_{\mu_\mathbf{x},\mu_\mathbf{h},\mu_\mathbf{n}} = \left.\frac{\partial \mathbf{y}}{\partial \mathbf{h}}\right|_{\mu_\mathbf{x},\mu_\mathbf{h},\mu_\mathbf{n}} = \mathbf{C}\mathrm{diag}\left(\frac{1}{1 + \exp\left(\mathbf{C}^{-1}(\mu_\mathbf{n} - \mu_\mathbf{x} - \mu_\mathbf{h})\right)}\right)\mathbf{C}^{-1} = \mathbf{G} \tag{4.29}$$

$$\left.\frac{\partial \mathbf{y}}{\partial \mathbf{n}}\right|_{\mu_\mathbf{x},\mu_\mathbf{h},\mu_\mathbf{n}} = \mathbf{I} - \mathbf{G}\ . \tag{4.30}$$

It is assumed that $\mathbf{x}$ is distributed as a GMM with mean $\mu_x = \{\mu_{\mathbf{x}_k}\}_{k=1}^K$ and covariance $\mathbf{\Sigma}_x = \{\mathbf{\Sigma}_{\mathbf{x}_k}\}_{k=1}^K$; $\mathbf{n}$ is Gaussian distributed with mean μ_n and covariance $\mathbf{\Sigma}_n$; $\mathbf{h}$ is constant along the utterance with value μ_h; and $\mathbf{x}$, $\mathbf{n}$ and $\mathbf{h}$ are statistically independent *a priori*. Being $\mathbf{x}$ GMM distributed, it is more appropriate to apply a different expansion for each component of the mixture. Then, if $\mathbf{y}$ belongs to the Gaussian k, we should write

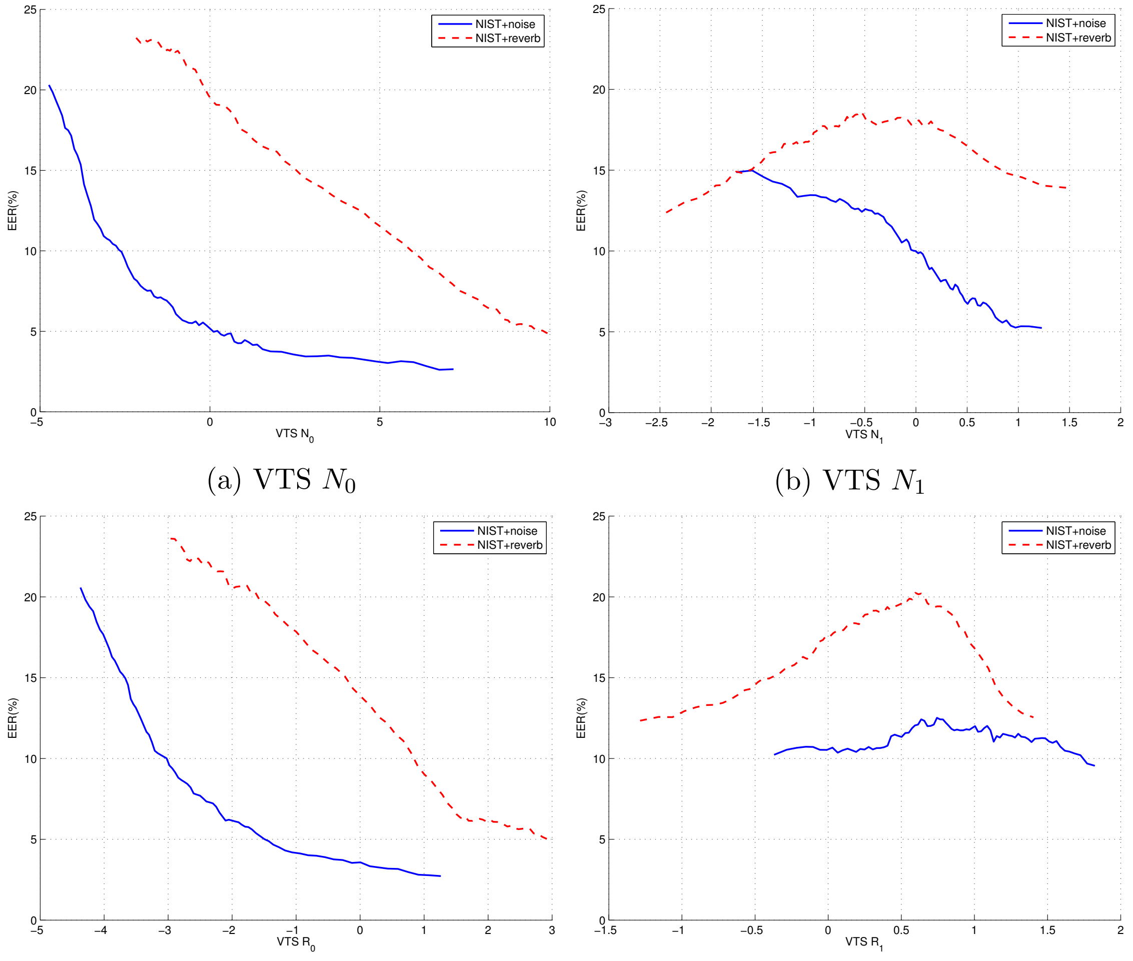
$$\mathbf{y} \approx \mu_{\mathbf{x}_k} + \mu_h + \mathrm{g}(\mu_{\mathbf{x}_k}, \mu_\mathbf{h}, \mu_\mathbf{n}) + \mathbf{G}_k(\mathbf{x} - \mu_\mathbf{x}) + (\mathbf{I} - \mathbf{G}_k)(\mathbf{n} - \mu_\mathbf{n})\ . \tag{4.31}$$

From (4.31), deriving the mean and variances of the GMM in the distorted space is straightforward. For the static MFCC, the mean vectors of the degraded GMM become

$$\mu_{\mathbf{y}_k} \approx \mu_{\mathbf{x}_k} + \mu_h + \mathrm{g}(\mu_{\mathbf{x}_k}, \mu_\mathbf{h}, \mu_\mathbf{n}) \tag{4.32}$$

and the covariance matrix

$$\mathbf{\Sigma}_{\mathbf{y}_k} \approx \mathbf{G}_k \mathbf{\Sigma}_{\mathbf{x}_k} \mathbf{G}_k^T + (\mathbf{I} - \mathbf{G}_k)\mathbf{\Sigma}_\mathbf{n}(\mathbf{I} - \mathbf{G}_k)^T \tag{4.33}$$



(a) VTS N_0

(b) VTS N_1

(c) VTS R_0

(d) VTS R_1

Figure 4.21: EER against VTS N and R measures.

For the means and variances corresponding to Δ and $\Delta\Delta$ the adaptation formulas are

$$\mu_{\Delta\mathbf{y}_k} \approx \mathbf{G}_k\mu_{\Delta\mathbf{x}_k} + (\mathbf{I} - \mathbf{G}_k)\mu_{\Delta n} \tag{4.34}$$

$$\mu_{\Delta\Delta\mathbf{y}_k} \approx \mathbf{G}_k\mu_{\Delta\Delta\mathbf{x}_k} + (\mathbf{I} - \mathbf{G}_k)\mu_{\Delta\Delta n} \tag{4.35}$$

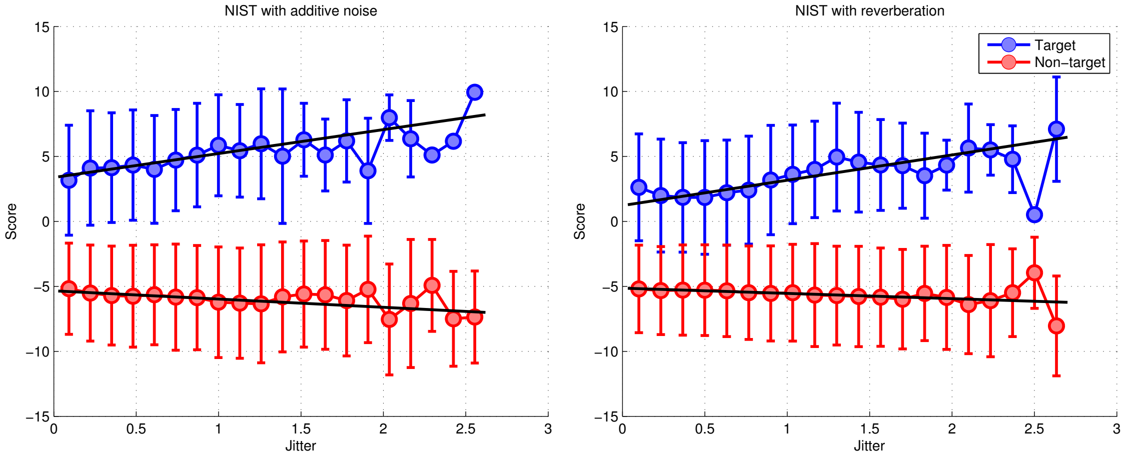
$$\mathbf{\Sigma}_{\Delta\mathbf{y}_k} \approx \mathbf{G}_k\mathbf{\Sigma}_{\Delta\mathbf{x}_k}\mathbf{G}_k^T + (\mathbf{I} - \mathbf{G}_k)\mathbf{\Sigma}_{\Delta n}(\mathbf{I} - \mathbf{G}_k)^T \tag{4.36}$$

$$\mathbf{\Sigma}_{\Delta\Delta\mathbf{y}_k} \approx \mathbf{G}_k\mathbf{\Sigma}_{\Delta\Delta\mathbf{x}_k}\mathbf{G}_k^T + (\mathbf{I} - \mathbf{G}_k)\mathbf{\Sigma}_{\Delta\Delta n}(\mathbf{I} - \mathbf{G}_k)^T. \tag{4.37}$$

The GMM of clean speech is previously trained on a set of clean signals. Then, for each degraded segment, the parameters $\mu_\mathbf{h}$, $\mu_\mathbf{n}$ and $\mathbf{\Sigma}_\mathbf{n}$ are estimated using an EM algorithm as described in [Li et al., 2009]. $\mu_\mathbf{n}$ and $\mathbf{\Sigma}_\mathbf{n}$ are usually initialized with silence frames of the utterance. The value of $\mu_\mathbf{h}$ is initialized to zeros.

We proposed to use the static means of noise $\mu_\mathbf{n}$ and channel $\mu_\mathbf{h}$ as starting point to derive some quality measures. We expected that $\mu_\mathbf{n}$ could help to infer the effect of additive noise on the SV performance and $\mu_\mathbf{h}$ the effect of reverberation. To reduce the cost of computing $\mu_\mathbf{n}$ and $\mu_\mathbf{h}$, we employed a GMM smaller than the one of the SV system. Our clean GMM was gender independent with 128 Gaussians and was trained on NIST SRE04–06.

The dimensionality of $\mu_\mathbf{n}$ and $\mu_\mathbf{h}$ is quite high so many pattern recognition techniques will not handle well these features. We applied a post-processing step of dimensionality



(a) (b)

Figure 4.22: Score range against jitter for NIST SRE10 + additive noise (a) and reverberation (b). Black lines plot the linear regression between score and jitter.

reduction to compress the information contained in them. First, we concatenated both means $\mu_{\mathbf{n}}$ and $\mu_{\mathbf{h}}$ into one feature vector μ. Afterwards, we applied two linear projections to μ based on linear discriminant analysis (LDA). The first projection $\mathbf{A}_N$ was trained on the dataset NIST SRE08 with additive noise where we took the SNR levels as the classes to discriminate. The second projection $\mathbf{A}_R$ was trained on NIST SRE08 with reverberation by taking the different reverberation times as classes. We denote by VTS N the measures with $\mathbf{A}_N$ and by VTS R the ones obtained with $\mathbf{A}_R$.

$$\begin{bmatrix}\text{VTS } N\\ \text{VTS } R\end{bmatrix}=\begin{bmatrix}\mathbf{A}_N\\ \mathbf{A}_R\end{bmatrix}\begin{bmatrix}\mu_{\mathbf{h}}\\ \mu_{\mathbf{n}}\end{bmatrix}\;. \tag{4.38}$$

We expected that VTS N will be correlated with the SV performance in the presence of noise while VTS R will be useful in the presence of reverberation.

4.9.2 Correlation with SV performance

Figure 4.21 plots the first two coefficients of VTS N and VTS R against the EER. For additive noise, curves evidence a correlation between N_0, N_1 and R_0 and EER. Low values of the features correspond to higher error rates. On the other hand R_1 is completely independent of the EER. Regarding the dataset with reverberation, we find that N_0 and R_0 are also clearly correlated with EER with low error rates for low values of the features and vice versa. N_1 and R_1 are also rather correlated with error with high error rates for central values and lower error rates for extreme values of the features. The minimum EER and ΔEER reached by N_0 and R_0 are comparable to those exhibited by the modulation index that is one of the most promising features shown until now. In the following chapters, we will expose that these measures were between the best performers distinguishing reliable and unreliable trials with classifiers based on Bayesian networks.

4.10 Jitter

4.10.1 Description

The jitter of the speech signal measures the cycle to cycle variation of the fundamental frequency f_0. It can be interpreted as a frequency modulation noise [Monzo et al., 2007]. We computed it by following the procedure described in [Monzo et al., 2008]. The pitch frequency was obtained with the same pitch detector that we used to compute our SNR, based on the RAPT algorithm [Talkin, 1995]. We used a logarithmic scale for the pitch that maps from Hertz to semitones

$$f_0' = 12 \log_2 \left(f_0 / f_{\text{ref}} \right) \text{ (semitones)} \tag{4.39}$$

where f_{ref} is the average pitch frequency of the segment. The normalization relative to the average frequency provides a better subjective representation of the frequency variations.

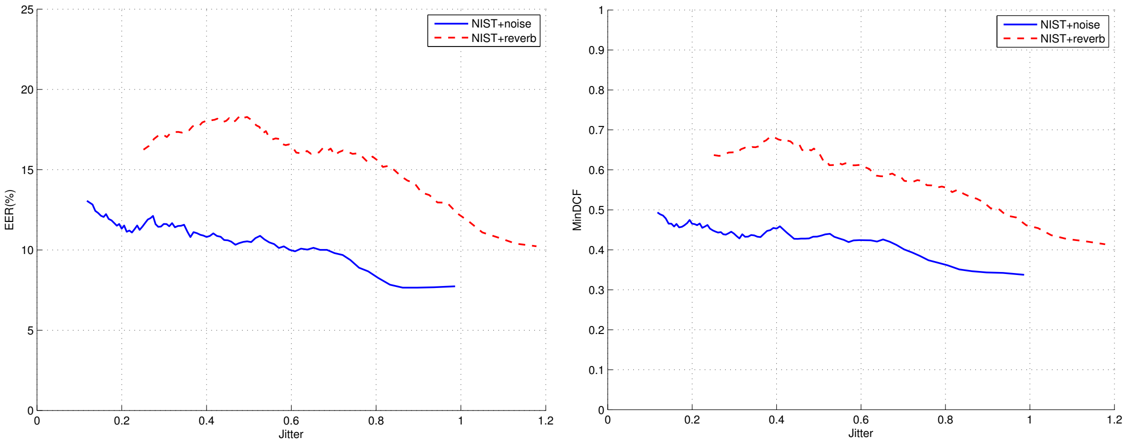
We compensated the effect of the sentence prosody on the pitch curve. With that purpose we detected the increase and decrease intervals of f_0' by analyzing the slope. We subtracted the result of applying linear regression to each interval from the original f_0'. Finally, the jitter of a frame i was computed from the compensated f_0' as

$$J_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \left(f_{0_i}'(j+1) - f_{0_i}'(j) \right)^2 \tag{4.40}$$

where N_i is the number of pitch cycles in the i^{th} frame.

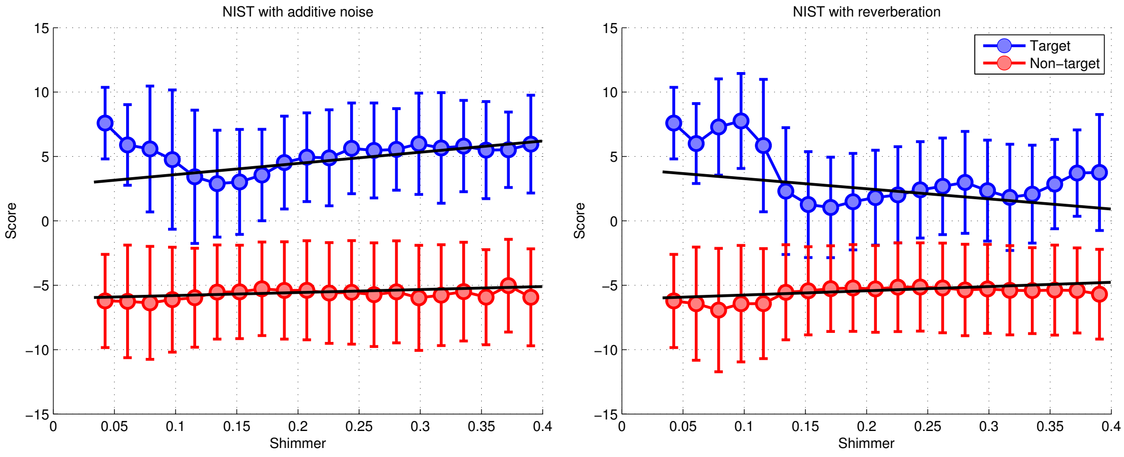
4.10.2 Correlation with SV performance

Figure 4.22 shows how score distributions evolve as a function of the test segment jitter. Figure 4.23 displays EER and minimum DCF against the jitter. For both noise and reverberation, the target distribution is more separated from the non-target one for higher jitter values. Accordingly, EER and DCF are also better for higher jitter. However the differences between low and high jitter are not as pronounced as for other measures. For



(a) EER (%) (b) Minimum DCF

Figure 4.23: SV performance against jitter.



(a) (b)

Figure 4.24: Score range against shimmer for NIST SRE10 + additive noise (a) and reverberation (b). Black lines plot the linear regression between score and shimmer.

example, ΔEER is around 6–8 that is much lower than the 20 that we showed for the modulation index. The minimum EER is also quite high (8–10%). Therefore, this measure of its own will not accurately detect reliable trials. Nevertheless, added to other measures could improve the performance of the reliability detector.

4.11 Shimmer

4.11.1 Description

The shimmer of a speech signal measures the cycle to cycle variation of the amplitude of the waveform. It can be interpreted as a amplitude modulation noise [Monzo et al., 2007]. We computed it by following the procedure described in [Monzo et al., 2008]. We computed the logarithm of the peak to peak amplitude for each period of f_0 in the voiced frames. The effect of the prosody was eliminated in the same manner described in Section 4.11.1 for the jitter.

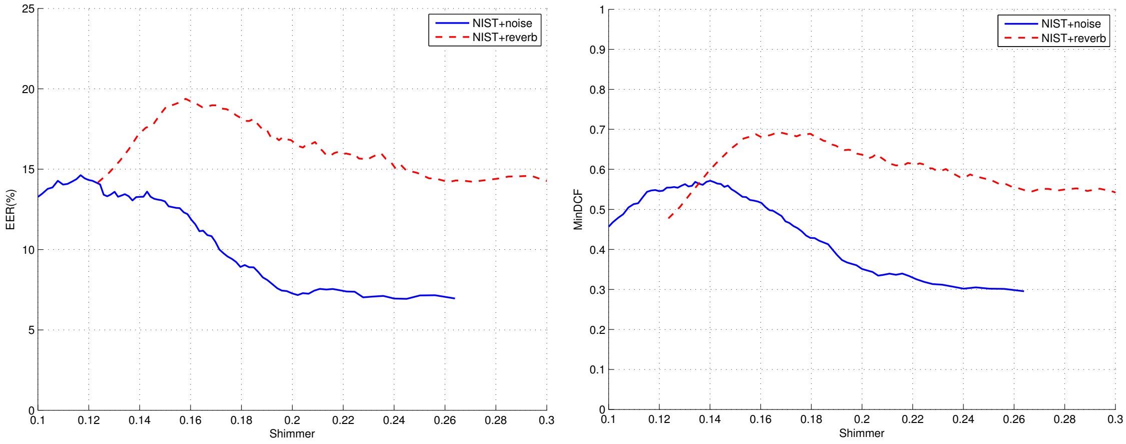
Finally, the shimmer of a frame i was computed as

$$S_i = \frac{1}{N_i}\sum_{j=1}^{N_i}\left(A_{\mathrm{pp}_i}(j+1) - A_{\mathrm{pp}_i}(j)\right)^2 \tag{4.41}$$

where A_{pp_i} is the logarithm of the peak to peak amplitude with prosody compensation and N_i is the number of pitch cycles in the i^{th} frame.

4.11.2 Correlation with SV performance

Figure 4.24 displays the target and non-target scores distributions against the shimmer of the segment. Figure 4.25 plots EER and minimum DCF against shimmer. Scores do not evolve linearly with shimmer, especially for reverberation. Scores get closer for shimmer around 0.15 and move away for lower and higher values. The performance in terms of EER



(a) EER (%)

(b) Minimum DCF

Figure 4.25: SV performance against shimmer.

and DCF, for additive noise, is close to the one of jitter with similar minimum and Δ values. For reverberation, it performs poorly with a very high minimum EER (14%).

4.12 Saturation Detection

4.12.1 Description

Saturation generally occurs in systems where signals are converted from analog to digital. The samples that overflow the dynamic range of the converter are saturated. The saturation level depends on the number of bits of the converter. Saturation can considerably degrade performance in speech and speaker recognition systems. We developed a saturation detector that considers that the saturation level is unknown *a priori*. In essence, the saturation level can be lower than maximum dynamic range of the file. This assumption is based on the possibility that the signal can suffer different amplitude changes in the transmission channel after it saturates. Besides, we would observe different positive and negative saturation levels if we removed the DC-offset after saturation.

Figure 4.26 shows the block diagram that implements our saturation detector. The algorithm was based on several measures. First, we considered that saturated samples should have almost identical values close to the saturation levels. Therefore, frames having a large number of local maxima of similar value are likely to be saturated. We clustered together frames with similar local maxima. If a local maximum appears many times inside a frame or in a cluster of frames we marked all the frames of that cluster as saturated.

Another indicator of saturation was based on the fact that standard speech samples follow a Laplace distribution. That distribution changes for saturated speech because the concentration of samples in the high part of the histogram grows. We measured the deviation from normal speech by computing the ratio between the number of samples in the high and low parts of the histogram.

Finally, saturation is a non-linear distortion and, as such, it makes new harmonics to appear in high frequencies. We measured this effect with the ratio between the energy in high frequency (3.5–4 kHz) and medium-high frequency (3–3.5 kHz).

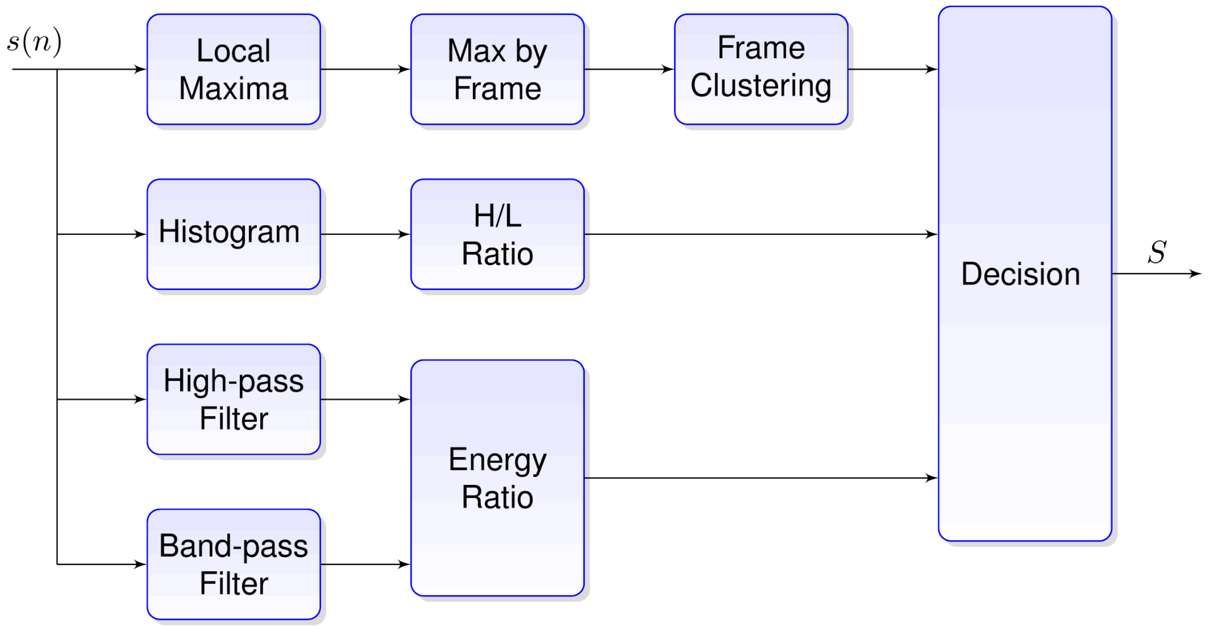


Figure 4.26: Block diagram of the saturation detector.

This three measures (number local maxima by frame, high/low histogram ratio and high/medium-high energy ratio) were combined by an heuristic method to decide which frames are saturated.

4.12.1.1 Local maxima by frame

Saturated samples are all clipped to values close to the saturation level. Thus, in saturated signals we find many samples with almost identical values. Besides, those samples are local maxima, that is, adjacent samples have smaller absolute values. We looked for frames having a large number of those maxima and marked them as candidates to be saturated. We denote by M the set of local maxima of the frame. We defined the set S_1 of samples candidates to be saturated as

$$S_1 = \left\{ s(n) \in M \,\middle|\, \frac{\max(M) - s(n)}{\max(M)} < \xi_{\text{cluster}} \right\} \tag{4.42}$$

where the threshold ξ_{cluster} was selected empirically. Additionally, we considered that samples next to the saturated local maxima could also be saturated if their value were close to the local maxima. We defined the set S_2 of saturated candidates next to samples in S_1 as

$$S_2 = \left\{ s(n) \,\middle|\, \exists x \in \{s(n-1), s(n+1)\} \cap S_1,\ \frac{x - s(n)}{x} < \delta_{\text{sat}} \right\}\ . \tag{4.43}$$

Finally, the candidates to be saturated samples are the ones that belong to the set $S = S_1 \cup S_2$. The number of elements of S, $|S|$ is a measure of the degree of saturation of a frame. We applied this procedure for negative and positive samples by separate because for signals with had DC offset when they were saturated the positive and negative saturation levels will be different.

4.12.1.2 Frame clustering

In a complete signal, there are frames with a high number saturated samples and others with one or two samples. If we just apply a threshold on $|S|$ to decide whether a frame is saturated, we will detect the formers but not the latter. To detect all the saturated frames we clustered together all the frames with similar absolute maxima. From now on, we denote by S_i the samples candidates to be saturated in the frame F_i. We define n as the current number of clusters. We define A as the set of frames that does not belong to any cluster C_j with $j = 1, \ldots, n$.

$$A = \{F_i | F_i \notin C_j, j = 1, \ldots, n\} \ . \tag{4.44}$$

The frame clustering procedure is listed in Algorithm 1. It starts with zero clusters and all the frames in A. While A is not empty, it selects the frame from A that has the largest absolute maximum x_n and it creates a cluster with all the frames with absolute maxima close to x_n.

Algorithm 1 Frame clustering algorithm.

```
$n = 0$
$A = \{F_i, \forall i\}$
while $A \neq \varnothing$ do
    $n = n + 1$
    $x_n = \max_i (\{\max(S_i) | F_i \in A\})$
    $C_n = \left\{F_i \in A | \frac{x_n - \max(S_i)}{x_n} < \xi_{\text{cluster}}\right\}$
    $A = A - C_n$
end while
```

For every cluster, we calculated the total number of saturated candidates, the average candidates by frame and the maximum candidates by frame:

$$s_j = \left| \bigcup_{F_i \in C_j} S_i \right| \tag{4.45}$$

$$m_j = \frac{s_j}{|C_j|} \tag{4.46}$$

$$x_j = \max_i \{|S_i| \, |F_i \in C_j\} \tag{4.47}$$

with $j = 1, \ldots, n$.

Each cluster corresponding to one of the saturation levels presents a high s_j value. We took the N clusters with larger s_j where $N = 1$ if there is one side of the conversation in each channel and $N = 2$ if both sides of the conversation are summed into the same channel. Then, we imposed a threshold to m_j and x_j to decide whether the cluster frames were saturated.

4.12.1.3 High/Low histogram ratio

While the samples of normal speech follow a Laplace distribution, saturated speech distributes concentrating a larger number of samples in high absolute values, see Figure 4.27.

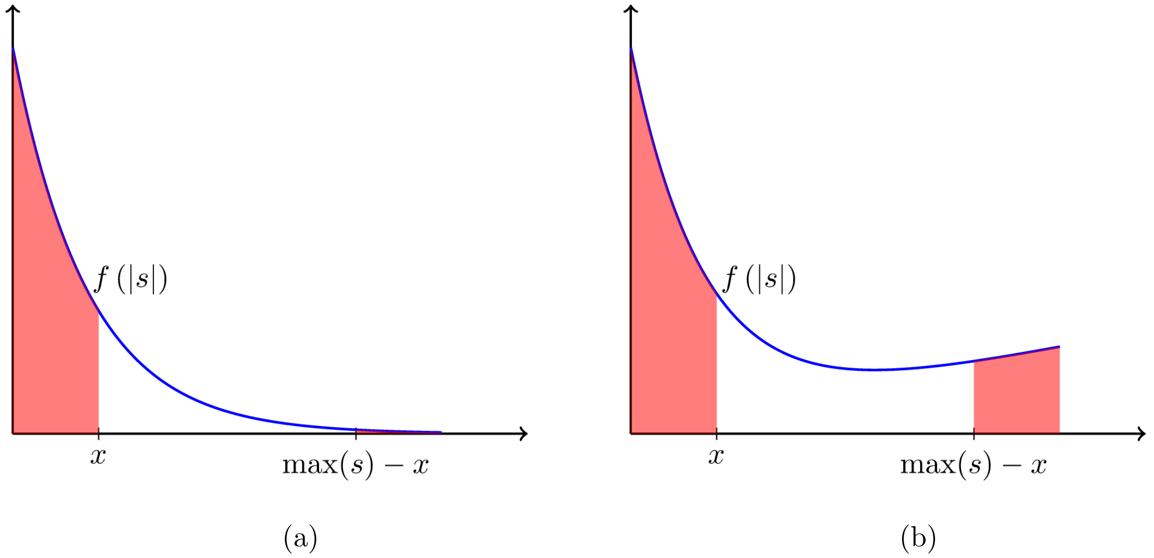


Figure 4.27: Histograms of non saturated (a) and saturated (b) signals.

We measured this effect by calculating the ratio between the number of samples in the high part of the histogram and in the low part of the histogram:

$$R(x) = \frac{\int_{\max(|s|)-x}^{\max(|s|)} f(|s|)\ \mathrm{d}s}{\int_0^x f(|s|)\ \mathrm{d}s} \tag{4.48}$$

where x defines the range of values that we consider high and low.

Not all the files have the same optimum value for x. To avoid fixing the value of x, we supposed that the optimum x has a probability distribution $p(x)$ in the domain $[0, max(|s|)/2]$, and we computed the expectation of R:

$$R = \int_0^{\max(|s|)/2} R(x)p(x)\ \mathrm{d}x\ . \tag{4.49}$$

We observed that results are not very dependent on the form of $p(x)$. Finally, we chose a triangular distributed $p(x)$.

4.12.1.4 High/Medium-high energy ratio

When a signal is saturated it experiences a non-linear distortion. Non-linear distortions move energy from some frequencies to others. Therefore, we could detect saturation by searching for energy in frequencies where it should not be. In telephone speech, that is in frequencies over 3.5 kHz. We computed the ratio between the energy in the interval 3.5-4 kHz and 3-3.5 kHz. This ratio should be higher for saturated speech than for non-saturated. Figure 4.28 shows the comparison between the spectrum of a signal before and after applying a strong saturation.

4.12.1.5 Decision

To decide if a frame is saturated we used a heuristic algorithm based on combining the all three measures explained above. We decided that a frame was saturated if it belonged to

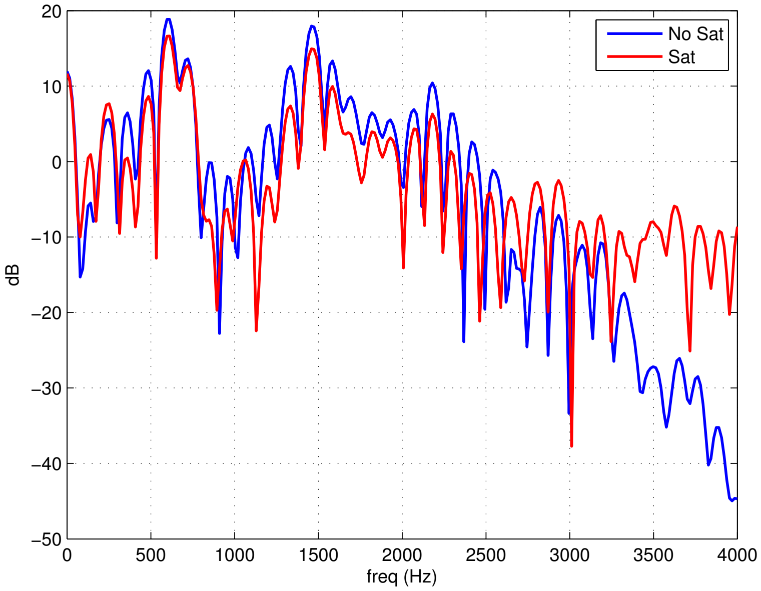
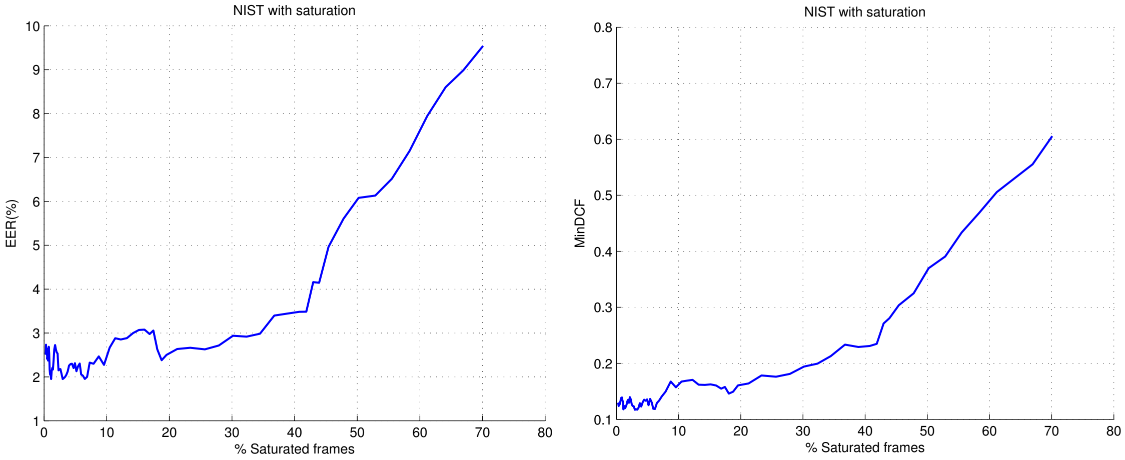


Figure 4.28: Saturated vs. no saturated spectrum.

cluster marked as likely saturated and if the high/low histogram and high/medium-high frequency ratios were over a threshold.

4.12.2 Correlation with SV performance

We saturated the test part of NIST SRE10 in different degrees as described in Section 3.4.2.3. Figure 4.29 plots EER and minimum DCF against the detected percentage of saturated frames in the test segment. A low percentage of saturation did not affect too much the error rates. It was from 40% of saturated frames that the EER and DCF started to grow faster. A frame was considered saturated with just one saturated sample. This results indicate that, to note the effect of saturation on the MFCC, we need to have many saturated samples in the frame. That only happens when we detect a high rate of saturated frames.



(a) EER (%)

(b) Minimum DCF

Figure 4.29: SV performance against detected percentage of saturated frames.

4.13 Combining Quality Measures

4.13.1 Description

We could combine the measures above described just by concatenating them into one feature vector. However, high dimensionality vectors may not be appropriate as input for some pattern recognition algorithms like models with Gaussian mixture distributions. In those cases, some dimensionality reduction preprocessing is recommended. We applied the same technique that we used in Section 4.9 to reduce the dimensionality of the VTS parameters. We formed a feature vector $\mathbf{x}$ by concatenating: signal-to-noise ratio, modulation index, entropy, UBM log-likelihood, VTS N_0, VTS N_1, jitter and shimmer. Then, we trained two LDA projections on NIST SRE08, one $\mathbf{B}_N$ to discriminate between noise levels and another one $\mathbf{B}_R$ to discriminate between reverberation times. We denote by N the feature resulting by projecting $\mathbf{x}$ with $\mathbf{B}_N$ and by R the one obtained by projecting with $\mathbf{B}_R$.

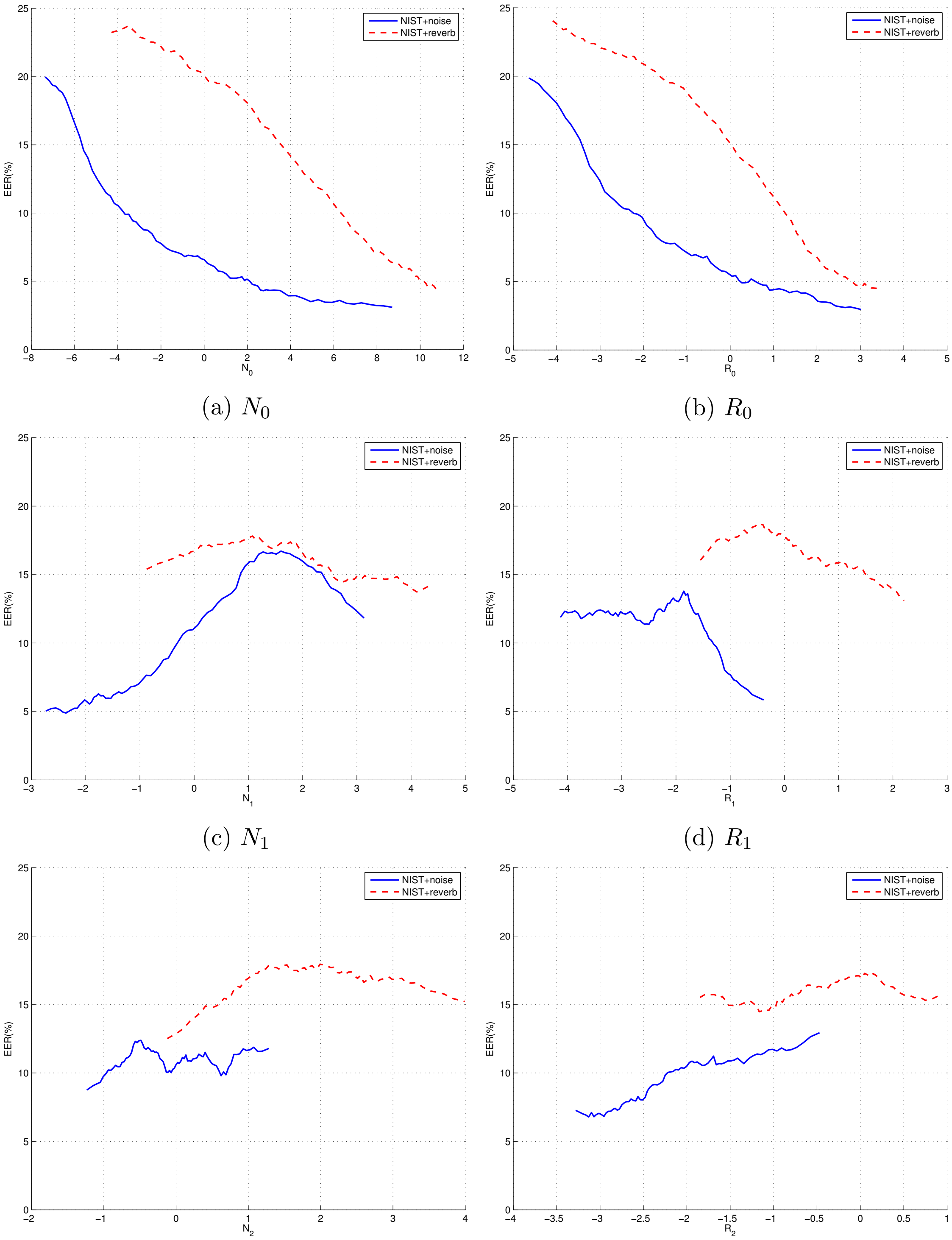
$$\begin{bmatrix} N \\ R \end{bmatrix} = \begin{bmatrix} \mathbf{B}_N \\ \mathbf{B}_R \end{bmatrix} \mathbf{x} \; . \tag{4.50}$$

4.13.2 Correlation with SV performance

Figure 4.30 plots the first three coefficients of N and R against EER. Regarding the dataset with additive noise, the figure proves that N_0, N_1 and R_0 are very correlated with performance. The difference between the EER at different values of the features is significant. N_0 and R_0 present minimum EER and ΔEER comparable with those of modulation index and VTS parameters. For N_1, ΔEER is about 30% smaller. For values of R_1 under -1.5, (the most part of the curve), the error rate is constant; for values above -1.5 the error rate decreases. N_2 is independent of the error. Finally, R_2 presents some dependency but it is small compared to N_0 and R_0. Now, looking at the curves for reverberation, we detect again an obvious relation between performance and N_0 and R_0. N_2 and R_1 denote a mild dependency with EER but far from that of N_0 and R_0. The curves for N_1 and R_2 are almost flat so we can deduce that they are independent of the EER.

4.14 Comparison of Quality Measures

In the following chapters, we will model the relation between the reliability of the speaker verification decisions and the quality measures with Bayesian networks. We will compare networks employing different quality measures and, thus, we will determine which ones contain more information regarding speaker verification performance. However, the figures presented in this chapter relating EER and DCF with each measure can provide some insights about what we should expect. As we said in previous sections, a good quality measure should be able to discriminate between the best and the worst trials. With this in mind, we conjectured that a quality measure is good if some values of the measure correspond to trials that are mostly well classified while other values always correspond to trials badly classified. On the other hand, if the quality measure is bad the rate of miss classified trials will be independent of the value of the measure. In other words, considering the figures that plot EER against the value of the measure, a measure should be better than others if it exhibits a larger difference between the highest and lowest error rate. Besides, the measures with larger difference of error rates should present lower minimum error rates.



(a) N_0

(b) R_0

(c) N_1

(d) R_1

(e) N_2

(f) R_2

Figure 4.30: EER against N and R measures.

Table 4.1: Ranking of quality measures ordered by ΔEER

(a) Additive noise.

Rank	Measure	Min EER(%)	ΔEER(%)
1	MI	2.91	19.96
2	VTS R_0	2.72	17.86
3	VTS N_0	2.64	17.67
4	R_0	2.95	16.90
5	N_0	3.08	16.89
6	SNR	3.59	15.24
7	# frames	4.87	13.11
8	UBM LLk	4.42	12.89
9	i-vector JFA LLk	4.77	11.96
10	N_1	5.03	11.56
11	i-vector PLDA LLk	6.61	10.03
12	Entropy	4.21	11.34
13	VTS N_1	5.23	9.68
14	R_1	6.05	7.44
15	Shimmer	7.14	7.23
16	R_2	6.96	5.97
17	Jitter	7.73	5.33
18	N_2	8.75	3.63
19	VTS R_1	9.54	2.88

(b) Reverberation.

Rank	Measure	Min EER(%)	ΔEER(%)
1	MI	3.88	20.19
2	R_0	4.39	19.64
3	UBM LLk	4.55	19.37
4	N_0	4.45	18.79
5	i-vector JFA LLk	4.83	18.72
6	VTS R_0	5.00	18.60
7	VTS N_0	4.81	18.42
8	SNR	4.90	16.53
9	Entropy	6.09	15.92
10	i-vector PLDA LLk	10.02	10.04
11	Jitter	10.23	8.01
12	VTS R_1	12.34	7.91
13	VTS N_1	12.37	6.05
14	R_1	13.09	5.64
15	N_2	12.51	5.36
16	Shimmer	14.07	5.31
17	N_1	13.95	3.86
18	# frames	13.62	3.47
19	R_2	14.47	2.80

Table 4.1a orders the quality measures by EER difference (ΔEER) for the dataset with additive noise and Table 4.1b do it for the dataset with reverberation. The maximum ΔEER for both noise and reverberation is around 20. We considered that the best measures are the ones having ΔEER larger than 75% of the maximum (ΔEER¿15). Thus, the best measure for detecting performance degradation in the presence of additive noise are modulation index, VTS N_0, VTS R_0, N_0, R_0 and signal-to-noise ratio. Regarding reverberation the best ones are modulation index, R_0, UBM log-likelihood, N_0, the likelihood of the i-vector extractor, VTS R_0, VTS N_0, signal-to-noise ratio and entropy. These measures also have lower minimum error rates, they are lower than 3.6 for noise and lower than 6.1 for reverberation. The fact that minimum error rates are larger for reverberation than for noise indicates that small reverberation affect performance but it is not detected by our quality measures and only when the reverberation becomes larger the effect can be noted in the measures. Looking at Table 3.2 EER under 6 correspond to reverberation times lower than 75 msec. For additive noise, looking at Table 3.1 an EER under 3.6 corresponds to clean and 20 dB trials.

Correlated measures are expected to provide similar reliability detection performance. We computed the normalized cross correlation between measures for both datasets noisy and reverberant. We found that modulation index, VTS N_0, VTS R_0, N_0 and R_0 are strongly correlated (> 0.9). We found also some correlation (> 0.7) between those measures and SNR and between UBM log-likelihood and entropy.

4.15 Summary

In this chapter, we described a set of quality measures that can be used to estimate the reliability of the speaker verification decisions. Some of them have been previously used in other works: signal-to-noise ratio, spectral entropy, number of speech frames, log-likelihood of the MFCC given the UBM, jitter and shimmer. The rest are novel contributions of this thesis: likelihood of the features given i-vector extractor model, log-likelihood of the i-vector given the PLDA model, VTS parameters, and saturation detection.

The algorithms to estimate SNR and saturation were developed as part of our work. Our SNR estimator takes advantage of the properties of voiced speech. Energy of voiced speech is contained in multiples of the pitch frequency. To estimate the speech power, a comb filter samples the energy in the multiples of the pitch frequency. Another filter infers the noise power by sampling in the frequencies between multiples. The ratio between both energies is related to the SNR of the signal. Then, we applied an heuristic procedure to calibrate that ratio into a good approximation of the SNR.

We designed a saturation detector that could be used in situations where the saturation level is unknown. The algorithm is based on three measures. First, we look for local maxima in the speech signal. If there are many local maxima with the same value it is probable that the signal is saturated and that value correspond to the saturation level. Second, we check whether the distribution of speech samples deviates from the Laplace distribution. We do it by evaluating the ratio between the number of samples in high and low parts of the histogram. And third, we took into account that saturation is a non-linear distortion so it makes energy to appear in frequencies where there was not before. To quantify that, we compute the ratio between the energy in high and medium-high frequencies. Finally these three features are combined heuristically to decide which frames are saturated.

One of the more promising features where the ones that we called VTS parameters. The non-linear effect of noise and reverberation on MFCC was approximated by a linear function by applying vector Taylor series. That function allows us to compute the mean and variances of a GMM for noisy speech from the GMM trained on clean speech. The Taylor series coefficients depend on the mean and variances of the noise and channel in cepstral domain. We estimated them for each segment by EM iterations. We derived quality measures by reducing the dimensionality of the means of noise and channel with linear discriminant analysis optimized to discriminate between noise and reverberation levels.

We also combined several measures by concatenating them and applying LDA projections optimized to discriminate noise levels and reverberation times. We ranked our measures by their ability to distinguish between groups of trials with low and high EER. These rankings revealed that, for additive noise, modulation index, signal-to-noise ratio, VTS parameters and the combination off all measures are better. And for reverberation, we have the same as for noise adding the UBM log-likelihood and entropy.

Chapter 5

Reliability Estimation from the Speaker Verification Score and Quality Measures

5.1 Introduction

In the previous chapter, we presented some quality measures and proved that they are related to the speaker verification performance. Now, we need some kind of mathematical model to combine those measures into a unique value that expresses the reliability of each verification decision. Previous works on this matter were already commented in Section 3.2 of Chapter 3. In the present chapter, we revisit Richiardi's approach [Richiardi et al., 2005, Richiardi et al., 2006b] based on Bayesian networks (BN). The networks establish causal relationships between the random variables involved in the speaker verification process (SV score, quality measures, true trial label or trial decision, etc.). The trial reliability is one of those variables and can take two possible values: true if the SV decision is right or false if it is wrong. The BN facilitates computing the posterior probability for the reliability. The results shown in [Richiardi et al., 2006a] indicate that BNs outperform previous techniques for estimating the trial reliability.

In [Richiardi et al., 2006b], the authors only used SNR spectral entropy. Here, we extend Richiardi's work by introducing a wider set of measures. Besides, we compare some variants of the BN configuration where we modify the dependencies between variables.

This chapter is organized as follows. Section 5.2 defines reliability and describes the Bayesian networks for reliability estimation. The network is defined through its graphical model and the conditional distribution of each node given its parents. We considered several network variants: networks with or without the SV score; and networks with quality measures dependent or independent on the trial label. Section 5.3 presents experiments on NIST SRE with added noise and reverberation and other databases with real distortions. From all the measures listed in the previous chapter, we show results with the ones that performed better in combination with the BN. We obtained good results training the networks on SRE08 and testing on SRE10 but those networks did not generalize well for the rest of datasets. Finally, Section 5.4 summarizes the chapter.

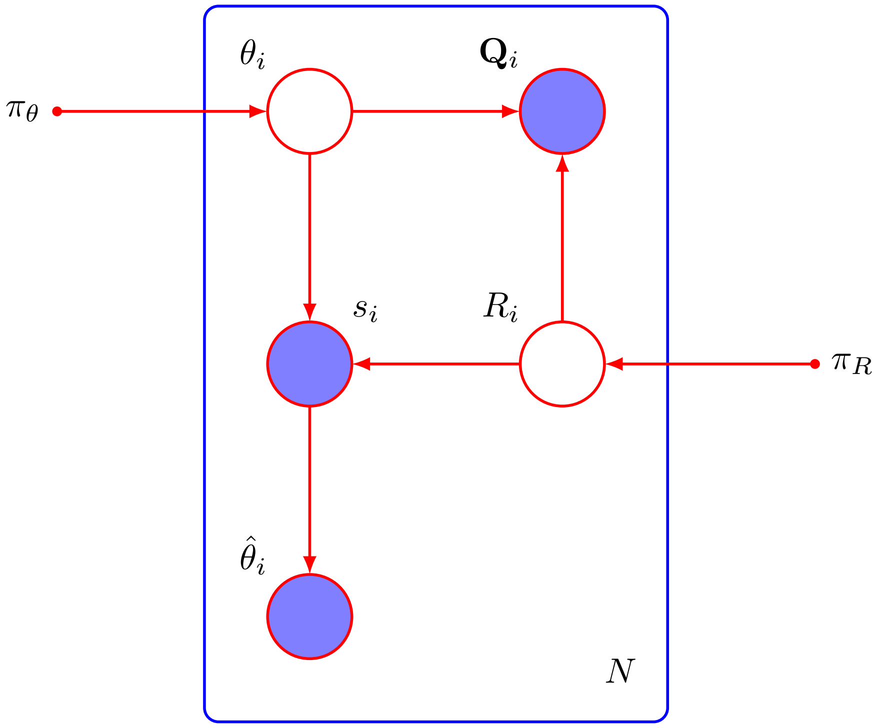


Figure 5.1: BN for reliability estimation based on score and quality measures.

5.2 Bayesian Networks for Reliability Estimation

5.2.1 Bayesian network description

To estimate a global measure of the trial reliability from the quality measures, we adopted the approach introduced in [Richiardi et al., 2005, Richiardi et al., 2006a]. These works model the relationships between the random variables involved in the verification process with a Bayesian network (BN). A Bayesian network is a directed graphical model [Bishop, 2006] that describes the dependencies of a set of random variables.

Figure 5.1 shows the BN that illustrates our problem. Empty nodes denote *hidden variables*, shaded nodes denote *observed variables* and small solid nodes denote *deterministic parameters*. A node or group of nodes surrounded by a box, called a *plate*, labeled with N indicates that there are N nodes of that kind (for example N trials). The arcs between the nodes point from the parent variables to the children variables. The arcs directionality can be interpreted as a cause-effect relationship, that is, the value of the children variables is conditioned by the value of its parents. Finally, to have the network completely defined we need the set of conditional probability distributions of each variable given its parents.

Following, we introduce the variables included in the graph. For each trial i, we have a label $\theta_i \in \{\mathcal{T}, \mathcal{N}\}$ where $\mathcal{T}$ is the hypothesis that the enrollment and test segments belong to the same speaker and $\mathcal{N}$ that they belong to different speakers. $\pi_\theta = (P_\mathcal{T}, P_\mathcal{N})$ is the hypothesis prior where $P_\mathcal{T}$ is the target prior and $P_\mathcal{N} = 1 - P_\mathcal{T}$ is the non-target prior. The variable $\hat{\theta}_i$ is the SV decision after applying a threshold ξ_θ to score s.

$$\hat{\theta}_i = \begin{cases} \mathcal{T} & \text{if } s_i \geq \xi_\theta \\ \mathcal{N} & \text{if } s_i < \xi_\theta \end{cases} \tag{5.1}$$

The variable $R_i \in \{\mathcal{R},\mathcal{U}\}$ is the trial reliability where $\mathcal{R}$ is the hypothesis that the decision is reliable and $\mathcal{U}$ that it is unreliable. Reliability is defined as

$$R_i = \begin{cases} \mathcal{R} & \text{if } \hat{\theta}_i = \theta_i \\ \mathcal{U} & \text{if } \hat{\theta}_i \neq \theta_i \end{cases} \tag{5.2}$$

Finally, $\pi_R = (P_{\mathcal{R}}, P_{\mathcal{U}})$ is the reliability prior where $P_{\mathcal{R}}$ is the prior probability of reliable decision and $P_{\mathcal{U}} = 1 - P_{\mathcal{R}}$ is the prior of non-reliable decision. θ_i and R_i are observed when training the network and hidden during the evaluation phase.

The SV score s_i depends on the true label of the trial θ_i and on the reliability R_i through a Gaussian conditional distribution

$$P\left(s_i|\theta_i = \theta, R_i = R\right) = \mathcal{N}\left(s_i|\mu_{s_{\theta R}}, \mathbf{\Lambda}^{-1}_{s_{\theta R}}\right) \ . \tag{5.3}$$

The score dependence on θ_i is justified because if $\theta_i = \mathcal{T}$, s_i is expected to be high and if $\theta_i = \mathcal{N}$, s_i is expected to be low. However, for target trials, s_i will be really high only if the trial is reliable, otherwise s_i will be low. The contrary will happen for non-targets, if the trial is reliable s_i will be low and it will be high otherwise. Because of that, we say that s_i also depends on R_i.

The quality measures of the trial $\mathbf{Q}_i$ also depend on θ_i and R_i and are modeled by a mixture of Gaussians

$$P\left(\mathbf{Q}_i|\theta_i = \theta, R_i = R\right) = \sum_{k=1}^{K} w_k \mathcal{N}\left(\mathbf{Q}_i|\mu_{\mathbf{Q}_{\theta R_k}}, \mathbf{\Lambda}^{-1}_{\mathbf{Q}_{\theta R_k}}\right) \ . \tag{5.4}$$

We say that $\mathbf{Q}_i$ depends on R_i because, for example, if a trial is unreliable due to noise in the speech, that will be detected for the quality measure SNR. We also add an arc between θ and $\mathbf{Q}$ because degradations can affect differently to target and non-target trials. For example, a small amount of noise reduces much the score of the targets but it does not increase the score of the non-targets.

We can write the joint probability distribution of the variables as a product of the conditional distributions of the BN:

$$P\left(s_i, \mathbf{Q}_i, R_i, \theta_i, \hat{\theta}_i|\pi_\theta, \pi_R\right) = P\left(s_i|R_i, \theta_i\right) P\left(\mathbf{Q}_i|R_i, \theta_i\right) P\left(\hat{\theta}_i|\theta_i, R_i\right) P\left(\theta_i|\pi_\theta\right) P\left(R_i|\pi_R\right) \ . \tag{5.5}$$

$P\left(\hat{\theta}_i|\theta_i, R_i\right)$ is a discrete distribution that is 1 if $\hat{\theta}_i = \theta_i$ and $R_i = \mathcal{R}$ or if $\hat{\theta}_i \neq \theta_i$ and $R = \mathcal{U}$ and it is 0 otherwise.

From (5.5), we can compute the posterior of R given the observed variables by applying the Bayes rule:

$$P\left(R_i|s_i, \mathbf{Q}_i, \hat{\theta}_i, \pi_\theta, \pi_R\right) = \frac{\sum_{\theta\in\{\mathcal{T},\mathcal{N}\}} P\left(s_i, \mathbf{Q}_i, R_i, \theta_i, \hat{\theta}_i|\pi_\theta, \pi_R\right)}{\sum_{R\in\{\mathcal{R},\mathcal{U}\}} \sum_{\theta\in\{\mathcal{T},\mathcal{N}\}} P\left(s_i, \mathbf{Q}_i, R_i, \theta_i, \hat{\theta}_i|\pi_\theta, \pi_R\right)} \ . \tag{5.6}$$

Before performing inference, we need to learn the parameters of the conditional distributions that define the network. The training samples must consist of a set of trials with different degradation levels. The labels θ of the training set are known, and

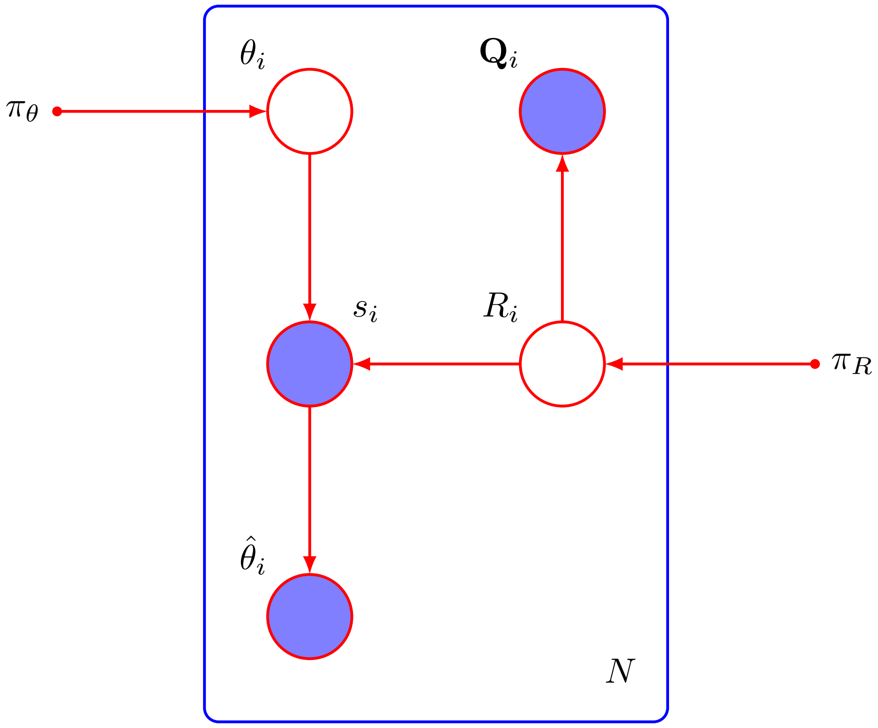


Figure 5.2: BN with $\mathbf{Q}$ independent of θ.

as the decisions $\hat{\theta}$ are observed, the reliabilities R also become observed. As all the variables are observed the distributions $P\left(s_i|\theta_i,R_i\right)$ can be trained by simple maximum likelihood. For the distributions $P\left(\mathbf{Q}_i|\theta_i,R_i\right)$, that are GMM, the variables that indicate the Gaussian occupations are still hidden. To train these GMM, we used standard EM iterations [Dempster et al., 1977].

This model has some drawbacks coming from the fact that, to train the network, we make hard decisions on the reliability variable of the training trials. First of all, the trial classification, as reliable or unreliable, depends on the SV threshold. That implies that if we change the operating point of the SV system the trials that are reliable or unreliable also change and we need to retrain the network. Secondly, for example, if we have two target trials with similar quality measures but with SV scores slightly under and over the threshold respectively, intuitively both should be given the same degree of reliability, but instead, one of them will be used to train the distributions conditioned on the reliable hypothesis and the other to train the distribution conditioned on the unreliable hypothesis. Thus, similar values of the measures are used to train opposite models what we think that damages the discriminative capabilities of the BN. In the next chapter, we will present an alternative model that overcomes these drawbacks and does not need to be retrained if we change the operating point of the system.

5.2.2 Bayesian network variants

Our Bayesian network differs from that in [Richiardi et al., 2006a], shown in Figure 5.2, in which ours adds a link from θ to $\mathbf{Q}$. In this manner, our BN takes into account that speech degradations can affect differently to targets and non-target trials while the other does not.

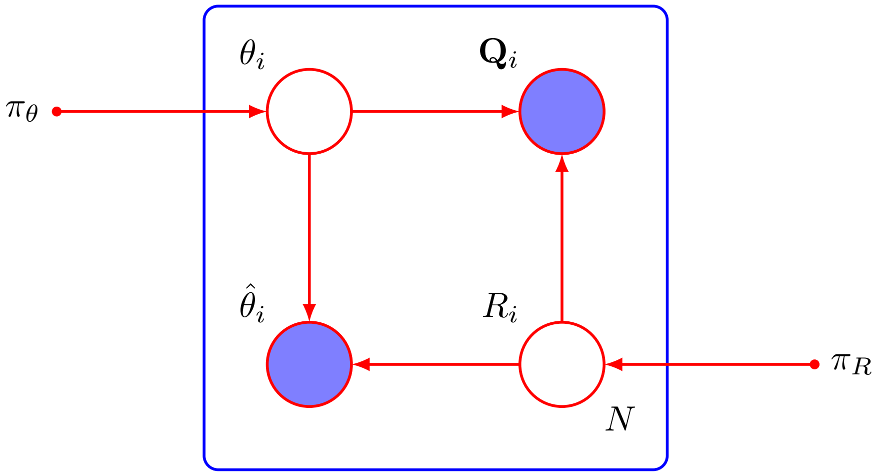


Figure 5.3: BN for reliability estimation based only on quality measures.

In Section 5.3, we show results comparing both BN configurations. The joint probability distribution of the BN with $\mathbf{Q}$ conditionally independent from θ is simplified as

$$P\left(s, \mathbf{Q}, R, \theta, \hat{\theta}|\pi_\theta, \pi_R\right) = P\left(s|R,\theta\right) P\left(\mathbf{Q}|R\right) P\left(\hat{\theta}|\theta, R\right) P\left(\theta|\pi_\theta\right) P\left(R|\pi_R\right) \tag{5.7}$$

where

$$P\left(\mathbf{Q}_i|\theta_i = \theta, R_i = R\right) = \sum_{k=1}^{K} w_k \mathcal{N}\left(\mathbf{Q}_i|\mu_{\mathbf{Q}_{R_k}}, \mathbf{\Lambda}^{-1}_{\mathbf{Q}_{R_k}}\right) \ . \tag{5.8}$$

We just need to plug (5.7) into (5.6) to compute the reliability.

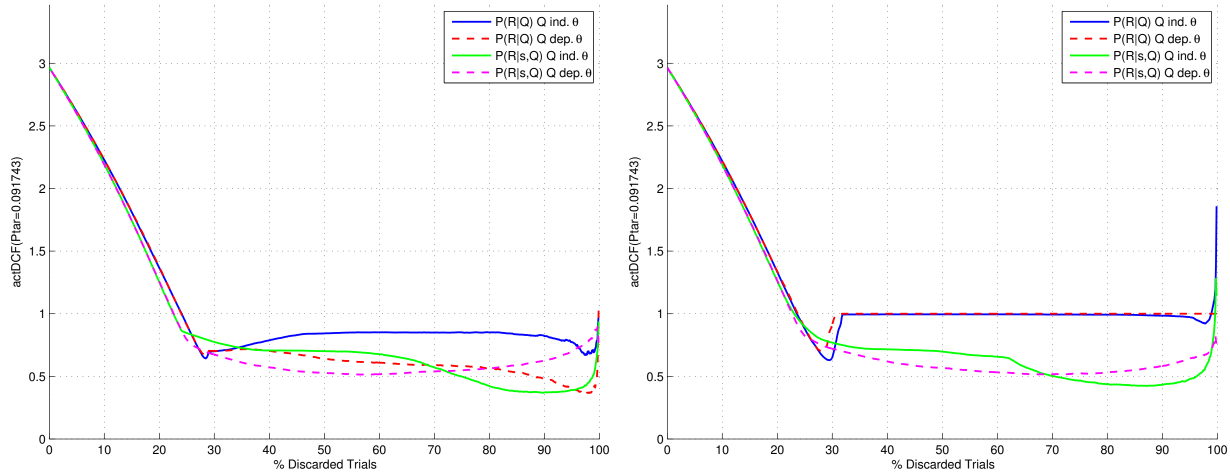
Previous models estimate the reliability given the score and quality measures. However, we also would like to assess the ability of the quality measures to estimate the reliability on their own without using the score. For that, we removed the score s from the network as shown in Figure 5.3. The joint distribution of this network is the same as in (5.5) and (5.7) removing the term $P\left(s|R,\theta\right)$.

5.3 Experiments

5.3.1 Experiments on NIST SRE with noise added

For the experiments in this section, we used data from NIST SRE with noise added as described in Section 3.4.2.1 of Chapter 3. This dataset has two trial lists: NIST SRE08 to train the reliability detection model and NIST SRE10 for evaluation. SV scores were computed used with system described in Section 3.4.1. If we pool all the noisy and clean trials of NIST SRE10 (all possible combinations of enrollment and test SNR), we obtain minimum DCF=0.99 and actual DCF=2.96. Our goal was to discard the unreliable trials in order to make the actual DCF lower than 1.0 while keeping fix the threshold that we set based on clean trials.

We start comparing different configurations of the Bayesian networks. Figure 5.4 compares the case where the reliability posterior is computed given the SV score and the



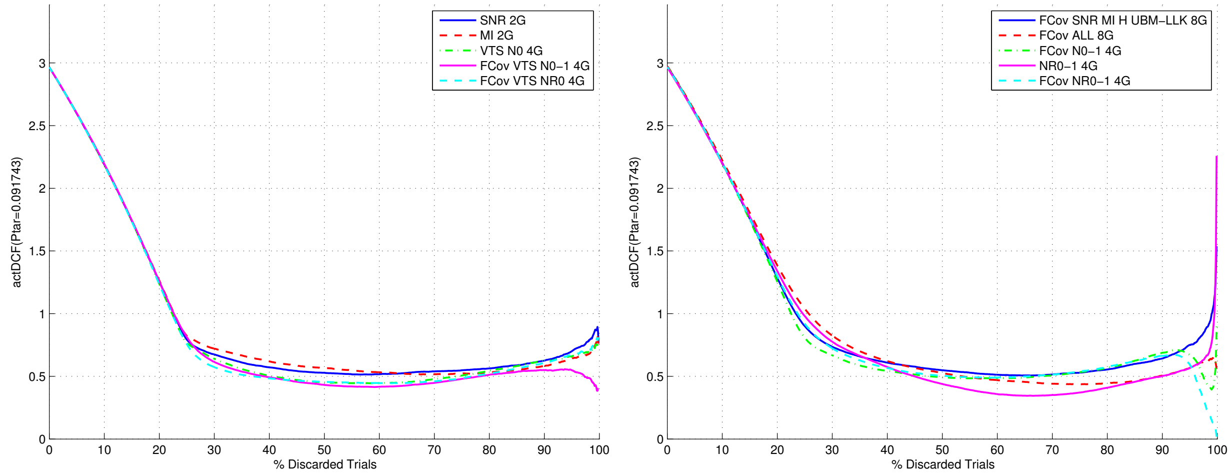
(a) Signal-to-noise ratio.

(b) Modulation index.

Figure 5.4: % Discarded trials vs. actDCF for different BN configurations.

quality measures, with the case where it is computed given the quality measures only. It also compares the case where the measures conditionally dependent of the label θ against the case they are independent. The figure plots the actual DCF against the percentage of discarded trials. The actual DCF was computed as described in Section 3.3.2, that is, we computed the DCF by taking into account only the trials classified as reliable. To plot these curves we put a varying threshold on the reliability posterior given by the network and, for each threshold, we computed the DCF and the percentage of discarded trials. The lower and steepy the curves are, the better. That means that we remove the worst trials first. When the curves stop descending and rise again, it means that the reliability detector is starting to discard trials that were correctly classified.

For this comparison we used networks based on two quality measures: signal-to-noise ratio and modulation index. We observe that all the networks behave similarly while



(a) Single feature.

(b) Multiple features.

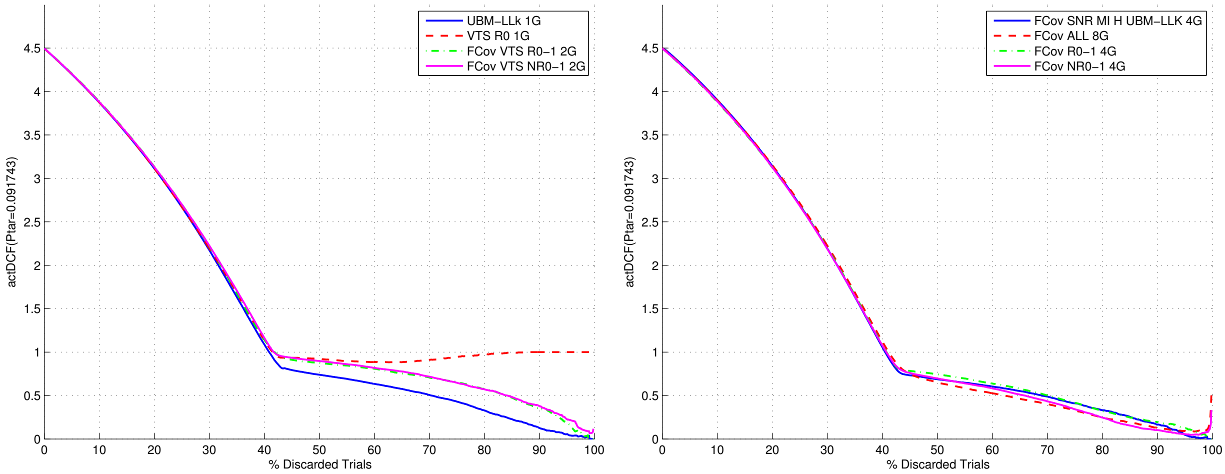
Figure 5.5: % Discarded trials vs. actDCF for NIST SRE10 + noise using several quality measures.

Table 5.1: % Discarded trials vs. actDCF for NIST SRE10 + noise

Q	K	% Discarded Trials 25	 50	 75
SNR	2	0.79	0.53	0.55
MI	2	0.82	0.57	0.52
VTS N_0	4	**0.76**	0.45	0.51
FCov VTS N_{0-1}	4	0.81	**0.43**	**0.48**
FCov VTS NR_0	4	**0.76**	0.46	**0.48**
FCov SNR MI H UBM-LLk	8	0.90	0.55	0.53
FCov All	8	1.04	0.53	0.44
FCov N_{0-1}	4	0.81	0.49	0.53
NR_{0-1}	4	0.98	**0.44**	**0.37**
FCov NR_{0-1}	4	0.92	0.50	0.54

removing the first 25% of trials. Those trials correspond to the ones with the lowest SNR pairs. In this point, the DCF dramatically improves from 2.96 to 0.80. From there, if we continue removing trials we observe differences between the curves. In general, the networks only based on quality measures perform worse than the networks based on score and measures. Only for the SNR and discarding more than 80% of the trials, the network with only measures is better. Comparing the curves from BN with measures conditionally dependent and independent of θ we find that, most of the time, the former are under the latter. However, if we were willing to discard more than 75% of the trials, we could achieve lower DCF with the network with SV scores and measures independent of θ. In most applications, we probably do not want to discard so many trials so we prefer the model with quality measures dependent on θ. From now on, we will show results only with that model.

Figure 5.5 and Table 5.1 compare the costs that we obtained by putting into the networks several quality measures from those described in Chapter 4. We divided the figure and the table into two parts: one where each BN only uses one quality measure (5.5a) and another where the BN combines several quality measures (5.5b). We tried more measures than those shown here. However for the sake of clarity, we only present the ones that yielded better performance. Following, we sum up the notation used for the measures. Signal-to-noise ratio is denoted by SNR, modulation index by MI, entropy by H and UBM log-likelihood by UBM-LLk. The term VTS N refers to the VTS coefficients with LDA trained to discriminate between different levels of SNR and, VTS R is the same but with LDA trained to discriminate between reverberation times. The numbers behind N or R indicate the output dimension of the LDA projection, for example N_{0-1} means that we keep the first two coefficients. When we write VTS NR, we mean that we concatenate the both LDA projections, the one for noise and the one for reverberation. When we say that we used *All* the measures we mean: SNR, MI, H, UBM-LLk, VTS NR_0, jitter and shimmer. In the same way as for the VTS parameters, the terms N, R and NR refer to LDA projections of *All* the quality measures to discriminate noise or/and reverberation times. We denote by *FCov* the networks that used full-covariance matrices for the mixtures $P\left(\mathbf{Q}|R,\theta\right)$, otherwise diagonal covariances were used. The table also shows the best number of components K of the mixtures for each quality measure. The performance of all the represented measures was



(a) Single feature.

(b) Multiple features.

Figure 5.6: % Discarded trials vs. actDCF for NIST SRE10 + reverberation using several quality measures.

quite similar. The best results were for the VTS based measures and *All* LDA projected.

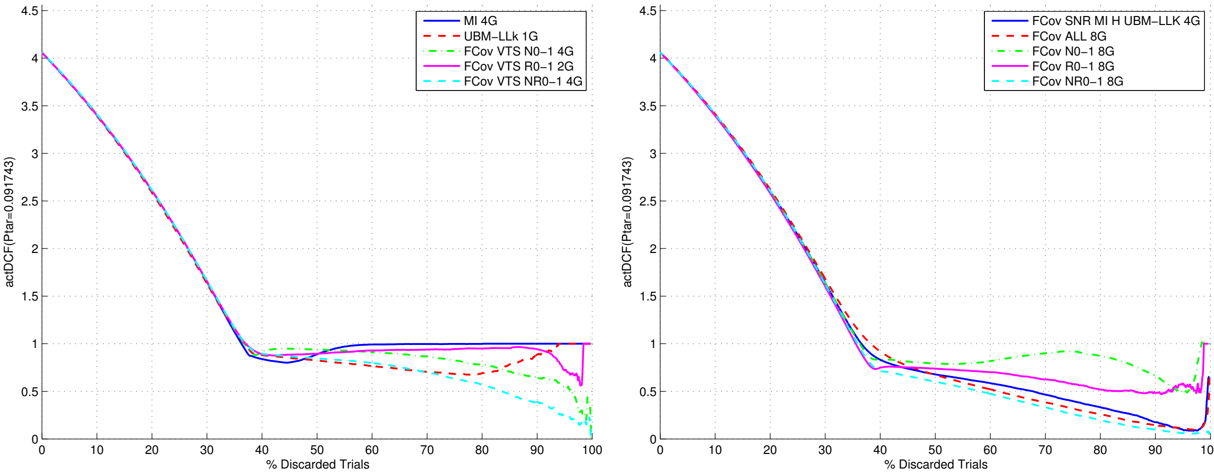
5.3.2 Experiments on NIST SRE with reverberation added

We also experimented on NIST SRE with reverberation added as described in Section 3.4.2.2. As in the previous section, trials created from NIST SRE08 were used to train the network and NIST SRE10 for evaluation. If we pool the NIST SRE10 trials with all possible reverberant conditions (all possible enrollment reverberation times against all the test reverberation times) our SV system obtains minimum DCF=0.99 and actual DCF=4.5.

Figure 5.6 and Table 5.2 display costs against percentage of trials discarded for BN using different measures. Subfigure 5.6a refers to networks using only one quality measure and Subfigure 5.6b to networks combining several measures. We only plot curves for the measures of Chapter 4 attaining lower costs. For the first 40% of discarded trials all measures performed the same, from there, we observe differences. The best results were achieved by

Table 5.2: % Discarded trials vs. actDCF for NIST SRE10 + reverberation

Q	K	% Discarded Trials		
		25	50	75
UBM-LLk	1	**2.67**	**0.74**	**0.43**
VTS R_0	1	**2.67**	0.92	0.94
FCov VTS R_{0-1}	2	2.70	0.87	0.65
FCov VTS NR_{0-1}	2	2.70	0.90	0.65
FCov SNR MI H UBM-LLk	4	2.69	0.69	0.41
FCov All	8	2.71	**0.65**	**0.33**
FCov R_{0-1}	4	2.69	0.74	0.42
FCov NR_{0-1}	4	**2.68**	0.70	0.34



(a) Single feature.

(b) Multiple features.

Figure 5.7: % Discarded trials vs. actDCF for NIST SRE10 + noise and reverberation using several quality measures.

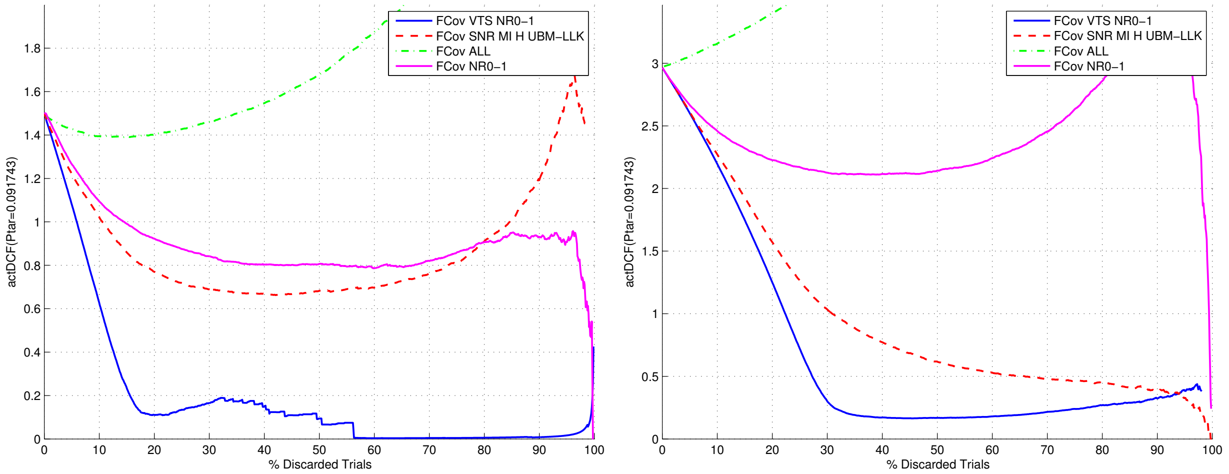
combining *All* the measures and with their LDA projection NR_{0-1}. Both curves are very close. After that, we find the UBM-LLk, the combination of SNR, MI, H and UBM-LLk; and the LDA projection R_{0-1}. Further, we have the LDA projections from the VTS parameters VTS R_{0-1} and VTS NR_{0-1}. The measure VTS R_0 performed badly, we needed to keep two dimensions of the projection to obtain good performance. The rest of measures tried but not shown here yielded worse results.

5.3.3 Experiments on NIST SRE with noise and reverberation added

Here, we pool the noisy and reverberant NIST SRE trial lists. We trained the BN with pooled NIST SRE08 and test on NIST SRE10. With this experiment, we intended to find

Table 5.3: % Discarded trials vs. actDCF for NIST SRE10 + noise and reverberation

Q	K	% Discarded Trials		
		25	50	75
MI	4	2.15	0.89	1.00
UBM-LLk	1	**2.13**	**0.82**	0.68
FCov VTS N_{0-1}	4	2.15	0.94	0.83
FCov VTS R_{0-1}	2	2.15	0.89	0.95
FCov VTS NR_{0-1}	4	2.15	0.85	**0.63**
FCov SNR MI H UBM-LLk	4	2.14	0.68	0.40
FCov All	8	2.17	0.67	0.32
FCov N_{0-1}	8	2.14	0.79	0.92
FCov R_{0-1}	8	**2.11**	0.74	0.58
FCov NR_{0-1}	8	2.14	**0.60**	**0.26**



(a) Agnitio Benchmark.

(b) Ahumada.

Figure 5.8: % Discarded trials vs. actDCF for Agnitio Benchmark and Ahumada datasets.

out if the network can cope with two types of degradations at the same time.

Figure 5.7 and Table 5.3 display costs against percentage of discarded trials for BN using different measures. Subfigure 5.7a refers to networks using only one quality measure and Subfigure 5.7b to networks combining several measures. Approximately, while discarding less than 35% of the trials all the BN behaved similarly. After that, we found that the best results were obtained by combining several measures. The best one was the projection NR_{0-1} from *All* the measures. It was followed by the concatenation of *All* the measures without LDA; and after that, the combination of SNR, MI, H and UBM-LLk. The best single feature was the LDA projection from VTS parameters VTS NR_{0-1}. The rest of measures shown were clearly worse. Results prove that we can use this type of networks to deal with different types of degradation.

5.3.4 Experiments on databases with real distortions

Finally, we experimented on databases with real distortions. We used the datasets Agnitio benchmark and Ahumada described in Sections 3.4.2.4 and 3.4.2.5. Getting access to large datasets with real distortions (noise, reverberation, etc) to train the reliability models is difficult. The aim of this experiments was to find out if we can use a BN trained on an artificially degraded dataset to detect unreliable trials on real databases. Thus, we trained the BN on NIST SRE08 with noise and reverberation added and we evaluated the model

Table 5.4: % Discarded trials vs. actDCF for Agnitio Benchmark and Ahumada.

(a) *Agnitio Benchmark.*

Q	% Discarded Trials		
	25	50	75
FCov VTS NR_{0-1}	**0.13**	**0.09**	**0.00**
FCov SNR MI H UBM-LLk	0.72	0.68	0.81
FCov All	1.43	1.68	2.35
FCov NR_{0-1}	0.87	0.81	0.86

(b) Ahumada.

Q	% Discarded Trials		
	25	50	75
FCov VTS NR_{0-1}	**0.72**	**0.17**	**0.24**
FCov SNR MI H UBM-LLk	1.27	0.62	0.46
FCov All	3.53	4.66	6.95
FCov NR_{0-1}	2.17	2.14	2.63

on Agnitio benchmark and Ahumada.

Figure 5.8a and Table 5.4a shows actual costs against percentage of discarded trials for Agnitio Benchmark. Figure 5.8b and Table 5.4b shows costs for Ahumada. In the previous sections, we only showed results with the quality measures that obtained the lowest costs on the evaluation set. Here, instead of that we show results with the measures that obtained the best results on NIST SRE10 + noise and reverberation. In most real scenarios, we are not going to count with a labeled dataset that allows us to choose the best measure. Thus, we think that it is more realistic to use the artificial dataset to choose the quality measure that we are going to use in our BN and to tune hyper-parameters like the optimum number of components of the GMM, etc. We discovered that some of the measures that performed well on NIST SRE10 performed poorly on these databases. The VTS NR_{0-1} is the only one that provides low costs while discarding a low number of trials (20–30%). The combination of SNR, MI, H, and UBM-LLk also performed quite well on Ahumada but not so well on Agnitio Benchmark where the lowest cost that it reached was 0.65. The combination of all the measures performed very bad making the cost to grow rapidly.

5.4 Summary

In this chapter, we revisited the work in [Richiardi et al., 2006a] about reliability estimation with Bayesian networks. We gave the mathematical foundations of the method and experimented on artificial and real databases. We trained our model on NIST SRE08 with noise and reverberation added and evaluated it on NIST SRE10 also with noise and reverberation, Agnitio benchmark and Ahumada. BN performance was measured by comparing the reduction of actual cost as we discard trials classified as unreliable.

We tried several configurations of the BN by altering the dependencies of the conditional distributions that define the network. In general, the best results were achieved by computing the reliability posterior based on the speaker verification score and the quality measures where the distribution of quality measures was assumed conditionally dependent on the trial label. Hence, we deduced that signal quality degradations affects differently to target and non-target trials. That means that the amount of noise needed to make a target trial unreliable is different than for a non-target.

We experimented with the quality measures described in Chapter 4 one by one and with several combinations of measures. We showed results with the ones that performed better. Most of those measures had not been evaluated before as inputs of this type of Bayesian network. Signal-to-noise ratio and modulation index were good measures for the dataset contaminated with noise and the UBM log-likelihood for the dataset with reverberation. The measures based on LDA dimensionality reduction of the VTS parameters (VTS NR_{0-1}) also performed well for both types of degradation. The BNs that combined multiple features also attained good performance. The combination of *All* the measures and its LDA dimensionality reduction (NR_{0-1}) were the best ones.

We evaluated the generalization capability of the method by training the BN on a synthetic dataset and evaluating in real datasets like Agnitio benchmark and Ahumada. Results evidenced that the quality measures that performed best on the synthetic dataset performed poorly or even made the cost grow over the original cost. Only the measure based the VTS parameters (VTS NR_{0-1}) performed well.

Chapter 6

Reliability Estimation by Modeling the Variability of the Speaker Verification Score in Adverse Environments

6.1 Introduction

In this chapter, we continue working on methods to estimate the reliability of the speaker verification decisions from a set of quality measures. The model that we considered in the previous chapter has several drawbacks, as we pointed out. Its main defect is that the trials considered reliable and unreliable change depending on the operating point of the speaker verification system. The operating point of the system is defined by the prior probability of finding a target trial. In practice, this affects to the selection of the SV threshold, a low prior implies a high threshold and vice versa. Thus, if we move the operating point the SV decisions change and, consequently, the trial reliability. In that event, we need to re-train our model.

Intending to overcome those problems, we propose to model SV scores and quality measures with a novel Bayesian network configured in a different manner. This BN does not, explicitly, contain a variable to indicate if the trial is reliable or not. Instead, we added a hidden variable meaning the score that the SV system would produce if the trial were not affected by any degradation. We called this variable *clean score*. Our BN models how the SV score deviates from the *clean score* for different types of distortions. The type of distortion of the was another hidden variable, which we called the *quality state*. Given the SV score and the quality measures, the BN allows to infer a posterior distribution for the hidden score and from it, we can decide about the trial reliability.

This chapter is organized as follows. Section 6.2 introduces our new Bayesian network that models variations of the score distributions on distorted trials. The network relates all the variables involved in the trial, i.e., the clean and noisy scores, the quality state, the quality measures and the trial label. For a given value of the quality state, we will observe a different distribution of quality measures and a different variation of the noisy score with respect to the clean score. This variation will also depend on whether the trial is target or non-target. Section 6.3 defines the concept of reliability in this new context. The reliability posterior is the probability for the SV score to be over or under the threshold–depending on whether the trial was classified as target or non-target–, which is computed by integrating the posterior of the clean score. Section 6.4 explains how to compute the posterior distribution of the hidden score. Depending on which variables are hidden or observed, we distinguish several cases. The general case corresponds to a common trial, where both trial label and the type of distortions are unknown. Even in this case the posterior can be expressed in closed form. Section 6.5 shows how to employ our Bayesian network to compute an improved speaker verification likelihood ratio that intends to be a better approximation to the ideal score. This ratio is based on the target posterior given the observed score and the quality measures. It does not depend on the clean score or the quality state that are effectively integrated out. In Section 6.6, we present expectation maximization algorithms to train the parameters of the network. We considered three flavors depending on which variables are observed during the training phase. In the general case, the clean score and the distortion level are hidden. However, for databases with artificial distortions where we have clean and distorted versions of the same trial, we can assume the clean score as known. Furthermore, in and artificial dataset we probably know the type of distortion. Then, all variables are observed and training the network is trivial. In Section 6.7, we show the results of our experiments. We experimented on NIST SRE with noise and reverberation, Agnitio benchmark and Ahumada. We did two types of experiments: reliability detection and LLR improvement. The reliability detection consisted in rejecting unreliable trials so that the rest of trials provide low error rates. We used the best results in the previous chapter as baseline. We will see that, in most cases, the new network outperforms the baseline. Again, VTS parameters, modulation index and UBM log-likelihood were good quality measures. Besides, the new network generalizes better for the non NIST databases. Regarding the LLR improvement experiments, we computed error rates on the improved LLR without rejecting trials. We will see that it dramatically improved actual costs. Finally, Section 6.8 summarizes the conclusions of the chapter.

6.2 Bayesian Network to Model the Score Variability in Adverse Environments

The speaker verification score is usually a log-likelihood ratio between the probabilities of the trial data given the target and non-target hypothesis. As such, it gives an idea of how much the enrollment and test segments are alike. If the score is very high, it provides a large confidence that both segments have been uttered by the same person; and vice versa, if it is very low, we can be quite sure that both segments belong to different speakers. When the enrollment or test segments are recorded in environments that degrade the quality of the speech signal the score also degrades and loses its capacity to discriminate between target and non-targets.

We thought that counting with a method to model how the SV scores mutate when the speech is affected by different types of distortions would be very valuable. One of the applications of such model would be to recover a non-degraded version of the speaker verification score. For that purpose, we defined the Bayesian network illustrated by the graphical model in Figure 6.1. We remember that empty nodes denote *hidden variables*, shaded nodes denote *observed variables* and small solid nodes denote *deterministic parameters*. The *plate* surrounding the nodes indicates that we consider N trials.

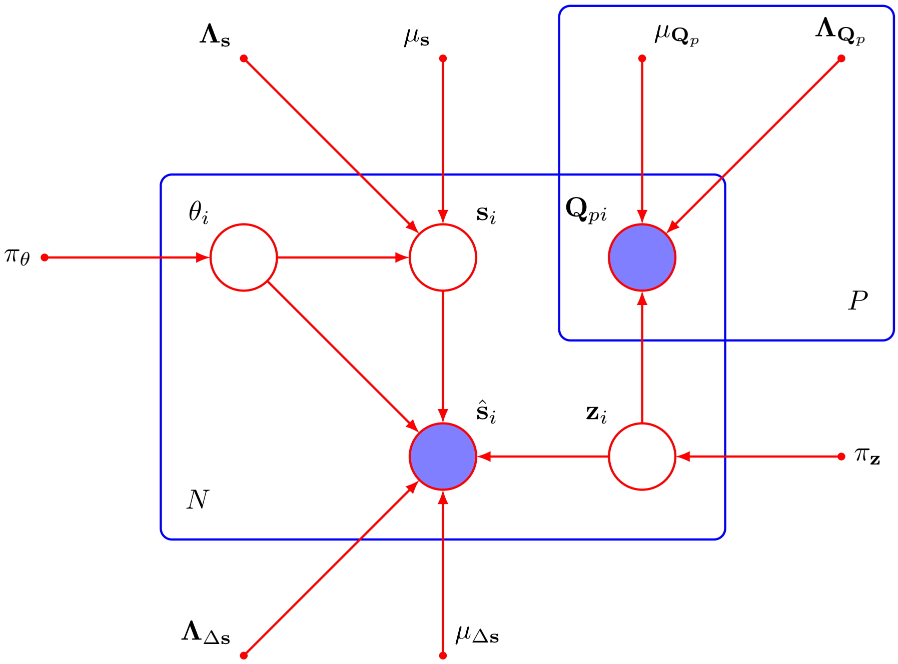


Figure 6.1: BN to model SV score variations in adverse environments.

Following, we explain the variables included in the graph. For each trial i, we have the corresponding score $\hat{\mathbf{s}}_i$ provided by the speaker verification system. We will refer to this score as *observed score* or *noisy score*.

The trial label $\theta_i \in \{\mathcal{T}, \mathcal{N}\}$ where $\mathcal{T}$ is the hypothesis that the enrollment and the test segments belong to the same speaker; and $\mathcal{N}$ the hypothesis that they belong to different speakers. It is observed in the training phase and hidden in the test phase. $\pi_\theta = (P_\mathcal{T}, P_\mathcal{N})$ is the hypothesis prior where $P_\mathcal{T}$ is the target prior and $P_\mathcal{N} = 1 - P_\mathcal{T}$ is the non-target prior.

We denote by $\mathbf{s}_i$ a hypothetical score that we would obtain if the trial were not affected by any source of degradation. In the general case, $\mathbf{s}_i$ is hidden. However, if we train the BN with a high quality database that we have degraded artificially (adding noise or reverberation), we can know which clean trial corresponds to each degraded trial. In this case $\mathbf{s}_i$ could be taken as observed and that makes the estimation of the parameters of the network simpler. We will refer to this score as *hidden score* or *clean score*. We put a Gaussian prior on $\mathbf{s}_i$ conditioned on the label θ_i:

$$P\left(\mathbf{s}_i|\theta_i = \theta\right) = \mathcal{N}\left(\mathbf{s}_i|\mu_{\mathbf{s}_\theta}, \mathbf{\Lambda}_{\mathbf{s}_\theta}^{-1}\right) \ . \tag{6.1}$$

There are two Gaussian distributions, one for the target hypothesis and another for the non-target.

The relation between $\mathbf{s}_i$ and $\hat{\mathbf{s}}_i$ was assumed linear:

$$\hat{\mathbf{s}}_i = \mathbf{s}_i + \Delta\mathbf{s}_i \tag{6.2}$$

where $\Delta\mathbf{s}_i$ is described by a Gaussian conditional distribution as defined below.

The variable $\mathbf{z}_i$ is called the *quality state*. It is a 1-of-K binary vector with elements z_{ik} for $k = 1, \ldots, K$. It represents the different types and/or levels of degradation that the trial could suffer. For example, it could correspond to signal-to-noise ratio levels. If we supposed that our recordings can have 6 different SNR of enrollment and test, the variable $\mathbf{z}_i$ could take 36 different values. We can write the prior distribution of $\mathbf{z}$ as

$$P(\mathbf{z}_i) = \prod_{k=1}^{K} \pi_{z_k}^{z_{ik}} \tag{6.3}$$

where the weights $\pi_{\mathbf{z}}$ represent the prior probabilities for each degradation type.

The distribution of $\Delta\mathbf{s}$ is conditioned on $\mathbf{z}_i$ and θ_i so

$$P(\hat{\mathbf{s}}_i|\mathbf{s}_i, z_{ik}=1, \theta_i=\theta) = \mathcal{N}\left(\hat{\mathbf{s}}_i|\mathbf{s}_i + \mu_{\Delta\mathbf{s}_{k\theta}}, \mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta}}^{-1}\right) \ . \tag{6.4}$$

There are $2 \times K$ different distributions, one for each pair of values of $\mathbf{z}_i$ and θ_i.

The observed quality measures are denoted by $\mathbf{Q}_{ip}$ with $p = 1, \ldots, P$. We have assumed a very general model where the subscript p divides the quality measures into groups of measures that are supposed to be independent between them given $\mathbf{z}_i$. In this way, we can force independence between variables that we think that should not be correlated. We define the set $\mathbf{Q}_i = \{\mathbf{Q}_{ip}\}_{p=1}^{P}$. Then, the distribution of $\mathbf{Q}_i$ conditioned on the quality state $\mathbf{z}_k$ is a block diagonal Gaussian:

$$P(\mathbf{Q}_i|z_{ik}=1) = \prod_{p=1}^{P} \mathcal{N}\left(\mathbf{Q}_{ip}|\mu_{\mathbf{Q}_{pk}}, \mathbf{\Lambda}_{\mathbf{Q}_{pk}}^{-1}\right) \ . \tag{6.5}$$

Finally, we denote by $\mathcal{M}$ the set of all the model parameters, $\mathcal{M} = \left(\mu_{\mathbf{s}}, \mathbf{\Lambda}_{\mathbf{s}}, \mu_{\Delta\mathbf{s}}, \mathbf{\Lambda}_{\Delta\mathbf{s}}, \mu_{\mathbf{Q}_p}, \mathbf{\Lambda}_{\mathbf{Q}_p}, \pi_{\mathbf{z}}\right)$.

Even though, the scores $\mathbf{s}_i$ and $\hat{\mathbf{s}}_i$ are unidimensional, we denoted them in boldface, as vectors, because this model could be generalized to the case where we have several speaker verification systems. In that case, they would be vectors and we could do score fusion with the help of the network. However, in this thesis, all the experiments were performed with only one speaker verification system.

6.3 Reliability Definition

In the BN of the previous chapter, the trial reliability was a variable that was part of the Bayesian network. In the new BN, we exchange the reliability for the hidden score. Now, we are going to define the reliability concept in this new context.

In a traditional speaker verification system, the decision $\hat{\theta}$ is taken by thresholding the SV score $\hat{s}$ as

$$\hat{\theta} = \begin{cases} \mathcal{T} & \text{if } \hat{s} \geq \xi_\theta \\ \mathcal{N} & \text{if } \hat{s} < \xi_\theta \end{cases} \tag{6.6}$$

where ξ_θ is the decision threshold. From now on, we will omit the subscript i in the variables when the equations refer to only one trial.

To know if that decision is reliable, first, we used the Bayesian network to compute the posterior distribution of the clean score s given the observed score and the quality measures $P(s|\hat{s}, \mathbf{Q})$ (We omit the rest of dependencies in the distribution to keep the notation uncluttered). Then, we defined the probability of reliable decision as

$$P(R = \mathcal{R}|\hat{s}, \mathbf{Q}) = \begin{cases} P(s > \xi_\theta|\hat{s}, \mathbf{Q}) & \text{if } \hat{\theta} = \mathcal{T} \\ P(s < \xi_\theta|\hat{s}, \mathbf{Q}) & \text{if } \hat{\theta} = \mathcal{N} \end{cases} \tag{6.7}$$

where

$$P\left(s > \xi_\theta | \hat{s}, \mathbf{Q}\right) = \int_{\xi_\theta}^{\infty} P\left(s|\hat{s}, \mathbf{Q}\right)\, \mathrm{d}s \tag{6.8}$$

$$P\left(s < \xi_\theta | \hat{s}, \mathbf{Q}\right) = \int_{-\infty}^{\xi_\theta} P\left(s|\hat{s}, \mathbf{Q}\right)\, \mathrm{d}s\ . \tag{6.9}$$

This definition of reliability is based on checking the coherence between the decision taken from the observed score and the decision that we would take from the posterior of the hidden score.

Finally, we decided if the trial is reliable $\mathcal{R}$ or unreliable $\mathcal{U}$ by thresholding the reliability posterior

$$\hat{R} = \begin{cases} \mathcal{R} & \text{if } P\left(R = \mathcal{R}|\hat{s}, \mathbf{Q}\right) \geq \xi_R \\ \mathcal{U} & \text{if } P\left(R = \mathcal{R}|\hat{s}, \mathbf{Q}\right) < \xi_R \end{cases} \tag{6.10}$$

where ξ_R is the reliability threshold.

6.4 Posterior Distribution of the Hidden Score

6.4.1 General case

According to (6.7), to compute the reliability posterior we need the posterior of the hidden score. In the general case, with θ and $\mathbf{z}$ hidden, the posterior of $\mathbf{s}$ is a mixture of Gaussians:

$$P\left(\mathbf{s}|\hat{\mathbf{s}}, \mathbf{Q}\right) = \sum_{\theta \in \{\mathcal{T}, \mathcal{N}\}} \sum_{k=1}^{K} P\left(\theta, z_k = 1|\hat{\mathbf{s}}, \mathbf{Q}\right) \mathcal{N}\left(\mathbf{s}|\mu'_{\mathbf{s}_{k\theta}}, \mathbf{\Lambda}'^{-1}_{\mathbf{s}_{k\theta}}\right) \tag{6.11}$$

where

$$\mathbf{\Lambda}'_{\mathbf{s}_{k\theta}} = \mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta}} + \mathbf{\Lambda}_{\mathbf{s}_\theta} \tag{6.12}$$

$$\mu'_{\mathbf{s}_{k\theta}} = \mathbf{\Lambda}'^{-1}_{\mathbf{s}_{k\theta}} \left(\mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta}} \left(\hat{\mathbf{s}} - \mu_{\Delta\mathbf{s}_{k\theta}}\right) + \mathbf{\Lambda}_{\mathbf{s}_\theta} \mu_{\mathbf{s}_\theta}\right)\ . \tag{6.13}$$

The weights of the mixture are given by

$$P\left(\theta, z_k = 1|\hat{\mathbf{s}}, \mathbf{Q}\right) = \frac{P\left(\hat{\mathbf{s}}|\theta, z_k = 1\right) P\left(\mathbf{Q}|z_k = 1\right) P\left(\theta\right) \pi_{z_k}}{\sum_{\theta \in \{\mathcal{T}, \mathcal{N}\}} \sum_{k=1}^{K} P\left(\hat{\mathbf{s}}|\theta, z_k = 1\right) P\left(\mathbf{Q}|z_k = 1\right) P\left(\theta\right) \pi_{z_k}} \tag{6.14}$$

where we need to evaluate the distribution of the observed score given the pair $(\theta,\, z_k = 1)$. It can be proven that is Gaussian:

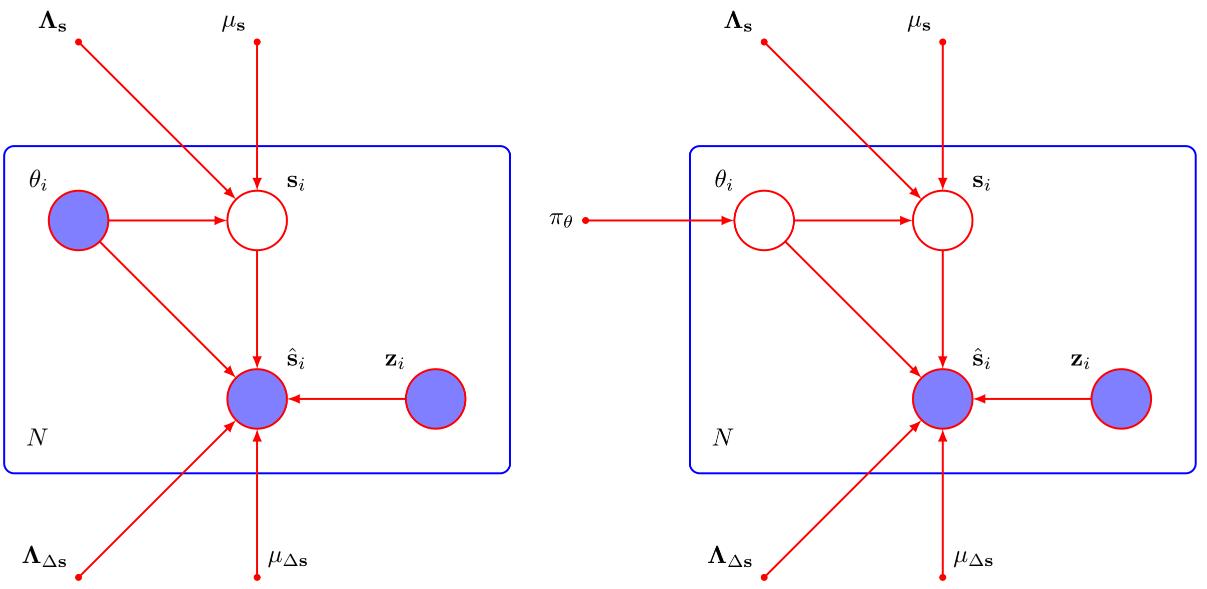
$$P\left(\hat{\mathbf{s}}|\theta, z_k = 1\right) = \mathcal{N}\left(\hat{\mathbf{s}}|\mu'_{\hat{\mathbf{s}}_{k\theta}}, \mathbf{\Lambda}'^{-1}_{\hat{\mathbf{s}}_{k\theta}}\right) \tag{6.15}$$

with

$$\mathbf{\Lambda}'_{\hat{\mathbf{s}}_{k\theta}} = \mathbf{\Lambda}_{\mathbf{s}_\theta} \mathbf{\Lambda}'^{-1}_{\mathbf{s}_{k\theta}} \mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta}} = \left(\mathbf{\Lambda}^{-1}_{\mathbf{s}_\theta} + \mathbf{\Lambda}^{-1}_{\Delta\mathbf{s}_{k\theta}}\right)^{-1} \tag{6.16}$$

$$\mu'_{\hat{\mathbf{s}}_{k\theta}} = \mu_{\mathbf{s}_\theta} + \mu_{\Delta\mathbf{s}_{k\theta}}\ . \tag{6.17}$$

Derivations for these equations can be found in Appendix B.



(a) θ and $\mathbf{z}$ observed.

(b) θ hidden and $\mathbf{z}$ observed.

Figure 6.2: Particular cases of BN to model SV score variations.

Then, we obtained the probability of reliable trial with (6.11), (6.7) and the formula for integral of the univariate Gaussian:

$$P\left(R=\mathcal{R}|\hat{s},\mathbf{Q}\right)=\begin{cases}\frac{1}{2}-\frac{1}{2}\sum_{\theta\in\{\mathcal{T},\mathcal{N}\}}\sum_{k=1}^{K}P\left(\theta,z_k=1|\hat{\mathbf{s}},\mathbf{Q}\right)\operatorname{erf}\left(\sqrt{\frac{\mathbf{\Lambda}'_{\mathbf{s}_{k\theta}}}{2}}\left(\xi_\theta-\mu'_{\mathbf{s}_{k\theta}}\right)\right) \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\text{if } \hat{\theta}=\mathcal{T} \\ \frac{1}{2}+\frac{1}{2}\sum_{\theta\in\{\mathcal{T},\mathcal{N}\}}\sum_{k=1}^{K}P\left(\theta,z_k=1|\hat{\mathbf{s}},\mathbf{Q}\right)\operatorname{erf}\left(\sqrt{\frac{\mathbf{\Lambda}'_{\mathbf{s}_{k\theta}}}{2}}\left(\xi_\theta-\mu'_{\mathbf{s}_{k\theta}}\right)\right) \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\text{if } \hat{\theta}=\mathcal{N}\end{cases}\tag{6.18}$$

where erf is the Gaussian error function.

6.4.2 Case with θ and z observed

We could consider the unrealistic case where θ and $\mathbf{z}$ are observed. We say that it is unrealistic because θ_{t} is precisely what we want to find out in the test phase. This case is illustrated in the graphical model of Figure 6.2a. The clean score posterior in (6.11) simplifies to an unique Gaussian:

$$P\left(\mathbf{s}|\hat{\mathbf{s}},\theta,z_k=1\right)=\mathcal{N}\left(\mathbf{s}|\mu'_{\mathbf{s}_{k\theta}},\mathbf{\Lambda}'^{-1}_{\mathbf{s}_{k\theta}}\right)\tag{6.19}$$

and probability of reliable trial reduces to:

$$P\left(R=\mathcal{R}|\hat{s},\theta,z_k=1\right)=\begin{cases}\frac{1}{2}\left(1-\operatorname{erf}\left(\sqrt{\frac{\mathbf{\Lambda}'_{\mathbf{s}_{k\theta}}}{2}}\left(\xi_\theta-\mu'_{\mathbf{s}_{k\theta}}\right)\right)\right) & \text{if } \hat{\theta}=\mathcal{T} \\ \frac{1}{2}\left(1+\operatorname{erf}\left(\sqrt{\frac{\mathbf{\Lambda}'_{\mathbf{s}_{k\theta}}}{2}}\left(\xi_\theta-\mu'_{\mathbf{s}_{k\theta}}\right)\right)\right) & \text{if } \hat{\theta}=\mathcal{N}\end{cases}.\tag{6.20}$$

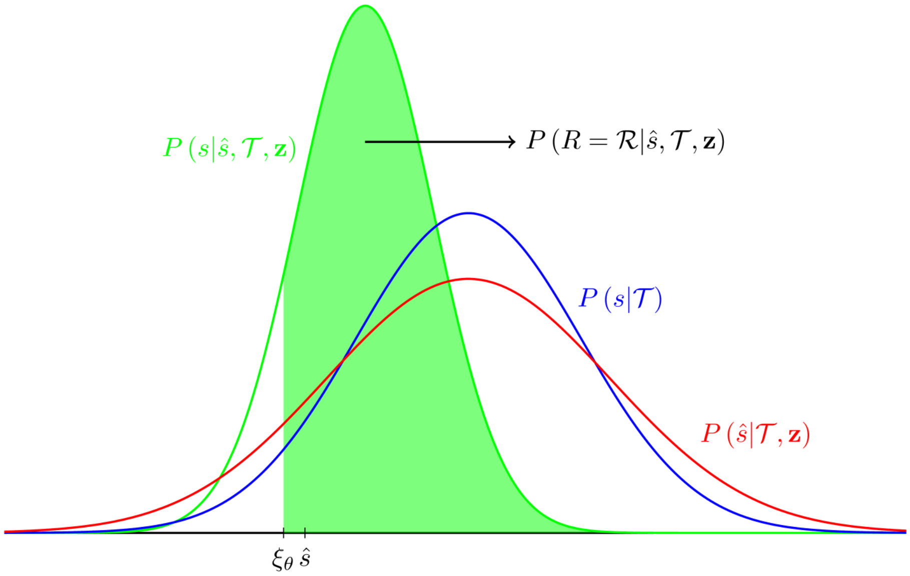


Figure 6.3: Score distributions involved in the calculus of $P(R=\mathcal{R}|\hat{s},\theta=\mathcal{T},z_k=1)$

Figure 6.3 shows an example of the score distributions involved in the calculus of $P(R=\mathcal{R}|\hat{s},\theta=\mathcal{T},z_k=1)$ with k corresponding to the pair of SNR of enrollment and test (10 dB, 10 dB). The blue curve is the prior distribution of the scores for clean target trials $P(s|\theta=\mathcal{T})$. The red curve is the distribution of observed scores for noisy trials $P(\hat{s}|\theta=\mathcal{T},z_k=1)$. The green curve is the posterior of the clean score $P(\mathbf{s}|\hat{\mathbf{s}},\theta=\mathcal{T},z_k=1)$ and the filled area under the curve represents the reliability posterior for the trial. Even though the observed score $\hat{s}$ is only slightly over the threshold ξ_θ most of the area under the posterior of s is filled so the trial has a high reliability. That happens because, as $\theta=\mathcal{T}$ is known, the posterior that we obtain reaffirms the target decision.

6.4.3 Case with θ hidden and $\mathbf{z}$ observed

Now, we consider a more realistic case where θ is hidden but $\mathbf{z}$ is observed. That is, we do not know if the trial is target or non-target but we know the type of degradation to which it is subject. This case is illustrated in the graphical model of Figure 6.2b Then, the posterior of s is a mixture of two Gaussians

$$P(\mathbf{s}|\hat{\mathbf{s}},z_k=1)=\sum_{\theta\in\{\mathcal{T},\mathcal{N}\}}P(\theta|\hat{\mathbf{s}},z_k=1)\,\mathcal{N}\left(\mathbf{s}|\mu'_{\mathbf{s}_{k\theta}},\mathbf{\Lambda}'^{-1}_{\mathbf{s}_{k\theta}}\right) \tag{6.21}$$

where the weights of the mixture are the posteriors of the labels θ given the observed score $\hat{s}$ and the quality state $\mathbf{z}$. They are computed as

$$P(\theta=\mathcal{T}|\hat{\mathbf{s}},z_k=1)=\frac{P(\hat{\mathbf{s}}|\mathcal{T},z_k=1)\,P_{\mathcal{T}}}{\sum_{\theta\in\{\mathcal{T},\mathcal{N}\}}P(\hat{\mathbf{s}}|\theta,z_k=1)\,P(\theta)} \tag{6.22}$$

$$=\frac{1}{1+\exp\left(-\ln R(\hat{\mathbf{s}},z_k=1)-\operatorname{logit}(P_{\mathcal{T}})\right)} \tag{6.23}$$

where

$$\ln R(\hat{\mathbf{s}},z_k=1)=\ln P(\hat{\mathbf{s}}|\mathcal{T},z_k=1)-\ln P(\hat{\mathbf{s}}|\mathcal{N},z_k=1) \tag{6.24}$$

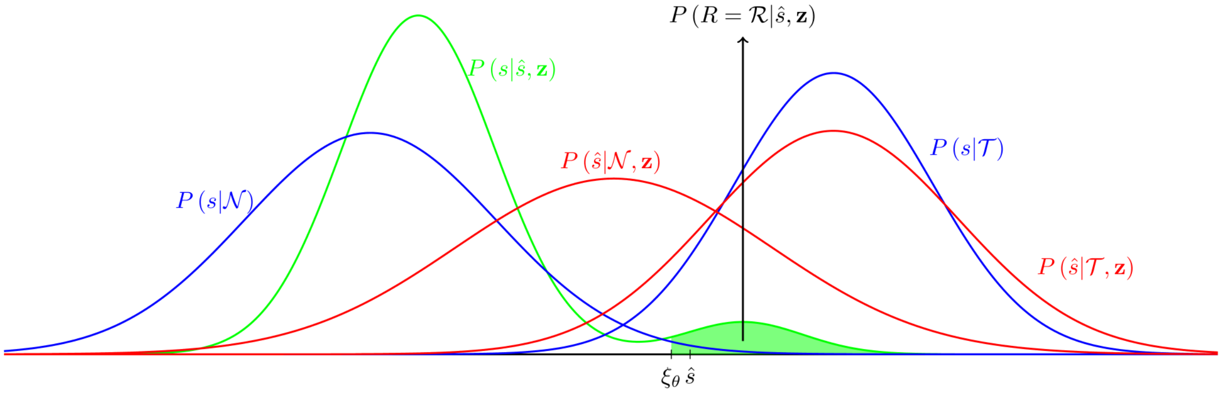


Figure 6.4: Score distributions involved in the calculus of $P(R = \mathcal{R}|\hat{s}, z_k = 1)$

and the distributions $P(\hat{\mathbf{s}}|\mathcal{N}, z_k = 1)$ are given by (6.15).

Figure 6.4 shows an example of the score distributions involved in the calculus of $P(R = \mathcal{R}|\hat{s}, z_k = 1)$ with k corresponding to the pair of SNR of enrollment and test (10dB, 10dB). The blue curves are the prior distributions for the scores of the clean target and non-target trials $P(s|\theta)$. The red curves are the distributions for the observed scores of noisy trials $P(\hat{s}|\theta, z_k = 1)$. The mean of the noisy targets distribution does not change much with respect to the one of clean targets but the noisy non-targets distribution is very shifted up. Both noisy distributions present larger variance than their clean counterparts. The green curve is the posterior distribution of the clean score $P(s|\hat{\mathbf{s}}, z_k = 1)$ and the filled area under the curve is the posterior reliability of the trial. As we saw above, the posterior of s is a mixture of two Gaussians. To compute the weights of the mixture we plugged-in the observed score $\hat{s}$ into (6.22). In the figure, we observe that the values of $P(\hat{s}|\mathcal{T}, \mathbf{z})$ and $P(\hat{s}|\mathcal{N}, \mathbf{z})$ are close but as the target prior $P_\mathcal{T} = 0.091$ (Effective prior used in NIST SRE before year 2010) was low, the Gaussian corresponding to the non-target hypothesis obtained a much larger weight. Thus, the resulting posterior tell us that it is very likely that the hidden score s is much lower than the observed score $\hat{s}$ so the reliability of the decision taken based on $\hat{s}$ is very low.

6.5 Quality Dependent Likelihood Ratio

We can apply this BN, not only to detect unreliable trials, but to obtain an improved SV likelihood ratio. Normally, the output of the SV system is a likelihood ratio and, from it, we can compute the target posterior. By inverting the formula, we can obtain the likelihood ratio from the target posterior given by the network:

$$\mathrm{LR}(\hat{\mathbf{s}}, Q) = \frac{P(\theta = \mathcal{T}|\hat{\mathbf{s}}, \mathbf{Q})}{1 - P(\theta = \mathcal{T}|\hat{\mathbf{s}}, \mathbf{Q})} \frac{1 - P_\mathcal{T}}{P_\mathcal{T}} \tag{6.25}$$

where

$$P(\theta = \mathcal{T}|\hat{\mathbf{s}}, \mathbf{Q}) = \sum_{k=1}^{K} P(\theta = \mathcal{T}, z_k = 1|\hat{\mathbf{s}}, \mathbf{Q}) \tag{6.26}$$

and the values of $P(\theta = \mathcal{T}, z_k = 1|\hat{\mathbf{s}}, \mathbf{Q})$ are given by (6.14). This ratio depends on the observed score and the quality measures but not on the hidden score or the quality states that are effectively integrated out. In applications where we need to classify all the trials, we can substitute the standard ratio given by the SV system by this new one. Given that this ratio takes into account an additional source of information like the quality measures, we should expect an improved performance.

6.6 Bayesian Network Training

The parameters of the Bayesian network need to be estimated from a development dataset that include a large amount of trials with different degradation types and/or levels. We assumed that the labels θ of the development set are known. We distinguished three cases according to which variables are hidden and observed: **s** and **z** observed; **s** observed and **z** hidden; and the general case where both variables are hidden.

6.6.1 Training with s and z observed

Let us consider a development set of trials artificially degraded from a high quality set. For example, we could create that dataset by adding noise or reverberation to the enrollment and/or test segments involved in each trial. In a case like this, we can track which clean trial is associated to each degraded trial. We assumed that a clean trial i has equal observed and hidden scores $s_i = \hat{s}_i$ and that any trial j obtained by degrading the trial i has a hidden score (*clean score*) $s_j = \hat{s}_i$. Thus, s is observed when training the network.

Besides, we can assign a different value of the quality state to each kind of distortion. For example, if we had a dataset with 6 different SNR levels for enrollment and another 6 levels for test the quality state **z** would be able to take 36 possible values, one by each SNR pair. As we know which distortion we applied to each trial, **z** is also observed.

Estimating the parameters of the network reduces to compute the means and variances $\left(\mu_{\mathbf{s}}, \mathbf{\Lambda}_{\mathbf{s}}, \mu_{\Delta\mathbf{s}}, \mathbf{\Lambda}_{\Delta\mathbf{s}}, \mu_{\mathbf{Q}_p}, \mathbf{\Lambda}_{\mathbf{Q}_p}\right)$ of the Gaussian conditional distributions $P(\mathbf{s}_i|\theta_i)$, $P(\hat{\mathbf{s}}_i|\mathbf{s}_i, \mathbf{z}_i, \theta_i)$, and $P(\mathbf{Q}_i|\mathbf{z}_i)$; and the weights $\pi_{\mathbf{z}}$ that indicate the priors of each distortion type. As in this case, all the distributions are conditioned on observed variables, computing those parameters by maximum likelihood is straightforward.

6.6.2 Training with s observed and z hidden

Another possible scenario consists of training with an artificial dataset, which allows us to observe the *clean score* **s**, while assuming that **z** is hidden. In this case, the algorithm clusters automatically the types of distortions present in the dataset based on the values of the quality measures. Having a hidden variable, the training procedure is not as simple as in the previous section and we need to apply the EM algorithm [Bishop, 2006].

6.6.2.1 E-step

In the E-step, we have to compute the posterior of the hidden variables given the observed variables $P\left(\mathbf{z}_i|\hat{\mathbf{s}}_i,\mathbf{s}_i,\mathbf{Q}_i,\theta_i\right)$:

$$P\left(z_{ik}=1|\hat{\mathbf{s}}_i,\mathbf{s}_i,\mathbf{Q}_i,\theta_i\right)=\frac{\pi_{z_k}P\left(\hat{\mathbf{s}}_i|\mathbf{s}_i,\theta_i,z_{ik}=1\right)P\left(\mathbf{s}_i|\theta_i\right)P\left(\mathbf{Q}_i|z_{ik}=1\right)}{\sum_{k=1}^{K}\pi_{z_k}P\left(\hat{\mathbf{s}}_i|\mathbf{s}_i,\theta_i,z_{ik}=1\right)P\left(\mathbf{s}_i|\theta_i\right)P\left(\mathbf{Q}_i|z_{ik}=1\right)} \tag{6.27}$$

where we have to plug-in Equations (6.1), (6.4) and (6.5). We defined $\gamma(z_{ik}) = P\left(z_{ik}=1|\hat{\mathbf{s}}_i,\mathbf{s}_i,\mathbf{Q}_i,\theta_i\right)$ to keep the following equations uncluttered.

6.6.2.2 M-step

In the M-step, we maximize the EM auxiliary function:

$$\mathcal{Q}(\mathcal{M})=\sum_{i=1}^{N}\mathrm{E}\left[\ln P\left(\hat{\mathbf{s}}_i,\mathbf{Q}_i,\mathbf{s}_i|\theta_i,\mathbf{z}_i\right)\right]\ . \tag{6.28}$$

By maximizing with respect to $\pi_{\mathbf{z}}$, we obtain:

$$\pi_{z_k}=\frac{N_{z_k}}{\sum_{k=1}^{K}N_{z_k}} \tag{6.29}$$

where

$$N_{z_k}=\sum_{i=1}^{N}\gamma(z_{ik})\ . \tag{6.30}$$

By maximizing with respect to $\mu_{\mathbf{Q}_{pk}}$ and $\mathbf{\Lambda}_{\mathbf{Q}_{pk}}$, we obtain:

$$\mu_{\mathbf{Q}_{pk}}=\frac{1}{N_{z_k}}\sum_{i=1}^{N}\gamma(z_{ik})\mathbf{Q}_{ip} \tag{6.31}$$

$$\mathbf{\Lambda}_{\mathbf{Q}_{pk}}^{-1}=\frac{1}{N_{z_k}}\sum_{i=1}^{N}\gamma(z_{ik})\left(\mathbf{Q}_{ip}-\mu_{\mathbf{Q}_{pk}}\right)\left(\mathbf{Q}_{ip}-\mu_{\mathbf{Q}_{pk}}\right)^T\ . \tag{6.32}$$

If we maximize respect to $\mu_{\mathbf{s}}$ and $\mathbf{\Lambda}_{\mathbf{s}}$ we obtain:

$$\mu_{\mathbf{s}_\theta}=\frac{1}{N_\theta}\sum_{i=1}^{N}t_{i\theta}\mathbf{s}_i \tag{6.33}$$

$$\mathbf{\Lambda}_{\mathbf{s}_\theta}^{-1}=\frac{1}{N_\theta}\sum_{i=1}^{N}t_{i\theta}\mathbf{s}_i\mathbf{s}_i^T-\mu_{\mathbf{s}_\theta}\mu_{\mathbf{s}_\theta}^T \tag{6.34}$$

where $t_{i\theta}=1$ if $\theta_i=\theta$ and $t_{i\theta}=0$ if $\theta_i\neq\theta$; and $N_\theta=\sum_{i=1}^{N}t_{i\theta}$.

Finally, we maximize respect to $\mu_{\Delta\mathbf{s}}$ and $\mathbf{\Lambda}_{\Delta\mathbf{s}}$ obtaining

$$\mu_{\Delta\mathbf{s}_{k\theta}}=\frac{1}{N_{\theta z_k}}\sum_{i=1}^{N}\gamma(\theta_i,z_{ik})\left(\hat{\mathbf{s}}_i-\mathbf{s}_i\right) \tag{6.35}$$

$$\mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta}}^{-1}=\frac{1}{N_{\theta z_k}}\sum_{i=1}^{N}\gamma(\theta_i,z_{ik})\left(\hat{\mathbf{s}}_i-\mathbf{s}_i\right)\left(\hat{\mathbf{s}}_i-\mathbf{s}_i\right)^T-\mu_{\Delta\mathbf{s}_{k\theta}}\mu_{\Delta\mathbf{s}_{k\theta}}^T \tag{6.36}$$

where we defined

$$\gamma(\theta_i, z_{ik}) = t_{i\theta}\gamma(z_{ik}) \tag{6.37}$$

$$N_{\theta z_k} = \sum_{i=1}^{N} \gamma(\theta_i, z_{ik}) \; . \tag{6.38}$$

6.6.2.3 EM initialization heuristics

The EM algorithm needs that we initialize the parameters of the network before we can start to iterate. First, we initialized the distributions $P(\mathbf{Q}|z_k = 1) = \prod_{p=1}^{P} P(\mathbf{Q}_p|z_k = 1)$. For that, we trained a GMM with diagonal covariance Gaussians for $\mathbf{Q}$ where the GMM occupations were randomly initialized. Then, we assigned each component of the GMM to one distribution $P(\mathbf{Q}_p|z_k = 1)$. The weights of the GMM were employed to initialize the values of π_{z_k}. Then, we computed the membership probabilities of each trial to each quality state z_k given the quality measures:

$$P(z_{ik} = 1|\mathbf{Q}_i) = \frac{\pi_{z_k} P(\mathbf{Q}_i|z_{ik} = 1)}{\sum_{k=1}^{K} \pi_{z_k} P(\mathbf{Q}_i|z_{ik} = 1)} \; . \tag{6.39}$$

We approximated $\gamma(z_{ik}) \approx P(z_{ik} = 1|\mathbf{Q}_i)$ and initialized $\mu_{\Delta \mathbf{s}_{k\theta}}$ and $\mathbf{\Lambda}_{\Delta \mathbf{s}_{k\theta}}$ by applying (6.35) and (6.36).

Initialization of $\mu_{\mathbf{s}_\theta}$ and $\mathbf{\Lambda}_{\mathbf{s}_\theta}$ is straightforward because θ and $\mathbf{s}$ are observed.

6.6.3 General case, training with s and z hidden

We may want to train the Bayesian network on a real dataset instead of doing it on a synthetic dataset. Then, we will not have any clean trial corresponding to each noisy trial so we will have to train the network assuming $\mathbf{s}$ hidden. The degradation type of each trial will also be unknown so $\mathbf{z}$ will be hidden. This is the most general case that we can find and we solved it with another EM algorithm.

6.6.3.1 E-step

In the E-step we compute $P(\mathbf{s}_i, \mathbf{z}_i|\hat{\mathbf{s}}_i, \mathbf{Q}_i, \theta_i)$

$$P(\mathbf{s}_i, \mathbf{z}_i|\hat{\mathbf{s}}_i, \mathbf{Q}_i, \theta_i) = P(\mathbf{s}_i|\hat{\mathbf{s}}_i, \theta_i, \mathbf{z}_i) \, P(\mathbf{z}_i|\hat{\mathbf{s}}_i, \mathbf{Q}_i, \theta_i) \; . \tag{6.40}$$

The first term is given by (6.19).

The second term is

$$P(z_{ik} = 1|\hat{\mathbf{s}}_i, \mathbf{Q}_i, \theta_i) = \frac{\pi_{z_k} P(\hat{\mathbf{s}}_i|z_{ik} = 1, \theta_i) \, P(\mathbf{Q}_i|z_{ik} = 1)}{\sum_{k=1}^{K} \pi_{z_k} P(\hat{\mathbf{s}}_i|z_{ik} = 1, \theta_i) \, P(\mathbf{Q}_i|z_{ik} = 1)} \tag{6.41}$$

where we plug-in (6.15) and (6.5). We define $\gamma(z_{ik}) = P(z_{ik} = 1|\hat{\mathbf{s}}_i, \mathbf{Q}_i, \theta_i)$ to keep the following equations uncluttered.

6.6.3.2 M-step

In the M-step, we maximize the EM auxiliary function:

$$\mathcal{Q}(\mathcal{M}) = \sum_{i=1}^{N} \mathrm{E}\left[\ln P\left(\hat{\mathbf{s}}_i, \mathbf{Q}_i, \mathbf{s}_i, \mathbf{z}_i|\theta_i\right)\right] \ . \tag{6.42}$$

The equations for $\pi_{\mathbf{z}}$, $\mu_{\mathbf{Q}_{pk}}$ and $\boldsymbol{\Lambda}_{\mathbf{Q}_{pk}}$ are the same as for the case with $\mathbf{s}$ observed. Now, we maximize with respect to $\mu_{\mathbf{s}}$ and $\boldsymbol{\Lambda}_{\mathbf{s}}$ obtaining:

$$\mu_{\mathbf{s}_\theta} = \frac{1}{N_\theta}\sum_{i=1}^{N}\sum_{k=1}^{K}\gamma(\theta_i, z_{ik})\mu'_{\mathbf{s}_{ik\theta}} \tag{6.43}$$

$$\boldsymbol{\Lambda}_{\mathbf{s}_\theta}^{-1} = \frac{1}{N_\theta}\sum_{i=1}^{N}\sum_{k=1}^{K}\gamma(\theta_i, z_{ik})\left(\boldsymbol{\Lambda}'^{-1}_{\mathbf{s}_{k\theta}} + \mu'_{\mathbf{s}_{ik\theta}}\mu'^{T}_{\mathbf{s}_{ik\theta}}\right) - \mu_{\mathbf{s}_\theta}\mu_{\mathbf{s}_\theta}^{T} \tag{6.44}$$

$$\tag{6.45}$$

where we defined $t_{i\theta} = 1$ if $\theta_i = \theta$, and

$$\gamma(\theta_i, z_{ik}) = t_{i\theta}\gamma(z_{ik}) \tag{6.46}$$

$$N_\theta = \sum_{i=1}^{N} t_{i\theta} \ . \tag{6.47}$$

Finally, we maximize with respect to $\mu_{\Delta\mathbf{s}}$ and $\boldsymbol{\Lambda}_{\Delta\mathbf{s}}$ and obtain:

$$\mu_{\Delta\mathbf{s}_{k\theta}} = \frac{1}{N_{\theta z_k}}\sum_{i=1}^{N}\gamma(\theta_i, z_{ik})\left(\hat{\mathbf{s}}_i - \mu'_{\mathbf{s}_{ik\theta}}\right) \tag{6.48}$$

$$\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}^{-1} = \frac{1}{N_{\theta z_k}}\sum_{i=1}^{N}\gamma(\theta_i, z_{ik})\left(\hat{\mathbf{s}}_i - \mu'_{\mathbf{s}_{ik\theta}}\right)\left(\hat{\mathbf{s}}_i - \mu'_{\mathbf{s}_{ik\theta}}\right)^{T} + \boldsymbol{\Lambda}'^{-1}_{\mathbf{s}_{k\theta}} - \mu_{\Delta\mathbf{s}_{k\theta}}\mu_{\Delta\mathbf{s}_{k\theta}}^{T} \tag{6.49}$$

where we defined

$$N_{\theta z_k} = \sum_{i=1}^{N}\gamma(\theta_i, z_{ik}) \ . \tag{6.50}$$

6.6.3.3 EM initialization heuristics

For the general case, the distributions of $\mathbf{Q}$, $P\left(\mathbf{Q}|z_k = 1\right)$ and the weights π_{z_k} were initialized as explained in Section 6.6.2.3.

Again, we computed the membership probabilities $P\left(z_{ik} = 1|\mathbf{Q}_i\right)$ with (6.39). We approximated $\gamma(z_{ik}) \approx P\left(z_{ik} = 1|\mathbf{Q}_i\right)$ and use it to estimate the observed score distributions $P\left(\hat{\mathbf{s}}|\theta, \mathbf{z}_k = 1\right)$ by approximating their means and variances as

$$\mu'_{\hat{\mathbf{s}}_{k\theta}} = \frac{1}{N_{\theta z_k}}\sum_{i=1}^{N}\gamma(\theta_i, z_{ik})\hat{\mathbf{s}}_i \tag{6.51}$$

$$\boldsymbol{\Lambda}'_{\hat{\mathbf{s}}_{k\theta}} = \frac{1}{N_{\theta z_k}}\sum_{i=1}^{N}\gamma(\theta_i, z_{ik})\hat{\mathbf{s}}_i\hat{\mathbf{s}}_i^{T} - \mu'_{\hat{\mathbf{s}}_{k\theta}}\mu'_{\hat{\mathbf{s}}_{k\theta}} \ . \tag{6.52}$$

For each quality level k we compute the Fisher ratio between the both distributions $P\left(\hat{\mathbf{s}}|\theta=\mathcal{T},\mathbf{z}_k=1\right)$ and $P\left(\hat{\mathbf{s}}|\theta=\mathcal{N},\mathbf{z}_k=1\right)$. When the Fisher ratio is high the target and non-target score distributions are more separated and the error rate is lower. We assumed that the component k^* with the highest Fisher ratio corresponds to the clean trials. Then, we divided the mean and variances of $P\left(\hat{\mathbf{s}}|\theta,\mathbf{z}_{k^*}=1\right)$ between the distributions $P\left(\mathbf{s}|\theta\right)$ and $P\left(\hat{\mathbf{s}}|\mathbf{s},\theta,\mathbf{z}_{k^*}=1\right)$ considering that in k^* the value of Δs should be very small. We assigned all the mean and most of the variance to $P\left(\mathbf{s}|\theta\right)$–75% of the variance–and the rest to $P\left(\hat{\mathbf{s}}|\mathbf{s},\theta,\mathbf{z}_{k^*}=1\right)$–25% of the variance. Thus, we initialized the *clean score* the distribution $P\left(\mathbf{s}|\theta\right)$ as:

$$\boldsymbol{\mu}_{\mathbf{s}_\theta} = \boldsymbol{\mu}_{\hat{\mathbf{s}}_{k^*\theta}} \tag{6.53}$$

$$\boldsymbol{\Lambda}_{\mathbf{s}_\theta} = \boldsymbol{\Lambda}'_{\hat{\mathbf{s}}_{k^*\theta}}/0.75 \tag{6.54}$$

and the distribution $P\left(\hat{\mathbf{s}}|\mathbf{s},\theta,\mathbf{z}_{k^*}=1\right)$ was initialized as

$$\boldsymbol{\mu}_{\Delta\mathbf{s}_{k^*\theta}} = \mathbf{0} \tag{6.55}$$

$$\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k^*\theta}} = \boldsymbol{\Lambda}'_{\hat{\mathbf{s}}_{k^*\theta}}/0.25\;. \tag{6.56}$$

We distributed the variance of $P\left(\hat{\mathbf{s}}|\theta,\mathbf{z}_{k^*}=1\right)$ between $P\left(\mathbf{s}|\theta\right)$ and $P\left(\hat{\mathbf{s}}|\mathbf{s},\theta,\mathbf{z}_{k^*}=1\right)$ so that neither of them have an infinite precision.

Finally, for the rest of components, we initialized $P\left(\hat{\mathbf{s}}|\mathbf{s},\theta,\mathbf{z}_k=1\right)$ from $P\left(\hat{\mathbf{s}}|\theta,\mathbf{z}_k=1\right)$ and $P\left(\mathbf{s}|\theta\right)$. From Equations (6.17) and (6.16), we isolated $\boldsymbol{\mu}_{\Delta\mathbf{s}_{k\theta}}$ and $\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}$:

$$\boldsymbol{\mu}_{\Delta\mathbf{s}_{k\theta}} = \boldsymbol{\mu}'_{\hat{\mathbf{s}}_{k\theta}} - \boldsymbol{\mu}_{\mathbf{s}_\theta} \tag{6.57}$$

$$\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}} = \left(\boldsymbol{\Lambda}'^{-1}_{\mathbf{s}_{k\theta}} - \boldsymbol{\Lambda}^{-1}_{\mathbf{s}_\theta}\right)^{-1} \tag{6.58}$$

If $\boldsymbol{\Lambda}'^{-1}_{\mathbf{s}_{k\theta}} - \boldsymbol{\Lambda}^{-1}_{\mathbf{s}_\theta}$ is not positive definite we just did

$$\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}} = \boldsymbol{\Lambda}'_{\mathbf{s}_{k\theta}}\;. \tag{6.59}$$

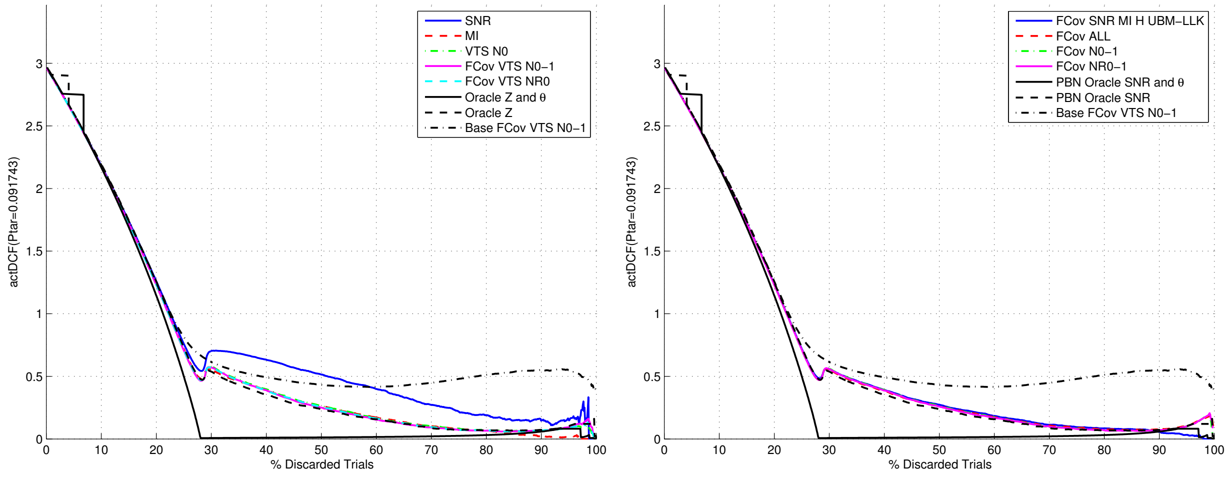
6.7 Experiments

6.7.1 Experiments on NIST SRE with noise added

For these experiments, we used the dataset described in Section 3.4.2.1, a dataset artificially degraded by adding noise to telephone signals from NIST SRE08 and SRE10. Trials from SRE08 were used to train the BN and SRE10 to evaluate the reliability detection performance. By pooling all the noisy trials of NIST SRE10, our SV system obtained actual DCF=2.96. We discarded unreliable trials to reduce it to values lower than one.

6.7.1.1 BN trained with s and $\mathbf{z}$ observed

As explained in Section 6.6.1, we could define a quality state for each pair of enrollment and test signal-to-noise ratios (SNR_{enroll},SNR_{tst}). Then, for our dataset, $\mathbf{z}$ could take 36 different values. As we degraded the datasets ourselves, we know the actual SNR of the segments and we can train the network assuming that the quality states $\mathbf{z}$ are observed variables. Besides, for each noisy trial we know the corresponding clean trial so the *clean*



(a) Single measure.

(b) Multiple measures.

Figure 6.5: % Discarded trials vs. actual DCF for NIST SRE10 + noise using BN trained with $\hat{s}$ and $\mathbf{z}$ observed.

score s is also observed. In this scenario where all the variables are observed, training the network by maximum likelihood is simple, as explained in Section 6.6.1.

In the test phase, both s and $\mathbf{z}$ were hidden and their posteriors were estimated as shown in Section 6.4.1. We also did two cheating experiments. The first one consisted in computing the posterior of s given observed labels θ and quality states $\mathbf{z}$, that is the case shown in Section 6.4.2. This case is never going to happen given that the purpose of SV is precisely estimating θ but this ideal scenario provides a lower bound to the actual DCF that we could obtain. For the second one, we computed the posterior of s given observed $\mathbf{z}$ and hidden θ. This scenario is also unrealistic because for most of the datasets the noise level is unknown, but it also provides a reference point for comparison.

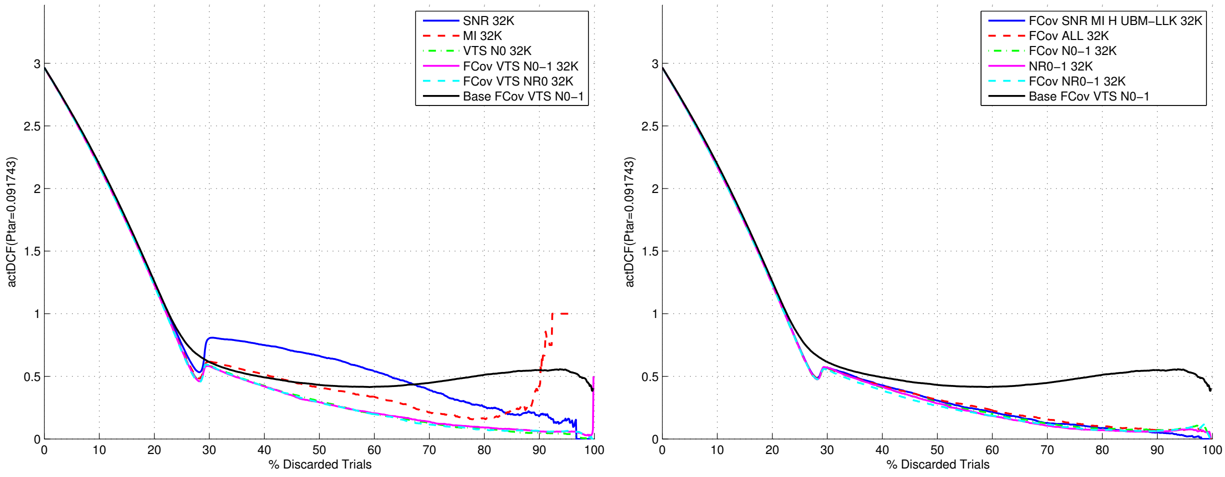
From the posterior of s, we computed the posterior probability for taking a reliable decision with (6.7). The reliability posterior $P\left(R|\hat{s},\mathbf{Q}\right)$ was thresholded to decide which trials were unreliable and needed to be discarded. By applying a varying threshold we obtained curves plotting actual DCF against percentage of discarded trials. In this context the DCF was computed taken into account only the trials classified as reliable as we explained in Section 3.3.2. Figure 6.5 compares curves of actual cost against percentage of discarded trials for several quality measures and for both cheating experiments. Besides, we added a curve with the best result achieved by the BN employed in the previous chapter. We took that as baseline. In Table 6.1, we show the costs values obtained by sampling the curves in 25%, 50% and 75% of discarded trials. We divided the figure and the table into two parts: one where each BN only uses one quality measure (Figure 6.5a) and another where the BN combines several quality measures (Figure 6.5b). As in the previous chapter, to simplify the visualization of the results, we only present results with the measures that provided lower costs. Following, we remind the notation used for the measures. Signal-to-noise ratio is denoted by SNR, modulation index by MI, entropy by H and UBM log-likelihood by UBM-LLk. The term VTS N refers to the VTS coefficients with LDA trained to discriminate between different SNR levels and, VTS R is the same but with LDA trained to discriminate between reverberation times. The numbers behind N or R indicate the output dimension of the LDA projection that we keep, for example N_{0-1} means that we

Table 6.1: % Discarded trials vs. actual DCF for NIST SRE10 + noise with the BN trained with $\hat{s}$ and $\mathbf{z}$ observed.

$\mathbf{Q}$	% Discarded Trials		
	25	50	75
Oracle $\mathbf{z}$ and θ	0.5	0.01	0.02
Oracle $\mathbf{z}$	0.75	0.23	0.07
SNR	0.76	0.51	0.22
MI	0.72	0.26	0.08
VTS N_0	**0.71**	0.26	0.08
FCov VTS N_{0-1}	**0.71**	**0.25**	**0.07**
FCov VTS NR_0	**0.71**	**0.25**	**0.07**
FCov SNR MI H UBM-LLk	0.72	0.27	0.09
FCov ALL	**0.71**	**0.26**	0.09
FCov N_{0-1}	**0.71**	**0.26**	**0.08**
FCov NR_{0-1}	**0.71**	**0.26**	**0.08**
Baseline FCov VTS NR_0	0.76	0.46	0.48

keep the first two coefficients. When we write VTS NR, we mean that we concatenate the both LDA projections, the one for noise and the one for reverberation. When we say that we use *All* the measures we mean: SNR, MI, H, UBM-LLk, VTS NR_0, jitter and shimmer. In the same way as for the VTS parameters, the terms N, R and NR refer to LDA projections of *All* the quality measures to discriminate noise or/and reverberation times. We denote by *FCov* the networks where the distribution of the quality measures conditioned to each quality state $P(\mathbf{Q}|z_k = 1)$ for $k = 1, \ldots, K$ is a full-covariance Gaussian. Otherwise, it is a product of full-covariance bi-dimensional Gaussians (equivalent to block diagonal Gaussian). There was one bi-Gaussian for each quality measure. The first dimension of each Gaussian corresponded to the enrollment segment measure, and the second one corresponded to the test segment measure.

As expected, the cheating experiment with observed test θ and $\mathbf{z}$ (Oracle SNR and θ) attains the best performance. It reaches a cost almost zero by discarding 30% of the trials. The experiment with observed $\mathbf{z}$ (Oracle SNR) does not reaches costs so low, but it is still much better than the baseline. For the baseline, we obtained the best results with the VTS based features. The new BN and the baseline performed similarly if we discard less than 15% of the trials but, from there, the proposed approach performed much better. The baseline did not provide costs lower than 0.38 while the new approach reached much lower costs. For the rest of experiments, $\mathbf{z}$ was hidden in test and it had to be estimated from the quality measures. Fortunately, we obtained almost the same curves as with $\mathbf{z}$ observed. Only the BN with SNR is worse. Thus, we can reduce the actual cost from 2.96 to values like 0.71, 0.25 and 0.07 if the application allows us to discard 25, 50 or 75% of the trials. Other measures like UBM-LLk, Entropy or jitter did not attained good results.



(a) Single measure.

(b) Multiple measures.

Figure 6.6: % Discarded trials vs. actual DCF for NIST SRE10 + noise using BN trained with $\hat{s}$ observed and $\mathbf{z}$ hidden.

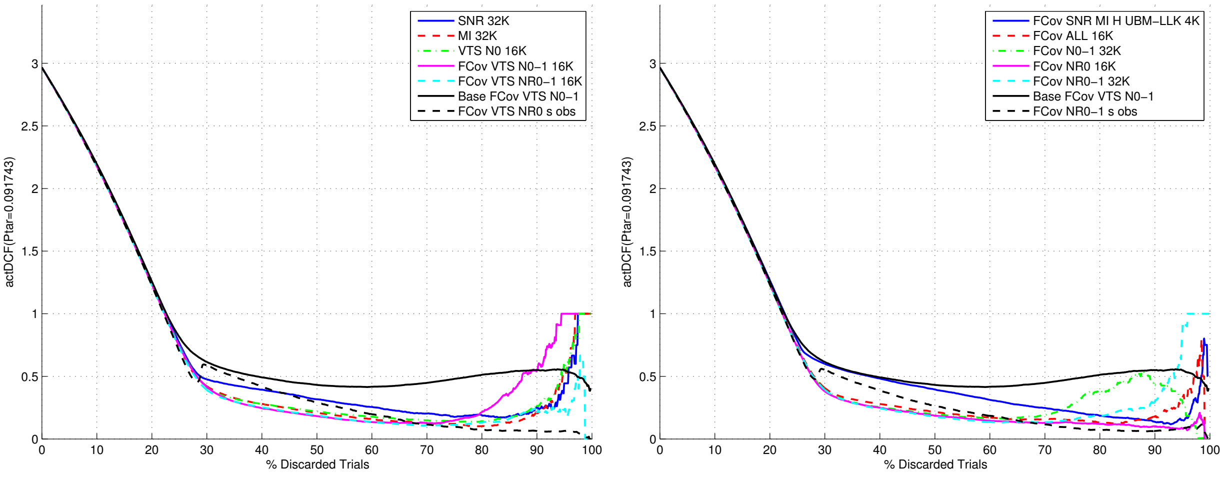
6.7.1.2 BN trained with s observed and $\mathbf{z}$ hidden

If we did not know the sources of degradation of the training dataset or we just wanted the algorithm to decide the quality states unsupervisedly, we would train the BN with $\mathbf{z}$ hidden. Here, we still would assume that the *clean scores* s were observed to train the BN. For that, we used the EM algorithm in Section 6.6.2.

Figure 6.6 and Table 6.2 compare actual cost against percentage of discarded trials for the best quality measures evaluated. Subfigure 6.6a corresponds to networks using only one measure and Subfigure 6.6b to networks that combine multiple measures. The column K in

Table 6.2: % Discarded trials vs. actual DCF for NIST SRE10 + noise with the BN trained with $\hat{s}$ observed and $\mathbf{z}$ hidden.

Q	K	% Discarded Trials 25	 50	 75
SNR	32	0.75	0.66	0.31
MI	32	0.72	0.42	0.18
VTS N_0	32	0.71	0.30	**0.09**
FCov VTS N_{0-1}	32	0.71	0.29	0.11
FCov VTS NR_0	32	**0.70**	**0.29**	0.10
FCov SNR MI H UBM-LLk	32	0.72	0.31	0.11
FCov All	32	0.72	0.31	0.12
FCov N_{0-1}	32	**0.71**	0.29	0.10
NR_{0-1}	32	0.72	0.28	**0.08**
FCov NR_{0-1}	32	**0.71**	**0.26**	0.09
FCov VTS NR_0 Train $\mathbf{z}$ obs.	36	0.71	0.25	0.07
Baseline FCov VTS NR_0	–	0.76	0.46	0.48



(a) Single measure.

(b) Multiple measures.

Figure 6.7: % Discarded trials vs. actual DCF for NIST SRE10 + noise using BN trained with $\hat{s}$ and $\mathbf{z}$ hidden.

the table indicates the optimum number of quality states that was 32 for all the measures. Compared with the BN trained with $\mathbf{z}$ observed, all measures, except signal-to-noise ration and modulation index, reached the same performance. Compared with the baseline, only the SNR produced a curve worse than the baseline.

6.7.1.3 BN trained with s and $\mathbf{z}$ hidden

The most general case that we could consider consists in training the BN with hidden s and $\mathbf{z}$. We would need this to train the BN on a database where each noisy trial does not have a corresponding clean trial that can provide the *clean score*. That is the case of databases captured in noisy environments instead of being artificially degraded by adding noise. Then, to train the network we need to apply the EM algorithm in Section 6.6.3.

Figure 6.7 and Table 6.3 shows actual costs against percentage of discarded trials. Subfigure 6.6a plots results with only one measure and Subfigure 6.6b combines multiple measures. As there are two hidden variables, training the network is more challenging and we expected an important performance lost. On the contrary, we achieved very good performance even outperforming the network trained with s observed. While discarding less than 70% of trials, the cost obtained by training with s hidden are lower than those obtained with s observed. We speculate that it is due to the fact that the original SRE trials, that we assumed clean, are not completely clean actually. In this case, the original trials were treated in the same manner as the noisy trials, i.e., having observed and hidden scores. Then, the algorithm could, in theory, find a better estimation for the *clean scores* s. The best performance was obtained with the measure NR_0, closely followed by the vector of all the features, NR_{0-1}, VTS NR_{0-1}, VTS N_0 and modulation index. The worse performance is given by the SNR and the combination of SNR MI H UBM-LLk, however, they still outperformed the baseline.

In conclusion, we evaluated three methods of training our BN and proved that, in all cases, it was able to outperform the baseline. According to the curves, the best measures to determine the reliability for a dataset with additive noise are modulation index and

Table 6.3: % Discarded trials vs. actual DCF for NIST SRE10 + noise with the BN trained with $\hat{s}$ and $\mathbf{z}$ hidden.

Q	K	% Discarded Trials		
		25	50	75
SNR	32	0.76	0.32	0.18
MI	32	0.73	0.21	0.12
VTS N_0	16	0.74	0.22	0.15
FCov VTS N_{0-1}	16	0.74	0.18	0.15
FCov VTS NR_{0-1}	16	**0.72**	**0.19**	**0.11**
FCov SNR MI H UBM-LLk	4	0.78	0.40	0.22
FCov All	16	0.74	0.21	0.15
FCov N_{0-1}	32	**0.73**	0.20	0.27
FCov NR_0	16	**0.73**	0.19	**0.13**
FCov NR_{0-1}	32	**0.73**	**0.18**	0.16
FCov VTS NR_0 train s obs.	32	**0.70**	**0.29**	0.10
Baseline FCov VTS NR_0	–	0.76	0.46	0.48

the features derived from the VTS parameters. The combination of all the features with or without LDA dimensionality reduction also yielded good results but not provided a significant gain compared to employing only one measure.

6.7.1.4 Analysis of score distributions

For the case where all the variables were observed we can easily compute the distributions of Δs given each pair of SNR and the label θ, $P\left(\Delta s|\theta, SNR_{\text{enroll}}, SNR_{\text{tst}}\right)$. We have 2×36 different Δs distributions some which appear in Figure 6.8. Each subplot represents the distributions for a pair of SNR values. The distributions given the target and non-target hypothesis are plotted in blue and red respectively.

When one of the segments is clean and the other is noisy (row 1), if the trial is target the mean of Δs decreases rapidly as the noise increases. If the trial is non-target, the mean of Δs grows slowly with the noise. With a small amount of noise in both segments (row 2), if the trial is target the mean of Δs is near zero, but, if the trial is non-target the mean of Δs becomes quite large. Finally, with a large noise level in both sides (row 3), the mean score of the targets decreases a little and the mean score of the non-targets increases significantly. Besides, in all cases, the variance of Δs grows fast with the noise. These graphs prove that Δs is very dependent on both noise and trial labeling so the dependencies that we included in our graphical model were correct.

We can also compare the distributions obtained by applying the different flavors of the EM algorithm. We center on the networks that utilized the measure NR_{0-1}. Figure 6.9 compares the *clean score* distributions $P\left(s|\theta\right)$ obtained by training with s observed and hidden. The distributions estimated with s hidden have approximately the same means as the ones obtained with s observed. However, the variances are smaller. When training with s observed we force $s = \hat{s}$ and $\Delta s = 0$. That makes the variance of Δs for the clean state to be small. On the contrary, by training with s hidden, the trials without noise added are

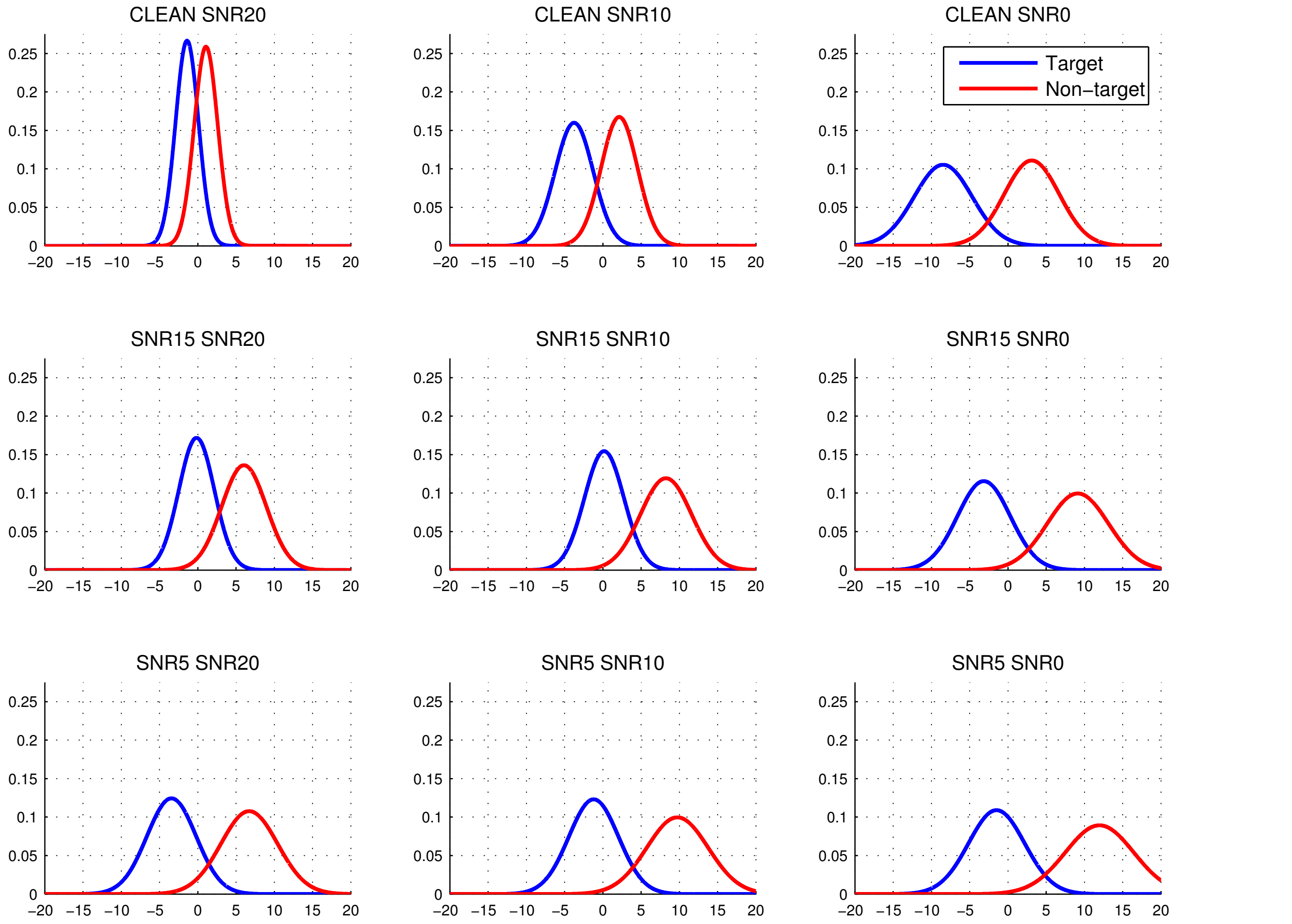


Figure 6.8: $P\left(\Delta s|\theta, SNR_{\text{enroll}}, SNR_{\text{tst}}\right)$ for several values of $(SNR_{\text{enroll}}, SNR_{\text{tst}})$.

treated in the same manner as the rest. This allows the algorithm to try to obtain a better estimate of their *clean score*. That implies $|\Delta s| > 0$ and therefore larger variance of Δs. Earlier, we proved that

$$\boldsymbol{\Lambda}'^{-1}_{\hat{\mathbf{s}}_{k\theta}} = \boldsymbol{\Lambda}^{-1}_{\mathbf{s}_\theta} + \boldsymbol{\Lambda}^{-1}_{\Delta\mathbf{s}_{k\theta}} \ , \tag{6.60}$$

so if $\boldsymbol{\Lambda}^{-1}_{\Delta\mathbf{s}_{k\theta}}$ becomes larger $\boldsymbol{\Lambda}^{-1}_{\mathbf{s}_\theta}$ has to reduce. Interestingly, the score distributions obtained in this manner are almost symmetric and not overlapped. Thus, if we could see this hidden score and put the SV threshold at zero, the error rate would be zero.

Now, we direct our attention to the Δs distributions in Figure 6.10. These distributions were computed with the trials assigned to a quality state that, approximately, corresponded to the SNR pair (15dB, 10dB). It compares the distributions obtained with the three training types. For both, targets and non-targets, the three Gaussians have similar means. In variance terms, the variances of the two distributions trained with s observed are also close. However, the variance of the Gaussians trained with s hidden is evidently smaller. This behavior was not only detected in the quality state depicted in the figure but also in the rest. Intuitively, we could think that if, as seen in Figure 6.9, $\boldsymbol{\Lambda}^{-1}_{\mathbf{s}_\theta}$ trained with s hidden is smaller, $\boldsymbol{\Lambda}^{-1}_{\Delta\mathbf{s}_{k\theta}}$ should be larger to maintain the value of $\boldsymbol{\Lambda}'^{-1}_{\hat{\mathbf{s}}_{k\theta}}$. On the contrary, we found that the three variances are smaller. This can be explained from the fact that leaving s

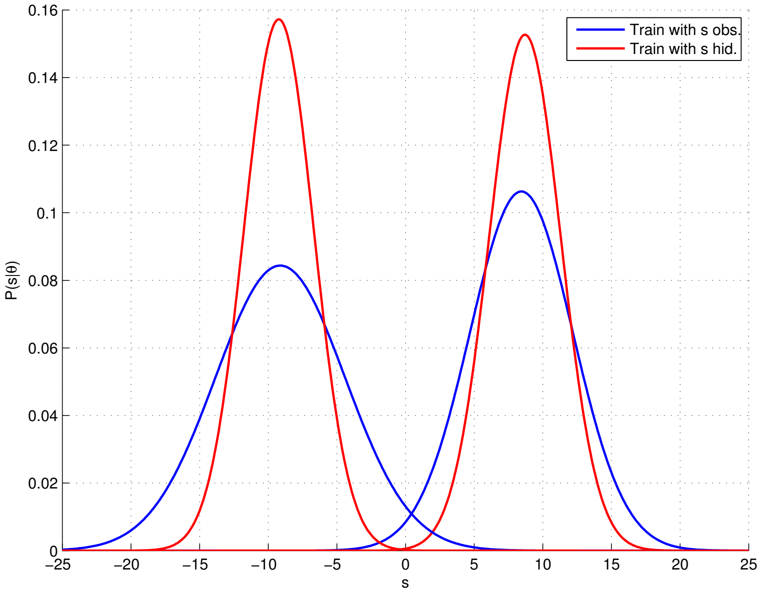
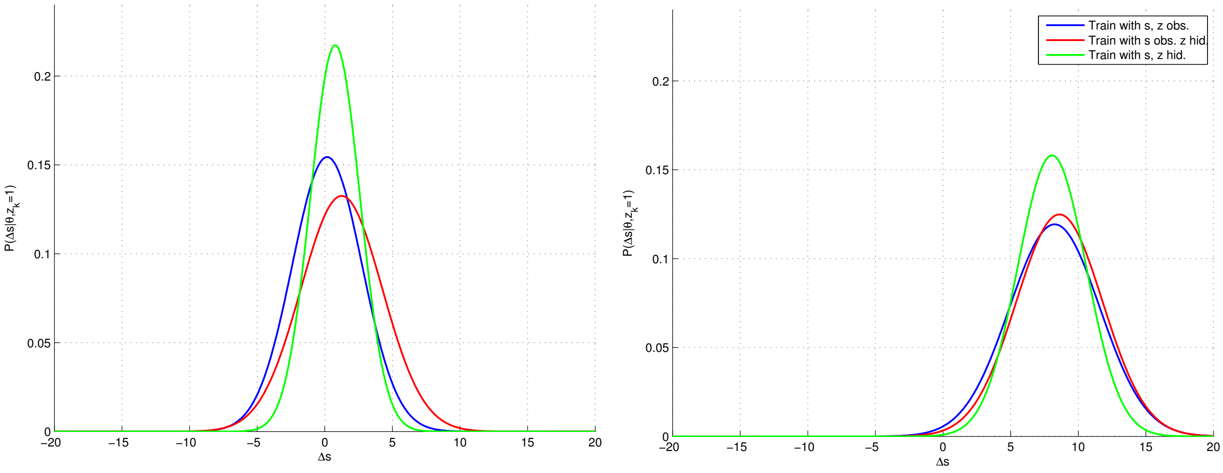


Figure 6.9: $P\left(s|\theta\right)$ with $\theta \in \{\mathcal{T},\mathcal{N}\}$, comparison between training with s observed vs. hidden.

hidden (or Δs hidden) implies a very different assignation of the trials to each quality state (Compare (6.41) and (6.27)). By training with s hidden we obtained quality states whose observed score distributions are more concentrated. We think that this helps to a better partitioning of the trials into quality states.

6.7.1.5 Quality dependent likelihood ratio

As we showed in Section 6.5 we can use the BN to obtain a refined version of the SV likelihood ratio. Then, instead of discarding unreliable trials, we maintain all the trials but



(a) Target trials distributions.

(b) Non-target trials distributions.

Figure 6.10: $P\left(\Delta s|\theta, z_k = 1\right)$ for quality state k approximately corresponding to trials with $SNR_{\text{enroll}} = 15$ dB and $SNR_{\text{tst}} = 10dB$, comparison of different flavors of the BN training

Table 6.4: EER(%)/DCF of quality dependent LLR on NIST SRE10+noise.

Q	K	EER(%)	minDCF	actDCF
Baseline	–	22.88	0.99	2.96
Train with s observed				
SNR	32	19.98	0.83	0.88
MI	32	18.35	0.78	0.81
FCov VTS N_{0-1}	32	18.01	0.76	0.81
FCov VTS NR_0	32	**17.60**	**0.75**	0.81
FCov All	32	18.14	0.78	**0.80**
FCov N_{0-1}	32	17.95	0.77	0.81
FCov NR_{0-1}	32	17.93	0.77	**0.80**
Train with s hidden				
SNR	32	20.08	0.83	0.97
MI	32	18.55	0.79	0.99
FCov VTS N_{0-1}	16	18.21	0.77	0.98
FCov VTS NR_{0-1}	16	18.09	**0.76**	**0.91**
FCov All	16	18.34	0.78	0.95
FCov N_{0-1}	32	17.87	0.78	0.95
FCov NR_{0-1}	32	**17.71**	0.78	0.95

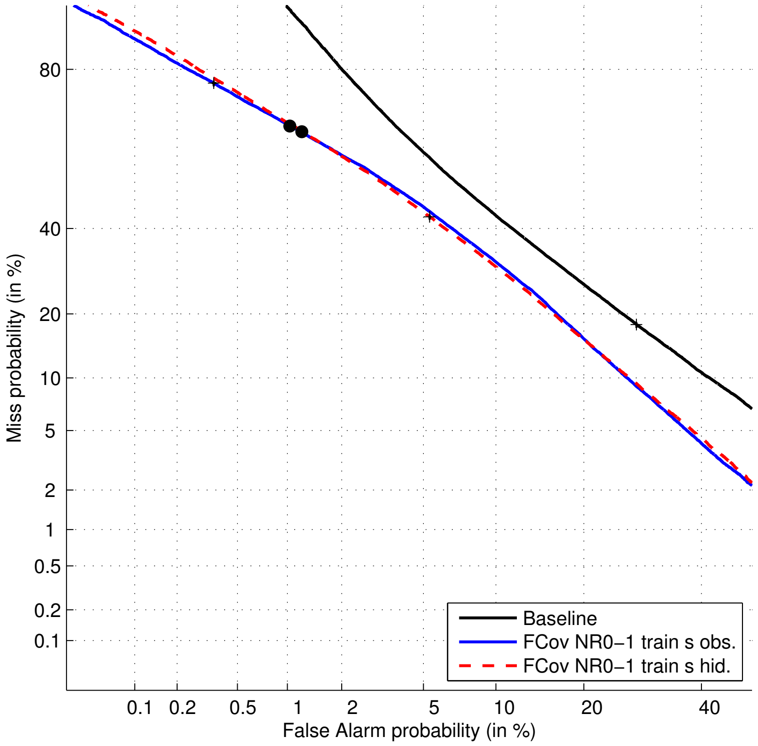
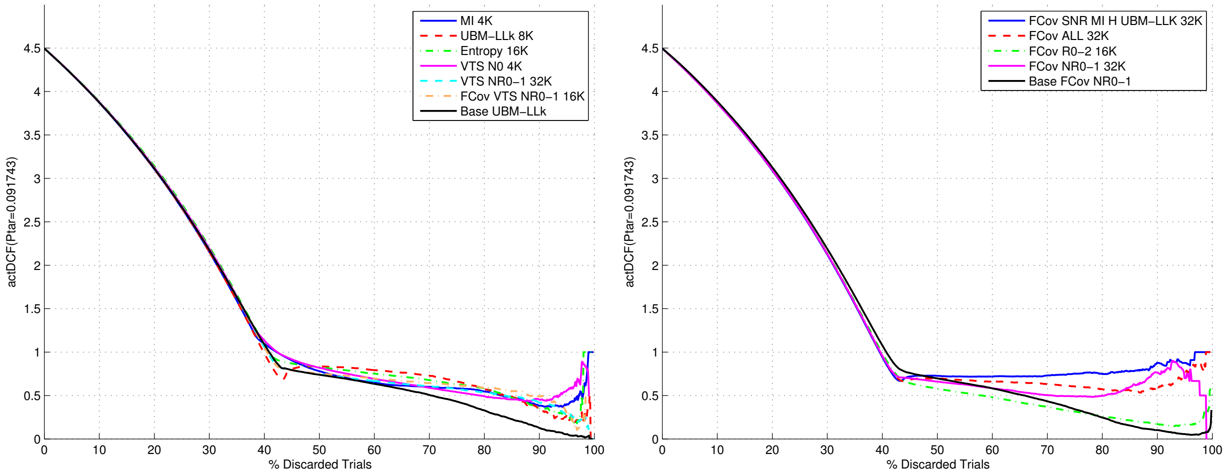


Figure 6.11: DET curves obtained from the quality dependent LLR with measure NR_{0-1}



(a) Single measure.

(b) Multiple measures.

Figure 6.12: % Discarded trials vs. actual DCF for NIST SRE10 + reverb. using BN trained with $\hat{s}$ and $\mathbf{z}$ hidden.

we substitute the standard SV score $\hat{s}$ by the ratio in (6.25).

In Table 6.4, we present results in terms of EER and DCF for BNs trained with hidden quality estates and s observed or hidden. All the measures performed quite similarly. We observe a moderate improvement in terms of EER and minimum DCF; and a large one in terms of actual DCF. This proves the utility of this model for score calibration in noisy datasets. The networks trained with s observed reached better calibration, for the measure NR_{0-1} actual cost was 15% better. Compared with the baseline we improved by around 22% in terms of EER, around 23% in terms of minimum DCF and between 68 and 72 % in terms of actual DCF.

Figure 6.11 compares DET curves for the standard and quality dependent likelihood ratios. All the curves used the measure NR_{0-1}. Both the BN trained with s observed and hidden obtained almost identical curves. We observe that there is an important improvement in all the operating points of the curve. The distance between curves increases in the low false alarm region, which is beneficial to reduce our DCF.

6.7.2 Experiments on NIST SRE with reverberation added

Here, we present experiment on NIST SRE with reverberation added as described in Section 3.4.2.2. Trials created from NIST SRE08 were used to train the network and trials from NIST SRE10 were used for evaluation. By pooling the NIST SRE10 trials with all possible degradations, our SV system provided minimum DCF=0.99 and actual DCF=4.5. We trained the Bayesian networks assuming hidden $\mathbf{z}$ and s.

6.7.2.1 Reliability detection

Figure 6.12 and Table 6.5 displays actual costs against percentage of discarded trials. Subfigure 6.12a plots results with only one measure and Subfigure 6.12b combines multiple measures. Curves are very similar while we discard the first 40% of the trials, i.e., the trials corresponding to the conditions with longer reverberation times. After that, we find

Table 6.5: % Discarded trials vs. actual DCF for NIST SRE10 + reverb. with the BN trained with $\hat{s}$ and $\mathbf{z}$ hidden.

Q	K	% Discarded Trials		
		25	50	75
MI	4	**2.66**	0.78	0.58
UBM-LLk	8	**2.66**	0.84	0.66
Entropy	16	2.70	0.83	0.63
VTS N_0	4	2.68	0.82	**0.54**
VTS NR_{0-1}	32	2.67	**0.74**	0.58
FCov VTS NR_{0-1}	16	2.67	0.75	0.62
FCov SNR MI H UBM-LLk	32	**2.63**	0.73	0.75
FCov All	32	2.64	0.69	0.60
FCov R_{0-2}	16	2.65	0.58	**0.31**
FCov NR_{0-1}	32	2.64	**0.66**	0.49
Baseline UBM-LLk	1	**2.67**	0.74	0.43
Baseline FCov NR_{0-1}	4	2.68	**0.70**	**0.34**

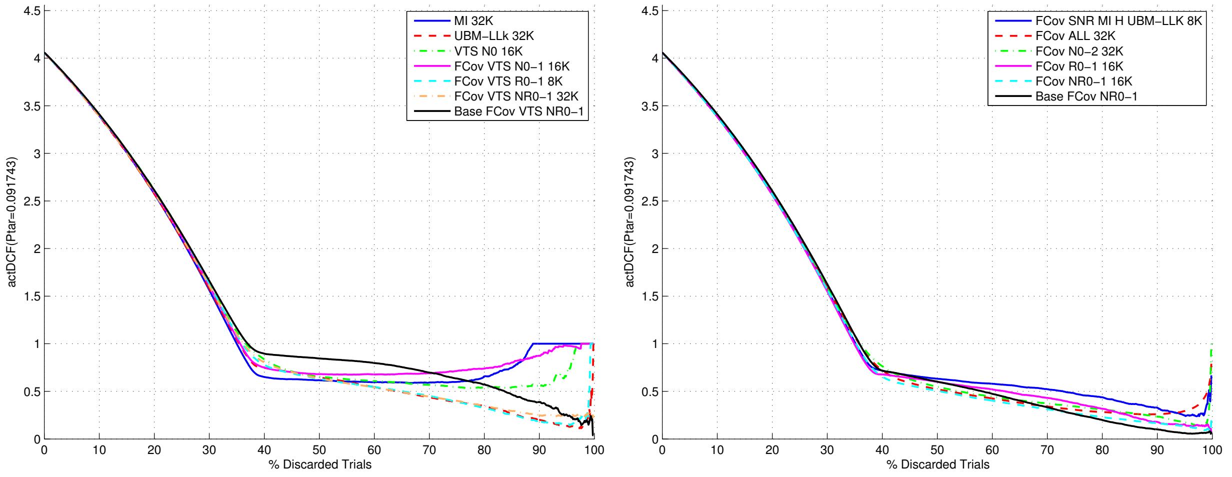
notable differences. Contrary to what happened for the dataset with noise, the baselines are better than the new BN for most of the measures. Only the measure R_{0-2} performs clearly better than the baseline if we discard less than 80% of trials. If we discard less than 60% of the trials the measures NR_{0-1}, VTS NR_{0-1} and modulation index perform similar to the baseline.

6.7.2.2 Quality dependent likelihood ratio

If instead of rejecting unreliable trials we classify all the trials with the quality dependent likelihood ratio given by the BN we obtain the results shown in Table 6.6. As for the dataset with additive noise, we note very small different between quality measures. Using the new ratio, EER improves by 18–28%, minimum DCF by 5–9% and actual DCF by 77–80%. These percentages are even larger than those obtained for the data with additive noise. The

Table 6.6: EER(%)/DCF of quality dependent LLR on NIST SRE10+reverberation with BN trained with s and $\mathbf{z}$ hidden.

Q	K	EER(%)	minDCF	actDCF
Baseline	–	33.52	0.99	4.50
MI	4	27.33	0.93	0.93
UBM-LLk	8	26.62	0.92	0.92
FCov VTS NR_{0-1}	16	28.43	0.94	1.00
FCov SNR MI H UBM-LLk	32	**23.96**	**0.90**	**0.90**
FCov All	32	24.78	0.91	0.91
FCov R_{0-2}	16	25.36	0.91	0.91
FCov NR_{0-1}	32	25.63	0.91	0.92



(a) Single measure.

(b) Multiple measures.

Figure 6.13: % Discarded trials vs. actual DCF for NIST SRE10 + noise and reverb. using BN trained with $\hat{s}$ and $\mathbf{z}$ hidden.

huge improvement of the actual cost deserves special mention given that it makes the SV system usable in that operating point. Remember that a cost larger than one means that classifying all the trials as non-target is more optimal than using the SV system. Again, the BN proves its utility for score calibration in adverse environments.

6.7.3 Experiments on NIST SRE with noise and reverberation added

We also pooled the noisy and reverberant NIST SRE trial lists. We trained the BN with pooled NIST SRE08 and tested on NIST SRE10. With this experiment, we intended to find out if the network can cope with several types of degradations at the same time.

6.7.3.1 Reliability detection

Figure 6.13 and Table 6.7 shows actual costs against percentage of discarded trials. Subfigure 6.13a plots results with only one measure and Subfigure 6.13b combines multiple measures. Again, only the best measures are shown to facilitate the results visualization. The baselines for this condition were very good so improvement was difficult. Curves are very similar for the first 35% of discarded trials, those corresponding to trials with higher noises and reverberation times. From there, we find important differences dependent of the quality measures used. The networks using *All* the measures and their LDA projection NR_{0-1} outperform the baseline if we discard less than 70% of the trials. Other measures that also yielded quite good results were N_{0-2}, R_{0-1}, VTS NR_{0-1} and VTS R_{0-1}.

We want to make note that if we consider the networks that use only one quality measure, the new BN clearly outperforms the baseline. For that reason and despite that there are small differences between both BN when using the measures NR_{0-1}, we trust in the superior modeling capacity of the new model.

An additional problem that we face when working with a reliability detector consists in choosing its operating point. One option could be to select the threshold that minimizes

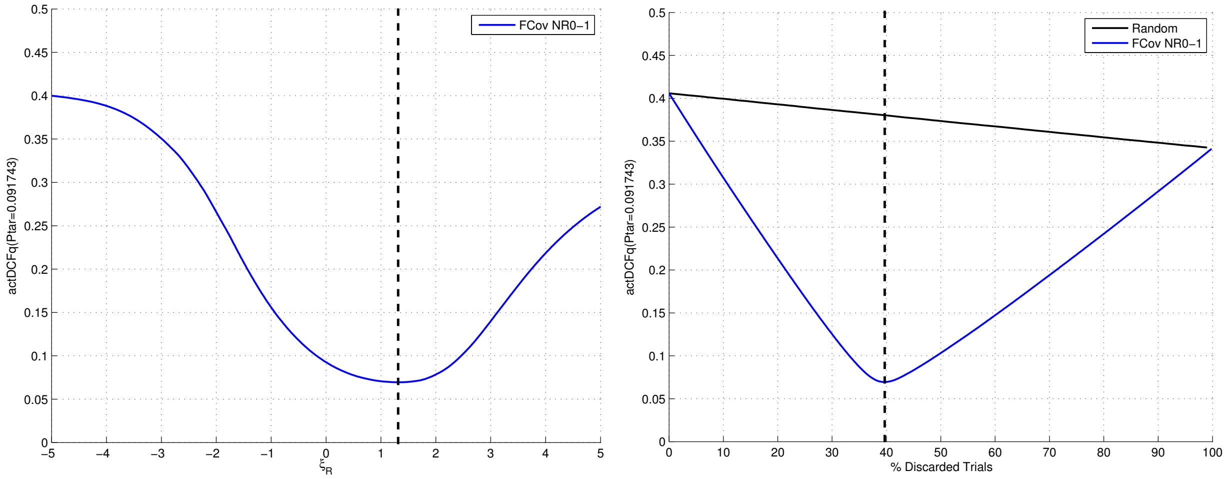
Table 6.7: % Discarded trials vs. actual DCF for NIST SRE10 + noise and reverb. with the BN trained with $\hat{s}$ and $\mathbf{z}$ hidden.

Q	K	% Discarded Trials		
		25	50	75
MI	32	**2.09**	**0.62**	0.60
UBM-LLk	32	2.10	0.65	**0.39**
VTS N_0	16	2.15	0.65	0.55
FCov VTS N_{0-1}	16	2.13	0.68	0.71
FCov VTS R_{0-1}	8	2.15	0.63	**0.39**
FCov VTS NR_{0-1}	32	2.12	0.63	0.40
FCov SNR MI H UBM-LLk	8	2.09	0.63	0.48
FCov All	32	2.10	0.52	0.31
FCov N_{0-2}	32	2.09	0.55	0.34
FCov R_{0-1}	16	**2.08**	0.60	0.38
FCov NR_{0-1}	16	2.09	**0.51**	**0.27**
Baseline FCov VTS NR_{0-1}	–	2.15	0.85	**0.63**
Baseline FCov NR_{0-1}	–	2.14	**0.60**	**0.26**

the actual cost but then, we would reject all the trials as can seen in Figure 6.13. Another option would be to reject a fixed percentage of trials. However, as happens for the MI curve in Figure 6.13a, rejecting more trials does not mean a lower cost so it is difficult to select the best percentage. Instead, we propose to apply the extended cost function C_Q defined in Section 3.3.4. This function is based on the standard DCF where we added two new terms to take into account the cost of rejecting reliable trials. There is a cost for rejecting well classified targets $C_{\mathrm{FU}\mathcal{T}}$ and another for non-targets $C_{\mathrm{FU}\mathcal{N}}$. For C_Q to make sense, the cost of rejecting well classified trials must be lower than the cost of accepting bad classified trials–these are miss and false acceptance costs of the standard DCF. Otherwise, the optimum C_Q would always correspond to not rejecting trials. For this experiment, we chose $C_{\mathrm{FU}\mathcal{T}} = C_{\mathrm{Miss}}/2 = 5$ and $C_{\mathrm{FU}\mathcal{N}} = C_{\mathrm{FA}}/2 = 0.5$. Figure 6.14 depicts the extended cost against the reliability detector decision threshold ξ_R and against the % of rejected trials. For this example, we used the measures NR_{0-1}. The curves have a clear minimum marked by the dashed vertical line. By choosing the threshold corresponding to minimum C_Q ($\xi_R = 1.33$), we accepted 60% of trials, which provide EER=22.34% and actual DCF=0.66. That improved by 26% in terms of EER and by 83% in terms of DCF. In Figure 6.15, we compare the DET curve of the accepted trials with the one of all the trials. We note that the improvement is larger in the zone of the curve around the DCF operating point. This happens because, when computing the *clean score* posterior we need to make use of the target prior (see (6.14)).

6.7.3.2 Quality dependent likelihood ratio

Once again, we show results evaluating the quality dependent likelihood ratio given by the BN instead of rejecting unreliable trials. EER and cost are displayed in Table 6.8. We improved by 12–22% in terms of EER, 3–9% in terms of minimum DCF and 75–77%



(a) Reliability threshold ξ_R vs. C_Q.

(b) % Discarded trials vs. C_Q

Figure 6.14: C_Q for NIST SRE10 + noise and reverb.

in terms of actual DCF. They are similar values to those obtained for the two previous datasets. Again, there were small differences between measures what seems to indicate that, for these datasets, the choice of measure is not critical.

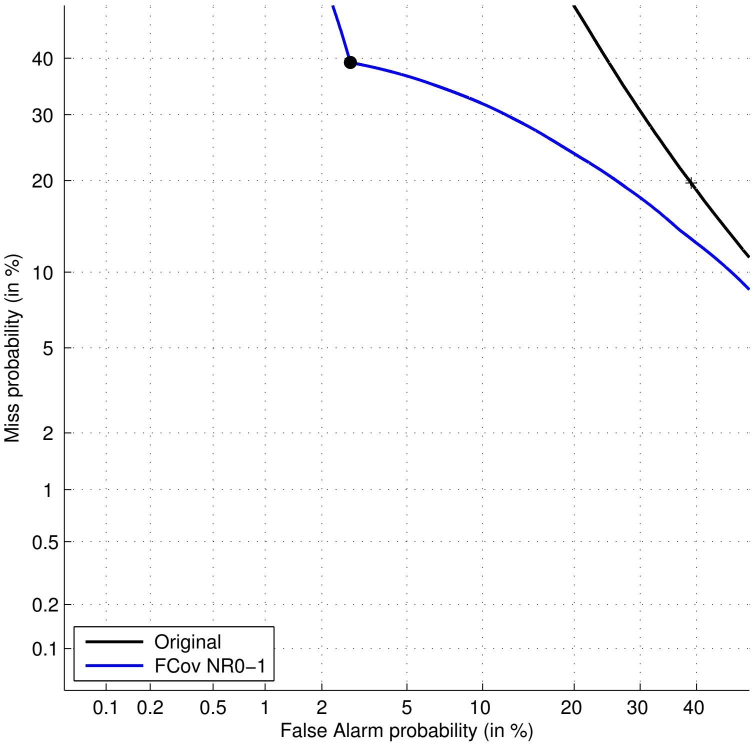


Figure 6.15: DET curve for NIST SRE10 + noise and reverb. in operating point given by minimum C_Q.

Table 6.8: EER(%)/DCF of quality dependent LLR on NIST SRE10+noise and reverberation.

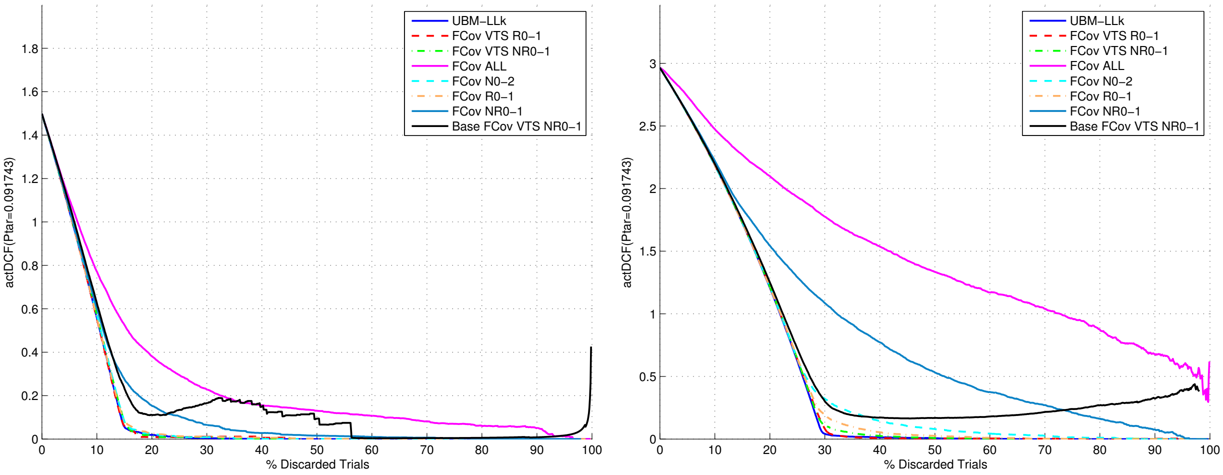
Q	K	EER(%)	minDCF	actDCF
Baseline	–	30.25	0.99	4.06
MI	32	26.20	0.92	0.93
UBM-LLk	32	25.62	0.96	1.01
FCov VTS NR_{0-1}	32	26.02	0.96	1.01
FCov SNR MI H UBM-LLk	8	25.30	0.93	0.96
FCov All	32	23.41	0.89	0.94
FCov NR_{0-1}	16	24.22	0.90	0.94

6.7.4 Experiments on databases with real distortions

As we did in the previous chapter with the baseline BN, we want to know if our new BN trained on artificial data can be used to detect unreliable trials on real databases. Again, we trained the BN on NIST SRE08 with noise and reverberation and evaluated it on Agnitio Benchmark and Ahumada. These datasets are described in Sections 3.4.2.4 and 3.4.2.5.

6.7.4.1 Reliability detection

Figure 6.16a and Table 6.9a plots actual cost against percentage of discarded trials for Agnitio Benchmark. Figure 6.16b and Table 6.9b shows costs for Ahumada. In previous experiments, we only showed results with the quality measures that reached better performance on the dataset under test. For this experiment, we supposed that is not possible to use the dataset under test to choose the best quality measure. This is a more realistic situation given that in most real cases we will not have the labels of the database under test. Instead, we present results with the quality measures that performed best on NIST SRE10



(a) Agnitio Benchmark.

(b) Ahumada.

Figure 6.16: % Discarded trials vs. actual DCF for Agnitio Benchmark and Ahumada datasets.

Table 6.9: % Discarded trials vs. actual DCF for Agnitio Benchmark and Ahumada.

(a) Agnitio Benchmark.

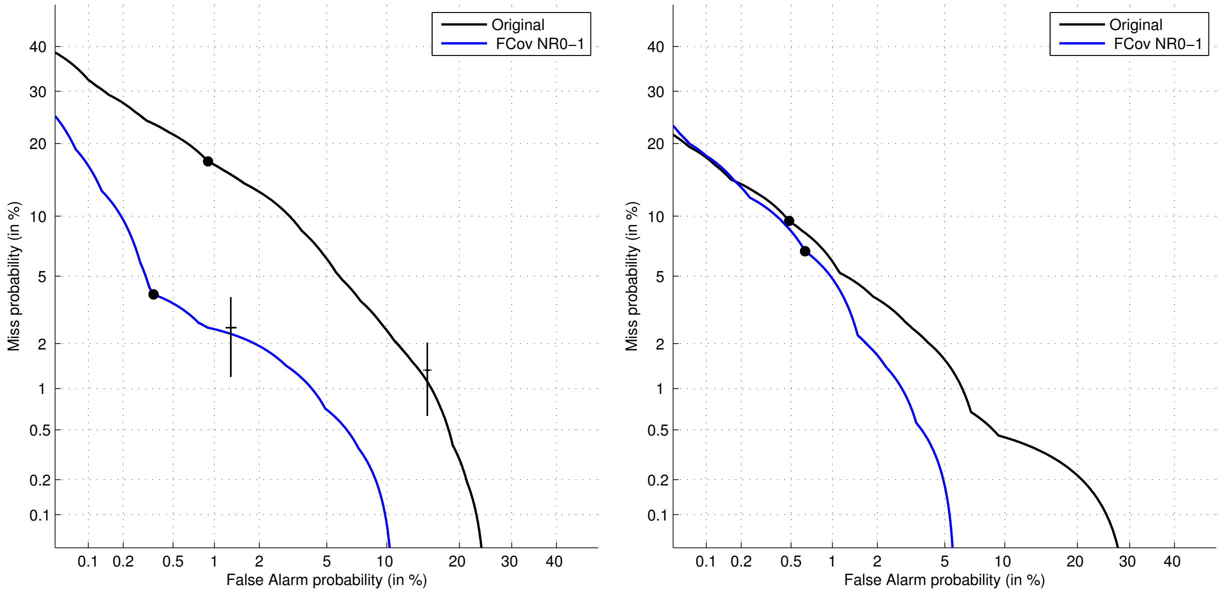
Q	% Discarded Trials		
	25	50	75
UBM-LLk	**0.01**	**0.00**	**0.00**
FCov VTS R_{0-1}	**0.01**	**0.00**	**0.00**
FCov VTS NR_{0-1}	**0.01**	**0.00**	**0.00**
FCov All	0.29	0.13	0.07
FCov N_{0-2}	**0.01**	**0.00**	**0.00**
FCov R_{0-1}	**0.01**	**0.00**	**0.00**
FCov NR_{0-1}	0.10	0.02	**0.00**
Baseline FCov VTS NR_{0-1}	**0.13**	**0.09**	0.00

(b) Ahumada.

Q	% Discarded Trials		
	25	50	75
UBM-LLk	**0.63**	**0.01**	**0.00**
FCov VTS R_{0-1}	0.64	**0.01**	**0.00**
FCov VTS NR_{0-1}	0.64	**0.01**	**0.00**
FCov All	1.93	1.33	0.96
FCov N_{0-2}	0.64	0.08	0.01
FCov R_{0-1}	0.65	0.03	**0.00**
FCov NR_{0-1}	1.29	0.53	0.21
Baseline FCov VTS NR_{0-1}	**0.72**	**0.17**	**0.24**

with noise and reverberation. We did the same in Section 5.3.4 with the baseline BN. Back then, we saw that most of the measures that performed well on NIST SRE10 performed badly on Agnitio Benchmark and Ahumada. Only the measure VTS NR_{0-1} provided good results. On the contrary, for the new BN, all the measures were usable. The worst measures, especially on Ahumada, were NR_{0-1} and the fusion of *All* measures. The rest of measures provided curves whose costs were always under the baseline. These results prove that the proposed network generalizes better than the baseline and it is better to be used in different datasets.

As operating point of the reliability detector, we chose to keep the same threshold that we used for NIST SRE10 with noise and reverberation. Even though, it is not the best measure for these two datasets, we show results with the measure NR_{0-1}. Thus, for Agnitio



(a) Agnitio Benchmark.

(b) Ahumada.

Figure 6.17: DET curves for Agnitio Benchmark and Ahumada datasets with reliability detection system working in operating point with minimum C_Q in NIST SRE10.

Table 6.10: EER(%)/DCF of quality dependent LLR on Agnitio benchmark and Ahumada.

(a) Agnitio Benchmark.

Q	EER(%)	minDCF	actDCF
Baseline	**5.46**	**0.26**	1.49
UBM-LLk	**5.85**	**0.31**	**0.33**
FCov VTS NR_{0-1}	8.32	0.42	0.63
FCov All	10.13	0.56	0.73
FCov NR_{0-1}	8.31	0.61	0.65

(b) Ahumada.

Q	EER(%)	minDCF	actDCF
Baseline	2.85	0.14	2.96
UBM-LLk	3.17	0.16	0.44
FCov VTS NR_{0-1}	7.67	0.42	0.43
FCov All	7.61	0.39	1.46
FCov NR_{0-1}	10.08	0.44	1.11

Benchmark we attained EER=1.95%, actual DCF=0.15 and rejected 20% of the trials. Compared with keeping all the trials, this meant an improvement of 64% in terms of EER and 90% in terms of DCF. For Ahumada, we obtained EER=1.84%, actual DCF=0.81 and pruned 39% of trials. That improved by 35% in terms of EER and by 72% in terms of actual cost. Figure 6.17 plots the corresponding DET curves for both datasets. For Agnitio benchmark we witness a large improvement along all the curve. For Ahumada, the distance between curves is smaller though the actual cost improved importantly.

6.7.4.2 Quality dependent likelihood ratio

Finally, we applied the BN trained on NIST SRE10 with noise and reverberation to compute quality dependent likelihood ratios. EER and DCF evaluated in the full trial lists are shown in Table 6.10. Contrary to what we obtained for NIST SRE10, EER and minimum DCF worsened, for some measures badly. On the other hand, actual DCF always decreased so, in practice, we obtained a better system for our operating point. The best performing measure for both databases was the UBM log-likelihood. Actual cost improved by 77% in Agnitio benchmark and by 85% in Ahumada. The worse measures were the combination of *All* measures and NR_{0-1}. They were also the worse in the reliability detection experiments. That is coherent because the posterior of the *clean score* needed to compute the reliability is a GMM whose weights depend on the posterior of θ (see (6.21)). Thus, a bad estimation of the target posterior, or equivalently of LLR_Q, implies bad score and reliability posteriors.

6.8 Summary

In this chapter, we presented a novel Bayesian network whose purpose was to model how SV score distributions diverge from the ideal ones when the segments involved in the SV trials are affected by some distortion like noise or reverberation. Our BN introduced the existence of two scores: one observed and another hidden. The observed score or *noisy score* is the one given by our SV system while the hidden score or *clean score* is an ideal score that we would obtain if the trial segments were high quality speech. The network has another hidden variable, the *quality state* that means the type of trial distortion. Each value of the quality estate is associated with a distribution of quality measures and with a distribution for the difference Δs between the clean and noisy scores. The network allows us to compute the posterior distributions for the hidden variables given the observed variables. We proved that we can do it even with three hidden variables involved, e.g. trial label, quality states and clean score. We also explained how to estimate the parameters of the network by expectation maximization iterations.

Our network can be employed for two purposes. First, to compute a posterior probability for the reliability of the trial decision. That is done by computing the probability that the *clean score* is over or under the threshold, what is obtained by integrating the posterior distribution of the *clean score*. Trials with low reliability are rejected and, in that way, we can assure that the rest of trials have a low error rate. Secondly, we can compute an improved SV likelihood ratio given the observed score and the quality measures. Thus, we can also apply this network to improve performance in applications where we require classifying all the trials. Actual DCF considerably improved by using the improved likelihood ratio.

We experimented on NIST SRE augmented with noise and reverberation, Agnitio benchmark and Ahumada datasets. The network was trained on NIST SRE08 and tested on the rest of databases. We used the quality measures in Chapter 4 as input to the network. We took the best results achieved in Chapter 5 as baseline for the experiments consisting in rejecting unreliable trials. For most of the measures the newly proposed BN outperformed the baseline BN with the same measure. If we compare the best results achieved by both networks, we determine that, for NIST SRE10 + noise the new network clearly outperformed the baseline; for NIST SRE10 + reverberation both were alike; and for NIST with noise and reverberation the new one was slightly better. In general, the best measures for the three datasets were VTS NR_{0-1} and NR_{0-1}. Besides, the modulation index and N_{0-1} worked well for noise; R_{0-2} for reverberation; and the UBM log-likelihood for noise and reverberation. The experiments on Agnitio and Ahumada datasets demonstrated that the new network generalizes better than the baseline.

We explained a procedure to select the operating point of the reliability detection system based on minimizing the extended cost C_Q. C_Q assigns different costs for accepting badly classified trials and for rejecting well classified trials. By applying this method on NIST with noise and reverberation, we rejected 40% of trials, EER improved by 22% and actual DCF by 83%.

Regarding the experiments with the quality dependent likelihood ratio, all the quality measures yielded similar performance. For NIST datasets, we improved as much EER as minimum and actual DCF. The most appealing result was that we were able to reduce actual cost to values lower than one from initial cost as high as 4.5 without having to discard trials. Thus, we can take decisions better than chance. On Agnitio and Ahumada, it did not improve EER and minimum cost but it calibrated the scores and improved actual cost by 77–85%.

Chapter 7

Bayesian Adaptation of Reliability Models

7.1 Introduction

In the two previous chapters, we presented methods based on Bayesian networks to estimate the reliability of speaker verification decisions. We experimented training the parameters of the networks on artificial datasets where we added noise and/or reverberation to signals assumed clean. The networks trained in this manner were evaluated on artificial and real datasets achieving good performance. However, speech signals can be degraded due to many factors. If the causes that degrade the dataset under test are very different from the distortions present in the dataset used to train the BN the performance of the reliability detector may fall severely. To overcome this issue, we need to be able to adapt the reliability model to a new domain given a small amount of development data.

In this chapter, we continue working on the Bayesian network presented in Chapter 6. We will explain how to adapt this kind of BN, trained on an outer domain dataset, to the target domain by applying Bayesian methods. The Bayesian framework allows us to include prior information in the training process. In our case, that prior will be the BN trained on outer domain data. Bayesian methods are useful when the amount of adaptation data is limited in which case the maximum likelihood solution would provide inaccurate estimates.

The chapter is organized as follows. Section 7.2 explains the theory to adapt the network introduced in the previous chapter from one domain to another with scarce training data by MAP estimation. The equations that arise are very similar to those of the MAP adaptation of a GMM [Gauvain and Lee, 1994]. Section 7.3 presents experiments on the MOBIO dataset. We compared results obtained with different measures and adapting different parameters of the network. The obtained good results by re-calibrating scores for MOBIO and, then, adapting the distributions that describe the quality measures and the variability of the SV score. The VTS parameters were among the best measures. Finally, in Section 7.4 we summarize the chapter.

7.2 Bayesian Adaptation of the Bayesian Network that Models Score Variability

In the previous chapter, we introduced a Bayesian network that allowed us to model how the score of a SV trial varies from the ideal score when the enrollment or test segments are degraded. This BN defined a hypothetical *clean score* s as the SV score that we would obtain from our SV system if the speech segments involved in the trial had optimum quality. In general, the *clean score* is hidden. On the other hand, there was an observed score $\hat{s}$ that was the one given by the SV system. We also defined another hidden variable $\mathbf{z}$, called quality states, that represents the distortion types and/or levels that we could find. For each value of $\mathbf{z}$ we are going to observe a different distribution of the quality measures and a different amount of variation of $\hat{s}$ with respect to s. The BN is defined by the following conditional distributions:

$$P\left(\mathbf{s}|\theta\right) = \mathcal{N}\left(\mathbf{s}|\mu_{\mathbf{s}_\theta}, \mathbf{\Lambda}_{\mathbf{s}_\theta}^{-1}\right) \tag{7.1}$$

$$P\left(\hat{\mathbf{s}}|\mathbf{s}, z_k = 1, \theta\right) = \mathcal{N}\left(\hat{\mathbf{s}}|\mathbf{s} + \mu_{\Delta\mathbf{s}_{k\theta}}, \mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta}}^{-1}\right) \tag{7.2}$$

$$P\left(\mathbf{Q}|z_{ik} = 1\right) = \prod_{p=1}^{P} \mathcal{N}\left(\mathbf{Q}_p|\mu_{\mathbf{Q}_{pk}}, \mathbf{\Lambda}_{\mathbf{Q}_{pk}}^{-1}\right) \tag{7.3}$$

$$P\left(\mathbf{z}\right) = \prod_{k=1}^{K} \pi_{z_k}^{z_k} \ . \tag{7.4}$$

The parameters of these distributions can be estimated by maximum likelihood by applying the different versions of the EM algorithm explained in Section 6.6. If the amount of available data to train the model is scarce, the maximum likelihood solution may provide bad estimates. In this scenario, we can use Bayesian adaptation. The difference between ML and the Bayesian approach is that, in the former, the parameters of the model $\mathcal{M}$ are simply point estimates, while in the latter they are random variables with prior distributions. Priors are usually computed from a dataset with enough data to provide a good estimate but that dataset does not exactly represent the target domain data. Given some data from the target domain $\mathbf{X}$, we compute the posterior distribution of the model given $\mathbf{X}$ and the prior Π as

$$P\left(\mathcal{M}|\mathbf{X}, \Pi\right) = \frac{P\left(\mathbf{X}|\mathcal{M}\right) P\left(\mathcal{M}|\Pi\right)}{\int P\left(\mathbf{X}|\mathcal{M}'\right) P\left(\mathcal{M}'|\Pi\right) \, \mathrm{d}\mathcal{M}'} \ . \tag{7.5}$$

The fully Bayesian approach employs the full posterior to integrate out the parameters of the model when computing the probabilities of the test data given the different hypothesis. In this way, it takes into account the uncertainty about the values of the parameters and provides more accurate predictions. However, in most cases, we cannot apply this technique because integrals involved are computationally non-tractable. Instead, the common practice is to make a point estimate of the model parameters by taking the mode of their posterior distribution:

$$\mathcal{M}_{\mathrm{MAP}} = \arg\max_{\mathcal{M}} P\left(\mathcal{M}|\mathbf{X}, \Pi\right) \ . \tag{7.6}$$

This is called *maximum a posteriori* (MAP) estimation.

The first step of the Bayesian approach consists in choosing the form of the prior distributions. The usual choice is selecting *conjugate priors* because it simplifies the mathematics involved. A prior is conjugate for a likelihood function $P(\mathbf{X}|\mathcal{M})$ if the resulting posterior has the same functional form as the prior. For the means and precisions of the Gaussians $P(s|\theta)$, $P(\hat{s}|s,\mathbf{z})$ and $P(\mathbf{Q}|\mathbf{z})$, we assigned Gaussian-Wishart priors (see Appendix A):

$$\begin{aligned}
P\left(\mu_{\mathbf{s}_\theta}, \mathbf{\Lambda}_{\mathbf{s}_\theta}\right) =& \mathcal{N}\left(\mu_{\mathbf{s}_\theta}|\mu_{\mathbf{s}_{\theta_0}}, (\beta_0\mathbf{\Lambda}_{\mathbf{s}_\theta})^{-1}\right)\mathcal{W}\left(\mathbf{\Lambda}_{\mathbf{s}_\theta}|\mathbf{\Lambda}_{\mathbf{s}_{\theta_0}}/\nu_0, \nu_0\right) && (7.7)\\
P\left(\mu_{\Delta\mathbf{s}_{k\theta}}, \mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta}}\right) =& \mathcal{N}\left(\mu_{\Delta\mathbf{s}_{k\theta}}|\mu_{\Delta\mathbf{s}_{k\theta_0}}, (\beta_0\mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta}})^{-1}\right)\mathcal{W}\left(\mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta}}|\mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta_0}}/\nu_0, \nu_0\right) && (7.8)\\
P\left(\mu_{\mathbf{Q}_{pk}}, \mathbf{\Lambda}_{\mathbf{Q}_{pk}}\right) =& \mathcal{N}\left(\mu_{\mathbf{Q}_{pk}}|\mu_{\mathbf{Q}_{pk_0}}, (\beta_0\mathbf{\Lambda}_{\mathbf{Q}_{pk}})^{-1}\right)\mathcal{W}\left(\mathbf{\Lambda}_{\mathbf{Q}_{pk}}|\mathbf{\Lambda}_{\mathbf{Q}_{pk_0}}/\nu_0, \nu_0\right)\ . && (7.9)
\end{aligned}$$

Besides, we put a Dirichlet prior on the weights of the quality states:

$$P(\pi_{\mathbf{z}}) = \mathrm{Dir}(\pi_{\mathbf{z}}|\alpha_0\pi_{\mathbf{z}_0}) = C(\alpha_0)\prod_{k=1}^{K}\pi_k^{\alpha_0\pi_{z_{k_0}}-1} \qquad (7.10)$$

where $C(\alpha_0)$ is the normalization constant. The hyper-parameters $\mu_{\mathbf{s}_{\theta_0}}$, $\mu_{\Delta\mathbf{s}_{k\theta_0}}$, $\mu_{\mathbf{Q}_{pk_0}}$, $\mathbf{\Lambda}_{\mathbf{s}_{\theta_0}}$, $\mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta_0}}$, $\mathbf{\Lambda}_{\mathbf{Q}_{pk_0}}$ and $\pi_{\mathbf{z}_0}$ are the means, variances and weights respectively of the BN trained by ML on the outer domain dataset. The parameters α_0, β_0 and ν_0 are the relevance factors for the weights, means and covariances respectively. They represent the effective number of samples used to train each parameter of the prior. In practice, they are chosen manually by the operator.

The EM procedure used for ML estimation can be modified to obtain the modes of the posterior densities [Gauvain and Lee, 1994] by adding the log prior to the objective function:

$$\mathcal{Q}_{\mathrm{MAP}}(\mathcal{M}) = \mathcal{Q}_{\mathrm{ML}}(\mathcal{M}) + \ln P(\mathcal{M}|\Pi)\ . \qquad (7.11)$$

The new EM algorithm has an E-step that is the same as for ML while the updating equations of the M-step result similar to the MAP equations for a GMM. For the weights, we obtain:

$$\begin{aligned}
\alpha_k =& \alpha_0\pi_{z_{k_0}} + N_k && (7.12)\\
\pi_{z_k} =& \frac{\alpha_k - 1}{\sum_{k=1}^{K}\alpha_k - K} && (7.13)
\end{aligned}$$

where N_k is the count of trials in the inner domain dataset assigned to the quality state k during the E-step.

For the means and precisions of the *clean score*, the MAP updates are

$$\begin{aligned}
\beta_\theta =& N_\theta + \beta_0 && (7.14)\\
\nu_\theta =& N_\theta + \nu_0 && (7.15)\\
\mu_{\mathbf{s}_\theta} =& \frac{1}{\beta_\theta}\left(\beta_0\mu_{\mathbf{s}_{\theta_0}} + N_\theta\mu_{\mathbf{s}_{\theta_{\mathrm{ML}}}}\right) && (7.16)\\
\mathbf{\Lambda}_{\mathbf{s}_\theta}^{-1} =& \frac{1}{\nu_\theta - d_{\mathbf{s}} - 1}\left(\nu_0\mathbf{\Lambda}_{\mathbf{s}_{\theta_0}}^{-1} + N_\theta\mathbf{\Lambda}_{\mathbf{s}_{\theta_{\mathrm{ML}}}}^{-1} + \frac{\beta_0 N_\theta}{\beta_\theta}\left(\mu_{\mathbf{s}_{\theta_{\mathrm{ML}}}} - \mu_{\mathbf{s}_{\theta_0}}\right)\left(\mu_{\mathbf{s}_{\theta_{\mathrm{ML}}}} - \mu_{\mathbf{s}_{\theta_0}}\right)^T\right) && (7.17)
\end{aligned}$$

for $\theta \in \{\mathcal{T},\mathcal{N}\}$; $d_{\mathbf{s}}$ is the dimension of the score vector that in our experiments is always 1; N_θ is the number of adaptation trials for the target and non-target hypothesis and; $\mu_{\mathbf{s}_{\theta_{\text{ML}}}}$ and $\mathbf{\Lambda}_{\mathbf{s}_{\theta_{\text{ML}}}}$ are the ML means and precisions obtained with the updating equations listed in Section 6.6.

Similarly, the equations for the means and precisions of the Δs distributions are

$$\beta_{k\theta} = N_{k\theta} + \beta_0 \tag{7.18}$$

$$\nu_{k\theta} = N_{k\theta} + \nu_0 \tag{7.19}$$

$$\mu_{\Delta\mathbf{s}_{k\theta}} = \frac{1}{\beta_{k\theta}}\left(\beta_0\mu_{\Delta\mathbf{s}_{k\theta_0}} + N_{k\theta}\mu_{\Delta\mathbf{s}_{k\theta_{\text{ML}}}}\right) \tag{7.20}$$

$$\begin{aligned}\mathbf{\Lambda}^{-1}_{\Delta\mathbf{s}_{k\theta}} =& \frac{1}{\nu_{k\theta} - d_{\mathbf{s}} - 1}\left(\nu_0\mathbf{\Lambda}^{-1}_{\Delta\mathbf{s}_{k\theta_0}} + N_{k\theta}\mathbf{\Lambda}^{-1}_{\Delta\mathbf{s}_{k\theta_{\text{ML}}}}\right.\\ &\left.+\frac{\beta_0 N_{k\theta}}{\beta_{k\theta}}\left(\mu_{\Delta\mathbf{s}_{k\theta_{\text{ML}}}} - \mu_{\Delta\mathbf{s}_{k\theta_0}}\right)\left(\mu_{\Delta\mathbf{s}_{k\theta_{\text{ML}}}} - \mu_{\Delta\mathbf{s}_{k\theta_0}}\right)^T\right)\end{aligned} \tag{7.21}$$

for $\theta \in \{\mathcal{T},\mathcal{N}\}$ and $k = 1,\ldots,K$. $N_{k\theta}$ is the number of adaptation trials for the target and non-target hypothesis assigned to the state k and; $\mu_{\Delta\mathbf{s}_{k\theta_{\text{ML}}}}$ and $\mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta_{\text{ML}}}}$ are the ML point estimates.

Finally, the means and precisions of the quality measures distributions are

$$\beta_k = N_k + \beta_0 \tag{7.22}$$

$$\nu_k = N_k + \nu_0 \tag{7.23}$$

$$\mu_{\mathbf{Q}_{pk}} = \frac{1}{\beta_k}\left(\beta_0\mu_{\mathbf{Q}_{pk_0}} + N_k\mu_{\mathbf{Q}_{pk_{\text{ML}}}}\right) \tag{7.24}$$

$$\mathbf{\Lambda}^{-1}_{\mathbf{Q}_{pk}} = \frac{1}{\nu_k - d_{\mathbf{Q}_p} - 1}\left(\nu_0\mathbf{\Lambda}^{-1}_{\mathbf{Q}_{pk_0}} + N_k\mathbf{\Lambda}^{-1}_{\mathbf{Q}_{pk_{\text{ML}}}} + \frac{\beta_0 N_k}{\beta_k}\left(\mu_{\mathbf{Q}_{pk_{\text{ML}}}} - \mu_{\mathbf{Q}_{pk_0}}\right)\left(\mu_{\mathbf{Q}_{pk_{\text{ML}}}} - \mu_{\mathbf{Q}_{pk_0}}\right)^T\right) \tag{7.25}$$

for $p = 1,\ldots,P$ and $k = 1,\ldots,K$; $d_{\mathbf{Q}_p}$ is the dimension of the measure vector $\mathbf{Q}_p$ and; $\mu_{\mathbf{Q}_{pk_{\text{ML}}}}$ and $\mathbf{\Lambda}_{\mathbf{Q}_{pk_{\text{ML}}}}$ are the ML point estimates.

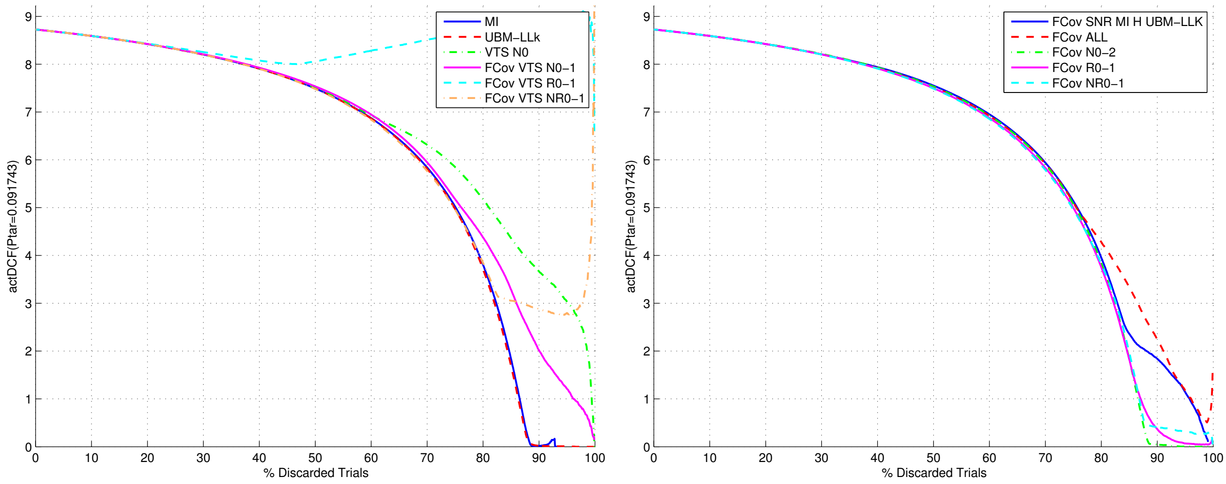
Note, that all the updating equations do, approximately, a weighted sum of the ML and the prior parameters. As the number of adapting trials increases, the parameters value approaches asymptotically to the ML solution.

7.3 Experiments

We experimented on the MOBIO database described in Section 3.4.2.6. The MOBIO dataset counts with a development trial list to do things like calibration and an evaluation list. Our prior BN was trained on NIST SRE08 with noise and reverberation with the version of the EM algorithm that takes the *clean score* s and the quality states $\mathbf{z}$ hidden. We used the MOBIO development for MAP adaptation of the BN and re-calibration and; the evaluation to validate our approach.

7.3.1 Reliability detection

Figure 7.1 plots actual costs against the percentage of discarded trials for the prior BN. This is the baseline of our experiments. We show results for the quality measures that



(a) Single measure.

(b) Multiple measures.

Figure 7.1: % Discarded trials vs. actual DCF for MOBIO for a BN trained on NIST SRE08 + noise and reverberation.

performed better on NIST SRE10. We remind the notation used for the quality measures. Signal-to-noise ratio is denoted by SNR, modulation index by MI, entropy by H and UBM log-likelihood by UBM-LLk. The term VTS N refers to the VTS coefficients with LDA trained to discriminate between different levels of SNR and, VTS R is the same but with LDA trained to discriminate between reverberation times. The numbers behind N or R indicate the output dimension of the LDA projection, for example N_{0-1} means that we keep the first two coefficients. When we write VTS NR, we mean that we concatenate the both

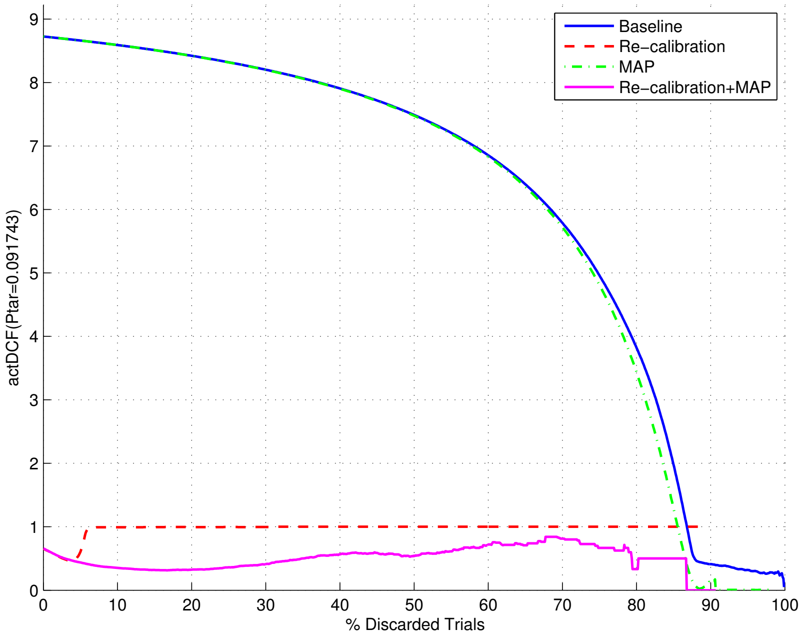


Figure 7.2: % Discarded trials vs. actual DCF for MOBIO, comparison of adaptation methods.

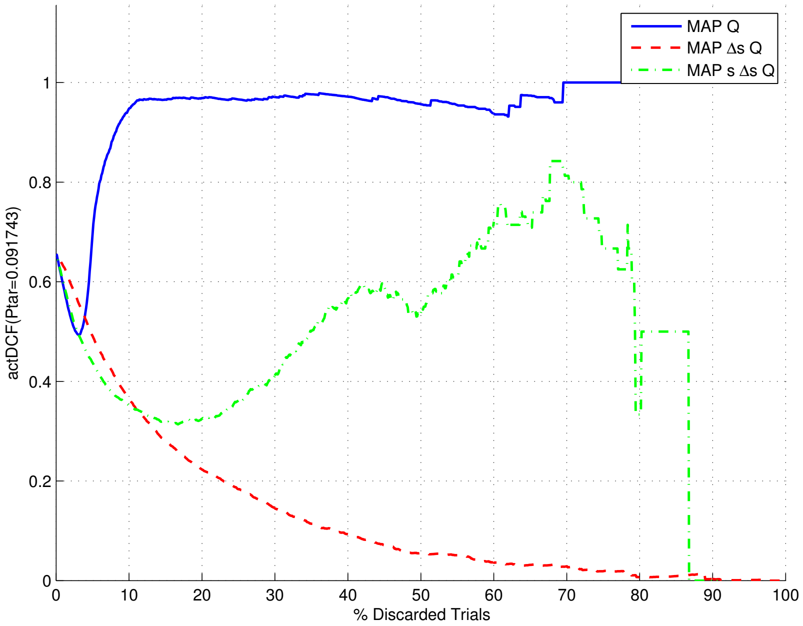


Figure 7.3: % Discarded trials vs. actual DCF for MOBIO, comparison between adapting the BN conditional distributions for $\mathbf{Q}$, $\Delta s + \mathbf{Q}$ or $s + \Delta s + \mathbf{Q}$.

LDA projections, the one for noise and the one for reverberation. When we say that we use *All* the measures we mean: SNR, MI, H, UBM-LLk, VTS NR_0, jitter and shimmer. In the same way as for the VTS parameters, the terms N, R and NR refer to LDA projections of *All* the quality measures to discriminate noise or/and reverberation times. We denote by *FCov* the networks where the distribution of the quality measures conditioned to each quality state $P(\mathbf{Q}|z_k = 1)$ for $k = 1, \dots, K$ is a full-covariance Gaussian. Otherwise, it is a product of full-covariance bi-dimensional Gaussians (equivalent to block diagonal Gaussian). We had one of these bi-dimensional Gaussians for each quality measure employed by the network. The first dimension of the Gaussian corresponded to the measure of the enrollment segment, and the second one corresponded to the measure of the test segment.

The actual cost starts being 8.71 and it decreases slowly as we discard the trials with lower estimated reliability. Once we discard 70% of the trials, the cost starts decreasing faster. We need to remove 87% to achieve a cost lower than one–with the measures MI, UBM-LLk, N_{0-2}, R_{0-2} and NR_{0-1}. If continue rejecting until 89%, we reach a cost under 0.1 with the measures MI, UBM-LLk and N_{0-2}. Despite that we need to discard 90% of trials to achieve good costs, the model performed well in the sense that the cost always decreased as we discard trials. If we compare these curves with the ones obtained in Section 6.7.4 for Agnitio benchmark and Ahumada, we observe that, in those datasets, we obtained a faster cost reduction (close to zero by just rejecting 30% of the trials). This could mean two things. On the one hand, we could think that our model works properly and that most of the trials of MOBIO are distorted. On the other hand, we could assume that the properties of MOBIO are different than the properties of NIST SRE and that we need to adapt of BN to this dataset. Following, we prove that we can achieve much better performance by adapting the BN.

We considered three adaptation possibilities. First, we re-calibrated the scores by logistic regression on the development set and evaluated the reliability with the BN without

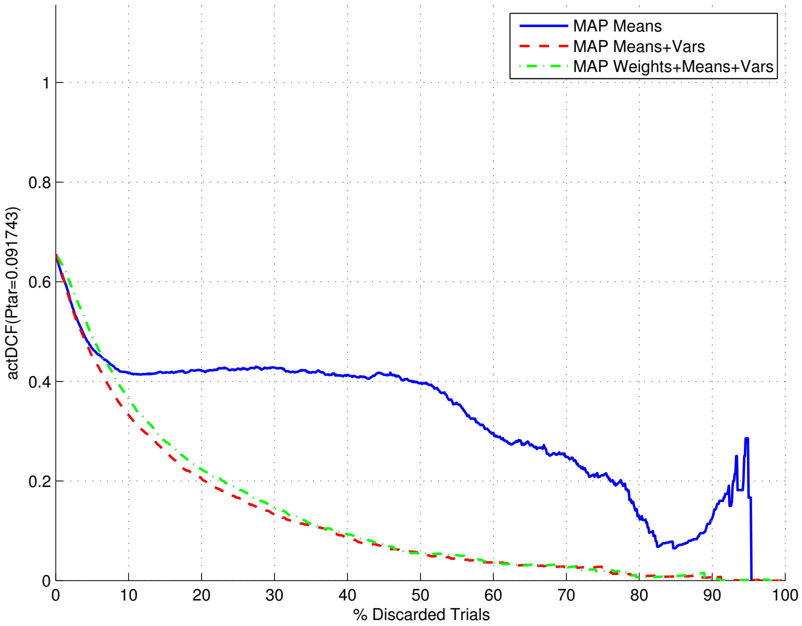
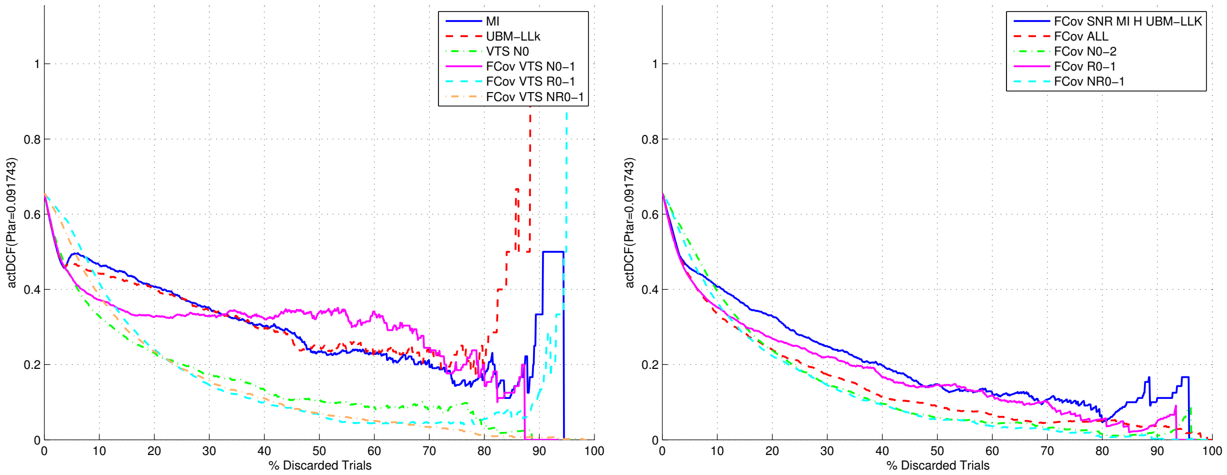


Figure 7.4: % Discarded trials vs. actual DCF for MOBIO, comparison between adapting mean, mean+variances or mean+variances+weights in the BN distributions.

adaptation. Second, we adapted the BN to MOBIO without score re-calibration. Finally, we re-calibrated the development and evaluation scores on the development set and adapted the BN. We adapted all the parameters of the BN with $\alpha_k = \beta_k = \nu_k = 25$. These three options are compared in Figure 7.2 using the measure NR_{0-1}. By re-calibrating we improved the actual cost from 8.71 to 0.61; however the BN without adaptation did not work well on the re-calibrated scores making the cost to grow until one if we reject trials. If we only adapt the BN, the curve is almost the same as without adaptation. The adapted curve is slightly under the baseline from the rejection rate of 70%. To reduce the cost while rejecting a small percentage of trials we needed to re-calibrate and adapt the BN at the same time. The lowest cost achieved in this manner was 0.31 by discarding 15% of the trials. The fact that re-calibration were needed to obtain a fast cost reduction seems to indicate that the clean score distributions for NIST SRE and MOBIO are very different. The rest of experiments that follow use both re-calibration and adaptation.

In Figure 7.3, we consider the possibility of adapting only some of the conditional distributions of the BN. First, we only adapted the parameters of the distribution of quality measures $P\left(\mathbf{Q}|\mathbf{z}\right)$ (denoted by Q); second, the distributions of quality measures and the distributions of the observed score given the *clean score* and the quality states $P\left(\hat{s}|s,\mathbf{z}\right)$ (denoted by Δs, Q); and third, all the distributions (denoted by s, Δs, Q). The best results were obtained by adapting everything but the *clean score* distribution. That means that the estimation of s that we obtained during training is not good enough to be employed to adapt the network. However, the curve corresponding to the second case is very good showing a monotonous reduction of the cost. In the following experiments, we only adapt the $\mathbf{Q}$ and Δs distributions.

We also compared the possibility of adapting different parameters of the distributions: only means; means and variances; and mean, variances and the weights of the quality states $\pi_{\mathbf{z}}$. The results are shown in Figure 7.4. By adapting the means, we obtained a



(a) Single measure.

(b) Multiple measure.

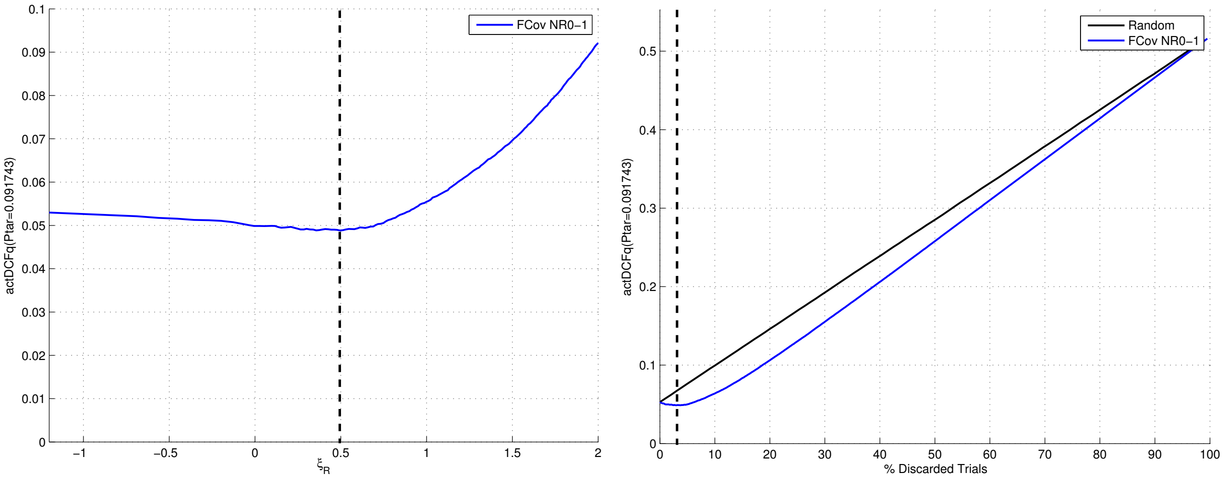
Figure 7.5: % Discarded trials vs. actual DCF for MOBIO with a BN MAP adapted and several quality measures.

good improvement compared to non-adapting but we need to adapt mean and variances to achieve the best performance. There was not a significant difference between also adapting the weights and not doing it.

Finally, Figure 7.5 and Table 7.5 compare different quality measures on a BN with score re-calibration and MAP adaption of the $\mathbf{Q}$ and Δs distributions. Figure 7.5a displays costs for BNs that use only one quality measure and Figure 7.5b for BNs that combine multiple measures. The best performing measures were VTS NR_{0-1}, the combination of *All* measures, N_{0-2} and NR_{0-1}. These results prove that our reliability models can be easily adapted to new databases with different characteristics.

Table 7.1: % Discarded trials vs. actual DCF for MOBIO with a BN MAP adapted and several quality measures.

Q	% Discarded Trials		
	25	50	75
MI	0.37	0.23	0.15
UBM-LLk	0.38	0.25	0.23
VTS N_0	0.20	0.11	0.09
FCov VTS N_{0-1}	0.33	0.34	0.19
FCov VTS R_{0-1}	**0.19**	**0.07**	0.05
FCov VTS NR_{0-1}	**0.19**	**0.07**	**0.03**
FCov SNR MI H UBM-LLK	0.28	0.15	0.10
FCov ALL	0.21	0.09	0.05
FCov N_{0-2}	0.19	**0.06**	0.03
FCov R_{0-1}	0.25	0.15	0.07
FCov NR_{0-1}	**0.18**	**0.06**	**0.02**
No Adapt.	8.32	7.50	4.97



(a) Reliability threshold ξ_R vs. C_Q.

(b) % Discarded trials vs. C_Q

Figure 7.6: C_Q for the MOBIO development dataset.

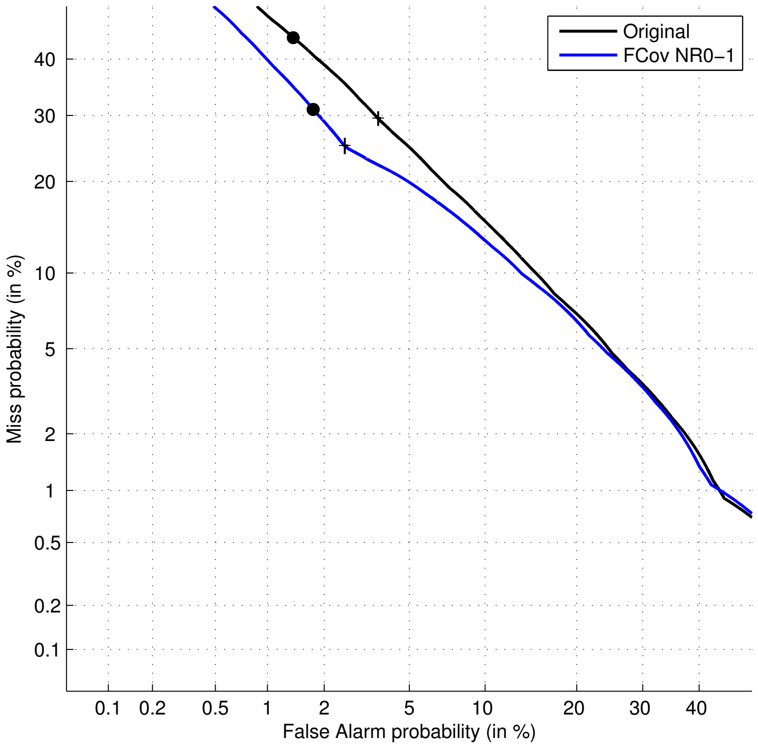


Figure 7.7: DET curve for MOBIO Eval in operating point given by minimum C_Q in MOBIO Dev.

Table 7.2: EER(%)/DCF of quality dependent LLR on MOBIO with different adaptation methods.

Q	EER(%)	minDCF	actDCF
Standard LLR			
Baseline	**12.37**	**0.57**	8.72
Baseline Re-cal	**12.37**	0.57	**0.66**
LLR(Q)			
No adapt.	19.21	0.92	1.11
Re-cal	13.95	0.63	0.97
MAP	13.27	0.59	0.62
Re-cal+MAP	**12.94**	**0.59**	**0.59**

As we did in the previous chapter, we can use the extended cost function C_Q to select the operating point of the reliability detector. We applied the reliability detector with measure NR_{0-1} to the development trials of MOBIO and plotted the C_Q curves in Figure 7.6. We chose the threshold that minimized C_Q ($\xi_R = 0.5$). By applying that threshold on the evaluation set, we kept 95% of the trials, which performed with EER=11.58% and actual DCF=0.5. Compared to keeping all the trails without re-calibration, EER improves by 4.2% and actual cost by 94.2%. Compared to using re-calibrated scores the actual cost improves by 18%. Figure 7.7 shows the DET curve that we obtain with the accepted trials. We see a larger improvement in the low false alarm region, the same as we saw for NIST SRE10.

7.3.2 Quality dependent likelihood ratio

In this section, instead of using the BN to reject unreliable trials, we use it to compute the quality dependent SV likelihood ratio explained in Section 6.5. Afterwards, error rates are computed on the new ratio. We tried different adaptation options whose performance appears in Table 7.2. The first block corresponds to the case of using the likelihood ratio from the standard verification system without using the BN. The Baseline correspond to the SV ratio calibrated on NIST SRE08 clean data and the second line (Re-cal) corresponds to the ratio calibrated on the development part of MOBIO. We observe that just re-calibrating we were able to reduce most of the gap between minimum and actual costs. On the other hand, the second block was obtained by using the quality dependent ratio. We used the measure NR_{0-1}, which obtained the best result in the reliability detection experiments. The first two lines of the block use the BN without adaptation, the first one on scores calibrated on NIST and the second one on the scores re-calibrated on MOBIO. Both reduced the actual cost with regard to the baseline but they were worse than just re-calibrating. Besides, the EER and minimum cost were worse than the baseline. The third line (MAP) used the network adapted to MOBIO without re-calibrating the scores. We adapted all the parameters of the network. In this case, the EER and minimum cost were slightly worse than the baseline but the actual cost improved and it was a little bit better than for the re-calibrated standard score. Finally, the fourth line (Re-cal+MAP) uses a BN adapted to MOBIO where the scores were previously re-calibrated. In this last case, we adapted all the parameters of the network but the means and variances of the prior *clean score* distributions $P(s|\theta)$. This option provides the lowest actual cost being, for all the fields, slightly better than adapting the BN without re-calibrating. The actual DCF improves by 93% compared to the baseline and by 10% compared to the baseline with re-calibration. With these results, we can say that, for this database, just re-calibrating scores is enough and that the extra gain that provides the BN may not be worthy compared to the complexity of the approach.

7.4 Summary

This chapter addressed the problem of adapting a reliability detection Bayesian network from one domain with a large amount of development data to another with scarce development data. We considered this problem because the degradations to which the signals are subjected can be different in each dataset. We decided to apply a Bayesian approach. The BN trained on the large dataset, e.g., NIST SRE with noise and reverberation, was the model *a priori*. Then, we computed the posterior distribution of the model given the adaptation data and the prior. The mode of the posterior was the adapted BN.

We experimented on the MOBIO dataset, which has two distinct trials lists: one for adaptation and/or calibration; and another for evaluation. Regarding the reliability detection experiments, to reduce the cost while rejecting a low number of trials, we needed to re-calibrate the scores before adapting the network. We compared the cost curves obtained by adapting different parameters of the networks. The Best results were achieved by adapting the means and variances of the quality and Δs conditional distributions while leaving the *clean score* s distributions fixed. As in the previous chapter, the measures VTS NR_{0-1} and NR_{0-1} were among the best performers.

Regarding the experiments with the quality dependent likelihood ratio, we attained a small improvement of actual DCF (10 %) compared to just re-calibrating the scores by logistic regression.

Part III

PLDA for Non-Colaborative Environments

Chapter 8

Handling Recordings Acquired in Multiple Environments with PLDA

8.1 Introduction

As we explained in Chapter 2, recent advancements, such as JFA and i-vectors, have allowed speaker verification systems to attain great performance. Error rates are especially low in conditions where recordings are acquired in clean environments and using always the same types of microphones and transmission channels. This is the scenario presented in NIST evaluations until SRE 2010. However, we evidenced in the Part II of this thesis that i-vectors are still quite vulnerable to noise and reverberation. The problem aggravates when we have trials that involve i-vectors recorded in very different channel conditions. For example, we can find situations like enrollment on clean speech and test on low signal-to-noise ratio speech; enrollment on telephone speech and test on a far-field microphone recording; mixed enrollment on several segments, each one of them captured from a different source; etc. In this chapter, we address the problem of how to model the i-vector distribution in this kind of scenarios.

The problem of multi-channel speaker recognition has been addressed before. The works in [Senoussaoui et al., 2010, Dehak et al., 2011a, Senoussaoui et al., 2011b, McLaren and Leeuwen, 2011a,McLaren and Leeuwen, 2011b,McLaren and Leeuwen, 2012], present similar approaches. Some kind of linear discriminant analysis (LDA) is applied to telephone and far-field microphone i-vectors to project them into a common space. Then, i-vectors are classified with cosine distance or probabilistic linear discriminant analysis (PLDA). The main difference between approaches is the method used to estimate the LDA projection matrix. In [Senoussaoui et al., 2010], two strategies were compared: weighting and pooling. Weighting is based on computing separate between and within class covariance matrices for telephone and microphone data. The weighted averages of the microphone and telephone matrices are used to compute the LDA projection. Pooling consists in training the LDA matrices on all the microphone and telephone data together. Weighting yielded the best results. In [Dehak et al., 2011a], authors project i-vectors by computing the speaker factors vector of a PLDA where the covariance of the residual term is only trained on telephone data and the eigen-channel matrix is trained on microphone data in such way that it captures the variability of the microphone data that is not already included in the telephone data. In [Senoussaoui et al., 2011b], i-vectors were projected with a heavy tail PLDA trained on pooled telephone and microphone. In [McLaren and Leeuwen, 2011a,McLaren and Leeuwen, 2011b, McLaren and Leeuwen, 2012], several ways of estimating and averaging the between and within class covariance matrices were studied.

A different approach was adopted in [Lei et al., 2012] where standard PLDA is trained using pooled clean and noisy data. In [Simonchik et al., 2012], authors trained three conditioned PLDA models (telephone, microphone and telephone+microphone). In the classification phase, the models are treated as components of a mixture of PLDA and Bayesian fusion of scores is implemented. In [Garcia-Romero et al., 2012], several PLDA variants are explored (condition dependent, pooled PLDA, tied PLDA) and the scores fused.

The standard PLDA model describes the inter-session variability between the i-vectors of a given speaker by a unique within-class covariance matrix. Intuition tells us that, as session variability is very dependent on the channel conditions, we should use different within-class matrices for each channel. Intending to approach the problem in a principled way, we tried a variant of Prince's tied PLDA [Prince and Elder, 2007] where i-vectors are modeled with a common between-class covariance but with a different within-class covariance depending on their channel type. This model can also be seen as a mixture of PLDA models where the speaker term is tied to be the same across the components. This framework allows pooling all the data available to estimate the PLDA parameters in such a way that the speaker space is estimated with all the data and the channel spaces are estimated only with the data of their corresponding channel.

This chapter is organized as follows: Section 8.2 describes the standard PLDA model, including the way of computing the latent variable posteriors and the multiple flavors of trial evaluation. Section 8.3 describes the proposed multi-channel PLDA framework. This covers computing the latent factors posteriors, EM algorithm form model training, multi-channel version of centering and whitening as previous step to length normalization, and trial evaluation variants. Section 8.4 describes our experimental setup including datasets, performance evaluation and system configuration. We experimented on NIST SRE12–dataset that includes different types of noise. The PLDA models were trained on NIST SRE04-10 augmented with additive noise. Section 8.5 shows results comparing different classifiers, different types of enrollment lists and trial evaluation methods. It also compares results with different noise types and levels. Finally, Section 8.6 summarizes the chapter.

8.2 Simplified PLDA

8.2.1 Model description

Probabilistic linear discriminant analysis (PLDA) [Prince and Elder, 2007] is a generative model that decomposes i-vectors into a speaker dependent term and channel dependent term. Thus, an i-vector ϕ_{ij} corresponding to the j^{th} recording of a speaker i can be written as

$$\phi_{ij} = \mu + \mathbf{V}\mathbf{y}_i + \mathbf{U}\mathbf{x}_{ij} + \epsilon_{ij} \tag{8.1}$$

where μ is a speaker independent term, $\mathbf{V}$ is a low-rank matrix of eigen-voices, $\mathbf{y}_i$ is the speaker factors vector, $\mathbf{U}$ is a low-rank eigen-channels matrix, $\mathbf{x}_{ij}$ is the channel factors vector and ϵ_{ij} is another channel offset accounting for the residual variability not included in $\mathbf{U}$. It is understood that the term *channel* is a synecdoche, representing the causes that make a speaker's recordings vary from one occasion to the next, rather than just physical transmission and recording channels. Variables $\mathbf{y}_i$, $\mathbf{x}_{ij}$ and ϵ_{ij} are hidden with priors:

$$\mathbf{y}_i \sim \mathcal{N}\left(\mathbf{y}_i|\mathbf{0},\mathbf{I}\right) \tag{8.2}$$

$$\mathbf{x}_{ij} \sim \mathcal{N}\left(\mathbf{x}_{ij}|\mathbf{0},\mathbf{I}\right) \tag{8.3}$$

$$\epsilon_{ij} \sim \mathcal{N}\left(\epsilon_{ij}|\mathbf{0},\mathbf{D}^{-1}\right) \tag{8.4}$$

where $\mathcal{N}$ denotes the Gaussian distribution; and $\mathbf{D}$ is a diagonal precision matrix.

The simplified version of PLDA (SPLDA) disregards the eigen-channel term by simply writing the i-vector as:

$$\phi_{ij} = \mu + \mathbf{V}\mathbf{y}_i + \epsilon_{ij} \tag{8.5}$$

where the prior for ϵ_{ij} is Gaussian

$$\epsilon_{ij} \sim \mathcal{N}\left(\epsilon_{ij}|\mathbf{0},\mathbf{W}^{-1}\right) \tag{8.6}$$

with full-precision matrix $\mathbf{W}$. If $\mathbf{U}$ is full-rank, simplified PLDA is equivalent to the full PLDA model where $\mathbf{W}^{-1} = \mathbf{U}\mathbf{U}^T + \mathbf{D}^{-1}$.

The parameters μ, $\mathbf{V}$ and $\mathbf{W}$ are trained on a development database by maximizing the data likelihood with expectation maximization (EM) iterations. Derivations for the EM equations can be found in Appendix C. We denote by $\mathcal{M}$ the set of all the model parameters.

8.2.2 Posterior of the hidden variables

The posterior of the hidden variables given the observed data is an useful distribution to train the model as well as to evaluate the trials. For this model, the posterior of the speaker variables $\mathbf{y}_i$ given the i-vectors $\mathbf{\Phi}_i$ of speaker i is a Gaussian distribution given by

$$P\left(\mathbf{y}_i|\mathbf{\Phi}_i,\mathcal{M}\right) = \mathcal{N}\left(\mathbf{y}_i|\mathbf{L}_i^{-1}\gamma_i,\mathbf{L}_i^{-1}\right) \tag{8.7}$$

where

$$\mathbf{L}_i = \mathbf{I} + N_i\mathbf{V}^T\mathbf{W}\mathbf{V} \tag{8.8}$$

$$\gamma_i = \mathbf{V}^T\mathbf{W}\overline{\mathbf{F}}_i\ ; \tag{8.9}$$

N_i are the number of segments of speaker i; and $\overline{\mathbf{F}}_i$ are the first order statistics of speaker i centered in μ:

$$\overline{\mathbf{F}}_i = \sum_{j=1}^{N_i} \phi_{ij} - \mu\ . \tag{8.10}$$

8.2.3 Trial evaluation

Given a set of enrollment i-vectors $\mathbf{\Phi}_{\text{enroll}} = \{\phi_1,\dots,\phi_N\}$ from a known speaker and a test i-vector ϕ_{tst} from an unknown speaker, trial evaluation consists in computing a likelihood ratio between the hypothesis that $\mathbf{\Phi}_{\text{enroll}}$ and ϕ_{tst} were uttered by the same speaker–target hypothesis $\mathcal{T}$–, and the hypothesis that they were uttered by different speakers–non-target hypothesis $\mathcal{N}$. In the PLDA context, the target hypothesis implies that all the i-vectors in $\mathbf{\Phi}_{\text{enroll}}$ and ϕ_{tst} share the same value of the speaker factor $\mathbf{y}$ while the non-target hypothesis implies that they do not. Thus, the likelihood ratio is computed as:

$$R\left(\mathbf{\Phi}_{\text{enroll}},\phi_{\text{tst}}\right)=\frac{P\left(\mathbf{\Phi}_{\text{enroll}},\phi_{\text{tst}}|\mathcal{T}\right)}{P\left(\mathbf{\Phi}_{\text{enroll}},\phi_{\text{tst}}|\mathcal{N}\right)}=\frac{\int P\left(\mathbf{\Phi}_{\text{enroll}},\phi_{\text{tst}}|\mathbf{y}\right)P\left(\mathbf{y}\right)\,\mathrm{d}\mathbf{y}}{\int P\left(\mathbf{\Phi}_{\text{enroll}}|\mathbf{y}_1\right)P\left(\mathbf{y}_1\right)\,\mathrm{d}\mathbf{y}_1\int P\left(\phi_{\text{tst}}|\mathbf{y}_2\right)P\left(\mathbf{y}_2\right)\,\mathrm{d}\mathbf{y}_2}\,. \tag{8.11}$$

We can avoid solving the integrals by manipulating the ratio as shown in [Brummer and De Villiers, 2010] where it is written as a function of the $\mathbf{y}$ posteriors:

$$R\left(\mathbf{\Phi}_{\text{enroll}},\phi_{\text{tst}}\right)=\left.\frac{P\left(\mathbf{y}|\mathbf{\Phi}_{\text{enroll}},\mathcal{M}\right)P\left(\mathbf{y}|\phi_{\text{tst}},\mathcal{M}\right)}{P\left(\mathbf{y}|\mathbf{\Phi}_{\text{enroll}},\phi_{\text{tst}},\mathcal{M}\right)P\left(\mathbf{y}\right)}\right|_{\mathbf{y}=\mathbf{y}_0}\,. \tag{8.12}$$

Note that the left hand side of the equation does not depend on $\mathbf{y}$ so neither does the right hand side, even though $\mathbf{y}$ appears explicitly. That means that, eventually, $\mathbf{y}$ will simplify so we can evaluate the ratio for whatever value $\mathbf{y}_0$ that we find convenient, for example, $\mathbf{y}_0=\mathbf{0}$.

The above equations are the more accurate from the point of view of the probability theory. This form of evaluation is popularly known among the community as *by the book* evaluation. However, results evidence that, when there are more than one enrollment segment, it does not provide the lowest error rates. It has been observed that trials with different number of enrollment segments exhibit different ranges of target and non-target scores. This miss-alignment causes a performance loss. Another issue is that, when we have many enrollment segments, we may be underestimating the covariance of the $\mathbf{y}$ posterior. That is, we are too over-confident about the value of the speaker factor. Equation (8.8) shows that the covariance of the posterior is smaller as the number of speaker samples increases. However, we obtain this result because the model assumes that the channel offsets of a given speaker are independent and identically distributed (i.i.d.), which is not true in practice. The inter-session variability term can depend on the speaker in many ways. For example, it depends on the telephone handsets that he owns. This fact implies that, each new i-vector from a given speaker adds less new information than the previous ones so it should not be counted as one new unit of information but less. Thus, the value of N_i in (8.8) should be lower than the actual number of i-vectors or speaker i and, thus, the covariance would be larger.

The method of i-vector averaging proposes to replace the enrollment i-vectors with an unique i-vector computed as the average of them:

$$R_{\text{iv}-\text{avg}}\left(\mathbf{\Phi}_{\text{enroll}},\phi_{\text{tst}}\right)=\frac{P\left(\frac{1}{N}\sum_{i=1}^{N}\phi_i,\phi_{\text{tst}}|\mathcal{T}\right)}{P\left(\frac{1}{N}\sum_{i=1}^{N}\phi_i,\phi_{\text{tst}}|\mathcal{N}\right)}\,. \tag{8.13}$$

This method does not suffer from the problems above described. Besides, i-vector averaging provided the lowest error rates in our experiments as we will show in Section 8.5.

Another method that attains a performance close to i-vector averaging is scoring averaging. The method consists in computing the log-likelihood ratios between each enrollment i-vector and the test i-vector and, afterward, the log-ratios are averaged:

$$\ln R_{\text{s}-\text{avg}}\left(\mathbf{\Phi}_{\text{enroll}},\phi_{\text{tst}}\right)=\frac{1}{N}\sum_{i=1}^{N}\ln\frac{P\left(\phi_i,\phi_{\text{tst}}|\mathcal{T}\right)}{P\left(\phi_i,\phi_{\text{tst}}|\mathcal{N}\right)}\,. \tag{8.14}$$

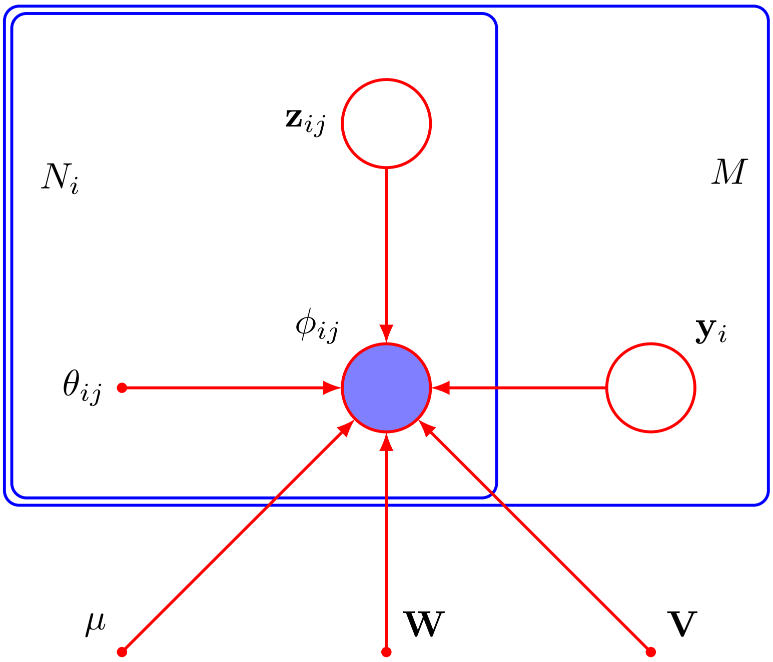


Figure 8.1: BN for multichannel SPLDA model.

This method has higher computational cost than i-vector averaging because it involves the calculus of many more likelihood ratios. However, in our experiments, it did not provide better performance.

8.3 Multi-Channel SPLDA

8.3.1 Model description

SPLDA deals with all types of inter-session variability with an unique within-class covariance. Intuition suggests that i-vector variability will be very different as a function of the type of channel (telephone, far-field microphone) or the noise type and level. In this section, we modify SPLDA to take into account the fact that each i-vector can be generated in different channel conditions. The new model, which we called multi-channel SPLDA (MCSPLDA), decompose the i-vector ϕ_{ij}, in the same manner than standard SPLDA:

$$\phi_{ij} = \mathbf{V}\mathbf{y}_i + \epsilon_{ij} \ , \tag{8.15}$$

but, this time, the distribution of the channel offset depends on the type of channel k where the i-vector is generated:

$$\epsilon_{ij}|z_{ijk} = 1 \sim \mathcal{N}\left(\epsilon_{ij}|\mu_k, \mathbf{W}_k^{-1}\right) \tag{8.16}$$

where μ_k is a channel dependent bias, and $\mathbf{W}_k$ is the channel dependent within-class precision matrix. We also define the sets $\mu = \{\mu_k\}_{k=1}^K$ and $\mathbf{W} = \{\mathbf{W}_k\}_{k=1}^K$. The variable $\mathbf{z}_{ij}$ indicates the type of channel of ϕ_{ij}. If there are K channel types, $\mathbf{z}_{ij}$ is a 1-of-K binary vector with elements z_{ijk} for $k = 1, \dots, K$, where $z_{ijk} = 1$ if ϕ_{ij} has been generated in channel k and $z_{ijk} = 0$ otherwise. For simplicity, we assumed that we have some kind of channel detector that provides the value of $\mathbf{z}_{ij}$, or at least a probability $P\left(z_{ijk} = 1\right)$ for it.

Figure 8.1 depicts the Bayesian network that describes this model. There are M speakers with N_i i-vectors per speaker. ϕ_{ij} are observed variables while $\mathbf{y}_i$ and $\mathbf{z}_{ij}$ are hidden; θ_{ij} are the speaker labels assumed known in the training phase; and μ, $\mathbf{W}$ and $\mathbf{V}$ are deterministic parameters derived by maximum likelihood. Additionally, we denote by $\mathbf{\Phi}_i$ the set of i-vectors of speaker i; and by $\mathbf{z}_i = \{\mathbf{z}_{i1}, \dots, \mathbf{z}_{iN_i}\}$ the channel assignments for i. The set of all the model parameters is again denoted by $\mathcal{M}$.

Note that MCSPLDA is equivalent to a mixture of PLDA models where $\mathbf{V}$ and $\mathbf{y}$ are tied across the components of the mixture. This model maintains the speaker space $\mathbf{V}$ independent of the channel; since speakers are human beings their voices should be the same in every recording environment. Furthermore, the model forces the speaker variable $\mathbf{y}_i$ to be unique regardless of the channel.

8.3.2 Posterior of the hidden variables

We need the posterior of the latent factors given the i-vectors to train the model by EM iterations and for efficient evaluation of the trial log-likelihood ratios. First, we find convenient to define the channel dependent sufficient statistics. The zeroth and first order sufficient statistics for speaker i and channel k are defined as:

$$N_{ik} = \sum_{j=1}^{N_i} P\left(z_{ijk} = 1\right) \tag{8.17}$$

$$\mathbf{F}_{ik} = \sum_{j=1}^{N_i} P\left(z_{ijk} = 1\right) \phi_{ij} \tag{8.18}$$

where N_i is the number of i-vectors of speaker i and $P\left(z_{ijk} = 1\right)$ is the probability for ϕ_{ij} to be generated by channel k. Besides, we define the channel centered statistics as:

$$\overline{\mathbf{F}}_{ik} = \mathbf{F}_{ik} - N_{ik}\mu_k \ . \tag{8.19}$$

It can be shown that, the posterior of $\mathbf{y}_i$ given $\mathbf{\Phi}_i$ and $\mathbf{z}_i$ is Gaussian distributed as:

$$P\left(\mathbf{y}_i|\mathbf{\Phi}_i, \mathbf{z}_i, \mathcal{M}\right) = \mathcal{N}\left(\mathbf{y}_i|\mathbf{L}_i^{-1}\gamma_i, \mathbf{L}_i^{-1}\right) \tag{8.20}$$

where

$$\mathbf{L}_i = \mathbf{I} + \sum_{k=1}^{K} N_{ik}\mathbf{V}^T\mathbf{W}_k\mathbf{V} \tag{8.21}$$

$$\gamma_i = \sum_{k=1}^{K} \mathbf{V}^T\mathbf{W}_k\overline{\mathbf{F}}_{ik} \ . \tag{8.22}$$

The derivation of this result can be found in Appendix D.

Note that (8.20) does not average the channel covariances to estimate the expectation of $\mathbf{y}_i$ as in other works like [McLaren and Leeuwen, 2012]. Instead, the channel dependent first order statistics $\overline{\mathbf{F}}_{ik}$ are multiplied by the precision matrix of their corresponding channel $\mathbf{W}_k$ and, then, summed. In theory, this model should robustly estimate the speaker identity variable when we have many i-vectors produced in a wide variety of channels.

8.3.3 Model training

The model is trained by expectation maximization maximum likelihood and minimum divergence iterations. Proofs for the equations in these sections can be found in Appendix D.

8.3.3.1 Maximum likelihood step

The likelihood of the training data is maximized by maximizing the EM auxiliary function $\mathcal{Q}(\mathcal{M})$ with respect to the model parameters:

$$\mathcal{Q}(\mathcal{M}) = \sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[\ln P\left(\mathbf{\Phi}_i|\mathbf{y}_i, \mathbf{z}_i, \mathcal{M}\right)\right] + \mathrm{E}_{\mathbf{Y}}\left[\ln P\left(\mathbf{y}_i\right)\right] \ . \tag{8.23}$$

If we derive $\mathcal{Q}$ with respect to μ_k and $\mathbf{V}$, we obtain that the optimum for μ_k is

$$\mu_k = \frac{1}{N_k}\left(\mathbf{F}_k - \mathbf{V}\mathbf{A}_k\right) \ . \tag{8.24}$$

with $k = 1, \ldots, K$; and the optimum for $\mathbf{V}$ is computed by solving the linear equation system:

$$\sum_{k=1}^{K}\left(\mathbf{B}_k^T \otimes \mathbf{W}_k\right) \mathrm{vec}(\mathbf{V}) = \mathrm{vec}(\mathbf{D}) \tag{8.25}$$

where $\otimes$ is the Kronecker product; vec() is the vectorization operator that converts the matrix into a column vector by stacking its columns; and we conveniently defined the following identities:

$$\mathbf{R}_{\mathbf{y}k} = \sum_{i=1}^{M} N_{ik}\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right] \tag{8.26}$$

$$\mathbf{A}_k = \sum_{i=1}^{M} N_{ik}\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right] \tag{8.27}$$

$$\mathbf{B}_k = \mathbf{R}_{\mathbf{y}k} - \frac{1}{N_k}\mathbf{A}_k\mathbf{A}_k^T \tag{8.28}$$

$$\mathbf{C}_k = \sum_{i=1}^{M} \mathbf{F}_{ik}\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]^T \tag{8.29}$$

$$\mathbf{D} = \sum_{k=1}^{K} \mathbf{W}_k\left(\mathbf{C}_k - \frac{1}{N_k}\mathbf{F}_k\mathbf{A}_k^T\right) \ . \tag{8.30}$$

We also need the expectations of the latent factors that are given by

$$\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right] = \gamma_i\mathbf{L}_i^{-1} \tag{8.31}$$

$$\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right] = \mathbf{L}_i^{-1} + \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]^T \ . \tag{8.32}$$

By deriving $\mathcal{Q}$ with respect to $\mathbf{W}_k^{-1}$, we obtain the optimum for the within-class covariances:

$$\begin{aligned}\mathbf{W}_k^{-1} = &\frac{1}{N_k}\left(\mathbf{S}_k - \mathbf{F}_k\mu_k^T - \mu_k\mathbf{F}_k^T + N_k\mu_k\mu_k^T\right. \\ &\left.-\mathbf{C}_k\mathbf{V}^T - \mathbf{V}\mathbf{C}_k^T + \mu_k\mathbf{A}_k^T\mathbf{V}^T + \mathbf{V}\mathbf{A}_k\mu_k^T + \mathbf{V}\mathbf{R}_{\mathbf{y}k}\mathbf{V}^T\right)\end{aligned} \tag{8.33}$$

where we need to plug-in the channel dependent global sufficient statistics:

$$N_k = \sum_{i=1}^{M}\sum_{j=1}^{N_i} P\left(z_{ijk}=1\right) \tag{8.34}$$

$$\mathbf{F}_k = \sum_{i=1}^{M}\sum_{j=1}^{N_i} P\left(z_{ijk}=1\right)\phi_{ij} \tag{8.35}$$

$$\mathbf{S}_k = \sum_{i=1}^{M}\sum_{j=1}^{N_i} P\left(z_{ijk}=1\right)\phi_{ij}\phi_{ij}^T \ . \tag{8.36}$$

Note that the means μ_k depend on the value of $\mathbf{V}$; $\mathbf{V}$ depend on $\mathbf{W}_k$; and the precisions $\mathbf{W}_k$ depend on both $\mathbf{V}$ and μ_k. These dependencies make necessary to update the parameters iteratively until convergence.

8.3.3.2 Minimum divergence step

A thorough discussion of minimum divergence (MD) estimation can be found in [Brummer, 2009]. To carry out the MD step, first, we need to assume a general prior for $\mathbf{y}$, instead of a standard normal prior ($\mathcal{N}\left(\mathbf{y}|\mathbf{0},\mathbf{I}\right)$):

$$P\left(\mathbf{y}\right) = \mathcal{N}\left(\mathbf{y}|\mu_{\mathbf{y}},\mathbf{\Lambda}_{\mathbf{y}}^{-1}\right) \ . \tag{8.37}$$

Thus, our model $\mathcal{M} = (\mu,\mathbf{V},\mathbf{W},\mu_{\mathbf{y}},\mathbf{\Lambda}_{\mathbf{y}})$ has two new parameters. The resulting model is an over-parametrized model, that is, the parametrization is redundant because we could find another set of parameters $\mathcal{M}' = (\mu',\mathbf{V}',\mathbf{W}')$ that meets the equivalence:

$$(\mu,\mathbf{V},\mathbf{W},\mu_{\mathbf{y}},\mathbf{\Lambda}_{\mathbf{y}}) \equiv (\mu',\mathbf{V}',\mathbf{W}',\mathbf{0},\mathbf{I}) \ . \tag{8.38}$$

If the maximum likelihood step converges without being trapped into a local maximum of the objective function, eventually, the prior of $\mathbf{y}$ will be standard normal. Meanwhile, the distribution of $\mathbf{y}$ is the above generic Gaussian. The MD step consists in estimating $\mu_{\mathbf{y}}$ and $\mathbf{\Lambda}_{\mathbf{y}}$, and, thereafter, obtaining the equivalent model that makes $\mu_{\mathbf{y}} = \mathbf{0}$ and $\mathbf{\Lambda}_{\mathbf{y}} = \mathbf{I}$. This step is called minimum divergence because, in practice, when we transform the model we are minimizing the divergence between the standard and the generic prior. If has been observed that applying the minimum divergence step between maximum likelihood steps helps to escape saddle point and speeds up the convergence of the EM algorithm.

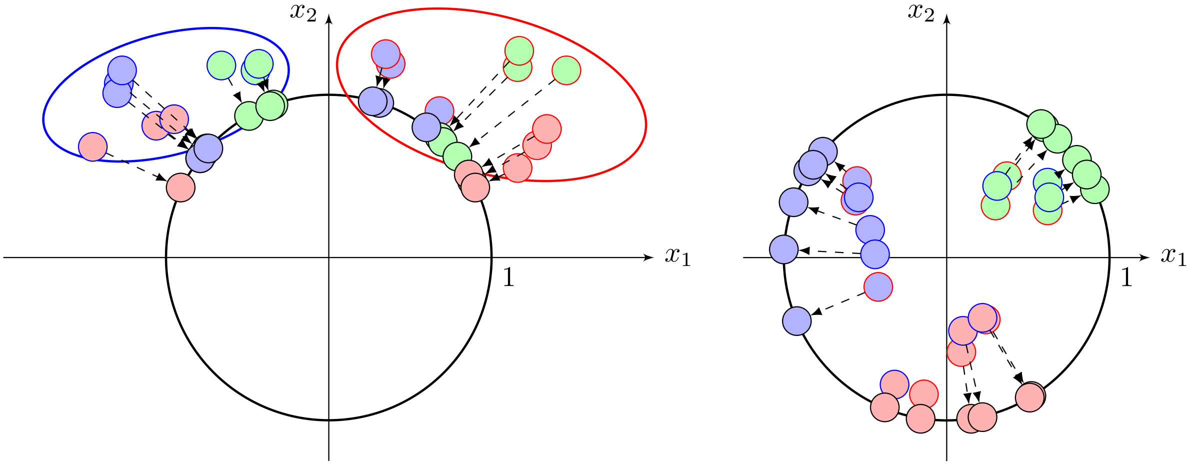
To obtain the optimum values for $\mu_{\mathbf{y}}$ and $\mathbf{\Lambda}_{\mathbf{y}}$, we maximize the term of the EM auxiliary that corresponds to the prior of $\mathbf{y}$:

$$\mathcal{Q}(\mu_{\mathbf{y}},\mathbf{\Lambda}_{\mathbf{y}}) = \sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[\ln\mathcal{N}\left(\mathbf{y}_i|\mu_{\mathbf{y}},\mathbf{\Lambda}_{\mathbf{y}}^{-1}\right)\right] \ . \tag{8.39}$$

We obtain that

$$\mu_{\mathbf{y}} = \frac{1}{M}\sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right] \tag{8.40}$$

$$\mathbf{\Sigma}_{\mathbf{y}} = \mathbf{\Lambda}_{\mathbf{y}}^{-1} = \frac{1}{M}\sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right] - \mu_{\mathbf{y}}\mu_{\mathbf{y}}^T \ . \tag{8.41}$$



(a) Without centering and whitening.

(b) With centering and whitening.

Figure 8.2: Multi-channel length normalization. Samples with borders in red belong to channel 1 and samples with borders in blue belong to channel 2. Length normalized samples have borders in black. The fill colors indicate different speakers.

Now, we need a transform $\mathbf{y} = \psi(\mathbf{y}')$ such as $\mathbf{y}'$ has a standard prior. That is

$$\mathbf{y} = \mu_{\mathbf{y}} + (\mathbf{\Sigma}_{\mathbf{y}}^{1/2})^T \mathbf{y}' \ . \tag{8.42}$$

Finally, this transform is used to make μ_k and $\mathbf{V}$ to absorb the effect of the non-standard prior:

$$\mu'_k = \mu_k + \mathbf{V}\mu_{\mathbf{y}} \tag{8.43}$$

$$\mathbf{V}' = \mathbf{V}(\mathbf{\Sigma}_{\mathbf{y}}^{1/2})^T \ , \tag{8.44}$$

where $\mathbf{\Sigma}_{\mathbf{y}}^{1/2}$ is the upper triangular Cholesky decomposition of $\mathbf{\Sigma}_{\mathbf{y}}$.

8.3.4 Multichannel i-vector length normalization

The technique popularly known as length normalization consist in dividing the i-vector by their magnitude so its norm become one:

$$\hat{\phi} = \frac{\phi}{\|\phi\|} \ . \tag{8.45}$$

The results shown in Chapter 2, evidence that by length normalizing the i-vector before classification, performance greatly improves. For high dimensional data, length normalization transforms the i-vector distribution, which is naturally heavy tailed, into something closer to a Gaussian [Garcia-Romero and Espy-Wilson, 2011]. Thus, we can apply the simple and computationally efficient Gaussian models to describe the i-vector distributions while attaining good performance.

In order to perform length normalization successfully, previously, we need to center and whiten the i-vector distribution. Centering places the i-vector distribution in the origin of coordinates. Meanwhile, whitening rotates and scales the distribution to decorrelate the

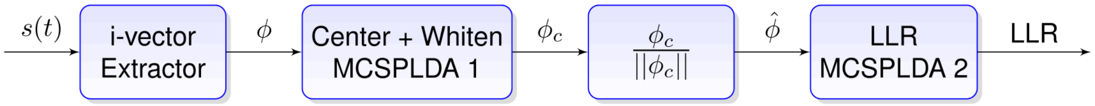


Figure 8.3: Trial evaluation with length normalization and MCSPLDA.

i-vector dimensions and equalize the variances in all directions. There are several ways of centering and whitening a distribution. One of them consists in computing the expectation of the speaker variable $\mathbf{y}$ of the SPLDA model i-vector by i-vector. As the prior of $\mathbf{y}$ is standard normal, the vectors obtained in this manner are, by definition, centered and whitened. If the matrix $\mathbf{V}$ of the SPLDA is full-rank we just center and whiten the i-vector but if it is not, we also reduce its dimension discriminatively. If we have i-vectors from different types of channels, we can perform channel dependent centering and whitening by computing the expectation of the speaker variable with a MCSPLDA model. That expectation is given by

$$\phi_c = \left(\mathbf{I} + \sum_{k=1}^{K} P\left(z_k = 1\right) \mathbf{V}^T \mathbf{W}_k \mathbf{V}\right)^{-1} \sum_{k=1}^{K} P\left(z_k = 1\right) \mathbf{V}^T \mathbf{W}_k \left(\phi - \mu_k\right) \tag{8.46}$$

where ϕ and ϕ_c are i-vector before and after centering and whitening; and $P\left(z_k = 1\right)$ are the probabilities for the i-vector to belong to each one of the channels. Note that we used soft values for $P\left(z_k = 1\right)$.

Figure 8.2 depicts the process of multi-channel length normalization. There are two channels types, samples with borders in red correspond to channel 1 and samples with borders in blue correspond to channel 2. Length normalized i-vectors from both channels have their borders in black. There are three speakers denoted by different fill colors (red, green and blue). Subfigure 8.2a plots the case where we length normalize the i-vectors without doing centering and whitening. We can see that samples from different speakers are projected very close together making i-vectors less discriminative. Meanwhile, there are samples from the same speaker that are projected to different sides of the unit hyper-sphere. Subfigure 8.2b plots the case where i-vectors are centered and whitened. In this case, normalized i-vectors are evenly distributed around the hyper-sphere. Besides, i-vectors from the same speaker are projected to the same region of the hyper-sphere.

Figure 8.3 shows the block diagram of a system that uses multi-channel length normalization. Note, that the system uses two different MCSPLDA models. The first model (MCSPLDA 1) is trained on non-normalized i-vectors. We use MCSPLDA 1 to compute the centered i-vectors ϕ_c. Then, we length normalize the i-vectors to obtain $\hat{\phi}$. Finally, trials are scored by a second MCSPLDA model. MCSPLDA 2 is trained on length normalized i-vectors.

8.3.5 Trial evaluation

As with SPLDA, we can adopt different approximations to evaluate likelihood ratios with MCSPLDA. The expression for the theoretically correct way of evaluation is equivalent to the one of SPLDA:

$$R\left(\mathbf{\Phi}_{\text{enroll}},\phi_{\text{tst}},\mathbf{z}_{\text{enroll}},\mathbf{z}_{\text{tst}}\right) =\frac{P\left(\mathbf{\Phi}_{\text{enroll}},\phi_{\text{tst}}|\mathbf{z}_{\text{enroll}},\mathbf{z}_{\text{tst}},\mathcal{T}\right)}{P\left(\mathbf{\Phi}_{\text{enroll}},\phi_{\text{tst}}|\mathbf{z}_{\text{enroll}},\mathbf{z}_{\text{tst}},\mathcal{N}\right)} \tag{8.47}$$

$$= \left.\frac{P\left(\mathbf{y}|\mathbf{\Phi}_{\text{enroll}},\mathbf{z}_{\text{enroll}},\mathcal{M}\right)P\left(\mathbf{y}|\phi_{\text{tst}},\mathbf{z}_{\text{tst}},\mathcal{M}\right)}{P\left(\mathbf{y}|\mathbf{\Phi}_{\text{enroll}},\phi_{\text{tst}},\mathbf{z}_{\text{enroll}},\mathbf{z}_{\text{tst}},\mathcal{M}\right)P\left(\mathbf{y}\right)}\right|_{\mathbf{y}=\mathbf{0}} , \tag{8.48}$$

where the posterior distributions also depend on the channel assignments ($\mathbf{z}_{\text{enroll}}$ and $\mathbf{z}_{\text{tst}}$) of the enrollment and test i-vectors, as shown in (8.20).

Having enrollment i-vectors from different channels, it may be the case that there are more i-vectors from one of the channels than from the others. If we attained perfect channel compensation, this would not be an issue. However, in practice, channel compensation is never ideal so the dominant channel has a larger weight on the $\mathbf{y}$ posterior. To prevent this, we proposed to scale the i-vector sufficient statistics (N_{ik}, $\overline{\mathbf{F}}_{ik}$) so that all channel types have the same number of effective i-vectors N_{eff}. The scaled statistics N'_{ik} and $\overline{\mathbf{F}}'_{ik}$ are

$$N'_{ik} = N_{\text{eff}}, \qquad \overline{\mathbf{F}}'_{ik} = \frac{N_{\text{eff}}}{N_{ik}}\overline{\mathbf{F}}_{ik}, \qquad k=1,\dots,K\ . \tag{8.49}$$

Then, we just plugged-in the scaled statistics into (8.20) to compute the $\mathbf{y}$ posterior. Thus, we maintain the average i-vector per channel constant while we make the i-vectors to count as if there were N_{eff} of them. Besides, the value of N_{eff} controls the variance of the posterior. Selecting a low N_{eff} the posterior becomes wider and avoids being over-confident about the value of $\mathbf{y}$, which may happen due to the i.i.d. assumption, as we explained before.

Another option is to do something similar to i-vector averaging. We make the enrollment i-vectors to count like only one i-vector by scaling the sufficient statistics as:

$$N'_{ik} = \frac{N_{ik}}{\sum_{k=1}^{K} N_{iK}}, \qquad \overline{\mathbf{F}}'_{ik} = \frac{1}{\sum_{k=1}^{K} N_{iK}}\overline{\mathbf{F}}_{ik}, \qquad k=1,\dots,K\ . \tag{8.50}$$

Thus, we can compare the i-vector averaging scores for SPLDA and MCSPLDA. Finally, we also can do scoring averaging:

$$\ln R_{\text{s}-\text{avg}}\left(\mathbf{\Phi}_{\text{enroll}},\phi_{\text{tst}},\mathbf{z}_{\text{enroll}},\mathbf{z}_{\text{tst}}\right) = \frac{1}{N}\sum_{i=1}^{N}\ln\frac{P\left(\phi_i,\phi_{\text{tst}}|\mathbf{z}_i,\mathbf{z}_{\text{tst}},\mathcal{T}\right)}{P\left(\phi_i,\phi_{\text{tst}}|\mathbf{z}_i,\mathbf{z}_{\text{tst}},\mathcal{N}\right)}\ . \tag{8.51}$$

where we score each enrollment i-vector against the test i-vector and, then, the scores are averaged.

8.3.6 Channel detection

When the channel where the i-vector was recorded is unknown, we estimate the probabilities $P\left(z_k\right)$ based on a set of quality measures $\mathbf{Q}$. Our measures were signal-to-noise ratio, modulation index, spectral entropy and UBM log-likelihood. These measures were computed as described in Chapter 4. For each channel type, we trained a mixture of Gaussians $P\left(\mathbf{Q}|\mathbf{z}\right)$. Then, the posterior probability for each channel was computed as:

$$P\left(z_k=1|\mathbf{Q}\right) = \frac{w_k P\left(\mathbf{Q}|z_k=1\right)}{\sum_{k=1}^{K} w_k P\left(\mathbf{Q}|z_k=1\right)} \tag{8.52}$$

where w_k were the channel priors. In our experiments, we defined 6 channel types (telephone and far-field microphones × 3 noise levels: clean, 15 dB and 6 dB) and trained a GMM of 8 Gaussians for each one of them.

8.4 Experimental Setup

8.4.1 Evaluation database

We evaluated the multiples PLDA flavors on the NIST SRE 2012 dataset [NIST Speech Group, 2012]. In previous evaluations, the enrollment data was released, together with the test data, at evaluation time. The core condition just consisted in comparing one enrollment segment with one test segment. In NIST SRE12, most of the target speakers were taken from evaluations NIST SRE06 to SRE10. To enroll each speaker, organizers allowed us to use all the available segments. Thus, the core condition, instead of posing a *1 side against 1 side* scenario, represented a *N sides versus 1 side* scenario. Another difference with previous evaluations was that we could use the enrollment speakers during the development phase of the system (training of the UBM, i-vector extractor, calibration, score normalization, etc.). Furthermore, NIST SRE12 proposed new challenges like speech with artificially added noise, speech collected in noisy environments and segments of different duration.

NIST SRE12 had 1918 target speakers in total (763 male + 1155 female), of which 1818 speakers were from previous evaluations and 100 speakers were new speakers–released at evaluation time. The evaluation proposed a core and an extended core condition with more trials that allowed us to compute the error rates more accurately. We present results on the extended condition. Because of the multiple types of test conditions and the 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. Rather, the core conditions trials were divided into five common conditions–subsets of trials that satisfy additional constraints–to better foster system comparison:

- Det1: All trials involving multiple segment enrollment and interview speech in test without added noise in test. It consisted of 23681 test segments and 22338663 trials.

- Det2: All trials involving multiple segment enrollment and phone call speech in test without added noise in test. It consisted of 24977 test segments and 12408306 trials.

- Det3: All trials involving multiple segment enrollment and interview speech with added noise in test. It consisted of 18449 test segments and 17254299 trials.

- Det4: All trials involving multiple segment enrollment and phone call speech with added noise in test. It consisted of 22058 test segments and 9603225 trials.

- Det5: All trials involving multiple segment enrollment and phone call speech intentionally collected in a noisy environment in test. It consisted of 5483 test segments and 5200758 trials.

As all common conditions involved enrollment with multiple segments and new speakers had only one enrollment segment, new speakers were not scored in the common conditions.

Noises added to the test segments were of different types: babble, HVAC (heating, ventilation, and air conditioning), and single speaker. The power of the noise was selected to produce signal-to-noise ratios of 15 and 6dB. Babble noises were created by adding a large number of conversations.

8.4.2 Development database

We created a development dataset that took into account the new scenario presented by NIST SRE12. The data from previous evaluations was divided into two parts:

- Training: This part included all the signals from SRE04–06 and 70% of the signals from SRE08 and SRE10. We used it to train UBM, JFA and PLDA; and to enroll the target speakers.

- Evaluation: We reserved a 30% of the speech in SRE08 and SRE10 to create a test set for training calibration and evaluating our system. It included short telephone calls, short and long interviews and 10 seconds calls.

Speech segments extracted from the same phonecall or interview (same ldc-id) were placed either in the training part or in the evaluation part but not in both.

Both parts of the dataset were augmented by adding babble and HVAC[1] noises with signal-to-noise ratios of 15 and 6 dB, following NIST SRE12 guidelines. Babble noises were created by averaging 1000 conversations from previous evaluations. Different noise samples were added to the training and evaluation sets. The power of the noise and speech signals was estimated with a psophometric filter and a VAD. Based on them, the power of the noise was modified to obtain the desired SNR. The noise added to telephone segments was filtered by a simulated telephone channel.

Adding noisy versions, our training set included 66457 male and 87826 female segments from 982 male and 1372 female speakers. The evaluation set included 15585 male and 21902 female segments. We scored all the NIST SRE12 target speakers from previous years against all the test segments. Thus, we produced 11267955 male and 23982690 female trials.

The enrollment lists were composed of segments from the training subset without added noise. We compare results with three different enrollment lists:

- **AllLDCId**: all the telephone and interview segments. Each interview was recorded simultaneously by 13 different channels. This list included all the channels.

- **1LDCId**: all the telephone segments and only one channel by interview.

- **PHN**: all the telephone segments without any interviews.

8.4.3 Performance measure

We report results in terms of the new primary cost function defined in [NIST Speech Group, 2012]. This was the average of two detection cost functions (DCF) computed in two different operating points: $P_{\mathcal{T}} = 0.01$ and 0.001 (NIST SRE10 operating point). This was intended to improve the stability of the cost measure and to increase the importance of good score calibration over a wider range of log-likelihoods. Another novelty was that NIST SRE12 computed separate false acceptance rates for known non-targets–impostors that are in the set of target speakers–and unknown non-targets–impostors that are not enrolled into the system. We do a weighted sum of both rates using the prior probability of known non-target P_{known}:

$$P_{\text{FA}} = P_{\text{known}} P_{\text{FA}|\text{known}} + (1 - P_{\text{known}}) P_{\text{FA}|\text{unk}} \tag{8.53}$$

[1] We downloaded HVAC noises from Freesound.org

where $P_{\text{known}} = 0.5$. Then, the DCF is computed normally as:

$$C(P_{\mathcal{T}}) = C_{\text{Miss}} P_{\mathcal{T}} P_{\text{Miss}} + C_{\text{FA}}(1 - P_{\mathcal{T}}) P_{\text{FA}} \ , \tag{8.54}$$

where the miss and false acceptance costs were $C_{\text{Miss}} = C_{\text{FA}} = 1$. To improve the intuitive meaning of the cost, it was normalized by dividing it by the best cost that could be obtained without knowledge of the input data:

$$C_{\text{Norm}}(P_{\mathcal{T}}) = \frac{C(P_{\mathcal{T}})}{C_{\text{Miss}} P_{\mathcal{T}}} \ . \tag{8.55}$$

Actual detection costs were computed by applying the Bayes decision threshold $-\text{logit}(P_{\mathcal{T}})$ to the calibrated scores.

The primary performance measure of the evaluation was the average:

$$C_{\text{Primary}} = \frac{C_{\text{Norm}}(0.01) + C_{\text{Norm}}(0.001)}{2} \ . \tag{8.56}$$

8.4.4 SV system configuration

8.4.4.1 i-Vector Extraction

The acoustic features of the system consisted of 20 MFCC ($C_0 - C_{19}$) with deltas and double deltas. Short-time Gaussianization [Pelecanos and Sridharan, 2001] was applied after silence removing.

Voice Activity Detection (VAD) was based on the long-term spectral divergence (LTSD) of the signal [Ramirez et al., 2004]. For phonecalls, where two channels are available, namely target channel and reference channel, the reference channel was used for cross-talk removal by comparing the energy in both channels. For interviews and microphone recorded phonecalls the channel energies were unbalanced so we could not remove the interviewer based on them. We followed the following steps:

1. We ran the VAD in both channels. We used a restrictive energy threshold on the reference channel to try to eliminate the target speaker. Then, we subtracted the VAD from the reference channel from the target channel.

2. If we eliminate more than 50% of the frames, we conjectured that we might be eliminating too much speech from the target speaker.

3. In that event, we ran a diarization system based on Bayesian information criterion and agglomerative hierarchical clustering (BIC + AHC) [Vaquero, 2011] on the reference channel. The cluster with more energy was assumed to correspond to the reference speaker. Then, we subtracted the reference speaker from the VAD of the target channel.

4. Finally, when we pruned more than 75% of the frames, we assumed that there was few speech from the interviewer and we kept all the frames of the target channel.

We trained full covariance, gender dependent UBMs with 2048 Gaussian components. We used a 600 dimension i-vector extractor. All UBM and i-vector extractors were trained on telephone data from the training part of our development dataset without added noise by ML+MD iterations.

We reduced the i-vector dimensionality to 400 by using SPLDA or MCSPLDA. That has the side effect of centering and whitening the i-vectors, as explained in Section 8.3.4. Then, we applied i-vector length normalization.

8.4.4.2 PLDA

SPLDA and MCSPLDA were trained on the pool of clean and added noise data from the training part the development dataset. Each system had two PLDA models, one for whitening the i-vectors–trained on plain i-vectors–and another for scoring the trials–trained on length normalized i-vectors. The types of both PLDA were matched, that is, we used either SPLDA in both steps or MCSPLDA in both steps.

The model parameters were estimated by ML+MD iterations. For MCSPLDA, we considered 6 channels: telephone and microphone times 3 noise levels (clean, 6dB and 15 dB). We tried two estimation procedures:

- Simple: $\mathbf{V}$, μ_k, and $\mathbf{W}_k$ for $k = 1, \dots, 6$ were estimated all together from scratch on pooled clean and noisy data.

- Progressive: We trained $\mathbf{V}$, μ_{phn} and $\mathbf{W}_{\text{phn}}$ on clean telephone data. Then, μ_{mic} and $\mathbf{W}_{\text{mic}}$ were initialized with the values of μ_{phn} and $\mathbf{W}_{\text{phn}}$, and re-estimated on clean microphone data. Finally, we added four μ_k and $\mathbf{W}_k$ for the channels with noise added, we initialized them with the parameters of the clean channels, and we re-estimate them on pooled clean and noisy data.

In the test phase, where the noise level was not given, we estimated the channel type with (8.52) where we chose $w_{\text{clean}} = w_{\text{15dB}} = w_{\text{6dB}} = 1/3$. We trained channel detection models on data from NIST SRE04–08 with noise added. The accuracy of this classifier for the training dataset was around 99%.

8.4.4.3 Calibration

We calibrated the scores by linear logistic regression with the Bosaris toolkit [Brummer and De Villiers, 2011] where we plugged-in a target prior $P_{\mathcal{T}} = 0.0055$, which is the average of the two operating points of the evaluation. Calibration was gender dependent.

We computed the compound likelihood ratio for the calibrated likelihood ratios as explained in [Brummer, 2012]. The idea of computing a compound likelihood ratio comes from the fact that NIST SRE12 rules allowed using the data from other target speakers to evaluate the trial. Thus, we can compute the posterior for the target hypothesis as:

$$P(\mathcal{T}|\mathcal{D}) = \frac{P_{\mathcal{T}} P(\mathcal{D}|S_1)}{(1-P_{\mathcal{T}})(1-P_{\text{known}})P(\mathcal{D}|\text{unk}) + P_{\mathcal{T}} P(\mathcal{D}|S_1) + (1-P_{\mathcal{T}})\frac{P_{\text{known}}}{M-1}\sum_{i=2}^{M} P(\mathcal{D}|S_i)} \tag{8.57}$$

where, without loss of generality, S_1 is the speaker under test and $S_2, \dots, S_M$ are the rest of speaker enrolled in the system; $\mathcal{D}$ are the enrollment i-vectors from all the speakers plus the test i-vector; and $P_{\text{known}} = 0.5$. We can write that posterior as a function of the likelihood ratios:

$$P(\mathcal{T}|\mathcal{D}) = \frac{P_{\mathcal{T}} R_1}{(1-P_{\mathcal{T}})(1-P_{\text{known}}) + P_{\mathcal{T}} R_1 + (1-P_{\mathcal{T}})\frac{P_{\text{known}}}{M-1}\sum_{i=2}^{M} R_i} . \tag{8.58}$$

If the likelihood ratios R_i are well-calibrated, this posterior allows to make cost-effective speaker recognition decisions. The posterior depends on the prior $P_{\mathcal{T}}$. In contrast, the likelihood ratios R_i provide prior independent information. We can obtain a likelihood ratio independent of $P_{\mathcal{T}}$, which we call compound likelihood ratio R', as:

$$R' = \frac{P(\mathcal{T}|\mathcal{D})}{1-P(\mathcal{T}|\mathcal{D})}\frac{1-P_{\mathcal{T}}}{P_{\mathcal{T}}} \tag{8.59}$$

$$= \frac{R_1}{1-P_{\text{known}}+\frac{P_{\text{known}}}{M-1}\sum_{i=2}^{M} R_i} \; . \tag{8.60}$$

By applying the threshold $-\text{logit}P_{\mathcal{T}}$ to R' we can easily evaluate the actual DCF of the system for multiple values of $P_{\mathcal{T}}$.

8.5 Experiments Results

8.5.1 Classifier Type and enrollment lists analysis

In Table 8.1, we compare our three classifiers (SPLDA, MCSPLDA with simple training mode and MCSPLDA with progressive training mode) evaluated with different scoring strategies (standard, i-vector statistics scaling, i-vector averaging and score averaging) and using our three different enrollment lists (AllLDCId, 1LDCId, PHN). The table shows minimum and actual values of C_{Primary}.

Clearly, the best scoring rule was i-vector averaging. It was the best for all classifiers and enrollment lists. The second best was score averaging, followed by statics scaling. The worst by far was the *by the book* scoring, which confirms the fact that it over-fits the estimation of the speaker variable $\mathbf{y}$. For example, for SPLDA with the list 1LDCId, i-vector averaging improved actual DCF by 44–52% with respect to the standard scoring.

Regarding the classifier type, MCSPLDA with progressive training performed the worst regardless of the enrollment list and the scoring type. SPLDA and MCSPLDA with simple training performed very closely, especially when using i-vector averaging. SPLDA performed around 14% better, in terms of actual cost, in the conditions without added noise (det1, det2 and det5) while MCSPLDA simple was around 5% better in conditions with artificial noise added (det3 and det4). We consider that a 5% of improvement is not really significant so the extra computational cost required by MCSPLDA would not be worthy.

Comparing enrollment lists, we found that the list that includes interviews recorded simultaneously by different microphones (AllLDCId) performed better in conditions with interviews in test (det1, det3). It was around 8–10% better than 1LDCId and around 14% better than PHN. On the other hand, the enrollment list with only telephone data (PHN) was the best for telephone tests. It was around 4-5% better than 1LDCId and around 22-33% better than AllLDCDId. As we did not achieved perfect telephone–microphone channel compensation, the number of enrollment segments of each type matters. We consider that, in general, 1LDCId is the best enrollment list since it offered the best equilibrium between telephone and interview performance.

8.5.2 Condition detection analysis

According to NIST SRE12 evaluation rules, the noise type and level of the test signals is suppose to be unknown. However, that information was available in the keys released *a*

Table 8.1: MinDCF/Actual DCF for several classifiers, scoring strategies and enrollment lists.

System							
Enroll.	Classif.	Scoring	det1	det2	det3	det4	det5
AllLDCId		std.	0.39/0.42	0.53/0.56	0.30/0.32	0.61/0.67	0.57/0.62
	SPLDA	iv-avg	**0.23**/**0.23**	0.21/0.27	0.18/0.18	0.26/0.35	0.24/0.33
		s-avg	0.25/0.26	0.30/0.34	0.20/0.20	0.35/0.43	0.34/0.40
		std.	0.47/0.50	0.55/0.59	0.31/0.33	0.53/0.60	0.59/0.65
	MCSPLDA	iv-scal	0.35/0.39	0.44/0.51	0.22/0.24	0.43/0.51	0.48/0.57
	simple	iv-avg	**0.23**/0.26	0.23/0.33	**0.17**/**0.17**	0.24/0.34	0.25/0.38
		s-avg	0.26/0.31	0.29/0.40	0.19/0.20	0.30/0.40	0.34/0.44
		std.	0.43/0.48	0.50/0.67	0.28/0.30	0.47/0.54	0.54/0.72
	MCSPLDA	iv-scal	0.35/0.40	0.40/0.66	0.23/0.24	0.45/0.51	0.43/0.71
	prog.	iv-avg	0.28/0.31	0.23/0.57	0.20/0.20	0.35/0.44	0.24/0.62
		s-avg	0.32/0.37	0.28/0.60	0.23/0.24	0.41/0.47	0.30/0.64
1LDCID		std.	0.45/0.46	0.38/0.40	0.35/0.36	0.46/0.51	0.42/0.46
	SPLDA	iv-avg	0.24/0.25	**0.17**/0.19	0.19/0.20	0.22/0.26	**0.18**/0.23
		s-avg	0.26/0.27	0.22/0.25	0.21/0.21	0.28/0.33	0.25/0.29
		std.	0.55/0.57	0.40/0.44	0.37/0.39	0.41/0.47	0.43/0.50
	MCSPLDA	iv-scal	0.34/0.39	0.31/0.38	0.22/0.23	0.31/0.39	0.33/0.43
	simple	iv-avg	0.25/0.29	0.18/0.23	0.18/0.19	0.20/0.26	0.19/0.27
		s-avg	0.28/0.33	0.24/0.30	0.20/0.21	0.25/0.31	0.26/0.34
	MCSPLDA	std.	0.65/0.68	0.43/0.55	0.53/0.55	0.48/0.56	0.48/0.61
	prog.	iv-scal	0.38/0.43	0.31/0.51	0.27/0.28	0.39/0.43	0.33/0.56
		iv-avg	0.34/0.36	0.20/0.35	0.26/0.26	0.29/0.33	0.21/0.41
		s-avg	0.38/0.42	0.23/0.40	0.31/0.31	0.34/0.38	0.25/0.45
PHN		std.	0.48/0.50	0.38/0.40	0.39/0.40	0.45/0.50	0.41/0.45
	SPLDA	iv-avg	0.25/0.27	**0.17**/**0.18**	0.20/0.21	0.23/0.27	0.19/**0.22**
		s-avg	0.27/0.28	0.22/0.23	0.22/0.22	0.29/0.32	0.25/0.27
		std.	0.60/0.63	0.41/0.45	0.43/0.45	0.43/0.49	0.45/0.51
	MCSPLDA	iv-scal	0.30/0.35	0.20/0.24	0.20/0.21	0.21/0.26	0.20/0.27
	simple	iv-avg	0.27/0.30	0.18/0.22	0.20/0.20	**0.20**/**0.25**	0.19/0.26
		s-avg	0.29/0.33	0.24/0.29	0.21/0.22	0.26/0.31	0.27/0.33
		std.	0.74/0.77	0.48/0.58	0.64/0.67	0.55/0.63	0.52/0.64
	MCSPLDA	iv-scal	0.43/0.48	0.24/0.38	0.34/0.35	0.34/0.37	0.25/0.44
	prog.	iv-avg	0.38/0.40	0.21/0.35	0.30/0.30	0.31/0.35	0.22/0.41
		s-avg	0.39/0.43	0.25/0.39	0.32/0.32	0.36/0.39	0.26/0.44

Table 8.2: Min/Actual DCF for MCSPLDA oracle vs. automatic noise detection.

System		det1	det2	det3	det4	det5
iv-scal	Oracle	0.35/0.41	0.37/0.44	**0.22/0.22**	0.33/0.40	0.39/0.54
	Automatic	**0.34/0.39**	**0.31/0.38**	**0.22**/0.23	**0.31/0.39**	**0.33/0.43**
iv-avg	Oracle	0.26/0.30	0.21/0.28	0.19/**0.19**	0.21/0.28	0.21/0.35
	Automatic	**0.25/0.29**	**0.18/0.23**	**0.18/0.19**	**0.20/0.26**	**0.19/0.27**

posteriori. In this section, we compare the results obtained using oracle noise level labels against automatic noise detection. The detection accuracy in our development dataset was 97% for telephone and 74% for interviews. In the evaluation set, the detection rates decayed to 62% for telephone and 47% for microphone.

Table 8.2 presents results for the MCSPLDA classifier with simple training where we used the 1LDCId enrollment list. Despite the strong degradation of the condition detection accuracy we observe very small differences between both. In general, automatic detection provided better results.

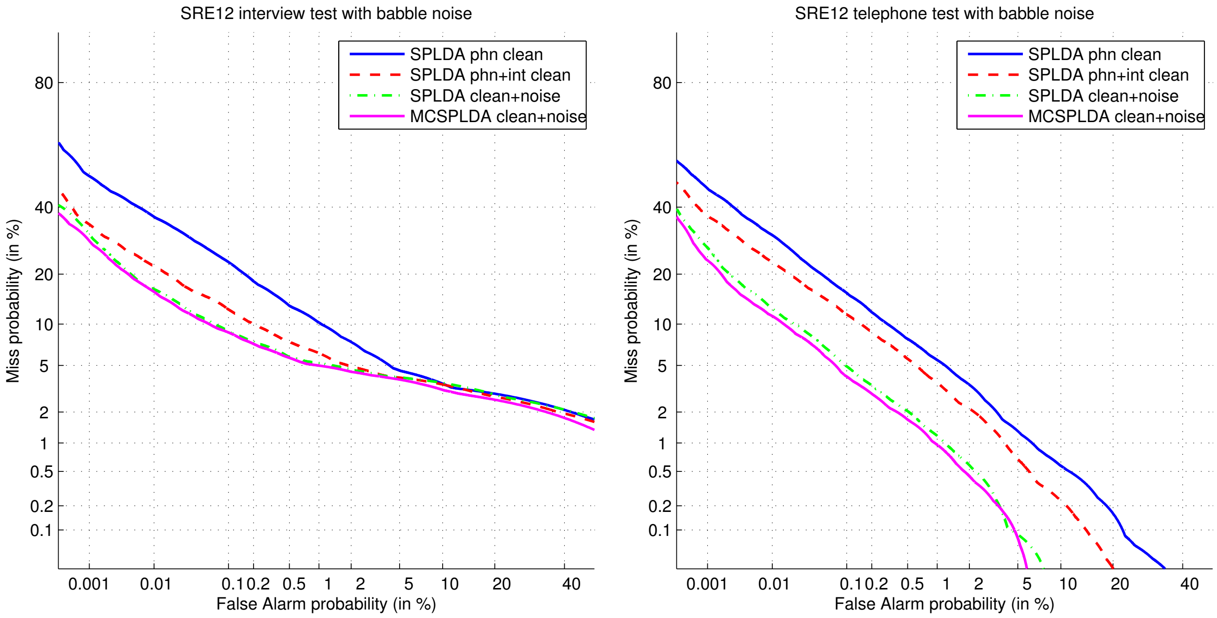
8.5.3 Noise effect analysis

In this section, we analyze the impact of the noise type and level. Table 8.3 compares results for four classifiers: SPLDA trained only on clean telephone data, SPLDA trained on pooled clean interview and telephone data, SPLDA trained on pooled clean and noisy telephone and interview data and MCSPLDA trained also on clean and noisy data from scratch (MCSPLDA simple). We used the enrollment list 1LDCId and the i-vector averaging scoring mode. In NIST SRE12, noise was only added to 300 seconds segments so, to provide a fair comparison, we computed the costs for the clean and noisy environment conditions in the table only accounting trials with 300 seconds tests. Babble and HVAC noises were added to interviews and; babble and single speaker noises to phonecalls.

By Training SPLDA with pooled clean telephone and interview data, we improved actual costs by 35–60% in interviews and by 0–40% with respect to training only with telephone. Contrary to intuition, adding interview data to the PLDA training did not damage the telephone conditions.

Table 8.3: Min/Actual DCF for different noise types and levels.

			Noise Type			Noise Level	
Test ch.	System	Clean	Babble	HVAC(int)/ Spkr(phn)	Noisy Env.	15 dB	6 dB
Int.	SPLDA phn clean	0.25/0.26	0.40/0.40	0.32/0.32		0.29/0.29	0.43/0.44
	SPLDA phn+int clean	**0.11/0.11**	0.25/0.26	0.17/0.18		**0.15/0.15**	0.28/0.29
	SPLDA clean+noise	0.12/0.12	**0.20**/0.22	0.17/**0.17**		0.16/0.16	0.22/0.24
	MCSPLDA clean+noise	**0.11**/0.12	**0.20/0.21**	**0.16/0.17**		**0.15/0.15**	**0.21/0.22**
Phn.	SPLDA phn clean	0.04/0.05	0.31/0.37	0.28/**0.33**	0.07/0.08	0.14/0.18	0.37/0.45
	SPLDA phn+int clean	**0.03/0.03**	0.25/0.31	**0.27/0.33**	**0.05/0.06**	0.10/0.13	0.33/0.40
	SPLDA clean+noise	0.04/0.04	0.17/0.22	0.31/0.35	0.08/0.09	0.09/**0.12**	0.28/0.34
	MCSPLDA clean+noise	0.04/0.04	**0.15/0.20**	0.29/0.38	**0.05**/0.08	**0.07/0.12**	**0.26/0.33**



(a) Interview test.

(b) Telephone test.

Figure 8.4: DET curves for NIST SRE12 babble noise tests.

For the SPLDA trained with clean interview and telephone data, in interviews, HVAC noise worsened actual cost by 63%. Adding noise to the training did not help much; minimum cost did not improve and actual cost improved only by 5%. Babble noise was more harmful. It worsened the cost by 136% and 933% in interviews and phonecalls respectively. In this case, adding noise to the training was more effective, actual cost improved by 15% in interviews and by 29% in telephone with respect to clean training. The worst noise consisted in adding speech of a single speaker to the segment, which degraded performance by 1000% in telephone tests. The noisy training did not help because we did not include that type of noise in our development. The same happened with the tests acquired in noisy environments. Real noise worsened actual cost by 100% and no gain was achieved by noisy training of the PLDA. These results prove that the approach of adding artificial noise to the development only helps if the noises of development and test are similar. Since we cannot add all possible noises to the development, other techniques should be explored.

Regarding the noise levels, for tests with 15 dB of signal-noise-ratio, we obtained small gain or not at all from noisy training; it was -6% in interviews and 8% in phonecalls. For tests of 6 dB, we obtained a noticeable gain of around 15-17%. Improvement in phonecalls was not larger because of the single speaker noise not included in training.

The results of the MCSPLDA classifier were very similar to the ones of SPLDA. Difference between both was always lower than 11%. For example, MCSPLDA was 9% better than SPLDA for telephone babble noise and 8% worse in single speaker noise.

Figure 8.4 displays DET curves for the trials with babble noise in test. This was the type of noise where we obtained larger gains from training with added noise. We can see that, for interviews, there is an improvement in all the low false alarm region. For telephone, the improvement was more or less constant along all the operating points in the curve.

8.6 Summary

In this chapter, we dealt with the problem of having i-vectors recorded in different conditions like different channel types, noise types or noise levels. Intending to approach the problem in a principled way, we introduced a PLDA variant, that we called multi-channel SPLDA (MCSPLDA), where the speaker space distribution is common to all types of channels and the channel space distribution is channel dependent. This model can be seen as a mixture of PLDA where the eigen-voices matrix $\mathbf{V}$ and the speaker factors $\mathbf{y}$ are shared across the components of the mixture. We compared this model with a standard SPLDA just trained on pooled clean and noisy telephone and interview data.

We experimented on the NIST SRE12 dataset that included test with artificially added noises (HVAC, babble and single speaker) as well as segments recorded in noisy environments. We proved that, if we train the PLDA on segments with the same type of noise than the test, we can obtain a significant performance improvement. However, no gain was observed for noises not included in training (single speaker noise and real noisy environment), which makes this method useless for some practical use cases.

Results of MCSPLDA did not differ much from those of SPLDA. It seems that training an unique within-class covariances with all the available data can be more robust than training one covariance per channel type.

Chapter 9

Fully Bayesian Evaluation of PLDA

9.1 Introduction

Bayesian inference is a statistical method that applies Bayes rule to compute the posterior probability for a hypothesis H given a set of observed data points $X = \{x_1, \dots, x_N\}$:

$$P(H|X) = \frac{P(X|H)\,P(H)}{P(X)} \ . \tag{9.1}$$

H can be a certain discrete event, a continuous variable, the parameters of a model, etc. This method requires choosing a hypothesis prior $P(H)$ that encodes the beliefs about H before any data is observed. $P(X|H)$ is called the *likelihood* function and represents the probability of observing X for a fixed value of H. Finally, $P(X)$ is the *marginal likelihood* of the observed data. As $P(X)$ is the same for all possible hypothesis, in many applications, it is disregarded. In contrast with the Bayesian approach, maximum likelihood inference limits itself to computing the value of H that maximizes the likelihood function $P(X|H)$ for the observed data. Thus, maximum likelihood only makes a point estimate for the best value for H while Bayesian inference provides a distribution that gives a different degree of confidence for all the possible values of the hypothesis.

To evaluate whether a new data point $\hat{x}$ has been generated by the same distribution as X, following the Bayesian approach, we marginalize over H using its posterior distribution:

$$P(\hat{x}|X) = \int P(\hat{x}|H)\,P(H|X)\ \mathrm{d}H \ . \tag{9.2}$$

In Bayesian theory, the distribution $P(\hat{x}|X)$ is referred as the *posterior predictive distribution* and when we use it we are doing *predictive inference*. In the non-Bayesian framework, the probability of a new data point is just approximated by the likelihood given the maximum likelihood estimate of H, $P(\hat{x}|H_{\mathrm{ML}})$. The advantage of the Bayesian method over maximum likelihood is that the former takes into account the uncertainty about the value of H while the latter does not. If the posterior $P(H|X)$ is sharply peaked, i.e. if the probability for H_{ML} is much larger than for the rest of values of H, the Bayesian and non Bayesian methods will produce similar outcome, that is:

$$P(\hat{x}|X) \approx P(\hat{x}|H_{\mathrm{ML}}) \ . \tag{9.3}$$

On the other hand, if the posterior assigns high probabilities for several values of H, the Bayesian solution should provide better predictions than the maximum likelihood approximation. As we will see, one of the factors that affect the width of the posterior is the number of samples in X. If we observe many samples we will be very confident about the value of H and we will obtain a very concentrated posterior. On the contrary, if we observe few samples the value of H will be uncertain and the posterior will be flatter.

Through the years, speaker recognition approaches have evolved from the pure maximum likelihood method to frameworks where we introduce more and more elements of the Bayesian philosophy. We can observe this by reviewing some landmark publications. The first approaches based on GMM modeled individual speakers by making maximum likelihood point estimates of the speaker models [Reynolds and Rose, 1995]. The work in [Reynolds et al., 2000] introduces the Bayesian approach by using a Universal GMM (UBM) as prior for the speaker models and the maximum of the posterior is taken as point estimate of the model parameters. That is called *maximum a posteriori* (MAP) or Bayesian adaptation. Then, the likelihood of the test data given the speaker models is still computed in a non-Bayesian way. In Chapter 7, we presented an example of MAP adaptation when we adapted our Bayesian network for reliability estimation of SV decisions from one domain to another.

Joint factor analysis (JFA) puts standard normal priors over speaker and channel factors. However, many works using JFA make MAP point estimates of the latent factors [Burget et al., 2007]. For trial evaluation, they compute the likelihood of the test segment given the point estimate of the speaker factor made on the enrollment segments and an estimate of the channel factor of the test file is used to remove inter-session variability. In [Kenny et al., 2007a], point estimates are made for speaker factors but channel factors are integrated out. With the advent of i-vectors [Dehak et al., 2009, Dehak et al., 2011b], of smaller dimension than JFA supervectors, it became computationally feasible to marginalize over both speaker and channel factors, as a principled way of computing speaker detection likelihood ratios in PLDA models [Kenny, 2010, Brummer and De Villiers, 2010].

Despite the progress made, neither of the above approaches takes into account the uncertainty about the parameters of the probabilistic models, i.e. the factor loading matrices of the JFA, i-vector extractor or the PLDA models. PLDA models are usually trained with the EM algorithm, which maximizes a lower bound of the likelihood function over a large development database (see Appendix C). In this chapter, we intend to go a step further by adopting a *fully Bayesian* approach. In a fully Bayesian treatment we assume that the model parameters are hidden variables with their corresponding priors, the same as the speaker and channel factors. Then, we can compute posterior distributions for them. To make predictions we marginalize with respect to all the latent variables. In particular, we will focus on the simplest case of PLDA model known as two covariance model or full-rank PLDA.

This chapter is organized as follows. In Section 9.2, we introduce the Bayesian version the two covariance model. In this model, the speaker's mean and the between-class covariance are hidden variables, with prior and posterior distributions, instead of point estimates. Section 9.3 explains how two evaluate the likelihood ratio in a fully Bayesian way and shows an alternative formulation that avoids to solve the involved integrals. As the model posteriors cannot be expressed in close form we used variational Bayes to compute approximate posteriors. Section 9.4 presents the VB procedure to obtain the posteriors assuming non-informative priors for the model parameters and Section 9.4 assuming informative priors. In Section 9.6, we explain our experimental setup including some optimizations made to speed-up the VB algorithm. Section 9.7 presents results on NIST SRE10 that prove that Bayesian evaluation dramatically improves the system

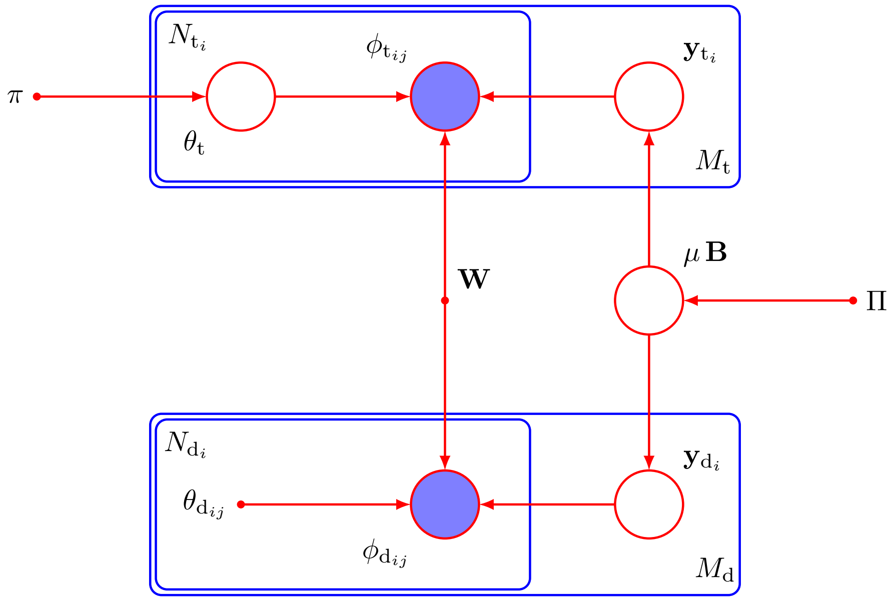


Figure 9.1: BN for fully Bayesian evaluation of two-covariance model.

performance. We compared results using non-informative priors to compute the LLR against informative priors; score normalization against non-normalized scores; and length normalization against the Bayesian approach. We found Bayesian evaluation and length normalization make score normalization useless. In certain conditions, both methods were complementary. Finally, Section 9.8 summarizes the chapter.

9.2 Bayesian Two-covariance Model

The two-covariance model is a generative model that is used to describe the i-vector between and within speaker distributions [Brummer and De Villiers, 2010]. We already introduced this model in Chapter 2. As a reminder, the model assumes that an observed i-vector ϕ_{ij} of speaker i can be written as the sum of two hidden variables:

$$\phi_{ij} = \mathbf{y}_i + \epsilon_{ij} \tag{9.4}$$

where $\mathbf{y}_i$ is called the speaker identity variable and ϵ_{ij} is the channel offset. The identity variable remains constant between different observations of the speaker, but the channel offset changes. This latent variables have Gaussian priors:

$$\mathbf{y}_i \sim \mathcal{N}\left(\mathbf{y}_i|\mu, \mathbf{B}^{-1}\right) \tag{9.5}$$

$$\epsilon_{ij} \sim \mathcal{N}\left(\epsilon_{ij}|\mathbf{0}, \mathbf{W}^{-1}\right) \tag{9.6}$$

where μ is a speaker independent mean; $\mathbf{B}^{-1}$ is the between-speaker covariance matrix and $\mathbf{W}^{-1}$ is the within-speaker covariance matrix. This is a simplified version of the standard PLDA model where both between and within speaker covariances are of full rank. Because of that, it is also known as full-rank PLDA.

Figure 9.1 shows the graphical model describing an almost fully Bayesian version of the two-covariance model. We say that it is almost fully Bayesian because we put priors over the parameters of the speaker space–mean μ and between-class precision $\mathbf{B}$–but we left the within-class precision $\mathbf{W}$ as a hyperparameter that we optimize by maximum likelihood. The motivation to subject only μ and $\mathbf{B}$ to Bayesian treatment came from the fact that the uncertainty about the speaker space depends on the number of speakers in the development set while the uncertainty of the channel space depends on the number of segments. We will show that in the following sections. In most datasets, for example NIST SRE, the number of development speakers is not very large compared to the i-vector dimension, so the posterior of $\mathcal{M}_{\mathbf{y}}$ may be quite flat. In contrast, the total number of speech segments (and therefore channels) in the development data is an order of magnitude larger, which should give a more peaked posterior for $\mathbf{W}$.

Note that the figure divides the data into two plates. The lower plate corresponds to the development data and the upper plate corresponds to the data of one trial, i.e. the enrollment and test segments. The whole database of development i-vectors is denoted by $\mathbf{\Phi}_{\mathrm{d}}$, while the trial i-vectors are denoted by $\mathbf{\Phi}_{\mathrm{t}}$. We shall also use $\mathbf{\Phi}$ to refer in general to any of both datasets. We assume that the speakers in the trial are not among the speakers in the development set. Let $\mathbf{Y}_{\mathrm{d}}$ and $\mathbf{Y}_{\mathrm{t}}$ respectively denote the hidden speaker identity variables of the development and test sets. $\mathbf{Y}$ can be used to refer to any of them.

The labels θ_{d} partition the development dataset into M_{d} speakers. $\theta_{\mathrm{t}} \in \{\mathcal{T}, \mathcal{N}\}$ is the hidden label of the trial where $\mathcal{T}$ is the hypothesis that the trial i-vectors belong to the same speaker and $\mathcal{N}$ to different speakers. For example, if we have one enrollment i-vector and one test i-vector and the trial is target $M_{\mathrm{t}} = 1$ and $N_{\mathrm{t}_1} = 2$. If the trial is non-target $M_{\mathrm{t}} = 2$ and $N_{\mathrm{t}_1} = N_{\mathrm{t}_2} = 1$. The hyperparameter π denotes the target prior.

Finally, we define $\mathcal{M} = (\mu, \mathbf{B}, \mathbf{W})$ and $\mathcal{M}_{\mathbf{y}} = (\mu, \mathbf{B})$. We have also the prior Π over $\mathcal{M}_{\mathbf{y}}$. We will discuss the selection of informative and non-informative priors in the following sections.

Looking at the graphical model and following the rules that we gave in Section 1.5.2, it is easy to determine whether some pair of variables are conditionally independent given some other set of variables. It is important to note that when $\mathcal{M}$ is given (observed), all variables (hidden and observed) of different speakers, in both development and trial sets, are independent. However, when the model is hidden the variables of all the speakers depend on each other.

9.3 Fully Bayesian Likelihood Ratios

In a non-fully Bayesian framework, the likelihood ratio between the target and non-target hypothesis for a speaker detection trial is computed as:

$$R_{\mathrm{p}}\left(\mathbf{\Phi}_{\mathrm{t}}, \mathcal{M}\right) = \frac{P\left(\mathbf{\Phi}_{\mathrm{t}} | \mathcal{T}, \mathcal{M}\right)}{P\left(\mathbf{\Phi}_{\mathrm{t}} | \mathcal{N}, \mathcal{M}\right)} \tag{9.7}$$

where $\mathbf{\Phi}_{\mathrm{t}}$ are the feature vectors corresponding to the enrollment and test segments, and $\mathcal{M}$ is a point estimate of the PLDA model. The numerator is the likelihood that both speech segments come from the same speaker and the denominator is the likelihood that they come from different speakers. The subscript p is a mnemonic for *plug-in*, because $\mathcal{M}$ is plugged into this formula.

As we pointed out before, the problem is that when the point-estimate is plugged into (9.7), all uncertainty about the values of the model parameters is ignored. The fully

Bayesian solution to this problem [Bishop, 2006] consists in marginalizing $\mathcal{M}$, to form the likelihood ratio:

$$R_{\mathrm{B}}\left(\mathbf{\Phi}_{\mathrm{t}},\mathbf{\Phi}_{\mathrm{d}},\theta_{\mathrm{d}},\Pi\right)=\frac{P\left(\mathbf{\Phi}_{\mathrm{t}}|\mathcal{T},\mathbf{\Phi}_{\mathrm{d}},\theta_{\mathrm{d}},\Pi\right)}{P\left(\mathbf{\Phi}_{\mathrm{t}}|\mathcal{N},\mathbf{\Phi}_{\mathrm{d}},\theta_{\mathrm{d}},\Pi\right)}=\frac{\int P\left(\mathbf{\Phi}_{\mathrm{t}}|\mathcal{T},\mathcal{M}\right)P\left(\mathcal{M}|\mathbf{\Phi}_{\mathrm{d}},\theta_{\mathrm{d}},\Pi\right)\,\mathrm{d}\mathcal{M}}{\int P\left(\mathbf{\Phi}_{\mathrm{t}}|\mathcal{N},\mathcal{M}\right)P\left(\mathcal{M}|\mathbf{\Phi}_{\mathrm{d}},\theta_{\mathrm{d}},\Pi\right)\,\mathrm{d}\mathcal{M}} \tag{9.8}$$

where $P\left(\mathcal{M}|\mathbf{\Phi}_{\mathrm{d}},\theta_{\mathrm{d}},\Pi\right)$ is the posterior of the development parameters given the development data. If there is little uncertainty about $\mathcal{M}$, i.e. when $P\left(\mathcal{M}|\mathbf{\Phi}_{\mathrm{d}},\theta_{\mathrm{d}},\Pi\right)$ is sharply peaked, then the much simpler plug-in recipe R_{p} would suffice for most purposes. But if there is too much model uncertainty, then R_{B} can provide better accuracy. This method has the side-effect that it also helps against dataset shift between development and test databases, because the predictive distributions that result, if you have a small amount of training data, are heavy-tailed. This side-effect disappears if you have a lot of training data, because then the predictive distributions become more Gaussian.

For the case of PLDA models, solving the integrals in (9.8) is intractable. Our approach to approximating R_{B} revolves around the model posterior. Specifically, by using Bayes' rule and the conditional independence assumptions encoded in the graphical model of Figure 9.1, we can express the integrals of (9.8) in terms of the model posterior. First, we use Bayes rule to decompose the joint distribution $P\left(\mathbf{\Phi}_{\mathrm{t}},\mathcal{M}|\theta_{\mathrm{t}},\mathbf{\Phi}_{\mathrm{d}},\theta_{\mathrm{d}},\Pi\right)$ as:

$$\begin{aligned}P\left(\mathbf{\Phi}_{\mathrm{t}},\mathcal{M}|\theta_{\mathrm{t}},\mathbf{\Phi}_{\mathrm{d}},\theta_{\mathrm{d}},\Pi\right)=&P\left(\mathbf{\Phi}_{\mathrm{t}}|\theta_{\mathrm{t}},\mathcal{M},\mathbf{\Phi}_{\mathrm{d}},\theta_{\mathrm{d}},\Pi\right)P\left(\mathcal{M}|\theta_{\mathrm{t}},\mathbf{\Phi}_{\mathrm{d}},\theta_{\mathrm{d}},\Pi\right)\\=&P\left(\mathcal{M}|\mathbf{\Phi}_{\mathrm{t}},\mathbf{\Phi}_{\mathrm{d}},\theta_{\mathrm{t}},\theta_{\mathrm{d}},\Pi\right)P\left(\mathbf{\Phi}_{\mathrm{t}}|\theta_{\mathrm{t}},\mathbf{\Phi}_{\mathrm{d}},\theta_{\mathrm{d}},\Pi\right)\ .\end{aligned} \tag{9.9}$$

We can use the conditional independences arising from the graphical model to simplify (9.9):

$$P\left(\mathbf{\Phi}_{\mathrm{t}}|\theta_{\mathrm{t}},\mathcal{M}\right)P\left(\mathcal{M}|\mathbf{\Phi}_{\mathrm{d}},\theta_{\mathrm{d}},\Pi\right)=P\left(\mathcal{M}|\mathbf{\Phi}_{\mathrm{t}},\mathbf{\Phi}_{\mathrm{d}},\theta_{\mathrm{t}},\theta_{\mathrm{d}},\Pi\right)P\left(\mathbf{\Phi}_{\mathrm{t}}|\theta_{\mathrm{t}},\mathbf{\Phi}_{\mathrm{d}},\theta_{\mathrm{d}},\Pi\right)\ . \tag{9.10}$$

Now, we can isolate $P\left(\mathbf{\Phi}_{\mathrm{t}}|\theta_{\mathrm{t}},\mathbf{\Phi}_{\mathrm{d}},\theta_{\mathrm{d}},\Pi\right)$ from (9.10) and substitute it into (9.8) to obtain:

$$R_{\mathrm{B}}\left(\mathbf{\Phi}_{\mathrm{t}},\mathbf{\Phi}_{\mathrm{d}},\theta_{\mathrm{d}},\Pi\right)=\frac{P\left(\mathbf{\Phi}_{\mathrm{t}}|\mathcal{T},\mathcal{M}\right)}{P\left(\mathbf{\Phi}_{\mathrm{t}}|\mathcal{N},\mathcal{M}\right)}\frac{P\left(\mathcal{M}|\mathbf{\Phi}_{\mathrm{t}},\mathbf{\Phi}_{\mathrm{d}},\mathcal{N},\theta_{\mathrm{d}},\Pi\right)}{P\left(\mathcal{M}|\mathbf{\Phi}_{\mathrm{t}},\mathbf{\Phi}_{\mathrm{d}},\mathcal{T},\theta_{\mathrm{d}},\Pi\right)} \tag{9.11}$$

$$=R_{\mathrm{p}}\left(\mathbf{\Phi}_{\mathrm{t}},\mathcal{M}\right)\frac{P\left(\mathcal{M}|\mathbf{\Phi}_{\mathrm{t}},\mathbf{\Phi}_{\mathrm{d}},\mathcal{N},\theta_{\mathrm{d}},\Pi\right)}{P\left(\mathcal{M}|\mathbf{\Phi}_{\mathrm{t}},\mathbf{\Phi}_{\mathrm{d}},\mathcal{T},\theta_{\mathrm{d}},\Pi\right)}\ . \tag{9.12}$$

If we analyze (9.12), we note that the left hand side (LHS) do not depend on $\mathcal{M}$ while the right hand side (RHS) does. That does not suppose a problem. It just means that the RHS is in fact independent of $\mathcal{M}$ because it simplifies. Thus, we can plug-in whatever value of $\mathcal{M}$ that we find convenient as long as we do not divide by zero.

Equation (9.12) gives an insightful interpretation of the Bayesian likelihood ratio as the plug-in ratio multiplied by a correction factor. We can plug bad estimate of the model $\hat{\mathcal{M}}$ into the formula and the correction factor will compensate for it. But the correction will be noticeable only if the posterior model densities at $\hat{\mathcal{M}}$ are considerably different for the two alternate conditionings. We interpret the correction factor in the following manner. When we plug in a model that excessively favors the target hypotheses of the trial, the denominator of the correction factor becomes larger than the numerator. Thus, it compensates for the over-confidence of the target hypothesis by reducing the total likelihood ratio. On the contrary, if the model favors the non-target hypotheses the numerator will be larger than the denominator. Finally, a fair model should be independent of the trial label, which implies a correction factor equal to one.

For the particular case of the model in Figure 9.1, where we do not marginalize over the within-class precision $\mathbf{W}$, the Bayesian likelihood ratio is written as:

$$R_{\mathrm{B}}\left(\mathbf{\Phi}_{\mathrm{t}},\mathbf{\Phi}_{\mathrm{d}},\theta_{\mathrm{d}},\mathbf{W},\Pi\right)=R_{\mathrm{p}}\left(\mathbf{\Phi}_{\mathrm{t}},\mathcal{M}_{\mathbf{y}},\mathbf{W}\right)\frac{P\left(\mathcal{M}_{\mathbf{y}}|\mathbf{\Phi}_{\mathrm{t}},\mathbf{\Phi}_{\mathrm{d}},\mathcal{N},\theta_{\mathrm{d}},\mathbf{W},\Pi\right)}{P\left(\mathcal{M}_{\mathbf{y}}|\mathbf{\Phi}_{\mathrm{t}},\mathbf{\Phi}_{\mathrm{d}},\mathcal{T},\theta_{\mathrm{d}},\mathbf{W},\Pi\right)}\;. \tag{9.13}$$

We have now transformed the problem of calculating integrals over model parameter into one of calculating model posteriors. Unfortunately, even for the simple two-covariance model, these posteriors cannot be expressed in closed form. We proposed to use a variational Bayes (VB) procedure to calculate approximate posteriors. In the next sections, we present the VB solutions for the two-covariance model assuming two different types of model priors: non-informative and informative.

9.4 Variational Inference with Non-Informative Priors

9.4.1 Non-informative priors

As first approximation, we assumed a non-informative prior (Jeffreys prior) for the parameters μ and $\mathbf{B}$ of the speaker space distribution (see Appendix A). A non-informative prior encodes the absence of information about μ and $\mathbf{B}$ other than the training data. With this prior no Gaussian should be preferred over others and it should be invariant to any translation or scaling of the measurement space. These conditions are satisfied by this distribution:

$$\begin{aligned} P\left(\mu,\mathbf{B}|\Pi\right) &= P\left(\mu|\mathbf{B},\Pi\right)P\left(\mathbf{B}|\Pi\right) & (9.14)\\ &= \lim_{k\to 0}\mathcal{N}\left(\mu|\mu_0,(k\mathbf{B})^{-1}\right)\mathcal{W}\left(\mathbf{B}|\mathbf{W}_0/k,k\right) & (9.15)\\ &= \alpha\left|\frac{\mathbf{B}}{2\pi}\right|^{1/2}|\mathbf{B}|^{-(d+1)/2} & (9.16)\end{aligned}$$

where $\mathcal{W}$ denotes a Wishart distribution and d the i-vector dimension. Since this density does not integrate to 1, it is improper and the symbol α is used to denote a normalizing constant which approaches zero. Note that the fact that the prior is improper does not mean that the posterior will be improper.

9.4.2 Variational Bayes likelihood ratio

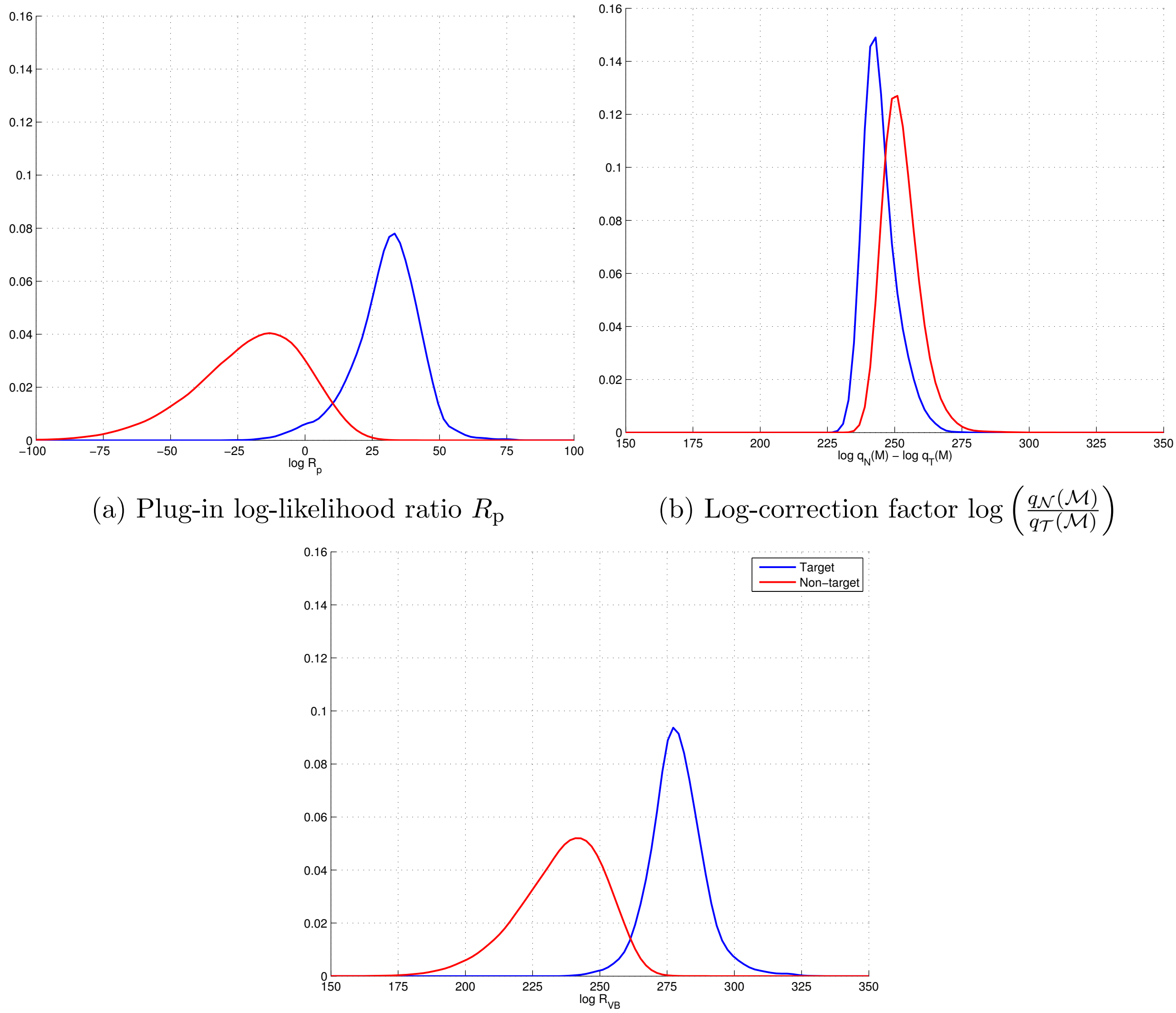
Our VB solution approximates the joint posterior distribution for the hidden identity variables and model parameters by a factorized distribution of the form:

$$P\left(\mathcal{M}_{\mathbf{y}},\mathbf{Y}_{\mathrm{t}},\mathbf{Y}_{\mathrm{d}}|\mathbf{\Phi}_{\mathrm{t}},\mathbf{\Phi}_{\mathrm{d}},\theta_{\mathrm{t}},\theta_{\mathrm{d}},\mathbf{W},\Pi\right)\approx q\left(\mathcal{M}_{\mathbf{y}},\mathbf{Y}_{\mathrm{t}},\mathbf{Y}_{\mathrm{d}}\right)=q\left(\mathcal{M}_{\mathbf{y}}\right)q\left(\mathbf{Y}_{\mathrm{t}},\mathbf{Y}_{\mathrm{d}}\right) \tag{9.17}$$

which ignores any posterior dependencies between the speaker variables $\mathbf{Y}$ and the model $\mathcal{M}_{\mathbf{y}}$. Note that we are not making further factorizing assumptions or restricting the functional form of the individual factors.

We complete our recipe by using the approximation $P\left(\mathcal{M}_{\mathbf{y}}|\mathbf{\Phi}_{\mathrm{t}},\mathbf{\Phi}_{\mathrm{d}},\theta_{\mathrm{t}},\theta_{\mathrm{d}},\Pi\right)\approx q\left(\mathcal{M}_{\mathbf{y}}\right)$ in (9.13), so that the VB approximation of the likelihood ratio becomes

$$R_{\mathrm{VB}}\left(\mathbf{\Phi}_{\mathrm{t}},\mathbf{\Phi}_{\mathrm{d}},\theta_{\mathrm{d}},\mathbf{W},\Pi\right)=R_{\mathrm{p}}\left(\mathbf{\Phi}_{\mathrm{t}},\mathcal{M}_{\mathbf{y}},\mathbf{W}\right)\frac{q_{\mathcal{N}}\left(\mathcal{M}_{\mathbf{y}}\right)}{q_{\mathcal{T}}\left(\mathcal{M}_{\mathbf{y}}\right)} \tag{9.18}$$



(a) Plug-in log-likelihood ratio R_{p}

(b) Log-correction factor $\log\left(\frac{q_{\mathcal{N}}(\mathcal{M})}{q_{\mathcal{T}}(\mathcal{M})}\right)$

(c) VB log-likelihood ratio R_{p}

Figure 9.2: Distributions of the log-likelihood ratios involved in the calculus of the approximated fully Bayesian log-likelihood ratio.

where $q_{\mathcal{T}}\left(\mathcal{M}_{\mathbf{y}}\right)$ and $q_{\mathcal{N}}\left(\mathcal{M}_{\mathbf{y}}\right)$ are the variational posteriors conditioned respectively on $\theta_{\mathrm{t}} = \mathcal{T}$ and $\theta_{\mathrm{t}} = \mathcal{N}$.

As this is just an approximation, the property that the likelihood ratio does not depend on the plug-in model $\hat{\mathcal{M}}_{\mathbf{y}}$ because it simplifies is no longer true. In our experiments we observed that plugging-in the maximum likelihood point estimate provided good results.

Figure 9.2 displays an example of the distributions of the likelihood ratios involved in the calculus of R_{VB}. The log-correction factor

$$\log\left(\frac{q_{\mathcal{N}}\left(\mathcal{M}_{\mathbf{y}}\right)}{q_{\mathcal{T}}\left(\mathcal{M}_{\mathbf{y}}\right)}\right) \approx \log\left(\frac{P\left(\mathcal{M}_{\mathbf{y}}|\mathbf{\Phi}_{\mathrm{t}},\mathbf{\Phi}_{\mathrm{d}},\mathcal{N},\theta_{\mathrm{d}},\mathbf{W},\Pi\right)}{P\left(\mathcal{M}_{\mathbf{y}}|\mathbf{\Phi}_{\mathrm{t}},\mathbf{\Phi}_{\mathrm{d}},\mathcal{T},\theta_{\mathrm{d}},\mathbf{W},\Pi\right)}\right) \tag{9.19}$$

exhibits a counterintuitive behavior. We could think that, to improve the plug-in ratio, the distribution of the correction factor of the targets trials should be to the right of the one of the non-target trials. However, we observe the opposite. We interpret this by assuming that our plug-in model, in general, favored the target decision. That makes the correction factor to reduce the ratios of the target trials and to increase the ratios of the non-targets. Thus, in the example, the separability of the target and non-target distributions improved. The Fisher ratio between both distributions increased from 4.45 for R_{p} to 5.13 for R_{VB}.

9.4.3 Variational distributions

According to variational Bayes theory [Bishop, 2006], given a set of visible variables $\mathbf{X}$ and a set of hidden variables $\mathbf{Z}$, if we approximate the posterior of $\mathbf{Z}$ by a factorized distribution

$$q(\mathbf{Z}) = \prod_{k=1}^{K} q_j(\mathbf{Z}_j) \ , \tag{9.20}$$

the optimum value for the $q_j(\mathbf{Z}_j)$ is given by the following equation:

$$\ln q_j^*(\mathbf{Z}_j) = \mathrm{E}_{i\neq j}\left[\ln P(\mathbf{X},\mathbf{Z})\right] + \mathrm{const} \ . \tag{9.21}$$

This equation means that the log of the optimum solution for factor q_j is estimated by taking the expectation of the log joint distribution over all hidden and visible variables with respect to the rest of factors $q_{i\neq j}$. The additive constant is needed to normalize the distribution to integrate to one.

VB is an iterative procedure. We first initialize the factors and then cycle through the factors $q_j(\mathbf{Z}_j)$ for $j = 1,\dots,K$ re-estimating each one with (9.21) until convergence.

To keep the notation uncluttered and without loss of generality, during the rest of the section we will abbreviate $\mathbf{\Phi} = (\mathbf{\Phi}_{\mathrm{t}}, \mathbf{\Phi}_{\mathrm{d}})$, $\mathbf{Y} = (\mathbf{Y}_{\mathrm{t}}, \mathbf{Y}_{\mathrm{d}})$ and $\theta = (\theta_{\mathrm{t}}, \theta_{\mathrm{d}})$.

In our Bayesian two-covariance model the joint distribution of all variables is given by

$$P(\mathbf{\Phi}, \mathcal{M}_{\mathbf{y}}, \mathbf{Y}|\theta, \mathbf{W}, \Pi) = P(\mathbf{\Phi}|\mathbf{Y}, \theta, \mathbf{W})\, P(\mathbf{Y}|\mathcal{M}_{\mathbf{y}})\, P(\mathcal{M}_{\mathbf{y}}|\Pi) \tag{9.22}$$

Now, applying (9.21), it is straightforward to obtain our variational distributions. In Appendix E, we show derivations for the equations in this section.

The optimum for the factor $q(\mathbf{Y})$ is given by a product of Gaussian distributions:

$$q^*(\mathbf{Y}) = \prod_{i=1}^{M} q^*(\mathbf{y}_i) = \prod_{i=1}^{M} \mathcal{N}\left(\mathbf{y}_i|\mathbf{L}_i^{-1}\gamma_i, \mathbf{L}_i^{-1}\right) \tag{9.23}$$

$$\mathbf{L}_i = \mathrm{E}_{\mathcal{M}_{\mathbf{y}}}[\mathbf{B}] + N_i\mathbf{W} \tag{9.24}$$

$$\gamma_i = \mathrm{E}_{\mathcal{M}_{\mathbf{y}}}[\mathbf{B}\mu] + \mathbf{W}\mathbf{F}_i \tag{9.25}$$

where N_i is the number of samples of speaker i and $\mathbf{F}_i = \sum_{j=1}^{N_i}\phi_{ij}$ are the first order sufficient statistics. The speaker identity variables $\mathbf{y}_i$ result independent a posteriori one from the other. Note that we have not forced that in any way but it originates naturally from the original factorization that we chose.

The optimum for the factor $q(\mathcal{M}_{\mathbf{y}})$ is a Gaussian-Wishart distribution.

$$q^*(\mathcal{M}_{\mathbf{y}}) = \mathcal{N}\left(\mu|\overline{\mu}, (M\mathbf{B})^{-1}\right)\mathcal{W}(\mathbf{B}|\mathbf{\Psi}, M) \quad \text{if } M > d \tag{9.26}$$

where we defined

$$\overline{\mu} = \frac{1}{M}\sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}[\mathbf{y}_i] \tag{9.27}$$

$$\mathbf{\Psi}^{-1} = \sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right] - M\overline{\mu}\overline{\mu}^T \ . \tag{9.28}$$

We have to remark that for this distribution to be proper we need the number of speakers M to be larger than the i-vectors dimensionality. Equation (9.26) evidences that M controls the width of the model posterior. The variance of μ reduces as M increases. Besides, it is also known that the degrees of freedom of the Wishart distribution control its width. Thus, as we mention previously, less speakers means a larger uncertainty about the model parameters.

In order to obtain the parameters of $q(\mathbf{Y})$, we need yet to evaluate $\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}[\mathbf{B}]$ and $\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}[\mathbf{B}\mu]$. Using the properties of the Wishart distribution [Bishop, 2006] we have

$$\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}[\mathbf{B}] = M\mathbf{\Psi} \qquad \mathrm{E}_{\mathcal{M}_{\mathbf{y}}}[\mathbf{B}\mu] = M\mathbf{\Psi}\overline{\mu}\ . \tag{9.29}$$

9.5 Variational Inference with Informative Priors

9.5.1 Approximation with informative priors

The approach taken in the previous section, in which we use all the development data to compute the likelihood ratio of each trial, has a large computational cost. For each trial, we need to evaluate several VB iterations and in each one of them we have to re-estimate the $q(\mathbf{y}_i)$ distributions for all the development speakers. In this section, we make a further approximation, by effectively fixing these $q(\mathbf{y}_i)$. This is achieved by first computing the VB posterior for $\mathcal{M}_{\mathbf{y}}$, conditioned only on the development data ($\mathbf{\Phi}_{\mathrm{d}}$,$\theta_{\mathrm{d}}$). In other words, the test data is not involved. Then we use this posterior to act as the prior, denoted by $P(\mathcal{M}_{\mathbf{y}}|\Pi_{\mathrm{d}})$, for further processing of the test data. Now for every trial, only the test i-vectors $\mathbf{\Phi}_{\mathrm{t}}$ are involved in the calculation of the likelihood ratio. This idea relies on the assumption that adding the trial data should not modify the posteriors of the development speaker identity variables significantly. Most development speakers have a large number of segments, so that the speaker identity posteriors are less affected by changes in the value of $\mathcal{M}_{\mathbf{y}}$. In contrast, test speakers, in many applications, have at most two segments. In summary, our Bayesian likelihood ratio is approximated as:

$$R_{\mathrm{B}}(\mathbf{\Phi}_{\mathrm{t}},\mathbf{\Phi}_{\mathrm{d}},\theta_{\mathrm{d}},\mathbf{W},\Pi) \approx R_{\mathrm{B}}(\mathbf{\Phi}_{\mathrm{t}},\mathbf{W},\Pi_{\mathrm{d}}) = R_{\mathrm{p}}(\mathbf{\Phi}_{\mathrm{t}},\mathcal{M}_{\mathbf{y}},\mathbf{W})\,\frac{P(\mathcal{M}_{\mathbf{y}}|\mathbf{\Phi}_{\mathrm{t}},\mathcal{N},\mathbf{W},\Pi_{\mathrm{d}})}{P(\mathcal{M}_{\mathbf{y}}|\mathbf{\Phi}_{\mathrm{t}},\mathcal{T},\mathbf{W},\Pi_{\mathrm{d}})}\ . \tag{9.30}$$

The model prior is now the variational posterior of the model given only the development data, which approximates the model posterior given the development data:

$$P(\mathcal{M}_{\mathbf{y}}|\Pi_{\mathrm{d}}) = q_{\mathrm{d}}(\mathcal{M}_{\mathbf{y}}) \approx P(\mathcal{M}_{\mathbf{y}}|\mathbf{\Phi}_{\mathrm{d}},\theta_{\mathrm{d}},\mathbf{W},\Pi)\ . \tag{9.31}$$

Thus, the VB factor $q_{\mathrm{d}}(\mathcal{M}_{\mathbf{y}})$ is Gaussian-Wishart distributed, which is a conjugate prior for the Gaussian distribution. As shown in (9.26), it is given by

$$q_{\mathrm{d}}(\mathcal{M}_{\mathbf{y}}) = \mathcal{N}\left(\mu|\overline{\mu}_{\mathrm{d}},(\beta_{\mathrm{d}}\mathbf{B})^{-1}\right)\mathcal{W}(\mathbf{B}|\mathbf{\Psi}_{\mathrm{d}},\nu_{\mathrm{d}}) \tag{9.32}$$

where

$$\overline{\mu}_{\mathrm{d}} = \frac{1}{M_{\mathrm{d}}}\sum_{i=1}^{M_{\mathrm{d}}}\mathrm{E}_{\mathbf{Y}}[\mathbf{y}_i] \tag{9.33}$$

$$\mathbf{\Psi}_{\mathrm{d}}^{-1} = \sum_{i=1}^{M_{\mathrm{d}}}\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right] - M_{\mathrm{d}}\overline{\mu}_{\mathrm{d}}\overline{\mu}_{\mathrm{d}}^T \tag{9.34}$$

and $\beta_{\mathrm{d}} = \nu_{\mathrm{d}} = M_{\mathrm{d}} > d$. Summations are now over the development speakers M_{d} only.

From this prior, now we proceed as in the non-informative case. We factorize the joint posterior of the latent variables as:

$$P\left(\mathcal{M}_{\mathbf{y}}, \mathbf{Y}_{\mathrm{t}} | \mathbf{\Phi}_{\mathrm{t}}, \theta_{\mathrm{t}}, \mathbf{W}, \Pi_{\mathrm{d}}\right) \approx q\left(\mathcal{M}_{\mathbf{y}}, \mathbf{Y}_{\mathrm{t}}\right) = q\left(\mathcal{M}_{\mathbf{y}}\right) q\left(\mathbf{Y}_{\mathrm{t}}\right) \ . \tag{9.35}$$

Then, we approximate $P\left(\mathcal{M}_{\mathbf{y}} | \mathbf{\Phi}_{\mathrm{t}}, \theta_{\mathrm{t}}, \mathbf{W}, \Pi_{\mathrm{d}}\right) \approx q\left(\mathcal{M}_{\mathbf{y}}\right)$ and compute the likelihood ratio as:

$$R_{\mathrm{VB}}\left(\mathbf{\Phi}_{\mathrm{t}}, \mathbf{W}, \Pi_{\mathrm{d}}\right) = R\left(\mathbf{\Phi}_{\mathrm{t}}, \mathcal{M}_{\mathbf{y}}, \mathbf{W}\right) \frac{q_{\mathcal{N}}\left(\mathcal{M}_{\mathbf{y}}\right)}{q_{\mathcal{T}}\left(\mathcal{M}_{\mathbf{y}}\right)} \tag{9.36}$$

where $q_{\mathcal{T}}\left(\mathcal{M}_{\mathbf{y}}\right)$ and $q_{\mathcal{N}}\left(\mathcal{M}_{\mathbf{y}}\right)$ are the variational posteriors conditioned respectively on $\theta_{\mathrm{t}} = \mathcal{T}$ and $\theta_{\mathrm{t}} = \mathcal{N}$.

9.5.2 Variational distributions

The variational distributions are calculated in a similar way to section 9.4.3. In fact, the optimum for the factor $q\left(\mathbf{Y}\right)$ is the same as in the non-informative case. The optimum for the factor $q\left(\mathcal{M}_{\mathbf{y}}\right)$ is again Gaussian-Wishart:

$$q^*\left(\mathcal{M}_{\mathbf{y}}\right) = \mathcal{N}\left(\mu | \overline{\mu}, \left(\beta \mathbf{B}\right)^{-1}\right) \mathcal{W}\left(\mathbf{B} | \mathbf{\Psi}, \nu\right) \tag{9.37}$$

where

$$\overline{\mathbf{y}} = \frac{1}{M_{\mathrm{t}}} \sum_{i=1}^{M_{\mathrm{t}}} \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right] \tag{9.38}$$

$$\mathbf{S}_{\mathbf{y}} = \sum_{i=1}^{M_{\mathrm{t}}} \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i \mathbf{y}_i^T\right] - M_{\mathrm{t}} \overline{\mathbf{y}}\overline{\mathbf{y}}^T \tag{9.39}$$

$$\beta = \beta_{\mathrm{d}} + M_{\mathrm{t}} \tag{9.40}$$

$$\nu = \nu_{\mathrm{d}} + M_{\mathrm{t}} \tag{9.41}$$

$$\overline{\mu} = \frac{1}{\beta}\left(\beta_{\mathrm{d}} \overline{\mu}_{\mathrm{d}} + M_{\mathrm{t}} \overline{\mathbf{y}}\right) \tag{9.42}$$

$$\mathbf{\Psi}^{-1} = \mathbf{\Psi}_{\mathrm{d}}^{-1} + \mathbf{S}_{\mathbf{y}} + \frac{\beta_{\mathrm{d}} M_{\mathrm{t}}}{\beta}\left(\overline{\mathbf{y}} - \overline{\mu}_{\mathrm{d}}\right)\left(\overline{\mathbf{y}} - \overline{\mu}_{\mathrm{d}}\right)^T \ . \tag{9.43}$$

In this case, summations are only over the trial speakers that can be one or two depending on whether we assume the target or the non-target hypothesis. A detailed derivation of these equations can be found in Appendix E.

9.6 Experimental Setup

9.6.1 SV system configuration

We performed experiments on all the conditions of NIST SRE 2010 except the summed channel ones. We present results in terms of EER and minimum DCF. The DCF was calculated for the NIST SRE10 operating point ($C_{\mathrm{Miss}} = 1$, $C_{\mathrm{FA}} = 1$, $P_{\mathcal{T}} = 0.001$) for the core, coreext and 8conv-coreext conditions; and at the NIST SRE08 operating point ($C_{\mathrm{Miss}} = 10$, $C_{\mathrm{FA}} = 1$, $P_{\mathcal{T}} = 0.01$) for the rest of them.

We used 400 dimensional i-vectors as features for our PLDA. They were extracted using 20 short-time Gaussianized MFCC plus deltas and double deltas and a 2048 component diagonal covariance UBM. The UBM, the i-vector extractor and the two-covariance model were gender dependent and they were trained on telephone data from NIST SRE04–06.

We conducted experiments with the Bayesian and the *plug-in* likelihood ratio with different preprocessing of the i-vectors. We present results with the plain i-vectors; length normalized i-vectors; and length normalized i-vectors followed by LDA to reduce dimensionality to 90. The LDA transform and the centering and whitening parameters were trained on the same development data as the two-covariance model.

9.6.2 i-Vector length normalization

i-Vector length normalization [Garcia-Romero and Espy-Wilson, 2011], presented in Section 2.4.4, prevents dataset shift by normalizing each i-vector by its magnitude. It makes the development and trial i-vector distributions closer and more Gaussian shaped. The Bayesian method has the side-effect that it also helps against dataset shift between training and test databases, because the predictive distributions that result, if you have a small amount of training data, are heavy-tailed. This side-effect disappears if you have a lot of training data, because then the distributions become more Gaussian.

We wanted to know if both approaches can be complementary: fully Bayesian combats over-fitting, while length norm combats bad modeling assumptions and dataset shift. However, if we make the training and test distributions closer, the problem of model over-fitting should not be so harmful. We compared the performances of both the length normalization and the Bayesian solution.

9.6.3 Speeding-up matrix inversions

The main disadvantage of the Bayesian approach is the large computational cost of the VB procedure needed to approximate the likelihood ratio. However, for the case with informative priors, we can speed up the algorithm by using the Woodbury matrix identity [Woodbury, 1950]–also known as matrix inversion lemma–:

$$(\mathbf{D}+\mathbf{U}\mathbf{V})^{-1} = \mathbf{D}^{-1} - \mathbf{D}^{-1}\mathbf{U}(\mathbf{I}+\mathbf{V}\mathbf{D}^{-1}\mathbf{U})^{-1}\mathbf{V}\mathbf{D}^{-1} \tag{9.44}$$

to calculate the matrix inversions involved. If $\mathbf{D}$ is diagonal computing its inverse is straightforward and if $\mathbf{U}$ and $\mathbf{V}$ are low-rank computing the inverse of $\mathbf{I}+\mathbf{V}\mathbf{D}^{-1}\mathbf{U}$ is much faster than computing the inverse of $\mathbf{D}+\mathbf{U}\mathbf{V}$. For example, let us suppose that we have $\mathbf{U}$ of size $d \times 2$ and $\mathbf{V}$ of size $2 \times d$, where d is the i-vector dimension. If we invert the matrix in the standard way, we need to invert a $d \times d$ matrix while by applying the identity we just invert $\mathbf{I}+\mathbf{V}\mathbf{D}^{-1}\mathbf{U}$ of size 2×2.

If we apply a preprocessing to the i-vectors consisting of linear discriminant analysis (LDA) plus within class covariance normalization (WCCN), the $\mathbf{B}$ and $\mathbf{W}$ matrices that we estimate on the pre-processed i-vectors become diagonal and identity respectively. Thus, we obtain that, in the first VB update of $q(\mathbf{Y})$, all $\mathbf{L}_i$ (see (9.24)) are also diagonal. Then,

$\mathbf{S}_{\mathbf{y}}$ in (9.39) becomes

$$\mathbf{S}_{\mathbf{y}} = \sum_{i=1}^{M} \mathbf{L}_i^{-1} + \mathrm{E}\left[\mathbf{y}_i\right] \mathrm{E}\left[\mathbf{y}_i\right]^T - M\overline{\mathbf{y}}\overline{\mathbf{y}}^T = \left[\sum_{i=1}^{M} \mathbf{L}_i^{-1}\right] + \left[\mathrm{E}\left[\mathbf{y}_1\right] \quad \ldots \quad \mathrm{E}\left[\mathbf{y}_M\right] \quad -M\overline{\mathbf{y}}\right] \begin{bmatrix} \mathrm{E}\left[\mathbf{y}_1\right] \\ \vdots \\ \mathrm{E}\left[\mathbf{y}_M\right] \\ \overline{\mathbf{y}} \end{bmatrix} , \tag{9.45}$$

that is a diagonal matrix plus a product of matrices of rank $M+1$. For the informative case $M \in \{1,2\} << d$, so we can speed-up matrix inversion by applying the lemma. For the non-informative case $M > d$ and we do not obtain any gain by applying the lemma. Following with the informative case, we substitute (9.45) into (9.43) to obtain:

$$\mathbf{\Psi}^{-1} = \left[\mathbf{\Psi}_{\mathrm{d}}^{-1} + \sum_{i=1}^{M} \mathbf{L}_i^{-1}\right] + \left[\mathrm{E}\left[\mathbf{y}_1\right] \quad \ldots \quad \mathrm{E}\left[\mathbf{y}_M\right] \quad -M\overline{\mathbf{y}} \quad \tfrac{\beta_{\mathrm{d}} M_{\mathrm{t}}}{\beta}\left(\overline{\mathbf{y}} - \overline{\mu}_{\mathrm{d}}\right)\right] \begin{bmatrix} \mathrm{E}\left[\mathbf{y}_1\right] \\ \vdots \\ \mathrm{E}\left[\mathbf{y}_M\right] \\ \overline{\mathbf{y}} \\ \overline{\mathbf{y}} - \overline{\mu}_{\mathrm{d}} \end{bmatrix} . \tag{9.46}$$

The matrix $\mathbf{\Psi}_{\mathrm{d}} = \mathbf{B}_0/M$, where $\mathbf{B}_0$ is the expected value of $\mathbf{B}$ given the development set. As we said above, after the i-vector preprocessing $\mathbf{B}_0$ is diagonal, so $\mathbf{\Psi}_{\mathrm{d}}^{-1}$ is diagonal and $\mathbf{\Psi}^{-1}$ is also the sum of a diagonal term and the product of two matrices of low rank $(M+2)$.

In the second iteration, we need to invert $\mathbf{\Psi}^{-1}$ to compute $\mathrm{E}\left[\mathbf{B}\right] = \nu\mathbf{\Psi}$ and with it, we update the distributions $q\left(\mathbf{Y}\right)$. We invert $\mathbf{\Psi}^{-1}$ by applying the lemma. We can realize that the inverse given by the lemma is also a diagonal matrix $(\mathbf{D}^{-1})$ plus a product of two low-rank matrices $(\mathbf{D}^{-1}\mathbf{U}$ and $-(\mathbf{I}+\mathbf{V}\mathbf{D}^{-1}\mathbf{U})^{-1}\mathbf{V}\mathbf{D}^{-1})$. Thus, if we continue adding low-rank products and diagonal matrices to it, we can also invert the resulting matrices by the same lemma. As all the matrices that we need to invert ($\mathbf{L}_i$ and $\mathbf{\Psi}^{-1}$) meet this condition, we can use the lemma to speed up the matrix inversions in all the VB iterations. For i-vectors of dimension 400, this method reduced the processing time by a factor of 6.

9.7 Experiment Results

9.7.1 Informative against non-informative priors

Table 9.1 compares the non Bayesian and Bayesian approaches on the NIST SRE10 core det5 condition (telephone vs. telephone, English, normal vocal effort). We did not use the extended condition because of the high computational cost of the Bayesian approach based on non-informative priors, where we need all the development data to compute the likelihood ratio of each trial. In this experiment, we did not apply any score normalization. The Bayesian approach dramatically improved EER, by around 50%. Improvement in the DCF operating point was not so evident, we observed some improvement for males–15% for non-inf. priors and 9% for inf.–but not for females–it worsened by 31% and 9% for non-inf. and inf. priors respectively.

The relative difference between using informative and non-informative priors was as small as 7% except for the female DCF where non-informative priors where 19% worse than

Table 9.1: EER(%)/MinDCF Bayesian two-covariance model with informative and non-informative priors on NIST SRE10 core det5.

System	male		female	
	EER	DCF	EER	DCF
2cov	5.42	0.46	4.35	**0.42**
Bay2cov non-inf. prior	2.68	**0.39**	**2.13**	0.55
Bay2cov inf. prior	**2.52**	0.42	2.30	0.46

informative ones. As consequence, we focused on the case with informative priors in the rest of experiments.

9.7.2 Score normalization analysis

Table 9.2 compares results with and without score normalization on the NIST SRE10 core det5 condition. We used S-Norm [Senoussaoui et al., 2010] with cohort utterances from SRE04 to SRE06 (1599 male, 2530 female). Again, we did not experiment on the extended condition because of the high cost of computing the score matrices of the enrollment and test i-vectors against all the cohort speakers, which S-Norm requires. For non length normalized i-vectors, score normalization improved the EER of the plug-in ratio. In the rest of cases, score normalization was harmful. For the Bayesian ratio without length normalization, score normalization worsened EER by around 30% and DCF by around 60%. For systems with

Table 9.2: EER(%)/MinDCF Bayesian two-covariance model with and without S-Norm on NIST SRE10 core det5.

System	male		female	
	EER	DCF	EER	DCF
Without S-Norm				
2cov	5.42	0.46	4.35	**0.42**
Bay2cov	2.52	**0.42**	2.30	0.46
LNorm+2cov	1.56	0.45	2.48	0.48
LNorm+Bay2cov	**1.53**	0.44	**2.10**	0.43
LNorm+LDA90+2cov	1.84	0.49	2.11	0.54
LNorm+LDA90+Bay2cov	1.97	0.50	2.13	0.57
With S-Norm				
2cov	3.14	0.59	2.83	0.72
Bay2cov	2.59	0.66	3.02	0.75
LNorm+2cov	**2.12**	**0.51**	2.60	**0.49**
LNorm+Bay2cov	2.34	0.53	2.63	0.52
LNorm+LDA90+2cov	2.78	0.55	**2.46**	0.61
LNorm+LDA90+Bay2cov	3.05	0.58	2.57	0.65

length normalization male EER worsened by 35–55%, female EER by 16–25% and DCF by 12–20%. The fact that score normalization damages the performance of length normalized i-vectors was already noted in Section 2.8.2.2.

If we compare length normalization against the Bayesian approach, we observe that length normalization obtained better EER for males and the Bayesian approach for females. In terms of DCF the difference between the Bayesian ratio and length normalization was not very significant (4–6%). We obtained the best result by combining both.

We did not benefit from dimensionality reduction, contrary to the findings in [Matejka et al., 2011]. Just for the system LNorm+LDA90+2cov, the female EER was better than the one of the corresponding system without LDA. We think that, since Bayesian approach reduces the risk of over-fitting, it allows us to better extract information from the 400 dimensional i-vectors. Therefore, the reduction of dimensionality should not be necessary.

9.7.3 Results core extended condition

Table 9.3 displays results for the NIST SRE10 core extended condition. The extended condition had more than 10 times the number of trials of the non-extended. As the operating point of SRE10 favored working in the low false alarm region of the DET curve, NIST introduced the extended condition to accurately measure very low false alarm rates. This decision was based on Doddington's "rule of 30" [Doddington, 2000], that states that we need at least 30 errors to be confident about the error rate.

As we explained in Chapter 2, the core condition included multiple types of enrollment and test segments so measuring performance by pooling all the trials together was not appropriate. Thus, NIST defined several common conditions grouping different types of trials. We experimented on the common conditions with normal vocal effort: interview against interview speech from the same microphone (det1), interview against interview speech from different microphones (det2), interview against telephone speech (det3), interview against telephone speech recorded over a room microphone (det4) and telephone against telephone speech (det5).

We first focus on the conditions with far-field microphones for both enrollment and test (det1, det2 and det4). The Bayesian likelihood ratio improved the baseline by an average 29% in terms of EER and by 2.5% in terms of minimum DCF. Comparing length normalization against the Bayesian approach, we found that the Bayesian method yielded better EER and DCF for all cases. The Bayesian ratio outperformed length normalization by an average 23% in terms of EER and by 5% in terms of DCF. Since the two-covariance models and the i-vector extractor were trained on telephone speech only, there was an important mismatch between the development and the trial data. It seems that the Bayesian approach coped with it better than the length normalization. Combining both approaches did not produce large performance differences with respect to employing the Bayesian approach only, in some cases it was better and in others it was worse. In average, adding length normalization to the Bayesian ratio worsened EER by 3.7% and improved DCF by 8%. Contrary to what happened in det5, dimensionality reduction improved performance for the Bayesian and non-Bayesian systems. In most cases the best performance was achieved for the system that combined length normalization, dimensionality reduction and Bayesian scoring. For example, in det4, EER improved by 22% in the version with dimensionality reduction with respect to the one without it.

For the telephone condition (det5) the results of the extended condition were similar

Table 9.3: EER(%)/MinDCF Bayesian two-covariance model on NIST SRE10 core extended common conditions 1-5.

CC	System	male		female	
		EER	DCF	EER	DCF
1	2cov	1.89	0.30	3.73	0.53
	Bay2cov	1.43	0.34	**2.81**	0.50
	LNorm+2cov	1.80	0.30	3.70	0.56
	LNorm+Bay2cov	1.31	0.26	3.34	**0.47**
	LNorm+LDA90+2cov	1.47	**0.25**	2.82	**0.47**
	LNorm+LDA90+Bay2cov	**1.15**	0.30	3.01	0.51
2	2cov	4.16	0.64	8.87	0.87
	Bay2cov	2.72	0.56	6.89	**0.83**
	LNorm+2cov	3.55	0.63	8.43	0.88
	LNorm+Bay2cov	2.73	**0.53**	7.39	**0.83**
	LNorm+LDA90+2cov	2.97	0.54	6.76	0.84
	LNorm+LDA90+Bay2cov	**2.59**	0.56	**6.63**	0.84
3	2cov	6.41	0.69	6.30	0.80
	Bay2cov	3.50	**0.64**	4.12	**0.66**
	LNorm+2cov	3.02	0.67	4.34	0.75
	LNorm+Bay2cov	**2.64**	**0.64**	**3.97**	0.71
	LNorm+LDA90+2cov	3.38	0.72	5.08	0.77
	LNorm+LDA90+Bay2cov	2.79	0.69	4.57	0.76
4	2cov	3.59	0.50	6.19	0.68
	Bay2cov	2.48	0.50	3.94	0.63
	LNorm+2cov	3.23	0.54	5.61	0.71
	LNorm+Bay2cov	2.63	0.43	4.60	0.63
	LNorm+LDA90+2cov	2.28	**0.37**	3.66	**0.60**
	LNorm+LDA90+Bay2cov	**2.01**	0.39	**3.62**	0.65
5	2cov	4.97	0.52	5.57	0.55
	Bay2cov	2.66	0.38	3.16	0.56
	LNorm+2cov	2.04	0.38	3.20	0.57
	LNorm+Bay2cov	**1.82**	**0.37**	2.95	**0.54**
	LNorm+LDA90+2cov	2.20	0.42	3.02	**0.54**
	LNorm+LDA90+Bay2cov	2.11	0.41	**2.88**	0.56

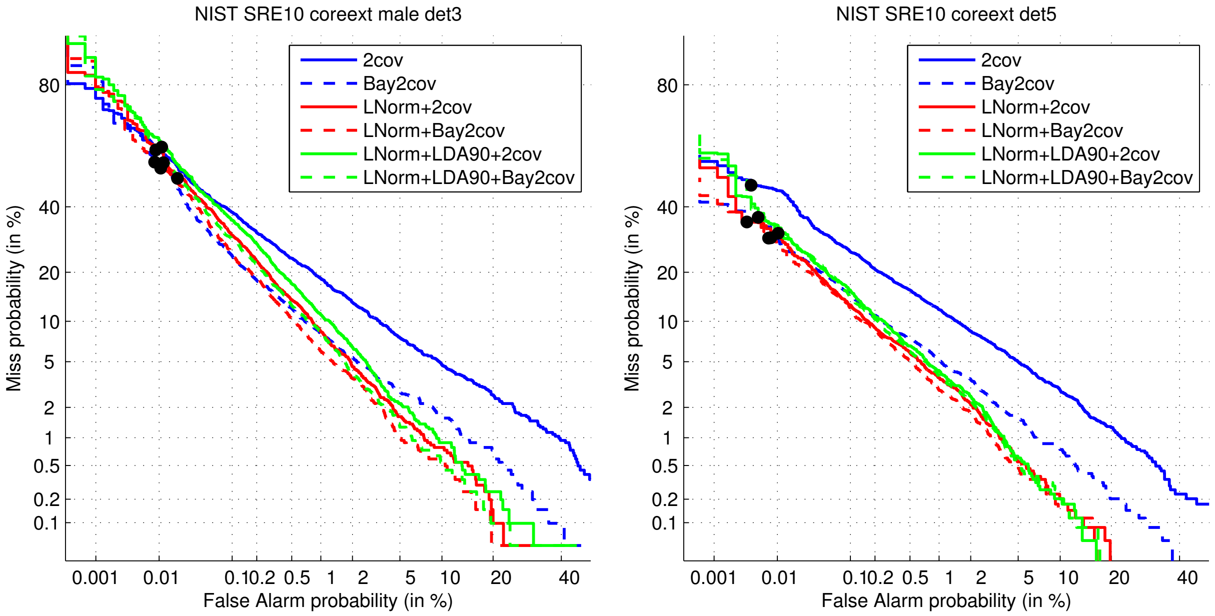


Figure 9.3: DET curves for the Bayesian two-covariance model on NIST SRE10 core extended male common conditions 3 and 5.

to the ones of the non-extended, reported in the previous section. Again, the best system was the one combining length normalization and Bayesian likelihood ratio without i-vector dimensionality reduction.

The cross-channel condition (det3) behaved similarly to det5. The best EER was for the combination of length normalization and the Bayesian scoring without dimensionality reduction. EER improved by 11% with respect to the system with only length normalization. The improvement in terms of DCF was smaller, by around 5%. Dimensionality reduction did not help.

Figure 9.3 shows DET curves for conditions det3 and det5 for male speakers. Both curves evidence that length normalization and the fully Bayesian likelihood ratio improve performance along most of the operating points of the curve. Improvement was larger in the low miss region of the curves. The fully Bayesian system with length normalization (red dashed curve) obtained a curve equal or better than the rest of systems for most operating points. The bold dots indicate the minimum DCF operating point. In that point, the difference between approaches was small, especially for condition det3.

In general, we concluded that the Bayesian approach helped more than length normalization in the conditions with a strong mismatch between development and trial datasets. In essence, it improved conditions det1, det2 and det4, which are microphone data while we trained our models only on telephone conversations. Nevertheless, for conditions with less mismatch (det3 and det5), results improved by evaluating the Bayesian likelihood ratio on length normalized i-vectors.

9.7.4 Results 8conv-core extended condition

Table 9.4 shows results 8conv-core extended condition. The best performances were achieved by the systems that included length normalization. Given the small differences in EER and DCF–an average 5%–, it was not clear whether the Bayesian approach helped length

Table 9.4: EER(%)/MinDCF Bayesian two-covariance model on NIST SRE10 8conv-core extended

	male		female	
	EER	DCF	EER	DCF
2cov	3.21	0.37	3.61	0.42
Bay2cov	1.75	0.24	2.07	0.33
LNorm+2cov	**0.70**	0.20	1.69	0.34
LNorm+Bay2cov	0.75	**0.18**	1.69	0.32
LNorm+LDA90+2cov	0.90	0.20	1.40	**0.23**
LNorm+LDA90+Bay2cov	0.87	0.19	**1.32**	0.24

normalization. Besides, dimensionality reduction improved females but worsened males. We concluded that when having several enrollment utterances the Bayesian approach contributes little.

9.7.5 Results 10sec conditions

Table 9.5 summarizes results for conditions with 10 second segments in test. For enrollment, we could have one conversation of 5 minutes (core), eight conversations (8conv) or one segment of 10 seconds (10sec). For males, the best system was the one with length normalization and Bayesian scoring. For females, the best was the one with only Bayesian scoring but it was not much better than the one that also included length normalization. In average, the system with length normalization plus Bayesian ratio improved by 9% in terms of EER and by 6% in terms of DCF with respect to the system with only length normalization. Again, the dimensionality reduction did not help.

9.8 Summary

In this chapter, we intended to calculate the speaker verification likelihood ratio in a fully Bayesian way by integrating-out the parameters of the PLDA model. In contrast, the classical likelihood ratio, which we called plug-in ratio, is computed by plugging a point estimate of the model $\hat{\mathcal{M}}$ into the involved likelihoods. The advantage of the Bayesian approach over the classical one consists in that the former takes into account the uncertainty about the value of the model parameters. The smaller the dataset used to train the model compared with the number of parameters the larger the uncertainty–e.g., when the number development speakers is not much larger than the i-vector dimension. Nonetheless, when we increase the amount of training data, uncertainty reduces and the Bayesian and the plug-in likelihood ratios should converge.

We presented a method to approximately estimate the Bayesian ratio for the particular case of the two-covariance model, also known as full-rank PLDA. The method takes advantage of the fact that the fully Bayesian likelihood ratio can be written as the plug-in ratio –i.e., the classical likelihood ratio where we plug-in a point estimate of the model $\hat{\mathcal{M}}$– multiplied by a correction factor. The correction factor is a ratio between model posteriors

Table 9.5: EER(%)/MinDCF Bayesian two-covariance model on NIST SRE10 conditions with 10 seconds tests.

Enroll.	System	male		female	
		EER	DCF	EER	DCF
core	2cov	10.19	0.51	10.99	0.54
	Bay2cov	**6.27**	0.36	**7.76**	**0.41**
	LNorm+2cov	7.10	0.36	8.84	0.45
	LNorm+Bay2cov	6.59	**0.34**	7.97	0.42
	LNorm+LDA90+2cov	8.88	0.43	9.22	0.51
	LNorm+LDA90+Bay2cov	7.95	0.39	8.76	0.45
8conv	2cov	9.23	0.39	8.88	0.41
	Bay2cov	5.84	0.26	**6.10**	**0.31**
	LNorm+2cov	4.02	0.25	7.57	0.39
	LNorm+Bay2cov	**3.90**	**0.23**	6.47	0.35
	LNorm+LDA90+2cov	5.71	0.32	7.86	0.43
	LNorm+LDA90+Bay2cov	5.29	0.30	7.36	0.40
10sec	2cov	17.96	0.71	16.31	0.80
	Bay2cov	14.37	0.63	**14.34**	**0.71**
	LNorm+2cov	14.78	**0.62**	16.80	0.78
	LNorm+Bay2cov	**13.43**	**0.62**	15.43	0.72
	LNorm+LDA90+2cov	16.30	0.73	18.17	0.79
	LNorm+LDA90+Bay2cov	15.03	0.69	15.89	0.75

given the trial data where, to compute the posterior in the numerator we assume that the trial is non-target; and, to compute the posterior in the denominator we assume that it is target. If these two posteriors are considerably different when we evaluate them at $\hat{\mathcal{M}}$ the correction factor can make a difference. As the model posteriors cannot be obtained in close form, we adopted a variational Bayes procedure to compute approximate posteriors.

We consider two approximations for computing the Bayesian ratio. In the first one, we assumed non-informative priors for the model parameters. Then, we computed the model posteriors given the trial plus the development data. The fact of using all the development data to compute the likelihood ratio of each trial implies a high computational cost. To alleviate this, we introduced a second approximation based on informative priors. In this second case, we started computing the model posterior given only the development data and the non-informative prior. Then, we used this posterior as informative prior when computing the model posteriors needed to evaluate each trial. Thus, the model posteriors in the correction factor are computed given only the trial data and the informative prior. In theory, the first approximation is more accurate because it takes into account that, when the model is unknown, the trial and development data are not independent. That means that the trial data affect the way in which the PLDA decompose the development i-vectors into a speaker term and a channel term. On the other hand, the second approximation is equivalent to assuming that the trial i-vectors do not affect the decomposition of the development ones. We thought that it is a reasonable approximation since the number of trial i-vector is very small, typically two, compared the number of development i-vectors. We obtained results that supported our hypothesis.

We presented results on several conditions of NIST SRE10. We compared the Bayesian approach with length normalization, another recent technique that also boosts performance in i-vectors systems. While length normalization intends to reduce the mismatch between development and trial datasets, the Bayesian solution intends to account for model uncertainty so we expected that both approaches could be complementary. Furthermore, if model posteriors are wide enough the Bayesian method also can help with database shift.

We observed that, for conditions with little mismatch between the development and the trial data (telephone data with segments of 5min. in both), length normalization usually outperformed the Bayesian scoring. However in most of the conditions, we obtained the best results by combining both techniques. For example in the core extended telephone-telephone condition, the combination of both improved EER by 55% with respect to the baseline, and by 9% with respect to only length normalization.

On the other hand, for conditions with a large mismatch between development (telephone segments of 5min.) and trial data (far-field microphone segments or telephone segments of 10secs.) the Bayesian approach outperformed length normalization. For example in the interview-interview with different microphones condition the Bayesian approach was 20% better than length normalization in terms of EER and 8.5% better in terms of DCF. And in the 10sec–10sec condition, EER and DCF of the Bayesian system was 9% and 5% better respectively.

Another interesting point was that both length normalization and Bayesian scoring did not need score normalization. However, they did not provide naturally well-calibrated likelihood ratios so score calibration was still needed to take Bayes optimal decisions.

We also evaluated what happened if we reduce the i-vector dimensionality by LDA. We observed that LDA was harmful except for the far-field microphone conditions. LDA reduction helps to alleviate model over-fitting by reducing the number of parameters to train. However, the Bayesian and length normalization methods seem to add robustness against over-fitting and do not need LDA.

Chapter 10

Bayesian Adaptation of PLDA to Domains with Scarce Development Data

10.1 Introduction

Database mismatch is a major handicap to introducing speaker recognition in certain applications. As we have seen, high performance methods like JFA and i-vectors are data driven approaches. That implies that models trained on a specific type of data may not generalize to other domains. For example, models trained on telephone English conversations will not achieve optimal performance on far-field microphone trials or on French conversations. In order to properly develop a speaker verification system, we need databases including large number of speakers and sessions per speaker. Given that NIST evaluations have driven speaker verification research in the last years and that NIST datasets are big enough, the efforts to deal with this issue had been limited. Now that speaker recognition is considered a mature technology there is a great interest to introduce it into more and more commercial applications. For most of those applications, the amount of available data is very scarce so researchers are starting to see the need for finding techniques that allow models trained on one domain, e.g., NIST, to be used on other domains.

In the previous chapter, we saw that techniques like i-vector length normalization and fully Bayesian evaluation of the likelihood ratio improve performance in conditions of database mismatch. Length normalization prevents dataset shift by making the development and trial i-vectors distributions closer and more Gaussian shaped [Garcia-Romero and Espy-Wilson, 2011]. The Bayesian approach primarily intends to combat model over-fitting by taking into account the uncertainty about the values of the model parameters. However, it also helps against database shift because, unless that we train models on a huge amount of data, the predictive distributions that result are heavy-tailed. We exposed that the Bayesian approach provided the largest benefits in conditions with more mismatch between development and test–development with 5 minutes telephone conversations and test with far-field microphones or 10 second segments.

The Bayesian approach has a high computational cost, which makes it difficult to apply in many circumstances. For this reason, in this chapter, we address the problem of database mismatch in a different manner. We assume that we have a model trained on a large development database from a domain different from the domain of interest. We also assume

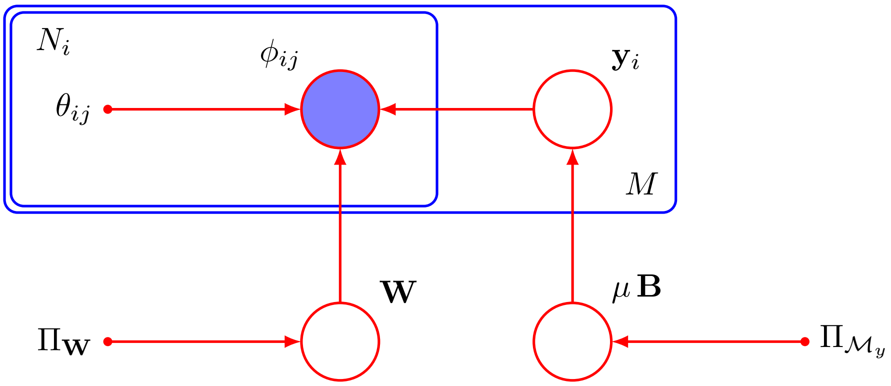


Figure 10.1: BN for Bayesian adaptation of the two-covariance model.

that we own a small amount of labeled data from the target domain. We will use the terms *out-of-domain* and *in-domain* to refer to the first and second dataset respectively. Then, we intend to do Bayesian adaptation of the out-of-domain PLDA model to the in-domain dataset. In chapter 7, we already employed this technique to adapt our Bayesian networks.

This chapter is organized as follows. Section 10.2 explains how to perform Bayesian adaptation of the two-covariance model. Section 10.3 shows the variational Bayes solution to approximate the model posteriors. Section 10.4 describes our experimental setup. We adapted a model trained on NIST SRE to the EVALITA09 dataset [Aversano, 2009]. Section 10.5 presents the results. We analyzed issues like the weight that the prior should have on the posterior; which model parameter is more important to adapt; importance of score normalization. We also compared length normalization against Bayesian adaptation. Bayesian adaptation performed better in conditions with shorter segments. Finally, Section 10.6 summarizes the chapter.

10.2 Bayesian Adaptation of the Two-covariance Model

In the previous chapter, we already introduced the fully Bayesian version of the two-covariance model. This version differed from the standard one in that model parameters are hidden variables with their corresponding probability distributions instead of fixed values. The Bayesian network in Figure 10.1 depicts the Bayesian two-covariance model. Note that there are some changes in the network with respect to Figure 9.1, in the preceding chapter. First, for Bayesian evaluation of the likelihood ratio, we only subjected the speaker space parameters (μ and $\mathbf{B}$) to Bayesian treatment, while now we also consider the within-class precision $\mathbf{W}$ as a hidden variable. Second, Figure 9.1 divides data into two plates: one for the development data and another for the trial data. In Figure 10.1, we eliminated the plate corresponding to the trial data because we did not intend to compute fully Bayesian likelihood ratios. We used the Bayesian model only to compute posteriors for the model parameters and, from them, to make *maximum a posteriori* point estimates.

We introduce some notation. We denote the database of out-of-domain i-vectors by $\mathbf{\Phi}_0$ and the in-domain i-vectors by $\mathbf{\Phi}$. The labels θ_0 partition the N_0 out-of-domain i-vectors into M_0 speakers while the labels θ partition the N in-domain i-vectors into M speakers.

The variables $\mathbf{Y}_0$ and $\mathbf{Y}$ are the speaker identity variables of the out-of-domain and in-domain datasets respectively. For convenience, we also define the sets of model parameters $\mathcal{M} = (\mu, \mathbf{B}, \mathbf{W})$ and $\mathcal{M}_{\mathbf{y}} = (\mu, \mathbf{B})$. Finally, we define the model priors:

$$P(\mathcal{M}|\Pi) = P(\mathcal{M}_{\mathbf{y}}|\Pi_{\mathcal{M}_{\mathbf{y}}})P(\mathbf{W}|\Pi_{\mathbf{W}}) \, . \tag{10.1}$$

In each step of adaptation process, we need different types of priors.

As explained in Chapter 7, (MAP) adaptation consists of three stages. First, we compute the posterior distribution of the parameters of the model given the out-of-domain dataset $P(\mathcal{M}|\mathbf{\Phi}_0, \theta_0, \Pi)$. In this stage, we assumed no prior knowledge about the parameters of the model. We do that by choosing a non-informative prior Π. Second, we compute the model posterior given the in-domain data $P(\mathcal{M}|\mathbf{\Phi}, \theta, \Pi_0)$ where we select an informative prior Π_0 that is just the posterior that we compute in the first stage:

$$P(\mathcal{M}|\Pi_0) = P(\mathcal{M}|\mathbf{\Phi}_0, \theta_0, \Pi) \, . \tag{10.2}$$

That is, the model posterior given the out-of-domain data is used as prior to compute the model posterior given the in-domain data. Finally, we make point estimates for μ, $\mathbf{B}$ and $\mathbf{W}$ by selecting the values that maximize the model posterior:

$$\mathcal{M}_{\text{MAP}} = \arg\max_{\mathcal{M}} P(\mathcal{M}|\mathbf{\Phi}_0, \theta_0, \Pi) \, . \tag{10.3}$$

The idea behind MAP adaptation is that, even when out-of-domain and in-domain datasets are so different that a model trained on the former does not perform well on the latter, both domains are close enough so that a small amount of in-domain data can bring the out-of-domain model close to the ideal in-domain model. In practice, in most MAP algorithms, the equations that we obtain consist in a weighted average of the prior and the maximum likelihood estimate of the parameters given the in-domain data. As we showed in the previous chapter, the model posteriors of the two-covariance model cannot be computed in an exact manner so we used variational inference.

10.3 Variational Inference for MAP Adaptation of the Two-Covariance Model

The variational Bayes procedure to obtain the model posteriors that we need to perform MAP adaptation is almost identical to the one described in the previous chapter. The only difference is that, in this chapter, we also consider $\mathbf{W}$ as a hidden variable. In this section we briefly summarize the equations needed. Full Derivation of these equation can be found in Appendix E.

10.3.1 Model posterior given the out-of-domain data

When computing the model posterior given the out-of-domain data, we assume a non-informative prior for the parameters μ, $\mathbf{B}$ and $\mathbf{W}$. Thus, the model posterior is only based on the training data. The prior is defined by these distributions:

$$P\left(\mathcal{M}|\Pi\right) = P\left(\mu,\mathbf{B}|\Pi_{\mathcal{M}_{\mathbf{y}}}\right) P\left(\mathbf{W}|\Pi_{\mathbf{W}}\right) \tag{10.4}$$

$$P\left(\mu,\mathbf{B}|\Pi_{\mathcal{M}_{\mathbf{y}}}\right) = \lim_{k\to 0} \mathcal{N}\left(\mu|\mu_0,(k\mathbf{B})^{-1}\right) \mathcal{W}\left(\mathbf{B}|\mathbf{B}_0/k,k\right) = \alpha \left|\frac{\mathbf{B}}{2\pi}\right|^{1/2} |\mathbf{B}|^{-(d+1)/2} \tag{10.5}$$

$$P\left(\mathbf{W}|\Pi_{\mathbf{W}}\right) = \lim_{k\to 0} \mathcal{W}\left(\mathbf{W}|\mathbf{W}_0/k,k\right) = \alpha \left|\mathbf{W}\right|^{-(d+1)/2} \tag{10.6}$$

where $\mathcal{W}$ denotes a Wishart distribution and d is the i-vector dimension. Since this density does not integrate to 1, it is improper and the symbol α is used to denote a normalizing constant which approaches zero.

Our VB solution approximates the joint posterior for all the latent variables by a factorized distribution of the form:

$$P\left(\mathcal{M},\mathbf{Y}_0|\mathbf{\Phi}_0,\theta_0,\Pi\right) \approx q\left(\mathcal{M},\mathbf{Y}\right) = q\left(\mathcal{M}\right) q\left(\mathbf{Y}_0\right) \tag{10.7}$$

which ignores any posterior dependencies between the speaker variables $\mathbf{Y}$ and the model $\mathcal{M}$.

We obtained that the optimum for the factor $q\left(\mathbf{Y}\right)$ is a product of Gaussians:

$$q^*\left(\mathbf{Y}_0\right) = \prod_{i=1}^{M_0} q^*\left(\mathbf{y}_i\right) = \prod_{i=1}^{M_0} \mathcal{N}\left(\mathbf{y}_i|\mathbf{L}_i^{-1}\gamma_i,\mathbf{L}_i^{-1}\right) \tag{10.8}$$

$$\mathbf{L}_i = \mathrm{E}_{\mathcal{M}}\left[\mathbf{B}\right] + N_i \mathrm{E}_{\mathcal{M}}\left[\mathbf{W}\right] \tag{10.9}$$

$$\gamma_i = \mathrm{E}_{\mathcal{M}}\left[\mathbf{B}\mu\right] + \mathrm{E}_{\mathcal{M}}\left[\mathbf{W}\right]\mathbf{F}_i \tag{10.10}$$

where N_i is the number of samples and $\mathbf{F}_i = \sum_{j=1}^{N_i} \phi_{ij}$ are the first order sufficient statistics of speaker i.

The optimum for the factor $q\left(\mathcal{M}\right)$ is also a product of factors:

$$q^*\left(\mathcal{M}\right) = q^*\left(\mathcal{M}_{\mathbf{y}}\right) q^*\left(\mathbf{W}\right)\ . \tag{10.11}$$

The factor $q^*\left(\mathcal{M}_{\mathbf{y}}\right)$ is a Gaussian-Wishart distribution:

$$q^*\left(\mathcal{M}_{\mathbf{y}}\right) = \mathcal{N}\left(\mu|\overline{\mu}_0,\left(\beta_{\mathbf{y}_0}\mathbf{B}\right)^{-1}\right) \mathcal{W}\left(\mathbf{B}|\mathbf{\Psi}_{\mathbf{y}_0},\nu_{\mathbf{y}_0}\right) \tag{10.12}$$

where we defined:

$$\beta_{\mathbf{y}_0} = \nu_{\mathbf{y}_0} = M_0 \tag{10.13}$$

$$\overline{\mu}_0 = \frac{1}{M_0}\sum_{i=1}^{M_0} \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right] \tag{10.14}$$

$$\mathbf{\Psi}_{\mathbf{y}_0}^{-1} = \sum_{i=1}^{M_0} \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right] - M_0\overline{\mu}_0\overline{\mu}_0^T\ . \tag{10.15}$$

The factor $q^*\left(\mathbf{W}\right)$ is Wishart distributed:

$$q^*\left(\mathbf{W}\right) = \mathcal{W}\left(\mathbf{W}|\mathbf{\Psi}_{\mathbf{W}_0},\nu_{\mathbf{W}_0}\right) \tag{10.16}$$

where we defined

$$\nu_{\mathbf{W}_0} = N_0 = \sum_{i=1}^{M_0} N_i \tag{10.17}$$

$$\mathbf{S} = \sum_{i=1}^{M_0} \sum_{j=1}^{N_i} \phi_{ij}\phi_{ij}^T \tag{10.18}$$

$$\mathbf{\Psi}_{\mathbf{W}_0}^{-1} = \mathbf{S} + \sum_{i=1}^{M_0} \left(N_i \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right] - \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]\mathbf{F}_i^T - \mathbf{F}_i\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]^T \right) . \tag{10.19}$$

Note that, even when we chose improper priors, the distributions $q(\mu, \mathbf{B})$ and $q(\mathbf{W})$ are proper as long as the number of speakers M be larger than the i-vector dimension. Also note that we only forced the factorization between the variables $\mathbf{Y}$ and $\mathcal{M}$; the additional factorizations that appear were not forced in any way. These additional factorizations are a consequence of the interaction between the assumed factorization and the conditional independence properties of the true joint distribution described by the Bayesian network in Figure 10.1.

We need to iterate by cycling between the calculus of $q(\mathbf{Y})$ and $q(\mathcal{M})$ until convergence. We can evaluate the convergence by tracking the VB lower bound detailed in Appendix E. Besides To compute the parameters of the variational factors, we need to evaluate some expectations:

$$\mathrm{E}_{\mathcal{M}}[\mathbf{B}] = \nu_{\mathbf{y}_0}\mathbf{\Psi}_{\mathbf{y}_0} \qquad \mathrm{E}_{\mathcal{M}}[\mathbf{B}\mu] = \nu_{\mathbf{y}_0}\mathbf{\Psi}_{\mathbf{y}_0}\overline{\mu}_0 \tag{10.20}$$

$$\mathrm{E}_{\mathcal{M}}[\mathbf{W}] = \nu_{\mathbf{W}_0}\mathbf{\Psi}_{\mathbf{W}_0} \tag{10.21}$$

$$\mathrm{E}_{\mathbf{Y}}[\mathbf{y}_i] = \mathbf{L}_i^{-1}\gamma_i \qquad \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right] = \mathbf{L}_i^{-1} + \mathrm{E}_{\mathbf{Y}}[\mathbf{y}_i]\,\mathrm{E}_{\mathbf{Y}}[\mathbf{y}_i]^T . \tag{10.22}$$

10.3.2 Model posterior given the in-domain data

Now, we adapt the model given the out-of-domain data to the target domain. For that, we use the variational factor $q(\mathcal{M})$ computed in the previous step as prior

$$P(\mathcal{M}|\Pi_0) = P(\mathcal{M}|\mathbf{\Phi}_0, \theta_0, \Pi) \approx q(\mathcal{M}|\mathbf{\Phi}_0, \theta_0, \Pi) \tag{10.23}$$

and compute the model posterior given the in-domain data.

We again factorized the posterior of all the latent variables using (10.7). We obtained that the optimum for the factor $q(\mathbf{Y})$ is the same as for the case with non-informative priors. The optimum for $q(\mathcal{M})$ also factorized into two Gaussian-Wishart factors: one for the speaker space and another for the channel space. These are given by

$$q^*(\mathcal{M}) = q^*(\mathcal{M}_{\mathbf{y}})\, q^*(\mathbf{W}) \tag{10.24}$$

$$q^*(\mathcal{M}_{\mathbf{y}}) = \mathcal{N}\left(\mu|\overline{\mu}, (\beta_{\mathbf{y}}\mathbf{B})^{-1}\right)\mathcal{W}(\mathbf{B}|\mathbf{\Psi}_{\mathbf{y}}, \nu_{\mathbf{y}}) \tag{10.25}$$

$$q^*(\mathbf{W}) = \mathcal{W}(\mathbf{W}|\mathbf{\Psi}_{\mathbf{W}}, \nu_{\mathbf{W}}) . \tag{10.26}$$

where

$$\begin{aligned}
\overline{\mathbf{y}} =& \frac{1}{M}\sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right] && (10.27)\\
\mathbf{S}_{\mathbf{y}} =& \sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right] - M\overline{\mathbf{y}}\overline{\mathbf{y}}^T && (10.28)\\
\mathbf{S}_{\mathbf{W}} =& \sum_{i=1}^{M}\sum_{j=1}^{N_i}\phi_{ij}\phi_{ij}^T + \sum_{i=1}^{M}\left(N_i\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right] - \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]\mathbf{F}_i^T - \mathbf{F}_i\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]^T\right) && (10.29)\\
\beta_{\mathbf{y}} =& \beta_{\mathbf{y}_0} + M && (10.30)\\
\nu_{\mathbf{y}} =& \nu_{\mathbf{y}_0} + M && (10.31)\\
\nu_{\mathbf{W}} =& \nu_{\mathbf{W}_0} + N && (10.32)\\
\overline{\mu} =& \frac{1}{\beta_{\mathbf{y}}}\left(\beta_{\mathbf{y}_0}\overline{\mu}_0 + M\overline{\mathbf{y}}\right) && (10.33)\\
\mathbf{\Psi}_{\mathbf{y}}^{-1} =& \mathbf{\Psi}_{\mathbf{y}_0}^{-1} + \mathbf{S}_{\mathbf{y}} + \frac{\beta_{\mathbf{y}_0}M}{\beta_{\mathbf{y}}}\left(\overline{\mathbf{y}} - \overline{\mu}_0\right)\left(\overline{\mathbf{y}} - \overline{\mu}_0\right)^T && (10.34)\\
\mathbf{\Psi}_{\mathbf{W}}^{-1} =& \mathbf{\Psi}_{\mathbf{W}_0}^{-1} + \mathbf{S}_{\mathbf{W}}\ . && (10.35)
\end{aligned}$$

Finally, we make MAP point estimates of the model parameters:

$$\begin{aligned}
\mu_{\mathrm{MAP}} =& \overline{\mu} && (10.36)\\
\mathbf{B}_{\mathrm{MAP}} =& (\nu_{\mathbf{y}} - d - 1)\mathbf{\Psi}_{\mathbf{y}} && (10.37)\\
\mathbf{W}_{\mathrm{MAP}} =& (\nu_{\mathbf{W}} - d - 1)\mathbf{\Psi}_{\mathbf{W}}\ . && (10.38)
\end{aligned}$$

We used these MAP estimates to evaluate the plug-in likelihood ratios of the trials in the usual way.

10.3.3 Controlling the weight of the prior on the posterior

Equations (10.27) to (10.35) evidence that the parameters $\beta_{\mathbf{y}_0} = \nu_{\mathbf{y}_0} = M_0$ and $\nu_{\mathbf{W}_0} = N_0$ control the weight that the prior has on the posterior. The larger M_0 and N_0 are the more in-domain data we need to take the model far from the prior. The values N_0 and M_0 also correspond to the degrees of freedom of the Wishart priors as well as modulating the variance of the μ Gaussian prior. The degrees of freedom of the Wishart are related to the width of the distribution, the lower they are the flatter and less informative the distribution is. The extreme case happens when we take the limit of the Wishart as the degrees of freedom approach 0, which results in the non-informative prior in (10.6). In practice, the degrees of freedom represent the number of samples that we have seen to train the model. If we observe many samples, there will be few uncertainty about the model parameters, which we will see reflected in a sharp distribution. In conclusion, the higher the values of M_0 and N_0 the sharper the prior. Also, a sharp prior means that the out-of-domain data weighs more on the in-domain posterior.

If we train our prior on a very large out-of-domain dataset and we have a small amount of adaptation data the MAP model will move very little from the prior model. Then, we will not obtain any benefits from the adaptation procedure. In those cases, we may want to reduce the influence of the prior on the posterior. We can do it just by lowering the values of $\beta_{\mathbf{y}_0}$, $\nu_{\mathbf{y}_0}$ and $\nu_{\mathbf{W}_0}$. However, we must also modify $\mathbf{\Psi}_{\mathbf{y}_0}$ and $\mathbf{\Psi}_{\mathbf{W}_0}$ so that the expected values of $\mathbf{B}$ and $\mathbf{W}$ do not change with respect to the original prior. Thus, we defined $\eta_{\mathbf{y}}$ a $\eta_{\mathbf{W}}$ and updated the prior parameters as:

$$\beta_{\mathbf{y}_0} = \nu_{\mathbf{y}_0} \leftarrow \eta_{\mathbf{y}} M_0 \qquad \mathbf{\Psi}_{\mathbf{y}_0} \leftarrow \eta_{\mathbf{y}}^{-1} \mathbf{\Psi}_{\mathbf{y}_0} \tag{10.39}$$

$$\nu_{\mathbf{W}_0} \leftarrow \eta_{\mathbf{W}} N_0 \qquad \mathbf{\Psi}_{\mathbf{W}_0} \leftarrow \eta_{\mathbf{W}}^{-1} \mathbf{\Psi}_{\mathbf{W}_0} \; . \tag{10.40}$$

When selecting $\eta_{\mathbf{y}}$ and $\eta_{\mathbf{W}}$, we must take into account that the resulting $\nu_{\mathbf{y}_0}$ and $\nu_{\mathbf{W}_0}$ need to be both larger than d so that the Wishart distributions of the prior be well defined:

$$\eta_{\mathbf{y}} \geq d/M_0 \qquad \eta_{\mathbf{W}} \geq d/N_0 \; . \tag{10.41}$$

We refer to the modified values of $\nu_{\mathbf{y}_0}$ and $\nu_{\mathbf{W}_0}$ as the *effective number of speakers and segments* used to train the prior. That is, we have a prior that provides the same expectations for $\mathcal{M}$ as the original prior but that behaves as trained with a smaller number of samples.

10.4 Experimental Setup

10.4.1 EVALITA09 Dataset

We experimented on the EVALITA09 dataset [Aversano, 2009]. EVALITA was an evaluation of natural language processing and speech tools for Italian. Telephone conversations were recorded over landline (PSTN) or mobile (GSM) channels. All recordings were in Italian language. The EVALITA09 SV task had guidelines similar to those of NIST evaluations. However, we found an important mismatch between EVALITA and NIST due to the language of the speakers and to the telephone channels, which had different spectral patterns to those that we find in NIST. For these reasons, we found it convenient for evaluating the adaptation task.

The database was split into three sets:

- Development set: speech data recorded by 30 male and 30 female speakers, during 20 sessions (10 PSTN calls + 10 GSM calls). The total duration of speech was 1200 minutes (∼1 minute per call). Calls were provided cut into small segments, thus we had 18000 short speech segments (9000 male + 9000 female). We can use this set to train UBM, JFA and PLDA models as well as for the score normalization cohorts.

- Enrollment set: data for speaker enrollment. It had 50 male and 50 female speakers. These speakers were different from those in the development set. Six training conditions were defined:

  - TC1: PSTN short (1 PSTN call, ∼1 minute per client).
  - TC2: GSM short (1 GSM call, ∼1 minute per client).
  - TC3: PSTN long (3 PSTN calls, ∼3 minutes per client).
  - TC4: GSM long (3 GSM calls, ∼3 minutes per client).
  - TC5: mixed short (1 PSTN + 1 GSM calls, ∼2 minutes per client).

  - TC6: mixed long (3 PSTN + 3 GSM calls, ∼6 minutes per client).

- Test set: Two test conditions were considered having 2071 trials each:
  - TS1: short (1 sequence of digits; ∼10 seconds).
  - TS2: long (1 sequence of digits, 4 short sentences, 2 isolated words; ∼30 seconds).

10.4.2 SV system configuration

The features for our PLDA were 400 dimension i-vectors. We extracted i-vectors from 20 MFCC with added deltas and double deltas with short-time Gaussianization. The UBM and i-vector extractor were gender dependent and based on 2040 diagonal Gaussians.

We used NIST SRE04–06 as out-of-domain development data. It included 529 male and 729 female speakers with a total of 7410 male and 9920 female conversations. We used NIST to train the UBM, i-vector extractor and our baseline two-covariance model as well as to obtain the prior model for the MAP adaptation.

PLDA models were also gender dependent. We had two PLDA baselines: A two-covariance model trained on NIST and a SPLDA model trained on the EVALITA development set. To train the two covariance model, we need the speaker number to be larger than the i-vector dimension (> 400). Since in EVALITA there were only 30 development speakers we could not train a two-covariance model on them. For that reason, the EVALITA baseline was based on the SPLDA model, which only requires the number of speakers to be larger than the rank of the eigen-voice matrix $\mathbf{V}$. We chose rank 25 for $\mathbf{V}$. We compared both baselines with several two-covariance models adapted from NIST to EVALITA.

In the experiments where we used length normalization, the centering and whitening parameters (mean and rotation matrices) were estimated in a manner matched with the classifier. That is, trained from NIST, from EVALITA or adapted from NIST to EVALITA.

Unless stated otherwise, we show results with S-Norm [Senoussaoui et al., 2010]. The cohorts were utterances from the EVALITA development set (9000 male + 9000 female segments).

10.5 Experiment Results

10.5.1 Analysis of the weight of the prior on the model posterior

Table 10.1 compares our two baselines with the systems with MAP adaptation in the condition TC6–TS2 (6 minutes enrollment against 30 second test). Results are in terms of EER and minimum DCF in the operating point defined in the EVALITA guidelines [Aversano, 2009] ($C_{\text{Miss}} = 10$, $C_{\text{FA}} = 1$, $P_{\mathcal{T}} = 0.5$). In this experiment, we did not length normalize i-vectors. We compared different adaptations where we tuned the influence of the prior on the posterior as we explained in Section 10.3.3. In essence, we adjusted the degrees of freedom in the prior Gaussian-Wishart distributions to obtain distributions with the same expectations than the original prior but that look like trained with less or more samples than the original. When we increase the effective number of samples, the prior is sharper and its weight on the posterior is larger and vice versa.

The table evidences that training only with EVALITA09 largely degraded performance with respect to NIST. The low number of in-domain speakers, which forced us to use a

Table 10.1: EER(%)/MinDCF EVALITA09 TC6 TS2 vs. effective number of speakers ($\nu_{\mathbf{y}_0}$) and segments (*nuwo*) in the prior distribution.

	male		female	
	EER	DCF	EER	DCF
NIST	2.99	0.098	1.37	0.089
EVITA09	6.08	0.279	7.02	0.232
Adapt $\eta_{\mathbf{y}} = \eta_{\mathbf{W}} = 1$	1.83	0.104	1.32	0.059
Adapt $\beta_{\mathbf{y}_0} = \nu_{\mathbf{y}_0} = 401$				
$\nu_{\mathbf{W}_0} = 401$	2.12	0.160	1.56	0.107
$\nu_{\mathbf{W}_0} = 500$	2.12	0.158	1.55	0.107
$\nu_{\mathbf{W}_0} = 750$	2.03	0.151	1.45	0.102
$\nu_{\mathbf{W}_0} = 1500$	1.77	0.141	1.32	0.090
$\nu_{\mathbf{W}_0} = 3000$	1.79	0.119	1.39	0.071
$\nu_{\mathbf{W}_0} = 6000$	1.83	0.106	1.35	0.061
$\nu_{\mathbf{W}_0} = 9000$	1.74	0.101	1.26	0.059
$\nu_{\mathbf{W}_0} = 12000$	**1.68**	0.089	1.26	0.058
$\nu_{\mathbf{W}_0} = 15000$	1.80	0.086	**1.17**	**0.048**
$\nu_{\mathbf{W}_0} = 18000$	1.79	**0.081**	**1.17**	**0.048**
Adapt $\beta_{\mathbf{y}_0} = \nu_{\mathbf{y}_0} = 500$				
$\nu_{\mathbf{W}_0} = 401$	2.17	0.160	1.56	0.112
$\nu_{\mathbf{W}_0} = 500$	2.16	0.160	1.56	0.112
$\nu_{\mathbf{W}_0} = 750$	2.12	0.154	1.50	0.107
$\nu_{\mathbf{W}_0} = 1500$	1.83	0.141	1.38	0.092
$\nu_{\mathbf{W}_0} = 3000$	1.79	0.121	1.39	0.071
$\nu_{\mathbf{W}_0} = 6000$	1.83	0.109	1.35	0.061
$\nu_{\mathbf{W}_0} = 9000$	1.80	0.104	1.31	0.059
$\nu_{\mathbf{W}_0} = 12000$	1.83	0.096	1.26	0.059
$\nu_{\mathbf{W}_0} = 15000$	1.80	0.089	1.18	0.051
$\nu_{\mathbf{W}_0} = 18000$	1.83	0.084	1.26	**0.048**

Table 10.2: EER(%)/MinDCF EVALITA09 TC6 TS2 vs. adapted parameters.

	male		female	
	EER	DCF	EER	DCF
NIST	2.99	0.098	1.37	0.089
EVITA09	6.08	0.279	7.02	0.232
Adapt μ	2.96	0.096	1.37	0.086
Adapt $\mu\mathbf{B}$	2.86	**0.073**	1.39	0.087
Adapt $\mu\mathbf{BW}$	**1.80**	0.086	**1.17**	**0.048**

very low-rank eigen-voice matrix, did not properly model the speaker space. However, by adapting the NIST model with these few speakers we obtained a nice improvement. In the case where we did not tune the degrees of freedom of the prior ($\eta_{\mathbf{y}} = \eta_{\mathbf{W}} = 1$), male EER improved by 39%, female EER by 3.6%, female DCF by 33% and only male DCF worsened by 6%.

We optimized performance by tuning $\beta_{\mathbf{y}_0}$, $\nu_{\mathbf{y}_0}$ and $\nu_{\mathbf{W}_0}$. The values of $\beta_{\mathbf{y}_0}$ and $\nu_{\mathbf{y}_0}$ started from 401 so that the prior be proper. The tuning of $\beta_{\mathbf{y}_0} = \nu_{\mathbf{y}_0}$ did not affect too much results because, in any case, all the values were much larger than the number of in-domain speakers ($\beta_{\mathbf{y}_0} = \nu_{\mathbf{y}_0} >> 30$). However, we improved performance by increasing $\nu_{\mathbf{W}_0}$. We could think that, having as much as 9000 development segments in EVALITA09, we should obtain good results by training $\mathbf{W}$ only on them. If that were the case, we should obtain the best error rates for low values of $\nu_{\mathbf{W}_0}$. On the contrary, we achieved the best results by selecting $\nu_{\mathbf{W}_0}$ $1.5-2$ times larger than the in-domain segments. This is explained because the EVALITA09 segments came from a reduced number of speakers. Even when PLDA assumes that channel offset is *a priori* independent of the speakers, in practice it is not. That means that those 9000 segments do not contribute as much information as, e.g., 9000 segments coming from hundreds of speakers. Thus, we prevented over-fitting by giving more weight to the prior.

For example with $\beta_{\mathbf{y}_0} = \nu_{\mathbf{y}_0} = 401$ and $\nu_{\mathbf{W}_0} = 15000$, we improved male EER by 40%, female EER by 15%, male DCF by 12% and female DCF by 46% with respect to NIST development. And with respect to the adapted model without tuning, they improved by 1.6, 11, 18 and 17% respectively. The experiments in next sections are always with $\beta_{\mathbf{y}_0} = \nu_{\mathbf{y}_0} = 401$ and $\nu_{\mathbf{W}_0} = 15000$.

10.5.2 Analysis different parameter adaptation

Besides of adapting all the model parameters, we studied the effect of adapting only μ or only μ and $\mathbf{B}$. Table 10.2 presents results for TC6–TS2. We did not length normalize the i-vectors. Results did not significantly improve when we did not adapt $\mathbf{W}$. Only the male DCF experienced a noticeable improvement of 25% when adapting μ and $\mathbf{B}$. The other metrics improved by less than 5%. We obtained the largest benefits when we also adapted the channel precision $\mathbf{W}$. EER improved by an average 26% with respect to adapting only μ and $\mathbf{B}$. Female DCF improved by 45% but male worsened by 17%, however it was still 12% better than the baseline. This behavior evidenced that inter-session variability was very different between NIST and EVALITA. Since we had very few in-domain speakers, we cannot decide if the lack of improvement obtained from adapting μ and $\mathbf{B}$ happened because

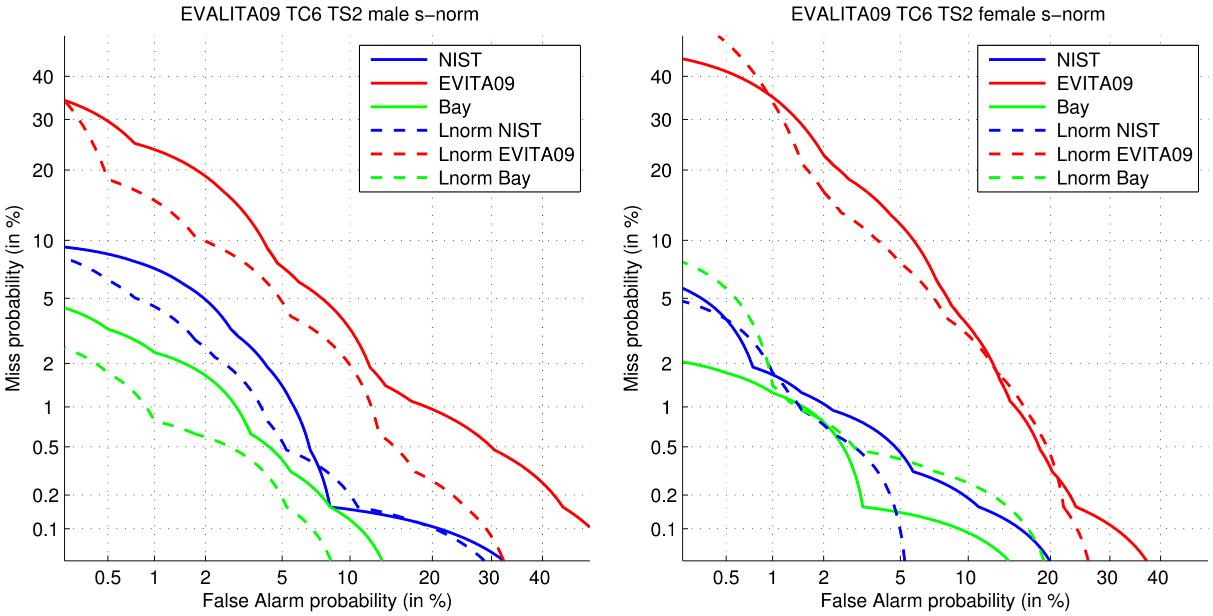


Figure 10.2: DET curves TC6–TS2 for male (left) and female (right).

we needed more speakers or because there was few difference between the speaker spaces of both datasets. We are inclined to the first hypothesis because we think the difference between English and Italian languages should be reflected in both spaces: speaker and channel.

10.5.3 Analysis S-Norm and length normalization

Experiments on NIST databases revealed that length normalization makes score normalization unnecessary [Garcia-Romero and Espy-Wilson, 2011, Senoussaoui et al., 2011a]. The results that we presented in Table 2.2 of Chapter 2 and Table 9.2 of Chapter 9 verified those findings. It has been claimed that score normalization is not necessary mainly because length normalization reduces mismatch between the development and test databases. In this experiment, we wanted to investigate whether this fact was also true for other datasets like EVALITA.

Table 10.3 presents results with length normalized i-vectors with and without

Table 10.3: EER(%)/MinDCF EVALITA09 TC6–TS2 with length normalization.

		male		female	
		EER	DCF	EER	DCF
No S-Norm	NIST	3.28	0.146	1.61	0.113
	EVITA09	5.60	0.245	6.43	0.247
	Adapt μ**BW**	**1.15**	**0.091**	**1.35**	**0.106**
S-Norm	NIST	2.23	0.100	1.25	**0.055**
	EVITA09	4.92	0.193	6.20	0.236
	Adapt μ**BW**	**0.93**	**0.068**	**1.18**	0.077

score normalization for condition TC6–TS2. Length normalization effectively improved performance in EVALITA for male trials. For females, it improved the baselines a little but worsened the adapted system. S-Norm was always beneficial.

Figure 10.2 compares DET curves with and without length normalization. All the curves include S-Norm. We observe that, for males, length normalization improved over most of the operating points. For females, the distance between curves with and without normalization is small. Besides, the male curves manifest the significant relative improvement between the baselines and the adapted system.

10.5.4 Results multiple conditions

Table 10.4 shows results on all the EVALITA09 conditions without and with length normalization. Length normalization did not achieve the best results in all conditions. For enrollment conditions TC1-3 and TC5 females, with length normalization, performance of the adapted model degraded with respect to the NIST model. For those conditions, the lowest error rates were for the adapted model without length normalization. In the rest of conditions, we obtained good results by combining adaptation and length normalization.

The overall conclusion that we drew from the table was that conditions with longer enrollment or test segments like TC6–TS2 benefited more from length normalization. Otherwise, using only model adaptation was better.

10.6 Summary

In this chapter, we explained how to adapt a two-covariance model from a domain with a large amount of development data to another one with scarce data. Adaptation was based on MAP estimation. First, we compute the posterior of the model parameters given the out-of-domain data and a non-informative prior. That posterior becomes an informative prior for the next step where we compute the posterior given the in-domain data. Finally, we make point estimates of the model parameters by maximizing this last posterior. Since the model posterior cannot be computed in closed form, we approximated it by applying variational Bayes.

We adapted models from NIST domain to EVALITA09. EVALITA09 presented an evaluation framework similar to NIST. However there was a significant mismatch between both dataset. EVALITA09 was recorded in Italian Language while NIST is dominated by English language. There were also differences in the telephone channels used in each one of them. As consequence, models trained on NIST did not performed optimally on EVALITA09. Development data in EVALITA09 was scarce and models trained only on it performed badly. With Bayesian adaptation, we improved EER by 15–40% and DCF by 12–46% with respect to the baseline, trained on NIST. The main gain came from the adaptation of the within-class precision $\mathbf{W}$. When we adapted only μ and $\mathbf{B}$ we obtained minor improvements.

We compared Bayesian adaptation with length normalization, which is supposed to reduce dataset shift. In EVALITA09, conditions with longer enrollment or test segments benefited from combining both techniques. In the rest of conditions, model adaptation performed better than length normalization. Contrary to what we had observed on NIST, length normalization needed score normalization to achieve optimal performance.

Table 10.4: EER(%)/MinDCF EVALITA09 Multiple conditions.

Cond.	System	Without Lnorm				With Lnorm			
		male		female		male		female	
		EER	DCF	EER	DCF	EER	DCF	EER	DCF
TC1–TS1	NIST	10.50	0.566	9.09	0.465	10.48	0.537	8.05	**0.417**
	EVITA09	17.05	0.667	12.75	0.594	14.47	0.607	12.38	0.635
	Adapt μ**BW**	**9.15**	**0.441**	**7.89**	0.458	12.16	0.680	11.72	0.622
TC1–TS2	NIST	6.02	0.352	4.28	**0.146**	5.27	0.287	3.86	0.159
	EVITA09	10.42	0.628	10.62	0.426	9.52	0.560	11.38	0.543
	Adapt μ**BW**	**4.88**	**0.326**	**3.87**	0.193	8.01	0.467	6.38	0.313
TC2–TS1	NIST	17.51	0.666	13.10	0.563	15.65	0.565	11.96	0.637
	EVITA09	18.59	0.740	18.37	0.785	17.28	0.674	15.47	0.785
	Adapt μ**BW**	**13.67**	**0.612**	**9.97**	**0.527**	14.94	0.700	14.22	0.663
TC2–TS2	NIST	11.79	0.417	5.64	0.318	10.68	0.492	5.92	0.295
	EVITA09	13.66	0.621	14.06	0.586	11.07	0.494	12.18	0.494
	Adapt μ**BW**	**9.92**	**0.383**	**5.02**	**0.254**	10.32	0.496	7.27	0.378
TC3–TS1	NIST	9.54	0.482	8.08	0.458	9.02	0.441	7.21	0.422
	EVITA09	13.94	0.637	12.51	0.562	11.98	0.554	11.87	0.548
	Adapt μ**BW**	**7.44**	**0.400**	**6.60**	**0.407**	9.67	0.536	8.90	0.494
TC3–TS2	NIST	4.38	0.243	3.60	0.137	3.96	0.247	3.07	0.136
	EVITA09	9.54	0.521	8.54	0.321	8.31	0.399	8.39	0.384
	Adapt μ**BW**	**3.69**	**0.231**	**2.90**	**0.159**	5.08	0.318	3.38	0.272
TC4–TS1	NIST	17.35	0.659	13.30	0.565	14.66	0.545	11.90	0.518
	EVITA09	16.30	0.749	16.21	0.719	15.10	0.678	13.75	0.637
	Adapt μ**BW**	12.45	**0.618**	**8.63**	**0.403**	**11.95**	0.622	10.37	0.478
TC4–TS2	NIST	10.84	0.443	4.44	0.272	9.98	0.401	4.15	0.253
	EVITA09	12.46	0.561	10.73	0.522	10.42	0.439	9.56	0.407
	Adapt μ**BW**	8.41	**0.392**	3.89	**0.246**	**7.94**	0.402	**3.77**	0.272
TC5–TS1	NIST	9.54	0.399	6.57	0.383	8.50	0.325	6.61	0.318
	EVITA09	14.03	0.545	12.84	0.588	10.94	0.468	10.73	0.579
	Adapt μ**BW**	6.41	**0.308**	**4.36**	**0.234**	**5.88**	0.328	8.04	0.362
TC5–TS2	NIST	3.81	0.189	1.58	0.089	3.30	0.156	1.22	0.058
	EVITA09	8.32	0.294	9.32	0.354	5.79	0.248	8.19	0.355
	Adapt μ**BW**	**2.60**	0.154	**1.53**	**0.076**	2.79	**0.146**	2.14	0.129
TC6–TS1	NIST	8.11	0.376	7.11	0.314	7.28	0.285	6.29	0.280
	EVITA09	12.59	0.499	12.03	0.512	9.63	0.409	10.58	0.477
	Adapt μ**BW**	5.12	0.297	**3.81**	**0.192**	**3.81**	**0.224**	5.85	0.273
TC6–TS2	NIST	2.99	0.098	1.37	0.089	2.23	0.100	1.25	0.055
	EVITA09	6.08	0.279	7.02	0.232	4.92	0.193	6.20	0.236
	Adapt μ**BW**	1.80	0.086	**1.17**	**0.048**	**0.93**	**0.068**	1.18	0.077

In the future, this work could be extended to do Bayesian adaptation of the UBM and the i-vector extractor.

Part IV

Attacks to Speaker Recognition Systems: Spoofing and Tampering

Chapter 11

Detecting Replay Spoofing Attacks on Speaker Verification Systems

11.1 Introduction

As we have seen in previous chapters, state-of-the-art speaker verification systems have achieved great performance in recent times. This has been possible thanks to the appearance of advanced modeling techniques like JFA [Kenny et al., 2008] or PLDA [Kenny, 2010], which compensate for channel mismatch and other inter-session variability effects. However, performance is usually measured in an ideal scenario where impostors do not try to disguise their voices to make then similar to the target speakers and where target speakers do not try to conceal their identities. For example, that is the case of NIST evaluations [NIST Speech Group, 2012]. In this chapter, we deal with spoofing attacks. Spoofing is the fact of impersonating somebody by employing techniques like playing a recording of the victim or voice transformation. Spoofing makes sense in SV applications related to access control: access to computers, bank accounts, restricted areas, etc.

All biometric modalities are subjected to the risk of being spoofed. Great efforts have been made to develop spoofing countermeasures for some of them. For example, there are several public databases [Anjos and Marcel, 2011, Chingovska et al., 2012] and competitions [Chakka et al., 2011] dedicated to spoofing in face recognition. On the contrary, research resources for spoofing in speaker verification have been scarce. However, this topic is currently drawing attention motivated by the desire of introducing this technology in new applications like telephone banking. Speaker recognition is specially suited for telephone applications where, since there is no human supervision, the exposure to attacks is elevated. Besides, risk increases because there are many to obtain a speech signal that can fool the recognition system. These techniques can be classified into four groups [Evans et al., 2013]: impersonation, speech synthesis, voice conversion and replay attacks. In this chapter, we focus on detecting replay attacks. We take into account attacks to text-independent systems as much as to text-dependent. The former just consists in playing a recording of the victim. Meanwhile, for the latter, the spoofer usually does not possess the exact utterance requested by the system, so he needs to create it by concatenating several excerpts from recordings of the target speaker. These low-technology spoofs are among the most dangerous because they are difficult to detect and they are easily available to impostors without any advanced technical knowledge.

This chapter is organized as follows. Section 11.2 reviews the state-of-the-art of different types of spoofing attacks (impersonation, speech conversion and synthesis, replay attacks). In Section 11.3, we describe the spoofing detection tasks: naive replay attacks and cut and paste attacks. We describe the assumptions that we need to make about how the attack is carried out. Section 11.4 explains our approach to detect signals created by concatenating words from several recordings. It was based on comparing MFCC and pitch contours of enrollment and test segments by aligning them with dynamic time warping. Section 11.5 is dedicated to detect recordings acquired by far-field microphones or replayed by a loudspeaker. These type of signals are suspects of being replay attacks. In Section 11.6, we present experiments on different databases with spoofing. The results revealed that spoofing perilously increases the number of false alarms of current SV systems. We performed spoofing detection experiments as well as fusing the spoofing detector with the SV score. We pruved that the fusion can reduce the false alarms without damaging the error rates of the normal trials. Finally, Section 11.7 summarizes de chapter.

11.2 Spoofing Types

11.2.1 Impersonation

Impersonation, imitation or mimicry refers to attacks where impostors alter their own voices to sound like the target speaker. Imitators tend to copy prosodic patterns like intonation, loudness and rhythm as well as lexical usage. However, there are physical differences between speakers that cannot be copied, which complicates producing an exact replica of the victim's voice. There are several studies about the consequences of imitation on speaker recognition; some of them disagree in their conclusions. The first ones date from the seventies. For example, in [Endres, 1971], imitators succeeded in varying their formants and pitch but they could not make them similar to those of the imitated persons. On the contrary, in [Lummis and Rosenberg, 1972], 27% of the utterances produced by the best mimics cheated the SV system while only 1.2% of the utterances of normal impostors were accepted. In the last decade, new studies have evaluated the vulnerability of modern SV approaches. In [Blomberg et al., 2004], a professional impersonator trains his voice to mimic two target speakers. Training was performed by listening and using the SV score as feedback. After training, test scores increased significantly. An acoustic analysis evidenced that the impersonator adjusted his formant positions for the target speaker. Results revealed a strong correlation between the second formant and the SV score. In [Lau et al., 2004, Lau et al., 2005], the authors concluded that non-professional impersonators can adapt their voices to overcome SV, but only when their natural voice is already similar to that of the target speaker. In such a case, professional linguistic impersonators are not necessarily better than non-professionals. For each impersonator, a spectral GMM system selected its closest, intermediate and furthest speaker from YOHO database. No imitator was accepted as the intermediate or further speaker. The authors hypothesized that speakers whose vowel space is similar to that of the imitator tend to be easily imitated, likely due to similar articulatory features. If the articulators are very different, it will be difficult or impossible to sufficiently modify the voice towards the target. Experiments in [Mariethoz and Bengio, 2006] show that professional imitators are better impostors than average people, yet their SV system perfectly separated client and impostor accesses. In [Farrus et al., 2008, Farrús et al., 2010] the authors tried to quantify how much a speaker is able to approximate others by employing a SV system based on a set of prosodic and voice source features. The parameters used in the experiment included word duration, word segments, means and ranges of fundamental frequency, as well as jitter and shimmer. Two professional impersonators imitating well-known politicians largely increased the identification error rates. The first study of the vulnerability of an i-vector based system [Gonzalez Hautamaki et al., 2013] consisted of a professional Finnish imitator impersonating five Finnish public figures. The attack slightly increased the false acceptance rate but not alarmingly.

None of the above works investigated countermeasures against impersonation. Regarding that, the work in [Amin et al., 2013], presents a metric to distinguish natural from disguised voices. It is based on exploring the amount of variation of certain glottal and vocal tract features in impersonators. The authors found that the effect of voice identity on vowel formants is highly dependent on vowel category so they developed a metric for voice disguise that treats formant variability on a vowel-by-vowel basis. This metric was correlated with the subjective ratings of a group of naive listeners. The general conclusion is that impersonation is mainly based on prosodic and stylistic cues so the spectral envelope of the speech remains more or less unchanged. Therefore, impersonators are more effective deceiving human listeners than state-of-the-art SV systems [Evans et al., 2013].

11.2.2 Speech synthesis

Speech synthesis is the artificial production of human speech. The main approaches for speech synthesis can be divided into two classes: those based on concatenating human speech samples, also called unit selection approaches; and those based on statistical parametric models like HMM. Having voice samples of the target speaker, it is relatively simple to set up a speech synthesizer able to fool a SV system. While unit selection approaches require a large amount of samples, a HMM speech synthesizer trained on other speakers can be adapted to the target speaker with few data [Zen et al., 2009,Yamagishi et al., 2009]. There are multiple works that prove the vulnerability of SV systems to synthetic speech. The first ones date from more than a decade ago. In [Masuko et al., 1999], an HMM-based speech synthesizer with models adapted to the target speakers [Masuko et al., 1996,Masuko et al., 1997] spoofed a HMM text-prompted SV system increasing the false acceptance rate from 0% to 70%. In [Masuko et al., 2000], pitch information was added to the SV system but it was not useful to reject synthetic speech. These early studies imposted a small number of speakers. Recently, we find studies employing larger number of speakers and state-of-the-art SV approaches. For example, the works in [De Leon et al., 2010a,De Leon et al., 2010b,De Leon et al., 2011,De Leon et al., 2012a] report false acceptance rates of 91% in the Wall Street Journal dataset when attacking GMM-UBM and GSV-SVM systems with a state-of-the-art HMM synthesizer. In [Galou and Chollet, 2011], the forensic SV tool BATVOX is attacked with synthetic speech obtaining similar outcome.

There have been several efforts to develop countermeasures to distinguish between natural and synthetic speech, however it is yet an open problem. In [Satoh et al., 2001], signals were classified as synthetic if the average inter-frame difference of the log-likelihood was under a threshold. This method assumed that synthetic frames, for a given phonetic unit, were much more similar than those of natural speech. The method was able to reduce the false acceptance rate of synthetic speech without increasing the false rejection rate of natural speech. Years later, two new measures were added to the inter-frame log-likelihood difference [De Leon et al., 2010a,De Leon et al., 2010b]: distance between the MFCC of two realizations of the same utterance aligned with dynamic time warping (DTW), and word-error-rate and sentence-error-rate from an automatic speech recognizer trained on human speech. However these works found those measures not to be robust enough to consistently detect synthetic speech. Other works focus on the difference between the phase of synthetic speech and the one of humans. As human auditory system is insensitive to phase, for simplicity, synthesizers use a minimum phase vocal tract model [Wu et al., 2012]. In [De Leon et al., 2011, De Leon et al., 2012a], relative phase shift detected synthetic speech with 100% accuracy. The disadvantage is that a discrimination model had to be trained for each target speaker of the dataset and for each TTS system. Other countermeasures are based on statistics over the pitch frequency of the signals [Ogihara et al., 2005, De Leon et al., 2012b]. We can take advantage of that pitch contours of HMM synthesizers are usually over-smoothed and that contours of synthesizers based on concatenating speech samples present discontinuities. Finally, we find a countermeasure borrowed from face recognition. Analysis of the sequence vectors with local binary patterns (LBP) was effective to detect synthetic speech and other artificial signals in [Alegre et al., 2013b].

11.2.3 Voice conversion

Voice conversion refers to different techniques to make a speaker's voice to sound like another person. Development of this technology whose development started at the end on the eighties [Abe et al., 1988, Valbret et al., 1992, Kain and Macon, 1998, Stylianou and Cappe, 1998, Stylianou et al., 1998] and it has reached maturity in the last decade [Stylianou, 2009]. Since SV technology became popular, it was clear that it could be vulnerable to voice conversion [Pellom and Hansen, 1999]. The work in [Perrot et al., 2005] used a converter based on voice coding to increase EER from 16% to 26% on the NIST SRE04 dataset. In [Matrouf et al., 2006, Bonastre et al., 2007], the transfer function of the impostor was transformed into the transfer function of the target speaker. For each frame, the target transfer function is estimated with a GMM model of the target speaker. The system has two tied sets of acoustic models. The first set models the features employed by the SV system; it is used to estimate the component resposivities of the GMM. The second set contains the parameters of the transfer functions. With this method EER increased from 8% to 60% on NIST SRE05 and from 6% to 28% on NIST SRE06. The work in [Kinnunen et al., 2012] experimented with a converter based on joint density GMM [Kain and Macon, 1998]. This method requires a parallel training corpus for the source and the target speaker. They tested two state-of-the art SV systems: SVM-GMM and JFA. The most robust was JFA where false acceptance rate in the EER operating point increased from 3% to 17% on a subset of NIST SRE06. In [Kons and Aronowitz, 2013], experiments on the West Fargo corpus [Aronowitz et al., 2011] show that a simple voice transformation technique can increase EER by 3–4 times. Authors evaluated three state-of-the-art SV systems with a text-dependent setup: i-vectors, GMM-NAP and HMM-NAP. In [Wu et al., 2013], the authors compare the vulnerability of text-dependent and text-independent SV systems to voice conversion. They experimented with two voice converters: one based on unit selection and another based on JD-GMM. Results on the RSR2015 dataset [Larcher et al., 2012] showed that text-dependent systems are more robust to spoofing. Other work related to voice conversion was presented in [Alegre et al., 2012b]. This work proves that selected short intervals of converted voiced speech produce very high scores. Artificial signals created from those intervals can cheat SV. Experiments on NIST SRE05 show that EER increases from

8.5% to 77% for a GMM-UBM system; and from 4.8% to 65% for JFA.

We find some recent works regarding the development of countermeasures to detect converted speech. Two measures to detect tone-like artificial signals were presented in [Alegre et al., 2012a]: high level features (HLF) based on detecting frame repetition and ITU-T P.563 quality assessment recommendation [ITU-T, 2004]. HLF yielded perfect spoofing detection for the type of signals evaluated. The countermeasure in [Alegre et al., 2013a] exploits the common shift applied to the spectral slope of consecutive speech frames when mapping the impostor towards the model of the target speaker. It yielded a spoofing detection EER of 2.7%. The same authors also applied local binary patterns (LBP) to voice conversion detection [Alegre et al., 2013b] but they obtained worse performance (EER=8%).

11.2.4 Replay attacks

Replay attacks consist in feeding the SV system with a recording acquired from the victim. If the SV system is text-dependent, it also includes the possibility of cutting and pasting short segments to build the spoofing sentence [Lindberg and Blomberg, 1999]. This kind of spoof is one of the most dangerous because high quality recordings are practically indistinguishable from real speech so they will obtain the same SV scores as the true target. Besides, the attacker does not need any advanced technical knowledge on the contrary of voice conversion or synthesis. There are few works addressing the problem of detecting replay attacks. Due to the difficulty of this task, some authors propose fusing speech and visual signals to detect imposture. The work in [Bredin et al., 2006] describes an algorithm to detect the lack of correspondence between speech and lip motion. However, there are already works that present methods to generate animated synthetic faces that synchronize speech and lip movement [Karam et al., 2009, Chollet et al., 2012].

11.3 Task Description

We worked on detecting two types of replay attack. In the first place, we considered the simple attack consisting in just playing a recording of the victim without any alteration. Text-independent SV systems are clearly vulnerable to this kind of attack as well as text-dependent systems with fixed pass-phrase. Systems that request the user to utter a different sentence for each access attempt (text-prompted system) are not vulnerable to this kind of attack. However, text-prompted systems can be attacked by manufacturing the pass-phrase by joining speech fragments (words, syllables) from different target speaker's recordings. We also worked on the detection of this attack, which we called *cut and paste* spoofing attack.

Detection of replay attacks is complicated unless we make some assumptions. First, we assumed that our SV system was intended for a telephone application where the handset is close to the speaker's mouth (close-talk). That implies that non-spoof signals will have high quality with low levels of noise and reverberation. Second, we expect that the victim does not collaborate with the spoofer to perpetrate the attack. That means that, probably, the criminal will have to record the victim from a certain distance and he will not obtain a high quality sample. Third, we supposed that the attacker will play the recording in front of the telephone handset by using a portable device (portable recorder, smartphone). The small loudspeakers in those devices exhibit frequency responses far from the ones of HI-FI equipment. Thus, our algorithm for replay attack detection combined two things:

discriminating between far-field and close-talk recordings and detecting that the speech signal has been generated by a loudspeaker.

Regarding the cut and paste attack, we used a SV system that is not text-dependent in the strict sense. The system is text-dependent in the sense that the length of the enrollment utterances is very sort what implies that target speakers only obtain a high verification score if the sentence uttered in the test is the same or, at least, similar to the enrollment ones. In our setup, the user was enrolled with three utterances of two different pass-phrases. In the test phase, the speaker is asked to utter any of the enrollment pass-phrases. Then, the *cut and paste* detection task was divided into two subtask:

- Pass-phrase detection: determining which one of the enrollment pass-phrases was uttered in the test segment.
- Cut and paste detection: detecting whether the test utterance was made by concatenating words. In this part we compared the test utterance with the enrollment utterances corresponding to the pass-phrase selected in the previous step.

If the original sentences, from which we extracted the words to make the spoofing sentence, were recorded by a far-field microphone, the fusion of both spoofing detectors improved performance.

11.4 Cut and Paste Detection System

The cut and paste detection algorithm was based on the distance between the feature contours of the test and reference segments. Contours were aligned by dynamic time warping (DTW) [Huang et al., 2001]. We hypothesized that these contours should be very different between a legitimate sentence and another one made of several recordings. The fact that the words in the pass-phrase are not in the same position than in the source recordings can make that intonation and energy patterns not to be the same as those in the enrollment segments. For example, discontinuities may appear in the pitch and energy contours or we may find a rising intonation pattern in the test sentence while we have a descending pattern in the enrollment segments and vice versa. The mood of the utterances can also be different. While in non-spoof enrollment and test sentence we find a more or less neutral intonation pattern, the spoofing sentences could present interrogative, exclamative patterns, etc. depending on the situation where they were recorded.

11.4.1 Features

The features employed by the system were:

- Log-pitch: the logarithm of the speech fundamental frequency f_p was computed with a pitch extractor based on the RAPT implementation [Talkin, 1995]. This implementation includes pitch tracking by dynamic programming.
- MFCC: 12 Mel Filtered Cepstral Coefficients (MFCC C1-C12) were extracted. Mean and variance normalization of the MFCC (CMVN) was used to mitigate channel mismatch between enrollment and test segments. We used our VAD based on [Ramirez et al., 2004] to prune leading and trailing silence segments. We did not remove the

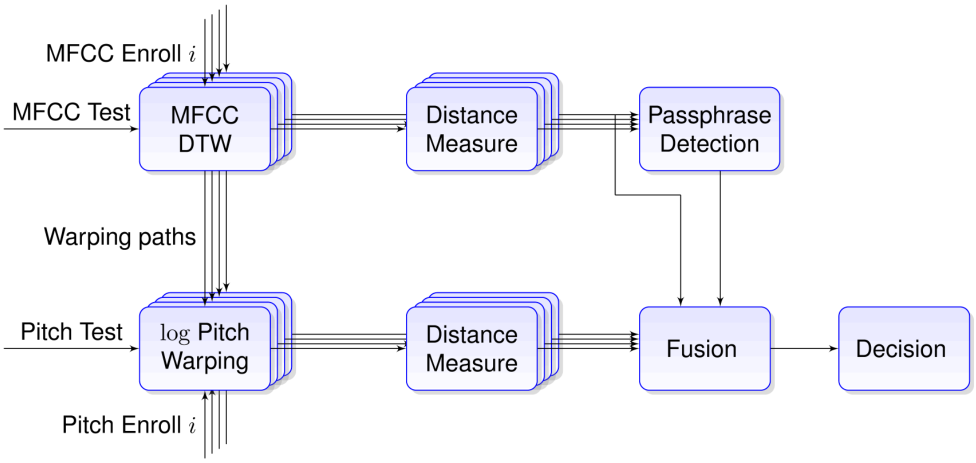


Figure 11.1: Cut and Paste matching algorithm.

silence segments in the middle of the sentence because it would damage the DTW alignment.

11.4.2 Matching algorithm

Algorithm 2 summarizes our matching algorithm to compare enrollment and test segments. First, our algorithm used the MFCC features to obtain the warping path that best aligns both sentences. Then, we used that path to align both log-pitch and MFCC contours. We observed that the warping obtained from MFCC was better than the one obtained based on pitch due to the halving and doubling pitch errors. We computed the Mahalanobis distance between enrollment and test aligned feature contours. We used the distance between MFCC contours to decide which phrase was uttered in the test segment. Finally, we fused the MFCC and log-pitch distances to decide if the test was spoof or non-spoof. Figure 11.1 shows a block diagram of the system.

In the next sections we explain more in detail each of the steps.

11.4.2.1 MFCC distance measure

The MFCC of the test segment t were aligned with each one of the enrollment utterances r_i of the target speaker by DTW. Then, we calculated the Mahalanobis distance:

$$d_{\text{MFCC}}(r_i,t)^2 = \frac{1}{T}\sum_{k=1}^{T}\sum_{j=1}^{D}\frac{\left(r_{w_{ij}}(k) - t_{w_j}(k)\right)^2}{\sigma^2_{w_{ij}}} \tag{11.1}$$

where r_{w_i} is the i^{th} warped reference signal, t_w is the warped test signal, j is the MFCC component index and k is the temporal index. We used the Mahalanobis distance to account for the different dynamic ranges of each of the MFCC components. The variances σ^2_{wi} were computed on the warped reference segment r_{w_i}.

11.4.2.2 Pitch Distance Measure

For the pitch distance, first, we warped the log-pitch contour of the reference and test signals by applying the warping path that we obtained from the DTW alignment of the MFCC. We

Algorithm 2 Cut and paste detection algorithm.

```
Given a test sentence $t$ and a set of enrollment sentences $r_i$.
for each pair of sentences $(r_i, t)$ do
    Perform joint DTW alignment between the 12 MFCC of both sentences.
    Use optimal warping path to warp MFCC and log-pitch contours.
    Calculate Mahalanobis distance between the warped MFCC of both sentences:
      • Compute the variances needed to evaluate the Mahalanobis distance on the warped
        MFCC of the reference signal $r_i$.
      • Normalize the distance by the number of samples.
    Calculate the Mahalanobis distance between the warped log-pitch of both sentences:
      • Take into account only points where pitch was detected on both signals.
      • Detect possible halving and doubling pitch errors (pitch in one signal almost double
        than in the other).
      • Compute the variances needed to evaluate the Mahalanobis distance on the warped
        log-pitch of the reference signal $r_i$.
      • Normalize the distance by the number of samples with pitch in both segments.
    Fuse MFCC and log-pitch distances with a weighted sum.
end for
Decide which pass-phrase was uttered by the speaker based on the minimum MFCC
distance.
The spoofing detection score is the minimum of the fused distances corresponding to the
pairs $(r_i, t)$ where $r_i$ contains the detected pass-phrase.
Threshold the score to take the final decision.
```

found that, in this manner, we obtained better alignments than doing DTW alignment based on pitch contours. Pitch detection errors like missing pitch segments or halving/doubling errors, in either the reference or the test signal, caused bad DTW alignments. The distances obtained in those cases were too large, which was especially harmful for non-spoof tests. Consequently, false acceptance rates (false spoof) increased dramatically.

Even with a good alignment of the pitch contours, there are frames whose pitch is missing in either the reference or the test segments. We computed distances accounting only on the frames that were marked as voiced in both segments.

To detect halving and doubling errors, for each frame, we checked if the doubled or halved version of the test pitch was almost the same as the reference pitch. Then, we replaced the test pitch value with the nearest one to the reference pitch. Written in mathematical notation:

$$\exists n \in \mathbb{Z} \ni r_w(k) - t_w(k) - n\log(2) < \epsilon \qquad \Rightarrow t_w(k) \leftarrow t_w(k) + n\log(2) \tag{11.2}$$

where $r_w(k)$ is the warped reference log-pitch and $t_w(k)$ is the warped test log-pitch.

We can see the effect of pitch detection errors in Figure 11.2. The figure shows the curves of miss probability and false acceptance probability against decision threshold for two spoofing detectors based on pitch. The left plot corresponds to a system where the pitch has been warped by DTW between the enrollment and test log-pitch contours. The

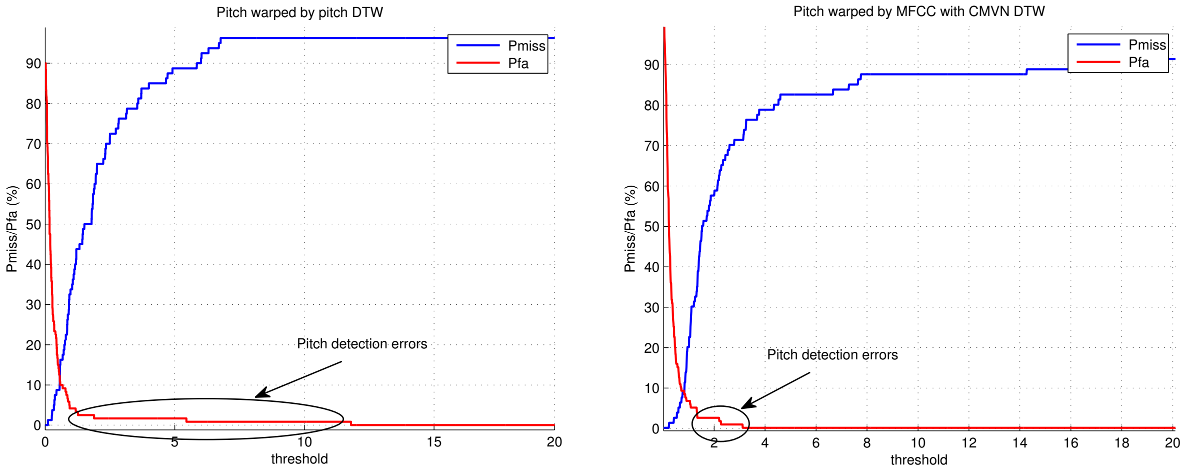


Figure 11.2: Effect of pitch errors on error rates.

right plot corresponds to a system where the pitch has been warped with the path obtained from the MFCC alignment. The figure reveals that, for the left one, there is point where we cannot reduce the false acceptance probability but selecting a very high threshold. However, if we do that we will not detect any true spoofs. On the contrary, for the right one, the effect of pitch detection errors is not so harmful. Thus, we can select an operating point for the spoofing detector where we do not have false alarms and yet we are able to detect a fair amount of spoofing attempts.

Once solved, as far as possible, the problem of pitch detection errors we can proceed to calculate the distance:

$$d_{\log-\text{pitch}}(r_i,t)^2 = \frac{1}{|V|}\sum_{k\in V}\frac{\left(r_{w_i}(k)-t_w(k)\right)^2}{\sigma^2_{w_i}} \tag{11.3}$$

$$V = \{k \ni r_{w_i}(k) > 0 \wedge t_w(k) > 0\} \tag{11.4}$$

where r_{w_i} is the warped log-pitch of the i^{th} reference signal and t_w is the warped pitch of the test signal. V is the set of frames that are voiced in both the reference and the test segments. The variances $\sigma^2_{w_i}$ were computed on the warped reference segment r_{w_i}.

11.4.2.3 Pass-phrase detection

In our setup, users authenticate themselves by uttering any pass-phrase among those recorded in the enrollment. We based on the MFCC distance to detect which pass-phrase the user selected. According to our results, MFCCs chose the right pass-phrase better than pitch. Not in vain, MFCCs are the standard features for speech recognition systems. We decided that the pass-phrase in the test segment should be the same as the pass-phrase in the enrollment segment with minimum d_{MFCC}:

$$\hat{s}(t) = s\left(r_n\right) \ni n = \underset{i}{\text{argmin}}\; d_{\text{MFCC}}(r_i,t) \tag{11.5}$$

where $s(f)$ is the pass-phrase uttered in the segment f.

11.4.2.4 Fusion

We computed a weighted sum of MFCC and log-pitch distances for the pairs (r_i, t) where r_i contains the same pass-phrase as t:

$$R = \left\{ r_i \ni s(r_i) = \hat{s(t)} \right\} \tag{11.6}$$

$$d(r_i, t) = w_1 d_{\text{MFCC}}(r_i, t) + w_2 d_{\text{log-pitch}}(r_i, t) \ni r_i \in R \ . \tag{11.7}$$

The output score of the spoofing detector was the minimum of those weighted distances:

$$d = \min_{r_i \in R} d(r_i, t) \ . \tag{11.8}$$

11.5 Far-Field Replay Attack Detection System

The far-field detector is based on a SVM working on a set of acoustic features extracted from the speech signal.

11.5.1 Features

For each recording, we selected a set of features that could detect two types of manipulations of the speech signal:

- The signal was acquired by a far-field microphone.
- The signal was replayed by a loudspeaker.

Currently, speaker verification systems are mostly used on telephone applications. That implies that the user is supposed to be close to the telephone handset. If we detect that the user was far from the handset during the recording we consider it a spoofing attempt. Far-field recordings present larger noise and reverberation levels than close-talk ones. As consequence, their spectra are flatter and their modulation index smaller.

A loudspeaker is the simplest means of injecting the spoofing signal into a phone-call. Probably, the impostor will use a easily transportable device like a smart-phone, with a small loudspeaker. This kind of loudspeakers presents bad frequency responses in the low part of the spectrum. Figure 11.3 shows a typical frequency response of a smart-phone loudspeaker that evidences that low frequencies, under 500 Hz, are strongly attenuated.

Following, we describe our features.

11.5.1.1 Spectral Ratio

We defined the spectral ratio (SR), for a frame n as:

$$\text{SR}(n) = \sum_{f=0}^{\text{NFFT}/2-1} \log\left(|X(f,n)|\right) \cos\left(\frac{(2f+1)\pi}{\text{NFFT}}\right) \tag{11.9}$$

where $X(f, n)$ is the spectrogram of the signal and NFFT is the size of the FFT. Speech frames were windowed with a Hamming window. Approximately, this measure represents the log-ratio between the energy in the frequency range 0–2 kHz and the one in 2–4 kHz. We averaged $SR(n)$ over all speech frames to compute a unique value for each segment. Spectrum flattening, due to noise and reverberation, implies lower values of SR.

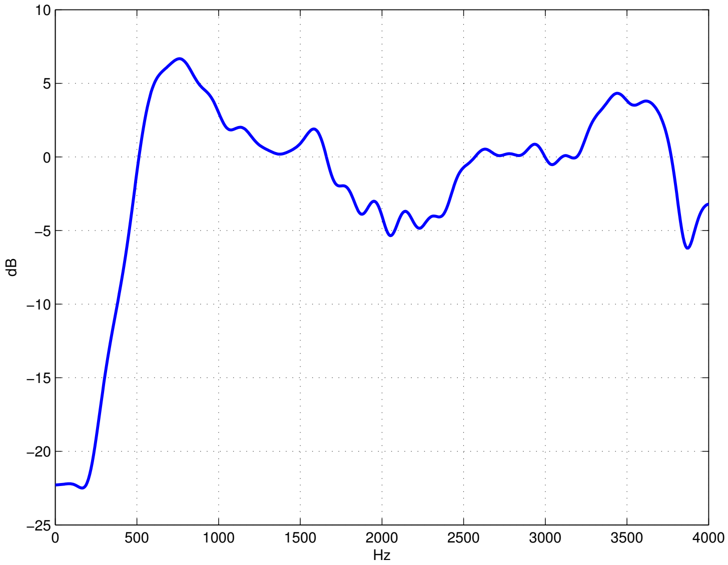


Figure 11.3: Typical frequency response of smartphone loudspeaker.

11.5.1.2 Low Frequency Ratio

We called low frequency ratio (LFR) to the log-ratio between the signal energy in 100–300 Hz and the one in 300–500 Hz. For a frame n, we calculated it as:

$$\mathrm{LFR}(n) = \sum_{f=100\mathrm{Hz}}^{300\mathrm{Hz}} \log\left(|X(f,n)|\right) - \sum_{f=300\mathrm{Hz}}^{500\mathrm{Hz}} \log\left(|X(f,n)|\right) \qquad (11.10)$$

where $X(f,n)$ is signal spectrogram. For each segment, we computed the average LFR after removing the silence frames. Signals replayed by a loudspeaker present lower values of LFR.

11.5.1.3 Modulation Index

Modulation index was already introduced in Section 4.2. The modulation index at time t is defined as

$$\mathrm{Indx}(t) = \frac{v_{\max}(t) - v_{\min}(t)}{v_{\max}(t) + v_{\min}(t)} \qquad (11.11)$$

where $v(t)$ is the envelope of the signal and $v_{\max}(t)$ and $v_{\min}(t)$ are the local maximum and minimum of the envelope in a region close to the time t. The envelope was approximated by the absolute value of the signal $s(t)$ down-sampled to 60 Hz. For each segment, we computed the average modulation index taking the frames whose indexes were above 0.75. Noise and reverberation cause higher values of $v_{\min}$ and lower modulation indexes.

11.5.1.4 Sub-band Modulation Index

If noise only affects a small frequency band it may not be noticed by the standard modulation index. We calculated the modulation index on several sub-bands to detect far-field recordings with colored noises. The sub-band modulation index was calculated by filtering the signal with a band-pass filter over the desired frequency range before feeding it to the standard modulation index algorithm. We computed indexes for the bands: 1–3 kHz, 1–2 kHz, 2–3 kHz, 0.5–1 kHz, 1–1.5 kHz, 1.5–2 kHz, 2–2.5 kHz, 2.5–3 kHz, 3–3.5 kHz.

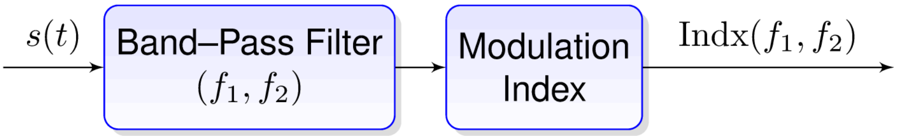


Figure 11.4: Sub-band modulation index calculation.

11.5.2 Classification algorithm

For each recording, we built a feature vector $\mathbf{x}$ with all the features described in the previous section:

$$\mathbf{x} = (\mathrm{SR}, \mathrm{LFR}, \mathrm{Indx}(0, 4\mathrm{kHz}), \ldots, \mathrm{Indx}(3\mathrm{kHz}, 3.5\mathrm{kHz}))\ . \tag{11.12}$$

Feature vectors were classified with SVM. We remind that the SVM classification function is defined as:

$$f(\mathbf{x}) = \sum_i \alpha_i \mathrm{k}(\mathbf{x}, \mathbf{x}_i) + b \tag{11.13}$$

where k is the kernel function, $\mathbf{x}$ is the test vector, and $\mathbf{x}_i$, α_i and b are the support vectors, the support vector weights, and the bias parameter that are estimated during the SVM training process. The kernel that best performed was the Gaussian kernel:

$$\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\gamma \left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2\right)\ . \tag{11.14}$$

We trained our classifier with the LIBSVM toolkit [Chang and Lin, 2011]. The training data was obtained from the NIST SRE08 database:

- Non spoofs: 1788 telephone signals from the NIST SRE08 train set.
- Spoofs: we created artificial spoofs from the far-field microphone interviews in the NIST SRE08 train set. We passed those signals through a loudspeaker and a telephone channel to simulate the conditions of a real spoof. We used two different loudspeakers: a USB loudspeaker for a desktop computer and a mobile device loudspeaker; and two different telephone channels: analog and digital. In this way, we obtained 1475 × 4 spoof signals.

11.6 Experiments

11.6.1 Databases

At the moment of this work, there were no publicly available databases for this task. In our experiments, we used two datasets provided by a third party.

11.6.1.1 Database 1: far-field

The first database was designed to detect replay attacks on text-independent speaker recognition. It consisted of only 5 speakers. Each speaker had 4 groups of signals:

- Originals: signals recorded by a close-talk microphone and transmitted over a telephone channel. There were 1 enrollment signal and 7 test signals per speaker. They were transmitted over different telephone channels: digital (enrollment and 3 test signals), analog wired (2 test signals) and analog wireless (2 test signals).

- Microphone: signals recorded simultaneously to the originals by a far-field microphone.
- Analog Spoof: the microphone test signals replayed on a telephone handset and transmitted over an analog channel.
- Digital Spoof: the microphone test signals replayed on a telephone handset and transmitted over a digital channel.

11.6.1.2 Database 2: far-field + cut and paste

This database was recorded to experiment with replay attacks on text-dependent speaker recognition. During the test phase, text-dependent systems ask the user to utter a given pass-phrase. The spoofing process consists in manufacturing the test utterance by cutting and pasting fragments of speech (words, syllables) obtained from victim's available recordings. The database consisted of two parts:

- Part 1: it consisted of 20 speakers. It included landline (T) signals for enrollment, non-spoof and spoof tests; and GSM (G) signals for spoofs tests.
- Part 2: it consisted of 10 speakers. It included landline and GSM signals for all enrollment, non-spoofs and spoof tests. It was recorded several months later than Part 1 and both had seven speakers in common.

Recordings were made in three sessions:

- Session 1: it was used for speaker enrollment. Each speaker recorded 3 utterances per channel type of 2 different pass-phrases (F_1, F_2). Each utterance was around 2 seconds long. The pass-phrases were:
  - F_1: ”Como manzanas en casa” (I eat apples at home).
  - F_2: ”Utilizo biometra de voz en mi trabajo” (I use voice biometrics at work).
- Session 2: it was used for non-spoofing access trials and consisted of 3 recordings per channel type for each of the sentences F_1 and F_2.
- Session 3: it was composed of different sentences and a long text that contained the same words as those that appear in the sentences F_1 and F_2. They were recorded by a far-field microphone. Spoofing trials were created from this session. First, the following excerpts were extracted from the recordings:
  - ”En mi trabajo” (at work).
  - ”Biometra de voz” (voice biometrics).
  - ”Utilizo” (I use).
  - ”Como” (I eat).
  - ”En casa” (at home).
  - ”Manzanas” (apples).

  Then, excerpts were concatenated to obtain 6 samples per pass-phrase. Finally, the signals were played on a telephone handset and transmitted over a landline or a GSM channel.

Three kinds of compositions were made:

- C_1: Words were in the same place (beginning, middle or end) than in the source sentences.
- C_2: Words were in different place than in the source sentences.
- C_3: Words were extracted from a long text.

We considered that compositions C_1 were non-realistic so we only experimented on C_2 and C_3.

Summing up, in this dataset we find these channel conditions:

- Telephone landline (T):
  - Enrollment and non-spoof speech recorded directly over landline telephone channel (Sessions 1 and 2).
  - Spoof sentences, recorded by a far-field microphone, and replayed with a loudspeaker on the telephone handset (Session 3).
- GSM (G):
  - Enrollment and non-spoof speech recorded directly over GSM mobile channel (Sessions 1 and 2 of Part 2).
  - Composed spoof sentences, recorded by microphone, and replayed with a speaker on a mobile phone (Session 3).

Note that the spoofs in this dataset included both types of manipulation: cut and paste and replay attack with far-field recording and loudspeaker.

11.6.2 Speaker verification system

We evaluated how spoofing attacks degrade the performance of a SV system based on JFA [Kenny et al., 2008]. The system was similar to the one described in Section 2.8.1. However, we trained it in a gender independent fashion to allow cross-gender trials. We extracted 20 MFCCs (C0-C19) plus first and second derivatives from the speech segments. After frame selection, features were short time Gaussianized as in [Pelecanos and Sridharan, 2001]. A gender independent Universal Background Model (UBM) of 2048 Gaussians was trained by EM iterations. Then, 300 eigen-voices $\mathbf{V}$ and 100 eigen-channels $\mathbf{U}$ were trained by EM ML+MD iterations. Speakers were enrolled by making MAP estimates of the speaker dependent factors ($\mathbf{y}$,$\mathbf{z}$). Trials were scored by the first order Taylor approximation of the log-likelihood ratio between the target and the UBM models, as shown in Equation (2.13). Scores were ZT-normalized and calibrated to log-likelihood ratios by linear logistic regression [Brummer and De Villiers, 2011] on the NIST SRE08 core trial list [NIST Speech Group, 2008]. We used telephone data from NIST SRE04-06 for UBM and JFA training; and for score normalization.

11.6.3 Speaker verification performance degradation

The results in this section evidence that spoofing degrades the performance of state-of-the-art SV. We experimented on both of our databases.

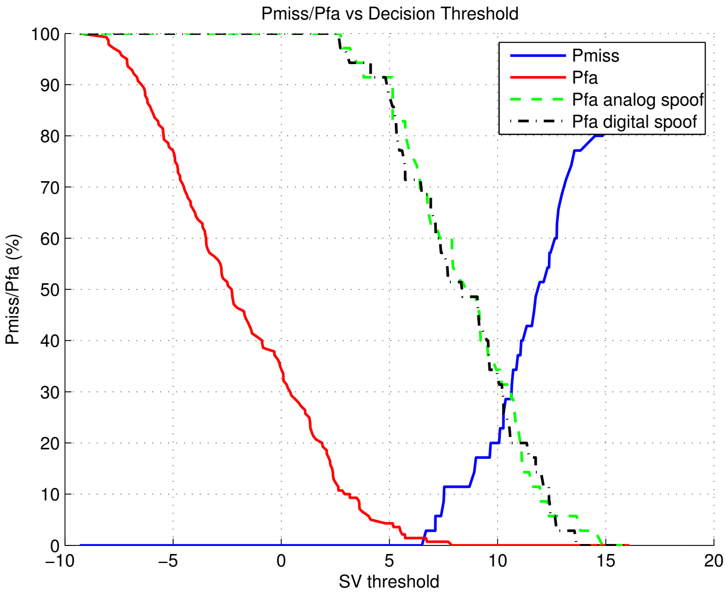


Figure 11.5: $P_{\mathrm{Miss}}/P_{\mathrm{FA}}$ against decision threshold for the spoofing Database 1.

11.6.3.1 Database 1

Database 1 had 35 legitimate target trials, 140 non-spoof non-target trials, 35 analog spoofs and 35 digital spoofs. Approximately, the enrollment segments were 60 seconds long and the test segments 5 seconds long. We obtained an EER of 0.71% for non-spoof trials. Figure 11.5 plots the miss and false acceptance rates of our SV system against the SV decision threshold for the four types of trials. The figure reveals that if we work on the EER operating point, we will accept 68% of the spoofing trials. Analog and digital spoofs presented similar false acceptance rates. To reduce the spoofing acceptance rates, we should increase the threshold at the cost of worsen the miss rate. For example, we can select a threshold that makes $P_{\mathrm{Miss}} = P_{\mathrm{FA-spoof}} = 28\%$.

Figure 11.6 shows the score distribution for each group of trials. The curves evidence a large overlap between the true targets and the spoofing attempts. Table 11.1 shows statistics for the score difference between a legitimate utterance and its spoofed version (Remember that non-spoof and spoof tests were recorded simultaneously). Note that some spoof tests obtained larger scores than their non-spoof counterparts.

Table 11.1: Score reduction due to replay attack for spoofing Database 1.

ΔLLR	Mean	Std	Median	Max	Min
Analog	3.38	2.42	3.47	9.70	-1.26
Digital	3.52	2.30	3.37	9.87	-1.68

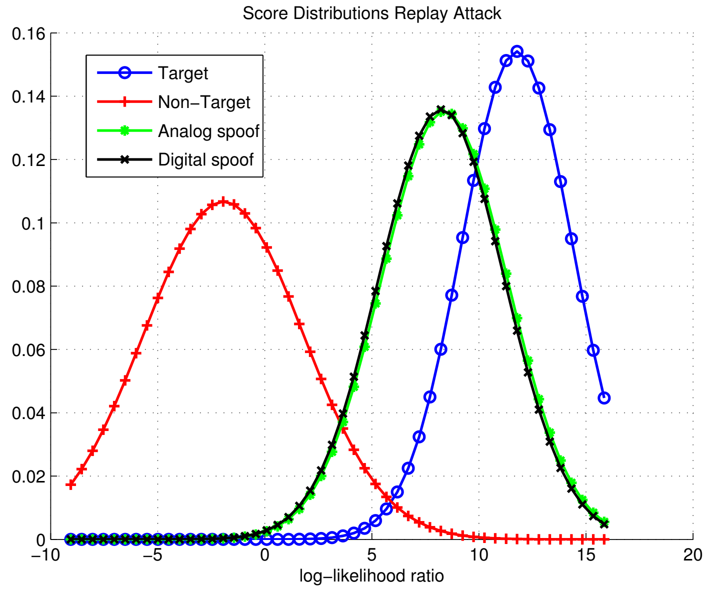


Figure 11.6: SV score distributions for the spoofing Database 1.

11.6.3.2 Database 2

We experimented on the parts one (P_1) and two (P_2) of the database separately. For P_1, we enrolled each speaker on 6 landline utterances; and evaluated 120 legitimate target trials, 2280 non-spoof non-targets, 80 landline spoofs and 80 GSM spoofs. For P_2, the speakers' models were derived from 12 utterances (6 landline + 6 GSM); and we scored 120 legitimate target trials (60 landline + 60 GSM), 1080 non-spoof non-target (540 landline + 540 GSM) and 80 spoofs (40 landline + 40 GSM).

With non-spoofing trials, we obtained EER=1.66% and EER=5.74% for P_1 and P_2 respectively. Figure 11.7 displays the miss and false acceptance rates against the SV decision threshold for the P_1 set. Working on the EER operating point 5% of landline spoofs are accepted but none of the GSM spoofs.

Figure 11.8 shows the score distributions for the trial sets in Database 2. For P_1, the spoofing scores were much lower than the true target scores but yet higher than the non-target scores. For P_2, the spoofing scores were lower than the non-target scores so none of them would cheat the system. It seems that the setup used to create the spoofing signals in P_2 was somehow different than the one used in P_1. As a result, the channel of P_2 spoofs was so different from the enrollment than JFA was not able to compensate it. Poor channel compensation can be explained by the length of the utterances. It is well known that, in short utterances, the channel factors cannot be properly estimated. Table 11.2 shows statistics for the difference between the target and the spoofing SV scores. Differences were calculated per speaker and pass-phrase and then averaged. ΔLLR were much larger than the ones in Database 1.

11.6.4 Experiments cut and paste detection

We tried our cut and paste detector on Database 2. We used both parts of the database despite that the trials in the Part 2 did not pass the SV system.

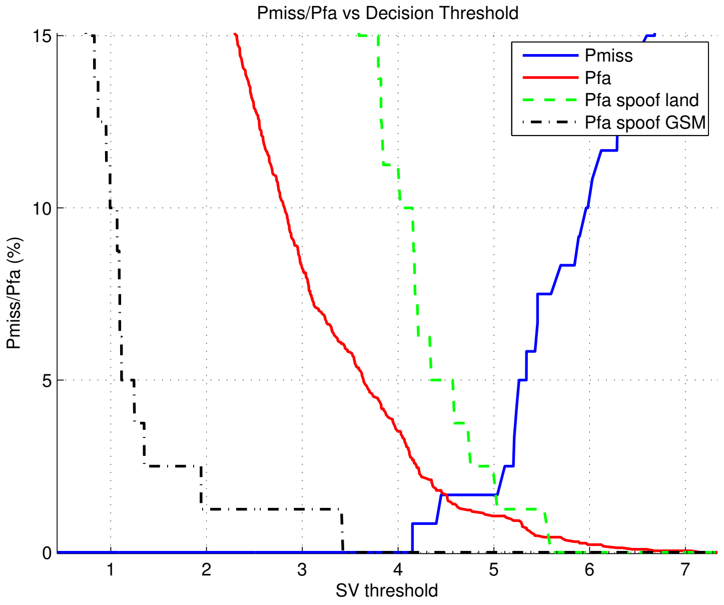
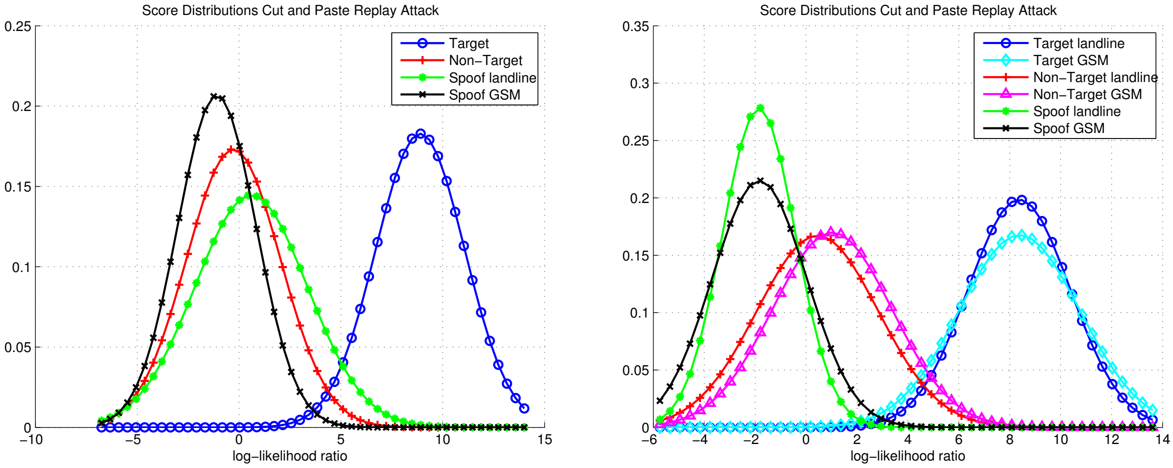


Figure 11.7: $P_{\text{Miss}}/P_{\text{FA}}$ vs decision threshold for Database 2 part 1.



(a) Part 1

(b) Part 2

Figure 11.8: SV score distributions for spoofing Database 2.

Table 11.2: Score reduction due to spoofing in Database 2.

ΔLLR		Mean	Std	Median	Max	Min
P_1	T	8.29	3.87	7.96	17.89	1.41
	G	9.98	2.96	9.56	18.51	5.40
P_2	T	10.21	2.51	9.76	17.78	6.86
	G	10.21	3.32	10.19	18.36	4.65

11.6.4.1 Results for multiple channel conditions

We experimented with different combinations of the type of telephone channels used for enrollment, non-spoof and spoof test. Table 11.3 presents the EER of spoofing detection (not speaker verification) obtained. We compare the detectors based on the MFCC, the log-pitch and the fusion of both. The nomenclature that defines each channel condition is ”enroll-channel_non-spoof-test-channel_spoof-test-channel”. Each one can be landline (T), GSM (G), or mixed channel (TG). For example, the condition T_T_TG indicates that we used landline for enrollment and non-spoof tests; and landline or GSM for spoof tests.

The MFCC produced better results than pitch. That was due to pitch detection errors and to that MFCCs are also very affected by the channel mismatch due to the spoofing creation procedure. We remind that the session used to create the spoofs was recorded by a far-field microphone. Thus, even for conditions like T_T_T there was still some channel mismatch between spoofs and non-spoofs. Despite that the MFCC were better, we think fusing both features is more robust to prevent spoofs made from high quality recordings. The fusion weights were tuned on the test data because we did not have any held-out set. The results were quite similar across channel conditions.

Table 11.4 shows pass-phrase identification error rates. The system always chose the correct pass-phrase in non-spoofing trials. This is important because otherwise, we would obtain a high false spoof detection rate. On the other hand, the error rate of the spoofing signals was very high. This, far from being a problem, improved the detection of true spoofs.

11.6.4.2 Results for sessions separated different time intervals

Part 2 of database 2 was recorded several months later than Part 1. Both parts had seven speakers in common. We used the data from those seven speakers to evaluate the performance of the algorithm when enrollment and test sessions are long time separated. We set up an experiment where the session 1 of Part 1 was used for enrollment and the rest of sessions of Part 1 and 2 for test. Table 11.5 shows the spoofing EER obtained. The top block of the table corresponds to experiments with the same type of telephone channel for all sessions (landline); and the bottom, to mixed channel types in test. The nomenclature that defines each condition is similar to the one in the previous section. We added two numbers to indicate the part and the session of the data. For example, condition T11_G21_T13 indicates that enrollment signals were landlines from the session 1 of Part 1, non-spoofs tests were GSM from the session 1 of Part 2; and spoofs tests were landlines from the session 3 of Part 1.

Table 11.3: EER(&) for cut and paste detection with multiple channel conditions.

EER(%)		MFCC	Pitch	Pitch + MFCC
P_1	T_T_T	0.00	6.88	0.00
	T_T_G	0.00	3.12	0.00
	T_T_TG	0.00	6.46	0.00
P_2	T_T_T	0.00	0.00	0.00
	T_G_G	0.00	0.00	0.00
	T_TG_TG	0.00	1.04	0.00
	G_T_T	0.00	2.92	0.00
	G_G_G	0.00	2.92	2.08
	G_TG_TG	0.00	3.96	1.04
	TG_T_T	0.00	2.92	0.00
	TG_G_G	0.00	0.83	0.00
	TG_TG_TG	0.00	2.50	0.00

Table 11.4: Pass-phrase detection error rates (%) for multiple channel conditions.

		$P(\text{error})$	$P(\text{error}\vert\text{non} - \text{spoof})$	$P(\text{error}\vert\text{spoof})$
P_1	T_T_T	13.00	0.00	32.50
	T_T_G	17.00	0.00	42.50
	T_T_TG	21.43	0.00	37.50
P_2	T_T_T	17.00	0.00	42.50
	T_G_G	15.00	0.00	37.50
	T_TG_TG	16.00	0.00	40.00
	G_T_T	17.00	0.00	42.50
	G_G_G	10.00	0.00	25.00
	G_TG_TG	13.50	0.00	33.75
	TG_T_T	19.00	0.00	47.50
	TG_G_G	12.00	0.00	30.00
	TG_TG_TG	15.50	0.00	38.75

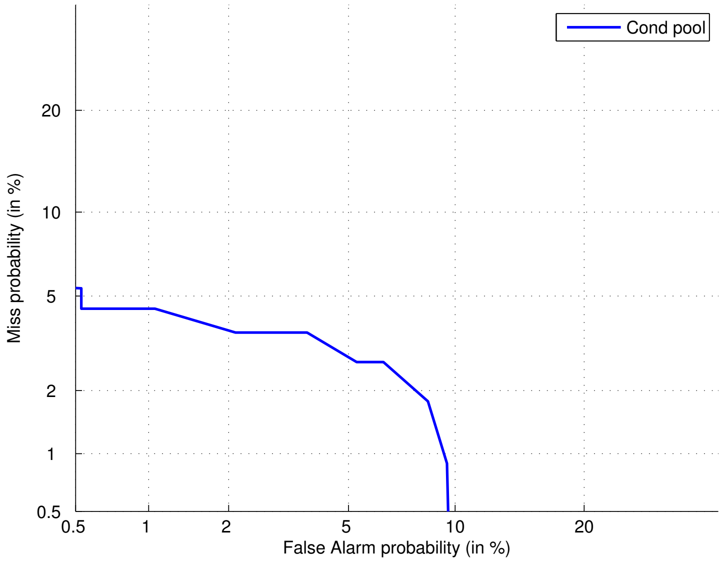


Figure 11.9: DET curve for the pooling of different channel conditions and session separations.

Both, the pitch and the MFCC, yielded worse results when the enrollment session was further in time from the non-spoofs tests, especially when there were mixed test channels. Results were better when we did not have channel mismatch between enrollment and test. There was not a significant difference between spoofs from Part 1 (close to the enrollment session) and the ones from Part 2 (far from the enrollment). Although in the previous section, we observed that the MFCCs were better than the log-pitch and that the fusion did not improve, Table 11.5 indicates that, when enrollment and test are distant, the MFCCs and the log-pitch provided similar performance. In this case, fusion clearly helped. We want to make note that we had very few data for this experiment, so we could not measure low error rates precisely. In fact, each classification error caused an EER absolute increment of about 3%. In Figure 11.9, we plot a DET curve for the detector fusing log-pitch and MFCC, where we pooled all the conditions of the table. EER was around 4%. Interestingly, the misses and false alarm rates were bounded, that is, independently of the operating point that we set for the spoofing detector, we always missed less than 5% of spoofs and the false alarms were less than 10%. We are aware that this behavior may be particular to this dataset and it may not generalize to others.

Table 11.6 shows the pass-phrase identification error rates for this experiment. The system was always able to select the right pass-phrase in non-spoofs even when enrollment and test segments were distant in time. For the spoofing trials, pass-phrase selection was almost random, but there is no reason why it should damage the detection of spoof tests.

11.6.5 Results far-field replay attack detection

In this section, we evaluate the performance of the countermeasure that detects far-field recordings and signals replayed by a loudspeaker. We experimented on Database 1 and 2.

Table 11.5: EER(%) for cut and paste detection with different distance between sessions.

EER(%)		MFCC	Pitch	Pitch + MFCC
Landline	T11_T12_T13	0.00	2.98	0.00
	T11_T12_T23	0.00	2.98	0.00
	T11_T21_T13	4.17	4.17	0.00
	T11_T21_T23	4.17	7.14	0.00
	T11_T22_T13	0.00	4.17	0.00
	T11_T22_T23	0.00	4.17	0.00
Mixed channel	T11_T12_TG13	0.00	2.08	0.00
	T11_T12_TG23	0.00	2.08	0.00
	T11_TG21_TG13	7.14	7.14	2.08
	T11_TG21_TG23	7.14	2.98	3.57
	T11_TG22_TG13	7.14	7.14	4.17
	T11_TG22_TG23	8.63	5.06	4.17

Table 11.6: Pass-phrase detection error rates (%) for different distance between sessions.

		$P(\text{error})$	$P(\text{error}\|\text{non}-\text{spoof})$	$P(\text{error}\|\text{spoof})$
Land line	T11_T12_T13	21.43	0.00	53.57
	T11_T12_T23	22.86	0.00	57.14
	T11_T21_T13	22.86	0.00	57.14
	T11_T21_T23	21.43	0.00	53.57
	T11_T22_T13	21.43	0.00	53.57
	T11_T22_T23	22.86	0.00	57.14
Mixed channel	T11_T12_TG13	27.55	0.00	48.21
	T11_T12_TG23	26.53	0.00	46.43
	T11_TG21_TG13	19.29	0.00	48.21
	T11_TG21_TG23	18.57	0.00	46.43
	T11_TG22_TG13	19.29	0.00	48.21
	T11_TG22_TG23	18.57	0.00	46.43

Table 11.7: EER(%) for far-field spoofing detection in Database 1.

Channel	Features	EER(%)
Analog orig. vs. Analog spoof	SR	20.00
	LFR	0.00
	MI	30.7
	Sb-MI	10.71
	SR,MI,Sb-MI	0.00
	SR,LFR,MI,Sb-MI	0.00
Digital orig. vs. Digital spoof	SR	36.07
	LFR	0.00
	MI	30.7
	Sb-MI	14.64
	SR,MI,Sb-MI	10.71
	SR,LFR,MI,Sb-MI	0.00
Analog+Dig orig. vs. Analog+Dig spoof	SR	37.32
	LFR	7.32
	MI	31.9
	Sb-MI	12.36
	SR,MI,Sb-MI	8.03
	SR,LFR,MI,Sb-MI	8.03

11.6.5.1 Database 1

Table 11.7 shows the spoofing detection EER for different features and channel conditions. The input to the SVM classifier were spectral ratio (SR), modulation index (MI), sub-band modulation indexes (Sb-MI) and low frequency ratio (LFR), as described in Section 11.5.1. We also combined SR, MI and Sb-MI; and SR, LFR, MI and Sb-MI. We considered three channel conditions: only analog telephone signals for spoof and non-spoofs, only digital telephone signals, and mixed channels signals. The LFR obtained the lowest EER in all conditions: EER=0% for conditions with one channel type and EER=7.32% for the mixed channel condition. The spectral ratio and the modulation indexes did not obtained very good results on their own, but combined (SR, MI, Sb-MI) they were close to the LFR in two of the three channel types. The set (SR, LFR, MI, Sb-MI) was not better than the LFR for this particular dataset. However, as LFR measures the impact of the loudspeaker on the signal and the rest of features measure the effect of the far-field acquisition, we would recommend using all the features for attacks where the criminal uses high quality equipment or is able to inject the signal into the telephone line without a loudspeaker.

Figure 11.10 plots the DET curve for the mixed channel condition with the detector based on all the features. Note that the miss rate was always lower than 10% independently of the operating point that we chose for the spoofing detector. On the other hand, false alarms could grow to more than 40%.

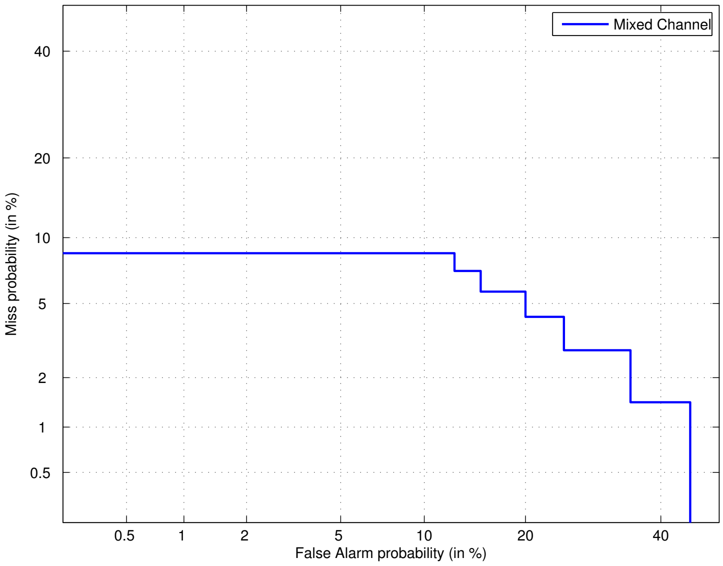


Figure 11.10: DET curve far-field spoofing detection curve for the Database 1.

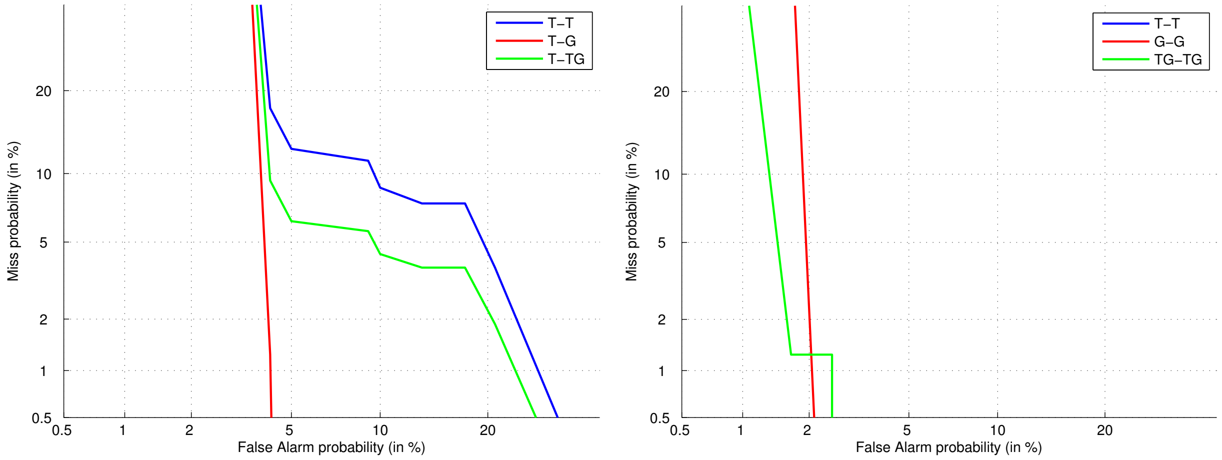
11.6.5.2 Database 2

Although Database 2 was mainly designed to detect cut and paste spoofs, we also applied the replay detector on it. Table 11.8 shows the spoofing detection EER for both parts of the database. We considered several combination of telephone channels for spoofs and non-spoofs. The nomenclature that defines each condition is: non-spoof-test-channel_spoof-test-channel. For example, T_TG means that we evaluated landline non-spoofs against mixed landline and GSM spoofs. Part 2 presented lower error rates, which means that spoofs were recorded in a way that degraded more the channel conditions compared to non-spoofs. We already noted this fact in Section 11.6.3.2 when we evaluated the SV performance in this database. For Part 1, the best result was for condition T_G, the one with more channel mismatch between spoofs and non-spoofs; and the worst was T_T, the one with less channel mismatch.

Figure 11.11 shows the spoofing detection DET curves. Except for condition T_T of the Part 2, we cannot reduce the false alarm rate as long as we want. If we want the false alarm rates to be under 5% 5% in Part 1, and under 2% in Part 2, the miss rate grows rapidly.

Table 11.8: EER(%) for far-field spoofing detection in Database 2.

		EER(%)
P_1	T_T	9.38
	T_G	2.71
	T_TG	5.62
P_2	T_T	0.00
	G_G	1.67
	TG_TG	1.46



(a) Part 1.

(b) Part 2.

Figure 11.11: DET far-field detection curves for Database 2.

11.6.6 Results fusion of cut and paste and far-field detectors

In Database 2, we had both types of spoofs simultaneously so it makes sense to fuse the spoofing detectors. We used the cut and paste detector with fusion of MFCC and pitch. Training the fusion with logistic regression did not work properly. Eventually, our fusion consisted in normalizing the scores in mean and variance and assigning the same weight to each of them. Table 11.9 shows the EER that we obtained for each channel condition and for the pool of all conditions. For Part 1, the result was worse than using only the cut and paste detector. For Part 2, we improved the two conditions that did not have zero error with the cut and paste detector.

Table 11.9: EER(%) for fusion of spoofing detectors in Database 2.

		EER(%)
P_1	T_T_T	5.00
	T_T_G	0.62
	T_T_TG	3.75
	Pool	3.12
P_2	T_T_T	0.00
	T_G_G	0.00
	T_TG_TG	0.00
	G_T_T	0.00
	G_G_G	0.00
	G_TG_TG	0.00
	TG_T_T	0.00
	TG_G_G	0.00
	TG_TG_TG	0.00
	Pool	0.00

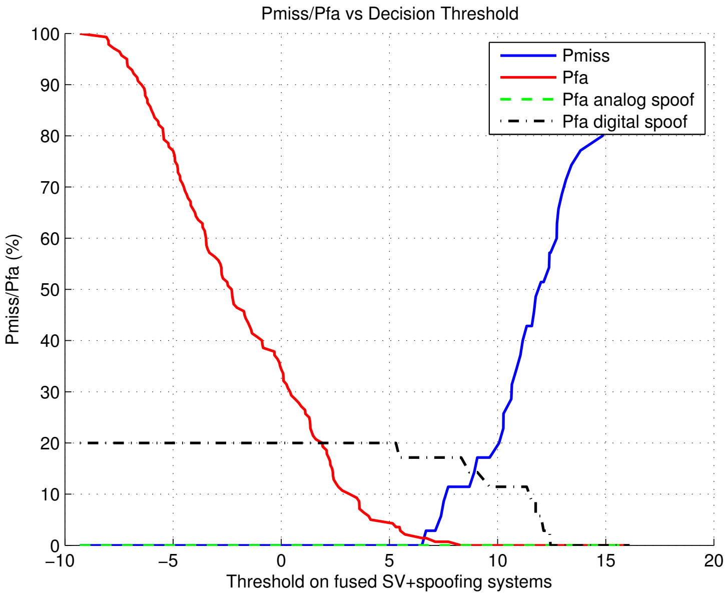


Figure 11.12: $P_{\mathrm{Miss}}/P_{\mathrm{FA}}$ against decision threshold for the fusion of SV and spoofing detection on Database 1.

11.6.7 Fusion of speaker verification and spoofing detection

Finally, we fused the spoofing detector and the SV system. We did a hard fusion where we rejected all the trials marked as spoof by the spoofing detector, i.e., we assigned them SV score equal to $-\infty$; the rest of trials kept the score provided by the SV system. We wanted the fused system to maintain the performance for non-spoofing trials close to the original SV system; and at the same time, to reduce the number of spoofing trials that deceive the system. We chose the threshold of the spoofing detector with that in mind. For not increasing the miss rate of the true target trials, we selected a high spoofing threshold.

We present results for Database 1 and for Part 1 of Database 2. For Database 1 we used the system based on far-field and replay detection; and for Database 2, we used the fusion of both detectors. Figure 11.12 refers to Database 1. It plots the miss and false acceptance rates against the decision threshold on the fused SV score. If we consider the EER operating point the percentage of accepted spoofs decreased from 68% to zero for analogs and to 17% for digitals. If we decide to work in the point with EER between misses and spoof acceptance, the miss rate is 14.3% instead of 28%, which was the value obtained without the spoofing detector.

Figure 11.13 shows results for Part 1 of Database 2. If we again consider the EER operating point the number of accepted spoofs decreases from 5% to 1.25% for landlines. Besides, all GSM spoofs were rejected no matter what SV decision threshold we choose. In exchange, we had a minimum miss probability of 1.25% because of true targets that the system marks as spoof.

11.7 Summary

In this chapter, we dealt with the problem of spoofing attacks to speaker recognition systems. Among all the types of attacks that we can find in the literature, we focused on replay

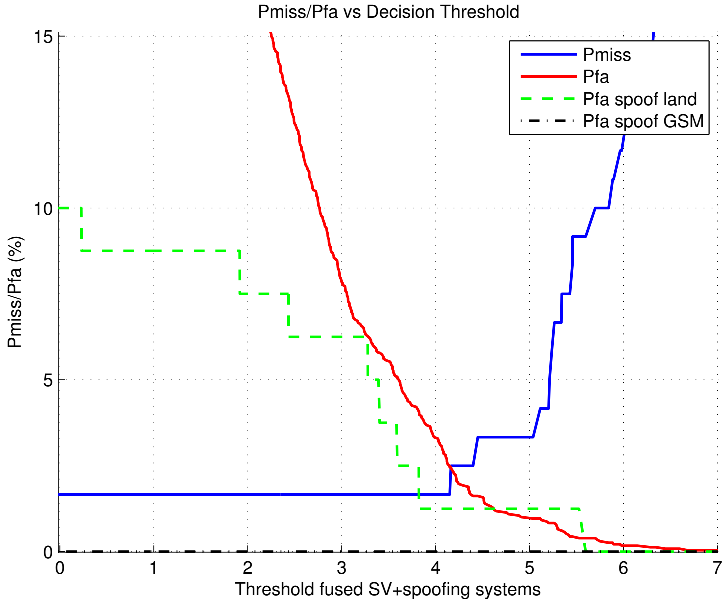


Figure 11.13: $P_{\text{Miss}}/P_{\text{FA}}$ against decision threshold for the fusion of SV and spoofing detection on Database 2 Part 1.

attacks. Replay attacks are major threads because they are low technology attacks so they are available even for criminals without speech processing knowledge. On the contrary, attacks like voice conversion or speech synthesis require higher levels of expertise.

We considered attacks to text-independent and text-dependent speaker verification systems. Text-independent and text-dependent systems with fixed pass-phrase are vulnerable to the naive attack consisting in recording the victim's voice and replaying it on the SV system. For text-dependent system with prompted pass-phrase, criminals need to carry out a more elaborate procedure consisting in extracting excerpts from different recordings and concatenating them to obtain the phrase requested by the system. We called that *cut and paste* replay attack.

We experimented on two databases. Database 1 was intended for text-independent SV. The spoofs in this database were simple replay attacks. Database 2 was intended for text-dependent SV and its spoofs included both types of attacks. On the one hand, the spoofs were created by cutting and pasting pieces of several recordings. On the other hand, the original recordings were acquired by far-field microphones and the composed signals were replayed by a loudspeaker.

In order to detect naive replay attacks, we made several assumptions. First, we assumed that the SV system was intended for a telephone application so a legitimate user should talk close to the telephone handset (close-talk recording). Second, we assume that the victim will not collaborate with the spoofer. Thus, the criminal should record the victim from a certain distance. A far-field recording has higher levels of noise and reverberation than a close-talk one. Third, we assumed that the criminal will replay the signal on the phone with a portable loudspeaker (mobile phone, tablet, etc.). This kind of loudspeakers produces a noticeable effect on the low frequencies of the signal. Summing up, our spoofing countermeasure was based on detecting far-field recordings replayed by a loudspeaker. To do it, we computed several features from the speech signal: spectral ratios and modulation indexes. They were classified by a SVM. We trained the SVM with data from NIST SRE08.

We used telephone data for non-spoofs examples and far-field microphone data for spoof examples. The microphone signals were filtered through loudspeaker and telephone channels to simulate the conditions of the real spoofs. Training with these artificial spoofs provided good error rates on real spoofs. On Database 1, the EER of spoofing detection was around 8% for mixed channel type. On Database 2, EER was between 0 and 9.38 % depending on the channel condition.

To detect cut and paste attacks, we assumed that the test pass-phrases were the same uttered during the enrollment phase. We computed distances between the enrollment and test MFCC and log-pitch feature contours. Those contours were aligned by DTW. We experimented with different telephone channels (landline and GSM) and different time distance between enrollment and test sessions. Most conditions yielded EER=0%. The worse condition, which corresponded to the case with more time distance between enrollment and non-spoof tests, obtained EER=4.17%.

We evaluated the vulnerability to spoofing attacks of a SV system based on JFA. Results evidenced very high acceptance rates of spoofs. Finally, we fused the SV score and the result of the spoofing detector significantly reducing the spoofing acceptance. Having the SV system working on the EER (for non-spoofs) operating point, spoofs acceptance reduced from 68% to 17% on Database 1; and from 5% to 1.25% on Database 2.

Spoofing attacks are one of the main barriers that we face to introduce speaker verification for security applications like telephone banking. Even thought, the research community is increasingly interested in attacking this problem, there is still a long road ahead. The wide variety of attacks that can be attempted makes spoofing detection a complex problem. Besides, the lack of publicly available databases for this task makes difficult to compare approaches.

Chapter 12

Detecting Tampering Attacks on Speaker Verification Systems

12.1 Introduction

If in the previous chapter we focused on the vulnerability of SV systems to impersonators, in this chapter, we deal with the opposite problem. Tampering attacks, also referred as voice disguise in the literature [Perrot et al., 2007], are defined as the deliberate action of a speaker who wants to modify his voice to hide his identity. This is a problem of great importance in the context of forensic speaker recognition. It has been observed that disguise happens in certain types of crime more frequently than others. Offenders usually attempt to disguise their voices in situations where they expect to be recorded or when they may be recognized by a listener who is familiar with their voice. Thus, tampering often happens in cases such as kidnapping, blackmail, threatening calls or even calls to emergency services. Disguise is more frequent in countries where these type of crimes are usual. For example, in Brazil, disguise is common in the numerous kidnapping cases. There, placing a pencil between the speaker's teeth is a frequent disguise method [Figueiredo and Britto, 1996]. Data from the German police reported in [Masthoff, 1996] show that, during the period 1989–1994, disguise happened in 52% of cases where the offender may expect to have their voice recorded. For blackmail cases, this percentage raised up to 69%.

There are many ways in which speakers can distort their voices. They can be classified into electronic and non-electronic [Rodman, 2003]. In [Masthoff, 1996], electronic disguise was reported to be relatively uncommon, occurring in only one to ten percent of voice disguise cases. However, nowadays with the development of Internet and the smartphone revolution, there is an increasing amount of available software that offers the possibility of changing somebody's voice [Scoompa, 2013, Kim, 2014, Zenital VOIP, 2014, Twiscon Software, 2013]. The main technique used by these programs, consists in modifying the pitch register by moving the mean or the pitch contour. It is also possible to add some specific effects like echo or robotic voice. On the other hand, non-electronic means include whisper, falsetto, foreign accent, change of speaking rate, imitation, pinched nostril, object in mouth, etc [Zhang and Tan, 2008]. The great variety of non-electronic disguises can be divided into four types according to the feature that we alter, as we show in Table 12.1 [Rodman, 2003]. *Phonation* refers to abnormal glottal activity; *phonemic* refers to the adoption of abnormal allophones; *prosodic* relates to intonation, segment length and speaking rate; and *deformation* involves forced physical changes in the vocal tract.

Table 12.1: Classification of non-electronic tampering methods [Rodman, 2003].

Phonation	Phonemic	Prosodic	Deformation
Raised pitch (falsetto)	Use of dialect	Intonation	Pinched nostrils
Lowered pitch	Foreign accent	Stress placement	Clenched Jaw
Creaky voice (glottal fry)	Speech defect	Segment lengthening or shortening	Use of bite blocks
Whisper	Mimicry	Speech tempo	Lip protrusion
Inspiratory	Hyper-nasal (velum lowered throughout)		Pulled cheeks
Raised or lowered larynx		Tongue holding	
			Objects in mouth
			Objects over mouth

In this chapter, we focus on some of those low technology attacks. In particular, we study disguises consisting in covering the speaker's mouth with the hand or a handkerchief; and denasalization by pinched nostril. We will show how these disguises degrade the performance of a state-of-the-art speaker verification system based on joint factor analysis. Moreover, we will present tampering detection experiments with SVM and GMM classifiers.

This chapter is organized as follows. Section 12.2 reviews previous works on voice disguise since the seventies until now. Section 12.3 describes the tampering detection systems, including features and classifiers. Section 12.4 describes our experimental setup and results. We experimented on four databases, two for each type of disguise. We evaluated the increase of misses in the SV system due to the tampering attack; performance of our tampering detectors; and the reduction of misses of tampering trials when we fuse the SV system and the tampering detector. Finally, Section 12.5 presents the conclusions of the chapter.

12.2 State of the Art

Research on voice disguise started in 1970s in the context of forensic science. In these first works, speaker identification and disguise detection tasks were carried out by just listening the speech recordings or by visual inspection of their spectrograms. Comparison of spectrograms of normal and disguised voices revealed strong variations in the formant structure [Endres, 1971]. While most early works asked speakers to freely disguise their voices in a manner that they felt that it would conceal their identities most effectively, the work in [Reich et al., 1976] studied a large variety of disguises: old–age, hoarse, hypernasal, slow-rate, and a free disguise of the speaker's own choosing. The authors proposed an open-set speaker identification task by spectrogram matching where reference signals always consisted of normal speech and test signals could consist of normal or disguised speech. The results proved that disguised speech significantly degraded performance. Later, the authors repeated the experiment performing speaker identification by listening instead of spectrogram comparison [Reich and Duke, 1979]. They achieved 92% of correct identification rate with normal speech. Rates dropped to 59–81% depending on the disguise type, results that were coherent with their previous work. The work in [McGlone et al., 1977], states that fundamental frequency, formant frequencies and bandwidths are significantly altered by disguise making speaker identification almost a matter of chance. In [Hollien and Majewski, 1977] three conditions were considered: normal speech, speech during stress and disguised speech. Speaker identification experiments by visualization of the long-term spectra found slightly reduced identification rates for speech during stress but a large degradation for disguised speech. Four years later, one of the first works on tampering detection showed that both naive and sophisticated listeners are able to recognize disguise with a high degree of reliability [Reich, 1981].

In [Shinan and Almeida, 1986], authors focused on disguises frequently found in criminal cases: heightened voice, lowered voice and pinched nose. They studied the formant transitions finding that there are regular deviations trends from one speech condition to the others and that transitions are more different between speakers than between conditions, which can be used to identify speakers. Another common disguise found in kidnapping cases consists in using a form of phonation known as *glottal fry* or *vocal creak* [Hirson and Duckworth, 1993]. This form of phonation occasionally occurs in normal speech but it is more frequent in pathological voices. Trained listeners were able to perform speaker identification with 65% accuracy on creaky voice, compared with 90% accuracy on normal speech. In that work, there were also automatic speaker identification tests made by comparing the power spectra of the [s] sound. They attained 50% accuracy when identifying one speaker out of 10, and 81% accuracy when identifying one speaker out of two. Creaky voice was also studied in [Moosmüller, 2001]. This work exposed that creaky voice is common in women that disguise their voices by lowering their pitch to imitate male's voice. The authors studied 750 creaky and modal vowels pronounced by 5 female and four male speakers. The analysis of formants evidenced that, for women, the second formant of creaky vowels is lower than the one of the same vowel and same speaker in normal mode. The effect on the male formants was inconsistent.

In [Orchard and Yarmey, 1995], whispering significantly worsened the identification performance of human listeners.

The work in [Figueiredo and Britto, 1996] considers a disguise that is common in kidnapping cases in Brazil and that consists in speaking with a pen between the front teeth. This report examined the formant shifts in Brazilian Portuguese vowels that occur with this type of disguise. The authors found that different speech segments are affected in different degrees. The most evident effect was the lowering of the high vowels.

In [Clark and Foulkes, 2007], the authors studied how electronic modifications of the pitch frequency affected speaker identification by humans. Listeners were trained to identify four voices. Most listeners performed above chance level except when the pitch frequency was reduced in 8 semitones.

The work in [Masthoff, 1996] explored the preferred forms of disguise of 20 German subjects. They asked the speakers to disguise their voice to obscure identity while retaining intelligibility. They found that the majority of the disguises included an alteration of phonation and that the subjects used one or two masking techniques simultaneously. Fifty five percent of speakers chose a single disguise method such as mimicking a foreign accent or altering their natural pitch. The remaining 45% chose multiple disguise methods. For example 15% chose a phonation change and a prosodic change; another 15% chose a phonation change and a phonemic change; and another 15% chose a prosodic change and a phonemic change. The authors noticed that when two speech characteristics are simultaneously changed, identification becomes significantly more difficult for human listeners. As only one or two phonetic parameters are changed some aspects of the vocal behavior remain undisguised and available for forensic examination. For example, speakers were unable to disguise strong regional dialect features. In [Moosmüller, 1997], it is also stated that dialect features can resist to voice disguise.

The work in [Künzel, 2000] investigated three types of disguise: raising fundamental frequency, lowering fundamental frequency, and denasalization. The author focused on the pitch frequency changes f_0. He found that there are differences between both genders with regard to the preferred disguise type as well as the articulatory alterations that they employed to create it. The results correlated with the experience in actual forensic casework showing that there is a relationship between the f_0 of a speaker normal speech and his preferred disguise. Speakers with higher than average f_0 tend to increase their f_0 level. This process may or may not involve a register change to falsetto. On the other hand, speaker with lower than average f_0 prefer to disguise their voices by lowering f_0 even more, sometimes obtaining creaky voice. This tendency is more often observed in males while females are reluctant to radically change their pitch pattern. The same dataset was used to evaluate the vulnerability to tampering of a modern speaker verification system [Künzel et al., 2004]. Until this work most literature regarding voice disguise referred to speaker identification by humans. With the popularization of automatic speaker recognition technology, the interest about how disguise affects these systems started to increase. The results indicated that, if the enrollment and test segment present the same type of disguise, voice disguise affects performance only marginally. However, if the enrollment is normal speech, significant degradation appears in the scores of 36%, 14% and 70% of the speakers for the disguises *high*, *low* and *denasalization* respectively.

Another work investigating the performance of a GMM based SV system on disguised speech is [Kajarekar et al., 2006]. In this case, speakers were instructed to try any voice modification without blocking their mouths or obstructing the path between the mouth and the microphone. Using the threshold optimized for the clean condition, the system falsely rejected 39% of the target trials with disguise in the test segment. False rejection reduced to 9% when the threshold was optimized for disguised speech. The authors compared the automatic system with the performance of human listeners. The results showed that, in general, the SV system outperforms humans when voices are disguised. In [Zhang and Tan, 2008], the authors investigate 10 types of disguises: raised and lowered pitch, fast and slow speech, whisper, pinched nostril, masking on mouth, use of bite block (pencil), objects in mouth (chewing gum) and foreign accents. Foreign accent did not affect the recognition. On the contrary, masking on mouth and whisper had the largest effect with 0% identification rate. Raised pitch was the second worst with 10% identification rate. Chewing gum, lowered pitch and pinched nostril also had an important effect with identification rates of 45%, 55% and 65%, respectively. The rest of disguises had weaker effects with identification rate above 85%. Although the effect of pinched nostril on speaker recognition was not as great as expected the voice quality was very degraded. Besides, they observed that some speakers are more easily recognized due to their idiosyncratic voice quality or poor disguising skill.

There are few works dealing with automatic disguise detection. Some serious attempts were presented in [Perrot and Chollet, 2008, Perrot et al., 2009, Chollet et al., 2012]. These works focused on detecting pinched nostril, hand over the mouth, high pitched voice and low pitched voice. A set of acoustic features including 12 MFCC with deltas, first two formants and; mean, maximum and minimum pitch frequency was used as input to SVM classifiers. Authors tried several architectures. The one yielding better results was a parallel architecture with five SVM classifiers, one for each one of the four types of disguises and another to discriminate *disguised/normal*. The output of the five classifiers is fused to obtain the final decision. For the pool of all the disguise types, EER around 20% was obtained.

In [Perrot et al., 2009], the authors evidenced that the performance of these classifiers highly degrades when disguised signals present additive noise.

In this chapter, we focus on two types of tampering methods: hand or handkerchief over the mouth; and denasalization by pinched nostrils. We evidence the performance degradation of a state-of-the-art SV system under these attacks and evaluate several classifiers for the task of automatic tampering detection.

12.3 Tampering Detection Systems

12.3.1 Features

In our experiments, the best performing features were the MFCC. MFCC were described in Section 2.3.1. We removed silence frames with our LTSD based VAD [Ramirez et al., 2004]. We did not apply any normalization like CMS or CMVN to the MFCC to not reduce the disguise effect on the features.

We also experimented with a wide range of other features however they did not outperformed the MFCC, neither on their own nor fused with the MFCC. Because of this, we will not present tampering detection results with these features. Among the features that we experimented with, we find spectral ratios like the ones that we used to detect replay attacks, described in Section 11.5.1. We also evaluated glottal properties, which were proposed in [Stevens and Hanson, 1994] to assess speech quality. They were open quotient, glottal opening, skewness of glottal pulse, rate of glottal closure and incompleteness of the glottal closure. They were computed by gradients between the amplitudes of different harmonics and that of the fundamental frequency as described in [Lugger and Yang, 2006]. Finally, we tried the quality measures proposed in [Monzo et al., 2007, Monzo et al., 2008]: spectral flatness, Hammarberg index, drop-off of spectral energy above 1 kHz, jitter and shimmer.

12.3.2 SVM classifiers

We used SVM and GMM classifiers. We tried two configurations for the training and evaluation of the SVM. In the first one, we applied the SVM on the average of the feature vectors of each speech file. Thus, the SVM evaluation function was

$$f(\mathbf{X}) = \sum_i \alpha_i \mathrm{k}(\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t, \mathbf{x}_i^*) + b \tag{12.1}$$

where k is the kernel function, $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ is the set of feature vectors in the test file, T is number of frames in $\mathbf{X}$, $\mathbf{x}_t$ is the t^{th} test feature; and $\mathbf{x}_i^*$, α_i and b are the support vectors, the support vector weights, and the bias parameter that are estimated during the SVM training. We tried linear, polynomial and Gaussian kernels. The Gaussian kernel performed the best:

$$\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\gamma \left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2\right) \ . \tag{12.2}$$

In the second configuration, we trained the SVM on all the feature vectors instead of doing it on the average. Then, the SVM was evaluated in a *frame by frame* fashion and the scores of each frame were averaged:

$$f(\mathbf{X}) = \frac{1}{T}\sum_{t=1}^{T}\sum_{i}\alpha_i \mathrm{k}(\mathbf{x}_t, \mathbf{x}_i^*) + b \ . \tag{12.3}$$

We also evaluated two configurations regarding the way to organize the SVM training data. In the first one, the training and evaluation data were from the same database. Given the small size of our datasets, we could not divide them into training and test sets. To avoid over-fitting we used a leave-one-out procedure. To be more precise, it was a leave-one-speaker-out procedure. That is, we trained a SVM per speaker with positive and negative samples from the rest of speakers. Then, we evaluate the SVM only for the segments of the corresponding speaker. Finally, the tampering detection scores from all the speakers were pooled together to compute error rates and plot DET curves.

In the second training configuration, we took advantage that we had two different databases for each type of tampering (covering mouth and denasalization). Although the SV results indicated that databases with the same type of tampering performed very different, we tried to train on one database and test on the other.

12.3.3 GMM classifier

We also tried a GMM based classifier. We trained a GMM for normal speech and another for disguised speech by maximum likelihood EM iterations [Bishop, 2006]. Test segments were evaluated by the log-likelihood ratio between both models:

$$\mathrm{LLR} = \frac{1}{T}\sum_{t=1}^{T}\ln P\left(\mathbf{x}_t|\mathrm{Normal}\right) - \ln P\left(\mathbf{x}_t|\mathrm{Tamper}\right) \ . \tag{12.4}$$

We used the same training configurations as the SVM: training and test on the same database with the leave-one-speaker-out procedure and training and test on different databases.

12.4 Experiments

12.4.1 Databases

At the moment of this work, there were no publicly available databases for this task. In our experiments, we used datasets provided by a third party. We focused on two types of tampering: covering the mouth and denasalization by pinched nostrils. We had two different databases for each one of them.

12.4.1.1 Covering Mouth Database 1

The disguise in this dataset consisted in covering the speaker's mouth with a handkerchief together with the hand making a shell. This type of disguise strongly distorted the spectral features that are used by current SV systems. This database consisted of 10 speakers with 3 sessions:

- Session 1: used for speaker enrollment. There were around 12 seconds of speech per speaker.

- Session 2: normal test signals. There were 120 short segments around 3 seconds long, 12 different phrases per speaker.

- Session 3: tampering test signals. There were another 120 short segments around 3 seconds long. Speakers repeated the same phrases as in session 2.

Signals were recorded over a landline telephone channel. With this database, we evaluated 120 normal target trials, 120 tampering target trials and 1080 normal non-target trials.

12.4.1.2 Covering Mouth Database 2

In this dataset, disguises were created by making a tube with the hands. The dataset consisted of 21 speakers with 3 sessions: enrollment, normal tests and tampering tests. Signals were recorded over landline and mobile channels. Speakers repeated the same sentences in the sessions 2 and 3, which were the same as in Database 1. With this database, we evaluated 252 normal target trials, 252 tampering target trials and 5040 normal non-target trials.

12.4.1.3 Denasalization Database 1

Denasalization consists in pinching the nostrils while the user is speaking. In this way, the sound wave is reflected back along the nasal cavity interfering with the wave in the pharynx. At certain frequencies both waves cancel each other introducing anti-resonances in the transfer function of the vocal tract. As covering the mouth, the spectral properties of the recorded signals change and damage the SV performance. The database consisted of 52 speakers. It included read and spontaneous speech recorded over a GSM channel. There were speech segments 60, 90 and 120 seconds long. Normal segments of 120 seconds were used for enrollment and the rest for test. We evaluated 198 normal targets trials, 165 tampering target trials and 10098 normal non-target trials.

12.4.1.4 Denasalization Database 2

The second denasalization dataset consisted of 10 speakers with three sessions:

- Session 1: used for enrollment. It was a text 30 seconds long.

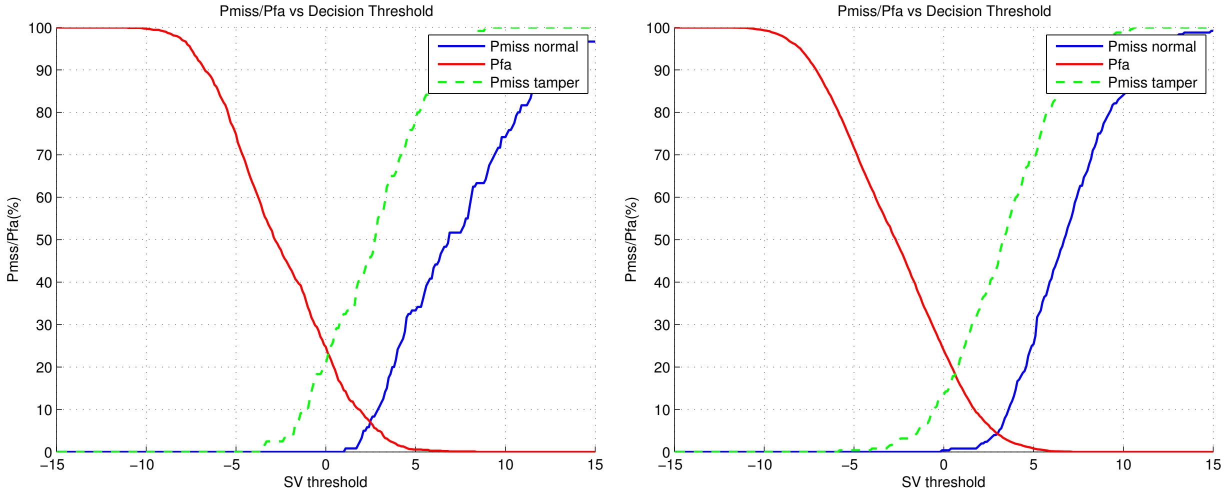
- Session 2: used for normal tests. There were 4 recordings of 4 different sentences.

- Session 3: used for denasalized tests. There were 4 recordings of the same 4 sentences used in session 2.

We evaluated 160 normal target trials, 1440 normal non-target trials and 160 tampering target trials.

12.4.2 Speaker verification performance degradation

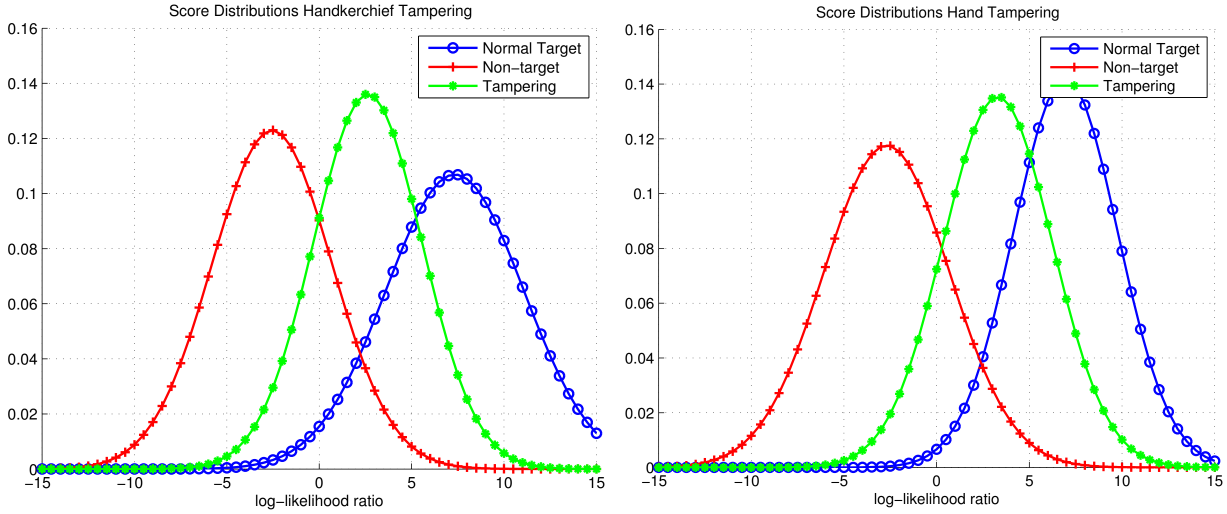
We evaluated how tampering increases the misses in our JFA SV system, which we described in Section 11.6.3 of the previous chapter.



(a) Database 1.

(b) Database 2.

Figure 12.1: $P_{\text{Miss}}/P_{\text{FA}}$ vs decision threshold for databases with tampering by covering mouth.



(a) Database 1.

(b) Database 2.

Figure 12.2: SV score distribution for databases with tampering by covering mouth.

Table 12.2: SV score reduction by covering mouth.

ΔLLR	Mean	Std	Median	Max	Min
Database 1	4.71	3.65	4.20	17.02	-1.95
Database 2	3.65	3.09	3.42	13.59	-5.31

12.4.2.1 Covering mouth

First, we focus on the datasets with disguise by covering the mouth. For non-tampering trials, Database 1 obtained EER=6.57%, and Database 2 EER=4.25%. Figure 12.1 displays the misses and false acceptances against the SV threshold for both datasets. If we place the operating point of the SV system in the threshold where $P_{\text{Miss}-\text{normal}} = P_{\text{FA}}$, we obtain a large amount of misses for the tampering target trials. For Database 1, we measured $P_{\text{Miss}-\text{tamper}} = 45.80\%$ and, for Database 2, $P_{\text{Miss}-\text{tamper}} = 44.05\%$. That means that miss rates multiply by 6 and 9 respectively. To reduce the misses of tampering trials we need to considerably lower the SV threshold in exchange of increasing the number false acceptances. For example, in the point with equal error rates between tampering misses and false acceptances, we obtain $P_{\text{Miss}-\text{tamper}} = P_{\text{FA}} = 22.5\%$ for Database 1 and $P_{\text{Miss}-\text{tamper}} = P_{\text{FA}} = 19.05\%$ for Database 2.

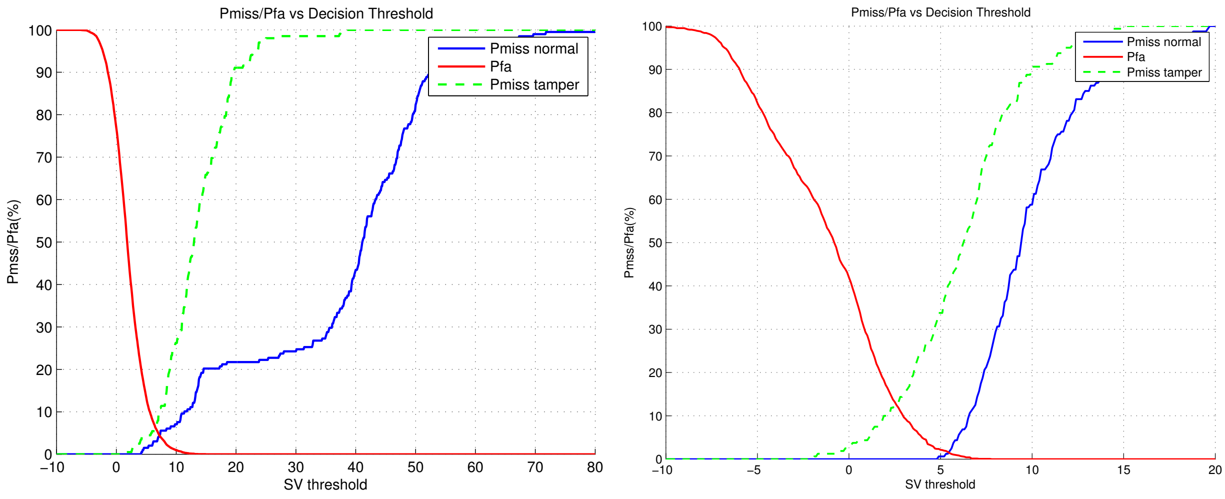
Figure 12.2 shows the score distributions for the different trial sets. The tampering trial distributions are approximately in the middle, between the normal targets and the non-targets. There is an important overlap between non-targets and tampering trials that explains the increase of misses that we saw above.

In Table 12.2, we present some statistics for the reduction of score ΔLLR between normal and tampering targets. Each ΔLLR was computed between segments of the same speaker uttering the same phrase. The table evidences that the disguises had different degree of success. For some trials the tampering did not succeed, obtaining larger score in the disguised version than in the normal one. On the other hand, some disguises achieved huge score drops.

12.4.2.2 Denasalization

The datasets with denasalization tampering behaved differently. For non-tampering trials, Database 1 obtained EER=4.09% and Database 2 EER=1.45%. Figure 12.3 shows misses and false acceptance rates against the SV threshold, and Figure 12.4 shows the score distributions for the different groups of trials. Table 12.3 shows statistics for the difference between the normal targets and tampering scores ΔLLR. For Database 2, ΔLLR was computed as in the databases with covering mouth disguise; that is, as the difference between trials with the same speaker and phrase. For Database 1, as speakers say different phrases in tampering than in normal segments, a ΔLLR was computed for each speaker by subtracting the average scores of his normal and tampering trials.

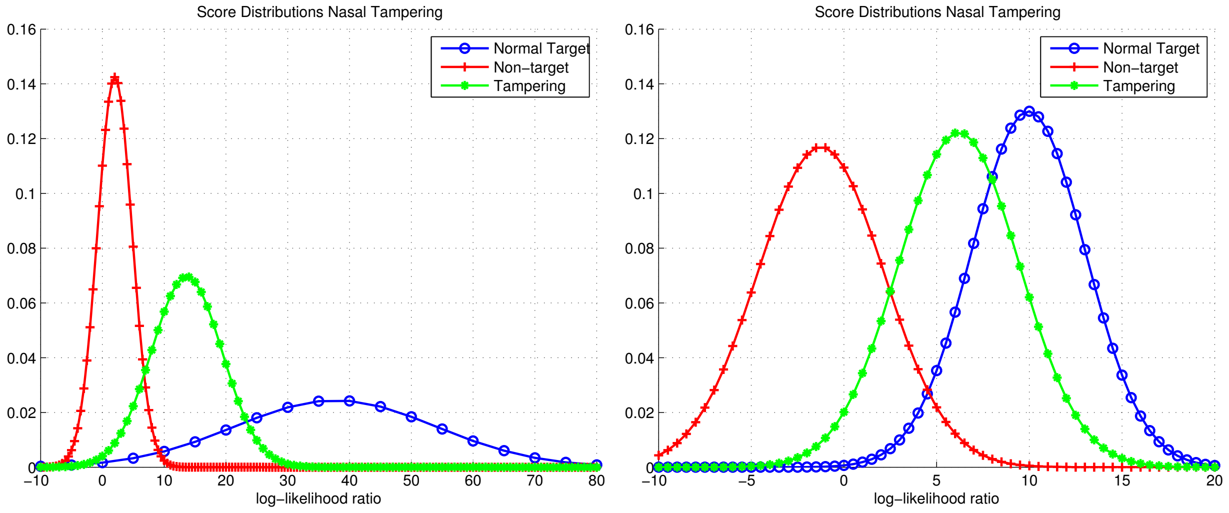
Database 1 presented very high target scores and, although tampering greatly dropped the scores, they remained larger than the tampering scores in the other databases. If we work in the EER operating point, we obtain $P_{\text{Miss}-\text{tamper}} = 10.89\%$ that is much lower than the misses in both covering mouth datasets. For Database 2, the resulting score distributions are much similar to those of covering mouth disguise. Again, working in the EER operating point, tampering increased the miss rate to $P_{\text{Miss}-\text{tamper}} = 40\%$–the miss rate multiply by 26. To reduce the miss rates, we can work in the point with equal tampering misses and false acceptances rates. Then, we obtain $P_{\text{Miss}-\text{tamper}} = P_{\text{FA}} = 6.43\%$ for Database 1, and $P_{\text{Miss}-\text{tamper}} = P_{\text{FA}} = 12.50\%$ for Database 2. Differences between both databases were mainly due to the longer duration of the segments in Database 1, which allows better inter-session compensation.



(a) Database 1.

(b) Database 2.

Figure 12.3: $P_{\mathrm{Miss}}/P_{\mathrm{FA}}$ vs decision threshold for databases with denasalization.



(a) Database 1.

(b) Database 2.

Figure 12.4: SV score distribution for databases with denasalization.

Table 12.3: SV score reduction by denasalization.

ΔLLR	Mean	Std	Median	Max	Min
Database 1	26.51	11.37	26.62	53.37	3.94
Database 2	3.75	2.76	3.27	12.24	-1.25

Table 12.4: EER(%) Covering mouth detector.

Features	Train leave-one-out						Train on Database 1		
	Database 1			Database 2			Database 2		
	SVM	SVM FbyF	GMM	SVM	SVM FbyF	GMM	SVM	SVM FbyF	GMM
C1-4	**2.22**	2.61	1.87	**16.13**	17.85	17.57	31.56	31.05	24.14
C1-6	3.88	0.71	**0.55**	19.20	16.54	16.32	36.41	33.53	**23.11**
C1-12	3.83	**0.00**	0.69	19.68	**14.73**	**15.16**	**31.31**	**24.83**	23.14

12.4.3 Results covering mouth detection

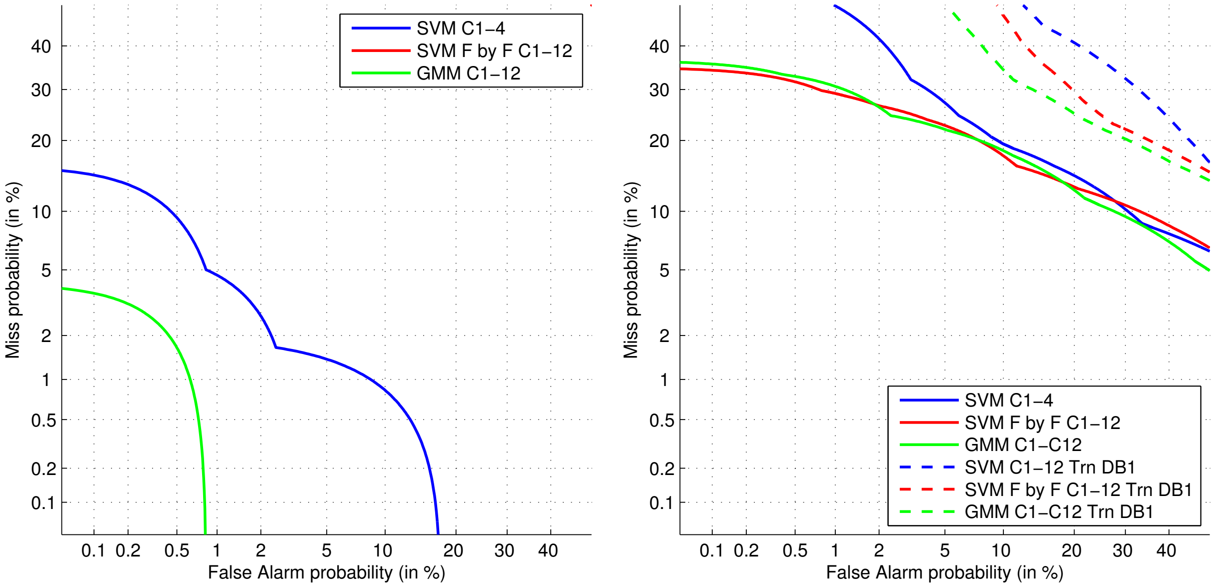
We applied the classifiers described in Section 12.3 to detect disguise by covering mouth. Table 12.4 shows tampering detection EER for both databases. Note that they are not SV error rates but tampering detection ones. We tried different MFCC dimension and three classifiers: SVM on the average feature of the file (SVM), SVM trained and tested in a frame by frame fashion (SVM FbyF), and GMM. The group of columns on the left refers to the case where classifiers are trained and evaluated with a leave-one-out procedure and the group on the right refers to the case were the classifier is trained on Database 1 and evaluated on Database 2. We tried different number of components for the GMM obtaining the best results with 64 Gaussians.

When training with leave-one-out, the SVM frame by frame attained the lowest error rates followed very closely by the GMM. Although in the previous section, we showed that covering with a handkerchief (Database 1) or only with the hand (Database 2) produced similar increments of the miss rate, here, we observe that the latter is much more difficult to detect. While for Database 1, we obtained perfect classification, for Database 2 the best EER was as high as 14.73%. When training on Database 1, the EER in Database 2 was even higher. In this case, the GMM produced the best result with EER=23.11%. The GMM generalized better between datasets while the SVM seemed to over-fit attaining higher error rates.

Figure 12.5 shows tampering DET curves for each classifier. We only plot the curve corresponding to the optimum number of MFCCs for each classifier. Regarding Database 1, the curve of the SVM frame by frame does not appear because the error is zero. The GMM also yielded quite good results; misses are always under 5% and false alarms under 1%. Regarding Database 2 with the leave-one-out setup, although in terms of EER the SVM frame by frame was slightly better than the GMM, the curve reveals that in some operating points the GMM is better and the differences between both curves are not significant. GMM and SVM frame by frame attained 0% false alarms while detecting more than 60% of the tampering attempts. The SVM on the average MFCC is clearly worse than the others in the low false alarm region of the curve. Regarding the classifiers trained on Database 1 and tested on Database 2, the GMM was better in all the operating points and, more significantly, in the low false alarm region.

12.4.4 Results denasalization detection

Table 12.5 shows tampering detection EER for both datasets with denasalization disguise. Again, we present results for our three classifiers with the two training modes (leave-one-out,



(a) Database 1.

(b) Database 2.

Figure 12.5: DET curves for detection of tampering by covering mouth.

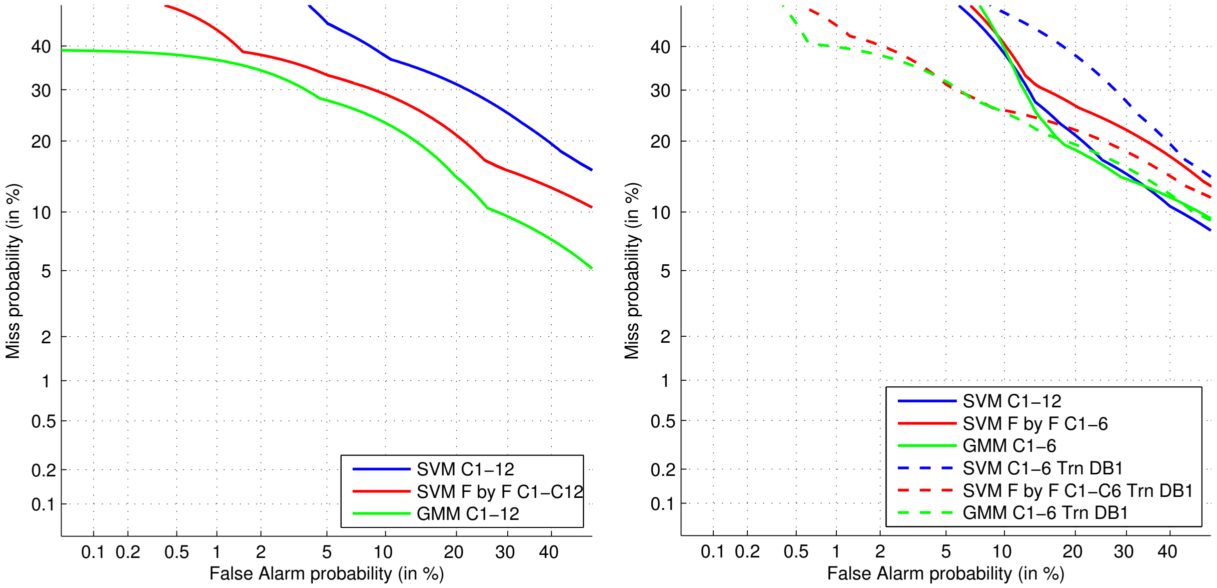
train on Database 1). The best number of components for the GMM was 64.

The best classifier was the GMM in all cases, in Database 1 with MFCC C1-12 and, in Database 2 with C1-C6. In Database 2, there was not much difference between training the GMM with leave-one-out or on Database 1. Both datasets presented high error rates. The best results were EER=16.98% for Database 1; and EER=18.97% for Database 2.

Figure 12.6 shows tampering DET curves for each classifier. We only plot the curve corresponding to the optimum MFCC dimension for each classifier. For Database 1, the GMM classifier was significantly better in all operating points. The largest difference happened for false alarms under 1%. The miss rate for very low false alarms was always under 40%. Regarding Database 2, for false alarms larger than 20%, the GMM trained with leave-one-out performed similarly to the one trained on Database 1. For false alarms under 20%, the GMM and SVM frame by frame trained on Database 1 were better. GMM and SVM frame by frame obtained very close curves but the GMM was better in most of the operating points.

Table 12.5: EER(%) Denasalization detector.

Features	Train leave-one-out						Train on Database 1		
	Database 1			Database 2			Database 2		
	SVM	SVM FbyF	GMM	SVM	SVM FbyF	GMM	SVM	SVM FbyF	GMM
C1-4	22.88	23.53	22.19	30.79	25.68	24.40	29.37	25.73	25.01
C1-6	22.82	20.92	19.06	25.74	**24.49**	**18.97**	**28.80**	**21.45**	**19.57**
C1-12	**21.25**	**20.56**	**16.98**	**19.43**	25.73	27.06	34.84	26.09	25.17



(a) Database 1.

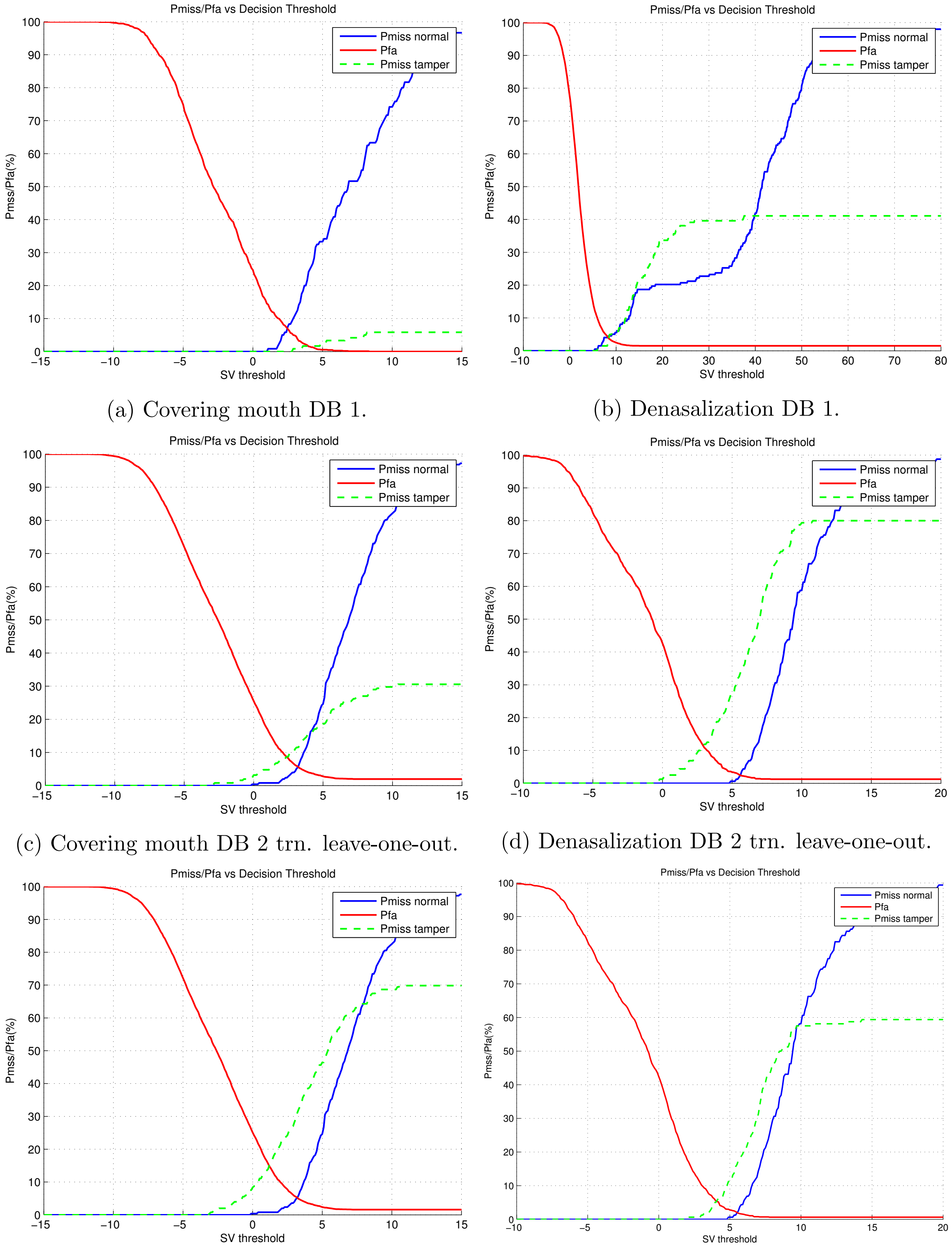
(b) Database 2.

Figure 12.6: DET curves for detection of tampering by denasalization.

12.4.5 Fusion of speaker verification and tampering detection

Finally, we fused the SV system and the tampering detector to improve the overall performance. We used the tampering detectors based on GMM given that they performed better than the SVM in most cases. We did a hard fusion were we marked as target all the trials that were classified as disguised, i.e., we assigned then SV score equal to $+\infty$. We are aware that this strategy may not make sense in many applications because that fact that the voice is disguised does not imply that it corresponds to the person that we are looking for. However, it could be useful for a surveillance application where we search for criminals in telephone calls. If we detect that somebody is disguising his voice, he may be a criminal and the call deserves further investigation. We wanted the tampering detector not to increase the false alarm rate of non-disguised non-targets. Therefore, we chose a high threshold for the score of the tampering detector.

Figure 12.7 displays the new curves $P_{\text{Miss}}/P_{\text{FA}}$ against SV threshold that we obtained with the fusion. The left column corresponds to the databases with covering mouth disguise and the right columns to the database with denasalization. We can compare this figure with Figures 12.1 and 12.3. We attained an evident reduction of the number of misses in Subfigures 12.7a, 12.7c and 12.7f. Table 12.6 compares the performance in the EER operating point of the SV with and without tampering detection. It shows the EER, the P_{Miss} of disguised targets and their relative improvement. $P_{\text{Miss-tamper}}$ improved by more than 50% in four of the six conditions while EER worsened by less than 40%. The best results were for the Databases 1 of both types of disguise were $P_{\text{Miss-tamper}}$ improved by more than 75% while EER worsened by less than 5%.



(a) Covering mouth DB 1.

(b) Denasalization DB 1.

(c) Covering mouth DB 2 trn. leave-one-out.

(d) Denasalization DB 2 trn. leave-one-out.

(e) Covering mouth DB 2 trn. on Database 1.

(f) Denasalization DB 2 trn. on Database 1.

Figure 12.7: $P_{\text{Miss}}/P_{\text{FA}}$ against decision threshold for the fusion of SV and tampering detection.

Table 12.6: EER(%) and $P_{\text{Miss}-\text{tamper}}$(%) in the EER operating point for the fusion of SV and tampering detection.

	Train leave-one-out				Train on Database 1	
	Database 1		Database 2		Database 2	
	EER	$P_{\text{Miss}-\text{tamper}}$	EER	$P_{\text{Miss}-\text{tamper}}$	EER	$P_{\text{Miss}-\text{tamper}}$
Covering mouth						
SV	6.57	44.80	4.25	44.05	4.25	44.05
SV + tampering	6.85	0.00	5.69	11.51	5.27	30.95
Rel. improvement	-4.26	100.00	-33.88	73.87	-24.00	29.73
Denasalization						
SV	4.09	10.89	1.45	40.00	1.45	40.00
SV + tampering	4.17	2.47	2.43	33.75	2.01	16.25
Rel. improvement	-1.95	77.31	-67.58	15.62	-38.62	59.37

12.5 Summary

In this chapter, we approached the problem of detecting voice disguised to avoid speaker recognition. We focused on two disguise methods: covering the mouth with the hand or a handkerchief; and denasalization by pinching the nostrils. We chose these methods because they do not require any technical knowledge so they can be carried out by any type of criminal.

We applied different features and classifiers to the task of tampering detection. MFCC were the features that performed the best. Regarding the classifiers, GMM and SVM evaluated in frame by frame fashion performed similarly in most of the conditions but the GMM seemed to be more robust to over-fitting.

We experimented on four datasets, two for each type of disguise. We tried two modes of training the classifiers: training and evaluation on the same database by using a leave-one-out procedure; and training on one of the datasets and evaluating on the other.

Using a JFA SV system, we showed that, disguised target trials present miss rates much higher than normal trials. For example, for a system working in the EER operating point, we observed $P_{\text{Miss}-\text{tamper}} \sim 40\%$ while $P_{\text{Miss}-\text{normal}}$ was always under 7%. Regarding the disguise detection experiments, we obtained different results depending on the dataset. For the database with disguise by covering the mouth with a handkerchief, tampering detection EER was as low as 0.55%. The disguise by covering the mouth with the hand was more difficult to detect with EER=15–23%. For the denasalization datasets, the EER was also quite high being between 17 and 20%.

Despite of the high error rates of the tampering detectors, the fusion of the SV system with the tampering detector attained a significant improvement. The fusion consisted in marking as targets all the trials that were classified as disguised. We are aware that the fact that the voice is disguise does not means that it corresponds to the target speaker. However, this strategy could make sense in some security applications. The threshold on the score of the tampering detector was set high for not increasing the low false alarm rates. In most conditions, the miss rate of tampering trials improved by more than 50% with small increase of the error rates of the normal trials.

Tampering is a major thread to speaker verification in the field of law enforcement. As it happens for spoofing attacks, there are still few works dealing with this problem. The wide variety of disguise methods that a criminal can use makes this problem a tough one. Besides, the lack of publicly available databases complicates the comparison of approaches and collaboration between institutions.

Part V

Conclusions

Chapter 13

Conclusions and Future Lines

13.1 Conclusions

In the last years, NIST evaluations have driven most speaker recognition research. Given the characteristics of NIST datasets, researchers had developed effective methods to characterize speakers and compensate speaker variability between different sessions. However, NIST presents an ideal scenario with relatively clean speech, collaborative users and sufficient data to train probabilistic models. However when applying speaker verification in real environments, we face some challenges that deserve further research. This thesis dealt with some of them. First, we worked on estimating the reliability of the speaker verification decisions. Second, we focused on the i-vector approach and the difficulties of modeling i-vector distributions when having recordings acquired in different conditions or when the training data is limited. In the last part of the thesis, we were interested in attacks to speaker recognition systems. We considered spoofing and tampering attacks. Both attacks have opposite intentions, while spoofing intends a impostor to impersonate a target speaker, tampering aims to conceal the speaker's identity. Following, we present the conclusions of each part of the thesis.

13.1.1 Quality Measures and Reliability

Speaker verification performance can decrease due to multiple causes: noise and reverberation in the speech signals, test languages and transmission channels different from those in the data used to train JFA and PLDA models, etc. Then, SV decisions become unreliable and we should not dare to assert whether the trial is target or non-target. This fact motivated us to work on reliability estimation. We applied Bayesian networks to model the causal relationships between the trial reliability, the speaker verification score and a set of quality measures computed from the enrollment and test utterances of the trial. The trials with low reliability are rejected. Thus, we can assure that the rest of trials present low error rates.

In Chapter 4, we described our quality measures in detail. These were selected because they carry information about noise and reverberation levels or channel type. Some measures had been used in previous works: signal-to-noise ratio, spectral entropy, number of speech frames, log-likelihood of the MFCC given the UBM, etc. Others are novel contributions of this thesis. Among them, we should point out the use of VTS parameters as quality measures. The VTS approach is well known in speech recognition in noise [Li et al., 2009].

VTS approximates linearly the non-linear effect of noise and reverberation on the MFCC by applying vector Taylor series. This linear function facilitates computing the means and variances of the noisy space GMM from the clean space GMM. The Taylor series coefficients depend on the mean and variances of the noise and channel in cepstral domain, which can be estimated by EM iterations. We stacked the noise and channel means and reduced dimensionality by linear discriminant analysis intending to create a measure sensible to noise and reverberation.

In Chapter 6, we introduced a novel Bayesian network whose purpose was to model how SV score distributions change when trial segments are distorted. This BN hypothesizes the existence of two scores: one observed and another hidden. The observed score or *noisy score* is the one given by the SV system while the hidden score or *clean score* is the ideal score that we would obtain if we had high quality speech. The network has another hidden variable, the *quality state* that means the type of distortion. Each value of the quality estate is associated with a distribution of quality measures and with a distribution for the difference between the clean and noisy scores. The network allows us to compute the posterior distributions for the hidden variables given the observed variables. The reliability posterior is the probability that the *clean score* is over or under the threshold, what is obtained by integrating the posterior distribution of the *clean score*. The parameters of the network can be estimated with the EM algorithm. We proved that it works even with three hidden variables involved, i.e., trial label, quality states and clean score.

We experimented on NIST SRE augmented with noise and reverberation, and two datasets with real distortions. The networks were trained on NIST SRE08 and tested on the rest of databases. We compared the proposed BN with previous models, described in Chapter 5. We compared the approaches by curves plotting actual DCF against the percentage of rejected trials. The best reliability estimator is the one that attains the lowest DCF while rejecting the lowest number of trials, as explained in Chapter 3. The new BN outperformed the baseline for trial added noise and real distortions. For reverberation, the results were alike. The best quality measures where the VTS parameters and the UBM log-likelihood.

In Chapter 3, we defined an extended DCF that add terms that penalize rejecting well classified trials. This extended DCF help to establish the operating point of the reliability estimator. For example, on NIST with noise and reverberation added, we selected and operating point where we rejected 40% of trials, EER improved by 22% and actual DCF by 83%.

We can also apply the proposed BN to compute an improved SV likelihood ratio. Thus, we can use the network in applications where we require classifying all the trials. The results evidenced that the improved ratio was better calibrated.

Signal distortions can be different in each dataset. Hence, in Chapter 7 we addressed the problem of adapting the BN from one domain with a large amount of training data to another one with scarce data. We proposed to employ *Maximum a posteriori* adaptation. We obtained good results if, besides of adapting the network, we re-calibrated the scores for the target database.

13.1.2 PLDA for Non-Collaborative Environments

In Chapter 2, we reviewed the evolution of speaker recognition technology in the last years. At the end of the chapter, we compared the performance of state-of-the-art system on a common datasets. Although JFA slightly outperformed i-vectors in the telephone–telephone conditions, i-vectors were better in conditions involving far-field microphones. For that reason, in Part III we focused on the i-vector paradigm and, more specifically, on PLDA as a mean of modeling the i-vector distributions. We addressed several issues concerning i-vector modeling.

In Chapter 8, we considered the problem of simultaneously having i-vectors recorded in different conditions like different channel types, noise types or noise levels. Intending to approach the problem in a principled way, we introduced a PLDA variant, that we called multi-channel SPLDA (MCSPLDA), where the speaker space distribution was common to all types of channels and the channel space distribution was channel dependent. We compared this model with a standard SPLDA just trained on pooled clean and noisy data. We experimented on the NIST SRE12, which included artificially added noises. MCSPLDA and SPLDA attained similar results. We concluded that training an unique within-class covariances with all the available data can be more robust than training one covariance per channel type. No gain was for noises not included in training with respect to training the PLDA with only clean data.

In theory, to properly estimate the parameters of the PLDA model we need a number of speakers much larger than the i-vector dimension as well as several recordings per speaker. However, we usually count with much less speakers. This implies a certain uncertainty about the values of the model parameters. Standard training methods like maximum likelihood make point estimates of the parameters ignoring that uncertainty. In Chapter 9, we proposed to compute a posterior distribution for the model parameters given the development data and use it to evaluate fully Bayesian likelihood ratios by integrating out the model parameters. Using model posteriors instead of point estimates, uncertainty is taken into account. As the integrals involved are intractable, we developed and approximated procedure that uses variational inference to compute posterior distributions. We compared the Bayesian approach with i-vector length normalization (intended to reduce mismatch between development and trial datasets). The improvement of both techniques was comparable. We obtained small gains by combining both approaches. The computational cost of the Bayesian approach is much larger than the one of length normalization, so it is only worth in some applications.

We also considered the problem of training PLDA for applications where the development data is scarce. In Chapter 10, we proposed to do MAP adaptation of PLDA from a domain with sufficient development data (out-of-domain) to the target domain. We used the same variational Bayes procedure developed for Chapter 9. In our experiments, performance improved by 15–40% with respect to the out-of-domain model. The main improvement came from the adaptation of the within-class covariance.

13.1.3 Spoofing and Tampering

The last part of the thesis was dedicated to spoofing and tampering attacks to speaker recognition systems. We focused on low effort attacks that criminals could perpetrate without needing any speech processing knowledge.

Chapter 11 dealt with replay attacks. Text-independent and text-dependent systems with fixed pass-phrase are vulnerable to the naive attack consisting in recording the victim's voice and replaying it on the SV system. For text-dependent systems with prompted pass-phrase, we need to concatenate excerpts from different recordings (cut and paste). To detect naive replay attacks, we assumed that the criminal should record the victim from a certain distance and replay the signal on the phone with a portable loudspeaker. Thus, the detector was based on a set of acoustic features and an SVM trained to detect far-field and loudspeaker channels. To detect cut and paste attacks, we computed distances between the enrollment and test MFCC and log-pitch feature contours. We showed that spoofing dramatically increases false acceptance of state-of-the-art SV systems. By fusing the output of spoofing and speaker detectors we reduced false alarms.

In Chapter 12, we worked on detecting voice disguised to avoid speaker detection. We focused on two disguise methods: covering the mouth with the hand or a handkerchief; and denasalization by pinching the nostrils. Disguised target trials presented miss rates much higher than normal trials. The tampering detector consisted of MFCC features and; SVM or GMM classifiers. Covering the mouth with a handkerchief had tampering detection EER was as low as 0.55%. Meanwhile, covering the mouth with the hand and denasalization had a much higher EER between 15 and 23%. Disguised target trials presented miss rates much higher than normal trials but using the tampering detector we attained a significant improvement.

Spoofing attacks are one of the main barriers that we face to introduce speaker verification for security applications like telephone banking. On the other hand, tampering is a major threat to speaker verification in the field of law enforcement. Even though, the research community is increasingly interested in addressing these problems, there is still a long road ahead. The wide variety of attacks that can be attempted makes this problem a complex one. Besides, the lack of publicly available databases complicates the comparison of approaches and collaboration between institutions.

13.2 Contributions of the Thesis

Following, we enumerate the main contributions, to our view, of this thesis. In the part considering quality measures and reliability, those contributions where:

- **Bayesian network to model the variability of the SV score in adverse environments:** we defined a BN that relates speech quality measures with the distribution of the SV score. The network allows us to compute a posterior distribution for the hypothetical score that we would obtain if we would have high quality speech. Applied to the task of reliability estimation, we showed that it outperformed previous works.

- **VTS parameters as quality measures:** vector Taylor series is a well known approximation used in speech recognition for computing the mean and variance of the noisy GMM from a clean GMM. One of the steps of this approach consists in computing the noise and channel means in cepstral domain by EM iteration. We introduced a novel quality measure based on stacking those noise and channel means and reducing their dimension by LDA. This measure performed well in our reliability experiments.

- **SNR estimation with comb filters:** we computed signal-to-noise ratios taking advantage of that the energy of voiced speech is mainly concentrated in multiples of its pitch frequency while the noise frequency distribution is more uniform. We used two complementary comb filters to estimate speech and noise powers. The ratio between those powers was calibrated to obtain the SNR. With this method, we can compute the SNR in signals without silence intervals, which we usually need to estimate the noise power.

- **Saturation detector:** we presented an algorithm to detect saturated frames in cases where the saturation level is unknown. This can happen if the signal suffers amplitude changes in the transmission channel after saturation. We used an heuristic algorithm combining three measures: repetition of local maxima, deviation of the speech distribution from the Laplace distribution, and appearance of harmonics in high frequencies.

The main contributions regarding i-vector modeling with PLDA were:

- **Multichannel PLDA:** we defined a mixture of PLDA models with eigen-voice matrix and speaker factors tied across the components of the mixture and different channel covariances for each component. This PLDA intended to model i-vectors from different sources in a principled way.

- **Bayesian two-covariance model:** The fully Bayesian two-covariance model or full-rank PLDA, assumed that the model parameters are hidden variables with prior and posterior distributions instead of point estimates. As the model posteriors could not be obtained in close form, we approximated them by variational Bayes. We used this model for two purposes:

  - **Evaluation of fully Bayesian likelihood ratios:** the Bayesian ratio integrates out the model parameters based on the model posterior. In this manner, model uncertainty is taken into account. This method greatly improved performance for non length-normalized i-vectors.
  - **MAP adaptation of the model:** adapting a PLDA model from one domain to another where the development data is scarce. Adapting the within-class covariance significantly improved the results.

Finally, the contributions related to attacks to SV systems were:

- **Naive replay attacks detector:** we assumed that replay attacks would be recorded by a far-field microphone and replayed on the telephone handset by a loudspeaker. We trained a SVM to distinguish between this type of signals and the rest.

- ***Cut and paste* detector:** we detected signals created by cutting and pasting excerpts from other signals. For that, we compared the MFCC and the log-pitch contours of the test signals with those of the enrollment. Contours were aligned by DTW. The method was based on the idea that cut and paste would create discontinuities and strange intonation and energy patterns. Both spoofing detector provided error rates close to 0% in the dataset evaluated.

- **Tampering detector:** we used MFCC features together with GMM and SVM to detect low effort tampering attacks. The results evidenced that detection of these attacks is challenging.

As a consequence of this work we applied for two European patents:

- Cut and Paste Spoofing Detection Using Dynamic Time Warping, Agnitio S.L., Madrid, Spain, 2009 European Patent App. No: 09771309.3–2225 PCT/EP2009008851.
- Estimation of Reliability in Speaker Recognition, Agnitio S.L., Madrid, Spain, 2013, European Patent App. No: 13165466.7–1910.

We also released several publications:

- Villalba, J., Vaquero, C., Lleida, E., Ortega, A., Miguel, A., Garcia, J. E., Saz, O. (2008). Experiencia del I3A en la Evaluación de Reconocimiento de Locutor NIST 2008. In Proceedings of the IV Jornadas de Reconocimiento Biometrico de Personas, JRBP 2008. Valladolid, Spain.
- Villalba, J., Lleida, E., Ortega, A., Vaquero, C., Miguel, A. (2009). I3A System for Evalita 2009 Speaker Verification Application Evaluation. In Proceedings of the 11th Conference of the Italian Association for Artificial Intelligence. Reggio Emilia, Italy.
- Villalba, J., Vaquero, C., Lleida, E., Ortega, A., Miguel, A. (2010). I3A NIST SRE2010 System Description. In Proceedings of the V Jornadas de Reconocimiento Biometrico de Personas, JRBP 2010 (pp. 241–250). Huesca, Spain.
- Villalba, J., Lleida, E. (2010). Speaker Verification Performance Degradation against Spoofing and Tampering Attacks. In Proceedings of Fala 2010 (pp. 131–134). Vigo, Spain.
- Vaquero, C., Villalba, J., Ortega, A., Lleida, E. (2011). Speaker Verification on Summed Channel Conditions with Confidence Measures. Computacion Y Sistemas, 15(1), 27–37.
- Villalba, J., Brummer, N. (2011). Towards Fully Bayesian Speaker Recognition: Integrating Out the Between-Speaker Covariance. In Proceedings of the 12th Annual Conference of the International Speech Communication Association, Interspeech 2011 (pp. 505–508). Florence, Italy: ISCA.
- Villalba, J., Brummer, N., Lleida, E. (2011). Fully Bayesian Likelihood Ratios vs i-vector Length Normalization in Speaker Recognition Systems. In NIST SRE11 Speaker Recognition Workshop. Atlanta, Georgia, USA.
- Villalba, J., Lleida, E. (2011). Detecting Replay Attacks from Far-Field Recordings on Speaker Verification Systems. In Biometrics and ID Management, Proceedings of the COST 2101 European Workshop, BioID 2011 (pp. 274–285). Brandenburg, Germany: Springer Berlin Heidelberg.
- Villalba, J., Lleida, E. (2011). Preventing Replay Attacks on Speaker Verification Systems. In Proceedings of the IEEE International Carnahan Conference on Security Technology, ICCST 2011 (pp. 284–291). Mataro, Spain: IEEE.

- Villalba, J., Lleida, E. (2012). Bayesian Adaptation of PLDA Based Speaker Recognition to Domains with Scarce Development Data. In Proceedings of Odyssey 2012 - The Speaker and Language Recognition Workshop. Singapore: COLIPS.

- Villalba, J., Lleida, E. (2012). Bayesian Two-Covariance Model for Speaker Recognition: A Comparative between Integrating Out the Speakers Mean, Between-Speaker and Within-Speaker Covariances. In Proceedings of the VI Jornadas de Reconocimiento Biometrico de Personas, JRBP 2012 (pp. 179–188). Las Palmas de Gran Canaria, Spain.

- Villalba, J., Lleida, E., Ortega, A., Miguel, A. (2012). I3A SRE12 System Description. In NIST SRE12 Speaker Recognition Workshop. Orlando, Florida, USA.

- Villalba, J., Lleida, E., Ortega, A., Miguel, A. (2012). Reliability Estimation of the Speaker Verification Decisions Using Bayesian Networks to Combine Information from Multiple Speech Quality Measures. In Proceedings of IberSpeech 2012, Advances in Speech and Language Technologies for Iberian Languages (pp. 1–10). Madrid, Spain: Springer-Verlag Berlin Heidelberg.

- Villalba, J., Lleida, E. (2013). Handling i-Vectors from Different Recording Conditions Using Multi-Channel Simplified PLDA in Speaker Recognition. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013 (pp. 6763–6767). Vancouver, British Columbia, Canada: IEEE.

- Villalba, J., Lleida, E., Ortega, A., Miguel, A. (2013). A New Bayesian Network to Assess the Reliability of Speaker Verification Decisions. In Proceedings of the 14th Annual Conference of the International Speech Communication Association, Interspeech 2013 (pp. 3132–3136). Lyon, France: ISCA.

- Villalba, J., Lleida, E., Ortega, A., Miguel, A. (2013). The I3A Speaker Recognition System for NIST SRE12: Post-evaluation Analysis. In Proceedings of the 14th Annual Conference of the International Speech Communication Association, Interspeech 2013 (pp. 3679–3683). Lyon, France: ISCA.

- Villalba, J., Diez, M., Varona, A., Lleida, E. (2013). Handling Recordings Acquired Simultaneously over Multiple Channels with PLDA. In Proceedings of the 14th Annual Conference of the International Speech Communication Association, Interspeech 2013 (pp. 2509–2513). Lyon, France: ISCA.

13.3 Future Lines

Finally, we propose some future research lines that can be considered from the work in this thesis. Regarding quality measures and reliability, we propose:

- **Expand the number of distortion types:** in this thesis, we experimented with a dataset with additive and convolutive noise. We should try new noise types, non-linear distortions and other degradations that could affect the speech signal.

- **New quality measures:** the quality measures used in this thesis performed well on additive noise but not so well on reverberation. We need better measures that can relate reverberation with trial reliability. We will also need measures for the new distortion that we consider.

In the field of speaker and channel models, we propose:

- **Noise compensation at feature or i-vector extraction levels:** in Chapter 8, we showed that multichannel PLDA did not improved standard PLDA in noisy conditions. This makes us think that compensating for noise at i-vector level may be a task too complicated. The fact that noise alters the Gaussian responsibilities given the UBM, which are then used to compute sufficient statistics and then i-vectors, probability has a highly non-linear effect on the i-vector. Thus, we think that may be more effective to compensate noise and other distortions in early stages of the pipeline. Whatever technique that can improve those responsibilities should significantly improve performance. Recent promising works about this topic use deep neural networks to compute responsibilities [Lei et al., 2014b] or VTS to obtain a noisy UBM from the clean UBM and compute sufficient statistics with it [Lei et al., 2013, Lei et al., 2014a, Martínez et al., 2014].

- **Fast evaluation of fully Bayesian likelihood ratios:** in Chapter 9, we showed that we can obtain some improvement from fully Bayesian likelihood ratios. However, the Bayesian approach has a huge computational cost compared to standard likelihood ratios, which makes it unfeasible for large scale applications. Finding faster approximations to evaluate Bayesian ratios is another interesting line of work.

- **Unsupervised adaptation of PLDA:** in most domains, obtaining labeled development data is difficult, expensive or just impossible. If we are fortunate we will obtain unlabeled data. That means that we know neither who speaks in each recording nor how many speakers there in the dataset. The logical step is extending the work in Chapter 10 to adapt PLDA with unlabeled data. We have already started to work on it [Villalba and Lleida, 2014].

- **Adaptation of UBM and i-vector extractors:** as well as adapting PLDA, we could think on creating a framework to jointly adapt UBM, i-vector extractor and PLDA.

Finally, regarding attacks to speaker verification systems we think that future research should be oriented towards:

- **Development of a public dataset:** to foster research on attacks to speaker recognition, we need publicly available databases and common evaluations protocols. This would favor meaningful comparison of approaches and collaboration between institutions.

Part VI

Appendices

Appendix A

Bayesian Inference of a Gaussian Distribution

A.1 Introduction

Here, we write the equations needed for Bayesian inference of multivariate Gaussian distributions with non-informative priors. We use precision matrix notation that is the one preferred along this thesis. The results given here will be useful for the derivation of our Bayesian PLDA model. This appendix is based on [Minka, 1998].

A.2 Inferring a Gaussian distribution

The multivariate Gaussian distribution is defined by:

$$P\left(\mathbf{x}|\mathbf{m},\boldsymbol{\Lambda}\right)=\mathcal{N}\left(\mathbf{x}|\mathbf{m},\boldsymbol{\Lambda}^{-1}\right)=\left|\frac{\boldsymbol{\Lambda}}{2\pi}\right|^{1/2}\exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{m})^T\boldsymbol{\Lambda}(\mathbf{x}-\mathbf{m})\right) \tag{A.1}$$

where $\mathbf{m}$ is the mean vector and $\boldsymbol{\Lambda}$ a full precision matrix.

We assume a non-informative prior Π for $\mathbf{m}$ and $\boldsymbol{\Lambda}$ (Jeffrey's Prior):

$$\begin{aligned}
P\left(\mathbf{m},\boldsymbol{\Lambda}|\Pi\right)&=P\left(\mathbf{m}|\boldsymbol{\Lambda},\Pi\right)P\left(\boldsymbol{\Lambda}|\Pi\right) &\text{(A.2)}\\
&=\lim_{k\to 0}\mathcal{N}\left(\mathbf{m}|\mathbf{m}_0,(k\boldsymbol{\Lambda})^{-1}\right)\mathcal{W}\left(\boldsymbol{\Lambda}|\mathbf{W}_0/k,k\right) &\text{(A.3)}\\
&=\alpha\left|\frac{\boldsymbol{\Lambda}}{2\pi}\right|^{1/2}\left|\boldsymbol{\Lambda}\right|^{-(d+1)/2}\ . &\text{(A.4)}
\end{aligned}$$

First, we obtain the posteriors for $\mathbf{m}$ and $\boldsymbol{\Lambda}$, given the observations $\mathbf{X}=\{\mathbf{x}_1,\mathbf{x}_2,\dots,\mathbf{x}_N\}$

$$\begin{aligned}
P\left(\mathbf{m},\boldsymbol{\Lambda}|\mathbf{X},\Pi\right)&=\frac{P\left(\mathbf{m},\boldsymbol{\Lambda}|\Pi\right)}{P\left(\mathbf{X}|\Pi\right)}\left|\frac{\boldsymbol{\Lambda}}{2\pi}\right|^{N/2}\exp\left(-\frac{1}{2}\sum_{i=1}^{N}(\mathbf{x}_i-\mathbf{m})^T\boldsymbol{\Lambda}(\mathbf{x}_i-\mathbf{m})\right) &\text{(A.5)}\\
&=\frac{P\left(\mathbf{m},\boldsymbol{\Lambda}|\Pi\right)}{P\left(\mathbf{X}|\Pi\right)}\left|\frac{\boldsymbol{\Lambda}}{2\pi}\right|^{N/2}\exp\left(-\frac{N}{2}(\mathbf{m}-\overline{\mathbf{x}})^T\boldsymbol{\Lambda}(\mathbf{m}-\overline{\mathbf{x}})\right)\exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{S}\boldsymbol{\Lambda}\right)\right) &\text{(A.6)}
\end{aligned}$$

where

$$\overline{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \tag{A.7}$$

$$\mathbf{S} = \sum_{i=1}^{N}\left(\mathbf{x}_i - \overline{\mathbf{x}}\right)\left(\mathbf{x}_i - \overline{\mathbf{x}}\right)^T \ . \tag{A.8}$$

Now, we need the marginal posterior for the precision $\mathbf{\Lambda}$:

$$P\left(\mathbf{\Lambda}|\mathbf{X},\Pi\right) = \int_{\mathbf{m}} P\left(\mathbf{m},\mathbf{\Lambda}|\mathbf{X},\Pi\right)\ \mathrm{d}\mathbf{m} \tag{A.9}$$

$$= \frac{1}{P\left(\mathbf{X}|\Pi\right)}\frac{\alpha}{\left|\mathbf{\Lambda}\right|^{(d+1)/2}}\frac{1}{N^{d/2}}\left|\frac{\mathbf{\Lambda}}{2\pi}\right|^{N/2}\exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{S}\mathbf{\Lambda}\right)\right)$$
$$\int_{\mathbf{m}}\left|\frac{N\mathbf{\Lambda}}{2\pi}\right|^{1/2}\exp\left(-\frac{N}{2}(\mathbf{m}-\overline{\mathbf{x}})^T\mathbf{\Lambda}(\mathbf{m}-\overline{\mathbf{x}})\right)\ \mathrm{d}\mathbf{m} \tag{A.10}$$

$$= \frac{1}{P\left(\mathbf{X}|\Pi\right)}\frac{\alpha}{\left|\mathbf{\Lambda}\right|^{(d+1)/2}}\frac{1}{N^{d/2}}\left|\frac{\mathbf{\Lambda}}{2\pi}\right|^{N/2}\exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{S}\mathbf{\Lambda}\right)\right) \tag{A.11}$$

$$= \frac{1}{P\left(\mathbf{X}|\Pi\right)}P\left(\mathbf{X}|\mathbf{\Lambda}\right)P\left(\mathbf{\Lambda}|\Pi\right) \tag{A.12}$$

Now, we apply the following relation

$$\int_{\mathbf{V}\geq 0}\left|\mathbf{V}\right|^{-k-(d+1)/2}\exp\left(-\mathrm{tr}\left(\mathbf{A}\mathbf{V}^{-1}\right)\right)\ \mathrm{d}\mathbf{V} = \int_{\mathbf{W}\geq 0}\left|\mathbf{W}\right|^{k-(d+1)/2}\exp\left(-\mathrm{tr}\left(\mathbf{A}\mathbf{W}\right)\right)\ \mathrm{d}\mathbf{W}$$
$$= \left|\mathbf{A}\right|^{-k}\pi^{d(d-1)/4}\prod_{i=1}^{d}\Gamma\left(k-(i-1)/2\right) \qquad \text{if } k > (d-1)/2 \text{ and } \left|\mathbf{A}\right| > 0 \tag{A.13}$$

to obtain the marginal likelihood of the data

$$P\left(\mathbf{X}|\Pi\right) = \int_{\mathbf{\Lambda}}\frac{\alpha}{\left|\mathbf{\Lambda}\right|^{(d+1)/2}}\frac{1}{N^{d/2}}\left|\frac{\mathbf{\Lambda}}{2\pi}\right|^{N/2}\exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{S}\mathbf{\Lambda}\right)\right)\ \mathrm{d}\mathbf{\Lambda} \tag{A.14}$$

$$= \begin{cases}\alpha\frac{\pi^{d(d-1)/4}\prod_{i=1}^{d}\Gamma((N+1-i)/2)}{N^{d/2}\left|\pi\mathbf{S}\right|^{N/2}} & N > d\\ \alpha\left(\pi\left|\mathbf{S}_0^{-1}\mathbf{S}\right|_+\left|\mathbf{S}_0\right|\right)^{-N/2} & N \leq d\end{cases} \tag{A.15}$$

where $\mathbf{S}_0 = (\overline{\mathbf{x}}-\mathbf{m}_0)(\overline{\mathbf{x}}-\mathbf{m}_0)^T + \mathbf{V}_0$

We plug (A.15) into (A.11) to obtain the marginal posterior of $\mathbf{\Lambda}$ that is Wishart distributed:

$$P\left(\mathbf{\Lambda}|\mathbf{X},\Pi\right) = \frac{1}{Z_{Nd}\left|\mathbf{\Lambda}\right|^{(d+1)/2}}\left|\frac{\mathbf{S}\mathbf{\Lambda}}{2}\right|^{N/2}\exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{S}\mathbf{\Lambda}\right)\right) \tag{A.16}$$

$$= \mathcal{W}\left(\mathbf{\Lambda}|\mathbf{S}^{-1},N\right) \qquad \text{if } N > d \tag{A.17}$$

where $Z_{Nd} = \pi^{d(d-1)/4}\prod_{i=1}^{d}\Gamma\left((N+1-i)/2\right)$.

Finally, we derive the joint posterior for $\mathbf{m}$ and $\boldsymbol{\Lambda}$ by plugging-in (A.15) into (A.6):

$$\begin{aligned}
P\left(\mathbf{m}, \boldsymbol{\Lambda} | \mathbf{X}, \Pi\right) = & \left|\frac{\boldsymbol{\Lambda}}{2\pi}\right|^{1/2} \left|\boldsymbol{\Lambda}\right|^{-(d+1)/2} \frac{N^{d/2} \left|\pi \mathbf{S}\right|^{N/2}}{Z_{Nd}} \left|\frac{\boldsymbol{\Lambda}}{2\pi}\right|^{N/2} \\
& \exp\left(-\frac{N}{2}(\mathbf{m}-\overline{\mathbf{x}})^T \boldsymbol{\Lambda} (\mathbf{m}-\overline{\mathbf{x}})\right) \exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{S}\boldsymbol{\Lambda}\right)\right) \qquad \text{(A.18)} \\
= & \frac{1}{Z_{Nd}\left|\boldsymbol{\Lambda}\right|^{(d+1)/2}} \left|\frac{N}{\pi \mathbf{S}}\right|^{1/2} \left|\frac{\mathbf{S}\boldsymbol{\Lambda}}{2}\right|^{(N+1)/2} \\
& \exp\left(-\frac{N}{2}(\mathbf{m}-\overline{\mathbf{x}})^T \boldsymbol{\Lambda} (\mathbf{m}-\overline{\mathbf{x}})\right) \exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{S}\boldsymbol{\Lambda}\right)\right) \qquad \text{(A.19)} \\
= & \left[\left|\frac{N\boldsymbol{\Lambda}}{2\pi}\right|^{1/2} \exp\left(-\frac{N}{2}(\mathbf{m}-\overline{\mathbf{x}})^T \boldsymbol{\Lambda} (\mathbf{m}-\overline{\mathbf{x}})\right)\right] \\
& \left[\frac{1}{Z_{Nd}\left|\boldsymbol{\Lambda}\right|^{(d+1)/2}} \left|\frac{\mathbf{S}\boldsymbol{\Lambda}}{2}\right|^{N/2} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{S}\boldsymbol{\Lambda}\right)\right)\right] \qquad \text{(A.20)} \\
= & \mathcal{N}\left(\mathbf{m} | \overline{\mathbf{x}}, \left(N\boldsymbol{\Lambda}\right)^{-1}\right) \mathcal{W}\left(\boldsymbol{\Lambda} | \mathbf{S}^{-1}, N\right) \quad \text{if } N > d \qquad \text{(A.21)}
\end{aligned}$$

Summing up:

$$P\left(\mathbf{m}, \boldsymbol{\Lambda} | \mathbf{X}, \Pi\right) = P\left(\mathbf{m} | \boldsymbol{\Lambda}, \mathbf{X}, \Pi\right) P\left(\boldsymbol{\Lambda} | \mathbf{X}, \Pi\right) \tag{A.22}$$

$$P\left(\mathbf{m} | \boldsymbol{\Lambda}, \mathbf{X}, \Pi\right) = \mathcal{N}\left(\mathbf{m} | \overline{\mathbf{x}}, \left(N\boldsymbol{\Lambda}\right)^{-1}\right) \tag{A.23}$$

$$P\left(\boldsymbol{\Lambda} | \mathbf{X}, \Pi\right) = \mathcal{W}\left(\boldsymbol{\Lambda} | \mathbf{S}^{-1}, N\right) \qquad \text{if } N > d \tag{A.24}$$

We can also calculate the marginal posterior for the mean $\mathbf{m}$:

$$\begin{aligned}
P\left(\mathbf{m} | \mathbf{X}, \Pi\right) = & \int_{\boldsymbol{\Lambda}} P\left(\mathbf{m}, \boldsymbol{\Lambda} | \mathbf{X}, \Pi\right) \, \mathrm{d}\boldsymbol{\Lambda} \qquad \text{(A.25)} \\
= & \frac{1}{P\left(\mathbf{X} | \Pi\right)} \int_{\boldsymbol{\Lambda}} \frac{\alpha}{\left|\boldsymbol{\Lambda}\right|^{(d+1)/2}} \frac{1}{N^{d/2}} \left|\frac{\boldsymbol{\Lambda}}{2\pi}\right|^{N/2} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{S}\boldsymbol{\Lambda}\right)\right) \\
& \left|\frac{N\boldsymbol{\Lambda}}{2\pi}\right|^{1/2} \exp\left(-\frac{N}{2}(\mathbf{m}-\overline{\mathbf{x}})^T \boldsymbol{\Lambda} (\mathbf{m}-\overline{\mathbf{x}})\right) \, \mathrm{d}\boldsymbol{\Lambda} \qquad \text{(A.26)} \\
= & \frac{1}{P\left(\mathbf{X} | \Pi\right)} \int_{\boldsymbol{\Lambda}} \frac{\alpha}{\left|\boldsymbol{\Lambda}\right|^{(d+1)/2}} \left|\frac{\boldsymbol{\Lambda}}{2\pi}\right|^{(N+1)/2} \\
& \exp\left(-\frac{1}{2}\mathrm{tr}\left(\left(\mathbf{S} + N\left(\mathbf{m}-\overline{\mathbf{x}}\right)\left(\mathbf{m}-\overline{\mathbf{x}}\right)^T\right)\boldsymbol{\Lambda}\right)\right) \, \mathrm{d}\boldsymbol{\Lambda} \qquad \text{(A.27)} \\
= & \frac{1}{P\left(\mathbf{X} | \Pi\right)} \frac{\alpha Z_{(N+1)d}}{\pi^{(N+1)d/2}} \left|\mathbf{S} + N\left(\mathbf{m}-\overline{\mathbf{x}}\right)\left(\mathbf{m}-\overline{\mathbf{x}}\right)^T\right|^{-(N+1)/2} \qquad \text{(A.28)} \\
= & \frac{Z_{(N+1)d} \, N^{d/2}}{Z_{Nd} \, \pi^{(N+1)d/2}} \left|\pi \mathbf{S}\right|^{N/2} \left|\mathbf{S} + N\left(\mathbf{m}-\overline{\mathbf{x}}\right)\left(\mathbf{m}-\overline{\mathbf{x}}\right)^T\right|^{-(N+1)/2} \qquad \text{(A.29)}
\end{aligned}$$

$$= \frac{\Gamma\left((N+1)/2\right)\ N^{d/2}}{\Gamma\left((N+1-d)/2\right),\pi^{d/2}} \left|\mathbf{S}\right|^{-1/2} \left|\mathbf{I} + N\mathbf{S}^{-1}\left(\mathbf{m}-\overline{\mathbf{x}}\right)\left(\mathbf{m}-\overline{\mathbf{x}}\right)^{T}\right|^{-(N+1)/2} \tag{A.30}$$

$$= \frac{\Gamma\left((N+1)/2\right)\ N^{d/2}}{\Gamma\left((N+1-d)/2\right),\pi^{d/2}} \left|\mathbf{S}\right|^{-1/2} \left|\mathbf{I} + N\mathbf{S}^{-1}\left(\mathbf{m}-\overline{\mathbf{x}}\right)\left(\mathbf{m}-\overline{\mathbf{x}}\right)^{T}\right|^{-(N+1)/2} \tag{A.31}$$

$$= \frac{\Gamma\left((N+1)/2\right)}{\Gamma\left((N+1-d)/2\right)} \left|\frac{N\mathbf{S}^{-1}}{\pi}\right|^{1/2} \left(1+N(\mathbf{m}-\overline{\mathbf{x}})^{T}\mathbf{S}^{-1}(\mathbf{m}-\overline{\mathbf{x}})\right)^{-(N+1)/2} \tag{A.32}$$

$$= \mathcal{T}\left(\mathbf{m}|\overline{\mathbf{x}}, \mathbf{S}/N^{2}, N+1-d\right) \qquad \text{if } N > d \tag{A.33}$$

where we used the matrix relation in [Minka, 2000]

$$|\mathbf{I}+\mathbf{BC}| = |\mathbf{I}+\mathbf{CB}| \tag{A.34}$$

and $\mathcal{T}$ is the Student's T distribution defined as

$$\mathcal{T}\left(\mathbf{x}|\mathbf{m}, \mathbf{\Lambda}^{-1}, N\right) = \frac{\Gamma\left((N+d)/2\right)}{\Gamma\left(N/2\right)} \left|\frac{\mathbf{\Lambda}}{N\pi}\right|^{1/2} \left(1+\frac{1}{N}(\mathbf{x}-\mathbf{m})^{T}\mathbf{\Lambda}(\mathbf{x}-\mathbf{m})\right)^{-(N+d)/2} . \tag{A.35}$$

A.3 Inferring a Gaussian distribution with given mean

Here, we particularize the results of the previous section for a distribution where the mean is known $\mathbf{m}_0$. The Jeffrey's prior Π for $\mathbf{\Lambda}$ is:

$$P\left(\mathbf{\Lambda}|\Pi\right) = \lim_{k\to 0} \mathcal{W}\left(\mathbf{\Lambda}|\mathbf{W}_0/k, k\right) \tag{A.36}$$

$$= \alpha \left|\mathbf{\Lambda}\right|^{-(d+1)/2} \tag{A.37}$$

The posterior for $\mathbf{\Lambda}$, given a set of observations $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ is

$$P\left(\mathbf{\Lambda}|\mathbf{X}, \Pi, \mathbf{m}_0\right) = \frac{P\left(\mathbf{\Lambda}|\Pi\right)}{P\left(\mathbf{X}|\Pi, \mathbf{m}_0\right)} \left|\frac{\mathbf{\Lambda}}{2\pi}\right|^{N/2} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{S\Lambda}\right)\right) \tag{A.38}$$

where

$$\mathbf{S} = \sum_{i=1}^{N} \left(\mathbf{x}_i - \mathbf{m}_0\right)\left(\mathbf{x}_i - \mathbf{m}_0\right)^{T} \ . \tag{A.39}$$

Now, we use (A.13) to compute $P\left(\mathbf{X}|\Pi\right)$:

$$P\left(\mathbf{X}|\Pi, \mathbf{m}_0\right) = \int_{\mathbf{\Lambda}} \frac{\alpha}{\left|\mathbf{\Lambda}\right|^{(d+1)/2}} \left|\frac{\mathbf{\Lambda}}{2\pi}\right|^{N/2} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{S\Lambda}\right)\right) \,\mathrm{d}\mathbf{\Lambda} \tag{A.40}$$

$$= \alpha\frac{\pi^{d(d-1)/4}\prod_{i=1}^{d}\Gamma\left((N+1-i)/2\right)}{\left|\pi\mathbf{S}\right|^{N/2}} \qquad N > d \, . \tag{A.41}$$

Finally, we plug (A.41) into (A.38) to obtain the marginal posterior of $\mathbf{\Lambda}$:

$$P\left(\mathbf{\Lambda}|\mathbf{X}, \Pi\right) = \frac{1}{Z_{Nd}\left|\mathbf{\Lambda}\right|^{(d+1)/2}} \left|\frac{\mathbf{S\Lambda}}{2}\right|^{N/2} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{S\Lambda}\right)\right) \quad \text{if } N > d \tag{A.42}$$

$$= \mathcal{W}\left(\mathbf{\Lambda}|\mathbf{S}^{-1}, N\right) \tag{A.43}$$

where $Z_{Nd} = \pi^{d(d-1)/4}\prod_{i=1}^{d}\Gamma\left((N+1-i)/2\right)$.

Appendix B

Proofs for Bayesian Network to Model Variations of the Speaker Verification Score Given Quality Measures

B.1 Bayesian Network

This appendix, we derive the equations related to the Bayesian network presented in Chapter 6. Figure B.1 illustrates the graphical model of the network.

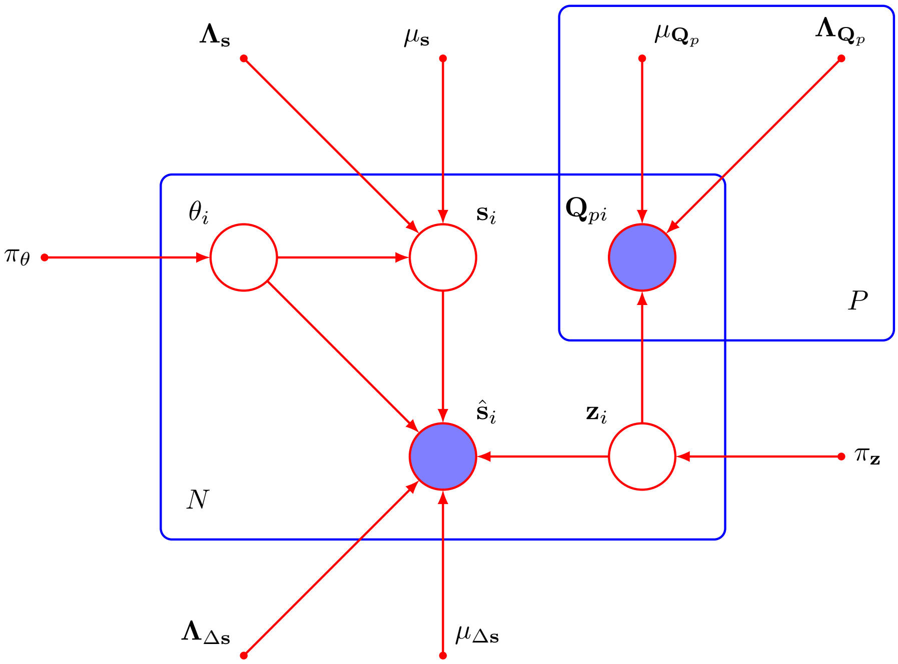


Figure B.1: BN to model SV score variations in adverse environments.

The nodes included in the Bayesian network are:

- $\hat{\mathbf{s}}_i$ is the *noisy observed score* given by the SV system.

- $\mathbf{s}_i$ is the *hidden clean score.* The relation between $\mathbf{s}_i$ and $\hat{\mathbf{s}}_i$ is

$$\hat{\mathbf{s}}_i = \mathbf{s}_i + \Delta\mathbf{s}_i \ . \tag{B.1}$$

- $\mathbf{z}_i$ are the states of quality. It is a 1-of-K binary vector with elements z_{ik} with $k = 1, \dots, K$. Each possible value of $\mathbf{z}_i$ corresponds to a different type or level of distortion of the speech segments.

- $\mathbf{Q}_{ip}$ are the observed quality measures. We consider that there are P groups of quality measures that are independent between them given $\mathbf{z}_i$. Thus, we can force independence between variables that we think that should not be correlated. We denote by $\mathbf{Q}_i = \{\mathbf{Q}_{ip}\}_{p=1}^{P}$ the set of measures of trial i.

- θ_i is the labeling of the trial. $\theta_i \in \{\mathcal{T}, \mathcal{N}\}$ where $\mathcal{T}$ is the target hypothesis and $\mathcal{N}$ is the non-target hypothesis.

- $\pi_\theta = (P_\mathcal{T}, P_\mathcal{N})$ is the hypothesis prior.

- $\pi_\mathbf{z}$ are the priors over the quality states.

The conditional distributions that define the nodes of the network are:

$$\begin{aligned} P\left(\mathbf{s}_i|\theta_i = \theta\right) =& \mathcal{N}\left(\mathbf{s}|\mu_{\mathbf{s}_\theta}, \mathbf{\Lambda}_{\mathbf{s}_\theta}^{-1}\right) & \text{(B.2)} \\ P\left(\hat{\mathbf{s}}_i|\mathbf{s}_i, z_{ik} = 1, \theta_i = \theta\right) =& \mathcal{N}\left(\hat{\mathbf{s}}_i|\mathbf{s}_i + \mu_{\Delta\mathbf{s}_{k\theta}}, \mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta}}^{-1}\right) & \text{(B.3)} \\ P\left(\mathbf{Q}_i|z_{ik} = 1\right) =& \prod_{p=1}^{P} \mathcal{N}\left(\mathbf{Q}_{ip}|\mu_{\mathbf{Q}_{pk}}, \mathbf{\Lambda}_{\mathbf{Q}_{pk}}^{-1}\right) & \text{(B.4)} \\ P\left(\mathbf{z}_i\right) =& \prod_{k=1}^{K} \pi_{z_k}^{z_{ik}} \ . & \text{(B.5)} \end{aligned}$$

B.2 Posterior Distribution of the Hidden Score

B.2.1 Case with θ and $\mathbf{z}$ observed

Here, we derive the posterior $P\left(\mathbf{s}|\hat{\mathbf{s}}, \theta, \mathbf{z}\right)$. First, we need the joint distribution:

$$\begin{aligned} P\left(\mathbf{s}, \hat{\mathbf{s}}|\theta, z_k = 1\right) =& P\left(\hat{\mathbf{s}}|\mathbf{s}, \theta, z_k = 1\right) P\left(\mathbf{s}|\theta\right) & \text{(B.6)} \\ =& \mathcal{N}\left(\hat{\mathbf{s}}|\mathbf{s} + \mu_{\Delta\mathbf{s}_{k\theta}}, \mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta}}^{-1}\right) \mathcal{N}\left(\mathbf{s}|\mu_{\mathbf{s}_\theta}, \mathbf{\Lambda}_{\mathbf{s}_\theta}^{-1}\right) \ . & \text{(B.7)} \end{aligned}$$

Then, we develop the posterior

$$\begin{aligned} P\left(\mathbf{s}|\hat{\mathbf{s}}, \theta, z_k = 1\right) =& \frac{P\left(\mathbf{s}, \hat{\mathbf{s}}|\theta, z_k = 1\right)}{P\left(\hat{\mathbf{s}}|\theta, z_k = 1\right)} & \text{(B.8)} \\ =& \frac{\mathcal{N}\left(\hat{\mathbf{s}}|\mathbf{s} + \mu_{\Delta\mathbf{s}_{k\theta}}, \mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta}}^{-1}\right) \mathcal{N}\left(\mathbf{s}|\mu_{\mathbf{s}_\theta}, \mathbf{\Lambda}_{\mathbf{s}_\theta}^{-1}\right)}{\int \mathcal{N}\left(\hat{\mathbf{s}}|\mathbf{s} + \mu_{\Delta\mathbf{s}_{k\theta}}, \mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta}}^{-1}\right) \mathcal{N}\left(\mathbf{s}|\mu_{\mathbf{s}_\theta}, \mathbf{\Lambda}_{\mathbf{s}_\theta}^{-1}\right) \, d\mathbf{s}} & \text{(B.9)} \end{aligned}$$

where the numerator is

$$
\begin{align}
&\mathcal{N}\left(\hat{\mathbf{s}}|\mathbf{s}+\mu_{\Delta\mathbf{s}_{k\theta}},\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}^{-1}\right)\mathcal{N}\left(\mathbf{s}|\mu_{\mathbf{s}_\theta},\boldsymbol{\Lambda}_{\mathbf{s}_\theta}^{-1}\right) \tag{B.10}\\
&=\left|\frac{\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}}{2\pi}\right|^{1/2}\exp\left(-\frac{1}{2}(\hat{\mathbf{s}}-\mathbf{s}-\mu_{\Delta\mathbf{s}_{k\theta}})^T\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}(\hat{\mathbf{s}}-\mathbf{s}-\mu_{\Delta\mathbf{s}_{k\theta}})\right) \notag\\
&\quad\times\left|\frac{\boldsymbol{\Lambda}_{\mathbf{s}_\theta}}{2\pi}\right|^{1/2}\exp\left(-\frac{1}{2}(\mathbf{s}-\mu_{\mathbf{s}_\theta})^T\boldsymbol{\Lambda}_{\mathbf{s}_\theta}(\mathbf{s}-\mu_{\mathbf{s}_\theta})\right) \tag{B.11}\\
&=\frac{1}{2\pi}\left|\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\right|^{1/2}\exp\Bigg(-\frac{1}{2}\mathbf{s}^T\left(\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}+\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\right)\mathbf{s}+\mathbf{s}^T\left(\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}\left(\hat{\mathbf{s}}-\mu_{\Delta\mathbf{s}_{k\theta}}\right)+\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\mu_{\mathbf{s}_\theta}\right) \notag\\
&\quad-\frac{1}{2}\left(\hat{\mathbf{s}}-\mu_{\Delta\mathbf{s}_{k\theta}}\right)^T\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}\left(\hat{\mathbf{s}}-\mu_{\Delta\mathbf{s}_{k\theta}}\right)-\frac{1}{2}\mu_{\mathbf{s}_\theta}^T\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\mu_{\mathbf{s}_\theta}\Bigg) \tag{B.12}\\
&=\frac{1}{2\pi}\left|\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\right|^{1/2}\exp\left(-\frac{1}{2}\gamma_{k\theta}\right)\exp\left(-\frac{1}{2}(\mathbf{s}-\mu'_{\mathbf{s}_{k\theta}})^T\boldsymbol{\Lambda}'_{\mathbf{s}_{k\theta}}(\mathbf{s}-\mu'_{\mathbf{s}_{k\theta}})\right) \tag{B.13}
\end{align}
$$

and

$$
\begin{align}
\boldsymbol{\Lambda}'_{\mathbf{s}_{k\theta}} &=\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}+\boldsymbol{\Lambda}_{\mathbf{s}_\theta} \tag{B.14}\\
\mu'_{\mathbf{s}_{k\theta}} &=\boldsymbol{\Lambda}'^{-1}_{\mathbf{s}_{k\theta}}\left(\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}\left(\hat{\mathbf{s}}-\mu_{\Delta\mathbf{s}_{k\theta}}\right)+\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\mu_{\mathbf{s}_\theta}\right) \tag{B.15}\\
\gamma_{k\theta} &=\left(\hat{\mathbf{s}}-\mu_{\Delta\mathbf{s}_{k\theta}}\right)^T\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}\left(\hat{\mathbf{s}}-\mu_{\Delta\mathbf{s}_{k\theta}}\right)+\mu_{\mathbf{s}_\theta}^T\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\mu_{\mathbf{s}_\theta}-\mu'^T_{\mathbf{s}_{k\theta}}\boldsymbol{\Lambda}'_{\mathbf{s}_{k\theta}}\mu'_{\mathbf{s}_{k\theta}}\ . \tag{B.16}
\end{align}
$$

We can expand $\gamma_{k\theta}$:

$$
\begin{align}
\gamma_{k\theta} &=\left(\hat{\mathbf{s}}-\mu_{\Delta\mathbf{s}_{k\theta}}\right)^T\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}\left(\hat{\mathbf{s}}-\mu_{\Delta\mathbf{s}_{k\theta}}\right)+\mu_{\mathbf{s}_\theta}^T\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\mu_{\mathbf{s}_\theta} \notag\\
&\quad-\left(\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}\left(\hat{\mathbf{s}}-\mu_{\Delta\mathbf{s}_{k\theta}}\right)+\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\mu_{\mathbf{s}_\theta}\right)^T\left(\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}+\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\right)^{-1}\left(\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}\left(\hat{\mathbf{s}}-\mu_{\Delta\mathbf{s}_{k\theta}}\right)+\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\mu_{\mathbf{s}_\theta}\right) \tag{B.17}\\
&=\left(\hat{\mathbf{s}}-\mu_{\Delta\mathbf{s}_{k\theta}}\right)^T\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}\left(\hat{\mathbf{s}}-\mu_{\Delta\mathbf{s}_{k\theta}}\right)+\mu_{\mathbf{s}_\theta}^T\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\mu_{\mathbf{s}_\theta} \notag\\
&\quad-\left(\hat{\mathbf{s}}-\mu_{\Delta\mathbf{s}_{k\theta}}\right)^T\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}\left(\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}+\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\right)^{-1}\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}\left(\hat{\mathbf{s}}-\mu_{\Delta\mathbf{s}_{k\theta}}\right) \notag\\
&\quad-\mu_{\mathbf{s}_\theta}^T\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\left(\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}+\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\right)^{-1}\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\mu_{\mathbf{s}_\theta} \notag\\
&\quad-2\mu_{\mathbf{s}_\theta}^T\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\left(\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}+\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\right)^{-1}\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}\left(\hat{\mathbf{s}}-\mu_{\Delta\mathbf{s}_{k\theta}}\right) \tag{B.18}\\
&=\left(\hat{\mathbf{s}}-\mu_{\Delta\mathbf{s}_{k\theta}}\right)^T\left(\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}-\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}\left(\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}+\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\right)^{-1}\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}\right)\left(\hat{\mathbf{s}}-\mu_{\Delta\mathbf{s}_{k\theta}}\right) \notag\\
&\quad+\mu_{\mathbf{s}_\theta}^T\left(\boldsymbol{\Lambda}_{\mathbf{s}_\theta}-\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\left(\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}+\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\right)^{-1}\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\right)\mu_{\mathbf{s}_\theta} \notag\\
&\quad-2\mu_{\mathbf{s}_\theta}^T\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\left(\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}+\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\right)^{-1}\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}\left(\hat{\mathbf{s}}-\mu_{\Delta\mathbf{s}_{k\theta}}\right) \tag{B.19}\\
&=\left(\hat{\mathbf{s}}-\mu_{\Delta\mathbf{s}_{k\theta}}\right)^T\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\left(\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}+\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\right)^{-1}\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}\left(\hat{\mathbf{s}}-\mu_{\Delta\mathbf{s}_{k\theta}}\right) \notag\\
&\quad+\mu_{\mathbf{s}_\theta}^T\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\left(\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}+\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\right)^{-1}\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}\mu_{\mathbf{s}_\theta} \notag\\
&\quad-2\mu_{\mathbf{s}_\theta}^T\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\left(\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}+\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\right)^{-1}\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}\left(\hat{\mathbf{s}}-\mu_{\Delta\mathbf{s}_{k\theta}}\right) \tag{B.20}\\
&=(\hat{\mathbf{s}}-(\mu_{\mathbf{s}_\theta}+\mu_{\Delta\mathbf{s}_{k\theta}}))^T\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\left(\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}+\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\right)^{-1}\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}(\hat{\mathbf{s}}-(\mu_{\mathbf{s}_\theta}+\mu_{\Delta\mathbf{s}_{k\theta}}))\ . \tag{B.21}
\end{align}
$$

Now, we solve the integral

$$\begin{aligned}
P\left(\hat{\mathbf{s}}|\theta, z_k = 1\right) = & \int \mathcal{N}\left(\hat{\mathbf{s}}|\mathbf{s} + \mu_{\Delta\mathbf{s}_{k\theta}}, \boldsymbol{\Lambda}^{-1}_{\Delta\mathbf{s}_{k\theta}}\right) \mathcal{N}\left(\mathbf{s}|\mu_{\mathbf{s}_\theta}, \boldsymbol{\Lambda}^{-1}_{\mathbf{s}_\theta}\right) \; \mathrm{d}\mathbf{s} \qquad \text{(B.22)} \\
= & \frac{1}{2\pi}\left|\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\right|^{1/2} \exp\left(-\frac{1}{2}\gamma_{k\theta}\right) \int \exp\left(-\frac{1}{2}(\mathbf{s}-\mu'_{\mathbf{s}_{k\theta}})^T \boldsymbol{\Lambda}'_{\mathbf{s}_{k\theta}}(\mathbf{s}-\mu'_{\mathbf{s}_{k\theta}})\right) \; \mathrm{d}\mathbf{s} \qquad \text{(B.23)} \\
= & \frac{1}{2\pi}\left|\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\right|^{1/2} \exp\left(-\frac{1}{2}\gamma_{k\theta}\right) \left|\frac{\boldsymbol{\Lambda}'_{\mathbf{s}_{k\theta}}}{2\pi}\right|^{-1/2} \qquad \text{(B.24)} \\
= & \left|\frac{\boldsymbol{\Lambda}_{\mathbf{s}_\theta}\boldsymbol{\Lambda}'^{-1}_{\mathbf{s}_{k\theta}}\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}}{2\pi}\right|^{1/2} \exp\left(-\frac{1}{2}\gamma_{k\theta}\right) \qquad \text{(B.25)}
\end{aligned}$$

By combining (B.9), (B.13) and (B.24), we obtain

$$P\left(\mathbf{s}|\hat{\mathbf{s}}, \theta, z_k = 1\right) = \left|\frac{\boldsymbol{\Lambda}'_{\mathbf{s}_{k\theta}}}{2\pi}\right|^{1/2} \exp\left(-\frac{1}{2}(\mathbf{s}-\mu'_{\mathbf{s}_{k\theta}})^T \boldsymbol{\Lambda}'_{\mathbf{s}_{k\theta}}(\mathbf{s}-\mu'_{\mathbf{s}_{k\theta}})\right) \; . \qquad \text{(B.26)}$$

And summing up:

$$P\left(\mathbf{s}|\hat{\mathbf{s}}, \theta, z_k = 1\right) = \mathcal{N}\left(\mathbf{s}|\mu'_{\mathbf{s}_{k\theta}}, \boldsymbol{\Lambda}'^{-1}_{\mathbf{s}_{k\theta}}\right) \qquad \text{(B.27)}$$

where

$$\begin{aligned}
\boldsymbol{\Lambda}'_{\mathbf{s}_{k\theta}} = & \boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}} + \boldsymbol{\Lambda}_{\mathbf{s}_\theta} \qquad \text{(B.28)} \\
\mu'_{\mathbf{s}_{k\theta}} = & \boldsymbol{\Lambda}'^{-1}_{\mathbf{s}_{k\theta}}\left(\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}}\left(\hat{\mathbf{s}} - \mu_{\Delta\mathbf{s}_{k\theta}}\right) + \boldsymbol{\Lambda}_{\mathbf{s}_\theta}\mu_{\mathbf{s}_\theta}\right) \qquad \text{(B.29)}
\end{aligned}$$

and

$$P\left(\hat{\mathbf{s}}|\theta, z_k = 1\right) = \mathcal{N}\left(\hat{\mathbf{s}}|\mu'_{\hat{\mathbf{s}}_{k\theta}}, \boldsymbol{\Lambda}'^{-1}_{\hat{\mathbf{s}}_{k\theta}}\right) \qquad \text{(B.30)}$$

where

$$\begin{aligned}
\boldsymbol{\Lambda}'_{\hat{\mathbf{s}}_{k\theta}} = & \boldsymbol{\Lambda}_{\mathbf{s}_\theta}\boldsymbol{\Lambda}'^{-1}_{\mathbf{s}_{k\theta}}\boldsymbol{\Lambda}_{\Delta\mathbf{s}_{k\theta}} \qquad \text{(B.31)} \\
\mu'_{\hat{\mathbf{s}}_{k\theta}} = & \mu_{\mathbf{s}_\theta} + \mu_{\Delta\mathbf{s}_{k\theta}} \; . \qquad \text{(B.32)}
\end{aligned}$$

B.2.2 Case with θ hidden and $\mathbf{z}$ observed

Now, we consider a more general case where θ is also hidden. Then the posterior is computing by integrating out θ:

$$\begin{aligned}
P\left(\mathbf{s}|\hat{\mathbf{s}}, z_k = 1\right) = & \sum_{\theta\in\{\mathcal{T},\mathcal{N}\}} P\left(\mathbf{s}, \theta|\hat{\mathbf{s}}, z_k = 1\right) \qquad \text{(B.33)} \\
= & \sum_{\theta\in\{\mathcal{T},\mathcal{N}\}} P\left(\mathbf{s}|\hat{\mathbf{s}}, \theta, z_k = 1\right) P\left(\theta|\hat{\mathbf{s}}, z_k = 1\right) \qquad \text{(B.34)} \\
= & \sum_{\theta\in\{\mathcal{T},\mathcal{N}\}} P\left(\theta|\hat{\mathbf{s}}, z_k = 1\right) \mathcal{N}\left(\mathbf{s}|\mu'_{\mathbf{s}_{k\theta}}, \boldsymbol{\Lambda}'^{-1}_{\mathbf{s}_{k\theta}}\right) \; . \qquad \text{(B.35)}
\end{aligned}$$

The posterior of $\mathbf{s}$ is a mixture of two Gaussians and the weights are the label posteriors $P(\theta|\hat{\mathbf{s}}, z_k = 1)$. The target posterior is computed as

$$P(\theta = \mathcal{T}|\hat{\mathbf{s}}, z_k = 1) = \frac{P(\hat{\mathbf{s}}|\mathcal{T}, z_k = 1) P_{\mathcal{T}}}{\sum_{\theta\in\{\mathcal{T},\mathcal{N}\}} P(\hat{\mathbf{s}}|\theta, z_k = 1) P(\theta)} \tag{B.36}$$

$$= \frac{1}{1 + \frac{P(\hat{\mathbf{s}}|\mathcal{N}, z_k=1)}{P(\hat{\mathbf{s}}|\mathcal{T}, z_k=1)} \frac{1-P_{\mathcal{T}}}{P_{\mathcal{T}}}} \tag{B.37}$$

$$= \frac{1}{1 + \exp(-\ln R(\hat{\mathbf{s}}, z_k = 1) - \text{logit}(P_{\mathcal{T}}))} \tag{B.38}$$

where

$$\ln R(\hat{\mathbf{s}}, z_k = 1) = \ln P(\hat{\mathbf{s}}|\mathcal{T}, z_k = 1) - \ln P(\hat{\mathbf{s}}|\mathcal{N}, z_k = 1) \tag{B.39}$$

and $Prob\hat{\mathbf{s}}|\mathcal{T}, z_k = 1$ is given by (B.30).

B.2.3 General case

Finally in the most general case, $\mathbf{z}$, θ and $\mathbf{s}$ are hidden. The posterior of $\mathbf{s}$ is computed by integrating out the rest of hidden variables:

$$P(\mathbf{s}|\hat{\mathbf{s}}, \mathbf{Q}) = \sum_{\theta\in\{\mathcal{T},\mathcal{N}\}} \sum_{k=1}^{K} P(\mathbf{s}, \theta, z_k = 1|\hat{\mathbf{s}}, \mathbf{Q}) \tag{B.40}$$

$$= \sum_{\theta\in\{\mathcal{T},\mathcal{N}\}} \sum_{k=1}^{K} P(\theta, z_k = 1|\hat{\mathbf{s}}, \mathbf{Q}) P(\mathbf{s}|\hat{\mathbf{s}}, \theta, z_k = 1) \tag{B.41}$$

$$= \sum_{\theta\in\{\mathcal{T},\mathcal{N}\}} \sum_{k=1}^{K} P(\theta, z_k = 1|\hat{\mathbf{s}}, \mathbf{Q}) \mathcal{N}\left(\mathbf{s}|\mu'_{\mathbf{s}_{k\theta}}, \mathbf{\Lambda}'^{-1}_{\mathbf{s}_{k\theta}}\right) \tag{B.42}$$

where

$$P(\theta, z_k = 1|\hat{\mathbf{s}}, \mathbf{Q}) = \frac{P(\hat{\mathbf{s}}, \mathbf{Q}|\theta, z_k = 1) P(\theta, z_k = 1)}{P(\hat{\mathbf{s}}, \mathbf{Q})} \tag{B.43}$$

$$= \frac{P(\hat{\mathbf{s}}|\theta, z_k = 1) P(\mathbf{Q}|z_k = 1) P(\theta) \pi_{z_k}}{\sum_{\theta\in\{\mathcal{T},\mathcal{N}\}} \sum_{k=1}^{K} P(\hat{\mathbf{s}}|\theta, z_k = 1) P(\mathbf{Q}|z_k = 1) P(\theta) \pi_{z_k}} . \tag{B.44}$$

The term $P(\mathbf{Q}|z_k = 1)$ is given by (B.4), $P(\hat{\mathbf{s}}|\theta, z_k = 1)$ is given by (B.30) and; $\mathbf{\Lambda}'_{\mathbf{s}_{k\theta}}$ and $\mu'_{\mathbf{s}_{k\theta}}$ are given by (B.28) and (B.29).

B.3 EM algorithm

B.3.1 General case, training with s and z hidden

In the general case, we assume $\mathbf{z}$ and $\mathbf{s}$ hidden, and θ known during training.

B.3.2 E-step

In the E-step we compute the posterior of the hidden variables, $P\left(\mathbf{s}_i,\mathbf{z}_i|\hat{\mathbf{s}}_i,\mathbf{Q}_i,\theta_i\right)$:

$$P\left(\mathbf{s}_i,\mathbf{z}_i|\hat{\mathbf{s}}_i,\mathbf{Q}_i,\theta_i\right)=P\left(\mathbf{s}_i|\hat{\mathbf{s}}_i,\theta_i,\mathbf{z}_i\right)P\left(\mathbf{z}_i|\hat{\mathbf{s}}_i,\mathbf{Q}_i,\theta_i\right)\ . \tag{B.45}$$

The first term is given by (B.27).

The second term is

$$P\left(z_{ik}=1|\hat{\mathbf{s}}_i,\mathbf{Q}_i,\theta_i\right)=\frac{P\left(\hat{\mathbf{s}}_i,\mathbf{Q}_i|z_{ik}=1,\theta_i\right)P\left(z_{ik}=1\right)}{P\left(\hat{\mathbf{s}}_i,\mathbf{Q}_i|\theta_i\right)} \tag{B.46}$$

$$=\frac{\pi_{z_k}P\left(\hat{\mathbf{s}}_i|z_{ik}=1,\theta_i\right)P\left(\mathbf{Q}_i|z_{ik}=1\right)}{\sum_{k=1}^{K}\pi_{z_k}P\left(\hat{\mathbf{s}}_i|z_{ik}=1,\theta_i\right)P\left(\mathbf{Q}_i|z_{ik}=1\right)} \tag{B.47}$$

where $P\left(\hat{\mathbf{s}}_i|z_{ik}=1,\theta_i\right)$ is given by (B.30). We define

$$\gamma(z_{ik})=P\left(z_{ik}=1|\hat{\mathbf{s}}_i,\mathbf{Q}_i,\theta_i\right) \tag{B.48}$$

to simplify the notation in following equations.

B.3.3 M-step

In the M-step, we maximize the EM auxiliary function: We maximize the EM auxiliary function:

$$\mathcal{Q}(\mathcal{M})=\sum_{i=1}^{N}\mathrm{E}\left[\ln P\left(\hat{\mathbf{s}}_i,\mathbf{Q}_i,\mathbf{s}_i,\mathbf{z}_i|\theta_i\right)\right]\ . \tag{B.49}$$

The joint probability of the observed and hidden variables can be decomposed as:

$$P\left(\hat{\mathbf{s}}_i,\mathbf{Q}_i,\mathbf{s}_i,\mathbf{z}_i|\theta_i\right)=P\left(\hat{\mathbf{s}}_i|\mathbf{s}_i,\mathbf{z}_i,\theta_i\right)P\left(\mathbf{Q}_i|\mathbf{z}_i\right)P\left(\mathbf{s}_i|\theta_i\right)P\left(\mathbf{z}_i\right)\ . \tag{B.50}$$

This allow us to write the auxiliary function as:

$$\mathcal{Q}(\mathcal{M})==\sum_{i=1}^{N}\mathrm{E}\left[\ln P\left(\hat{\mathbf{s}}_i|\mathbf{s}_i,\mathbf{z}_i,\theta_i\right)\right]+\mathrm{E}\left[\ln P\left(\mathbf{Q}_i|\mathbf{z}_i\right)\right]+\mathrm{E}\left[\ln P\left(\mathbf{s}_i|\theta_i\right)\right]+\mathrm{E}\left[\ln P\left(\mathbf{z}_i\right)\right] \tag{B.51}$$

First, we maximize respect to $\pi_{\mathbf{z}}$ with the restriction that $\sum_{k=1}^{K}\pi_{z_k}=1$. We obtain:

$$\pi_{z_k}=\frac{N_{z_k}}{\sum_{k=1}^{K}N_{z_k}} \tag{B.52}$$

where we defined

$$N_{z_k}=\sum_{i=1}^{N}\gamma(z_{ik})\ . \tag{B.53}$$

By maximizing with respect to $\mu_{\mathbf{Q}_p}$ and $\mathbf{\Lambda}_{\mathbf{Q}_p}$, we obtain:

$$\mu_{\mathbf{Q}_{pk}}=\frac{1}{N_{z_k}}\sum_{i=1}^{N}\gamma(z_{ik})\mathbf{Q}_{ip} \tag{B.54}$$

$$\mathbf{\Lambda}_{\mathbf{Q}_{pk}}^{-1}=\frac{1}{N_{z_k}}\sum_{i=1}^{N}\gamma(z_{ik})\left(\mathbf{Q}_{ip}-\mu_{\mathbf{Q}_{pk}}\right)\left(\mathbf{Q}_{ip}-\mu_{\mathbf{Q}_{pk}}\right)^T \tag{B.55}$$

We maximize respect to $\mu_{\mathbf{s}}$ and $\mathbf{\Lambda}_{\mathbf{s}}$ to obtain:

$$\mu_{\mathbf{s}_\theta} = \frac{1}{N_\theta}\sum_{i=1}^{N} t_{i\theta}\mathrm{E}\left[\mathbf{s}_i\right] \tag{B.56}$$

$$\mathbf{\Lambda}_{\mathbf{s}_\theta}^{-1} = \frac{1}{N_\theta}\sum_{i=1}^{N} t_{i\theta}\mathrm{E}\left[\left(\mathbf{s}_i-\mu_{\mathbf{s}_\theta}\right)\left(\mathbf{s}_i-\mu_{\mathbf{s}_\theta}\right)^T\right] \tag{B.57}$$

$$= \frac{1}{N_\theta}\sum_{i=1}^{N} t_{i\theta}\mathrm{E}\left[\mathbf{s}_i\mathbf{s}_i^T\right] - \mu_{\mathbf{s}_\theta}\mu_{\mathbf{s}_\theta}^T \tag{B.58}$$

where we defined $t_{i\theta}=1$ if $\theta_i=\theta$ and $t_{i\theta}=0$ if $\theta_i\neq\theta$; and $N_\theta=\sum_{i=1}^N t_{i\theta}$.

Now, we manipulate the term of the auxiliary that depends on $\mu_{\Delta\mathbf{s}}$ and $\mathbf{\Lambda}_{\Delta\mathbf{s}}$:

$$\mathcal{Q}(\mu_{\Delta\mathbf{s}},\mathbf{\Lambda}_{\Delta\mathbf{s}}) = \sum_{i=1}^{N}\mathrm{E}\left[\ln P\left(\hat{\mathbf{s}}_i|\mathbf{s}_i,\mathbf{z}_i,\theta_i\right)\right] \tag{B.59}$$

$$= \sum_{i=1}^{N}\sum_{\theta\in\{\mathcal{T},\mathcal{N}\}}\sum_{k=1}^{K} t_{i\theta}\left(\frac{1}{2}\gamma(z_{ik})\ln\left|\frac{\mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta}}}{2\pi}\right|\right.$$
$$\left.-\frac{1}{2}\mathrm{tr}\left(\mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta}}\mathrm{E}\left[z_{ik}\left(\hat{\mathbf{s}}_i-\left(\mathbf{s}_i+\mu_{\Delta\mathbf{s}_{k\theta}}\right)\right)\left(\hat{\mathbf{s}}_i-\left(\mathbf{s}_i+\mu_{\Delta\mathbf{s}_{k\theta}}\right)\right)^T\right]\right)\right) \tag{B.60}$$

$$= \frac{1}{2}\sum_{\theta\in\{\mathcal{T},\mathcal{N}\}}\sum_{k=1}^{K}\left(N_{\theta z_k}\ln\left|\frac{\mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta}}}{2\pi}\right|\right.$$
$$\left.-\mathrm{tr}\left(\mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta}}\sum_{i=1}^{N} t_{i\theta}\mathrm{E}\left[z_{ik}\left(\hat{\mathbf{s}}_i-\left(\mathbf{s}_i+\mu_{\Delta\mathbf{s}_{k\theta}}\right)\right)\left(\hat{\mathbf{s}}_i-\left(\mathbf{s}_i+\mu_{\Delta\mathbf{s}_{k\theta}}\right)\right)^T\right]\right)\right) \tag{B.61}$$

where we defined

$$\gamma(\theta_i,z_{ik}) = t_{i\theta}\gamma(z_{ik}) \tag{B.62}$$

$$N_{\theta z_k} = \sum_{i=1}^{N}\gamma(\theta_i,z_{ik}) \tag{B.63}$$

Deriving with respect to $\mu_{\Delta\mathbf{s}}$ and $\mathbf{\Lambda}_{\Delta\mathbf{s}}$, we obtain

$$\mu_{\Delta\mathbf{s}_{k\theta}} = \frac{1}{N_{\theta z_k}}\sum_{i=1}^{N}\left(\gamma(\theta_i,z_{ik})\hat{\mathbf{s}}_i - t_{i\theta}\mathrm{E}\left[z_{ik}\mathbf{s}_i\right]\right) \tag{B.64}$$

$$\mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta}}^{-1} = \frac{1}{N_{\theta z_k}}\sum_{i=1}^{N} t_{i\theta}\mathrm{E}\left[z_{ik}\left(\hat{\mathbf{s}}_i-\mathbf{s}_i-\mu_{\Delta\mathbf{s}_{k\theta}}\right)\left(\hat{\mathbf{s}}_i-\mathbf{s}_i-\mu_{\Delta\mathbf{s}_{k\theta}}\right)^T\right] \tag{B.65}$$

$$= \frac{1}{N_{\theta z_k}}\sum_{i=1}^{N} t_{i\theta}\mathrm{E}\left[z_{ik}\left(\hat{\mathbf{s}}_i-\mathbf{s}_i\right)\left(\hat{\mathbf{s}}_i-\mathbf{s}_i\right)^T\right] - \mu_{\Delta\mathbf{s}_{k\theta}}\mu_{\Delta\mathbf{s}_{k\theta}}^T \tag{B.66}$$

To complete the formulas, we need to compute the expectations:

$$\begin{aligned}
\mathrm{E}\left[\mathbf{s}_i\right] &= \sum_{k=1}^{K} \gamma(z_{ik}) \mu'_{\mathbf{s}_{ik\theta}} && \text{(B.67)}\\
\mathrm{E}\left[\mathbf{s}_i \mathbf{s}_i^T\right] &= \sum_{k=1}^{K} \gamma(z_{ik}) \left(\boldsymbol{\Lambda}'^{-1}_{\mathbf{s}_{k\theta}} + \mu'_{\mathbf{s}_{ik\theta}} \mu'^T_{\mathbf{s}_{ik\theta}}\right) && \text{(B.68)}\\
\mathrm{E}\left[z_{ik} \mathbf{s}_i\right] &= \gamma(z_{ik}) \mu'_{\mathbf{s}_{ik\theta}} && \text{(B.69)}\\
\mathrm{E}\left[z_{ik} \mathbf{s}_i \mathbf{s}_i^T\right] &= \gamma(z_{ik}) \left(\boldsymbol{\Lambda}'^{-1}_{\mathbf{s}_{k\theta}} + \mu'_{\mathbf{s}_{ik\theta}} \mu'^T_{\mathbf{s}_{ik\theta}}\right) && \text{(B.70)}\\
\mathrm{E}\left[z_{ik} \left(\hat{\mathbf{s}}_i - \mathbf{s}_i\right) \left(\hat{\mathbf{s}}_i - \mathbf{s}_i\right)^T\right] &= \gamma(z_{ik}) \left(\hat{\mathbf{s}}_i \hat{\mathbf{s}}_i^T - \hat{\mathbf{s}}_i \mu'^T_{\mathbf{s}_{ik\theta}} - \mu'_{\mathbf{s}_{ik\theta}} \hat{\mathbf{s}}_i^T + \boldsymbol{\Lambda}'^{-1}_{\mathbf{s}_{k\theta}} + \mu'_{\mathbf{s}_{ik\theta}} \mu'^T_{\mathbf{s}_{ik\theta}}\right) && \text{(B.71)}
\end{aligned}$$

Finally, we plug-in the expectations into (B.56), (B.58), (B.64) and (B.66) to obtain the final values of the parameters:

$$\begin{aligned}
\mu_{\mathbf{s}_\theta} &= \frac{1}{N_\theta} \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma(\theta_i, z_{ik}) \mu'_{\mathbf{s}_{ik\theta}} && \text{(B.72)}\\
\boldsymbol{\Lambda}^{-1}_{\mathbf{s}_\theta} &= \frac{1}{N_\theta} \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma(\theta_i, z_{ik}) \left(\boldsymbol{\Lambda}'^{-1}_{\mathbf{s}_{k\theta}} + \mu'_{\mathbf{s}_{ik\theta}} \mu'^T_{\mathbf{s}_{ik\theta}}\right) - \mu_{\mathbf{s}_\theta} \mu^T_{\mathbf{s}_\theta} && \text{(B.73)}\\
\mu_{\Delta\mathbf{s}_{k\theta}} &= \frac{1}{N_{\theta z_k}} \sum_{i=1}^{N} \gamma(\theta_i, z_{ik}) \left(\hat{\mathbf{s}}_i - \mu'_{\mathbf{s}_{ik\theta}}\right) && \text{(B.74)}\\
\boldsymbol{\Lambda}^{-1}_{\Delta\mathbf{s}_{k\theta}} &= \frac{1}{N_{\theta z_k}} \sum_{i=1}^{N} \gamma(\theta_i, z_{ik}) \left(\hat{\mathbf{s}}_i \hat{\mathbf{s}}_i^T - \hat{\mathbf{s}}_i \mu'^T_{\mathbf{s}_{ik\theta}} - \mu'_{\mathbf{s}_{ik\theta}} \hat{\mathbf{s}}_i^T + \mu'_{\mathbf{s}_{ik\theta}} \mu'^T_{\mathbf{s}_{ik\theta}}\right)\\
&\quad + \boldsymbol{\Lambda}'^{-1}_{\mathbf{s}_{k\theta}} - \mu_{\Delta\mathbf{s}_{k\theta}} \mu^T_{\Delta\mathbf{s}_{k\theta}} && \text{(B.75)}\\
&= \frac{1}{N_{\theta z_k}} \sum_{i=1}^{N} \gamma(\theta_i, z_{ik}) \left(\hat{\mathbf{s}}_i - \mu'_{\mathbf{s}_{ik\theta}}\right) \left(\hat{\mathbf{s}}_i - \mu'_{\mathbf{s}_{ik\theta}}\right)^T + \boldsymbol{\Lambda}'^{-1}_{\mathbf{s}_{k\theta}} - \mu_{\Delta\mathbf{s}_{k\theta}} \mu^T_{\Delta\mathbf{s}_{k\theta}} && \text{(B.76)}
\end{aligned}$$

B.3.4 Objective function

Here, we derive the EM auxiliary function to evaluate the convergence:

$$\mathcal{Q}(\mathcal{M}) = \sum_{i=1}^{N} \mathrm{E}\left[\ln P\left(\hat{\mathbf{s}}_i | \mathbf{s}_i, \mathbf{z}_i, \theta_i\right)\right] + \mathrm{E}\left[\ln P\left(\mathbf{Q}_i | \mathbf{z}_i\right)\right] + \mathrm{E}\left[\ln P\left(\mathbf{s}_i | \theta_i\right)\right] + \mathrm{E}\left[\ln P\left(\mathbf{z}_i\right)\right] \ . \quad \text{(B.77)}$$

The term $\sum_{i=1}^{N} \mathrm{E}\left[\ln P\left(\hat{\mathbf{s}}_i|\mathbf{s}_i, \mathbf{z}_i, \theta_i\right)\right]$:

$$\begin{aligned}
&\sum_{i=1}^{N} \mathrm{E}\left[\ln P\left(\hat{\mathbf{s}}_i|\mathbf{s}_i, \mathbf{z}_i, \theta_i\right)\right] \\
&= \sum_{\theta\in\{\mathcal{T},\mathcal{N}\}} \sum_{k=1}^{K} \frac{1}{2} N_{\theta z_k} \ln \left|\frac{\mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta}}}{2\pi}\right| \\
&- \frac{1}{2}\mathrm{tr}\left(\mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta}} \sum_{i=1}^{N} t_{i\theta}\mathrm{E}\left[z_{ik}\left(\hat{\mathbf{s}}_i - \left(\mathbf{s}_i + \mu_{\Delta\mathbf{s}_{k\theta}}\right)\right)\left(\hat{\mathbf{s}}_i - \left(\mathbf{s}_i + \mu_{\Delta\mathbf{s}_{k\theta}}\right)\right)^T\right]\right) \qquad \text{(B.78)}
\end{aligned}$$

$$\begin{aligned}
&= \sum_{\theta\in\{\mathcal{T},\mathcal{N}\}} \sum_{k=1}^{K} \frac{1}{2} N_{\theta z_k} \ln \left|\frac{\mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta}}}{2\pi}\right| \\
&- \frac{1}{2}\mathrm{tr}\left(\mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta}} \sum_{i=1}^{N} t_{i\theta}\mathrm{E}\left[z_{ik}\left(\left(\hat{\mathbf{s}}_i - \mathbf{s}_i\right) - \mu_{\Delta\mathbf{s}_{k\theta}}\right)\left(\left(\hat{\mathbf{s}}_i - \mathbf{s}_i\right) - \mu_{\Delta\mathbf{s}_{k\theta}}\right)^T\right]\right) \qquad \text{(B.79)}
\end{aligned}$$

$$\begin{aligned}
&= \sum_{\theta\in\{\mathcal{T},\mathcal{N}\}} \sum_{k=1}^{K} \frac{1}{2} N_{\theta z_k} \ln \left|\frac{\mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta}}}{2\pi}\right| \\
&- \frac{1}{2}\mathrm{tr}\left(\mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta}} \sum_{i=1}^{N} t_{i\theta}\left(\mathrm{E}\left[z_{ik}\left(\hat{\mathbf{s}}_i - \mathbf{s}_i\right)\left(\hat{\mathbf{s}}_i - \mathbf{s}_i\right)^T\right]\right.\right. \\
&\left.\left. -\mathrm{E}\left[z_{ik}\left(\hat{\mathbf{s}}_i - \mathbf{s}_i\right)\right]\mu_{\Delta\mathbf{s}_{k\theta}}^T - \mu_{\Delta\mathbf{s}_{k\theta}}\mathrm{E}\left[z_{ik}\left(\hat{\mathbf{s}}_i - \mathbf{s}_i\right)\right]^T + \gamma(z_{ik})\mu_{\Delta\mathbf{s}_{k\theta}}\mu_{\Delta\mathbf{s}_{k\theta}}^T\right)\right) \qquad \text{(B.80)}
\end{aligned}$$

$$\begin{aligned}
&= \sum_{\theta\in\{\mathcal{T},\mathcal{N}\}} \sum_{k=1}^{K} \frac{1}{2} N_{\theta z_k} \ln \left|\frac{\mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta}}}{2\pi}\right| \\
&- \frac{1}{2}\mathrm{tr}\left(\mathbf{\Lambda}_{\Delta\mathbf{s}_{k\theta}} \sum_{i=1}^{N} \gamma(\theta_i, z_{ik})\left(\left(\hat{\mathbf{s}}_i - \mu'_{\mathbf{s}_{ik\theta}}\right)\left(\hat{\mathbf{s}}_i - \mu'_{\mathbf{s}_{ik\theta}}\right)^T + \mathbf{\Lambda}'^{-1}_{\mathbf{s}_{k\theta}}\right.\right. \\
&\left.\left. - \left(\hat{\mathbf{s}}_i - \mu'_{\mathbf{s}_{ik\theta}}\right)\mu_{\Delta\mathbf{s}_{k\theta}}^T - \mu_{\Delta\mathbf{s}_{k\theta}}\left(\hat{\mathbf{s}}_i - \mu'_{\mathbf{s}_{ik\theta}}\right) + \mu_{\Delta\mathbf{s}_{k\theta}}\mu_{\Delta\mathbf{s}_{k\theta}}^T\right)\right) \ . \qquad \text{(B.81)}
\end{aligned}$$

The term $\sum_{i=1}^{N} \mathrm{E}\left[\ln P\left(\mathbf{Q}_i|\mathbf{z}_i\right)\right]$:

$$\begin{aligned}
\sum_{i=1}^{N} \mathrm{E}\left[\ln P\left(\mathbf{Q}_i|\mathbf{z}_i\right)\right] =& \sum_{p=1}^{P}\sum_{k=1}^{K} \frac{1}{2} N_{z_k} \ln \left|\frac{\mathbf{\Lambda}_{\mathbf{Q}_{pk}}}{2\pi}\right| \\
&- \frac{1}{2}\mathrm{tr}\left(\mathbf{\Lambda}_{\mathbf{Q}_{pk}} \sum_{i=1}^{N} \gamma(z_{ik})\left(\mathbf{Q}_{ip} - \mu_{\mathbf{Q}_{pk}}\right)\left(\mathbf{Q}_{ip} - \mu_{\mathbf{Q}_{pk}}\right)^T\right) \ . \qquad \text{(B.82)}
\end{aligned}$$

The term $\sum_{i=1}^{N} \mathrm{E}\left[\ln P\left(\mathbf{s}_i|\theta_i\right)\right]$

$$\begin{aligned}\sum_{i=1}^{N} \mathrm{E}\left[\ln P\left(\mathbf{s}_i|\theta_i\right)\right] =& \sum_{\theta\in\{\mathcal{T},\mathcal{N}\}} \frac{1}{2}N_\theta \ln\left|\frac{\mathbf{\Lambda}_{\mathbf{s}_\theta}}{2\pi}\right| \\ &-\frac{1}{2}\mathrm{tr}\left(\mathbf{\Lambda}_{\mathbf{s}_\theta}\sum_{i=1}^{N} t_{i\theta}\mathrm{E}\left[\left(\mathbf{s}_i-\mu_{\mathbf{s}_\theta}\right)\left(\mathbf{s}_i-\mu_{\mathbf{s}_\theta}\right)^T\right]\right) \qquad \text{(B.83)}\\ =& \sum_{\theta\in\{\mathcal{T},\mathcal{N}\}} \frac{1}{2}N_\theta \ln\left|\frac{\mathbf{\Lambda}_{\mathbf{s}_\theta}}{2\pi}\right| \\ &-\frac{1}{2}\mathrm{tr}\left(\mathbf{\Lambda}_{\mathbf{s}_\theta}\sum_{i=1}^{N} t_{i\theta}\left(\mathrm{E}\left[\mathbf{s}_i\mathbf{s}_i^T\right]-\mathrm{E}\left[\mathbf{s}_i\right]\mu_{\mathbf{s}_\theta}^T-\mu_{\mathbf{s}_\theta}\mathrm{E}\left[\mathbf{s}_i\right]^T+\mu_{\mathbf{s}_\theta}\mu_{\mathbf{s}_\theta}^T\right)\right) . \qquad \text{(B.84)}\end{aligned}$$

And finally, the term $\sum_{i=1}^{N} \mathrm{E}\left[\ln P\left(\mathbf{z}_i\right)\right]$:

$$\sum_{i=1}^{N} \mathrm{E}\left[\ln P\left(\mathbf{z}_i\right)\right] = \sum_{k=1}^{K} N_{z_k}\ln\pi_{z_k} . \qquad \text{(B.85)}$$

B.3.5 Case with s observed and z hidden

Now, we assume that only $\mathbf{z}$ is hidden and $\mathbf{s}$ and θ are known.

B.3.6 E-step

We compute the posterior of the hidden variable $P\left(\mathbf{z}|\hat{\mathbf{s}},\mathbf{s},\mathbf{Q},\theta\right)$

$$\begin{aligned}P\left(z_k=1|\hat{\mathbf{s}},\mathbf{s},\mathbf{Q},\theta\right) =& \frac{P\left(z_k=1,\hat{\mathbf{s}},\mathbf{s},\mathbf{Q}|\theta\right)}{P\left(\hat{\mathbf{s}},\mathbf{s},\mathbf{Q}|\theta\right)} \qquad \text{(B.86)}\\ =& \frac{\pi_{z_k}P\left(\hat{\mathbf{s}}|\mathbf{s},\theta,z_k=1\right)P\left(\mathbf{s}|\theta\right)P\left(\mathbf{Q}|z_k=1\right)}{\sum_{k=1}^{K}\pi_{z_k}P\left(\hat{\mathbf{s}}|\mathbf{s},\theta,z_k=1\right)P\left(\mathbf{s}|\theta\right)P\left(\mathbf{Q}|z_k=1\right)} . \qquad \text{(B.87)}\end{aligned}$$

We define

$$\gamma(z_k) = P\left(z_k=1|\hat{\mathbf{s}},\mathbf{s},\mathbf{Q},\theta\right) . \qquad \text{(B.88)}$$

B.3.7 M-step

Again, we maximize the EM auxiliary function:

$$\mathcal{Q}(\mathcal{M}) = \sum_{i=1}^{N} \mathrm{E}\left[\ln P\left(\hat{\mathbf{s}}_i|\mathbf{s}_i,\mathbf{z}_i,\theta_i\right)\right]+\mathrm{E}\left[\ln P\left(\mathbf{Q}_i|\mathbf{z}_i\right)\right]+\mathrm{E}\left[\ln P\left(\mathbf{s}_i|\theta_i\right)\right]+\mathrm{E}\left[\ln P\left(\mathbf{z}_i\right)\right] \qquad \text{(B.89)}$$

The equations for $\pi_{\mathbf{z}}$, $\mu_{\mathbf{Q}_{pk}}$ and $\mathbf{\Lambda}_{\mathbf{Q}_{pk}}$ are the same as for the general case. Thus we have:

$$\pi_{z_k} = \frac{N_{z_k}}{\sum_{k=1}^{K} N_{z_k}} \qquad \text{(B.90)}$$

where

$$N_{z_k} = \sum_{i=1}^{N} \gamma(z_{ik}) \ , \tag{B.91}$$

$$\mu_{\mathbf{Q}_{pk}} = \frac{1}{N_{z_k}} \sum_{i=1}^{N} \gamma(z_{ik}) \mathbf{Q}_{ip} \tag{B.92}$$

$$\boldsymbol{\Lambda}_{\mathbf{Q}_{pk}}^{-1} = \frac{1}{N_{z_k}} \sum_{i=1}^{N} \gamma(z_{ik}) \left(\mathbf{Q}_{ip} - \mu_{\mathbf{Q}_{pk}}\right) \left(\mathbf{Q}_{ip} - \mu_{\mathbf{Q}_{pk}}\right)^T \tag{B.93}$$

We maximize with respect to $\mu_{\mathbf{s}}$ and $\boldsymbol{\Lambda}_{\mathbf{s}}$ to obtain:

$$\mu_{\mathbf{s}_\theta} = \frac{1}{N_\theta} \sum_{i=1}^{N} t_{i\theta} \mathbf{s}_i \tag{B.94}$$

$$\boldsymbol{\Lambda}_{\mathbf{s}_\theta}^{-1} = \frac{1}{N_\theta} \sum_{i=1}^{N} t_{i\theta} \left(\mathbf{s}_i - \mu_{\mathbf{s}_\theta}\right) \left(\mathbf{s}_i - \mu_{\mathbf{s}_\theta}\right)^T \tag{B.95}$$

$$= \frac{1}{N_\theta} \sum_{i=1}^{N} t_{i\theta} \mathbf{s}_i \mathbf{s}_i^T - \mu_{\mathbf{s}_\theta} \mu_{\mathbf{s}_\theta}^T \tag{B.96}$$

We write the terms of $\mathcal{Q}$ depending on $\mu_{\Delta \mathbf{s}}$ and $\boldsymbol{\Lambda}_{\Delta \mathbf{s}}$ as:

$$\mathcal{Q}(\mu_{\Delta \mathbf{s}}, \boldsymbol{\Lambda}_{\Delta \mathbf{s}}) = \sum_{i=1}^{N} \mathrm{E}\left[\ln P\left(\hat{\mathbf{s}}_i | \mathbf{s}_i, \mathbf{z}_i, \theta_i\right)\right] \tag{B.97}$$

$$\begin{aligned} = & \sum_{i=1}^{N} \sum_{\theta \in \{\mathcal{T}, \mathcal{N}\}} \sum_{k=1}^{K} t_{i\theta} \left(\frac{1}{2} \gamma(z_{ik}) \ln \left| \frac{\boldsymbol{\Lambda}_{\Delta \mathbf{s}_{k\theta}}}{2\pi} \right| \right. \\ & \left. - \frac{1}{2} \mathrm{tr} \left(\boldsymbol{\Lambda}_{\Delta \mathbf{s}_{k\theta}} \gamma(z_{ik}) \left(\hat{\mathbf{s}}_i - (\mathbf{s}_i + \mu_{\Delta \mathbf{s}_{k\theta}})\right) \left(\hat{\mathbf{s}}_i - (\mathbf{s}_i + \mu_{\Delta \mathbf{s}_{k\theta}})\right)^T \right) \right) \end{aligned} \tag{B.98}$$

$$\begin{aligned} = & \frac{1}{2} \sum_{\theta \in \{\mathcal{T}, \mathcal{N}\}} \sum_{k=1}^{K} \left(N_{\theta z_k} \ln \left| \frac{\boldsymbol{\Lambda}_{\Delta \mathbf{s}_{k\theta}}}{2\pi} \right| \right. \\ & \left. - \mathrm{tr} \left(\boldsymbol{\Lambda}_{\Delta \mathbf{s}_{k\theta}} \sum_{i=1}^{N} \gamma(\theta_i, z_{ik}) \left(\hat{\mathbf{s}}_i - (\mathbf{s}_i + \mu_{\Delta \mathbf{s}_{k\theta}})\right) \left(\hat{\mathbf{s}}_i - (\mathbf{s}_i + \mu_{\Delta \mathbf{s}_{k\theta}})\right)^T \right) \right) \end{aligned} \tag{B.99}$$

and maximize to obtain:

$$\mu_{\Delta \mathbf{s}_{k\theta}} = \frac{1}{N_{\theta z_k}} \sum_{i=1}^{N} \gamma(\theta_i, z_{ik}) \left(\hat{\mathbf{s}}_i - \mathbf{s}_i\right) \tag{B.100}$$

$$\boldsymbol{\Lambda}_{\Delta \mathbf{s}_{k\theta}}^{-1} = \frac{1}{N_{\theta z_k}} \sum_{i=1}^{N} \gamma(\theta_i, z_{ik}) \left(\hat{\mathbf{s}}_i - \mathbf{s}_i - \mu_{\Delta \mathbf{s}_{k\theta}}\right) \left(\hat{\mathbf{s}}_i - \mathbf{s}_i - \mu_{\Delta \mathbf{s}_{k\theta}}\right)^T \tag{B.101}$$

$$= \frac{1}{N_{\theta z_k}} \sum_{i=1}^{N} \gamma(\theta_i, z_{ik}) \left(\hat{\mathbf{s}}_i - \mathbf{s}_i\right) \left(\hat{\mathbf{s}}_i - \mathbf{s}_i\right)^T - \mu_{\Delta \mathbf{s}_{k\theta}} \mu_{\Delta \mathbf{s}_{k\theta}}^T \ . \tag{B.102}$$

B.3.8 Objective function

The EM auxiliary to evaluate convergence is:

$$\mathcal{Q}(\mathcal{M}) = \sum_{i=1}^{N} \mathrm{E}\left[\ln P\left(\hat{\mathbf{s}}_i|\mathbf{s}_i, \mathbf{z}_i, \theta_i\right)\right] + \mathrm{E}\left[\ln P\left(\mathbf{Q}_i|\mathbf{z}_i\right)\right] + \mathrm{E}\left[\ln P\left(\mathbf{s}_i|\theta_i\right)\right] + \mathrm{E}\left[\ln P\left(\mathbf{z}_i\right)\right] \quad \text{(B.103)}$$

The terms 2 and 4 are the same as for the general case and are given by (B.82) and (B.85). The rest of terms are different.

The term $\sum_{i=1}^{N} \mathrm{E}\left[\ln P\left(\hat{\mathbf{s}}_i|\mathbf{s}_i, \mathbf{z}_i, \theta_i\right)\right]$ is

$$\begin{aligned}\sum_{i=1}^{N} \mathrm{E}\left[\ln P\left(\hat{\mathbf{s}}_i|\mathbf{s}_i, \mathbf{z}_i, \theta_i\right)\right] = &\sum_{\theta\in\{\mathcal{T},\mathcal{N}\}} \sum_{k=1}^{K} \frac{1}{2} N_{\theta z_k} \ln \left|\frac{\mathbf{\Lambda}_{\Delta \mathbf{s}_{k\theta}}}{2\pi}\right| \\ &- \frac{1}{2}\mathrm{tr}\left(\mathbf{\Lambda}_{\Delta \mathbf{s}_{k\theta}} \sum_{i=1}^{N} \gamma(\theta_i, z_{ik}) \left(\left(\hat{\mathbf{s}}_i - \mathbf{s}_i\right) - \mu_{\Delta \mathbf{s}_{k\theta}}\right) \left(\left(\hat{\mathbf{s}}_i - \mathbf{s}_i\right) - \mu_{\Delta \mathbf{s}_{k\theta}}\right)^T\right) \ .\end{aligned} \quad \text{(B.104)}$$

And the term $\sum_{i=1}^{N} \mathrm{E}\left[\ln P\left(\mathbf{s}_i|\theta_i\right)\right]$ is

$$\begin{aligned}\sum_{i=1}^{N} \mathrm{E}\left[\ln P\left(\mathbf{s}_i|\theta_i\right)\right] = &\sum_{\theta\in\{\mathcal{T},\mathcal{N}\}} \frac{1}{2} N_\theta \ln \left|\frac{\mathbf{\Lambda}_{\mathbf{s}_\theta}}{2\pi}\right| \\ &- \frac{1}{2}\mathrm{tr}\left(\mathbf{\Lambda}_{\mathbf{s}_\theta} \sum_{i=1}^{N} t_{i\theta} \left(\mathbf{s}_i - \mu_{\mathbf{s}_\theta}\right) \left(\mathbf{s}_i - \mu_{\mathbf{s}_\theta}\right)^T\right) \ .\end{aligned} \quad \text{(B.105)}$$

Appendix C

EM for PLDA

C.1 Introduction

Probabilistic linear discriminant analysis [Prince and Elder, 2007] is a linear generative model commonly used to describe the distribution of the speakers' observations (i-vector). The model decomposes each i-vector into a speaker dependent term and a channel term. The channel term represents the inter-session variability between different observations of the same speaker. While i-vectors are observed variables, the speaker and channel terms are hidden. As in other models with hidden variables like GMM or HMM, we apply the expectation-maximization algorithm (EM) to estimate the parameters of the model. In this appendix, we derive the equations of the EM-algorithm needed to train the multiple flavors of PLDA employed in this thesis. Models differ in the assumptions made about the prior distributions of speaker and channel spaces. The two-covariance model assumes that both the speaker and channel spaces are of full-rank. The simplified PLDA model (SPLDA) assumes that the speaker space is low-rank. Finally, the full PLDA model complicates SPLDA by restricting the channel covariance to have the form $\mathbf{U}\mathbf{U}^T + \mathbf{D}$ where $\mathbf{U}$ is a low-rank matrix and $\mathbf{D}$ a diagonal matrix.

C.2 Definition of Sufficient Statistics

Let's assume that we are given M speakers with N_i observations by speaker. We denote by ϕ_{ij} the j^{th} observation of the speaker i and by $\mathbf{\Phi}_i$ the set of all the observations of i. We define some sufficient statistics that simplify the computations. The first-order and second-order statistics for speaker i are:

$$\mathbf{F}_i = \sum_{j=1}^{N_i} \phi_{ij} \tag{C.1}$$

$$\mathbf{S}_i = \sum_{j=1}^{N_i} \phi_{ij}\phi_{ij}^T \tag{C.2}$$

and the centered statistics are:

$$\overline{\mathbf{F}}_i = \mathbf{F}_i - N_i\mu \tag{C.3}$$

$$\overline{\mathbf{S}}_i = \sum_{j=1}^{N_i} (\phi_{ij} - \mu)(\phi_{ij} - \mu)^T = \mathbf{S}_i - \mu\mathbf{F}_i^T - \mathbf{F}_i\mu^T + N_i\mu\mu^T \tag{C.4}$$

where μ is a speaker and channel independent mean of the observations.

We also define the global statistics as:

$$N = \sum_{i=1}^{M} N_i \tag{C.5}$$

$$\mathbf{F} = \sum_{i=1}^{M} \mathbf{F}_i \qquad \overline{\mathbf{F}} = \sum_{i=1}^{M} \overline{\mathbf{F}}_i \tag{C.6}$$

$$\mathbf{S} = \sum_{i=1}^{M} \mathbf{S}_i \qquad \overline{\mathbf{S}} = \sum_{i=1}^{M} \overline{\mathbf{S}}_i\ . \tag{C.7}$$

C.3 Two-Covariance Model

C.3.1 Model definition

The two-covariance model, also called full-rank PLDA, assumes that an i-vector ϕ_{ij} of the session j of speaker i can be written as:

$$\phi_{ij} = \mathbf{y}_i + \epsilon_{ij} \tag{C.8}$$

where $\mathbf{y}_i$ is the speaker identity variable and ϵ_{ij} is a channel offset.

The following priors are assumed for the speaker and channel distributions:

$$\mathbf{y}_i \sim \mathcal{N}\left(\mathbf{y}_i|\mu, \mathbf{B}^{-1}\right) \tag{C.9}$$

$$\epsilon_{ij} \sim \mathcal{N}\left(\epsilon_{ij}|\mathbf{0}, \mathbf{W}^{-1}\right) \tag{C.10}$$

where $\mathcal{N}$ denotes the Gaussian distribution; μ is the speakers mean; $\mathbf{B}^{-1}$ is the between class covariance matrix, $\mathbf{W}^{-1}$ is the within class covariance matrix; and $\mathbf{B}$ and $\mathbf{W}$ are the precision matrices. We denote by $\mathcal{M} = (\mu, \mathbf{B}, \mathbf{W})$ the set of all the parameters of the model.

C.3.2 EM algorithm

C.3.2.1 E-step

In the E-step we calculate the posterior distribution of the hidden variables $\mathbf{y}_i$:

$$P\left(\mathbf{y}_i|\mathbf{\Phi}_i, \mathcal{M}\right) = \frac{P\left(\mathbf{\Phi}_i|\mathbf{y}_i, \mathbf{W}\right) P\left(\mathbf{y}_i|\mu, \mathbf{B}\right)}{P\left(\mathbf{\Phi}_i|\mathcal{M}\right)} \tag{C.11}$$

so

$$\begin{aligned}
\ln P\left(\mathbf{y}_i|\mathbf{\Phi}_i,\mathcal{M}\right) &= \sum_{j=1}^{N_i} \ln P\left(\phi_{ij}|\mathbf{y}_i,\mathbf{W}\right) + \ln P\left(\mathbf{y}_i|\mu,\mathbf{B}\right) + \text{const} && \text{(C.12)}\\
&= -\frac{1}{2}\sum_{j=1}^{N_i}(\phi_{ij}-\mathbf{y}_i)^T\mathbf{W}(\phi_{ij}-\mathbf{y}_i) - \frac{1}{2}(\mathbf{y}_i-\mu)^T\mathbf{B}(\mathbf{y}_i-\mu) + \text{const} && \text{(C.13)}\\
&= \mathbf{y}_i^T\mathbf{W}\mathbf{F}_i - \frac{1}{2}N_i\mathbf{y}_i^T\mathbf{W}\mathbf{y}_i - \frac{1}{2}\mathbf{y}_i^T\mathbf{B}\mathbf{y}_i + \mathbf{y}_i^T\mathbf{B}\mu + \text{const} && \text{(C.14)}\\
&= -\frac{1}{2}\mathbf{y}_i^T\left(\mathbf{B}+N_i\mathbf{W}\right)\mathbf{y}_i + \mathbf{y}_i^T\left(\mathbf{B}\mu+\mathbf{W}\mathbf{F}_i\right) + \text{const}\ . && \text{(C.15)}
\end{aligned}$$

Equation (C.15) has the form of a Gaussian distribution so:

$$\begin{aligned}
P\left(\mathbf{y}_i|\mathbf{\Phi}_i,\mathcal{M}\right) &= \mathcal{N}\left(\mathbf{y}_i|\mathbf{L}_i^{-1}\gamma_i,\mathbf{L}_i^{-1}\right) && \text{(C.16)}\\
\mathbf{L}_i &= \mathbf{B}+N_i\mathbf{W} && \text{(C.17)}\\
\gamma_i &= \mathbf{B}\mu+\mathbf{W}\mathbf{F}_i\ . && \text{(C.18)}
\end{aligned}$$

C.3.2.2 M-step

In the M-step, we maximize the auxiliary function $\mathcal{Q}(\mathcal{M})$:

$$\begin{aligned}
\mathcal{Q}(\mathcal{M}) &= \sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[\ln P\left(\mathbf{\Phi}_i,\mathbf{y}_i|\mathcal{M}\right)\right] && \text{(C.19)}\\
&= \sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[\ln P\left(\mathbf{\Phi}_i|\mathbf{y}_i,\mathbf{W}\right)\right] + \mathrm{E}_{\mathbf{Y}}\left[\ln P\left(\mathbf{y}_i|\mu,\mathbf{B}\right)\right] && \text{(C.20)}\\
&= \mathcal{Q}(\mathbf{W}) + \mathcal{Q}(\mu,\mathbf{B})\ . && \text{(C.21)}
\end{aligned}$$

From (C.9), the term $\mathcal{Q}(\mu,\mathbf{B})$ is

$$\mathcal{Q}(\mu,\mathbf{B}) = \frac{M}{2}\ln|\mathbf{B}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{B}\sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}\left[\left(\mathbf{y}_i-\mu\right)\left(\mathbf{y}_i-\mu\right)^T\right]\right) + \text{const}\ . \qquad \text{(C.22)}$$

We derive $\mathcal{Q}$ with respect to μ:

$$\begin{aligned}
\frac{\partial\mathcal{Q}(\mu,\mathbf{B})}{\partial\mu} &= \frac{1}{2}\sum_{i=1}^{M}\mathbf{B}\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i-\mu\right] = \mathbf{0} \quad \Longrightarrow && \text{(C.23)}\\
\mu &= \frac{1}{M}\sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]\ . && \text{(C.24)}
\end{aligned}$$

We derive $\mathcal{Q}$ with respect to $\mathbf{B}$:

$$\frac{\partial\mathcal{Q}(\mu,\mathbf{B})}{\partial\mathbf{B}} = \frac{M}{2}\left(2\mathbf{B}^{-1}-\mathrm{diag}(\mathbf{B}^{-1})\right) - \frac{1}{2}\left(2\mathbf{K}-\mathrm{diag}(\mathbf{K})\right) = \mathbf{0} \qquad \text{(C.25)}$$

where $\mathbf{K} = \sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[(\mathbf{y}_i - \mu)(\mathbf{y}_i - \mu)^T\right]$, so

$$\mathbf{B}^{-1} = \frac{1}{M}\sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[(\mathbf{y}_i - \mu)(\mathbf{y}_i - \mu)^T\right] = \frac{1}{M}\sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right] - \mu\mu^T \ . \tag{C.26}$$

From (C.10), the term $\mathcal{Q}(\mathbf{W})$ is

$$\mathcal{Q}(\mathbf{W}) = \frac{N}{2}\ln|\mathbf{W}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{W}\sum_{i=1}^{M}\sum_{j=1}^{N_i}\mathrm{E}_{\mathbf{Y}}\left[(\phi_{ij} - \mathbf{y}_i)(\phi_{ij} - \mathbf{y}_i)^T\right]\right) + \mathrm{const}\ . \tag{C.27}$$

We derive $\mathcal{Q}$ with respect to $\mathbf{W}$:

$$\frac{\partial\mathcal{Q}(\mathbf{W})}{\partial\mathbf{W}} = \mathbf{0} \implies \tag{C.28}$$

$$\mathbf{W}^{-1} = \frac{1}{N}\sum_{i=1}^{M}\sum_{j=1}^{N_i}\mathrm{E}_{\mathbf{Y}}\left[(\phi_{ij} - \mathbf{y}_i)(\phi_{ij} - \mathbf{y}_i)^T\right] \tag{C.29}$$

$$= \frac{1}{N}\left(\mathbf{S} - \sum_{i=1}^{M}\mathbf{F}_i\mathrm{E}_{\mathbf{Y}}[\mathbf{y}_i]^T - \sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}[\mathbf{y}_i]\,\mathbf{F}_i^T + \sum_{i=1}^{M}N_i\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right]\right)\ . \tag{C.30}$$

C.3.2.3 Objective function

We can check the convergence by computing the marginal likelihood of the data in each iteration of the EM-algorithm. For a given speaker i, by applying Bayes rule, it can be written as

$$P(\mathbf{\Phi}_i|\mathcal{M}) = \frac{P(\mathbf{\Phi}_i|\mathbf{y}_0,\mathcal{M})\,P(\mathbf{y}_0)}{P(\mathbf{y}_0|\mathbf{\Phi}_i,\mathcal{M})} \tag{C.31}$$

where $\mathbf{y}_0$ can adopt whatever value that we decide as long as the denominator is not zero. By taking (C.9), (C.10) and (C.16); $\mathbf{y}_0 = \mathrm{E}_{\mathbf{Y}}[\mathbf{y}_i]$; and summing for all the speakers, the total likelihood of the data is:

$$\begin{aligned}\ln P(\mathbf{\Phi}|\theta,\mathcal{M}) = &-\frac{Nd}{2}\ln(2\pi) + \frac{N}{2}\ln|\mathbf{W}| - \frac{1}{2}\mathrm{tr}(\mathbf{W}\mathbf{C}_{\mathbf{W}}) \\ &+ \frac{M}{2}\ln|\mathbf{B}| - \frac{1}{2}\mathrm{tr}(\mathbf{B}\mathbf{C}_{\mathbf{B}}) - \frac{1}{2}\sum_{i=1}^{M}\ln|\mathbf{L}_i|\end{aligned} \tag{C.32}$$

where

$$\mathbf{C}_{\mathbf{B}} = \sum_{i=1}^{M}(\mathrm{E}_{\mathbf{Y}}[\mathbf{y}_i] - \mu)(\mathrm{E}_{\mathbf{Y}}[\mathbf{y}_i] - \mu)^T \tag{C.33}$$

$$= \sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}[\mathbf{y}_i]\,\mathrm{E}_{\mathbf{Y}}[\mathbf{y}_i]^T - \left(\sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}[\mathbf{y}_i]\right)\mu^T - \mu\left(\sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}[\mathbf{y}_i]\right)^T + M\mu\mu^T \tag{C.34}$$

and

$$\mathbf{C_W} = \sum_{i=1}^{M}\sum_{j=1}^{N_i} \left(\phi_{ij} - \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]\right)\left(\phi_{ij} - \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]\right)^T \tag{C.35}$$

$$= \mathbf{S} - \sum_{i=1}^{M} \mathbf{F}_i \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]^T - \sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right] \mathbf{F}_i^T + \sum_{i=1}^{M} N_i \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right] \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]^T \ . \tag{C.36}$$

C.4 Simplified PLDA

C.4.1 Model definition

For simplified PLDA (SPLDA), an i-vector ϕ_{ij} of speaker i is written as:

$$\phi_{ij} = \mu + \mathbf{V}\mathbf{y}_i + \epsilon_{ij} \tag{C.37}$$

where μ is a speaker independent term, $\mathbf{V}$ is a low-rank eigen-voices matrix, $\mathbf{y}_i$ is the speaker factor vector, and ϵ_{ij} is a channel offset.

We assume the following priors for the variables:

$$\mathbf{y}_i \sim \mathcal{N}\left(\mathbf{y}_i|\mathbf{0}, \mathbf{I}\right) \tag{C.38}$$

$$\epsilon_{ij} \sim \mathcal{N}\left(\epsilon_{ij}|\mathbf{0}, \mathbf{W}^{-1}\right) \tag{C.39}$$

where $\mathcal{N}$ denotes the Gaussian distribution; and $\mathbf{W}$ is a full within class precision matrix. The set of all the model parameters is denoted by $\mathcal{M} = (\mu, \mathbf{V}, \mathbf{W})$.

C.4.2 Data conditional likelihood

The likelihood of the data given the hidden variables, for speaker i, is

$$\ln P\left(\mathbf{\Phi}_i|\mathbf{y}_i, \mathcal{M}\right) = \sum_{j=1}^{N_i} \ln \mathcal{N}\left(\phi_{ij}|\mu + \mathbf{V}\mathbf{y}_i, \mathbf{W}^{-1}\right) \tag{C.40}$$

$$= \frac{N_i}{2} \ln \left|\frac{\mathbf{W}}{2\pi}\right| - \frac{1}{2}\sum_{j=1}^{N_i} (\phi_{ij} - \mu - \mathbf{V}\mathbf{y}_i)^T \mathbf{W} (\phi_{ij} - \mu - \mathbf{V}\mathbf{y}_i) \tag{C.41}$$

$$= \frac{N_i}{2} \ln \left|\frac{\mathbf{W}}{2\pi}\right| - \frac{1}{2}\mathrm{tr}\left(\mathbf{W}\overline{\mathbf{S}}_i\right) + \mathbf{y}_i^T \mathbf{V}^T \mathbf{W} \overline{\mathbf{F}}_i - \frac{N_i}{2} \mathbf{y}_i^T \mathbf{V}^T \mathbf{W} \mathbf{V} \mathbf{y}_i \tag{C.42}$$

We can write this likelihood in another form, useful for the M-step, by defining:

$$\tilde{\mathbf{y}}_i = \begin{bmatrix}\mathbf{y}_i \\ 1\end{bmatrix}, \quad \tilde{\mathbf{V}} = \begin{bmatrix}\mathbf{V} & \mu\end{bmatrix} \ . \tag{C.43}$$

Then, we obtain

$$
\begin{aligned}
\ln P\left(\mathbf{\Phi}_i|\mathbf{y}_i,\mathcal{M}\right) =&\sum_{j=1}^{N_i}\ln\mathcal{N}\left(\phi_{ij}|\tilde{\mathbf{V}}\tilde{\mathbf{y}}_i,\mathbf{W}^{-1}\right) && \text{(C.44)}\\
=&\frac{N_i}{2}\ln\left|\frac{\mathbf{W}}{2\pi}\right|-\frac{1}{2}\sum_{j=1}^{N_i}(\phi_{ij}-\tilde{\mathbf{V}}\tilde{\mathbf{y}}_i)^T\mathbf{W}(\phi_{ij}-\tilde{\mathbf{V}}\tilde{\mathbf{y}}_i) && \text{(C.45)}\\
=&\frac{N_i}{2}\ln\left|\frac{\mathbf{W}}{2\pi}\right|-\frac{1}{2}\mathrm{tr}\left(\mathbf{W}\mathbf{S}_i\right)+\tilde{\mathbf{y}}_i^T\tilde{\mathbf{V}}^T\mathbf{W}\mathbf{F}_i-\frac{N_i}{2}\tilde{\mathbf{y}}_i^T\tilde{\mathbf{V}}^T\mathbf{W}\tilde{\mathbf{V}}\tilde{\mathbf{y}}_i && \text{(C.46)}\\
=&\frac{N_i}{2}\ln\left|\frac{\mathbf{W}}{2\pi}\right|-\frac{1}{2}\mathrm{tr}\left(\mathbf{W}\left(\mathbf{S}_i-2\mathbf{F}_i\tilde{\mathbf{y}}_i^T\tilde{\mathbf{V}}^T+N_i\tilde{\mathbf{V}}\tilde{\mathbf{y}}_i\tilde{\mathbf{y}}_i^T\tilde{\mathbf{V}}^T\right)\right)\;. && \text{(C.47)}
\end{aligned}
$$

C.4.3 Posterior of the hidden variables

The posterior of $\mathbf{y}$ is given by

$$
P\left(\mathbf{y}_i|\mathbf{\Phi}_i,\mathcal{M}\right)=\frac{P\left(\mathbf{\Phi}_i|\mathbf{y}_i,\mathcal{M}\right)P\left(\mathbf{y}_i\right)}{P\left(\mathbf{\Phi}_i|\mathcal{M}\right)}\;. \qquad \text{(C.48)}
$$

By using (C.38) and (C.42), we obtain:

$$
\begin{aligned}
\ln P\left(\mathbf{y}_i|\mathbf{\Phi}_i,\mathcal{M}\right) =&\ln P\left(\mathbf{\Phi}_i|\mathbf{y}_i,\mathcal{M}\right)+\ln P\left(\mathbf{y}_i\right)+\mathrm{const} && \text{(C.49)}\\
=&\mathbf{y}_i^T\mathbf{V}^T\mathbf{W}\overline{\mathbf{F}}_i-\frac{N_i}{2}\mathbf{y}_i^T\mathbf{V}^T\mathbf{W}\mathbf{V}\mathbf{y}_i-\frac{1}{2}\mathbf{y}_i^T\mathbf{y}_i+\mathrm{const} && \text{(C.50)}\\
=&\mathbf{y}_i^T\mathbf{V}^T\mathbf{W}\overline{\mathbf{F}}_i-\frac{1}{2}\mathbf{y}_i^T\left(\mathbf{I}+N_i\mathbf{V}^T\mathbf{W}\mathbf{V}\right)\mathbf{y}_i+\mathrm{const}\;. && \text{(C.51)}
\end{aligned}
$$

Equation (C.51) has the form of a Gaussian distribution:

$$
P\left(\mathbf{y}_i|\mathbf{\Phi}_i,\mathcal{M}\right)=\mathcal{N}\left(\mathbf{y}_i|\mathbf{L}_i^{-1}\gamma_i,\mathbf{L}_i^{-1}\right) \qquad \text{(C.52)}
$$

where

$$
\begin{aligned}
\mathbf{L}_i =&\mathbf{I}+N_i\mathbf{V}^T\mathbf{W}\mathbf{V} && \text{(C.53)}\\
\gamma_i =&\mathbf{V}^T\mathbf{W}\overline{\mathbf{F}}_i\;. && \text{(C.54)}
\end{aligned}
$$

C.4.4 Marginal likelihood of the data

The marginal likelihood of the data is

$$
P\left(\mathbf{\Phi}_i|\mathcal{M}\right)=\frac{P\left(\mathbf{\Phi}_i|\mathbf{y}_0,\mathcal{M}\right)P\left(\mathbf{y}_0\right)}{P\left(\mathbf{y}_0|\mathbf{\Phi}_i,\mathcal{M}\right)} \qquad \text{(C.55)}
$$

where we can plug-in whatever $\mathbf{y}_0$ as the denominator is not zero. By taking (C.42), (C.38) and (C.52); and $\mathbf{y}_0=\mathbf{0}$, we obtain:

$$
\ln P\left(\mathbf{\Phi}_i|\mathcal{M}\right)=\frac{N_i}{2}\ln\left|\frac{\mathbf{W}}{2\pi}\right|-\frac{1}{2}\mathrm{tr}\left(\mathbf{W}\overline{\mathbf{S}}_i\right)-\frac{1}{2}\ln\left|\mathbf{L}_i\right|+\frac{1}{2}\gamma_i^T\mathbf{L}_i^{-1}\gamma_i\;. \qquad \text{(C.56)}
$$

C.4.5 EM algorithm

C.4.5.1 E-step

In the E-step we calculate the posterior of $\mathbf{y}$ by (C.52).

C.4.5.2 M-step ML

We maximize the EM auxiliary function $\mathcal{Q}(\mathcal{M})$:

$$\mathcal{Q}(\mathcal{M}) = \sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[\ln P\left(\mathbf{\Phi}_i, \mathbf{y}_i | \mathcal{M}\right)\right] \tag{C.57}$$

$$= \sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[\ln P\left(\mathbf{\Phi}_i | \mathbf{y}_i, \mathcal{M}\right)\right] + \mathrm{E}_{\mathbf{Y}}\left[\ln P\left(\mathbf{y}_i\right)\right] \ . \tag{C.58}$$

We develop $\mathcal{Q}$ by (C.46):

$$\mathcal{Q}(\mathcal{M}) = \frac{N}{2}\ln|\mathbf{W}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{W}\sum_{i=1}^{M}\left(\mathbf{S}_i - 2\mathbf{F}_i\mathrm{E}_{\mathbf{Y}}\left[\tilde{\mathbf{y}}_i\right]^T\tilde{\mathbf{V}}^T + N_i\tilde{\mathbf{V}}\mathrm{E}_{\mathbf{Y}}\left[\tilde{\mathbf{y}}_i\tilde{\mathbf{y}}_i^T\right]\tilde{\mathbf{V}}^T\right)\right) + \text{const} \tag{C.59}$$

$$= \frac{N}{2}\ln|\mathbf{W}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{W}\left(\mathbf{S} - 2\left(\sum_{i=1}^{M}\mathbf{F}_i\mathrm{E}_{\mathbf{Y}}\left[\tilde{\mathbf{y}}_i\right]^T\right)\tilde{\mathbf{V}}^T + \tilde{\mathbf{V}}\left(\sum_{i=1}^{M}N_i\mathrm{E}_{\mathbf{Y}}\left[\tilde{\mathbf{y}}_i\tilde{\mathbf{y}}_i^T\right]\right)\tilde{\mathbf{V}}^T\right)\right) + \text{const} \ . \tag{C.60}$$

We define the accumulators:

$$\mathbf{R}_{\tilde{\mathbf{y}}} = \sum_{i=1}^{M} N_i\mathrm{E}_{\mathbf{Y}}\left[\tilde{\mathbf{y}}_i\tilde{\mathbf{y}}_i^T\right] \tag{C.61}$$

$$\mathbf{C} = \sum_{i=1}^{M}\mathbf{F}_i\mathrm{E}_{\mathbf{Y}}\left[\tilde{\mathbf{y}}_i\right]^T \ . \tag{C.62}$$

Then, the auxiliary function looks like

$$\mathcal{Q}(\mathcal{M}) = \frac{N}{2}\ln|\mathbf{W}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{W}\left(\mathbf{S} - 2\mathbf{C}\tilde{\mathbf{V}}^T + \tilde{\mathbf{V}}\mathbf{R}_{\tilde{\mathbf{y}}}\tilde{\mathbf{V}}^T\right)\right) + \text{const} \ . \tag{C.63}$$

We derive $\mathcal{Q}$ with respect to $\tilde{\mathbf{V}}$:

$$\frac{\partial\mathcal{Q}(\mathcal{M})}{\partial\tilde{\mathbf{V}}} = \mathbf{C} - \tilde{\mathbf{V}}\mathbf{R}_{\tilde{\mathbf{y}}} = \mathbf{0} \quad \Longrightarrow \tag{C.64}$$

$$\tilde{\mathbf{V}} = \mathbf{C}\mathbf{R}_{\tilde{\mathbf{y}}}^{-1} \ . \tag{C.65}$$

We derive $\mathcal{Q}$ with respect to $\mathbf{W}$:

$$\frac{\partial\mathcal{Q}(\mathcal{M})}{\partial\mathbf{W}} = \frac{N}{2}\left(2\mathbf{W}^{-1} - \mathrm{diag}(\mathbf{W}^{-1})\right) - \frac{1}{2}\left(\mathbf{K} + \mathbf{K}^T - \mathrm{diag}(\mathbf{K})\right) = \mathbf{0} \tag{C.66}$$

where $\mathbf{K} = \mathbf{S} - 2\mathbf{C}\tilde{\mathbf{V}}^T + \tilde{\mathbf{V}}\mathbf{R}_{\tilde{\mathbf{y}}}\tilde{\mathbf{V}}^T$, so

$$\begin{aligned} \mathbf{W}^{-1} =& \frac{1}{N}\frac{\mathbf{K}+\mathbf{K}^T}{2} && \text{(C.67)} \\ =& \frac{1}{N}\left(\mathbf{S} - \tilde{\mathbf{V}}\mathbf{C}^T - \mathbf{C}\tilde{\mathbf{V}}^T + \tilde{\mathbf{V}}\mathbf{R}_{\tilde{\mathbf{y}}}\tilde{\mathbf{V}}^T\right) && \text{(C.68)} \\ =& \frac{1}{N}\left(\mathbf{S} - \tilde{\mathbf{V}}\mathbf{C}^T\right) \ . && \text{(C.69)} \end{aligned}$$

Finally, we just need to evaluate the expectations $\mathrm{E}_{\mathbf{Y}}\left[\tilde{\mathbf{y}}_i\right]$ and $\mathrm{E}_{\mathbf{Y}}\left[\tilde{\mathbf{y}}_i\tilde{\mathbf{y}}_i^T\right]$:

$$\begin{aligned} \mathrm{E}_{\mathbf{Y}}\left[\tilde{\mathbf{y}}_i\right] =& \begin{bmatrix} \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right] \\ 1 \end{bmatrix} && \text{(C.70)} \\ \mathrm{E}_{\mathbf{Y}}\left[\tilde{\mathbf{y}}_i\tilde{\mathbf{y}}_i^T\right] =& \begin{bmatrix} \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right] & \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right] \\ \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]^T & 1 \end{bmatrix} && \text{(C.71)} \\ \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right] =& \mathbf{L}_i^{-1}\gamma_i && \text{(C.72)} \\ \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right] =& \mathbf{L}_i^{-1} + \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]^T \ . && \text{(C.73)} \end{aligned}$$

C.4.5.3 M-step MD

The minimum divergence (MD) step consists in minimizing the KL distance between the true and the assumed prior for the hidden variables. It has been observed that MD helps to avoid saddle points and speeds convergence up [Brummer, 2009]. We, temporally, need to assume an over-parametrized model with a general prior for the hidden variables:

$$P\left(\mathbf{y}_i\right) = \mathcal{N}\left(\mathbf{y}_i|\mu_{\mathbf{y}}, \mathbf{\Lambda}_{\mathbf{y}}^{-1}\right) \ . \qquad \text{(C.74)}$$

Now, we maximize the term of the EM auxiliary that depends on the new parameters $\mu_{\mathbf{y}}$ and $\mathbf{\Lambda}_{\mathbf{y}}$:

$$\begin{aligned} \mathcal{Q}(\mu_{\mathbf{y}}, \mathbf{\Lambda}_{\mathbf{y}}) =& \sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[\ln\mathcal{N}\left(\mathbf{y}_i|\mu_{\mathbf{y}}, \mathbf{\Lambda}_{\mathbf{y}}^{-1}\right)\right] && \text{(C.75)} \\ =& \frac{M}{2}\ln\left|\mathbf{\Lambda}_{\mathbf{y}}\right| - \frac{1}{2}\mathrm{tr}\left(\mathbf{\Lambda}_{\mathbf{y}}\sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}\left[\left(\mathbf{y}_i-\mu_{\mathbf{y}}\right)\left(\mathbf{y}_i-\mu_{\mathbf{y}}\right)^T\right]\right) + \text{const} \ . && \text{(C.76)} \end{aligned}$$

We derive $\mathcal{Q}$ with respect to $\mu_{\mathbf{y}}$

$$\begin{aligned} \frac{\partial\mathcal{Q}(\mu_{\mathbf{y}}, \mathbf{\Lambda}_{\mathbf{y}})}{\partial\mu_{\mathbf{y}}} =& \frac{1}{2}\sum_{i=1}^{M}\mathbf{\Lambda}_{\mathbf{y}}\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i-\mu_{\mathbf{y}}\right] = \mathbf{0} \quad \Longrightarrow && \text{(C.77)} \\ \mu_{\mathbf{y}} =& \frac{1}{M}\sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right] \ . && \text{(C.78)} \end{aligned}$$

We derive $\mathcal{Q}$ with respect to $\mathbf{\Lambda}_{\mathbf{y}}^{-1}$:

$$\frac{\partial\mathcal{Q}(\mu_{\mathbf{y}}, \mathbf{\Lambda}_{\mathbf{y}})}{\partial\mathbf{\Lambda}_{\mathbf{y}}^{-1}} = \frac{M}{2}\left(2\mathbf{\Lambda}_{\mathbf{y}}^{-1} - \mathrm{diag}(\mathbf{\Lambda}_{\mathbf{y}}^{-1})\right) - \frac{1}{2}\left(2\mathbf{K} - \mathrm{diag}(\mathbf{K})\right) = \mathbf{0} \qquad \text{(C.79)}$$

where $\mathbf{K} = \sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[(\mathbf{y}_i - \mu_{\mathbf{y}})(\mathbf{y}_i - \mu_{\mathbf{y}})^T\right]$, so

$$\boldsymbol{\Sigma}_{\mathbf{y}} = \boldsymbol{\Lambda}_{\mathbf{y}}^{-1} = \frac{1}{M}\sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[(\mathbf{y}_i - \mu_{\mathbf{y}})(\mathbf{y}_i - \mu_{\mathbf{y}})^T\right] = \frac{1}{M}\sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right] - \mu_{\mathbf{y}}\mu_{\mathbf{y}}^T\ . \tag{C.80}$$

To minimize the divergence between the standard and the general prior we need to find a transform $\mathbf{y} = \psi(\mathbf{y}')$ such as $\mathbf{y}'$ has a standard prior. That is

$$\mathbf{y} =\mu_{\mathbf{y}} + (\boldsymbol{\Sigma}_{\mathbf{y}}^{1/2})^T\mathbf{y}'\ . \tag{C.81}$$

By applying this transform, we make the effect of the non-standard priors to be absorbed into μ and $\mathbf{V}$:

$$\mu' =\mu + \mathbf{V}\mu_{\mathbf{y}} \tag{C.82}$$
$$\mathbf{V}' =\mathbf{V}(\boldsymbol{\Sigma}_{\mathbf{y}}^{1/2})^T \tag{C.83}$$

where $\boldsymbol{\Sigma}_{\mathbf{y}}^{1/2}$ is the upper triangular Cholesky decomposition of $\boldsymbol{\Sigma}_{\mathbf{y}}$.

C.4.5.4 Objective function

Convergence can be checked by computing the marginal likelihood of the development data given the model, i.e., this is Equation (C.56) summed for all the speakers:

$$\ln P(\boldsymbol{\Phi}|\mathcal{M}) = \frac{N}{2}\ln\left|\frac{\mathbf{W}}{2\pi}\right| - \frac{1}{2}\mathrm{tr}\left(\mathbf{W}\overline{\mathbf{S}}\right) - \frac{1}{2}\sum_{i=1}^{M}\ln|\mathbf{L}_i| + \frac{1}{2}\sum_{i=1}^{M}\gamma_i^T\mathbf{L}_i^{-1}\gamma_i\ . \tag{C.84}$$

C.5 PLDA

C.5.1 Model definition

The general PLDA model supposes that an i-vector ϕ_{ij} of the session j of the speaker i is written as:

$$\phi_{ij} = \mu + \mathbf{V}\mathbf{y}_i + \mathbf{U}\mathbf{x}_{ij} + \epsilon_{ij} \tag{C.85}$$

where μ is a speaker independent term, $\mathbf{V}$ is a low-rank eigen-voices matrix, $\mathbf{y}_i$ is the speaker factors vector, $\mathbf{U}$ is a low-rank eigen-channels matrix, $\mathbf{x}_{ij}$ is the channel factors vector and ϵ_{ij} is an offset accounting for the residual variability not included in $\mathbf{U}$.

We assume the following priors:

$$\mathbf{y}_i \sim \mathcal{N}(\mathbf{y}_i|\mathbf{0},\mathbf{I}) \tag{C.86}$$
$$\mathbf{x}_{ij} \sim \mathcal{N}(\mathbf{x}_{ij}|\mathbf{0},\mathbf{I}) \tag{C.87}$$
$$\epsilon_{ij} \sim \mathcal{N}\left(\epsilon_{ij}|\mathbf{0},\mathbf{D}^{-1}\right) \tag{C.88}$$

where $\mathcal{N}$ denotes the Gaussian distribution; and $\mathbf{D}$ is a diagonal precision matrix. ϕ is observed variable while $\mathbf{y}$ and $\mathbf{x}$ are hidden. We denote by $\mathbf{X}_i$ the set of all the channel factors of speaker i and by $\mathcal{M} = (\mu, \mathbf{V}, \mathbf{U}, \mathbf{D})$ the set of all the model parameters.

C.5.2 Data conditional likelihood

The likelihood of the data given the hidden variables, for speaker i, is

$$\ln P\left(\boldsymbol{\Phi}_i|\mathbf{y}_i,\mathbf{X}_i,\mathcal{M}\right)=\sum_{j=1}^{N_i}\ln\mathcal{N}\left(\phi_{ij}|\mu+\mathbf{V}\mathbf{y}_i+\mathbf{U}\mathbf{x}_{ij},\mathbf{D}^{-1}\right) \tag{C.89}$$

$$=\frac{N_i}{2}\ln\left|\frac{\mathbf{D}}{2\pi}\right|-\frac{1}{2}\sum_{j=1}^{N_i}(\phi_{ij}-\mu-\mathbf{V}\mathbf{y}_i-\mathbf{U}\mathbf{x}_{ij})^T\mathbf{D}(\phi_{ij}-\mu-\mathbf{V}\mathbf{y}_i-\mathbf{U}\mathbf{x}_{ij}) \tag{C.90}$$

$$=\frac{N_i}{2}\ln\left|\frac{\mathbf{D}}{2\pi}\right|-\frac{1}{2}\mathrm{tr}\left(\mathbf{D}\overline{\mathbf{S}}_i\right)+\mathbf{y}^T\mathbf{V}^T\mathbf{D}\overline{\mathbf{F}}_i-\frac{N_i}{2}\mathbf{y}_i^T\mathbf{V}^T\mathbf{D}\mathbf{V}\mathbf{y}$$
$$+\sum_{j=1}^{N_i}\mathbf{x}_{ij}^T\mathbf{U}^T\mathbf{D}\left(\phi_{ij}-\mu\right)-\mathbf{y}_i^T\mathbf{V}^T\mathbf{D}\mathbf{U}\mathbf{x}_{ij}-\frac{1}{2}\mathbf{x}_{ij}^T\mathbf{U}^T\mathbf{D}\mathbf{U}\mathbf{x}_{ij}\;. \tag{C.91}$$

We can write this likelihood in other form by defining:

$$\tilde{\mathbf{y}}_{ij}=\begin{bmatrix}\mathbf{y}_i\\ \mathbf{x}_{ij}\\ 1\end{bmatrix}\,,\qquad\qquad \tilde{\mathbf{V}}=\begin{bmatrix}\mathbf{V} & \mathbf{U} & \mu\end{bmatrix}\;. \tag{C.92}$$

Then, we obtain

$$\ln P\left(\boldsymbol{\Phi}_i|\mathbf{y}_i,\mathbf{X}_i,\mathcal{M}\right)=\sum_{j=1}^{N_i}\ln\mathcal{N}\left(\phi_{ij}|\tilde{\mathbf{V}}\tilde{\mathbf{y}}_{ij},\mathbf{D}^{-1}\right) \tag{C.93}$$

$$=\frac{N_i}{2}\ln\left|\frac{\mathbf{D}}{2\pi}\right|-\frac{1}{2}\sum_{j=1}^{N_i}(\phi_{ij}-\tilde{\mathbf{V}}\tilde{\mathbf{y}}_{ij})^T\mathbf{D}(\phi_{ij}-\tilde{\mathbf{V}}\tilde{\mathbf{y}}_{ij}) \tag{C.94}$$

$$=\frac{N_i}{2}\ln\left|\frac{\mathbf{D}}{2\pi}\right|-\frac{1}{2}\mathrm{tr}\left(\mathbf{D}\mathbf{S}_i\right)+\sum_{j=1}^{N_i}\tilde{\mathbf{y}}_{ij}^T\tilde{\mathbf{V}}^T\mathbf{D}\phi_{ij}-\frac{1}{2}\tilde{\mathbf{y}}_{ij}^T\tilde{\mathbf{V}}^T\mathbf{D}\tilde{\mathbf{V}}\tilde{\mathbf{y}}_{ij} \tag{C.95}$$

$$=\frac{N_i}{2}\ln\left|\frac{\mathbf{D}}{2\pi}\right|-\frac{1}{2}\mathrm{tr}\left(\mathbf{D}\left(\mathbf{S}_i+\sum_{j=1}^{N_i}-2\phi_{ij}\tilde{\mathbf{y}}_{ij}^T\tilde{\mathbf{V}}^T+\tilde{\mathbf{V}}\tilde{\mathbf{y}}_{ij}\tilde{\mathbf{y}}_{ij}^T\tilde{\mathbf{V}}^T\right)\right)\;. \tag{C.96}$$

C.5.3 Posterior of the hidden variables

The posterior of the hidden variables can be decomposed into two factors by applying the product rule:

$$P\left(\mathbf{y}_i,\mathbf{X}_i|\boldsymbol{\Phi}_i,\mathcal{M}\right)=P\left(\mathbf{X}_i|\mathbf{y}_i,\boldsymbol{\Phi}_i,\mathcal{M}\right)P\left(\mathbf{y}_i|\boldsymbol{\Phi}_i,\mathcal{M}\right) \tag{C.97}$$

C.5.3.1 Conditional posterior of $\mathbf{X}_i$

The conditional posterior of $\mathbf{X}_i$ is

$$P\left(\mathbf{X}_i|\mathbf{y}_i,\boldsymbol{\Phi}_i,\mathcal{M}\right)=\frac{P\left(\boldsymbol{\Phi}_i|\mathbf{y}_i,\mathbf{X}_i,\mathcal{M}\right)P\left(\mathbf{X}_i\right)}{P\left(\boldsymbol{\Phi}_i|\mathbf{y}_i,\mathcal{M}\right)}\;. \tag{C.98}$$

By using (C.87) and (C.91), we obtain

$$\ln P\left(\mathbf{X}_i|\mathbf{y}_i,\boldsymbol{\Phi}_i,\mathcal{M}\right) = \ln P\left(\boldsymbol{\Phi}_i|\mathbf{y}_i,\mathbf{X}_i,\mathcal{M}\right) + \ln P\left(\mathbf{X}_i|\mathcal{M}\right) + \text{const} \tag{C.99}$$

$$= \sum_{j=1}^{N_i} \mathbf{x}_{ij}^T\mathbf{U}^T\mathbf{D}\left(\phi_{ij}-\mu\right) - \mathbf{y}^T\mathbf{V}^T\mathbf{D}\mathbf{U}\mathbf{x}_{ij}$$

$$- \frac{1}{2}\mathbf{x}_{ij}^T\mathbf{U}^T\mathbf{D}\mathbf{U}\mathbf{x}_{ij} - \frac{1}{2}\mathbf{x}_{ij}^T\mathbf{x}_{ij} + \text{const} \tag{C.100}$$

$$= \sum_{j=1}^{N_i} \mathbf{x}_{ij}^T\mathbf{U}^T\mathbf{D}\left(\phi_{ij}-\mu-\mathbf{V}\mathbf{y}_i\right) - \frac{1}{2}\mathbf{x}_{ij}^T\mathbf{L}_{\mathbf{x}}\mathbf{x}_{ij} + \text{const} \tag{C.101}$$

$$= \sum_{j=1}^{N_i} \mathbf{x}_{ij}^T\zeta_{ij} - \frac{1}{2}\mathbf{x}_{ij}^T\mathbf{L}_{\mathbf{x}}\mathbf{x}_{ij} + \text{const} \tag{C.102}$$

where

$$\zeta_{ij} = \mathbf{U}^T\mathbf{D}\left(\phi_{ij}-\mu-\mathbf{V}\mathbf{y}_i\right) = \tilde{\zeta}_{ij} - \mathbf{J}\mathbf{y}_i \tag{C.103}$$

$$\tilde{\zeta}_{ij} = \mathbf{U}^T\mathbf{D}\left(\phi_{ij}-\mu\right) \tag{C.104}$$

$$\mathbf{J} = \mathbf{U}^T\mathbf{D}\mathbf{V} \tag{C.105}$$

$$\mathbf{L}_{\mathbf{x}} = \mathbf{I} + \mathbf{U}^T\mathbf{D}\mathbf{U}\ . \tag{C.106}$$

Equation (C.102) has the form of a product of Gaussian distributions. Therefore

$$P\left(\mathbf{X}_i|\mathbf{y}_i,\boldsymbol{\Phi}_i,\mathcal{M}\right) = \prod_{j=1}^{N_i}\mathcal{N}\left(\mathbf{x}_{ij}|\overline{\mathbf{x}}_{ij},\mathbf{L}_{\mathbf{x}}^{-1}\right) \tag{C.107}$$

where

$$\overline{\mathbf{x}}_{ij} = \mathbf{L}_{\mathbf{x}}^{-1}\zeta_{ij}\ . \tag{C.108}$$

C.5.3.2 Posterior of $\mathbf{y}_i$

The marginal posterior of $\mathbf{y}$ is

$$P\left(\mathbf{y}_i|\boldsymbol{\Phi}_i,\mathcal{M}\right) = \frac{P\left(\boldsymbol{\Phi}_i|\mathbf{y}_i,\mathcal{M}\right)P\left(\mathbf{y}_i\right)}{P\left(\boldsymbol{\Phi}_i|\mathcal{M}\right)}\ . \tag{C.109}$$

Now, we apply Bayes theorem to write

$$P\left(\boldsymbol{\Phi}_i,\mathbf{X}_i|\mathbf{y}_i,\mathcal{M}\right) = P\left(\boldsymbol{\Phi}_i|\mathbf{y}_i,\mathbf{X}_i,\mathcal{M}\right)P\left(\mathbf{X}_i|\mathbf{y}_i,\mathcal{M}\right) = P\left(\mathbf{X}_i|\boldsymbol{\Phi}_i,\mathbf{y}_i,\mathcal{M}\right)P\left(\boldsymbol{\Phi}_i|\mathbf{y}_i,\mathcal{M}\right)\ . \tag{C.110}$$

Simplifying:

$$P\left(\boldsymbol{\Phi}_i|\mathbf{y}_i,\mathbf{X}_i,\mathcal{M}\right)P\left(\mathbf{X}_i\right) = P\left(\mathbf{X}_i|\boldsymbol{\Phi}_i,\mathbf{y}_i,\mathcal{M}\right)P\left(\boldsymbol{\Phi}_i|\mathbf{y}_i,\mathcal{M}\right)\ . \tag{C.111}$$

Then, isolating $P\left(\mathbf{y}_i|\boldsymbol{\Phi}_i,\mathcal{M}\right)$:

$$P\left(\mathbf{y}_i|\boldsymbol{\Phi}_i,\mathcal{M}\right) = \left.\frac{P\left(\boldsymbol{\Phi}_i|\mathbf{y}_i,\mathbf{X}_i,\mathcal{M}\right)P\left(\mathbf{X}_i\right)P\left(\mathbf{y}_i\right)}{P\left(\mathbf{X}_i|\boldsymbol{\Phi}_i,\mathbf{y}_i,\mathcal{M}\right)P\left(\boldsymbol{\Phi}_i|\mathcal{M}\right)}\right|_{\mathbf{X}_i=\mathbf{0}}\ . \tag{C.112}$$

Note that $P(\mathbf{y}_i|\mathbf{\Phi}_i,\mathcal{M})$ is not conditioned on $\mathbf{X}_i$ so we can plug-in whatever value that we choose.

By Equations (C.86), (C.91) and (C.102), we obtain:

$$\begin{aligned}
\ln P(\mathbf{y}_i|\mathbf{\Phi}_i,\mathcal{M}) =& \ln P(\mathbf{\Phi}_i|\mathbf{y}_i,\mathbf{X}_i,\mathcal{M}) + \ln P(\mathbf{y}) - \ln P(\mathbf{X}_i|\mathbf{\Phi}_i,\mathbf{y}_i,\mathcal{M}) + \text{const} \qquad \text{(C.113)}\\
=&\mathbf{y}^T\mathbf{V}^T\mathbf{D}\overline{\mathbf{F}}_i - \frac{N_i}{2}\mathbf{y}_i^T\mathbf{V}^T\mathbf{D}\mathbf{V}\mathbf{y}_i - \frac{1}{2}\mathbf{y}_i^T\mathbf{y}_i + \frac{1}{2}\sum_{j=1}^{N_i}\overline{\mathbf{x}}_{ij}^T\mathbf{L}_\mathbf{x}\overline{\mathbf{x}}_{ij} + \text{const} \qquad \text{(C.114)}\\
=&\mathbf{y}^T\mathbf{V}^T\mathbf{D}\overline{\mathbf{F}}_i - \frac{1}{2}\mathbf{y}_i^T\left(\mathbf{I} + N_i\mathbf{V}^T\mathbf{D}\mathbf{V}\right)\mathbf{y}_i\\
&+\frac{1}{2}\sum_{j=1}^{N_i}\left(\phi_{ij}-\mu-\mathbf{V}\mathbf{y}_i\right)^T\mathbf{D}\mathbf{U}\mathbf{L}_\mathbf{x}^{-1}\mathbf{U}^T\mathbf{D}\left(\phi_{ij}-\mu-\mathbf{V}\mathbf{y}_i\right) + \text{const} \qquad \text{(C.115)}\\
=&\mathbf{y}^T\mathbf{V}^T\mathbf{D}\overline{\mathbf{F}}_i - \frac{1}{2}\mathbf{y}_i^T\left(\mathbf{I} + N_i\mathbf{V}^T\mathbf{D}\mathbf{V}\right)\mathbf{y}_i\\
&+\frac{1}{2}\sum_{j=1}^{N_i}\left(\phi_{ij}-\mu\right)^T\mathbf{D}\mathbf{U}\mathbf{L}_\mathbf{x}^{-1}\mathbf{U}^T\mathbf{D}\left(\phi_{ij}-\mu\right)\\
&-2\mathbf{y}_i^T\mathbf{V}^T\mathbf{D}\mathbf{U}\mathbf{L}_\mathbf{x}^{-1}\mathbf{U}^T\mathbf{D}\left(\phi_{ij}-\mu\right)\\
&+\mathbf{y}_i^T\mathbf{V}^T\mathbf{D}\mathbf{U}\mathbf{L}_\mathbf{x}^{-1}\mathbf{U}^T\mathbf{D}\mathbf{V}\mathbf{y}_i + \text{const} \qquad \text{(C.116)}\\
=&\mathbf{y}^T\mathbf{V}^T\left(\mathbf{D}-\mathbf{D}\mathbf{U}\mathbf{L}_\mathbf{x}^{-1}\mathbf{U}^T\mathbf{D}\right)\overline{\mathbf{F}}_i\\
&-\frac{1}{2}\mathbf{y}_i^T\left(\mathbf{I}+N_i\mathbf{V}^T\left(\mathbf{D}-\mathbf{D}\mathbf{U}\mathbf{L}_\mathbf{x}^{-1}\mathbf{U}^T\mathbf{D}\right)\mathbf{V}\right)\mathbf{y}_i + \text{const}\,. \qquad \text{(C.117)}
\end{aligned}$$

That is a Gaussian distribution:

$$P(\mathbf{y}_i|\mathbf{\Phi}_i,\mathcal{M}) = \mathcal{N}\left(\mathbf{y}_i|\overline{\mathbf{y}}_i,\mathbf{L}_{\mathbf{y}_i}^{-1}\right) \tag{C.118}$$

where

$$\begin{aligned}
\mathbf{L}_{\mathbf{y}_i} =& \mathbf{I} + N_i\mathbf{V}^T\left(\mathbf{D}-\mathbf{D}\mathbf{U}\mathbf{L}_\mathbf{x}^{-1}\mathbf{U}^T\mathbf{D}\right)\mathbf{V} = \mathbf{I} + N_i\left(\mathbf{V}^T\mathbf{D}\mathbf{V} - \mathbf{J}^T\mathbf{L}_\mathbf{x}^{-1}\mathbf{J}\right) \qquad \text{(C.119)}\\
\gamma_i =& \mathbf{V}^T\left(\mathbf{D}-\mathbf{D}\mathbf{U}\mathbf{L}_\mathbf{x}^{-1}\mathbf{U}^T\mathbf{D}\right)\overline{\mathbf{F}}_i = \tilde{\gamma}_i - \mathbf{J}^T\mathbf{L}_\mathbf{x}^{-1}\tilde{\zeta}_i \qquad \text{(C.120)}\\
\tilde{\gamma}_i =& \mathbf{V}^T\mathbf{D}\overline{\mathbf{F}}_i \qquad \text{(C.121)}\\
\tilde{\zeta}_i =& \sum_{j=1}^{N_i}\tilde{\zeta}_{ij} \qquad \text{(C.122)}\\
\overline{\mathbf{y}}_i =& \mathbf{L}_{\mathbf{y}_i}^{-1}\gamma_i\,. \qquad \text{(C.123)}
\end{aligned}$$

C.5.4 Marginal likelihood of the data

The marginal likelihood for speaker i data is

$$P(\mathbf{\Phi}_i|\mathcal{M}) = \left.\frac{P(\mathbf{\Phi}_i|\mathbf{y}_i,\mathbf{X}_i,\mathcal{M})\,P(\mathbf{y}_i)\,P(\mathbf{X}_i)}{P(\mathbf{X}_i|\mathbf{y}_i,\mathbf{\Phi}_i,\mathcal{M})\,P(\mathbf{y}_i|\mathbf{\Phi}_i,\mathcal{M})}\right|_{\mathbf{y}_i=\mathbf{0},\mathbf{X}_i=\mathbf{0}}\,. \tag{C.124}$$

Taking (C.91), (C.86), (C.87), (C.102) and (C.118):

$$\begin{aligned}\ln P\left(\mathbf{\Phi}_i|\mathcal{M}\right) =& \frac{N_i}{2}\ln\left|\frac{\mathbf{D}}{2\pi}\right| - \frac{1}{2}\mathrm{tr}\left(\mathbf{D}\overline{\mathbf{S}}_i\right) - \frac{N_i}{2}\ln\left|\mathbf{L}_{\mathbf{x}}\right| + \frac{1}{2}\sum_{j=1}^{N_i}\tilde{\zeta}_{ij}^T\mathbf{L}_{\mathbf{x}}^{-1}\tilde{\zeta}_{ij} \\ & - \frac{1}{2}\ln\left|\mathbf{L}_{\mathbf{y}_i}\right| + \frac{1}{2}\gamma_i^T\mathbf{L}_{\mathbf{y}_i}^{-1}\gamma_i \ . \end{aligned} \tag{C.125}$$

The fourth term can be developed as

$$\sum_{j=1}^{N_i}\tilde{\zeta}_{ij}^T\mathbf{L}_{\mathbf{x}}^{-1}\tilde{\zeta}_{ij} = \sum_{j=1}^{N_i}\left(\phi_{ij}-\mu\right)^T\mathbf{D}\mathbf{U}\mathbf{L}_{\mathbf{x}}^{-1}\mathbf{U}^T\mathbf{D}\left(\phi_{ij}-\mu\right) \tag{C.126}$$

$$= \mathrm{tr}\left(\mathbf{D}\mathbf{U}\mathbf{L}_{\mathbf{x}}^{-1}\mathbf{U}^T\mathbf{D}\sum_{j=1}^{N_i}\left(\phi_{ij}-\mu\right)\left(\phi_{ij}-\mu\right)^T\right) \tag{C.127}$$

$$= \mathrm{tr}\left(\mathbf{D}\mathbf{U}\mathbf{L}_{\mathbf{x}}^{-1}\mathbf{U}^T\mathbf{D}\overline{\mathbf{S}}_i\right) \ . \tag{C.128}$$

Finally, we obtain:

$$\begin{aligned}\ln P\left(\mathbf{\Phi}_i|\mathcal{M}\right) =& \frac{N_i}{2}\ln\left|\frac{\mathbf{D}}{2\pi}\right| - \frac{1}{2}\mathrm{tr}\left(\left(\mathbf{D}-\mathbf{D}\mathbf{U}\mathbf{L}_{\mathbf{x}}^{-1}\mathbf{U}^T\mathbf{D}\right)\overline{\mathbf{S}}_i\right) - \frac{N_i}{2}\ln\left|\mathbf{L}_{\mathbf{x}}\right| \\ & - \frac{1}{2}\ln\left|\mathbf{L}_{\mathbf{y}_i}\right| + \frac{1}{2}\gamma_i^T\mathbf{L}_{\mathbf{y}_i}^{-1}\gamma_i \ . \end{aligned} \tag{C.129}$$

C.5.5 EM algorithm

C.5.5.1 E-step

In the E-step, we calculate the posterior of $\mathbf{y}$ and $\mathbf{X}$ by (C.97).

C.5.5.2 M-step ML

We maximize the EM auxiliary function $\mathcal{Q}(\mathcal{M})$

$$\mathcal{Q}(\mathcal{M}) = \sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y},\mathbf{X}}\left[\ln P\left(\mathbf{\Phi}_i,\mathbf{y}_i,\mathbf{X}_i|\mathcal{M}\right)\right] \tag{C.130}$$

$$= \sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y},\mathbf{X}}\left[\ln P\left(\mathbf{\Phi}_i|\mathbf{y}_i,\mathbf{X}_i,\mathcal{M}\right)\right] + \mathrm{E}_{\mathbf{Y},\mathbf{X}}\left[\ln P\left(\mathbf{y}_i\right)\right] + \mathrm{E}_{\mathbf{Y},\mathbf{X}}\left[\ln P\left(\mathbf{X}_i\right)\right] \ . \tag{C.131}$$

By taking (C.96), that is

$$\mathcal{Q}(\mathcal{M}) = \frac{N}{2}\ln\left|\frac{\mathbf{D}}{2\pi}\right| - \frac{1}{2}\mathrm{tr}\left(\mathbf{D}\left(\mathbf{S} + \sum_{i=1}^{M}\sum_{j=1}^{N_i} -2\phi_{ij}\mathrm{E}_{\mathbf{Y},\mathbf{X}}\left[\tilde{\mathbf{y}}_{ij}\right]^T\tilde{\mathbf{V}}^T + \tilde{\mathbf{V}}\mathrm{E}_{\mathbf{Y},\mathbf{X}}\left[\tilde{\mathbf{y}}_{ij}\tilde{\mathbf{y}}_{ij}^T\right]\tilde{\mathbf{V}}^T\right)\right) \ . \tag{C.132}$$

We define the accumulators:

$$\mathbf{R}_{\tilde{\mathbf{y}}} = \sum_{i=1}^{M}\sum_{j=1}^{N_i}\mathrm{E}_{\mathbf{Y}}\left[\tilde{\mathbf{y}}_{ij}\tilde{\mathbf{y}}_{ij}^T\right] \tag{C.133}$$

$$\mathbf{C} = \sum_{i=1}^{M}\sum_{j=1}^{N_i}\phi_{ij}\mathrm{E}_{\mathbf{Y}}\left[\tilde{\mathbf{y}}_{ij}\right]^T ; . \tag{C.134}$$

Then, the auxiliary looks like

$$\mathcal{Q}(\mathcal{M}) = \frac{N}{2} \ln |\mathbf{D}| - \frac{1}{2} \mathrm{tr} \left(\mathbf{D} \left(\mathbf{S} - 2\mathbf{C}\tilde{\mathbf{V}}^T + \tilde{\mathbf{V}}\mathbf{R}_{\tilde{\mathbf{y}}}\tilde{\mathbf{V}}^T \right) \right) + \text{const} \,. \tag{C.135}$$

We derive $\mathcal{Q}$ with respect to $\tilde{\mathbf{V}}$:

$$\frac{\partial \mathcal{Q}(\mathcal{M})}{\partial \tilde{\mathbf{V}}} = \mathbf{C} - \tilde{\mathbf{V}}\mathbf{R}_{\tilde{\mathbf{y}}} = \mathbf{0} \quad \Longrightarrow \tag{C.136}$$

$$\tilde{\mathbf{V}} = \mathbf{C}\mathbf{R}_{\tilde{\mathbf{y}}}^{-1} \,. \tag{C.137}$$

We derive $\mathcal{Q}$ with respect to $\mathbf{D}$:

$$\frac{\partial \mathcal{Q}(\mathcal{M})}{\partial \mathbf{D}} = \frac{N}{2} \left(2\mathbf{D}^{-1} - \mathrm{diag}(\mathbf{D}^{-1}) \right) - \frac{1}{2} \left(\mathbf{K} + \mathbf{K}^T - \mathrm{diag}(\mathbf{K}) \right) = \mathbf{0} \tag{C.138}$$

where $\mathbf{K} = \mathbf{S} - 2\mathbf{C}\tilde{\mathbf{V}}^T + \tilde{\mathbf{V}}\mathbf{R}_{\tilde{\mathbf{y}}}\tilde{\mathbf{V}}^T$, so

$$\mathbf{D}^{-1} = \frac{1}{N} \frac{\mathbf{K} + \mathbf{K}^T}{2} \tag{C.139}$$

$$= \frac{1}{N} \left(\mathbf{S} - \tilde{\mathbf{V}}\mathbf{C}^T - \mathbf{C}\tilde{\mathbf{V}}^T + \tilde{\mathbf{V}}\mathbf{R}_{\tilde{\mathbf{y}}}\tilde{\mathbf{V}}^T \right) \tag{C.140}$$

$$= \frac{1}{N} \left(\mathbf{S} - \tilde{\mathbf{V}}\mathbf{C}^T \right) \,. \tag{C.141}$$

Finally, we need to evaluate the expectations $\mathrm{E}_{\mathbf{Y}}\left[\tilde{\mathbf{y}}_{ij}\right]$ and $\mathrm{E}_{\mathbf{Y}}\left[\tilde{\mathbf{y}}_{ij}\tilde{\mathbf{y}}_{ij}^T\right]$ and compute $\mathbf{R}_{\tilde{\mathbf{y}}}$ and $\mathbf{C}$. For $\mathbf{C}$, we have

$$\mathbf{C} = \sum_{i=1}^{M} \sum_{j=1}^{N_i} \phi_{ij} \mathrm{E}_{\mathbf{Y},\mathbf{X}} \left[\tilde{\mathbf{y}}_{ij}\right]^T = \sum_{i=1}^{M} \sum_{j=1}^{N_i} \phi_{ij} \begin{bmatrix} \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right] \\ \mathrm{E}_{\mathbf{Y},\mathbf{X}}\left[\mathbf{x}_{ij}\right] \\ 1 \end{bmatrix}^T = \begin{bmatrix} \mathbf{C}_{\mathbf{y}} & \mathbf{C}_{\mathbf{x}} & \mathbf{F} \end{bmatrix} \tag{C.142}$$

where

$$\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right] = \overline{\mathbf{y}}_i \tag{C.143}$$

$$\mathrm{E}_{\mathbf{Y},\mathbf{X}}\left[\mathbf{x}_{ij}\right] = \mathrm{E}_{\mathbf{Y}}\left[\overline{\mathbf{x}}_{ij}\right] = \mathbf{L}_{\mathbf{x}}^{-1} \left(\tilde{\zeta}_{ij} - \mathbf{J}\overline{\mathbf{y}}_i \right) \tag{C.144}$$

and

$$\mathbf{C}_{\mathbf{y}} = \sum_{i=1}^{M} \sum_{j=1}^{N_i} \phi_{ij} \overline{\mathbf{y}}_i^T = \sum_{i=1}^{M} \mathbf{F}_i \overline{\mathbf{y}}_i^T \tag{C.145}$$

$$\mathbf{C}_{\mathbf{x}} = \sum_{i=1}^{M} \sum_{j=1}^{N_i} \phi_{ij} \left(\tilde{\zeta}_{ij} - \mathbf{J}\overline{\mathbf{y}}_i \right)^T \mathbf{L}_{\mathbf{x}}^{-1} \tag{C.146}$$

$$= \left(\sum_{i=1}^{M} \sum_{j=1}^{N_i} \phi_{ij} \left(\mathbf{U}^T \mathbf{D} \left(\phi_{ij} - \mu \right) \right)^T - \phi_{ij} \overline{\mathbf{y}}_i^T \mathbf{J}^T \right) \mathbf{L}_{\mathbf{x}}^{-1} \tag{C.147}$$

$$= \left(\sum_{i=1}^{M} \sum_{j=1}^{N_i} \phi_{ij} \left(\phi_{ij} - \mu \right)^T \mathbf{D}\mathbf{U} - \phi_{ij} \overline{\mathbf{y}}_i^T \mathbf{J}^T \right) \mathbf{L}_{\mathbf{x}}^{-1} \tag{C.148}$$

$$= \left(\left(\mathbf{S} - \mathbf{F}\mu^T \right) \mathbf{D}\mathbf{U} - \sum_{i=1}^{M} \sum_{j=1}^{N_i} \phi_{ij} \overline{\mathbf{y}}_i^T \mathbf{J}^T \right) \mathbf{L}_{\mathbf{x}}^{-1} \tag{C.149}$$

$$= \left(\left(\mathbf{S} - \mathbf{F}\mu^T \right) \mathbf{D}\mathbf{U} - \mathbf{C}_{\mathbf{y}} \mathbf{J}^T \right) \mathbf{L}_{\mathbf{x}}^{-1} \,. \tag{C.150}$$

For $\mathbf{R}_{\tilde{\mathbf{y}}}$, we have

$$\mathbf{R}_{\tilde{\mathbf{y}}} = \begin{bmatrix} \mathbf{R}_{\mathbf{y}} & \mathbf{R}_{\mathbf{yx}} & \mathbf{R}_{\mathbf{y1}} \\ \mathbf{R}_{\mathbf{xy}} & \mathbf{R}_{\mathbf{x}} & \mathbf{R}_{\mathbf{x1}} \\ \mathbf{R}_{\mathbf{y1}}^T & \mathbf{R}_{\mathbf{x1}}^T & N \end{bmatrix} \tag{C.151}$$

where

$$\mathbf{R}_{\mathbf{y1}} = \sum_{i=1}^{M} N_i \mathrm{E}_{\mathbf{Y}} \left[\mathbf{y}_i\right] = \sum_{i=1}^{M} N_i \overline{\mathbf{y}}_i \tag{C.152}$$

$$\mathbf{R}_{\mathbf{x1}} = \sum_{i=1}^{M} \sum_{j=1}^{N_i} \mathrm{E}_{\mathbf{Y},\mathbf{X}} \left[\mathbf{x}_{ij}\right] = \mathbf{L}_{\mathbf{x}}^{-1} \left(\sum_{i=1}^{M} \sum_{j=1}^{N_i} \mathbf{U}^T \mathbf{D} \left(\phi_{ij} - \mu\right) - \mathbf{J}\overline{\mathbf{y}}_i \right) \tag{C.153}$$

$$= \mathbf{L}_{\mathbf{x}}^{-1} \left(\mathbf{U}^T \mathbf{D} \overline{\mathbf{F}} - \mathbf{J}\mathbf{R}_{\mathbf{y1}}\right) \tag{C.154}$$

$$\mathbf{R}_{\mathbf{y}} = \sum_{i=1}^{M} N_i \mathrm{E}_{\mathbf{Y}} \left[\mathbf{y}_i \mathbf{y}_i^T\right] = \sum_{i=1}^{M} N_i \left(\mathbf{L}_{\mathbf{y}_i}^{-1} + \overline{\mathbf{y}}_i \overline{\mathbf{y}}_i^T\right) \tag{C.155}$$

$$\mathbf{R}_{\mathbf{xy}} = \sum_{i=1}^{M} \sum_{j=1}^{N_i} \mathrm{E}_{\mathbf{Y},\mathbf{X}} \left[\mathbf{x}_{ij} \mathbf{y}_i^T\right] = \sum_{i=1}^{M} \sum_{j=1}^{N_i} \mathrm{E}_{\mathbf{Y}} \left[\mathbf{L}_{\mathbf{x}}^{-1} \left(\tilde{\zeta}_{ij} - \mathbf{J}\mathbf{y}_i\right) \mathbf{y}_i^T\right] \tag{C.156}$$

$$= \sum_{i=1}^{M} \mathbf{L}_{\mathbf{x}}^{-1} \left(\mathbf{U}^T \mathbf{D} \sum_{j=1}^{N_i} \left(\phi_{ij} - \mu\right) \overline{\mathbf{y}}_i^T - N_i \mathbf{J} \mathrm{E}_{\mathbf{Y}} \left[\mathbf{y}_i \mathbf{y}_i^T\right] \right) \tag{C.157}$$

$$= \mathbf{L}_{\mathbf{x}}^{-1} \left(\mathbf{U}^T \mathbf{D} \left(\mathbf{C}_{\mathbf{y}} - \mu \sum_{i=1}^{M} N_i \overline{\mathbf{y}}_i^T \right) - \mathbf{J}\mathbf{R}_{\mathbf{y}} \right) \tag{C.158}$$

$$= \mathbf{L}_{\mathbf{x}}^{-1} \left(\mathbf{U}^T \mathbf{D} \left(\mathbf{C}_{\mathbf{y}} - \mu \mathbf{R}_{\mathbf{y1}}^T\right) - \mathbf{J}\mathbf{R}_{\mathbf{y}}\right) \tag{C.159}$$

$$= \mathbf{L}_{\mathbf{x}}^{-1} \left(\mathbf{U}^T \mathbf{D} \overline{\mathbf{C}}_y - \mathbf{J}\mathbf{R}_{\mathbf{y}}\right) \tag{C.160}$$

$$\mathbf{R}_{\mathbf{x}} = \sum_{i=1}^{M} \sum_{j=1}^{N_i} \mathrm{E}_{\mathbf{Y},\mathbf{X}} \left[\mathbf{x}_{ij} \mathbf{x}_{ij}^T\right] \tag{C.161}$$

$$= \sum_{i=1}^{M} \sum_{j=1}^{N_i} \mathrm{E}_{\mathbf{Y}} \left[\mathrm{E}_{\mathbf{X}|\mathbf{Y}} \left[\mathbf{x}_{ij} \mathbf{x}_{ij}^T\right]\right] \tag{C.162}$$

$$= \sum_{i=1}^{M} \sum_{j=1}^{N_i} \mathrm{E}_{\mathbf{Y}} \left[\mathbf{L}_{\mathbf{x}}^{-1} + \mathrm{E}_{\mathbf{X}|\mathbf{Y}} \left[\mathbf{x}_{ij}\right] \mathrm{E}_{\mathbf{X}|\mathbf{Y}} \left[\mathbf{x}_{ij}\right]^T\right] \tag{C.163}$$

$$= N\mathbf{L}_{\mathbf{x}}^{-1} + \mathbf{L}_{\mathbf{x}}^{-1} \sum_{i=1}^{M} \sum_{j=1}^{N_i} \mathrm{E}_{\mathbf{Y}} \left[\left(\mathbf{U}^T \mathbf{D} \left(\phi_{ij} - \mu\right) - \mathbf{J}\mathbf{y}_i\right) \left(\mathbf{U}^T \mathbf{D} \left(\phi_{ij} - \mu\right) - \mathbf{J}\mathbf{y}_i\right)^T \right] \mathbf{L}_{\mathbf{x}}^{-1} \tag{C.164}$$

$$= N\mathbf{L}_{\mathbf{x}}^{-1} + \mathbf{L}_{\mathbf{x}}^{-1} \left(\mathbf{U}^T \mathbf{D} \overline{\mathbf{S}} \mathbf{D} \mathbf{U} - \mathbf{U}^T \mathbf{D} \overline{\mathbf{C}}_{\mathbf{y}} \mathbf{J}^T - \mathbf{J} \overline{\mathbf{C}}_{\mathbf{y}}^T \mathbf{D} \mathbf{U} + \mathbf{J} \mathbf{R}_{\mathbf{y}} \mathbf{J}^T \right) \mathbf{L}_{\mathbf{x}}^{-1} \tag{C.165}$$

and

$$\overline{\mathbf{C}}_{\mathbf{y}} = \mathbf{C}_{\mathbf{y}} - \mu \mathbf{R}_{\mathbf{y1}}^T \, . \tag{C.166}$$

C.5.5.3 M-step MD

For the MD step, we assume an over-parametrized model with a general prior for the hidden variables:

$$P(\mathbf{y}_i) = \mathcal{N}\left(\mathbf{y}_i|\mu_{\mathbf{y}}, \mathbf{\Lambda}_{\mathbf{y}}^{-1}\right) \tag{C.167}$$

$$P(\mathbf{x}_{ij}|\mathbf{y}_i) = \mathcal{N}\left(\mathbf{x}_{ij}|\mathbf{H}\mathbf{y}_i + \mu_{\mathbf{x}}, \mathbf{\Lambda}_{\mathbf{x}}^{-1}\right)\ . \tag{C.168}$$

Now, we maximize the term of the EM auxiliary that depends on the new prior:

$$\mathcal{Q}(\mu_{\mathbf{y}}, \mathbf{\Lambda}_{\mathbf{y}}, \mathbf{H}, \mu_{\mathbf{x}}, \mathbf{\Lambda}_{\mathbf{x}}) = \sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[\ln \mathcal{N}\left(\mathbf{y}_i|\mu_{\mathbf{y}}, \mathbf{\Lambda}_{\mathbf{y}}^{-1}\right)\right] + \sum_{j=1}^{N_i} \mathrm{E}_{\mathbf{Y},\mathbf{X}}\left[\ln \mathcal{N}\left(\mathbf{x}_{ij}|\mathbf{H}\mathbf{y}_i + \mu_{\mathbf{x}}, \mathbf{\Lambda}_{\mathbf{x}}^{-1}\right)\right] \tag{C.169}$$

$$\begin{aligned} =&\frac{M}{2}\ln|\mathbf{\Lambda}_{\mathbf{y}}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{\Lambda}_{\mathbf{y}} \sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[(\mathbf{y}_i - \mu_{\mathbf{y}})(\mathbf{y}_i - \mu_{\mathbf{y}})^T\right]\right) + \frac{N}{2}\ln|\mathbf{\Lambda}_{\mathbf{x}}| \\ &- \frac{1}{2}\mathrm{tr}\left(\mathbf{\Lambda}_{\mathbf{x}} \sum_{i=1}^{M}\sum_{j=1}^{N_i} \mathrm{E}_{\mathbf{Y},\mathbf{X}}\left[(\mathbf{x}_{ij} - \mathbf{H}\mathbf{y}_i - \mu_{\mathbf{x}})(\mathbf{x}_{ij} - \mathbf{H}\mathbf{y}_i - \mu_{\mathbf{x}})^T\right]\right) \\ &+ \mathrm{const} \end{aligned} \tag{C.170}$$

We derive $\mathcal{Q}$ with respect to $\mu_{\mathbf{y}}$

$$\frac{\partial \mathcal{Q}(\mu_{\mathbf{y}}, \mathbf{\Lambda}_{\mathbf{y}}, \mathbf{H}, \mu_{\mathbf{x}}, \mathbf{\Lambda}_{\mathbf{x}})}{\partial \mu_{\mathbf{y}}} = \frac{1}{2}\sum_{i=1}^{M} \mathbf{\Lambda}_{\mathbf{y}} \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i - \mu_{\mathbf{y}}\right] = \mathbf{0} \quad \Longrightarrow \tag{C.171}$$

$$\mu_{\mathbf{y}} = \frac{1}{M}\sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]\ . \tag{C.172}$$

We derive $\mathcal{Q}$ with respect to $\mathbf{\Lambda}_{\mathbf{y}}$:

$$\frac{\partial \mathcal{Q}(\mu_{\mathbf{y}}, \mathbf{\Lambda}_{\mathbf{y}}, \mathbf{H}, \mu_{\mathbf{x}}, \mathbf{\Lambda}_{\mathbf{x}})}{\partial \mathbf{\Lambda}_{\mathbf{y}}} = \frac{M}{2}\left(2\mathbf{\Lambda}_{\mathbf{y}}^{-1} - \mathrm{diag}(\mathbf{\Lambda}_{\mathbf{y}}^{-1})\right) - \frac{1}{2}\left(2\mathbf{K} - \mathrm{diag}(\mathbf{K})\right) = \mathbf{0} \tag{C.173}$$

where $\mathbf{K} = \sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[(\mathbf{y}_i - \mu_{\mathbf{y}})(\mathbf{y}_i - \mu_{\mathbf{y}})^T\right]$, so

$$\mathbf{\Sigma}_{\mathbf{y}} = \mathbf{\Lambda}_{\mathbf{y}}^{-1} \tag{C.174}$$

$$= \frac{1}{M}\sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[(\mathbf{y}_i - \mu_{\mathbf{y}})(\mathbf{y}_i - \mu_{\mathbf{y}})^T\right] \tag{C.175}$$

$$= \frac{1}{M}\sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right] - \mu_{\mathbf{y}}\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]^T - \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]\mu_{\mathbf{y}}^T + \mu_{\mathbf{y}}\mu_{\mathbf{y}}^T \tag{C.176}$$

$$= \frac{1}{M}\sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right] - \mu_{\mathbf{y}}\mu_{\mathbf{y}}^T\ . \tag{C.177}$$

We derive $\mathcal{Q}$ with respect to $\mu_{\mathbf{x}}$:

$$\frac{\partial\mathcal{Q}(\mu_{\mathbf{y}},\mathbf{\Lambda}_{\mathbf{y}},\mathbf{H},\mu_{\mathbf{x}},\mathbf{\Lambda}_{\mathbf{x}})}{\partial\mu_{\mathbf{x}}} = \mathbf{\Lambda}_{\mathbf{x}}\sum_{i=1}^{M}\sum_{j=1}^{N_i}\mathrm{E}_{\mathbf{Y},\mathbf{X}}\left[\mathbf{x}_{ij}-\mathbf{H}\mathbf{y}_i-\mu_{\mathbf{x}}\right]=\mathbf{0} \quad \Longrightarrow \tag{C.178}$$

$$\mu_{\mathbf{x}} = \frac{1}{N}\left(\sum_{i=1}^{M}\sum_{j=1}^{N_i}\mathrm{E}_{\mathbf{X}}\left[\mathbf{x}_{ij}\right]-\mathbf{H}\sum_{i=1}^{M}N_i\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]\right) \tag{C.179}$$

$$= \frac{1}{N}\left(\mathbf{R}_{\mathbf{x1}}-\mathbf{H}\mathbf{R}_{\mathbf{y1}}\right)\ . \tag{C.180}$$

We derive $\mathcal{Q}$ with respect to $\mathbf{H}$:

$$\frac{\partial\mathcal{Q}(\mu_{\mathbf{y}},\mathbf{\Lambda}_{\mathbf{y}},\mathbf{H},\mu_{\mathbf{x}},\mathbf{\Lambda}_{\mathbf{x}})}{\partial\mathbf{H}} = \mathbf{\Lambda}_{\mathbf{x}}\sum_{i=1}^{M}\sum_{j=1}^{N_i}\mathrm{E}_{\mathbf{Y},\mathbf{X}}\left[\left(\mathbf{x}_{ij}-\mathbf{H}\mathbf{y}_i-\mu_{\mathbf{x}}\right)\mathbf{y}_i^T\right]=\mathbf{0} \tag{C.181}$$

$$\Longrightarrow \mathbf{R}_{\mathbf{xy}}-\mathbf{H}\mathbf{R}_{\mathbf{y}}-\mu_{\mathbf{x}}\mathbf{R}_{1y}=\mathbf{0} \tag{C.182}$$

$$\Longrightarrow \mathbf{R}_{\mathbf{xy}}-\mathbf{H}\mathbf{R}_{\mathbf{y}}-\frac{1}{N}\left(\mathbf{R}_{\mathbf{x1}}-\mathbf{H}\mathbf{R}_{\mathbf{y1}}\right)\mathbf{R}_{1y}=\mathbf{0} \tag{C.183}$$

$$\Longrightarrow \mathbf{R}_{\mathbf{xy}}-\frac{1}{N}\mathbf{R}_{\mathbf{x1}}\mathbf{R}_{1y}-\mathbf{H}\left(\mathbf{R}_{\mathbf{y}}-\frac{1}{N}\mathbf{R}_{\mathbf{y1}}\mathbf{R}_{1y}\right)=\mathbf{0} \quad \Longrightarrow \tag{C.184}$$

$$\mathbf{H}=\left(\mathbf{R}_{\mathbf{xy}}-\frac{1}{N}\mathbf{R}_{\mathbf{x1}}\mathbf{R}_{1y}\right)\left(\mathbf{R}_{\mathbf{y}}-\frac{1}{N}\mathbf{R}_{\mathbf{y1}}\mathbf{R}_{1y}\right)^{-1}\ . \tag{C.185}$$

Finally, we derive $\mathcal{Q}$ with respect to $\mathbf{\Lambda}_{\mathbf{x}}$:

$$\frac{\partial\mathcal{Q}(\mu_{\mathbf{y}},\mathbf{\Lambda}_{\mathbf{y}},\mathbf{H},\mu_{\mathbf{x}},\mathbf{\Lambda}_{\mathbf{x}})}{\partial\mathbf{\Lambda}_{\mathbf{x}}} = \frac{N}{2}\left(2\mathbf{\Lambda}_{\mathbf{x}}^{-1}-\mathrm{diag}(\mathbf{\Lambda}_{\mathbf{x}}^{-1})\right)-\frac{1}{2}\left(2\mathbf{K}-\mathrm{diag}(\mathbf{K})\right)=\mathbf{0} \tag{C.186}$$

where $\mathbf{K}=\sum_{i=1}^{M}\sum_{j=1}^{N_i}\mathrm{E}_{\mathbf{Y},\mathbf{X}}\left[\left(\mathbf{x}_{ij}-\mathbf{H}\mathbf{y}_i-\mu_{\mathbf{x}}\right)\left(\mathbf{x}_{ij}-\mathbf{H}\mathbf{y}_i-\mu_{\mathbf{x}}\right)^T\right]$, so

$$\mathbf{\Sigma}_{\mathbf{x}} = \mathbf{\Lambda}_{\mathbf{x}}^{-1} \tag{C.187}$$

$$\begin{aligned}= &\frac{1}{N}\left(\mathbf{R}_{\mathbf{x}}-\mathbf{R}_{\mathbf{xy}}\mathbf{H}^T-\mathbf{H}\mathbf{R}_{\mathbf{xy}}^T-\mathbf{R}_{\mathbf{x1}}\mu_{\mathbf{x}}^T-\mu_{\mathbf{x}}\mathbf{R}_{\mathbf{x1}}^T+\mathbf{H}\mathbf{R}_{\mathbf{y}}\mathbf{H}^T\right.\\ &\left.+\mathbf{H}\mathbf{R}_{\mathbf{y1}}\mu_{\mathbf{x}}^T+\mu_{\mathbf{x}}\mathbf{R}_{\mathbf{y1}}^T\mathbf{H}^T+N\mu_{\mathbf{x}}\mu_{\mathbf{x}}^T\right)\end{aligned} \tag{C.188}$$

$$\begin{aligned}= &\frac{1}{N}\left(\mathbf{R}_{\mathbf{x}}-\mathbf{R}_{\mathbf{xy}}\mathbf{H}^T-\mathbf{H}\mathbf{R}_{\mathbf{xy}}^T+\mathbf{H}\mathbf{R}_{\mathbf{y}}\mathbf{H}^T\right.\\ &\left.-\left(\mathbf{R}_{\mathbf{x1}}-\mathbf{H}\mathbf{R}_{\mathbf{y1}}\right)\mu_{\mathbf{x}}^T-\mu_{\mathbf{x}}\left(\mathbf{R}_{\mathbf{x1}}-\mathbf{H}\mathbf{R}_{\mathbf{y1}}\right)^T+N\mu_{\mathbf{x}}\mu_{\mathbf{x}}^T\right)\end{aligned} \tag{C.189}$$

$$= \frac{1}{N}\left(\mathbf{R}_{\mathbf{x}}-\mathbf{R}_{\mathbf{xy}}\mathbf{H}^T-\mathbf{H}\mathbf{R}_{\mathbf{xy}}^T+\mathbf{H}\mathbf{R}_{\mathbf{y}}\mathbf{H}^T-\left(\mathbf{R}_{\mathbf{x1}}-\mathbf{H}\mathbf{R}_{\mathbf{y1}}\right)\mu_{\mathbf{x}}^T\right) \tag{C.190}$$

The transform $(\mathbf{y},\mathbf{x})=\psi(\mathbf{y}',\mathbf{x}')$ such as $\mathbf{y}'$ and $\mathbf{x}'$ has a standard prior is

$$\mathbf{y} = \mu_{\mathbf{y}}+(\mathbf{\Sigma}_{\mathbf{y}}^{1/2})^T\mathbf{y}' \tag{C.191}$$

$$\mathbf{x} = \mu_{\mathbf{x}}+\mathbf{H}\mathbf{y}+(\mathbf{\Sigma}_{\mathbf{x}}^{1/2})^T\mathbf{x}' \tag{C.192}$$

$$= \mu_{\mathbf{x}}+\mathbf{H}\mu_{\mathbf{y}}+\mathbf{H}(\mathbf{\Sigma}_{\mathbf{y}}^{1/2})^T\mathbf{y}'+(\mathbf{\Sigma}_{\mathbf{x}}^{1/2})^T\mathbf{x}' \tag{C.193}$$

where $\mathbf{\Sigma}^{1/2}$ denotes the upper triangular Cholesky decomposition of $\mathbf{\Sigma}$.

The effect of the new prior can be absorbed into μ, $\mathbf{V}$ and $\mathbf{U}$ by applying that transform:

$$\mathbf{U}' = \mathbf{U}(\mathbf{\Sigma}_{\mathbf{x}}^{-1/2})^T \tag{C.194}$$

$$\mathbf{V}' = (\mathbf{V} + \mathbf{U}\mathbf{H})\,(\mathbf{\Sigma}_{\mathbf{y}}^{-1/2})^T \tag{C.195}$$

$$\mu' = \mu + (\mathbf{V} + \mathbf{U}\mathbf{H})\,\mu_{\mathbf{y}} + \mathbf{U}\mu_{\mathbf{x}}\ . \tag{C.196}$$

C.5.5.4 Objective function

Convergence can be checked by tracking the marginal likelihood of the data. This consists in accumulate Equation (C.129) for all speakers:

$$\begin{aligned}\ln P\left(\mathbf{\Phi}|\mathcal{M}\right) =& \frac{N}{2}\ln\left|\frac{\mathbf{D}}{2\pi}\right| - \frac{1}{2}\mathrm{tr}\left(\left(\mathbf{D} - \mathbf{D}\mathbf{U}\mathbf{L}_{\mathbf{x}}^{-1}\mathbf{U}^T\mathbf{D}\right)\overline{\mathbf{S}}\right) - \frac{N}{2}\ln|\mathbf{L}_{\mathbf{x}}| \\ & - \frac{1}{2}\sum_{i=1}^{M}\ln|\mathbf{L}_{\mathbf{y}_i}| + \frac{1}{2}\sum_{i=1}^{M}\gamma_i^T\mathbf{L}_{\mathbf{y}_i}^{-1}\gamma_i\ .\end{aligned} \tag{C.197}$$

Appendix D

EM for Multi-channel SPLDA

D.1 Introduction

In this appendix, we derive the equations need to estimate the parameters of the multi-channel SPLDA (MCSPLDA) model described in Chapter 8. MCSPLDA is a variant of SPLDA where the within-class covariance that we use to model each i-vector depends on the type of channel where it comes from. For example, we could have a within-class covariance for telephone recordings and another one for far-field microphone recordings.

D.2 Model Description

Multi-channel SPLDA is a linear generative model where an i-vector ϕ_{ij} of speaker i is written as

$$\phi_{ij} = \mathbf{V}\mathbf{y}_i + \epsilon_{ij} \tag{D.1}$$

where $\mathbf{V}$ is the eigenvoices matrix, $\mathbf{y}_i$ is the speaker factor vector, and ϵ_{ij} is a channel offset. We introduce a variable $\mathbf{z}_{ij}$ that indicates the type of channel that generates ϵ_{ij}. $\mathbf{z}_{ij}$ is a 1-of-K binary vector with elements z_{ijk} for $k = 1, \ldots, K$. We assume that some kind of channel detector provides the type of channel for each speaker or at least the probability $P\left(z_{ijk}\right)$ for it.

We put the following priors on the variables:

$$\mathbf{y}_i \sim \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \mathbf{I}\right) \tag{D.2}$$

$$\epsilon_{ij}|z_{ijk} = 1 \sim \mathcal{N}\left(\epsilon_{ij}|\mu_k, \mathbf{W}_k^{-1}\right) \tag{D.3}$$

where $\mathcal{N}$ denotes a Gaussian distribution; μ_k is a channel dependent mean; and $\mathbf{W}_k$ is the channel dependent within class precision matrix. We define $\mu = \{\mu_k\}_{k=1}^K$ and $\mathbf{W} = \{\mathbf{W}_k\}_{k=1}^K$. The denote by $\mathcal{M} = (\mu, \mathbf{V}, \mathbf{W})$ the set of all the model parameters.

D.3 Definitions of Sufficient Statistics

We define channel dependent sufficient statistics for speaker i as:

$$N_{ik} = \sum_{j=1}^{N_i} P\left(z_{ijk} = 1\right) \tag{D.4}$$

$$\mathbf{F}_{ik} = \sum_{j=1}^{N_i} P\left(z_{ijk} = 1\right) \phi_{ij} \tag{D.5}$$

$$\mathbf{S}_{ik} = \sum_{j=1}^{N_i} P\left(z_{ijk} = 1\right) \phi_{ij}\phi_{ij}^T \ . \tag{D.6}$$

We define the channel centered statistics as

$$\overline{\mathbf{F}}_{ik} = \mathbf{F}_{ik} - N_{ik}\mu_k \tag{D.7}$$

$$\overline{\mathbf{S}}_{ik} = \sum_{j=1}^{N_i} P\left(z_{ijk} = 1\right) \left(\phi_{ij} - \mu_k\right) \left(\phi_{ij} - \mu_k\right)^T = \mathbf{S}_{ik} - \mu_k \mathbf{F}_{ik}^T - \mathbf{F}_{ik}\mu_k^T + N_{ik}\mu_k\mu_k^T \ . \tag{D.8}$$

Finally, we define the global statistics:

$$N_k = \sum_{i=1}^{M} N_{ik} \tag{D.9}$$

$$\mathbf{F}_k = \sum_{i=1}^{M} \mathbf{F}_{ik} \tag{D.10}$$

$$\overline{\mathbf{F}}_k = \sum_{i=1}^{M} \overline{\mathbf{F}}_{ik} \tag{D.11}$$

$$\mathbf{S}_k = \sum_{i=1}^{M} \mathbf{S}_{ik} \tag{D.12}$$

$$\overline{\mathbf{S}}_k = \sum_{i=1}^{M} \overline{\mathbf{S}}_{ik} \ . \tag{D.13}$$

D.4 Data Conditional Likelihood

The likelihood of the data given the hidden variables for speaker i is

$$\begin{aligned}\ln P\left(\mathbf{\Phi}_i|\mathbf{y}_i,\mathbf{z}_i,\mathcal{M}\right) =& \sum_{j=1}^{N_i}\sum_{k=1}^{K} P\left(z_{ijk}=1\right)\ln\mathcal{N}\left(\phi_{ij}|\mu_k+\mathbf{V}\mathbf{y}_i,\mathbf{W}_k^{-1}\right) \qquad \text{(D.14)}\\ =& \sum_{k=1}^{K}\frac{N_{ik}}{2}\ln\left|\frac{\mathbf{W}_k}{2\pi}\right| \\ & -\frac{1}{2}\sum_{j=1}^{N_i} P\left(z_{ijk}=1\right)\left(\phi_{ij}-\mu_k-\mathbf{V}\mathbf{y}_i\right)^T\mathbf{W}_k(\phi_{ij}-\mu_k-\mathbf{V}\mathbf{y}_i) \qquad \text{(D.15)}\\ =& \sum_{k=1}^{K}\frac{N_{ik}}{2}\ln\left|\frac{\mathbf{W}_k}{2\pi}\right|-\frac{1}{2}\mathrm{tr}\left(\mathbf{W}_k\overline{\mathbf{S}}_{ik}\right)+\mathbf{y}_i^T\mathbf{V}^T\mathbf{W}_k\overline{\mathbf{F}}_{ik}-\frac{N_{ik}}{2}\mathbf{y}_i^T\mathbf{V}^T\mathbf{W}_k\mathbf{V}\mathbf{y}_i \\ & \qquad \text{(D.16)}\end{aligned}$$

We can write this likelihood in another form, useful for the M-step derivations:

$$\begin{aligned}\ln P\left(\mathbf{\Phi}_i|\mathbf{y}_i,\mathbf{z}_i,\mathcal{M}\right) =& \sum_{k=1}^{K}\frac{N_{ik}}{2}\ln\left|\frac{\mathbf{W}_k}{2\pi}\right|-\frac{1}{2}\mathrm{tr}\Big(\mathbf{W}_k\Big(\mathbf{S}_{ik}-2\mathbf{F}_{ik}\mu_k^T+N_{ik}\mu_k\mu_k^T \\ & -2\left(\mathbf{F}_{ik}-N_{ki}\mu_k\right)\mathbf{y}_i^T\mathbf{V}^T+N_{ik}\mathbf{V}\mathbf{y}_i\mathbf{y}_i^T\mathbf{V}^T\Big)\Big)\ . \end{aligned} \tag{D.17}$$

D.5 Posterior of the Hidden Variables

The posterior of $\mathbf{y}_i$ is given by

$$P\left(\mathbf{y}_i|\mathbf{\Phi}_i,\mathbf{z}_i,\mathcal{M}\right)=\frac{P\left(\mathbf{\Phi}_i|\mathbf{y}_i,\mathbf{z}_i,\mathcal{M}\right)P\left(\mathbf{y}_i\right)}{P\left(\mathbf{\Phi}_i|\mathbf{z}_i,\mathcal{M}\right)}\ . \tag{D.18}$$

Taking (D.2) and (D.16), we obtain:

$$\begin{aligned}\ln P\left(\mathbf{y}_i|\mathbf{\Phi}_i,\mathbf{z}_i,\mathcal{M}\right) =& \ln P\left(\mathbf{\Phi}_i|\mathbf{y}_i,\mathbf{z}_i,\mathcal{M}\right)+\ln P\left(\mathbf{y}_i\right)+\mathrm{const} \qquad \text{(D.19)}\\ =& \sum_{k=1}^{K}\mathbf{y}_i^T\mathbf{V}^T\mathbf{W}_k\overline{\mathbf{F}}_{ik}-\frac{N_{ik}}{2}\mathbf{y}_i^T\mathbf{V}^T\mathbf{W}_k\mathbf{V}\mathbf{y}_i-\frac{1}{2}\mathbf{y}_i^T\mathbf{y}_i+\mathrm{const} \qquad \text{(D.20)}\\ =& \mathbf{y}_i^T\sum_{k=1}^{K}\mathbf{V}^T\mathbf{W}_k\overline{\mathbf{F}}_{ik}-\frac{1}{2}\mathbf{y}_i^T\left(\mathbf{I}+\sum_{k=1}^{K}N_{ik}\mathbf{V}^T\mathbf{W}_k\mathbf{V}\right)\mathbf{y}_i+\mathrm{const} \qquad \text{(D.21)}\end{aligned}$$

Equation (D.21) has the form of a Gaussian distribution so, finally, we obtain:

$$P\left(\mathbf{y}_i|\mathbf{\Phi}_i,\mathbf{z}_i,\mathcal{M}\right)=\mathcal{N}\left(\mathbf{y}_i|\mathbf{L}_i^{-1}\gamma_i,\mathbf{L}_i^{-1}\right) \tag{D.22}$$

where

$$\mathbf{L}_i = \mathbf{I}+\sum_{k=1}^{K}N_{ik}\mathbf{V}^T\mathbf{W}_k\mathbf{V} \tag{D.23}$$

$$\gamma_i = \sum_{k=1}^{K}\mathbf{V}^T\mathbf{W}_k\overline{\mathbf{F}}_{ik}\,. \tag{D.24}$$

D.6 Marginal Likelihood of the Data

The marginal likelihood of the data for speaker i is

$$P\left(\mathbf{\Phi}_i|\mathbf{z}_i,\mathcal{M}\right)=\frac{P\left(\mathbf{\Phi}_i|\mathbf{y}_0,\mathbf{z}_i,\mathcal{M}\right)P\left(\mathbf{y}_0\right)}{P\left(\mathbf{y}_0|\mathbf{\Phi}_i,\mathbf{z}_i,\mathcal{M}\right)} \tag{D.25}$$

where $\mathbf{y}_0$ can be whatever as long as the denominator is not zero. By taking (D.16), (D.2) and (D.22); and $\mathbf{y}_0=\mathbf{0}$, we obtain:

$$\ln P\left(\mathbf{\Phi}_i|\mathbf{z}_i,\mathcal{M}\right)=\sum_{k=1}^{K}\frac{N_{ik}}{2}\ln\left|\frac{\mathbf{W}_k}{2\pi}\right|-\frac{1}{2}\sum_{k=1}^{K}\mathrm{tr}\left(\mathbf{W}_k\overline{\mathbf{S}}_{ik}\right)-\frac{1}{2}\ln\left|\mathbf{L}_i\right|+\frac{1}{2}\gamma_i^T\mathbf{L}_i^{-1}\gamma_i\ . \tag{D.26}$$

D.7 EM algorithm

D.7.1 E-step

In the E-step we calculate the posterior of $\mathbf{y}$ with (D.22).

D.7.2 M-step ML

We maximize the EM auxiliary function $\mathcal{Q}(\mathcal{M})$:

$$\mathcal{Q}(\mathcal{M})=\sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}\left[\ln P\left(\mathbf{\Phi}_i,\mathbf{y}_i|\mathbf{z}_i,\mathcal{M}\right)\right] \tag{D.27}$$

$$=\sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}\left[\ln P\left(\mathbf{\Phi}_i|\mathbf{y}_i,\mathbf{z}_i,\mathcal{M}\right)\right]+\mathrm{E}_{\mathbf{Y}}\left[\ln P\left(\mathbf{y}_i\right)\right]\ . \tag{D.28}$$

If we plug-in (D.17), we obtain:

$$\begin{aligned}\mathcal{Q}(\mathcal{M})=&\sum_{k=1}^{K}\frac{N_k}{2}\ln\left|\mathbf{W}_k\right|-\frac{1}{2}\mathrm{tr}\left(\mathbf{W}_k\sum_{i=1}^{M}\left(\mathbf{S}_{ik}-2\mathbf{F}_{ik}\mu_k^T+N_{ik}\mu_k\mu_k^T\right.\right.\\&\left.\left.-2\left(\mathbf{F}_{ik}-N_{ki}\mu_k\right)\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i^T\right]\mathbf{V}^T+N_{ik}\mathbf{V}\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right]\mathbf{V}^T\right)\right)+\text{const}\end{aligned} \tag{D.29}$$

$$\begin{aligned}=&\sum_{k=1}^{K}\frac{N_k}{2}\ln\left|\mathbf{W}_k\right|\\&-\frac{1}{2}\mathrm{tr}\left(\mathbf{W}_k\left(\mathbf{S}_k-2\mathbf{F}_k\mu_k^T+N_k\mu_k\mu_k^T\right.\right.\\&-2\left(\sum_{i=1}^{M}\mathbf{F}_{ik}\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]^T\right)\mathbf{V}^T+2\mu_k\left(\sum_{i=1}^{M}N_{ki}\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]^T\right)\mathbf{V}^T\\&\left.\left.+\mathbf{V}\left(\sum_{i=1}^{M}N_{ik}\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right]\right)\mathbf{V}^T\right)\right)+\text{const}\ .\end{aligned} \tag{D.30}$$

Now, we find convenient to define

$$\mathbf{R}_{\mathbf{y}k} = \sum_{i=1}^{M} N_{ik} \mathrm{E}_{\mathbf{Y}} \left[\mathbf{y}_i \mathbf{y}_i^T \right] \tag{D.31}$$

$$\mathbf{A}_k = \sum_{i=1}^{M} N_{ik} \mathrm{E}_{\mathbf{Y}} \left[\mathbf{y}_i \right] \tag{D.32}$$

$$\mathbf{C}_k = \sum_{i=1}^{M} \mathbf{F}_{ik} \mathrm{E}_{\mathbf{Y}} \left[\mathbf{y}_i \right]^T \ . \tag{D.33}$$

Thus, we can write the EM objective in a more compact form:

$$\begin{aligned}\mathcal{Q}(\mathcal{M}) = & \sum_{k=1}^{K} \frac{N_k}{2} \ln |\mathbf{W}_k| - \frac{1}{2} \mathrm{tr} \left(\mathbf{W}_k \left(\mathbf{S}_k - 2\mathbf{F}_k \mu_k^T + N_k \mu_k \mu_k^T \right.\right. \\ & \left.\left. -2\mathbf{C}_k \mathbf{V}^T + 2\mu_k \mathbf{A}_k^T \mathbf{V}^T + \mathbf{V} \mathbf{R}_{\mathbf{y}k} \mathbf{V}^T \right) \right) + \mathrm{const}\end{aligned} \tag{D.34}$$

We derive $\mathcal{Q}$ with respect to μ_k:

$$\frac{\partial \mathcal{Q}(\mathcal{M})}{\partial \mu_k} = \mathbf{W}_k \left(\mathbf{F}_k - N_k \mu_k - \mathbf{V} \mathbf{A}_k \right) = \mathbf{0} \tag{D.35}$$

$$\Longrightarrow \mu_k = \frac{1}{N_k} \left(\mathbf{F}_k - \mathbf{V} \mathbf{A}_k \right) \ . \tag{D.36}$$

We derive $\mathcal{Q}$ with respect $\mathbf{V}$:

$$\frac{\partial \mathcal{Q}(\mathcal{M})}{\partial \mathbf{V}} = \sum_{k=1}^{K} \mathbf{W}_k \left(\mathbf{C}_k - \mu_k \mathbf{A}_k^T - \mathbf{V} \mathbf{R}_{\mathbf{y}k} \right) = \mathbf{0} \ . \tag{D.37}$$

We substitute (D.36) into (D.37):

$$\sum_{k=1}^{K} \mathbf{W}_k \left(\mathbf{C}_k - \frac{1}{N_k} \left(\mathbf{F}_k - \mathbf{V} \mathbf{A}_k \right) \mathbf{A}_k^T - \mathbf{V} \mathbf{R}_{\mathbf{y}k} \right) = \mathbf{0} \tag{D.38}$$

$$\Longrightarrow \sum_{k=1}^{K} \mathbf{W}_k \mathbf{V} \left(\mathbf{R}_{\mathbf{y}k} - \frac{1}{N_k} \mathbf{A}_k \mathbf{A}_k^T \right) = \sum_{k=1}^{K} \mathbf{W}_k \left(\mathbf{C}_k - \frac{1}{N_k} \mathbf{F}_k \mathbf{A}_k^T \right) \tag{D.39}$$

Additionally, we define

$$\mathbf{B}_k = \mathbf{R}_{\mathbf{y}k} - \frac{1}{N_k} \mathbf{A}_k \mathbf{A}_k^T \tag{D.40}$$

$$\mathbf{D} = \sum_{k=1}^{K} \mathbf{W}_k \left(\mathbf{C}_k - \frac{1}{N_k} \mathbf{F}_k \mathbf{A}_k^T \right) \ . \tag{D.41}$$

Thus, we obtain that to compute $\mathbf{V}$ we need to solve:

$$\sum_{k=1}^{K} \mathbf{W}_k \mathbf{V} \mathbf{B}_k = \mathbf{D} \tag{D.42}$$

To solve this equation we have to apply the property of the Kronecker product $(\mathbf{B}^T \otimes \mathbf{A})\mathrm{vec}(\mathbf{X}) = \mathrm{vec}(\mathbf{AXB})$. Finally, we have to solve the linear system of equations:

$$\sum_{k=1}^{K} \left(\mathbf{B}_k^T \otimes \mathbf{W}_k\right) \mathrm{vec}(\mathbf{V}) = \mathrm{vec}(\mathbf{D}) \tag{D.43}$$

For $\mathbf{W}_k$, we have that

$$\frac{\partial \mathcal{Q}(\mathcal{M})}{\partial \mathbf{W}_k} = \frac{N_k}{2}\left(2\mathbf{W}_k^{-1} - \mathrm{diag}(\mathbf{W}_k^{-1})\right) - \frac{1}{2}\left(\mathbf{K} + \mathbf{K}^T - \mathrm{diag}(\mathbf{K})\right) = \mathbf{0} \tag{D.44}$$

where

$$\mathbf{K} = \mathbf{S}_k - 2\mathbf{F}_k\mu_k^T + N_k\mu_k\mu_k^T - 2\mathbf{C}_k\mathbf{V}^T + 2\mu_k\mathbf{A}_k^T\mathbf{V}^T + \mathbf{V}\mathbf{R}_{\mathbf{y}k}\mathbf{V}^T \ . \tag{D.45}$$

Then, isolating $\mathbf{W}_k$:

$$\begin{aligned}\mathbf{W}_k^{-1} =& \frac{1}{N_k}\frac{\mathbf{K}+\mathbf{K}^T}{2} \qquad\qquad (\mathrm{D.46})\\ =& \frac{1}{N_k}\left(\mathbf{S}_k - \mathbf{F}_k\mu_k^T - \mu_k\mathbf{F}_k^T + N_k\mu_k\mu_k^T\right.\\ & \left.-\mathbf{C}_k\mathbf{V}^T - \mathbf{V}\mathbf{C}_k^T + \mu_k\mathbf{A}_k^T\mathbf{V}^T + \mathbf{V}\mathbf{A}_k\mu_k^T + \mathbf{V}\mathbf{R}_{\mathbf{y}k}\mathbf{V}^T\right) \ . \qquad (\mathrm{D.47})\end{aligned}$$

We have to update iteratively μ_k, $\mathbf{V}$ and $\mathbf{W}_k$.

D.7.3 M-step MD

In the minimum divergence step, we assume a general prior for $\mathbf{y}$, instead of a standard normal prior:

$$P(\mathbf{y}) = \mathcal{N}\left(\mathbf{y}|\mu_{\mathbf{y}}, \mathbf{\Lambda}_{\mathbf{y}}^{-1}\right) \tag{D.48}$$

To obtain the optimum values for $\mu_{\mathbf{y}}$ and $\mathbf{\Lambda}_{\mathbf{y}}$, we maximize the term of the EM auxiliary that corresponds to the prior:

$$\begin{aligned}\mathcal{Q}(\mu_{\mathbf{y}}, \mathbf{\Lambda}_{\mathbf{y}}) =& \sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[\ln \mathcal{N}\left(\mathbf{y}_i|\mu_{\mathbf{y}}, \mathbf{\Lambda}_{\mathbf{y}}^{-1}\right)\right] \qquad (\mathrm{D.49})\\ =& \frac{M}{2}\ln|\mathbf{\Lambda}_{\mathbf{y}}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{\Lambda}_{\mathbf{y}} \sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[(\mathbf{y}_i - \mu_{\mathbf{y}})(\mathbf{y}_i - \mu_{\mathbf{y}})^T\right]\right) + \mathrm{const} \ . \qquad (\mathrm{D.50})\end{aligned}$$

We derivate with respect to μ_y:

$$\frac{\partial \mathcal{Q}(\mu_{\mathbf{y}}, \mathbf{\Lambda}_{\mathbf{y}})}{\partial \mu_{\mathbf{y}}} = \frac{1}{2}\sum_{i=1}^{M} \mathbf{\Lambda}_{\mathbf{y}} \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i - \mu_{\mathbf{y}}\right] = \mathbf{0} \quad \Longrightarrow \tag{D.51}$$

$$\mu_{\mathbf{y}} = \frac{1}{M}\sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right] \ . \tag{D.52}$$

We derivate with respect to $\boldsymbol{\Lambda}_{\mathbf{y}}^{-1}$:

$$\frac{\partial \mathcal{Q}(\mu_{\mathbf{y}}, \boldsymbol{\Lambda}_{\mathbf{y}})}{\partial \boldsymbol{\Lambda}_{\mathbf{y}}^{-1}} = \frac{M}{2}\left(2\boldsymbol{\Lambda}_{\mathbf{y}}^{-1} - \mathrm{diag}(\boldsymbol{\Lambda}_{\mathbf{y}}^{-1})\right) - \frac{1}{2}\left(2\mathbf{S} - \mathrm{diag}(\mathbf{S})\right) = \mathbf{0} \tag{D.53}$$

where $\mathbf{S} = \sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[(\mathbf{y}_i - \mu_{\mathbf{y}})(\mathbf{y}_i - \mu_{\mathbf{y}})^T\right]$, so

$$\boldsymbol{\Sigma}_{\mathbf{y}} = \boldsymbol{\Lambda}_{\mathbf{y}}^{-1} = \frac{1}{M}\sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[(\mathbf{y}_i - \mu_{\mathbf{y}})(\mathbf{y}_i - \mu_{\mathbf{y}})^T\right] = \frac{1}{M}\sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right] - \mu_{\mathbf{y}}\mu_{\mathbf{y}}^T \; . \tag{D.54}$$

To minimize the divergence between the general prior and the standard normal prior, first, we have to find a transform $\mathbf{y} = \psi(\mathbf{y}')$ such as $\mathbf{y}'$ has a standard prior. That is

$$\mathbf{y} = \mu_{\mathbf{y}} + (\boldsymbol{\Sigma}_{\mathbf{y}}^{1/2})^T \mathbf{y}' \; . \tag{D.55}$$

This transform can be used to make μ_k and $\mathbf{V}$ to absorb the effect of the non-standard prior:

$$\mu_k' = \mu_k + \mathbf{V}\mu_{\mathbf{y}} \tag{D.56}$$

$$\mathbf{V}' = \mathbf{V}(\boldsymbol{\Sigma}_{\mathbf{y}}^{1/2})^T \; , \tag{D.57}$$

where $\boldsymbol{\Sigma}_{\mathbf{y}}^{1/2}$ is the upper triangular Cholesky decomposition.

D.7.4 Objective function

We can check the convergence of the EM algorithm by evaluating the marginal likelihood of the training data, i.e., it is Equation (D.26) accumulated for all the speakers

$$\ln P\left(\boldsymbol{\Phi}|\mathcal{M}\right) = \sum_{k=1}^{K} \frac{N_k}{2} \ln\left|\frac{\mathbf{W}_k}{2\pi}\right| - \frac{1}{2}\sum_{k=1}^{K} \mathrm{tr}\left(\mathbf{W}_k\overline{\mathbf{S}}_k\right) - \frac{1}{2}\sum_{i=1}^{M} \ln|\mathbf{L}_i| + \frac{1}{2}\sum_{i=1}^{M} \gamma_i^T \mathbf{L}_i^{-1}\gamma_i \; . \tag{D.58}$$

Appendix E

Variational Inference for Bayesian Two-Covariance Model

E.1 Introduction

In this appendix, we derive the variational Bayes solution of the Bayesian two-covariance model used in Chapters 9 and 10. The difference between the fully Bayesian and the classical versions of the model consists in that the fully Bayesian version assumes that the model parameters are hidden variables with priors over them while in the standard version they are deterministic variables. While deterministic variables are generally derived by maximum likelihood estimation, for hidden variables, we can compute posterior distributions. Posterior distributions present the advantage that they include information regarding the uncertainty about the value of the parameters.

E.2 The Bayesian Two-Covariance Model

Figure E.1 depicts the graphical model of our Bayesian two-covariance model. As we have already explained in previous chapters, the model decomposes each i-vector ϕ_{ij} of the session j of speaker i as:

$$\phi_{ij} = \mathbf{y}_i + \epsilon_{ij} \tag{E.1}$$

where $\mathbf{y}_i$ is the speaker identity variable and ϵ_{ij} is a channel offset. We assume Gaussian priors for the speaker and channel spaces:

$$\mathbf{y}_i \sim \mathcal{N}\left(\mathbf{y}_i|\mu, \mathbf{B}^{-1}\right) \tag{E.2}$$

$$\epsilon_{ij} \sim \mathcal{N}\left(\epsilon_{ij}|\mathbf{0}, \mathbf{W}^{-1}\right) \tag{E.3}$$

where μ is the speaker independent mean; $\mathbf{B}$ is the between class precision matrix, and $\mathbf{W}$ is the within class precision matrix. We denote by $\mathcal{M} = (\mu, \mathbf{B}, \mathbf{W})$ the set of all the parameters of the model and by $\mathcal{M} = (\mu, \mathbf{B})$ the parameters of speaker space.

In the Bayesian version of the model, we have the priors over the model parameters, $\Pi_{\mathcal{M}_\mathbf{y}}$ and $\Pi_\mathbf{W}$. We denote by $\Pi = \left(\Pi_{\mathcal{M}_\mathbf{y}}, \Pi_\mathbf{W}\right)$ the set of model priors.

Additionally, we denote by $\mathbf{\Phi}$ the set of all the i-vectors in the database, by $\mathbf{\Phi}_i$ the i-vectors of the i^{th} speaker, $\mathbf{Y}$ the set of all speaker variables of the dataset, and by θ the

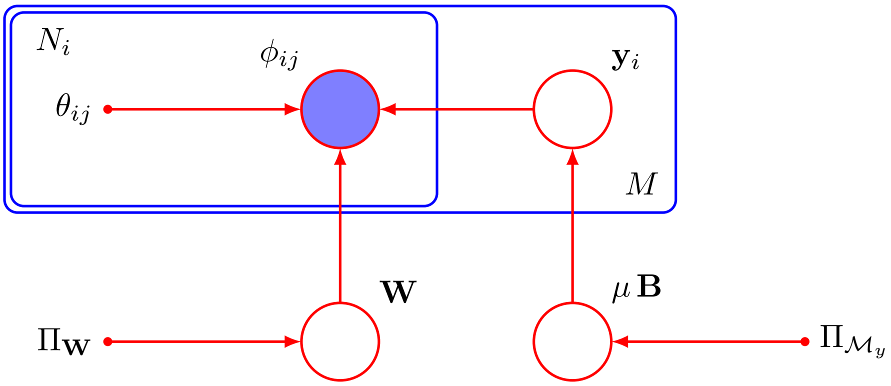


Figure E.1: Graphical model of the Bayesian two-covariance model.

labels that indicate the assignations of i-vectors to speakers. We assume that we have M speakers with N_i i-vectors per speaker. The total number of i-vector is $N = \sum_{i=1}^{M} N_i$.

Computing the posterior distribution of the hidden variables of the fully Bayesian model in close form is non-tractable. We approximated the posteriors by applying variational Bayes (VB) inference [Bishop, 2006]. We present equations for the case where we assume non-informative priors and for the case with informative conjugate priors.

E.3 Variational Inference with Non-Informative Priors

E.3.1 Model priors

A non-informative prior (Jeffreys prior) encodes the absence of information about μ, $\mathbf{B}$ and $\mathbf{W}$ other than the training data. With a non-informative prior no Gaussian should be preferred over others and it should be invariant to any translation or scaling of the measurement space. We introduced the non-informative priors for the parameters of Gaussian distributions in Appendix A. We can use the same kind of prior for the Gaussian distributions of the two-covariance model. Then, model prior can be written as:

$$P\left(\mathcal{M}|\Pi\right) = P\left(\mu, \mathbf{B}|\Pi_{\mathcal{M}_\mathbf{y}}\right) P\left(\mathbf{W}|\Pi_\mathbf{W}\right) \tag{E.4}$$

where

$$\begin{aligned} P\left(\mu, \mathbf{B}|\Pi_{\mathcal{M}_\mathbf{y}}\right) &= P\left(\mu|\mathbf{B}, \Pi_{\mathcal{M}_\mathbf{y}}\right) P\left(\mathbf{B}|\Pi_{\mathcal{M}_\mathbf{y}}\right) && \text{(E.5)}\\ &= \lim_{k\to 0} \mathcal{N}\left(\mu|\mu_0, (k\mathbf{B})^{-1}\right) \mathcal{W}\left(\mathbf{B}|\mathbf{B}_0/k, k\right) && \text{(E.6)}\\ &= \alpha \left|\frac{\mathbf{B}}{2\pi}\right|^{1/2} |\mathbf{B}|^{-(d+1)/2} && \text{(E.7)}\\ P\left(\mathbf{W}|\Pi_\mathbf{W}\right) &= \lim_{k\to 0} \mathcal{W}\left(\mathbf{W}|\mathbf{W}_0/k, k\right) && \text{(E.8)}\\ &= \alpha\, |\mathbf{W}|^{-(d+1)/2} && \text{(E.9)} \end{aligned}$$

where $\mathcal{W}$ denotes a Wishart distribution and d the dimensionality of the i-vector. Since this densities do not integrate to 1, they are improper and the symbol α is introduced to denote a normalizing constant which approaches zero. Note that an improper prior does not implies an improper posterior.

E.3.2 Variational distributions

In order to derive the VB equations of the model we need to write down the joint distribution of all the observed and hidden variables:

$$P\left(\boldsymbol{\Phi},\mathcal{M},\mathbf{Y}|\theta,\Pi\right)=P\left(\boldsymbol{\Phi}|\mathbf{Y},\mathcal{M}_{\mathbf{y}},\mathbf{W},\theta,\Pi\right)P\left(\mathbf{Y}|\mathcal{M}_{\mathbf{y}},\mathbf{W},\theta,\Pi\right)P\left(\mathcal{M}_{\mathbf{y}}|\mathbf{W},\theta,\Pi\right)P\left(\mathbf{W}|\theta,\Pi\right)\ . \tag{E.10}$$

By looking the conditional dependencies described in the figure, we can simplify (E.10) into:

$$P\left(\boldsymbol{\Phi},\mathcal{M},\mathbf{Y}|\theta,\Pi\right)=P\left(\boldsymbol{\Phi}|\mathbf{Y},\mathbf{W},\theta\right)P\left(\mathbf{Y}|\mathcal{M}_{\mathbf{y}}\right)P\left(\mathcal{M}_{\mathbf{y}}|\Pi_{\mathcal{M}_{\mathbf{y}}}\right)P\left(\mathbf{W}|\Pi_{\mathbf{W}}\right)\ . \tag{E.11}$$

We found convenient to choose the following partition for the joint posterior of the latent variables of the model:

$$q\left(\mathcal{M},\mathbf{Y}\right)=q\left(\mathcal{M}\right)q\left(\mathbf{Y}\right)\ . \tag{E.12}$$

The optimum for each one of the factors $q\left(\mathbf{Z}_i\right)$ is obtained by taking the expectation of the joint distribution in (E.11) w.r.t. the rest of factors $q\left(\mathbf{Z}_{j\neq i}\right)$. We iterate across the factors until that the distributions converge.

Thus, the optimum for the factor $q\left(\mathbf{Y}\right)$ is given by

$$\ln q^{*}\left(\mathbf{Y}\right)=\mathrm{E}_{\mathcal{M}}\left[\ln P\left(\boldsymbol{\Phi},\mathcal{M},\mathbf{Y}|\theta,\Pi\right)\right]+\mathrm{const} \tag{E.13}$$

By plugging-in (E.11) we obtain:

$$\ln q^{*}\left(\mathbf{Y}\right)=\mathrm{E}_{\mathcal{M}}\left[\ln P\left(\boldsymbol{\Phi}|\mathbf{Y},\mathbf{W},\theta\right)\right]+\mathrm{E}_{\mathcal{M}}\left[\ln P\left(\mathbf{Y}|\mathcal{M}_{\mathbf{y}}\right)\right]+\mathrm{const} \tag{E.14}$$

where the terms that do not depend on $\mathbf{Y}$ can be absorbed into the additive constant const.

Now, we substitute (E.2) and (E.3) into (E.14) and, again, absorb any term that does not depend on $\mathbf{Y}$ into the additive constant. Thus, we obtain:

$$\ln q^{*}\left(\mathbf{Y}\right)=\sum_{i=1}^{M}\sum_{j=1}^{N_i}\mathrm{E}_{\mathcal{M}}\left[\ln P\left(\phi_{ij}|\mathbf{y}_i,\mathbf{W}\right)\right]+\sum_{i=1}^{M}\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\ln P\left(\mathbf{y}_i|\mathcal{M}_{\mathbf{y}}\right)\right]+\mathrm{const} \tag{E.15}$$

$$\begin{aligned}&=-\frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{N_i}\mathrm{E}_{\mathcal{M}}\left[(\phi_{ij}-\mathbf{y}_i)^T\mathbf{W}(\phi_{ij}-\mathbf{y}_i)\right]\\&\quad-\frac{1}{2}\sum_{i=1}^{M}\mathrm{E}_{\mathcal{M}}\left[(\mathbf{y}_i-\mu)^T\mathbf{B}(\mathbf{y}_i-\mu)\right]+\mathrm{const}\end{aligned} \tag{E.16}$$

$$\begin{aligned}&=\sum_{i=1}^{M}\mathbf{y}_i^T\mathrm{E}_{\mathcal{M}}\left[\mathbf{W}\right]\mathbf{F}_i-\frac{1}{2}\sum_{i=1}^{M}N_i\mathbf{y}_i^T\mathrm{E}_{\mathcal{M}}\left[\mathbf{W}\right]\mathbf{y}_i\\&\quad-\frac{1}{2}\sum_{i=1}^{M}\mathbf{y}_i^T\mathrm{E}_{\mathcal{M}}\left[\mathbf{B}\right]\mathbf{y}_i+\sum_{i=1}^{M}\mathbf{y}_i^T\mathrm{E}_{\mathcal{M}}\left[\mathbf{B}\mu\right]+\mathrm{const}\end{aligned} \tag{E.17}$$

$$=\sum_{i=1}^{M}-\frac{1}{2}\mathbf{y}_i^T\left(\mathrm{E}_{\mathcal{M}}\left[\mathbf{B}\right]+N_i\mathrm{E}_{\mathcal{M}}\left[\mathbf{W}\right]\right)\mathbf{y}_i+\mathbf{y}_i^T\left(\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\mathbf{B}\mu\right]+\mathrm{E}_{\mathcal{M}}\left[\mathbf{W}\right]\mathbf{F}_i\right)+\mathrm{const} \tag{E.18}$$

where $\mathbf{F}_i$ is first order sufficient statistic for speaker i:

$$\mathbf{F}_i = \sum_{j=1}^{N_i} \phi_{ij} \ . \tag{E.19}$$

We note that (E.18) has the form of the sum of logarithms of Gaussian distributions. Thus, we identify the mean and variance of the distributions and write:

$$\begin{aligned} q^*(\mathbf{Y}) &= \prod_{i=1}^{M} q^*(\mathbf{y}_i) && \text{(E.20)} \\ q^*(\mathbf{y}_i) &= \mathcal{N}\left(\mathbf{y}_i | \mathbf{L}_i^{-1}\gamma_i, \mathbf{L}_i^{-1}\right) && \text{(E.21)} \\ \mathbf{L}_i &= \mathrm{E}_{\mathcal{M}}[\mathbf{B}] + N_i \mathrm{E}_{\mathcal{M}}[\mathbf{W}] && \text{(E.22)} \\ \gamma_i &= \mathrm{E}_{\mathcal{M}_\mathbf{y}}[\mathbf{B}\mu] + \mathrm{E}_{\mathcal{M}}[\mathbf{W}]\, \mathbf{F}_i \ . && \text{(E.23)} \end{aligned}$$

Note that the factorization of $q^*(\mathbf{Y})$ into one Gaussian per speaker has not been forced in any way but it arises naturally from the factorization (E.12).

In the same manner, we derive the optimum for the factor $q(\mathcal{M})$:

$$\ln q^*(\mathcal{M}) = \mathrm{E}_{\mathbf{Y}}[\ln P(\mathbf{\Phi}, \mathcal{M}, \mathbf{Y} | \theta, \Pi)] + \text{const} \ . \tag{E.24}$$

By applying (E.11), we obtain:

$$\begin{aligned} \ln q^*(\mathcal{M}) = & \left[\mathrm{E}_{\mathbf{Y}}[\ln P(\mathbf{Y}|\mathcal{M}_\mathbf{y})] + \mathrm{E}_{\mathbf{Y}}\left[\ln P\left(\mathcal{M}_\mathbf{y}|\Pi_{\mathcal{M}_\mathbf{y}}\right)\right]\right] \\ & + \left[\mathrm{E}_{\mathbf{Y}}[\ln P(\mathbf{\Phi}|\mathbf{Y}, \mathbf{W}, \theta)] + \mathrm{E}_{\mathbf{Y}}[\ln P(\mathbf{W}|\Pi_\mathbf{W})]\right] + \text{const} && \text{(E.25)} \\ & = \ln q^*(\mathcal{M}_\mathbf{y}) + \ln q^*(\mathbf{W}) \ . && \text{(E.26)} \end{aligned}$$

The model $q^*(\mathcal{M})$ is decomposed into two independent factors, one for the speaker space and another for the channel space. Again, we did not force this factorization but it arose naturally.

For $q^*(\mathcal{M}_\mathbf{y})$, we substitute (E.2) and (E.7) into (E.25) and absorb any term that does not depend on μ or $\mathbf{B}$ into the additive constant:

$$\begin{aligned} \ln q^*(\mathcal{M}_\mathbf{y}) = & \sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}[\ln P(\mathbf{y}_i|\mathcal{M}_\mathbf{y})] + \mathrm{E}_{\mathbf{Y}}[\ln P(\mu, \mathbf{B}|\Pi)] + \text{const} && \text{(E.27)} \\ = & \frac{M}{2} \ln |\mathbf{B}| - \frac{1}{2} \sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[(\mathbf{y}_i - \mu)^T \mathbf{B} (\mathbf{y}_i - \mu)\right] + \frac{1}{2} \ln |\mathbf{B}| - \frac{d+1}{2} \ln |\mathbf{B}| + \text{const} \ . && \text{(E.28)} \end{aligned}$$

Now, we define:

$$\begin{aligned} \bar{\mathbf{y}} = & \frac{1}{M} \sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}[\mathbf{y}_i] && \text{(E.29)} \\ \mathbf{S}_\mathbf{y} = & \sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[(\mathbf{y}_i - \bar{\mathbf{y}})(\mathbf{y}_i - \bar{\mathbf{y}})^T\right] = \sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i \mathbf{y}_i^T\right] - M \bar{\mathbf{y}} \bar{\mathbf{y}}^T \ . && \text{(E.30)} \end{aligned}$$

We use them to develop (E.28):

$$\begin{aligned}\ln q^*\left(\mathcal{M}_{\mathbf{y}}\right) =& \frac{M}{2}\ln|\mathbf{B}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{B}\sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}\left[\left(\mathbf{y}_i-\mu\right)\left(\mathbf{y}_i-\mu\right)^T\right]\right)\\ &+\frac{1}{2}\ln|\mathbf{B}| - \frac{d+1}{2}\ln|\mathbf{B}| + \mathrm{const} &\text{(E.31)}\\ =& \frac{M}{2}\ln|\mathbf{B}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{B}\sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}\left[\left(\left(\mathbf{y}_i-\overline{\mathbf{y}}\right)-\left(\mu-\overline{\mathbf{y}}\right)\right)\left(\left(\mathbf{y}_i-\overline{\mathbf{y}}\right)-\left(\mu-\overline{\mathbf{y}}\right)\right)^T\right]\right)\\ &+\frac{1}{2}\ln|\mathbf{B}| - \frac{d+1}{2}\ln|\mathbf{B}| + \mathrm{const} &\text{(E.32)}\\ =& \frac{M}{2}\ln|\mathbf{B}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{B}\left(\sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}\left[\left(\mathbf{y}_i-\overline{\mathbf{y}}\right)\left(\mathbf{y}_i-\overline{\mathbf{y}}\right)^T\right]+M\left(\mu-\overline{\mathbf{y}}\right)\left(\mu-\overline{\mathbf{y}}\right)^T\right)\right)\\ &+\frac{1}{2}\ln|\mathbf{B}| - \frac{d+1}{2}\ln|\mathbf{B}| + \mathrm{const} &\text{(E.33)}\\ =& \left[\frac{1}{2}\ln|\mathbf{B}| - \frac{M}{2}(\mu-\overline{\mathbf{y}})^T\mathbf{B}(\mu-\overline{\mathbf{y}})\right] + \left[\frac{M}{2}\ln|\mathbf{B}| - \frac{d+1}{2}\ln|\mathbf{B}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{B}\mathbf{S}_{\mathbf{y}}\right)\right]\\ &+\mathrm{const}\;. &\text{(E.34)}\end{aligned}$$

If we compare equations (E.34) and (A.20), we see that $q^*\left(\mathcal{M}_{\mathbf{y}}\right)$ is Gaussian-Wishart distributed:

$$q^*\left(\mathcal{M}_{\mathbf{y}}\right) = \mathcal{N}\left(\mu|\overline{\mu},(M\mathbf{B})^{-1}\right)\mathcal{W}\left(\mathbf{B}|\mathbf{\Psi}_{\mathbf{y}},M\right) \quad \text{if } M>d \tag{E.35}$$

where

$$\begin{aligned}\overline{\mu} =& \overline{\mathbf{y}} &\text{(E.36)}\\ \mathbf{\Psi}_{\mathbf{y}} =& \mathbf{S}_{\mathbf{y}}^{-1}\;. &\text{(E.37)}\end{aligned}$$

For $q^*(\mathbf{W})$, we substitute (E.3), and (E.9) into (E.25) and absorb any term that does not depend on $\mathbf{W}$ into the additive constant:

$$\begin{aligned}\ln q^*(\mathbf{W}) =& \sum_{i=1}^{M}\sum_{j=1}^{N_i}\mathrm{E}_{\mathbf{Y}}\left[\ln P\left(\phi_{ij}|\mathbf{y}_i,\mathbf{W}\right)\right] + \mathrm{E}_{\mathbf{Y}}\left[\ln P\left(\mathbf{W}|\Pi_{\mathbf{W}}\right)\right] + \mathrm{const} &\text{(E.38)}\\ =& \frac{N}{2}\ln|\mathbf{W}| - \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{N_i}\mathrm{E}_{\mathbf{Y}}\left[(\phi_{ij}-\mathbf{y}_i)^T\mathbf{W}(\phi_{ij}-\mathbf{y}_i)\right] - \frac{d+1}{2}\ln|\mathbf{W}| + \mathrm{const}\\ && \text{(E.39)}\\ =& \frac{N}{2}\ln|\mathbf{W}| - \frac{d+1}{2}\ln|\mathbf{W}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{W}\sum_{i=1}^{M}\sum_{j=1}^{N_i}\mathrm{E}_{\mathbf{Y}}\left[\left(\phi_{ij}-\mathbf{y}_i\right)\left(\phi_{ij}-\mathbf{y}_i\right)^T\right]\right) + \mathrm{const}\\ && \text{(E.40)}\\ =& \ln|\mathbf{W}| - \frac{d+1}{2}\ln|\mathbf{W}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{W}\mathbf{S}_{\mathbf{W}}\right) + \mathrm{const} &\text{(E.41)}\end{aligned}$$

where

$$\mathbf{S} = \sum_{i=1}^{M}\sum_{j=1}^{N_i} \phi_{ij}\phi_{ij}^T \tag{E.42}$$

$$\mathbf{S_W} = \sum_{i=1}^{M}\sum_{j=1}^{N_i} \mathrm{E}_{\mathbf{Y}}\left[\left(\phi_{ij} - \mathbf{y}_i\right)\left(\phi_{ij} - \mathbf{y}_i\right)^T\right] \tag{E.43}$$

$$= \mathbf{S} + \sum_{i=1}^{M}\left(N_i \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right] - \mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]\mathbf{F}_i^T - \mathbf{F}_i\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]^T\right) . \tag{E.44}$$

If we compare (E.41) with (A.20), we see that $q^*(\mathbf{W})$ is Wishart distributed:

$$q^*(\mathbf{W}) = \mathcal{W}\left(\mathbf{W}|\mathbf{\Psi_W}, N\right) \quad \text{if } N > d \tag{E.45}$$

where

$$\mathbf{\Psi_W} = \mathbf{S}_\mathbf{W}^{-1} . \tag{E.46}$$

Finally, we need to evaluate the expectations $\mathrm{E}_{\mathcal{M}}[\mathbf{B}]$, $\mathrm{E}_{\mathcal{M}}[\mathbf{B}\mu]$, $\mathrm{E}_{\mathcal{M}}[\mathbf{W}]$, $\mathrm{E}_{\mathbf{Y}}[\mathbf{y}_i]$ and $\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right]$. Using the properties of the Gaussian and Wishart distributions [Bishop, 2006] we obtain:

$$\mathrm{E}_{\mathcal{M}}[\mathbf{B}] = M\mathbf{\Psi_y} \tag{E.47}$$

$$\mathrm{E}_{\mathcal{M}}[\mathbf{B}\mu] = \int_{\mathbf{B}}\int_{\mu} \mathbf{B}\mu\mathcal{N}\left(\mu|\overline{\mu},(M\mathbf{B})^{-1}\right)\mathcal{W}\left(\mathbf{B}|\mathbf{\Psi_y}, M\right)\,\mathrm{d}\mu\,\mathrm{d}\mathbf{B} \tag{E.48}$$

$$= \int_{\mathbf{B}} \mathbf{B}\int_{\mu} \mu\mathcal{N}\left(\mu|\overline{\mu},(M\mathbf{B})^{-1}\right)\,\mathrm{d}\mu\,\mathcal{W}\left(\mathbf{B}|\mathbf{\Psi_y}, M\right)\,\mathrm{d}\mathbf{B} \tag{E.49}$$

$$= \int_{\mathbf{B}} \mathbf{B}\mathcal{W}\left(\mathbf{B}|\mathbf{\Psi_y}, M\right)\,\mathrm{d}\mathbf{B}\,\overline{\mu} = M\mathbf{\Psi_y}\overline{\mu} \tag{E.50}$$

$$\mathrm{E}_{\mathcal{M}}[\mathbf{W}] = N\mathbf{\Psi_W} \tag{E.51}$$

$$\mathrm{E}_{\mathbf{Y}}[\mathbf{y}_i] = \mathbf{L}_i^{-1}\gamma_i \tag{E.52}$$

$$\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right] = \mathbf{L}_i^{-1} + \mathrm{E}_{\mathbf{Y}}[\mathbf{y}_i]\,\mathrm{E}_{\mathbf{Y}}[\mathbf{y}_i]^T = \mathbf{L}_i^{-1} + \mathbf{L}_i^{-1}\gamma_i\gamma_i^T\mathbf{L}_i^{-1} . \tag{E.53}$$

E.3.3 Variational lower bound

In this section, we derive the equation to compute the variational lower bound. The lower bound is an approximation of the complete data likelihood and can be use to evaluate the convergence of the VB algorithm.

$$\begin{aligned}
\mathcal{L} = &\int_{\mathbf{Y}}\int_{\mathbf{W}}\int_{\mathbf{B}}\int_{\mu} q\left(\mu, \mathbf{B}, \mathbf{W}, \mathbf{Y}\right) \ln\left(\frac{P\left(\mathbf{\Phi}, \mu, \mathbf{B}, \mathbf{W}, \mathbf{Y}|\theta, \Pi\right)}{q\left(\mu, \mathbf{B}, \mathbf{W}, \mathbf{Y}\right)}\right) \mathrm{d}\mu\, \mathrm{d}\mathbf{B}\, \mathrm{d}\mathbf{W}\, \mathrm{d}\mathbf{Y} && \text{(E.54)}\\
=&\mathrm{E}_{\mathcal{M},\mathbf{Y}}\left[\ln P\left(\mathbf{\Phi}, \mu, \mathbf{B}, \mathbf{W}, \mathbf{Y}|\theta, \Pi\right)\right] - \mathrm{E}_{\mathcal{M},\mathbf{Y}}\left[\ln q\left(\mu, \mathbf{B}, \mathbf{W}, \mathbf{Y}\right)\right] && \text{(E.55)}\\
=&\mathrm{E}_{\mathbf{W},\mathbf{Y}}\left[\ln P\left(\mathbf{\Phi}|\mathbf{Y}, \mathbf{W}, \theta\right)\right] + \mathrm{E}_{\mathcal{M}_{\mathbf{y}},\mathbf{Y}}\left[\ln P\left(\mathbf{Y}|\mathcal{M}_{\mathbf{y}}\right)\right]\\
&+ \mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\ln P\left(\mathcal{M}_{\mathbf{y}}|\Pi_{\mathcal{M}_{\mathbf{y}}}\right)\right] + \mathrm{E}_{\mathbf{W}}\left[\ln P\left(\mathbf{W}|\Pi_{\mathbf{W}}\right)\right]\\
&- \mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\ln q\left(\mu, \mathbf{B}\right)\right] - \mathrm{E}_{\mathbf{W}}\left[\ln q\left(\mathbf{W}\right)\right] - \mathrm{E}_{\mathbf{Y}}\left[\ln q\left(\mathbf{Y}\right)\right]\ . && \text{(E.56)}
\end{aligned}$$

We evaluate $\mathrm{E}_{\mathbf{W},\mathbf{Y}}\left[\ln P\left(\mathbf{\Phi}|\mathbf{Y}, \mathbf{W}, \theta\right)\right]$:

$$\begin{aligned}
\mathrm{E}_{\mathbf{W},\mathbf{Y}}\left[\ln P\left(\mathbf{\Phi}|\mathbf{Y}, \mathbf{W}, \theta\right)\right] =&\sum_{i=1}^{M}\sum_{j=1}^{N_i}\mathrm{E}_{\mathbf{W},\mathbf{Y}}\left[\ln P\left(\phi_{ij}|\mathbf{y}_i, \mathbf{W}\right)\right] && \text{(E.57)}\\
=&\frac{N}{2}\mathrm{E}_{\mathbf{W}}\left[\ln|\mathbf{W}|\right] - \frac{Nd}{2}\ln(2\pi)\\
&- \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{N_i}\mathrm{E}_{\mathbf{W},\mathbf{Y}}\left[(\phi_{ij} - \mathbf{y}_i)^T\mathbf{W}(\phi_{ij} - \mathbf{y}_i)\right] && \text{(E.58)}\\
=&\frac{N}{2}\ln\tilde{\mathbf{W}} - \frac{Nd}{2}\ln(2\pi)\\
&- \frac{1}{2}\mathrm{tr}\left(\mathrm{E}_{\mathbf{W}}\left[\mathbf{W}\right]\sum_{i=1}^{M}\sum_{j=1}^{N_i}\mathrm{E}_{\mathbf{Y}}\left[\left(\phi_{ij} - \mathbf{y}_i\right)\left(\phi_{ij} - \mathbf{y}_i\right)^T\right]\right) && \text{(E.59)}\\
=&\frac{N}{2}\ln\tilde{\mathbf{W}} - \frac{Nd}{2}\ln(2\pi) - \frac{1}{2}\mathrm{tr}\left(\mathrm{E}_{\mathbf{W}}\left[\mathbf{W}\right]\mathbf{S}_{\mathbf{W}}\right) && \text{(E.60)}
\end{aligned}$$

where we used the properties of the Wishart distribution [Bishop, 2006] to define:

$$\ln\tilde{\mathbf{W}} \equiv \mathrm{E}_{\mathbf{W}}\left[\ln|\mathbf{W}|\right] = \sum_{i=1}^{d}\psi\left(\frac{N+1-i}{2}\right) + d\ln 2 + \ln|\mathbf{\Psi}_{\mathbf{W}}|\ . \tag{E.61}$$

If we compute the lower bound just after updating $q^*\left(\mathbf{W}\right)$, $\mathrm{E}_{\mathbf{W}}\left[\mathbf{W}\right] = N\mathbf{S}_{\mathbf{W}}^{-1}$ and we can simplify:

$$\mathrm{E}_{\mathbf{W},\mathbf{Y}}\left[\ln P\left(\mathbf{\Phi}|\mathbf{Y}, \mathbf{W}, \theta\right)\right] =\frac{N}{2}\ln\tilde{\mathbf{W}} - \frac{Nd}{2}\left(\ln(2\pi) + 1\right)\ . \tag{E.62}$$

Note that, if we evaluate the lower bound between the update of $q\left(\mathbf{Y}\right)$ and $q\left(\mathbf{W}\right)$, we should use (E.60).

Now, we evaluate $\mathrm{E}_{\mathcal{M}_{\mathbf{y}},\mathbf{Y}}\left[\ln P\left(\mathbf{Y}|\mathcal{M}_{\mathbf{y}}\right)\right]$:

$$\begin{aligned}
\mathrm{E}_{\mathcal{M}_{\mathbf{y}},\mathbf{Y}}\left[\ln P\left(\mathbf{Y}|\mathcal{M}_{\mathbf{y}}\right)\right] =& \sum_{i=1}^{M}\mathrm{E}_{\mathcal{M}_{\mathbf{y}},\mathbf{Y}}\left[\ln P\left(\mathbf{y}_i|\mu,\mathbf{B}\right)\right] && \text{(E.63)}\\
=&\frac{M}{2}\mathrm{E}_{\mathbf{B}}\left[\ln|\mathbf{B}|\right]-\frac{Md}{2}\ln(2\pi)-\frac{1}{2}\sum_{i=1}^{M}\mathrm{E}_{\mathcal{M}_{\mathbf{y}},\mathbf{Y}}\left[(\mathbf{y}_i-\mu)^T\mathbf{B}(\mathbf{y}_i-\mu)\right] \\
& && \text{(E.64)}\\
=&\frac{M}{2}\mathrm{E}_{\mathbf{B}}\left[\ln|\mathbf{B}|\right]-\frac{Md}{2}\ln(2\pi)\\
&-\frac{1}{2}\sum_{i=1}^{M}\left(\mathrm{E}_{\mathcal{M}_{\mathbf{y}},\mathbf{Y}}\left[\mathbf{y}_i^T\mathbf{B}\mathbf{y}_i\right]-2\mathrm{E}_{\mathcal{M}_{\mathbf{y}},\mathbf{Y}}\left[\mathbf{y}_i^T\mathbf{B}\mu\right]+\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\mu^T\mathbf{B}\mu\right]\right)\\
& && \text{(E.65)}\\
=&\frac{M}{2}\mathrm{E}_{\mathbf{B}}\left[\ln|\mathbf{B}|\right]-\frac{Md}{2}\ln(2\pi)+M\overline{\mathbf{y}}^T\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\mathbf{B}\mu\right]\\
&-\frac{1}{2}\mathrm{tr}\left(\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\mathbf{B}\right]\sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right]+M\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\mathbf{B}\mu\mu^T\right]\right)\ . && \text{(E.66)}
\end{aligned}$$

To evaluate (E.66) we need to calculate some new expectations. Using the properties of the Gaussian and Wishart distributions [Bishop, 2006] we have

$$\ln\tilde{\mathbf{B}}\equiv\mathrm{E}_{\mathbf{B}}\left[\ln|\mathbf{B}|\right]=\sum_{i=1}^{d}\psi\left(\frac{M+1-i}{2}\right)+d\ln 2+\ln|\mathbf{\Psi}_{\mathbf{y}}| \tag{E.67}$$

and $\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\mathbf{B}\mu\mu^T\right]$ is

$$\begin{aligned}
\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\mathbf{B}\mu\mu^T\right]=&\int_{\mathbf{B}}\int_{\mu}\mathbf{B}\mu\mu^T\mathcal{N}\left(\mu|\overline{\mu},(M\mathbf{B})^{-1}\right)\mathcal{W}\left(\mathbf{B}|\mathbf{\Psi}_{\mathbf{y}},M\right)\,\mathrm{d}\mu\,\mathrm{d}\mathbf{B} && \text{(E.68)}\\
=&\int_{\mathbf{B}}\mathbf{B}\int_{\mu}\mu\mu^T\mathcal{N}\left(\mu|\overline{\mu},(M\mathbf{B})^{-1}\right)\,\mathrm{d}\mu\,\mathcal{W}\left(\mathbf{B}|\mathbf{\Psi}_{\mathbf{y}},M\right)\,\mathrm{d}\mathbf{B} && \text{(E.69)}\\
=&\int_{\mathbf{B}}\mathbf{B}\left((M\mathbf{B})^{-1}+\overline{\mu}\overline{\mu}^T\right)\mathcal{W}\left(\mathbf{B}|\mathbf{\Psi}_{\mathbf{y}},M\right)\,\mathrm{d}\mathbf{B} && \text{(E.70)}\\
=&M^{-1}\mathbf{I}+\int_{\mathbf{B}}\mathbf{B}\mathcal{W}\left(\mathbf{B}|\mathbf{\Psi}_{\mathbf{y}},M\right)\,\mathrm{d}\mathbf{B}\,\overline{\mu}\overline{\mu}^T && \text{(E.71)}\\
=&M^{-1}\mathbf{I}+M\mathbf{\Psi}_{\mathbf{y}}\overline{\mu}\overline{\mu}^T\ . && \text{(E.72)}
\end{aligned}$$

For the particular case where we evaluate the lower bound after updating $q^*\left(\mu,\mathbf{B}\right)$ we can go on simplifying. We plug-in (E.30), (E.47), (E.50),(E.67) and (E.72) into (E.66) and use that $\overline{\mu} = \overline{\mathbf{y}}$ and $\mathbf{\Psi}_\mathbf{y} = \mathbf{S}_\mathbf{y}^{-1}$:

$$\begin{aligned}
\mathrm{E}_{\mathcal{M}_\mathbf{y},\mathbf{Y}}\left[\ln P\left(\mathbf{Y}|\mathcal{M}_\mathbf{y}\right)\right] =& \frac{M}{2}\ln\tilde{\mathbf{B}} - \frac{Md}{2}\ln(2\pi) + M\overline{\mathbf{y}}^T M\mathbf{S}_\mathbf{y}^{-1}\overline{\mathbf{y}} \\
& -\frac{1}{2}\mathrm{tr}\left(M\mathbf{S}_\mathbf{y}^{-1}\sum_{i=1}^{M}\mathrm{E}_\mathbf{Y}\left[\mathbf{y}_i\mathbf{y}_i^T\right] + M\left(M^{-1}\mathbf{I} + M\mathbf{S}_\mathbf{y}^{-1}\overline{\mathbf{y}}\overline{\mathbf{y}}^T\right)\right) && \text{(E.73)} \\
=& \frac{M}{2}\ln\tilde{\mathbf{B}} - \frac{Md}{2}\ln(2\pi) \\
& -\frac{1}{2}\mathrm{tr}\left(\mathbf{I} + M\mathbf{S}_\mathbf{y}^{-1}\left(\sum_{i=1}^{M}\mathrm{E}_\mathbf{Y}\left[\mathbf{y}_i\mathbf{y}_i^T\right] - M\overline{\mathbf{y}}\overline{\mathbf{y}}^T\right)\right) && \text{(E.74)} \\
=& \frac{M}{2}\ln\tilde{\mathbf{B}} - \frac{Md}{2}\ln(2\pi) - \frac{1}{2}\mathrm{tr}\left(\mathbf{I} + M\mathbf{S}_\mathbf{y}^{-1}\mathbf{S}_\mathbf{y}\right) && \text{(E.75)} \\
=& \frac{M}{2}\ln\tilde{\mathbf{B}} - \frac{Md}{2}\ln(2\pi) - \frac{1}{2}\mathrm{tr}\left((M+1)\mathbf{I}\right) && \text{(E.76)} \\
=& \frac{M}{2}\ln\tilde{\mathbf{B}} - \frac{Md}{2}\ln(2\pi) - \frac{(M+1)d}{2}\ . && \text{(E.77)}
\end{aligned}$$

Note that if we evaluate the lower bound after updating $q^*(\mathbf{Y})$ and before $q^*(\mu, \mathbf{B})$, $\overline{\mu} \neq \overline{\mathbf{y}}$ and $\mathbf{\Psi}_\mathbf{y} \neq \mathbf{S}_\mathbf{y}^{-1}$ and we should use (E.66).

Now, we evaluate $\mathrm{E}_{\mathcal{M}_\mathbf{y}}\left[\ln P\left(\mathcal{M}_\mathbf{y}|\Pi_{\mathcal{M}_\mathbf{y}}\right)\right]$:

$$\mathrm{E}_{\mathcal{M}_\mathbf{y}}\left[\ln P\left(\mathcal{M}_\mathbf{y}|\Pi_{\mathcal{M}_\mathbf{y}}\right)\right] = \ln\alpha - \frac{d}{2}\ln(2\pi) - \frac{d}{2}\ln\tilde{\mathbf{B}}\ . \qquad \text{(E.78)}$$

Now, we evaluate $\mathrm{E}_\mathbf{W}\left[\ln P\left(\mathbf{W}|\Pi_\mathbf{W}\right)\right]$:

$$\mathrm{E}_{\mathcal{M}_\mathbf{y}}\left[\ln P\left(\mathbf{W}|\Pi_\mathbf{W}\right)\right] = \ln\alpha - \frac{d+1}{2}\ln\tilde{\mathbf{W}}\ . \qquad \text{(E.79)}$$

Now, we evaluate $\mathrm{E}_{\mathcal{M}_\mathbf{y}}\left[\ln q\left(\mu, \mathbf{B}\right)\right]$:

$$\begin{aligned}
\mathrm{E}_{\mathcal{M}_\mathbf{y}}\left[\ln q\left(\mu,\mathbf{B}\right)\right] =& \mathrm{E}_{\mathcal{M}_\mathbf{y}}\left[\ln\mathcal{N}\left(\mu|\overline{\mu},(M\mathbf{B})^{-1}\right)\right] + \mathrm{E}_\mathbf{B}\left[\ln\mathcal{W}\left(\mathbf{B}|\mathbf{\Psi}_\mathbf{y},M\right)\right] && \text{(E.80)} \\
=& \frac{d}{2}\ln\left(\frac{M}{2\pi}\right) + \frac{1}{2}\ln\tilde{\mathbf{B}} - \frac{M}{2}\mathrm{E}_{\mathcal{M}_\mathbf{y}}\left[(\mu-\overline{\mu})^T\mathbf{B}(\mu-\overline{\mu})\right] - \mathrm{H}\left[q\left(\mathbf{B}\right)\right] && \text{(E.81)}
\end{aligned}$$

where $\mathrm{H}\left[q\left(\mathbf{B}\right)\right]$ is the entropy of the Wishart distribution [Bishop, 2006]

$$\begin{aligned}
\mathrm{H}\left[q\left(\mathbf{B}\right)\right] =& \mathrm{H}\left[\mathcal{W}\left(\mathbf{B}|\mathbf{\Psi}_\mathbf{y},M\right)\right] && \text{(E.82)} \\
=& -\ln B\left(\mathbf{\Psi}_\mathbf{y},M\right) - \frac{M-d-1}{2}\ln\tilde{\mathbf{B}} + \frac{Md}{2} && \text{(E.83)} \\
B(\mathbf{W},N) =& \frac{1}{2^{Nd/2}Z_{Nd}}\left|\mathbf{W}\right|^{-N/2} && \text{(E.84)}
\end{aligned}$$

and

$$\begin{aligned}
\mathrm{E}_{\mathcal{M}_\mathbf{y}}\left[(\mu-\overline{\mu})^T\mathbf{B}(\mu-\overline{\mu})\right] =& \mathrm{E}_{\mathcal{M}_\mathbf{y}}\left[\mu^T\mathbf{B}\mu\right] - 2\overline{\mu}^T\mathrm{E}_{\mathcal{M}_\mathbf{y}}\left[\mathbf{B}\mu\right] + \overline{\mu}^T\mathrm{E}_\mathbf{B}\left[\mathbf{B}\right]\overline{\mu} && \text{(E.85)} \\
=& \mathrm{tr}\left(\mathrm{E}_{\mathcal{M}_\mathbf{y}}\left[\mathbf{B}\mu\mu^T\right]\right) - \overline{\mu}^T\mathrm{E}_\mathbf{B}\left[\mathbf{B}\right]\overline{\mu} && \text{(E.86)} \\
=& \mathrm{tr}\left(M^{-1}\mathbf{I} + M\mathbf{S}_\mathbf{y}^{-1}\overline{\mu}\overline{\mu}^T\right) - \overline{\mu}^T M\mathbf{S}_\mathbf{y}^{-1}\overline{\mu} && \text{(E.87)} \\
=& dM^{-1}\ . && \text{(E.88)}
\end{aligned}$$

By plugging (E.88) into (E.81):

$$\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\ln q\left(\mu, \mathbf{B}\right)\right] = \frac{d}{2}\ln\left(\frac{M}{2\pi}\right) + \frac{1}{2}\ln\tilde{\mathbf{B}} - \frac{d}{2} - \mathrm{H}\left[q\left(\mathbf{B}\right)\right] \ . \tag{E.89}$$

Now, we evaluate $\mathrm{E}_{\mathbf{W}}\left[\ln q\left(\mathbf{W}\right)\right]$:

$$\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\ln q\left(\mathbf{W}\right)\right] = \mathrm{E}_{\mathbf{W}}\left[\ln \mathcal{W}\left(\mathbf{W}|\mathbf{\Psi}_{\mathbf{W}}, N\right)\right] = -\mathrm{H}\left[q\left(\mathbf{W}\right)\right] \tag{E.90}$$

where $\mathrm{H}\left[q\left(\mathbf{W}\right)\right]$ is the entropy of the Wishart distribution [Bishop, 2006]

$$\begin{aligned}\mathrm{H}\left[q\left(\mathbf{W}\right)\right] =& \mathrm{H}\left[\mathcal{W}\left(\mathbf{W}|\mathbf{\Psi}_{\mathbf{W}}, N\right)\right] & \text{(E.91)}\\ =& -\ln B\left(\mathbf{\Psi}_{\mathbf{W}}, N\right) - \frac{N-d-1}{2}\ln\tilde{\mathbf{W}} + \frac{Nd}{2} \ . & \text{(E.92)}\end{aligned}$$

Finally, we evaluate $\mathrm{E}_{\mathbf{Y}}\left[\ln q\left(\mathbf{Y}\right)\right]$:

$$\begin{aligned}\mathrm{E}_{\mathbf{Y}}\left[\ln q\left(\mathbf{Y}\right)\right] =& \sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}\left[\ln\mathcal{N}\left(\mathbf{y}_i|\mathbf{L}_i^{-1}\gamma_i, \mathbf{L}_i^{-1}\right)\right] & \text{(E.93)}\\ =& -\frac{Md}{2}\ln(2\pi) + \frac{1}{2}\sum_{i=1}^{M}\ln|\mathbf{L}_i| - \frac{1}{2}\sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}\left[(\mathbf{y}_i - \mathbf{L}_i^{-1}\gamma_i)^T\mathbf{L}_i(\mathbf{y}_i - \mathbf{L}_i^{-1}\gamma_i)\right] & \text{(E.94)}\\ =& -\frac{Md}{2}\ln(2\pi) + \frac{1}{2}\sum_{i=1}^{M}\ln|\mathbf{L}_i| \\ & -\frac{1}{2}\sum_{i=1}^{M}\mathrm{tr}\left(\mathbf{L}_i\mathrm{E}_{\mathbf{Y}}\left[\left(\mathbf{y}_i - \mathbf{L}_i^{-1}\gamma_i\right)\left(\mathbf{y}_i - \mathbf{L}_i^{-1}\gamma_i\right)^T\right]\right) & \text{(E.95)}\\ =& -\frac{Md}{2}\ln(2\pi) + \frac{1}{2}\sum_{i=1}^{M}\ln|\mathbf{L}_i| - \frac{1}{2}\sum_{i=1}^{M}\mathrm{tr}\left(\mathbf{L}_i\mathbf{L}_i^{-1}\right) & \text{(E.96)}\\ =& -\frac{Md}{2}(\ln(2\pi)+1) + \frac{1}{2}\sum_{i=1}^{M}\ln|\mathbf{L}_i| \ . & \text{(E.97)}\end{aligned}$$

E.4 Variational Inference with Informative Conjugate Priors

E.4.1 Model priors

In this section, we put an informative prior over the model parameters. We chose Gaussian Wishart priors that are conjugate priors for the Gaussian distribution:

$$\begin{aligned}P\left(\mathcal{M}_{\mathbf{y}}|\Pi_{\mathcal{M}_{\mathbf{y}}}\right) =& \mathcal{N}\left(\mu|\overline{\mu}_0, \left(\beta_{\mathbf{y}_0}\mathbf{B}\right)^{-1}\right)\mathcal{W}\left(\mathbf{B}|\mathbf{\Psi}_{\mathbf{y}_0}, \nu_{\mathbf{y}_0}\right) & \text{(E.98)}\\ P\left(\mathbf{W}|\Pi_{\mathbf{W}}\right) =& \mathcal{W}\left(\mathbf{W}|\mathbf{\Psi}_{\mathbf{W}_0}, \nu_{\mathbf{W}_0}\right) \ . & \text{(E.99)}\end{aligned}$$

E.4.2 Variational distributions

We write again the joint distribution of all the random variables:

$$P(\mathbf{\Phi}, \mathcal{M}, \mathbf{Y}|\theta, \Pi) = P(\mathbf{\Phi}|\mathbf{Y}, \mathbf{W}, \theta)\, P(\mathbf{Y}|\mathcal{M}_{\mathbf{y}})\, P\left(\mathcal{M}_{\mathbf{y}}|\Pi_{\mathcal{M}_{\mathbf{y}}}\right) P(\mathbf{W}|\Pi_{\mathbf{W}}) \ . \tag{E.100}$$

We assume a partition of the posterior into two factors:

$$q(\mathcal{M}, \mathbf{Y}) = q(\mathcal{M})\, q(\mathbf{Y}) \ . \tag{E.101}$$

The optimum for the factor $q(\mathbf{Y})$ is the same as for the non-informative case:

$$\begin{aligned} q^*(\mathbf{Y}) =& \prod_{i=1}^{M} q^*(\mathbf{y}_i) && \text{(E.102)}\\ q^*(\mathbf{y}_i) =& \mathcal{N}\left(\mathbf{y}_i|\mathbf{L}_i^{-1}\gamma_i, \mathbf{L}_i^{-1}\right) && \text{(E.103)}\\ \mathbf{L}_i =& \mathrm{E}_{\mathcal{M}_{\mathbf{y}}}[\mathbf{B}] + N_i \mathrm{E}_{\mathbf{W}}[\mathbf{W}] && \text{(E.104)}\\ \gamma_i =& \mathrm{E}_{\mathcal{M}_{\mathbf{y}}}[\mathbf{B}\mu] + \mathrm{E}_{\mathbf{W}}[\mathbf{W}]\, \mathbf{F}_i \ . && \text{(E.105)} \end{aligned}$$

The optimum for the factor $q(\mathcal{M})$ is given by

$$\ln q^*(\mathcal{M}) = \mathrm{E}_{\mathbf{Y}}[\ln P(\mathbf{\Phi}, \mathcal{M}, \mathbf{Y}|\theta, \Pi)] + \text{const} \ . \tag{E.106}$$

By plugging (E.100), we obtain again that the model factor can be decomposed into two independent factors:

$$\begin{aligned} \ln q^*(\mathcal{M}) = & \left[\mathrm{E}_{\mathbf{Y}}[\ln P(\mathbf{Y}|\mathcal{M}_{\mathbf{y}})] + \mathrm{E}_{\mathbf{Y}}\left[\ln P\left(\mathcal{M}_{\mathbf{y}}|\Pi_{\mathcal{M}_{\mathbf{y}}}\right)\right]\right] \\ & + [\mathrm{E}_{\mathbf{Y}}[\ln P(\mathbf{\Phi}|\mathbf{Y}, \mathbf{W}, \theta)] + \mathrm{E}_{\mathbf{Y}}[\ln P(\mathbf{W}|\Pi_{\mathbf{W}})]] + \text{const} && \text{(E.107)}\\ = & \ln q^*(\mathcal{M}_{\mathbf{y}}) + \ln q^*(\mathbf{W}) \ . && \text{(E.108)} \end{aligned}$$

First, we compute $q^*(\mu, \mathbf{B})$. We substitute (E.2), and (E.98) into (E.107) and absorb any term that does not depend on μ or $\mathbf{B}$ into the additive constant:

$$\ln q^*(\mathcal{M}_{\mathbf{y}}) = \sum_{i=1}^{M} \mathrm{E}_{\mathbf{Y}}[\ln P(\mathbf{y}_i|\mathcal{M}_{\mathbf{y}})] + \mathrm{E}_{\mathbf{Y}}\left[\ln P\left(\mu, \mathbf{B}|\Pi_{\mathcal{M}_{\mathbf{y}}}\right)\right] + \text{const} \tag{E.109}$$

$$\begin{aligned} =& \frac{M}{2}\ln|\mathbf{B}| - \frac{1}{2}\sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}\left[(\mathbf{y}_i - \mu)^T\mathbf{B}(\mathbf{y}_i - \mu)\right] \\ & + \frac{1}{2}\ln|\mathbf{B}| - \frac{\beta_{\mathbf{y}_0}}{2}(\mu - \overline{\mu}_0)^T\mathbf{B}(\mu - \overline{\mu}_0) \\ & + \frac{\nu_{\mathbf{y}_0} - d - 1}{2}\ln|\mathbf{B}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{B}\mathbf{\Psi}_{\mathbf{y}_0}^{-1}\right) + \text{const} \ . && \text{(E.110)} \end{aligned}$$

Now, we define:

$$\bar{\mathbf{y}} =\frac{1}{M}\sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right] \tag{E.111}$$

$$\mathbf{S}_{\mathbf{y}} =\sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}\left[\left(\mathbf{y}_i-\bar{\mathbf{y}}\right)\left(\mathbf{y}_i-\bar{\mathbf{y}}\right)^T\right]=\sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right]-M\bar{\mathbf{y}}\bar{\mathbf{y}}^T \tag{E.112}$$

$$\beta_{\mathbf{y}} =\beta_{\mathbf{y}_0}+M \tag{E.113}$$

$$\nu_{\mathbf{y}} =\nu_{\mathbf{y}_0}+M \tag{E.114}$$

$$\bar{\mu} =\frac{1}{\beta_{\mathbf{y}}}\left(\beta_{\mathbf{y}_0}\bar{\mu}_0+M\bar{\mathbf{y}}\right) \tag{E.115}$$

$$\mathbf{\Psi}_{\mathbf{y}}^{-1} =\mathbf{\Psi}_{\mathbf{y}_0}^{-1}+\mathbf{S}_{\mathbf{y}}+\frac{\beta_{\mathbf{y}_0}M}{\beta_{\mathbf{y}}}\left(\bar{\mathbf{y}}-\bar{\mu}_0\right)\left(\bar{\mathbf{y}}-\bar{\mu}_0\right)^T\ . \tag{E.116}$$

Now, we can write (E.110) as

$$\begin{aligned}
\ln q^*\left(\mathcal{M}_{\mathbf{y}}\right) =&\frac{M}{2}\ln|\mathbf{B}|-\frac{1}{2}\mathrm{tr}\left(\mathbf{B}\left(\sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}\left[\left(\mathbf{y}_i-\bar{\mathbf{y}}\right)\left(\mathbf{y}_i-\bar{\mathbf{y}}\right)^T\right]+M\left(\mu-\bar{\mathbf{y}}\right)\left(\mu-\bar{\mathbf{y}}\right)^T\right)\right)\\
&+\frac{1}{2}\ln|\mathbf{B}|-\frac{\beta_{\mathbf{y}_0}}{2}(\mu-\bar{\mu}_0)^T\mathbf{B}(\mu-\bar{\mu}_0)\\
&+\frac{\nu_{\mathbf{y}_0}-d-1}{2}\ln|\mathbf{B}|-\frac{1}{2}\mathrm{tr}\left(\mathbf{B}\mathbf{\Psi}_{\mathbf{y}_0}^{-1}\right)+\mathrm{const}
\end{aligned} \tag{E.117}$$

$$\begin{aligned}
=&-\frac{1}{2}\mathrm{tr}\left(\mathbf{B}\left(M\left(\mu-\bar{\mathbf{y}}\right)\left(\mu-\bar{\mathbf{y}}\right)^T+\beta_{\mathbf{y}_0}\left(\mu-\bar{\mu}_0\right)\left(\mu-\bar{\mu}_0\right)^T\right)\right)\\
&+\frac{1}{2}\ln|\mathbf{B}|+\frac{\nu_{\mathbf{y}_0}+M-d-1}{2}\ln|\mathbf{B}|-\frac{1}{2}\mathrm{tr}\left(\mathbf{B}\left(\mathbf{\Psi}_{\mathbf{y}_0}^{-1}+\mathbf{S}_{\mathbf{y}}\right)\right)+\mathrm{const}
\end{aligned} \tag{E.118}$$

$$\begin{aligned}
=&-\frac{1}{2}\mathrm{tr}\left(\mathbf{B}\left(M\left(\mu\mu^T-\mu\bar{\mathbf{y}}^T-\bar{\mathbf{y}}\mu^T+\bar{\mathbf{y}}\bar{\mathbf{y}}^T\right)+\beta_{\mathbf{y}_0}\left(\mu\mu^T-\mu\bar{\mu}_0^T-\bar{\mu}_0\mu^T+\bar{\mu}_0\bar{\mu}_0^T\right)\right)\right)\\
&+\frac{1}{2}\ln|\mathbf{B}|+\frac{\nu_{\mathbf{y}}-d-1}{2}\ln|\mathbf{B}|-\frac{1}{2}\mathrm{tr}\left(\mathbf{B}\left(\mathbf{\Psi}_{\mathbf{y}_0}^{-1}+\mathbf{S}_{\mathbf{y}}\right)\right)+\mathrm{const}
\end{aligned} \tag{E.119}$$

$$\begin{aligned}
=&-\frac{1}{2}\mathrm{tr}\left(\mathbf{B}\left(\left(M+\beta_{\mathbf{y}_0}\right)\left(\mu\mu^T\right.\right.\right.\\
&-\frac{1}{M+\beta_{\mathbf{y}_0}}\mu\left(M\bar{\mathbf{y}}+\beta_{\mathbf{y}_0}\bar{\mu}_0\right)^T-\frac{1}{M+\beta_{\mathbf{y}_0}}\left(M\bar{\mathbf{y}}+\beta_{\mathbf{y}_0}\bar{\mu}_0\right)\mu^T\\
&\left.\left.\left.+\frac{1}{M+\beta_{\mathbf{y}_0}}\left(M\bar{\mathbf{y}}\bar{\mathbf{y}}^T+\beta_{\mathbf{y}_0}\bar{\mu}_0\bar{\mu}_0^T\right)\right)\right)\right)\\
&+\frac{1}{2}\ln|\mathbf{B}|+\frac{\nu_{\mathbf{y}}-d-1}{2}\ln|\mathbf{B}|-\frac{1}{2}\mathrm{tr}\left(\mathbf{B}\left(\mathbf{\Psi}_{\mathbf{y}_0}^{-1}+\mathbf{S}_{\mathbf{y}}\right)\right)+\mathrm{const}
\end{aligned} \tag{E.120}$$

$$\begin{aligned}
=&\frac{1}{2}\ln|\mathbf{B}|-\frac{\beta_{\mathbf{y}}}{2}(\mu-\bar{\mu})^T\mathbf{B}(\mu-\bar{\mu})\\
&-\frac{1}{2}\mathrm{tr}\left(\mathbf{B}\left(M\bar{\mathbf{y}}\bar{\mathbf{y}}^T+\beta_{\mathbf{y}_0}\bar{\mu}_0\bar{\mu}_0^T-\frac{1}{M+\beta_{\mathbf{y}_0}}\left(M\bar{\mathbf{y}}+\beta_{\mathbf{y}_0}\bar{\mu}_0\right)\left(M\bar{\mathbf{y}}+\beta_{\mathbf{y}_0}\bar{\mu}_0\right)^T\right)\right)\\
&+\frac{\nu_{\mathbf{y}}-d-1}{2}\ln|\mathbf{B}|-\frac{1}{2}\mathrm{tr}\left(\mathbf{B}\left(\mathbf{\Psi}_{\mathbf{y}_0}^{-1}+\mathbf{S}_{\mathbf{y}}\right)\right)+\mathrm{const}
\end{aligned} \tag{E.121}$$

$$
\begin{aligned}
=&\frac{1}{2}\ln|\mathbf{B}| - \frac{\beta_{\mathbf{y}}}{2}(\mu-\overline{\mu})^T\mathbf{B}(\mu-\overline{\mu}) + \frac{\nu_{\mathbf{y}}-d-1}{2}\ln|\mathbf{B}| \\
&-\frac{1}{2}\mathrm{tr}\left(\mathbf{B}\left(\boldsymbol{\Psi}_{\mathbf{y}_0}^{-1}+\mathbf{S}_{\mathbf{y}}+\frac{\beta_{\mathbf{y}_0}M}{M+\beta_{\mathbf{y}_0}}\left(\overline{\mathbf{y}}-\overline{\mu}_0\right)\left(\overline{\mathbf{y}}-\overline{\mu}_0\right)^T\right)\right)+\mathrm{const} \qquad \text{(E.122)}\\
=&\left[\frac{1}{2}\ln|\mathbf{B}| - \frac{\beta_{\mathbf{y}}}{2}(\mu-\overline{\mu})^T\mathbf{B}(\mu-\overline{\mu})\right] + \left[\frac{\nu_{\mathbf{y}}-d-1}{2}\ln|\mathbf{B}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{B}\boldsymbol{\Psi}_{\mathbf{y}}^{-1}\right)\right]+\mathrm{const} \qquad \text{(E.123)}
\end{aligned}
$$

obtaining that $q^*\left(\mathcal{M}_{\mathbf{y}}\right)$ is Gaussian-Wishart distributed:

$$q^*\left(\mathcal{M}_{\mathbf{y}}\right)=\mathcal{N}\left(\mu|\overline{\mu},\left(\beta_{\mathbf{y}}\mathbf{B}\right)^{-1}\right)\mathcal{W}\left(\mathbf{B}|\boldsymbol{\Psi}_{\mathbf{y}},\nu_{\mathbf{y}}\right) \quad \text{if } \nu_{\mathbf{y}}>d\,. \qquad \text{(E.124)}$$

For $q^*\left(\mathbf{W}\right)$, we substitute (E.3), and (E.99) into (E.107) and absorb any term that does not depend on $\mathbf{W}$ into the additive constant:

$$
\begin{aligned}
\ln q^*\left(\mathbf{W}\right)=&\sum_{i=1}^{M}\sum_{j=1}^{N_i}\mathrm{E}_{\mathbf{Y}}\left[\ln P\left(\phi_{ij}|\mathbf{y}_i,\mathbf{W}\right)\right]+\mathrm{E}_{\mathbf{Y}}\left[\ln P\left(\mathbf{W}|\Pi_{\mathbf{W}}\right)\right]+\mathrm{const} \qquad \text{(E.125)}\\
=&\frac{N}{2}\ln|\mathbf{W}|-\frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{N_i}\mathrm{E}_{\mathbf{Y}}\left[(\phi_{ij}-\mathbf{y}_i)^T\mathbf{W}(\phi_{ij}-\mathbf{y}_i)\right]\\
&+\frac{\nu_{\mathbf{W}_0}-d-1}{2}\ln|\mathbf{W}|-\frac{1}{2}\mathrm{tr}\left(\mathbf{W}\boldsymbol{\Psi}_{\mathbf{W}_0}^{-1}\right)+\mathrm{const}\,. \qquad \text{(E.126)}
\end{aligned}
$$

We define

$$
\begin{aligned}
\mathbf{S}_{\mathbf{W}}=&\sum_{i=1}^{M}\sum_{j=1}^{N_i}\mathrm{E}_{\mathbf{Y}}\left[\left(\phi_{ij}-\mathbf{y}_i\right)\left(\phi_{ij}-\mathbf{y}_i\right)^T\right] \qquad \text{(E.127)}\\
\nu_{\mathbf{W}}=&\nu_{\mathbf{W}_0}+N \qquad \text{(E.128)}\\
\boldsymbol{\Psi}_{\mathbf{W}}^{-1}=&\boldsymbol{\Psi}_{\mathbf{W}_0}^{-1}+\mathbf{S}_{\mathbf{W}} \qquad \text{(E.129)}
\end{aligned}
$$

and write (E.126) as

$$\ln q^*\left(\mathbf{W}\right)=\frac{\nu_{\mathbf{W}}-d-1}{2}\ln|\mathbf{W}|-\frac{1}{2}\mathrm{tr}\left(\mathbf{W}\boldsymbol{\Psi}_{\mathbf{W}}^{-1}\right)+\mathrm{const}\,. \qquad \text{(E.130)}$$

Thus, $q^*\left(\mathbf{W}\right)$ is Wishart distributed:

$$q^*\left(\mathbf{W}\right)=\mathcal{W}\left(\mathbf{W}|\boldsymbol{\Psi}_{\mathbf{W}},\nu_{\mathbf{W}}\right) \quad \text{if } \nu_{\mathbf{W}}>d\,. \qquad \text{(E.131)}$$

Finally, we need to evaluate the expectations $\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\mathbf{B}\right]$, $\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\mathbf{B}\mu\right]$, $\mathrm{E}_{\mathbf{W}}\left[\mathbf{W}\right]$, $\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]$ and $\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right]$ by applying the properties of the Gaussian and Wishart distributions [Bishop, 2006]:

$$
\begin{aligned}
\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\mathbf{B}\right]=&\nu_{\mathbf{y}}\boldsymbol{\Psi}_{\mathbf{y}} \qquad \text{(E.132)}\\
\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\mathbf{B}\mu\right]=&\nu_{\mathbf{y}}\boldsymbol{\Psi}_{\mathbf{y}}\overline{\mu} \qquad \text{(E.133)}\\
\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\mathbf{W}\right]=&\nu_{\mathbf{W}}\boldsymbol{\Psi}_{\mathbf{W}} \qquad \text{(E.134)}\\
\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]=&\mathbf{L}_i^{-1}\gamma_i \qquad \text{(E.135)}\\
\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right]=&\mathbf{L}_i^{-1}+\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\right]^T\ . \qquad \text{(E.136)}
\end{aligned}
$$

E.4.3 Variational lower bound

In this section, we evaluate the lower bound used to check the convergence of our algorithm.

$$\mathcal{L} = \int_{\mathbf{Y}}\int_{\mathbf{W}}\int_{\mathbf{B}}\int_{\mu} q\left(\mu,\mathbf{B},\mathbf{W},\mathbf{Y}\right)\ln\left(\frac{P\left(\mathbf{\Phi},\mu,\mathbf{B},\mathbf{W},\mathbf{Y}|\theta,\Pi\right)}{q\left(\mu,\mathbf{B},\mathbf{W},\mathbf{Y}\right)}\right) \mathrm{d}\mu\,\mathrm{d}\mathbf{B}\,\mathrm{d}\mathbf{W}\,\mathrm{d}\mathbf{Y} \tag{E.137}$$

$$=\mathrm{E}_{\mathcal{M},\mathbf{Y}}\left[\ln P\left(\mathbf{\Phi},\mu,\mathbf{B},\mathbf{W},\mathbf{Y}|\theta,\Pi\right)\right] - \mathrm{E}_{\mathcal{M},\mathbf{Y}}\left[\ln q\left(\mu,\mathbf{B},\mathbf{W},\mathbf{Y}\right)\right] \tag{E.138}$$

$$\begin{aligned}
=&\mathrm{E}_{\mathbf{W},\mathbf{Y}}\left[\ln P\left(\mathbf{\Phi}|\mathbf{Y},\mathbf{W},\theta\right)\right] + \mathrm{E}_{\mathcal{M}_{\mathbf{y}},\mathbf{Y}}\left[\ln P\left(\mathbf{Y}|\mathcal{M}_{\mathbf{y}}\right)\right]\\
&+\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\ln P\left(\mathcal{M}_{\mathbf{y}}|\Pi_{\mathcal{M}_{\mathbf{y}}}\right)\right] + \mathrm{E}_{\mathbf{W}}\left[\ln P\left(\mathbf{W}|\Pi_{\mathbf{W}}\right)\right]\\
&-\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\ln q\left(\mu,\mathbf{B}\right)\right] - \mathrm{E}_{\mathbf{W}}\left[\ln q\left(\mathbf{W}\right)\right] - \mathrm{E}_{\mathbf{Y}}\left[\ln q\left(\mathbf{Y}\right)\right]\ .
\end{aligned} \tag{E.139}$$

We evaluate $\mathrm{E}_{\mathbf{Y}}\left[\ln P\left(\mathbf{\Phi}|\mathbf{Y},\mathbf{W},\theta\right)\right]$:

$$\mathrm{E}_{\mathbf{Y}}\left[\ln P\left(\mathbf{\Phi}|\mathbf{Y},\mathbf{W},\theta\right)\right] = \frac{N}{2}\mathrm{E}_{\mathbf{W}}\left[\ln|\mathbf{W}|\right] - \frac{Nd}{2}\ln(2\pi) - \frac{1}{2}\mathrm{tr}\left(\mathrm{E}_{\mathbf{W}}\left[\mathbf{W}\right]\mathbf{S}_{\mathbf{W}}\right) \tag{E.140}$$

$$=\frac{N}{2}\ln\tilde{\mathbf{W}} - \frac{Nd}{2}\ln(2\pi) - \frac{\nu_{\mathbf{W}}}{2}\mathrm{tr}\left(\mathbf{\Psi}_{\mathbf{W}}\mathbf{S}_{\mathbf{W}}\right) \tag{E.141}$$

where

$$\ln\tilde{\mathbf{W}} \equiv \mathrm{E}_{\mathbf{W}}\left[\ln|\mathbf{W}|\right] = \sum_{i=1}^{d}\psi\left(\frac{\nu_{\mathbf{W}}+1-i}{2}\right) + d\ln 2 + \ln|\mathbf{\Psi}_{\mathbf{W}}|\ . \tag{E.142}$$

Now, we evaluate $\mathrm{E}_{\mathcal{M}_{\mathbf{y}},\mathbf{Y}}\left[\ln P\left(\mathbf{Y}|\mathcal{M}_{\mathbf{y}}\right)\right]$:

$$\begin{aligned}
\mathrm{E}_{\mathcal{M}_{\mathbf{y}},\mathbf{Y}}\left[\ln P\left(\mathbf{Y}|\mathcal{M}_{\mathbf{y}}\right)\right] =& \frac{M}{2}\mathrm{E}_{\mathbf{B}}\left[\ln|\mathbf{B}|\right] - \frac{Md}{2}\ln(2\pi) + M\overline{\mathbf{y}}^{T}\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\mathbf{B}\mu\right]\\
&-\frac{1}{2}\mathrm{tr}\left(\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\mathbf{B}\right]\sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^{T}\right] + M\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\mathbf{B}\mu\mu^{T}\right]\right)\ .
\end{aligned} \tag{E.143}$$

To evaluate (E.143) we need to calculate some new expectations. Using the properties of the Gaussian and Wishart distributions [Bishop, 2006] we have

$$\ln\tilde{\mathbf{B}} \equiv \mathrm{E}_{\mathbf{B}}\left[\ln|\mathbf{B}|\right] = \sum_{i=1}^{d}\psi\left(\frac{\nu_{\mathbf{y}}+1-i}{2}\right) + d\ln 2 + \ln|\mathbf{\Psi}_{\mathbf{y}}| \tag{E.144}$$

and

$$\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\mathbf{B}\mu\mu^{T}\right] = \int_{\mathbf{B}}\int_{\mu}\mathbf{B}\mu\mu^{T}\mathcal{N}\left(\mu|\overline{\mu},\left(\beta_{\mathbf{y}}\mathbf{B}\right)^{-1}\right)\mathcal{W}\left(\mathbf{B}|\mathbf{\Psi}_{\mathbf{y}},\nu_{\mathbf{y}}\right)\,\mathrm{d}\mu\,\mathrm{d}\mathbf{B} \tag{E.145}$$

$$=\int_{\mathbf{B}}\mathbf{B}\int_{\mu}\mu\mu^{T}\mathcal{N}\left(\mu|\overline{\mu},\left(\beta_{\mathbf{y}}\mathbf{B}\right)^{-1}\right)\,\mathrm{d}\mu\,\mathcal{W}\left(\mathbf{B}|\mathbf{\Psi}_{\mathbf{y}},\nu_{\mathbf{y}}\right)\,\mathrm{d}\mathbf{B} \tag{E.146}$$

$$=\int_{\mathbf{B}}\mathbf{B}\left(\left(\beta_{\mathbf{y}}\mathbf{B}\right)^{-1} + \overline{\mu\mu}^{T}\right)\mathcal{W}\left(\mathbf{B}|\mathbf{\Psi}_{\mathbf{y}},\nu_{\mathbf{y}}\right)\,\mathrm{d}\mathbf{B} \tag{E.147}$$

$$=\beta_{\mathbf{y}}^{-1}\mathbf{I} + \int_{\mathbf{B}}\mathbf{B}\mathcal{W}\left(\mathbf{B}|\mathbf{\Psi}_{\mathbf{y}},\nu_{\mathbf{y}}\right)\,\mathrm{d}\mathbf{B}\,\overline{\mu\mu}^{T} \tag{E.148}$$

$$=\beta_{\mathbf{y}}^{-1}\mathbf{I} + \nu_{\mathbf{y}}\mathbf{\Psi}_{\mathbf{y}}\overline{\mu\mu}^{T}\ . \tag{E.149}$$

Now, we can plug (E.112), (E.132), (E.133), (E.144) and (E.149) into (E.143)

$$\begin{aligned}
\mathrm{E}_{\mathcal{M}_{\mathbf{y}},\mathbf{Y}}\left[\ln P\left(\mathbf{Y}|\mathcal{M}_{\mathbf{y}}\right)\right] =& \frac{M}{2}\ln\tilde{\mathbf{B}} - \frac{Md}{2}\ln(2\pi) + M\overline{\mathbf{y}}^T\nu_{\mathbf{y}}\boldsymbol{\Psi}_{\mathbf{y}}\overline{\mu} \\
&- \frac{1}{2}\mathrm{tr}\left(\nu_{\mathbf{y}}\boldsymbol{\Psi}_{\mathbf{y}}\sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right] + M\left(\beta_{\mathbf{y}}^{-1}\mathbf{I} + \nu_{\mathbf{y}}\boldsymbol{\Psi}_{\mathbf{y}}\overline{\mu}\overline{\mu}^T\right)\right) \qquad \text{(E.150)} \\
=& \frac{M}{2}\ln\tilde{\mathbf{B}} - \frac{Md}{2}\ln(2\pi) + M\overline{\mathbf{y}}^T\nu_{\mathbf{y}}\boldsymbol{\Psi}_{\mathbf{y}}\overline{\mu} \\
&- \frac{1}{2}\mathrm{tr}\left(\frac{M}{\beta_{\mathbf{y}}}\mathbf{I} + \nu_{\mathbf{y}}\boldsymbol{\Psi}_{\mathbf{y}}\left(\sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}\left[\mathbf{y}_i\mathbf{y}_i^T\right] + M\overline{\mu}\overline{\mu}^T\right)\right) \qquad \text{(E.151)} \\
=& \frac{M}{2}\ln\tilde{\mathbf{B}} - \frac{Md}{2}\ln(2\pi) - \frac{Md}{2\beta_{\mathbf{y}}} \\
&- \frac{M\nu_{\mathbf{y}}}{2}(\overline{\mu} - \overline{\mathbf{y}})^T\boldsymbol{\Psi}_{\mathbf{y}}(\overline{\mu} - \overline{\mathbf{y}}) - \frac{\nu_{\mathbf{y}}}{2}\mathrm{tr}\left(\boldsymbol{\Psi}_{\mathbf{y}}\mathbf{S}_{\mathbf{y}}\right)\ . \qquad \text{(E.152)}
\end{aligned}$$

Now, we evaluate $\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\ln P\left(\mathcal{M}_{\mathbf{y}}|\Pi_{\mathcal{M}_{\mathbf{y}}}\right)\right]$:

$$\begin{aligned}
\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\ln P\left(\mathcal{M}_{\mathbf{y}}|\Pi_{\mathcal{M}_{\mathbf{y}}}\right)\right] =& \mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\ln\mathcal{N}\left(\mu|\overline{\mu}_0,(\beta_{\mathbf{y}_0}\mathbf{B})^{-1}\right)\right] + \mathrm{E}_{\mathbf{B}}\left[\ln\mathcal{W}\left(\mathbf{B}|\boldsymbol{\Psi}_{\mathbf{y}_0},\nu_{\mathbf{y}_0}\right)\right] \qquad \text{(E.153)} \\
=& \frac{d}{2}\ln\left(\frac{\beta_{\mathbf{y}_0}}{2\pi}\right) + \frac{1}{2}\ln\tilde{\mathbf{B}} - \frac{\beta_{\mathbf{y}_0}}{2}\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[(\mu - \overline{\mu}_0)^T\mathbf{B}(\mu - \overline{\mu}_0)\right] \\
&+ \ln B\left(\boldsymbol{\Psi}_{\mathbf{y}_0},\nu_{\mathbf{y}_0}\right) + \frac{\nu_{\mathbf{y}_0} - d - 1}{2}\ln\tilde{\mathbf{B}} - \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Psi}_{\mathbf{y}_0}^{-1}\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[B\right]\right) \\
& \qquad \text{(E.154)}
\end{aligned}$$

where

$$\begin{aligned}
\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[(\mu - \overline{\mu}_0)^T\mathbf{B}(\mu - \overline{\mu}_0)\right] =& \mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\mu^T\mathbf{B}\mu\right] - 2\overline{\mu}_0^T\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\mathbf{B}\mu\right] + \overline{\mu}_0^T\mathrm{E}_{\mathbf{B}}\left[\mathbf{B}\right]\overline{\mu}_0 \qquad \text{(E.155)} \\
=& \mathrm{tr}\left(\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\mathbf{B}\mu\mu^T\right]\right) - 2\overline{\mu}_0^T\nu_{\mathbf{y}}\boldsymbol{\Psi}_{\mathbf{y}}\overline{\mu} + \overline{\mu}_0^T\nu_{\mathbf{y}}\boldsymbol{\Psi}_{\mathbf{y}}\overline{\mu}_0 \qquad \text{(E.156)} \\
=& \mathrm{tr}\left(\beta_{\mathbf{y}}^{-1}\mathbf{I} + \nu_{\mathbf{y}}\boldsymbol{\Psi}_{\mathbf{y}}\overline{\mu}\overline{\mu}^T\right) - 2\overline{\mu}_0^T\nu_{\mathbf{y}}\boldsymbol{\Psi}_{\mathbf{y}}\overline{\mu} + \overline{\mu}_0^T\nu_{\mathbf{y}}\boldsymbol{\Psi}_{\mathbf{y}}\overline{\mu}_0 \qquad \text{(E.157)} \\
=& \frac{d}{\beta_{\mathbf{y}}} + \nu_{\mathbf{y}}(\overline{\mu} - \overline{\mu}_0)^T\boldsymbol{\Psi}_{\mathbf{y}}(\overline{\mu} - \overline{\mu}_0)\ . \qquad \text{(E.158)}
\end{aligned}$$

We plug (E.158) into (E.154):

$$\begin{aligned}
\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\ln P\left(\mathcal{M}_{\mathbf{y}}|\Pi_{\mathcal{M}_{\mathbf{y}}}\right)\right] =& \frac{d}{2}\ln\left(\frac{\beta_{\mathbf{y}_0}}{2\pi}\right) + \frac{1}{2}\ln\tilde{\mathbf{B}} - \frac{d\beta_{\mathbf{y}_0}}{2\beta_{\mathbf{y}}} - \frac{\beta_{\mathbf{y}_0}\nu_{\mathbf{y}}}{2}(\overline{\mu} - \overline{\mu}_0)^T\boldsymbol{\Psi}_{\mathbf{y}}(\overline{\mu} - \overline{\mu}_0) \\
&+ \ln B\left(\boldsymbol{\Psi}_{\mathbf{y}_0},\nu_{\mathbf{y}_0}\right) + \frac{\nu_{\mathbf{y}_0} - d - 1}{2}\ln\tilde{\mathbf{B}} - \frac{\nu_{\mathbf{y}}}{2}\mathrm{tr}\left(\boldsymbol{\Psi}_{\mathbf{y}_0}^{-1}\boldsymbol{\Psi}_{\mathbf{y}}\right)\ . \qquad \text{(E.159)}
\end{aligned}$$

Now, we evaluate $\mathrm{E}_{\mathbf{W}}\left[\ln P\left(\mathbf{W}|\Pi_{\mathbf{W}}\right)\right]$

$$\begin{aligned}
\mathrm{E}_{\mathbf{W}}\left[\ln P\left(\mathbf{W}|\Pi_{\mathbf{W}}\right)\right] =& \mathrm{E}_{\mathbf{W}}\left[\ln\mathcal{W}\left(\mathbf{W}|\boldsymbol{\Psi}_{\mathbf{W}_0},\nu_{\mathbf{W}_0}\right)\right] \qquad \text{(E.160)} \\
=& \ln B\left(\boldsymbol{\Psi}_{\mathbf{W}_0},\nu_{\mathbf{W}_0}\right) + \frac{\nu_{\mathbf{W}_0} - d - 1}{2}\ln\tilde{\mathbf{W}} - \frac{\nu_{\mathbf{W}}}{2}\mathrm{tr}\left(\boldsymbol{\Psi}_{\mathbf{W}_0}^{-1}\boldsymbol{\Psi}_{\mathbf{W}}\right)\ . \qquad \text{(E.161)}
\end{aligned}$$

Now, we evaluate $\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\ln q\left(\mu,\mathbf{B}\right)\right]$

$$\begin{aligned}
\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\ln q\left(\mu,\mathbf{B}\right)\right] =& \mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\ln\mathcal{N}\left(\mu|\overline{\mu},(\beta_{\mathbf{y}}\mathbf{B})^{-1}\right)\right] + \mathrm{E}_{\mathbf{B}}\left[\ln\mathcal{W}\left(\mathbf{B}|\boldsymbol{\Psi}_{\mathbf{y}},\nu_{\mathbf{y}}\right)\right] \qquad \text{(E.162)} \\
=& \frac{d}{2}\ln\left(\frac{\beta_{\mathbf{y}}}{2\pi}\right) + \frac{1}{2}\ln\tilde{\mathbf{B}} - \frac{\beta_{\mathbf{y}}}{2}\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[(\mu - \overline{\mu})^T\mathbf{B}(\mu - \overline{\mu})\right] - \mathrm{H}\left[q\left(\mathbf{B}\right)\right] \qquad \text{(E.163)}
\end{aligned}$$

where $\mathrm{H}\left[q\left(\mathbf{B}\right)\right]$ is the entropy of the Wishart distribution [Bishop, 2006]

$$\begin{aligned}
\mathrm{H}\left[q\left(\mathbf{B}\right)\right] =& \mathrm{H}\left[\mathcal{W}\left(\mathbf{B}|\boldsymbol{\Psi}_{\mathbf{y}},\nu_{\mathbf{y}}\right)\right] && \text{(E.164)}\\
=& -\ln B\left(\boldsymbol{\Psi}_{\mathbf{y}},\nu_{\mathbf{y}}\right)-\frac{\nu_{\mathbf{y}}-d-1}{2}\ln\tilde{\mathbf{B}}+\frac{\nu_{\mathbf{y}}d}{2} && \text{(E.165)}\\
B(\mathbf{W},N) =& \frac{1}{2^{Nd/2}Z_{Nd}}\left|\mathbf{W}\right|^{-N/2} && \text{(E.166)}
\end{aligned}$$

and

$$\begin{aligned}
\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[(\mu-\overline{\mu})^T\mathbf{B}(\mu-\overline{\mu})\right] =& \mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\mu^T\mathbf{B}\mu\right]-2\overline{\mu}^T\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\mathbf{B}\mu\right]+\overline{\mu}^T\mathrm{E}_{\mathbf{B}}\left[\mathbf{B}\right]\overline{\mu} && \text{(E.167)}\\
=& \mathrm{tr}\left(\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\mathbf{B}\mu\mu^T\right]\right)-\overline{\mu}^T\mathrm{E}_{\mathbf{B}}\left[\mathbf{B}\right]\overline{\mu} && \text{(E.168)}\\
=& \mathrm{tr}\left(\beta_{\mathbf{y}}^{-1}\mathbf{I}+\nu_{\mathbf{y}}\boldsymbol{\Psi}_{\mathbf{y}}\overline{\mu}\overline{\mu}^T\right)-\overline{\mu}^T\nu_{\mathbf{y}}\boldsymbol{\Psi}_{\mathbf{y}}\overline{\mu} && \text{(E.169)}\\
=& d\beta_{\mathbf{y}}^{-1}\;. && \text{(E.170)}
\end{aligned}$$

We plug (E.170) into (E.163):

$$\mathrm{E}_{\mathcal{M}_{\mathbf{y}}}\left[\ln q\left(\mu,\mathbf{B}\right)\right] =\frac{d}{2}\ln\left(\frac{\beta_{\mathbf{y}}}{2\pi}\right)+\frac{1}{2}\ln\tilde{\mathbf{B}}-\frac{d}{2}-\mathrm{H}\left[q\left(\mathbf{B}\right)\right]\;. \qquad \text{(E.171)}$$

Now, we evaluate $\mathrm{E}_{\mathbf{W}}\left[\ln q\left(\mathbf{W}\right)\right]$

$$\mathrm{E}_{\mathbf{W}}\left[\ln q\left(\mathbf{W}\right)\right] =\mathrm{E}_{\mathbf{W}}\left[\ln\mathcal{W}\left(\mathbf{W}|\boldsymbol{\Psi}_{\mathbf{W}},\nu_{\mathbf{W}}\right)\right]=-\mathrm{H}\left[q\left(\mathbf{W}\right)\right] \qquad \text{(E.172)}$$

where $\mathrm{H}\left[q\left(\mathbf{W}\right)\right]$ is the entropy of the Wishart distribution [Bishop, 2006].

$$\begin{aligned}
\mathrm{H}\left[q\left(\mathbf{W}\right)\right] =& \mathrm{H}\left[\mathcal{W}\left(\mathbf{W}|\boldsymbol{\Psi}_{\mathbf{W}},\nu_{\mathbf{W}}\right)\right] && \text{(E.173)}\\
=& -\ln B\left(\boldsymbol{\Psi}_{\mathbf{W}},\nu_{\mathbf{W}}\right)-\frac{\nu_{\mathbf{W}}-d-1}{2}\ln\tilde{\mathbf{W}}+\frac{\nu_{\mathbf{W}}d}{2}\;. && \text{(E.174)}
\end{aligned}$$

Finally, we evaluate $\mathrm{E}_{\mathbf{Y}}\left[\ln q\left(\mathbf{Y}\right)\right]$:

$$\begin{aligned}
\mathrm{E}_{\mathbf{Y}}\left[\ln q\left(\mathbf{Y}\right)\right] =& \sum_{i=1}^{M}\mathrm{E}_{\mathbf{Y}}\left[\ln\mathcal{N}\left(\mathbf{y}_i|\mathbf{L}_i^{-1}\gamma_i,\mathbf{L}_i^{-1}\right)\right] && \text{(E.175)}\\
=& -\frac{Md}{2}(\ln(2\pi)+1)+\frac{1}{2}\sum_{i=1}^{M}\ln\left|\mathbf{L}_i\right|\;. && \text{(E.176)}
\end{aligned}$$

Bibliography

[Abe et al., 1988] Abe, M., Nakamura, S., Shikano, K., and Kuwabara, H. (1988). Voice conversion through vector quantization. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 1988*, pages 655–658, New York, NY, USA. IEEE. Available from: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=196671.

[Acero et al., 2000] Acero, A., Deng, L., Kristjansson, T., and Zhang, J. (2000). HMM adaptation using vector taylor series for noisy speech recognition. In *Proceedings of the 6th International Conference on Spoken Language Processing, ICSLP - Interspeech 2000*, pages 869–872, Beijing, China. ISCA. Available from: http://research.microsoft.com/pubs/76536/2000-alexac-icslpb.pdf.

[Alam et al., 2011] Alam, M. J., Ouellet, P., Kenny, P., and Shaughnessy, D. (2011). Comparative Evaluation of Feature Normalization Techniques for Speaker Verification. In Travieso-González, C. and Alonso-Hernández, J., editors, *Advances in Nonlinear Speech Processing, 5th International Conference on Nonlinear Speech Processing, NOLISP 2011*, volume 7015 of *Lecture Notes in Computer Science*, pages 246–253, Las Palmas de Gran Canaria, Spain. Springer-Verlag Berlin, Heidelberg. Available from: http://dx.doi.org/10.1007/978-3-642-25020-0_32.

[Alegre et al., 2013a] Alegre, F., Amehraye, A., and Evans, N. (2013a). Spoofing countermeasures to protect automatic speaker verification from voice conversion. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013*, pages 3068–3072, Vancouver, British Columbia, Canada. IEEE. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6638222.

[Alegre et al., 2013b] Alegre, F., Vipperla, R., Amehraye, A., and Evans, N. (2013b). A New Speaker Verification Spoofing Countermeasure Based on Local Binary Patterns. In *Proceedings of the 14th Annual Conference of the International Speech Communication Association, Interspeech 2013*, pages 940–944, Lyon, France. ISCA.

[Alegre et al., 2012a] Alegre, F., Vipperla, R., and Evans, N. (2012a). Spoofing countermeasures for the protection of automatic speaker recognition from attacks with artificial signals. In *Proceedings of the 13th Annual Conference of the International Speech Communication Association, Interspeech 2012*, Portland, Oregon, USA. ISCA.

[Alegre et al., 2012b] Alegre, F., Vipperla, R., Evans, N., and Fauve, B. (2012b). On the Vulnerability of Automatic Speaker Recognition to Spoofing Attacks with Artificial

Signals. In *Proceedings of the 20th European Signal Processing Conference, EUSIPCO 2012*, pages 36–40, Bucharest, Romania. IEEE.

[Alexander et al., 2004] Alexander, A., Botti, F., Dessimoz, D., and Drygajlo, A. (2004). The effect of mismatched recording conditions on human and automatic speaker recognition in forensic applications. *Forensic Science International*, 146 Suppl:S95–S99. Available from: http://www.ncbi.nlm.nih.gov/pubmed/15639600.

[Amin et al., 2013] Amin, T. B., German, J. S., and Marziliano, P. (2013). Detecting voice disguise from speech variability: Analysis of three glottal and vocal tract measures. *The Journal of the Acoustical Society of America*, 134(5):4068. Available from: http://link.aip.org/link/JASMAN/v134/i5/p4068/s4&Agg=doi.

[Anjos and Marcel, 2011] Anjos, A. and Marcel, S. (2011). Counter-measures to photo attacks in face recognition: A public database and a baseline. In *Proceedings of the 2011 International Joint Conference on Biometrics (IJCB)*, pages 1–7, Washington, DC, USA. IEEE. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6117503.

[Aronowitz et al., 2011] Aronowitz, H., Hoory, R., Pelecanos, J., Nahamoo, D., and Heights, Y. (2011). New Developments in Voice Biometrics for User Authentication. In *Proceedings of the 12th Annual Conference of the International Speech Communication Association, Interspeech 2011*, pages 17–20, Florence, Italy. ISCA. Available from: https://docs.google.com/file/d/0B6JITGelC0ySOTAxNGFkM2EtNGNlNy00N2U5LWEyNzQtOWE4NGY5MjdjNTll/edit?hl=en_US.

[Atal, 1974] Atal, B. S. (1974). Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. *Journal of the Acoustical Society of America*, 55(6):1304–1322. Available from: http://link.aip.org/link/?JASMAN/55/1304/1.

[Auckenthaler et al., 2000] Auckenthaler, R., Carey, M., and Lloyd-Thomas, H. (2000). Score Normalization for Text-Independent Speaker Verification Systems. *Digital Signal Processing*, 10(1-3):42–54. Available from: http://linkinghub.elsevier.com/retrieve/pii/S1051200499903603.

[Aversano, 2009] Aversano, G. (2009). Evalita 2009 Speaker Identity Verification - Application
Track Guidelines. Available from: http://www.evalita.it/sites/evalita.fbk.eu/files/doc2009/Guidelines_evalita09_SIV-Application_track.pdf.

[Bengio et al., 2002] Bengio, S., Marcel, C., Marcel, S., and Mariethoz, J. (2002). Confidence Measures for Multimodal Identity Verification. *Information Fusion*, 3(4):267–276. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.8.3606.

[Bimbot et al., 2004] Bimbot, F., Bonastre, J.-F., Fredouille, C., Gravier, G., Magrin-Chagnolleau, I., Meignier, S., Merlin, T., Ortega-García, J., Petrovska-Delacrétaz, D., and Reynolds, D. A. (2004). A Tutorial on Text-Independent Speaker Verification. *EURASIP*

Journal on Advances in Signal Processing, 2004(4):430–451. Available from: http://www.hindawi.com/journals/asp/2004/101962.abs.html.

[Bishop, 2006] Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer Science+Business Media, LLC.

[Blomberg et al., 2004] Blomberg, M., Elenius, D., and Zetterholm, E. (2004). Speaker verification scores and acoustic analysis of a professional impersonator. In Branderud, P. and Traunmuller, H., editors, *Proceedings of Fonetik 2004: The 17th Swedish Phonetics Conference*, pages 84–87, Stockholm, Sweden. Available from: http://www.speech.kth.se/prod/publications/files/1035.pdf.

[Bonastre et al., 2007] Bonastre, J.-F., Matrouf, D., and Fredouille, C. (2007). Artificial impostor voice transformation effects on false acceptance rates. In *Proceedings of the 8th Annual Conference of the International Speech Communication Association, Interspeech 2007*, pages 2053–2056, Antwerp, Belgium. ISCA.

[Bredin et al., 2006] Bredin, H., Miguel, A., Witten, I. H., and Chollet, G. (2006). Detecting replay attacks in audiovisual identity verification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2006*, pages 621–624, Toulouse, France. IEEE. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.126.3171.

[Brummer, 2006] Brummer, N. (2006). Focal toolkit. Available from: https://sites.google.com/site/nikobrummer/focal.

[Brummer, 2009] Brummer, N. (2009). The EM algorithm and Minimum Divergence. Technical report, Agnitio Research, Cape Town, South Africa. Available from: https://sites.google.com/site/nikobrummer/EMandMINDIV.pdf.

[Brummer, 2010a] Brummer, N. (2010a). Bayesian PLDA. Technical report, Agnitio Research, Cape Town, South Africa. Available from: https://sites.google.com/site/nikobrummer/bplda.pdf.

[Brummer, 2010b] Brummer, N. (2010b). EM for Probabilistic LDA. Technical Report February, Agnitio Research, Cape Town, South Africa. Available from: https://sites.google.com/site/nikobrummer/EMforPLDA.pdf.

[Brummer, 2010c] Brummer, N. (2010c). *Measuring, refining and calibrating speaker and language information extracted from speech*. PhD thesis, University of Stellenbosch. Available from: https://scholar.sun.ac.za/handle/10019.1/5139.

[Brummer, 2012] Brummer, N. (2012). LLR Transformation for SRE12. Technical Report December, Agnitio Research, Cape Town, South Africa. Available from: https://sites.google.com/site/bosaristoolkit/sre12/llrTrans.pdf.

[Brummer et al., 2006] Brummer, N., Burget, L., Černocký, J., Glembek, O., Grézl, F., Karafiát, M., Leeuwen, D. V., Matejka, P., Schwarz, P., and Strasheim, A. (2006). Fusion of heterogeneous speaker recognition systems in the STBU submission for the NIST speaker recognition evaluation 2006. *IEEE Transactions on Audio, Speech and Language*

Processing, 15(7):2072–2084. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4291590&tag=1.

[Brummer and De Villiers, 2010] Brummer, N. and De Villiers, E. (2010). The Speaker Partitioning Problem. In *Proceedings of Odyssey 2010 - The Speaker and Language Recognition Workshop*, pages 194–201, Brno, Czech Republic. ISCA. Available from: http://www.isca-speech.org/archive_open/archive_papers/odyssey_2010/papers/od10_034.pdf.

[Brummer and De Villiers, 2011] Brummer, N. and De Villiers, E. (2011). The BOSARIS Toolkit: Theory, Algorithms and Code for Surviving the New DCF. In *NIST SRE11 Speaker Recognition Workshop*, pages 1–23, Atlanta, Georgia, USA. Available from: https://sites.google.com/site/nikobrummer/bosaris_toolkit_full_paper.pdf.

[Brummer and Preez, 2006] Brummer, N. and Preez, J. D. (2006). Application-independent evaluation of speaker detection. *Computer Speech and Language*, 20(2-3):230–275. Available from: http://linkinghub.elsevier.com/retrieve/pii/S0885230805000483.

[Brummer and Strasheim, 2009] Brummer, N. and Strasheim, A. (2009). AGNITIO's Speaker Recognition System for EVALITA 2009. In *Proceedings of the 11th Conference of the Italian Association for Artificial Intelligence*, Reggio Emilia, Italy. Available from: http://www.evalita.it/sites/evalita.fbk.eu/files/proceedings2009/SpeakerIdentityVerification/Application/SIV_APPLICATION_AGNITIO.pdf.

[Burget et al., 2007] Burget, L., Matejka, P., Schwarz, P., Glembek, O., and Černocký, J. (2007). Analysis of feature extraction and channel compensation in GMM speaker recognition system. *IEEE Transactions on Audio, Speech and Language Processing*, 15(7):1979–1986. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4291591&tag=1.

[Burget et al., 2011] Burget, L., Plchot, O., Cumani, S., Glembek, O., Matejka, P., and Brummer, N. (2011). Discriminatively Trained Probabilistic Linear Discriminant Analysis for Speaker Verification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2011*, pages 4832–4835, Prague, Czech Republic. IEEE. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5947437.

[Campbell et al., 2009] Campbell, J. P., Shen, W., Campbell, W. M., Schwartz, R., Bonastre, J.-F., and Matrouf, D. (2009). Forensic speaker recognition. *IEEE Signal Processing Magazine*, 26(2):95–103. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4806209.

[Campbell et al., 2004] Campbell, W. M., Campbell, J. P., Reynolds, D. A., Jones, D., and Leek, T. (2004). Phonetic speaker recognition with support vector machines. In *Advances in Neural Information Processing Systems*, pages 1377–1384. MIT Press. Available from: http://papers.nips.cc/paper/2523-phonetic-speaker-recognition-with-support-vector-machines.pdf.

[Campbell et al., 2005] Campbell, W. M., Reynolds, D. A., Campbell, J. P., and Brady, K. J. (2005). Estimating and evaluating confidence for forensic speaker recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2005*, pages 717–720, Philadelphia, Pennsylvania, USA. IEEE. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.208.199.

[Campbell et al., 2006a] Campbell, W. M., Sturim, D. E., and Reynolds, D. A. (2006a). Support vector machines using GMM supervectors for speaker verification. *IEEE Signal Processing Letters*, 13(5):308–311. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1618704.

[Campbell et al., 2006b] Campbell, W. M., Sturim, D. E., Reynolds, D. A., and Solomonoff, A. (2006b). SVM Based Speaker Verification using a GMM Supervector Kernel and NAP Variability Compensation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2006*, volume 1, pages 97–100, Toulouse, France. IEEE. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1659966.

[Campbell et al., 2007] Campbell, W. M., Sturim, D. E., Shen, W., Reynolds, D. A., and Navratil, J. (2007). The MIT-LL/IBM 2006 Speaker Recognition System: High-Performance Reduced-Complexity Recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2007*, volume 4, pages 217–220, Honolulu, Hawaii, USA. IEEE. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4218076.

[Chakka et al., 2011] Chakka, M. M., Anjos, A., Marcel, S., Tronci, R., Muntoni, D., Fadda, G., Pili, M., Sirena, N., Murgia, G., Ristori, M., Roli, F., Li, S. Z., Schwartz, W. R., Rocha, A., Pedrini, H., Lorenzo-Navarro, J., Castrillon-Santana, M., Maatta, J., Hadid, A., and Pietikainen, M. (2011). Competition on counter measures to 2-D facial spoofing attacks. In *Proceedings of the 2011 International Joint Conference on Biometrics (IJCB)*, pages 1–6, Washington, DC, USA. IEEE. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6117509.

[Chang and Lin, 2011] Chang, C.-C. and Lin, C.-J. (2011). *LIBSVM: a library for support vector machines*. Available from: http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[Chetouani et al., 2009] Chetouani, M., Faundez, M., Gas, B., and Zarader, J. L. (2009). Investigation on LP-residual presentations for speaker identification. *Pattern Recognition*, 42(3):487–494.

[Chingovska et al., 2012] Chingovska, I., Anjos, A., and Marcel, S. (2012). On the Effectiveness of Local Binary Patterns in Face Anti-spoofing. In *Proceedings of the International Conference of the Biometrics Special Interest Group, BIOSIG 2012*, pages 1–7, Darmstadt, Germany. IEEE. Available from: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6313548.

[Chollet et al., 2012] Chollet, G., Perrot, P., Karam, W., Mokbel, C., Kanade, S., and Petrovska-Delacrétaz, D. (2012). Identities, forgeries and disguises. *International Journal* *of Information Technology and Management*, 11(1/2):138–152. Available from: http://www.inderscience.com/link.php?id=44070.

[Cieri et al., 2007] Cieri, C., Corson, L., Graff, D., and Walker, K. (2007). Resources for New Research Directions in Speaker Recognition: The Mixer 3, 4 and 5 Corpora. In *Proceedings of the 8th Annual Conference of the International Speech Communication Association, Interspeech 2007*, pages 950–953, Antwerp, Belgium. ISCA. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.78.7500.

[Clark and Foulkes, 2007] Clark, J. and Foulkes, P. (2007). Identification of voices in electronically disguised speech. *International Journal of Speech Language and the Law*, 14(2):195–221. Available from: http://www.equinoxjournals.com/ojs/index.php/IJSLL/article/view/3820.

[Cristianini and Shawe-Taylor, 2000] Cristianini, N. and Shawe-Taylor, J. (2000). *An introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press. Available from: http://eprints.ecs.soton.ac.uk/9578/.

[Cumani et al., 2011] Cumani, S., Brummer, N., Burget, L., and Laface, P. (2011). Fast discriminative speaker verification in the i-vector space. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2011*, pages 4852–4855, Prague, Czech Republic. IEEE. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5947442.

[Cumani et al., 2012] Cumani, S., Glembek, O., Brummer, N., De Villiers, E., and Laface, P. (2012). Gender independent discriminative speaker recognition in i-vector space. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2012*, pages 4361–4364, Kyoto, Japan. IEEE. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6288885.

[Davis and Mermelstein, 1980] Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics Speech and Signal Processing*, 28(4):357–366. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1163420.

[De Leon et al., 2010a] De Leon, P. L., Apsingekar, V. R., Pucher, M., and Yamagishi, J. (2010a). Revisiting the security of speaker verification systems against imposture using synthetic speech. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2010*, pages 1798–1801, Dallas, TX, USA. IEEE. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5495413.

[De Leon et al., 2011] De Leon, P. L., Hernaez, I., Saratxaga, I., Pucher, M., and Yamagishi, J. (2011). Detection of synthetic speech for the problem of imposture. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2011*, pages 4844–4847, Prague, Czech Republic. IEEE. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5947440.

[De Leon et al., 2010b] De Leon, P. L., Pucher, M., and Yamagishi, J. (2010b). Evaluation of the Vulnerability of Speaker Verification to Synthetic Speech. In *Proceedings of Odyssey 2010 - The Speaker and Language Recognition Workshop*, pages 151–158, Brno, Czech Republic. ISCA. Available from: http://www.isca-speech.org/archive_open/archive_papers/odyssey_2010/papers/od10_028.pdf.

[De Leon et al., 2012a] De Leon, P. L., Pucher, M., Yamagishi, J., Hernaez, I., and Saratxaga, I. (2012a). Evaluation of Speaker Verification Security and Detection of HMM-Based Synthetic Speech. *IEEE Transactions on Audio, Speech and Language Processing*, 20(8):2280–2290. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6205335.

[De Leon et al., 2012b] De Leon, P. L., Stewart, B., and Yamagishi, J. (2012b). Synthetic Speech Discrimination using Pitch Pattern Statistics Derived from Image Analysis. In *Proceedings of the 13th Annual Conference of the International Speech Communication Association, Interspeech 2012*, Portland, Oregon, USA. ISCA. Available from: http://www.ece.nmsu.edu/~pdeleon/Research/Publications/Interspeech_2012_3.pdf.

[Dehak et al., 2009] Dehak, N., Dehak, R., Kenny, P., Brummer, N., Ouellet, P., and Dumouchel, P. (2009). Support Vector Machines versus Fast Scoring in the Low-Dimensional Total Variability Space for Speaker Verification. In *Proceedings of the 10th Annual Conference of the International Speech Communication Association, Interspeech 2009*, pages 1559–1562, Brighton, UK. ISCA. Available from: http://www.isca-speech.org/archive/interspeech_2009/i09_1559.html.

[Dehak et al., 2008] Dehak, N., Dehak, R., Kenny, P., and Dumouchel, P. (2008). Comparison between factor analysis and GMM support vector machines for speaker verification. In *Proceedings of Odyssey 2008 - The Speaker and Language Recognition Workshop*, Stellenbosch, South Africa. ISCA. Available from: http://www.isca-speech.org/archive_open/odyssey_2008/od08_009.html.

[Dehak et al., 2007] Dehak, N., Dumouchel, P., and Kenny, P. (2007). Modeling Prosodic Features With Joint Factor Analysis for Speaker Verification. *IEEE Transactions on Audio, Speech and Language Processing*, 15(7):2095–2103. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4291597.

[Dehak et al., 2011a] Dehak, N., Karam, Z. N., Reynolds, D. A., Campbell, W. M., and Glass, J. R. (2011a). A Channel-Blind System for Speaker Verification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2011*, pages 4536–4539, Prague, Czech Republic. IEEE. Available from: http://groups.csail.mit.edu/sls/publications/2011/Dehak2_ICASSP2011.pdf.

[Dehak et al., 2011b] Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., and Ouellet, P. (2011b). Front-End Factor Analysis For Speaker Verification. *IEEE Transactions on Audio, Speech and Language Processing*, 19(4):788 – 798. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5545402&tag=1.

[Dempster et al., 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal*

Statistical Society B, 39(1):1–22. Available from: http://links.jstor.org/sici?sici=0035-9246(1977)39:1<1:MLFIDV>2.0.CO;2-Z.

[Doddington, 2000] Doddington, G. R. (2000). The NIST speaker recognition evaluation - Overview, methodology, systems, results, perspective. *Speech Communication*, 31(2-3):225–254. Available from: http://dx.doi.org/10.1016/S0167-6393(99)00080-1.

[Doddington, 2001] Doddington, G. R. (2001). Speaker recognition based on idiolectal differences between speakers. In *Proceedings of the 7th European Conference on Speech Comunication and Technology, Interspeech 2001*, pages 2521–2524, Aalborg, Denmark. ISCA. Available from: http://perso.telecom-paristech.fr/~chollet/Biblio/Congres/Audio/Eurospeech01/CDROM/papers/page2521.pdf.

[Doddington, 2012] Doddington, G. R. (2012). The Effect of Target/Non-Target Age Difference on Speaker Recognition Performance. In *Proceedings of Odyssey 2012 - The Speaker and Language Recognition Workshop*, pages 263–267, Singapore. ISCA.

[Dunn et al., 2001] Dunn, R. B., Quatieri, T. F., Reynolds, D. A., and Campbell, J. P. (2001). Speaker recognition from coded speech and the effects of score normalization. In *Proceedings of the 35th Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 1562 – 1567, Pacific Grove, California, USA. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=987749.

[Endres, 1971] Endres, W. (1971). Voice Spectrograms as a Function of Age, Voice Disguise, and Voice Imitation. *The Journal of the Acoustical Society of America*, 49(6B):1842. Available from: http://link.aip.org/link/JASMAN/v49/i6B/p1842/s1&Agg=doi.

[ETSI, 2007] ETSI (2007). ETSI Standard Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Advanced front-end feature extraction algorithm; Compression algorithms. Available from: http://pda.etsi.org/pda/queryform.asp.

[Evans et al., 2013] Evans, N., Kinnunen, T., and Yamagishi, J. (2013). Spoofing and Countermeasures for Automatic Speaker Verification. In *Proceedings of the 14th Annual Conference of the International Speech Communication Association, Interspeech 2013*, pages 925–929, Lyon, France. ISCA.

[Farrus et al., 2008] Farrus, M., Wagner, M., Anguita, J., and Hernando, J. (2008). How vulnerable are prosodic features to professional imitators? In *Proceedings of Odyssey 2008 - The Speaker and Language Recognition Workshop*, Stellenbosch, South Africa. ISCA. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.160.7218.

[Farrús et al., 2010] Farrús, M., Wagner, M., Erro, D., and Hernando, J. (2010). Automatic Speaker Recognition as a Measurement of Voice Imitation and Conversion. *International Journal of Speech Language and the Law*, 17(1):119–142. Available from: http://www.equinoxjournals.com/IJSLL/article/view/6328.

[Feld et al., 2010] Feld, M., Schwartz, T., and Müller, C. (2010). This Is Me: Using Ambient Voice Patterns for In-Car Positioning. In de Ruyter, B., Wichert, R., Keyson, D., Markopoulos, P., Streitz, N., Divitini, M., Georgantas, N., and Mana Gomez, A., editors, *Proceedings of Ambient Intelligence - First International Joint Conference, AmI 2010*, volume 6439 of *Lecture Notes in Computer Science*, pages 290–294. Springer Berlin / Heidelberg, Malaga, Spain. Available from: http://dx.doi.org/10.1007/978-3-642-16917-5_33.

[Ferrer et al., 2011] Ferrer, L., Bratt, H., Burget, L., Černocký, J., Graciarena, M., Lawson, A., Lei, Y., Matejka, P., Plchot, O., and Scheffer, N. (2011). Promoting robustness for speaker modeling in the community: the PRISM evaluation set. In *NIST SRE11 Speaker Recognition Workshop*, Atlanta, Georgia, USA. Available from: www-speech.sri.com/papers/sre11-ferrer.pdf?

[Ferrer et al., 2008] Ferrer, L., Graciarena, M., Zymnis, A., and Shriberg, E. (2008). System Combination Using Auxiliary Information for Speaker Verification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2008*, pages 4853 – 4856, Las Vegas, Nevada, USA. IEEE. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4518744.

[Figueiredo and Britto, 1996] Figueiredo, R. M. D. and Britto, H. D. S. (1996). A report on the acoustic effects of one type of disguise. *Forensic Linguistics*, 3(1):168–175. Available from: https://www.equinoxpub.com/journals/index.php/IJSLL/article/view/17246.

[Furui, 1981] Furui, S. (1981). Cepstral analysis technique for automatic speaker verification. *IEEE Transactions on Acoustics Speech and Signal Processing*, 29(2):254–272. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1163530.

[Furui, 1986] Furui, S. (1986). Speaker-independent isolated word recognition using dynamic features of speech spectrum. *IEEE Transactions on Acoustics Speech and Signal Processing*, 34(1):52–59. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1164788.

[Galou and Chollet, 2011] Galou, G. and Chollet, G. (2011). Synthetic Voice Forgery in the Forensic Context: a short tutorial. In *Forensic Speech and Audio Analysis Working Group (ENFSI-FSAAWG)*, Rome (Italy). Available from: http://biblio.telecom-paristech.fr/cgi-bin/download.cgi?id=11627.

[Garcia-Romero and Espy-Wilson, 2011] Garcia-Romero, D. and Espy-Wilson, C. Y. (2011). Analysis of I-vector Length Normalization in Speaker Recognition Systems. In *Proceedings of the 12th Annual Conference of the International Speech Communication Association, Interspeech 2011*, pages 249–252, Florence, Italy. ISCA. Available from: http://www.isr.umd.edu/Labs/SCL/publications/conference/dgromero_is11%_lnorm_final.pdf.

[Garcia-Romero et al., 2004] Garcia-Romero, D., Fierrez-Aguilar, J., Gonzalez-Rodriguez, J., and Ortega-García, J. (2004). On the use of quality measures for text-independent speaker recognition. In *Proceedings of Odyssey 2004 - The Speaker and Language Recognition Workshop*, Toledo, Spain. ISCA. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.60.5529&amp;rep=rep1&amp;type=pdf.

[Garcia-Romero et al., 2006] Garcia-Romero, D., Fierrez-Aguilar, J., Gonzalez-Rodriguez, J., and Ortega-García, J. (2006). Using quality measures for multilevel speaker recognition. *Computer Speech and Language*, 20(2-3):192–209. Available from: http://linkinghub.elsevier.com/retrieve/pii/S0885230805000537.

[Garcia-Romero et al., 2012] Garcia-Romero, D., Zhou, X., and Espy-Wilson, C. Y. (2012). Multicondition Training of Gaussian PLDA Models in i-Vector Space for Noise and Reverberation Robust Speaker Recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2012*, pages 4257–4260, Kyoto, Japan. IEEE. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6288859.

[Gauvain and Lee, 1994] Gauvain, J.-L. and Lee, C.-h. (1994). Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions On Speech And Audio Processing*, 2(2):291–298. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=279278.

[Gheorghe and Anton, 2008] Gheorghe, R. and Anton, G. (2008). Recursive Calculation of Mel-Cepstrum from LP Coefficients. Technical report. Available from: http://zeus.eed.usv.ro/SistemeDistribuite/2008/17RaduAnton.pdf.

[Gish and Schmidt, 1994] Gish, H. and Schmidt, M. (1994). Text-independent speaker identification. *IEEE Signal Processing Magazine*, 11(4):18–32. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=317924.

[Glembek et al., 2009] Glembek, O., Burget, L., Dehak, N., Brummer, N., and Kenny, P. (2009). Comparison of scoring methods used in speaker recognition with Joint Factor Analysis. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2009*, pages 4057–4060, Taipei, Taiwan. IEEE. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4960519.

[Gonzalez Hautamaki et al., 2013] Gonzalez Hautamaki, R., Kinnunen, T., Hautamaki, V., Leino, T., and Laukkanen, A.-m. (2013). I-vectors meet imitators: on vulnerability of speaker verification systems against voice mimicry. In *Proceedings of the 14th Annual Conference of the International Speech Communication Association, Interspeech 2013*, pages 930–934, Lyon, France. ISCA. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.310.2259.

[Gonzalez-Rodriguez et al., 2003] Gonzalez-Rodriguez, J., García-Gomar, M., Ramos, D., and Ortega-García, J. (2003). Robust likelihood ratio estimation in Bayesian forensic speaker recognition. In *Proceedings of the 8th European Conference on Speech Communication and Technology, Eurospeech 2003*, pages 693–696, Geneva, Switzerland. ISCA.

[Gonzalez-Rodriguez et al., 2007] Gonzalez-Rodriguez, J., Rose, P., Ramos, D., Toledano, D. T., and Ortega-García, J. (2007). Emulating DNA: Rigorous Quantification of Evidential Weight in Transparent and Testable Forensic Speaker Recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 15(7):2104–2115. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4291601.

[Grother and Tabassi, 2007] Grother, P. and Tabassi, E. (2007). Performance of biometric quality measures. *IEEE transactions on pattern analysis and machine intelligence*, 29(4):531–43. Available from: http://www.ncbi.nlm.nih.gov/pubmed/17299212.

[Gupta et al., 2005] Gupta, H., Hautamaki, V., Kinnunen, T., and Franti, P. (2005). Field Evaluation of Text-Dependent Speaker Recognition in an Access Control Application. In *Proceedings of the 10th International Conference Speech and Computer SPECOM 2005*, Patras, Greece. University of Patras. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.144.4678&amp;rep=rep1&amp;type=pdf.

[Harriero et al., 2009] Harriero, A., Ramos, D., Gonzalez-Rodriguez, J., and Fierrez-Aguilar, J. (2009). Analysis of the Utility of Classical and Novel Speech Quality Measures for Speaker Verification. In Tistarelli, M. and Nixon, M., editors, *Proceedings of the Third International Conference on Advances in Biometrics, ICB 2009*, volume 5558 of *Lecture Notes in Computer Science*, pages 434–442. Springer-Verlag Berlin, Heidelberg, Alghero, Italy. Available from: http://dx.doi.org/10.1007/978-3-642-01793-3_45.

[Hautamaki et al., 2011] Hautamaki, V., Lee, K., Kinnunen, T., Ma, B., and Li, H. (2011). Regularized logistic regression fusion for speaker verification. In *Proceedings of the 12th Annual Conference of the International Speech Communication Association, Interspeech 2011*, pages 2745–2748, Florence, Italy. ISCA. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.221.2350.

[Hautamaki et al., 2012] Hautamaki, V., Lee, K., and Larcher, A. (2012). Variational Bayes logistic regression as regularized fusion for NIST SRE 2010. In *Proceedings of Odyssey 2012 - The Speaker and Language Recognition Workshop*, Singapore. ISCA. Available from: http://www.cs.joensuu.fi/pages/tkinnu/webpage/pdf/regularized-fusion-Odyssey.pdf.

[Heck and Weintraub, 1997] Heck, L. P. and Weintraub, M. (1997). Handset-Dependent Background Models for Robust Text-Independent Speaker Recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 1997*, volume 2, pages 1071–1074, Munich, Bavaria, Germany. IEEE. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=596126.

[Hermansky, 1990] Hermansky, H. (1990). Perceptual linear predictive (PLP) analysis of speech. *Journal of the Acoustical Society of America*, 87(4):1738–1752. Available from: http://lectures.idiap.ch/winter2005-2006/ic-48/courseslide/PLP.pdf.

[Hermansky and Morgan, 1994] Hermansky, H. and Morgan, N. (1994). RASTA processing of speech. *IEEE Transactions On Speech And Audio Processing*, 2(4):578–589. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=326616.

[Higgins and Wohlford, 1986] Higgins, A. and Wohlford, R. (1986). A new method of text-independent speaker recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 1986*, volume 11, pages 869–872, Tokyo, Japan. IEEE. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1168975.

[Hirsch, 2005] Hirsch, H.-g. (2005). FaNT - Filtering and Noise Adding Tool. Available from: http://dnt.kr.hsnr.de/download.html.

[Hirsch and Pearce, 2000] Hirsch, H.-g. and Pearce, D. (2000). The Aurora Experimental Framework for the Performance Evaluation of Speech Recognition Systems under Noisy Conditions. In *Proceedings of the 6th International Conference on Spoken Language Processing, ICSLP - Interspeech 2000*, pages 16–19, Beijing, China. ISCA. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.121.1762&amp;rep=rep1&amp;type=pdf.

[Hirson and Duckworth, 1993] Hirson, A. and Duckworth, M. (1993). Glottal fry and voice disguise: a case study in forensic phonetics. *Journal of Biomedical Engineering*, 15(3):193–200. Available from: http://dx.doi.org/10.1016/0141-5425(93)90115-F.

[Hollien and Majewski, 1977] Hollien, H. and Majewski, W. (1977). Speaker identification by long-term spectra under normal and distorted speech conditions. *The Journal of the Acoustical Society of America*, 62(4):975–980. Available from: http://link.aip.org/link/JASMAN/v62/i4/p975/s1&Agg=doi.

[Hotelling, 1953] Hotelling, H. (1953). New Light on the Correlation Coefficient and its Transforms. *Journal of the Royal Statistical Society. Series B (Methodological)*, 15(2):193–232. Available from: http://www.jstor.org/stable/2983768.

[Huang et al., 2001] Huang, X., Acero, A., and Hon, H.-W. (2001). *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. Prentice Hall PTR. Available from: http://www.worldcat.org/isbn/0130226165.

[ITU-T, 2004] ITU-T (2004). Recommendation P.563: Single ended method for objective speech quality assessment in narrow-band telephony applications. Technical report, International Telecommunications Union. Available from: http://www.itu.int/rec/T-REC-P.563/en.

[Kain and Macon, 1998] Kain, A. and Macon, M. W. (1998). Spectral Voice Conversion For Text-To-Speech Synthesis. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 1998*, volume 1, pages 285–288, Seattle, Washington, USA. IEEE. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=674423.

[Kajarekar et al., 2006] Kajarekar, S. S., Bratt, H., Shriberg, E., and de Leon, R. (2006). A Study of Intentional Voice Modifications for Evading Automatic Speaker Recognition. In *Proceedings of the IEEE Odyssey 2006 - The Speaker and Language Recognition Workshop*, pages 1–6, San Juan, Puerto Rico. IEEE. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4013540.

[Kalinli et al., 2009] Kalinli, O., Seltzer, M. L., and Acero, A. (2009). Noise adaptive training using a vector taylor series approach for noise robust automatic speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2009*, pages 3825–3828, Taipei, Taiwan. IEEE. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4960461.

[Karam et al., 2009] Karam, W., Bredin, H., Greige, H., Chollet, G., and Mokbel, C. (2009). Talking-Face Identity Verification, Audiovisual Forgery, and Robustness Issues. *EURASIP Journal on Advances in Signal Processing*, 2009:1–15. Available from: http://asp.eurasipjournals.com/content/2009/1/746481.

[Kelly and Harte, 2011] Kelly, F. and Harte, N. (2011). Effects of Long-Term Ageing on Speaker Verification. In *Biometrics and ID Management, Proceedings of the COST 2101 European Workshop, BioID 2011*, volume 6583 of *Lecture Notes in Computer Science*, pages 113–124, Brandenburg, Germany. Springer Berlin Heidelberg. Available from: http://link.springer.com/chapter/10.1007/978-3-642-19530-3_11.

[Kenny, 2005] Kenny, P. (2005). Joint factor analysis of speaker and session variability : Theory and algorithms - Technical report CRIM-06/08-13. Technical report, CRIM, Montreal. Available from: http://www.crim.ca/perso/patrick.kenny/FAtheory.pdf.

[Kenny, 2010] Kenny, P. (2010). Bayesian Speaker Verification with Heavy-Tailed Priors. In *Proceedings of Odyssey 2010 - The Speaker and Language Recognition Workshop*, Brno, Czech Republic. ISCA. Available from: http://www.crim.ca/perso/patrick.kenny/kenny_Odyssey2010.pdf.

[Kenny et al., 2007a] Kenny, P., Boulianne, G., Ouellet, P., and Dumouchel, P. (2007a). Joint Factor Analysis versus Eigenchannels in Speaker Recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 15(4):1435–1447. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4156202.

[Kenny et al., 2007b] Kenny, P., Boulianne, G., Ouellet, P., and Dumouchel, P. (2007b). Speaker and session variability in GMM-based speaker verification. *IEEE Transactions on Audio, Speech and Language Processing*, 15(4):1448–1460. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4156203.

[Kenny et al., 2008] Kenny, P., Ouellet, P., Dehak, N., Gupta, V., and Dumouchel, P. (2008). A Study of Interspeaker Variability in Speaker Verification. *IEEE Transactions on Audio, Speech and Language Processing*, 16(5):980–988. Available from: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4531370.

[Kim, 2014] Kim, U. (2014). Ultra Voice Changer. Available from: http://play.google.com.

[Kinnunen, 2006] Kinnunen, T. (2006). Joint acoustic-modulation frequency for speaker recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2006*, pages 665–668, Toulouse, France. IEEE. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1660108&tag=1.

[Kinnunen and Alku, 2009] Kinnunen, T. and Alku, P. (2009). On separating glottal source and vocal tract information in telephony speaker verification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2009*, pages 4545–4548, Taipei, Taiwan. IEEE. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4960641.

[Kinnunen et al., 2008] Kinnunen, T., Lee, K. A., and Li, H. (2008). Dimension reduction of the modulation spectrogram for speaker verification. In *Proceedings of Odyssey 2008 - The Speaker and Language Recognition Workshop*, Stellenbosch, South Africa. ISCA.

[Kinnunen and Li, 2010] Kinnunen, T. and Li, H. (2010). An overview of text-independent speaker recognition: From features to supervectors. *Speech Communication*, 52(1):12–40. Available from: http://linkinghub.elsevier.com/retrieve/pii/S0167639309001289.

[Kinnunen et al., 2006] Kinnunen, T., Wei, C., Koh, E., Wang, L., Li, H., and Chng, E. S. (2006). Temporal discrete cosine transform: Towards longer term temporal features for speaker verification. In *Proceedings of the 5th International Symposium on Chinese Spoken Language Processing, ISCSLP 2006*, pages 547–558, Singapore. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.144.4023.

[Kinnunen et al., 2012] Kinnunen, T., Wu, Z.-z., Lee, K. A., Sedlak, F., Chng, E. S., and Li, H. (2012). Vulnerability of Speaker Verification Systems Against Voice Conversion Spoofing Attacks: the Case of Telephone Speech. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2012*, pages 4401–4404, Kyoto, Japan. IEEE. Available from: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6288895.

[Kockmann et al., 2011] Kockmann, M., Ferrer, L., Burget, L., Shriberg, E., and Černocký, J. (2011). Recent Progress in Prosodic Speaker Verification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2011*, pages 4556–4559, Prague, Czech Republic. IEEE. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5947368.

[Kons and Aronowitz, 2013] Kons, Z. and Aronowitz, H. (2013). Voice Transformation-Based Spoofing of Text-Dependent Speaker Verification Systems. In *Proceedings of the 14th Annual Conference of the International Speech Communication Association, Interspeech 2013*, pages 945–949, Lyon, France. ISCA.

[Koolwaaij et al., 2000] Koolwaaij, J., Boves, L., Jongebloed, H., and den Os, E. (2000). On model quality and evaluation in speaker verification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2000*, volume 6, pages 3759–3762, Istanbul, Turkey. IEEE. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=860220.

[Kryszczuk et al., 2007] Kryszczuk, K., Richiardi, J., Prodanov, P., and Drygajlo, A. (2007). Reliability-Based Decision Fusion in Multimodal Biometric Verification Systems. *EURASIP Journal on Advances in Signal Processing*, 2007:1–10. Available from: http://www.hindawi.com/journals/asp/2007/086572.abs.html.

[Künzel, 2000] Künzel, H. J. (2000). Effects of voice disguise on speaking fundamental frequency. *International Journal of Speech Language and the Law*, 7(2):150 – 179. Available from: http://www.equinoxjournals.com/IJSLL/article/viewArticle/595.

[Künzel et al., 2004] Künzel, H. J., Gonzalez-Rodriguez, J., and Ortega-García, J. (2004). Effect of Voice Disguise on the Performance of a Forensic Automatic Speaker Recognition System. In *Proceedings of Odyssey 2004 - The Speaker and Language Recognition Workshop*, pages 153–156, Toledo, Spain. ISCA. Available from: http://www.isca-speech.org/archive_open/odyssey_04/ody4_153.html.

[Larcher et al., 2012] Larcher, A., Lee, K. A., Ma, B., and Li, H. (2012). The RSR2015: Database for Text-Dependent Speaker Verification using Multiple Pass-Phrases. In *Proceedings of the 13th Annual Conference of the International Speech Communication Association, Interspeech 2012*, Portland, Oregon, USA. ISCA. Available from: http://www1.i2r.a-star.edu.sg/~kalee/interspeech2012_rsr2015.pdf.

[Lau et al., 2005] Lau, Y. W., Tran, D., and Wagner, M. (2005). Testing Voice Mimicry with the YOHO Speaker Verification Corpus. In *Proceedings of the 9th International Conference on Knowledge-Based Intelligent Information And Engineering Systems, KES 2005*, volume 3584 of *Lecture Notes in Computer Science*, pages 15–21, Melbourne, Australia. Springer-Verlag Berlin Heidelberg. Available from: http://link.springer.com/chapter/10.1007/11554028_3.

[Lau et al., 2004] Lau, Y. W., Wagner, M., and Tran, D. (2004). Vulnerability of speaker verification to voice mimicking. In *Proceedings of the International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004.*, pages 145–148, Hong Kong. IEEE. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1434021.

[Lei et al., 2012] Lei, Y., Burget, L., Ferrer, L., Graciarena, M., and Scheffer, N. (2012). Towards Noise-Robust Speaker Recognition Using Probabilistic Linear Discriminant Analysis. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2012*, pages 4253–4256, Kyoto, Japan. IEEE. Available from: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=6288858.

[Lei et al., 2013] Lei, Y., Burget, L., and Scheffer, N. (2013). A Noise Robust i-Vector Extractor Using Vector Taylor Series for Speaker Recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013*, pages 6788–6791, Vancouver, British Columbia, Canada. IEEE. Available from: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6638976.

[Lei and Hansen, 2009] Lei, Y. and Hansen, J. H. L. (2009). The Role of Age in Factor Analysis for Speaker Identification. In *Proceedings of the 10th Annual Conference of the International Speech Communication Association, Interspeech 2009*, pages 2371–2374, Brighton, UK. ISCA.

[Lei et al., 2014a] Lei, Y., McLaren, M., Ferrer, L., and Scheffer, N. (2014a). Simplified VTS-Based i-Vector Extraction in Noise-Robust Speaker Recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014*, pages 4065–4069, Florence, Italy. IEEE. Available from: http://www.sri.com/work/publications/simplified-vts-based-i-vector-extraction-noise-robust-speaker-recognition.

[Lei et al., 2014b] Lei, Y., Scheffer, N., Ferrer, L., and McLaren, M. (2014b). A Novel Scheme for Speaker Recognition Using a Phonetically-Aware Deep Neural Network. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014*, pages 1714–1718, Florence, Italy. IEEE. Available from: http://www.sri.com/work/publications/novel-scheme-speaker-recognition-using-phonetically-aware-deep-neural-network.

[Li et al., 2005] Li, D., Yang, Y., Wu, Z., and Wu, T. (2005). Emotion-State Conversion for Speaker Recognition. In Tao, J., Tan, T., and Picard, R., editors, *Affective Computing and Intelligent Interaction, Proceedings of the First International Conference, ACII 2005*, volume 3784 of *Lecture Notes in Computer Science*, pages 403–410, Beijing, China. Springer-Verlag Berlin, Heidelberg. Available from: http://dx.doi.org/10.1007/11573548_52.

[Li et al., 2007] Li, J., Deng, L., Yu, D., Gong, Y., and Acero, A. (2007). High-performance HMM adaptation with joint compensation of additive and convolutive distortions via vector Taylor series. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2007*, Kioto, Japan. IEEE. Available from: http://research.microsoft.com/apps/pubs/default.aspx?id=78299.

[Li et al., 2009] Li, J., Deng, L., Yu, D., Gong, Y., and Acero, A. (2009). A unified framework of HMM adaptation with joint compensation of additive and convolutive distortions. *Computer Speech and Language*, 23:389–405. Available from: http://research.microsoft.com/apps/pubs/default.aspx?id=80028.

[Li and Porter, 1988] Li, K. P. and Porter, J. E. (1988). Normalizations and selection of speech segments for speaker recognition scoring. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 1988*, volume 1, pages 595–598, New York, NY, USA. IEEE. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=196655.

[Lindberg and Blomberg, 1999] Lindberg, J. and Blomberg, M. (1999). Vulnerability in speaker verification a study of technical impostor techniques. In *Proceedings of the 6th European Conference on Speech Communication and Technology, Eurospeech 1999*, pages 1211–1214, Budapest, Hungary. ISCA. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.12.9603.

[Lu et al., 2009] Lu, L., Dong, Y., Zhao, X., Liu, J., and Wang, H. (2009). The effect of language factors for robust speaker recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2009*, pages 4217–4220, Taipei, Taiwan. IEEE. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4960559.

[Lugger and Yang, 2006] Lugger, M. and Yang, B. (2006). Classification of different speaking groups by means of voice quality parameters. In *ITG-Fachtagung Sprach-Kommunikation*, Kiel, Germany. Available from: http://www.iss.uni-stuttgart.de/forschung/veroeffentlichungen/lugger_itg2006.pdf.

[Lummis and Rosenberg, 1972] Lummis, R. C. and Rosenberg, A. E. (1972). Test of an Automatic Speaker Verification Method with Intensively Trained Professional Mimics. *The Journal of the Acoustical Society of America*, 51(1A):131–132. Available from: http://link.aip.org/link/?JAS/51/131/5.

[Ma et al., 2006] Ma, B., Zhu, D., Tong, R., and Li, H. (2006). Speaker cluster based GMM tokenization for speaker recognition. In *Proceedings of the 9th International Conference on Spoken Language Processing, ICSLP - Interspeech 2006*, pages 505–508, Pittsburgh, Pennsylvania, USA. ISCA. Available from: http://www.ntu.edu.sg/home/aseschng/SpeechTechWeb/members/TongRongWeb/papers/Interspeech2006_mabin.pdf.

[Mak and Yu, 2010] Mak, M.-W. and Yu, H.-B. (2010). Robust voice activity detection for interview speech in NIST speaker recognition evaluation. In *Proceedings of the APSIPA ASC*, Singapore. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.187.3727&rep=rep1&type=pdf.

[Marchetto et al., 2009] Marchetto, E., Avanzini, F., and Flego, F. (2009). An Automatic Speaker Recognition System for Intelligence Applications. In *Proceedings of the 17th European Signal Processing Conference, EUSIPCO 2009*, pages 1612–1616, Glasgow, Scotland. Curran Associates, Inc. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.184.5093.

[Mariethoz and Bengio, 2006] Mariethoz, J. and Bengio, S. (2006). Can a Professional Imitator Fool a GMM-Based Speaker Verification System? Technical report, IDIAP Resarch Institute, Martigny, Switzerland. Available from: http://publications.idiap.ch/index.php/publications/show/356.

[Martin et al., 1997] Martin, A. F., Doddington, G. R., Kamm, T., Ordowski, M., and Przybocki, M. A. (1997). The DET curve in assessment of detection task performance. In *Proceedings of the 5th European Conference on Speech Communication and Technology, Eurospeech 1997*, volume 4, pages 1895–1898, Rhodes, Greece. ISCA, ISCA. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.117.4489&amp;rep=rep1&amp;type=pdf.

[Martin and Przybocki, 2001] Martin, A. F. and Przybocki, M. A. (2001). The NIST speaker recognition evaluations: 1996-2001. In *Proceedings of Odyssey 2001 - The Speaker and Language Recognition Workshop*, pages 225–254, Crete, Greece. ISCA. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.61.2552&amp;rep=rep1&amp;type=pdf.

[Martínez et al., 2014] Martínez, D., Burget, L., Stafylakis, T., Lei, Y., Kenny, P., and Lleida, E. (2014). Unscented Transform for Ivector-Based Noisy Speaker Recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal*

Processing, ICASSP 2014, pages 4070–4074, Florence, Italy. IEEE. Available from: www.crim.ca/perso/patrick.kenny/Martinez_ICASSP2014.pdf.

[Masthoff, 1996] Masthoff, H. (1996). A report on a voice disguise experiment. *International Journal of Speech Language and the Law*, 3(1):160–167.

[Masuko et al., 1999] Masuko, T., Hitotsumatsu, T., Tokuda, K., and Kobayashi, T. (1999). On the security of hmm-based speaker verification systems against imposture using synthetic speech. In *Proceedings of the 6th European Conference on Speech Communication and Technology, Eurospeech 1999*, pages 1223–1226, Budapest, Hungary. ISCA. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.61.2528.

[Masuko et al., 2000] Masuko, T., Tokuda, K., and Kobayashi, T. (2000). Imposture using synthetic speech against speaker verification based on spectrum and pitch. In *Proceedings of the 6th International Conference on Spoken Language Processing, ICSLP - Interspeech 2000*, volume 2, pages 302–305, Beijing, China. ISCA. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.28.8230.

[Masuko et al., 1996] Masuko, T., Tokuda, K., Kobayashi, T., and Imai, S. (1996). Speech synthesis using hmms with dynamic features. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 1996*, volume 1, pages 389–392, Atlanta, Georgia, USA. IEEE. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=541114&tag=1.

[Masuko et al., 1997] Masuko, T., Tokuda, K., Kobayashi, T., and Imai, S. (1997). Voice characteristics conversion for HMM-based speech synthesis system. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 1997*, volume 3, pages 1611–1614, Munich, Bavaria, Germany. IEEE. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=598807.

[Matejka et al., 2011] Matejka, P., Glembek, O., Castaldo, F., Alam, M. J., Kenny, P., Burget, L., and Černocký, J. (2011). Full-Covariance UBM and Heavy-Tailed PLDA in I-Vector Speaker Verification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2011*, pages 4828 – 4831, Prague, Czech Republic. IEEE. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5947436&tag=1.

[Matrouf et al., 2006] Matrouf, D., Bonastre, J.-F., and Fredouille, C. (2006). Effect of Speech Transformation on Impostor Acceptance. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2006*, volume 1, pages 933–936, Toulouse, France. IEEE. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1660175.

[McCool et al., 2012] McCool, C., Marcel, S., Hadid, A., Pietikainen, M., Matejka, P., Cernock, J., Poh, N., Kittler, J., Larcher, A., Levy, C., Matrouf, D., Bonastre, J.-F., Tresadern, P., and Cootes, T. (2012). Bi-Modal Person Recognition on a Mobile Phone: Using Mobile Phone Data. In *IEEE ICME Workshop on Hot Topics in Mobile Multimedia*, pages 635–640, Melbourne, Australia. IEEE. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6266494.

[McGlone et al., 1977] McGlone, R. E., Hollien, P., and Hollien, H. (1977). Acoustic Analysis of Voice Disguise Related to Voice Identification. In *Proceedings of the International Conference on Crime Countermeasures*, pages 31–35.

[McGovern, 2004] McGovern, S. (2004). A Model for Room Acoustics. Available from: http://sgm-audio.com/research/rir/rir.html.

[McLaren and Leeuwen, 2011a] McLaren, M. and Leeuwen, D. V. (2011a). Source-normalised-and-weighted LDA for robust speaker recognition using i-vectors. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2011*, pages 5456–5459, Prague, Czech Republic. Centre for Language and Speech Technology, Radboud University Nijmegen, The Netherlands, IEEE. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5947593.

[McLaren and Leeuwen, 2011b] McLaren, M. and Leeuwen, D. V. (2011b). To Weight or not to Weight: Source-Normalised LDA for Speaker Recognition using i-vectors. In *Proceedings of the 12th Annual Conference of the International Speech Communication Association, Interspeech 2011*, pages 2709–2712, Florence, Italy. ISCA. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.228.3576.

[McLaren and Leeuwen, 2012] McLaren, M. and Leeuwen, D. V. (2012). Source-normalised LDA for robust speaker recognition using i-vectors from multiple speech sources. *IEEE Transactions on Audio, Speech and Language Processing*, 20(3):755–766. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5983477.

[Mengusoglu, 2004] Mengusoglu, E. (2004). *Confidence Measures for Speech/Speaker Recognition and Applications on Turkish LVCSR*. PhD thesis, Faculte Polytechnique de Mons. Available from: http://theses.eurasip.org/theses/292/confidence-measures-for-speech-speaker/.

[Minka, 1998] Minka, T. (1998). Inferring a Gaussian distribution. Technical report, MIT media Lab. Available from: http://research.microsoft.com/en-us/um/people/minka/papers/gaussian.html.

[Minka, 2000] Minka, T. (2000). Old and New Matrix Algebra Useful for Statistics. Technical report, MIT Media Lab. Available from: http://research.microsoft.com/en-us/um/people/minka/papers/matrix/.

[Monzo et al., 2007] Monzo, C., Alías, F., Iriondo, I., Gonzalvo, X., and Planet, S. (2007). Discriminating expressive speech styles by voice quality parameterization. In *Proceedings of the 16th International Congress of Phonetic Sciences, ICPhS 2007*, pages 2081–2084, Saarbrücken, Germany. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.177.8950.

[Monzo et al., 2008] Monzo, C., Iriondo, I., and Martinez, E. (2008). Procedimiento para la medida y la modificación del jitter y del shimmer aplicado a la síntesis del habla expresiva. In *Proceedings of the V Jornadas en Tecnologias del Habla*, pages 58–61, Bilbao, Spain.

[Moosmüller, 1997] Moosmüller, S. (1997). Phonological variation in speaker identification. *International Journal of Speech Language and the Law*, 4(1):29–47. Available from: http://www.equinoxpub.com/journals/index.php/IJSLL/article/viewArticle/17267.

[Moosmüller, 2001] Moosmüller, S. (2001). The influence of creaky voice on formant frequency changes. *International Journal of Speech Language and the Law*, 8(1):100–112. Available from: http://www.equinoxjournals.com/IJSLL/article/view/1693.

[Moreno et al., 1993] Moreno, A., Poch, D., Bonafonte, A., Lleida, E., Llisterri, J., Mariño, J., and Nadeu, C. (1993). Albayzin Speech Database: Design of the Phonetic Corpus. In *Proceedings of the 3rd European Conference on Speech Communications and Technology, Eurospeech 1993*, volume 1, pages 175–178, Berlin, Germany. ISCA. Available from: http://liceu.uab.cat/~joaquim/publicacions/Moreno_et_al_93_Albayzin_Phonetic_Corpus.pdf.

[Moreno et al., 1996] Moreno, P. J., Raj, B., and Stern, R. M. (1996). A vector Taylor series approach for environment-independent speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 1996*, volume II, pages 733–736, Atlanta, Georgia, USA. IEEE. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=543225.

[Murty and Yegnanarayana, 2006] Murty, K. and Yegnanarayana, B. (2006). Combining evidence from residual phase and MFCC features for speaker recognition. *IEEE Signal Processing Letters*, 13(1):52–55. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1561210.

[Nakasone and Beck, 2001] Nakasone, H. and Beck, S. D. (2001). Forensic automatic speaker recognition. In *Proceedings of Odyssey 2001 - The Speaker and Language Recognition Workshop*, pages 1–6, Crete, Greece. ISCA. Available from: http://www.isca-speech.org/archive_open/archive_papers/odyssey/odys_139.pdf.

[Niemi-Laitinen et al., 2005] Niemi-Laitinen, T., Saastamoinen, J., Kinnunen, T., and Franti, P. (2005). Applying MFCC-based automatic speaker recognition to GSM and forensic data. In *Proceedings of the Second Baltic Conference on Human Language Technologies (HLT2005)*, pages 317–322, Tallin, Estonia. Available from: http://cs.joensuu.fi/pages/tkinnu/webpage/pdf/HLT2005-forensic.pdf.

[NIST Speech Group, 2006] NIST Speech Group (2006). The NIST Year 2006 Speaker Recognition Evaluation Plan. Technical report, NIST. Available from: http://www.itl.nist.gov/iad/mig/tests/sre/2006/sre-06_evalplan-v9.pdf.

[NIST Speech Group, 2008] NIST Speech Group (2008). The NIST Year 2008 Speaker Recognition Evaluation Plan 1. Technical report, NIST. Available from: http://www.itl.nist.gov/iad/mig/tests/sre/2008/sre08_evalplan_release4.pdf.

[NIST Speech Group, 2010] NIST Speech Group (2010). The NIST Year 2010 Speaker Recognition Evaluation Plan. Technical report, NIST. Available from: http://www.itl.nist.gov/iad/mig/tests/sre/2010/NIST_SRE10_evalplan.r6.pdf.

[NIST Speech Group, 2012] NIST Speech Group (2012). The NIST Year 2012 Speaker Recognition Evaluation Plan. Technical report, NIST. Available from: http://www.nist.gov/itl/iad/mig/upload/NIST_SRE12_evalplan-v17-r1.pdf.

[Ogihara et al., 2005] Ogihara, A., Unno, H., and Shiozaki, A. (2005). Discrimination Method of Synthetic Speech Using Pitch Frequency against Synthetic Speech Falsification. *IEICE transactions on fundamentals of electronics, communications and computer*, 88(1):280–286.

[Oppenheim et al., 1999] Oppenheim, A. V., Schafer, R. W., and Buck, J. R. (1999). *Discrete-Time Signal Processing*. Prentice Hall, Inc.

[Orchard and Yarmey, 1995] Orchard, T. L. and Yarmey, A. D. (1995). The effects of whispers, voice-sample duration, and voice distinctiveness in criminal speaker identification. *Applied cognitive psychology*, 9(3):249–260. Available from: http://cat.inist.fr/?aModele=afficheN&cpsidt=3562962.

[Pelecanos and Sridharan, 2001] Pelecanos, J. and Sridharan, S. (2001). Feature warping for robust speaker verification. In *Proceedings of Odyssey 2001 - The Speaker and Language Recognition Workshop*, Crete, Greece. ISCA. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.155.4456.

[Pellom and Hansen, 1999] Pellom, B. and Hansen, J. H. L. (1999). An Experimental Study of Speaker Verification Sensitivity to Computer Voice-Altered Imposters. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 1999*, volume 2, pages 837–840, Phoenix, Arizona, USA. IEEE. Available from: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=759801.

[Perrot et al., 2005] Perrot, P., Aversano, G., Blouet, R., Charbit, M., and Chollet, G. (2005). Voice Forgery Using ALISP: Indexation in a Client Memory. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2005*, pages 17–20, Philadelphia, Pennsylvania, USA. IEEE. Available from: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1415039.

[Perrot et al., 2007] Perrot, P., Aversano, G., and Chollet, G. (2007). Voice disguise and automatic detection: review and perspectives. In Stylianou, Y., Faundez, M., and Esposito, A., editors, *Progress in Nonlinear Speech Processing*, pages 101–117. Springer-Verlag Berlin, Heidelberg. Available from: http://portal.acm.org/citation.cfm?id=1768226.1768233.

[Perrot and Chollet, 2008] Perrot, P. and Chollet, G. (2008). The question of disguised voice. *The Journal of the Acoustical Society of America*, 123(5):3878. Available from: http://link.aip.org/link/JASMAN/v123/i5/p3878/s1&Agg=doi.

[Perrot et al., 2009] Perrot, P., Morel, M., Razik, J., and Chollet, G. (2009). Vocal Forgery in Forensic Sciences. In *Proceedings of the Second International Conference Forensics in Telecommunications, Information and Multimedia, e-Forensics 2009*, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pages 179–185, Adelaide, Australia. Springer Berlin Heidelberg. Available from: http://link.springer.com/chapter/10.1007/978-3-642-02312-5_21.

[Pfister and Beutler, 2003] Pfister, B. and Beutler, R. (2003). Estimating the weight of evidence in forensic speaker verification. In *Proceedings of the 8th European Conference on Speech Communication and Technology, Eurospeech 2003*, pages 701–704, Geneva, Switzerland. ISCA. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.10.5693.

[Poh and Bengio, 2005] Poh, N. and Bengio, S. (2005). Improving Fusion with Margin-Derived Confidence In Biometric Authentication Tasks. In *Proceedings of the 5th International Conference on Audio- and Video-Based Biometric Person Authentication, AVBPA 2005*, volume 3546, pages 347–356, Hilton Rye Town, NY, USA. Springer. Available from: http://eprints.pascal-network.org/archive/00000865/.

[Prasanna et al., 2006] Prasanna, S. M., Gupta, C. S., and Yegnanarayana, B. (2006). Extraction of speaker-specific excitation information from linear prediction residual of speech. *Speech Communication*, 48(10):1243–1261. Available from: http://www.sciencedirect.com/science/article/pii/S0167639306000665.

[Prince and Elder, 2007] Prince, S. J. and Elder, J. H. (2007). Probabilistic Linear Discriminant Analysis for Inferences About Identity. In *Proceedings of the IEEE International Conference on Computer Vision, ICCV 2007*, pages 1–8, Rio de Janeiro, Brazil. IEEE. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4409052.

[Przybocki et al., 2007] Przybocki, M. A., Martin, A. F., and Le, A. N. (2007). NIST Speaker Recognition Evaluations Utilizing the Mixer Corpora - 2004, 2005, 2006. *IEEE Transactions on Audio, Speech and Language Processing*, 15(7):1951–1959. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4291612.

[Ramirez et al., 2004] Ramirez, J., Segura, J., Benitez, C., Torre, A. D. L., and Rubio, A. (2004). Voice activity detection with noise reduction and long-term spectral divergence estimation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2004*, volume 2, pages 1093–1096, Montreal, Quebec, Canada. IEEE. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1326452.

[Ramos et al., 2005] Ramos, D., Garcia-Romero, D., López, I., and Gonzalez-Rodriguez, J. (2005). Speaker Verification Using Fast Adaptive Tnorm Based on Kullback-Leibler Divergence. In *Proceedings of the Third COST 275 Workshop, Biometrics on the Internet*, pages 49–52, Hertfordshire, Hatfield, UK. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.99.3211&rep=rep1&type=pdf#page=71.

[Ramos et al., 2008] Ramos, D., Gonzalez-Rodriguez, J., Gonzalez-Dominguez, J., and Lucena-Molina, J. J. (2008). Addressing database mismatch in forensic speaker recognition with Ahumada III: a public real-casework database in Spanish. In *Proceedings of the 9th Annual Conference of the International Speech Communication Association, Interspeech 2008*, pages 1493–1496, Brisbane, Australia. ISCA. Available from: http://atvs.ii.uam.es/files/2008_Interspeech_RamosDatabaseMismatch_v7.pdf.

[Reich, 1981] Reich, A. R. (1981). Detecting the presence of vocal disguise in the male voice. *The Journal of the Acoustical Society of America*, 69(5):1458–1461. Available from: http://link.aip.org/link/JASMAN/v69/i5/p1458/s1&Agg=doi.

[Reich and Duke, 1979] Reich, A. R. and Duke, J. E. (1979). Effects of selected vocal disguises upon speaker identification by listening. *The Journal of the Acoustical Society of America*, 66(4):1023–1028. Available from: http://link.aip.org/link/JASMAN/v66/i4/p1023/s1&Agg=doi.

[Reich et al., 1976] Reich, A. R., Moll, K. L., and Curtis, J. F. (1976). Effects of selected vocal disguises upon spectrographic speaker identification. *The Journal of the Acoustical Society of America*, 60(4):919–925. Available from: http://link.aip.org/link/JASMAN/v60/i4/p919/s1&Agg=doi.

[Reynolds, 1995] Reynolds, D. A. (1995). Speaker identification and verification using Gaussian mixture speaker models. *Speech Communication*, 17(1-2):91–108. Available from: http://linkinghub.elsevier.com/retrieve/pii/016763939500009D.

[Reynolds, 2002] Reynolds, D. A. (2002). An overview of automatic speaker recognition technology. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2002*, volume 4, pages 4072–4075, Orlando, Florida, USA. IEEE. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1004813.

[Reynolds, 2003] Reynolds, D. A. (2003). Channel robust speaker verification via feature mapping. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2003*, volume 2, pages 53–56, Hong Kong. IEEE. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1202292.

[Reynolds et al., 2003] Reynolds, D. A., Andrews, W., Campbell, J. P., Navratil, J., Peskin, B., Adami, A., Klusacek, D., Abramson, J., Mihaescu, R., Godfrey, J., and Jones, D. (2003). The SuperSID project: exploiting high-level information for high-accuracy speaker recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2003*, volume 4, pages 784–787, Hong Kong. IEEE. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1202760.

[Reynolds and Carlson, 1995] Reynolds, D. A. and Carlson, B. (1995). Text-Dependent Speaker Verification Using Decoupled and Integrated Speaker and Speech Recognizers. In *Proceedings of the 4th European Conference on Speech Communication and Technology, Eurospeech 1995*, pages 647–650, Madrid, Spain. ISCA.

[Reynolds et al., 2009] Reynolds, D. A., Kenny, P., and Castaldo, F. (2009). A Study of New Approaches to Speaker Diarization. In *Proceedings of the 10th Annual Conference of the International Speech Communication Association, Interspeech 2009*, pages 1047–1050, Brighton, UK. ISCA. Available from: http://www.crim.ca/perso/patrick.kenny/IS090471.PDF.

[Reynolds et al., 2000] Reynolds, D. A., Quatieri, T. F., and Dunn, R. B. (2000). Speaker Verification Using Adapted Gaussian Mixture Models. *Digital Signal Processing*, 10(1-3):19–41. Available from: http://dx.doi.org/10.1006/dspr.1999.0361.

[Reynolds and Rose, 1995] Reynolds, D. A. and Rose, R. C. (1995). Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE Transactions On Speech And Audio Processing*, 3(1):72–83. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=365379.

[Richiardi and Drygajlo, 2008] Richiardi, J. and Drygajlo, A. (2008). Evaluation of Speech Quality Measures for the Purpose of Speaker Verification. In *Proceedings of Odyssey 2008 - The Speaker and Language Recognition Workshop*, Stellenbosch, South Africa. ISCA. Available from: http://www.isca-speech.org/archive_open/odyssey_2008/od08_005.html.

[Richiardi et al., 2006a] Richiardi, J., Drygajlo, A., and Prodanov, P. (2006a). Confidence and reliability measures in speaker verification. *Journal of the Franklin Institute*, 343(6):574–595. Available from: http://www.sciencedirect.com/science/article/pii/S0016003206001013.

[Richiardi et al., 2006b] Richiardi, J., Drygajlo, A., and Prodanov, P. (2006b). Speaker Verification with Confidence and Reliability Measures. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2006*, pages 641–644, Toulouse, France. IEEE. Available from: http://linkinghub.elsevier.com/retrieve/pii/S0016003206001013.

[Richiardi et al., 2005] Richiardi, J., Prodanov, P., and Drygajlo, A. (2005). A probabilistic measure of modality reliability in speaker verification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2005*, Philadelphia, Pennsylvania, USA. IEEE.

[Roberts, 2002] Roberts, P. (2002). Visa Gets Behind Voice Recognition,. *PCWorld*. Available from: http://www.pcworld.com/article/106142/visa_gets_behind_voice_recognition.html.

[Rodman, 2003] Rodman, R. D. (2003). Speaker Recognition of Disguised Voices: A Program for Research. Technical report, North Carolina State University, Raleigh, NC, USA. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.116.7877.

[Sadjadi and Hansen, 2013] Sadjadi, S. O. and Hansen, J. H. L. (2013). Robust Front-end Processing for Speaker Identification over Extremely Degraded Communication Channels. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013*, pages 7214 – 7218, Vancouver, British Columbia, Canada. IEEE. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6639063&tag=1.

[Satoh et al., 2001] Satoh, T., Masuko, T., Kobayashi, T., and Tokuda, K. (2001). A Robust Speaker Verification System against Imposture Using an HMM-based Speech Synthesis System. In *Proceedings of the 7th European Conference on Speech Comunication and Technology, Interspeech 2001*, pages 759–762, Aalborg, Denmark. ISCA.

[Schwartz et al., 1982] Schwartz, R., Roucos, S., and Berouti, M. (1982). The application of probability density estimation to text-independent speaker identification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 1982*, volume 7, pages 1649–1652, Paris, France. IEEE. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1171488.

[Scoompa, 2013] Scoompa (2013). Best Voice Changer. Available from: http://play.google.com.

[Senoussaoui et al., 2011a] Senoussaoui, M., Kenny, P., Brummer, N., De Villiers, E., and Dumouchel, P. (2011a). Mixture of PLDA Models in I-Vector Space for Gender-Independent Speaker Recognition. In *Proceedings of the 12th Annual Conference of the International Speech Communication Association, Interspeech 2011*, pages 25–28, Florence, Italy. ISCA. Available from: http://www.crim.ca/perso/patrick.kenny/MixturePLDA_Interspeech2011_SubmissionV.pdf.

[Senoussaoui et al., 2010] Senoussaoui, M., Kenny, P., Dehak, N., and Dumouchel, P. (2010). An i-vector Extractor Suitable for Speaker Recognition with both Microphone and Telephone Speech. In *Proceedings of Odyssey 2010 - The Speaker and Language Recognition Workshop*, Brno, Czech Republic. ISCA. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.180.3255.

[Senoussaoui et al., 2011b] Senoussaoui, M., Kenny, P., Dumouchel, P., and Castaldo, F. (2011b). Well-calibrated heavy tailed Bayesian speaker verification for microphone speech. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2011*, pages 4824–4827, Prague, Czech Republic. IEEE. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5947435.

[Shinan and Almeida, 1986] Shinan, L. and Almeida, A. (1986). The effects of voice disguise upon formant transition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 1986*, volume 11, pages 885–888, Tokyo, Japan. IEEE. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1168880.

[Shriberg et al., 2005] Shriberg, E., Ferrer, L., Kajarekar, S. S., Venkataraman, A., and Stolcke, A. (2005). Modeling prosodic feature sequences for speaker recognition. *Speech Communication*, 46(3-4):455–472. Available from: http://www.speech.sri.com/papers/speechcomm2005-snerfs.pdf.

[Simonchik et al., 2012] Simonchik, K., Pekhovsky, T., Shulipa, A., and Afanasyev, A. (2012). Supervised Mixture of PLDA Models for Cross-Channel Speaker Verification. In *Proceedings of the 13th Annual Conference of the International Speech Communication Association, Interspeech 2012*, Portland, Oregon, USA. ISCA.

[Solewicz and Koppel, 2005] Solewicz, Y. and Koppel, M. (2005). Considering Speech Quality in Speaker Verification Fusion. In *Proceedings of the 9th European Conference on Speech Communication and Technology, Interspeech 2005*, pages 2189–2192, Lisbon, Portugal. ISCA. Available from: http://eprints.pascal-network.org/archive/00001495/.

[Solomonoff et al., 2005] Solomonoff, A., Campbell, W. M., and Boardman, I. (2005). Advances in channel compensation for SVM speaker recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2005*, volume 1, pages 629 – 632, Philadelphia, Pennsylvania, USA. IEEE. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1415192&tag=1.

[Stevens and Hanson, 1994] Stevens, K. N. and Hanson, H. M. (1994). Classification of glottal vibration from acoustic measurements. *Vocal fold physiology*, pages 147–170.

[Sturim and Reynolds, 2005] Sturim, D. E. and Reynolds, D. A. (2005). Speaker adaptive cohort selection for Tnorm in text-independent speaker verification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2005*, pages 741–744, Philadelphia, Pennsylvania, USA. IEEE. Available from: http://extwebprod.ll.mit.edu/mission/communications/ist/publications/050319_Sturim.pdf.

[Stylianou, 2009] Stylianou, Y. (2009). Voice Transformation: A survey. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2009*, pages 3585–3588. IEEE. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4960401.

[Stylianou and Cappe, 1998] Stylianou, Y. and Cappe, O. (1998). A system for voice conversion based on probabilistic classification and a harmonic plus noise model. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 1998*, pages 281–284, Seattle, Washington, USA. IEEE. Available from: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=674422.

[Stylianou et al., 1998] Stylianou, Y., Cappe, O., and Moulines, E. (1998). Continuous probabilistic transform for voice conversion. *IEEE Transactions on Speech and Audio Processing*, 6(2):131–142. Available from: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=661472.

[Talkin, 1995] Talkin, D. (1995). A robust algorithm for pitch tracking (RAPT). In *Speech coding and synthesis*, chapter 14, pages 495–518. Elsevier Science. Available from: http://www.ee.columbia.edu/~dpwe/papers/Talkin95-rapt.pdf.

[Twiscon Software, 2013] Twiscon Software (2013). Simple Voice Changer. Available from: http://play.google.com.

[Valbret et al., 1992] Valbret, H., Moulines, E., and Tubach, J. P. (1992). Voice transformation using PSOLA technique. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 1992*, pages 145–148, San Francisco, California, USA. IEEE. Available from: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=225951.

[Vaquero, 2011] Vaquero, C. (2011). *Robust Diarization for Speaker Characterization*. PhD thesis, University of Zaragoza. Available from: http://zaguan.unizar.es/record/7065?ln=es.

[Vaquero et al., 2009] Vaquero, C., Scheffer, N., and Kajarekar, S. S. (2009). Impact of Prior Channel Information for Speaker Identification. In Tistarelli, M. and Nixon, M., editors, *Advances in Biometrics, Proceedings of the Third International Conference on Advances in Biometrics, ICB 2009*, volume 5558 of *Lecture Notes in Computer Science*, pages 443–453, Alghero, Italy. Springer-Verlag Berlin, Heidelberg. Available from: http://dx.doi.org/10.1007/978-3-642-01793-3_46.

[Villalba, 2011] Villalba, J. (2011). SPLDA. Technical report, University of Zaragoza, Zaragoza. Available from: http://www.mendeley.com/download/public/1833661/4877455762/15ff05f6c4b1f6319b77bee39add478f391e30a8/dl.pdf.

[Villalba and Lleida, 2014] Villalba, J. and Lleida, E. (2014). Unsupervised Adaptation of PLDA by Using Variational Bayes Methods. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014*, Florence, Italy. IEEE.

[Villalba et al., 2012] Villalba, J., Lleida, E., Ortega, A., and Miguel, A. (2012). I3A SRE12 System Description. In *NIST SRE12 Speaker Recognition Workshop*, Orlando, Florida, USA. Available from: http://www.mendeley.com/download/public/1833661/5363803174/ac7ad01c801c9fed90da012a9a03353fe159c43d/dl.pdf.

[Villalba et al., 2010] Villalba, J., Vaquero, C., Lleida, E., Ortega, A., and Miguel, A. (2010). I3A NIST SRE2010 System Description. In *Proceedings of the V Jornadas de Reconocimiento Biometrico de Personas, JRBP 2010*, pages 241–250, Huesca, Spain. Available from: http://www.mendeley.com/download/public/1833661/3949230302/8772cb1e3df09db42090e2318c333e2e33cfe7be/dl.pdf.

[Villalba et al., 2008] Villalba, J., Vaquero, C., Lleida, E., Ortega, A., Miguel, A., Garcia, J. E., Buera, L., and Saz, O. (2008). Experiencia del I3A en la Evaluación de Reconocimiento de Locutor NIST 2008. In *Proceedings of the IV Jornadas de Reconocimiento Biometrico de Personas, JRBP 2008*, Valladolid, Spain. Available from: http://www.mendeley.com/download/public/1833661/3948390932/fdd3f2ee82702e40498fa82175e0e764915752ad/dl.pdf.

[Woodbury, 1950] Woodbury, M. A. (1950). Inverting modified matrices. Technical report, Princeton University, Princeton.

[Wu et al., 2012] Wu, Z.-z., Chng, E. S., and Li, H. (2012). Detecting Converted Speech and Natural Speech for anti-Spoofing Attack in Speaker Recognition. In *Proceedings of the 13th Annual Conference of the International Speech Communication Association, Interspeech 2012*, Portland, OR, USA. ISCA. Available from: http://www3.ntu.edu.sg/home/wuzz/papers/IS2012_SyntheticDetection_v4.pdf.

[Wu et al., 2013] Wu, Z.-z., Larcher, A., Lee, K. A., Chng, E. S., Kinnunen, T., and Li, H. (2013). Vulnerability Evaluation of Speaker Verification Under Voice Conversion Spoofing: The Effect of Text Constraints. In *Proceedings of the 14th Annual Conference of the International Speech Communication Association, Interspeech 2013*, pages 950–954, Lyon, France. ISCA. Available from: http://www3.ntu.edu.sg/home/wuzz/papers/IS2013_spoofing_attack.pdf.

[Xiang et al., 2002] Xiang, B., Chaudhari, U. V., Navratil, J., Ramaswamy, G. N., and Gopinath, R. A. (2002). Short-time Gaussianization for robust speaker verification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2002*, volume 1, pages 681– 684, Orlando, Florida, USA. IEEE. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5743809&tag=1.

[Yamagishi et al., 2009] Yamagishi, J., Kobayashi, T., Nakano, Y., Ogata, K., and Isogai, J. (2009). Analysis of Speaker Adaptation Algorithms for HMM-Based Speech Synthesis and a Constrained SMAPLR Adaptation Algorithm. *IEEE Transactions on Audio, Speech and Language Processing*, 17(1):66–83. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4740153.

[Yang and Chen, 2012] Yang, Y. and Chen, L. (2012). Toward Emotional Speaker Recognition: Framework and Preliminary Results. In Zheng, W.-S., Sun, Z., Wang, Y., Chen, X., Yuen, P., and Lai, J., editors, *Biometric Recognition, Proceedings of the 7th Chinese Conference, CCBR 2012, , December 4-5*, volume 7701 of *Lecture Notes in Computer Science*, pages 235–242, Guangzhou, China. Springer-Verlag Berlin, Heidelberg. Available from: http://dx.doi.org/10.1007/978-3-642-35136-5_29.

[Zen et al., 2009] Zen, H., Tokuda, K., and Black, A. W. (2009). Statistical parametric speech synthesis. *Speech Communication*, 51(11):1039–1064. Available from: http://linkinghub.elsevier.com/retrieve/pii/S0167639309000648.

[Zenital VOIP, 2014] Zenital VOIP (2014). Voice changer calling. Available from: http://play.google.com.

[Zhang and Tan, 2008] Zhang, C. and Tan, T. (2008). Voice disguise and automatic speaker recognition. *Forensic science international*, 175(2-3):118–22. Available from: http://dx.doi.org/10.1016/j.forsciint.2007.05.019.

[Zheng et al., 2007] Zheng, N., Lee, T., and Ching, P. C. (2007). Integration of complementary acoustic features for speaker recognition. *IEEE Signal Processing Letters*, 14(3):181–184. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4100660&tag=1.

[Zhou et al., 2001] Zhou, G., Hansen, J. H. L., and Kaiser, J. F. (2001). Nonlinear feature based classification of speech under stress. *IEEE Transactions On Speech And Audio Processing*, 9(3):201–216. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=905995.