VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

Extensions to Probabilistic Linear Discriminant Analysis for Speaker Recognition

Rozšíření pro pravděpodobnostní lineární diskriminační analýzu
v rozpoznávání mluvčího

DISERTAČNÍ PRÁCE

PHD THESIS

AUTOR PRÁCE Ing. OLDŘICH PLCHOT
AUTHOR

VEDOUCÍ PRÁCE Ing. LUKÁŠ BURGET, Ph.D.
SUPERVISOR

BRNO 2014

Abstract

This thesis deals with probabilistic models for automatic speaker verification. In particular, the Probabilistic Linear Discriminant Analysis (PLDA) model, which models i–vector representation of speech utterances, is analyzed in detail. The thesis proposes extensions to the standard state-of-the-art PLDA model. The newly proposed Full Posterior Distribution PLDA also models the uncertainty associated with the i–vector generation process. A new discriminative approach to training the speaker verification system based on the PLDA model is also proposed.

When comparing the original PLDA with the model extended by considering the i–vector uncertainty, results obtained with the extended model show up to 20% relative improvement on tests with short segments. As the test segments get longer (more than one minute), the performance gain of the extended model is lower, but it is never worse than the baseline. Training data are, however, usually available in the form of segments which are sufficiently long and therefore, in such cases, there is no gain from using the extended model for training. Instead, the training can be performed with the original PLDA model and the extended model can be used if the task is to test on the short segments.

The discriminative classifier is based on classifying pairs of i–vectors into two classes representing target and non-target trials. The functional form for obtaining the score for every i–vector pair is derived from the PLDA model and training is based on the logistic regression minimizing the cross-entropy error function between the correct labeling of all trials and the probabilistic labeling proposed by the system. The results obtained with discriminatively trained system are similar to those obtained with generative baseline, but the discriminative approach shows the ability to output better calibrated scores. This property leads to a better actual verification performance on an unseen evaluation set, which is an important feature for real use scenarios.

Curriculum Vitae

Experience

2012-2014	**DARPA RATS**	Robust Automatic Transcription of Speech Working on dataset definitions, speaker and language recognition, calibration, fusion and final system delivery.
2012	**SRE 2012**	NIST Speaker Recognition Evaluation Developing speaker verification systems based on i–vectors
2011	**LRE 2011**	NIST Language Recognition Evaluation Developing language recognition systems based on i–vectors, involved in calibration and fusion.
2010–2011	**IARPA BEST**	Biometrics Exploitation Science & Technology Working on speaker verification, calibration, fusion and dataset definition
2010	**MOBIO**	EC FP7 project, Mobile Biometry Developing SVM-based speaker recognition system.
2010	**SRE 2010**	NIST Speaker Recognition Evaluation Working in BUT/Agnitio/CRIM team, responsible for SVM-based systems (processing neural-net-based, acoustic, and JFA-derived featureWs)
2009	**LRE 2009**	NIST Language Recognition Evaluation Working in BUT/Agnitio team, responsible for defining and preparation of data setd and building phonotactic SVM-based systems. Involved also in system calibration and fusion.
2008–2009	**EOARD**	EOARD 083066, Improving the capacity of language recognition systems to handle rare languages using radio broadcast data
2007	**LRE 2007**	'NIST Language Recognition Evaluation' Working in BUT team, responsible for data mining using radio broadcast data for language recognition.

Teaching

During my PhD studies, I was regularly teaching computer laboratories of the Signal and Systems course. For one year, I was responsible for lectures in Signal and Systems for international students. I was regularly teaching in computer laboratories of the Speech Signal Processing course. I was also giving lectures on speaker and language recognition in the Speech Signal Processing and Classification and Recognition courses.

Tertiary Education

Since 9/2007	**PhD studies** at FIT, Brno University of Technology Concentrated mainly on language identification and speaker recognition
9/2001 - 6/2007	**Computer Science and Engineering** at Brno University of Technology
Degree	**Diploma** with emphasis in **Computer networking** and **Pattern classification**)
Diploma Thesis *(9/2006 - 5/2007)*	**”User Oriented QoS System”** Using artificial neural networks to classify network data flows
Student Research Project *(9/2004 - 5/2005)*	**”Comparison of network stack of Linux and BSD kernels”**

Contents

Chapter 1

Introduction

Automatic speaker recognition (SRE) is a process of comparing bio-metric signals produced by the human vocal tract and answering the question to whom the given signal belongs or simply whether two signals were produced by the same individual.

Similarly to the DNA, image of the iris, contour lines of the fingerprints, etc. — voice is a common type of bio-metric data, which every individual can produce and which is easy to capture. Thanks to its nature of being easily obtained, the the bio-metric systems based on voice find a broad use in law-enforcement and intelligence. This property, however, is not desired in the authentication systems. Therefore, in such scenarios of using voice for authentication, the voice verification is usually combined with other methods like knowing the secret password or providing additional bio-metric signals. If the voice is to be a single source of bio-metric data and the system knows the supposedly secret content of the speech and is able to use this knowledge, then we consider the SRE system as *text-dependent*, otherwise we talk about a *text-independent* system.

Speech is also a very complex signal carrying not only the desired content, but also other various information. After it is produced by a vocal tract, which is characteristic to every speaker and therefore it inputs most of the speaker-related information to the signal, it passes through some environment to a point where it is recorded. This environment or *channel* has a great effect on the quality of such signal, which causes the degradation in performance of SRE systems. This behavior is, of course, an important topic for research and we will address it in this work as well.

An SRE system is built with an assumption that the information relevant to the speaker in the given recording is independent on the information related to channel, language, content (in case of the text-independent system), etc. Current state-of-the-art systems are designed to decouple the information contained in the signal into the speaker- and channel-related parts. As already mentioned, the problem can be viewed as answering two types of questions: (i) Who is speaking in this recording? — then we talk about the *speaker identification* or (ii-a) Is it the same speaker speaking in these two (or even more) recordings? or (ii-b) Is this speaker speaking in this recording? — then we talk about *speaker verification*.

Both questions (ii-a) and (ii-b) represent a so-called speaker verification *trial*. If the correct answer is “yes” then the trial is called a *target trial*. If “no” is the correct answer, then we talk about a *non-target trial*.

As we can see, speaker verification constitutes a two-class problem, where the task is to decide whether a test utterance belongs to a given speaker, or, equivalently, whether a set of recordings (e.g. one enrollment and one test utterance) belongs to the same speaker. These two very similar formulations are equivalent, but they correspond to two different discriminative approaches. We will address the latter formulation when describing a discriminative approach later.

An example for the verification task can be a scenario widely used by a law enforcement. Given some utterances belonging to a particular person, the goal is to search in a collection of data and find the recordings corresponding to the given person. A speaker verification can be turned into identification, by restricting the set of compared utterances.

Speaker identification is then a multi-class classification problem, where the task is to assign a correct label to the utterance, where each label corresponds to one of the speakers from the set of known speakers. The assumption, whether the test segment belongs to the set of known speakers, constitutes two classification problems: the *closed set identification* — the segment is always assumed to belong to one of the speakers, and the *open-set identification* — the segment does not have to belong to any of the speakers. The open-set problem is a more difficult scenario. If a new speaker is to be added to the known speaker set, a procedure called enrollment is carried out. It consists of collecting a sufficient amount of speech data, assigning it a unique speaker label and creating a corresponding *speaker model*.

1.1 Motivation and Contribution

My work on the topics of this thesis started when I was building subsystems for the NIST SRE 2010 in the team of people from Agnitio, Brno and Crim (ABC). Later during the 2010 BOSARIS workshop held in Brno, I was working on the analysis of systems submitted by the ABC team to the NIST SRE 2010. The main focus was on Probabilistic Linear Discriminant Analysis using i–vectors as features as it showed excellent results in the evaluations. At that time it was already becoming apparent that PLDA and i–vectors will become a new state-of-the-art in SRE. I was also working with Lukáš Burget on one of the research directions, where the goal was to formulate a discriminative way of training the PLDA-like model. The goal of obtaining a discriminatively trained SRE system based on the PLDA was successfully achieved [Burget et al., 2011, Cumani et al., 2011] and for a very short time (until the introduction of the i–vector length normalization [Garcia-Romero, 2011]), this technique was providing the best results. I continued my work on discriminative training, dataset design and calibration [Ferrer et al., 2012, Ferrer et al., 2011] as a member of BUT and SRI team in the IARPA Biometrics Exploitation & Science Technology (BEST) program. Later on, I was working with Sandro Cumani on various topics in SRE, the main being the extension of the PLDA model [Cumani et al., 2014], which takes into account the uncertainty about the i–vector. As the uncertainty of the i–vector estimate depends mainly on the duration of speech segments from which the i–vectors are extracted, the proposed extension turned out to be effective mainly for short segments. At the same time when developing the PLDA extension, I was also working both on a speaker- and language modeling, calibration and fusion [Plchot et al., 2013] for a DARPA RATS (Robust Automatic Transription of Speech) project in a team led by BBN Technologies. Working on RATS allowed me to compare generative PLDA with its discriminative counterpart in a very noisy and degraded acoustic environment.

1.1.1 Claims

The goal of this thesis is to investigate the contemporary state-of-the-art techniques in text-independent speaker verification field. The main focus is on the analysis and further improvement of the Probabilistic Linear Discriminant Analysis (PLDA). The main contributions can be summarized in the following points:

- **Analysis of the PLDA**: I analyzed the performance of presented methods on various datasets representing different levels of acoustic signal distortions and channel variabilities. Also a direct comparison of the main techniques considered as the state-of-the-art before introduction of PLDA is provided on a common dataset.
- **Extension of the PLDA**: The proposed extended PLDA model takes into account an uncertainty of the input features, which improves performance on the short speech segments with respect to the original PLDA model.
- **Discriminative training of the PLDA**: The proposed discriminative approach to PLDA model training offers an interesting alternative to the currently preferred generative approach. Presented results suggest that the discriminatively trained PLDA model offers well calibrated outputs and therefore poses as a viable option for a practical use.

Chapter 2

Gaussian Mixture Modeling of Acoustic Features

The main role of the GMM is to estimate an underlying distribution of acoustic features extracted from speech segments and inherently model the hidden classes, which are being formed by individual speakers, various acoustic channels or some other common properties. This ability of unsupervised modeling of classes is later exploited by a supervised algorithm focused on extracting the information about the distributions of particular classes, e.g. those associated with speaker identities.

Let us define a speech segment as a set of F-dimensional acoustic features: $\mathcal{X} = \{\mathbf{x}_1 \mathbf{x}_2 \ldots \mathbf{x}_\mathcal{T}\}$. A GMM [Bishop, 2006] is then defined as a weighted sum (mixture) of a set of C multivariate normal distributions of the form:

$$p(\mathbf{x}|\mathcal{G}) = \sum_{c=1}^{C} w^{(c)} \mathcal{N}\left(\mathbf{x}; \boldsymbol{\mu}^{(c)}, \boldsymbol{\Sigma}^{(c)}\right), \tag{2.1}$$

where $p(\mathbf{x}|\mathcal{G})$ is the probability of $\mathbf{x}$ given the GMM model $\mathcal{G}$ with C mixtures and $w^{(c)}$ are individual mixture weights, also called mixing coefficients, satisfying the constraints that $w^{(c)} \geq 0$ and $\sum_{c=1}^{C} w^{(c)} = 1$. $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}^{(c)}, \boldsymbol{\Sigma}^{(c)})$ is a F-variate Gaussian component PDF with mean $\boldsymbol{\mu}^{(c)}$ and covariance matrix $\boldsymbol{\Sigma}^{(c)}$:

$$\mathcal{N}\left(\mathbf{x}; \boldsymbol{\mu}^{(c)}, \boldsymbol{\Sigma}^{(c)}\right) = \frac{1}{(2\pi)^{F/2} |\boldsymbol{\Sigma}^{(c)}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}^{(c)})^{\mathrm{T}} \boldsymbol{\Sigma}^{(c)^{-1}} (\mathbf{x}-\boldsymbol{\mu}^{(c)})}. \tag{2.2}$$

The whole GMM $\mathcal{G}$ is then represented by parameters

$$\lambda = \left\langle w^{(c)}, \boldsymbol{\mu}^{(c)}, \boldsymbol{\Sigma}^{(c)} \right\rangle \qquad \text{with} \qquad c = 1 \ldots C, \tag{2.3}$$

or more conveniently by the supervectors and the matrix of stacked parameters as:

$$\lambda = \langle \mathbf{w}, \boldsymbol{\mu}, \boldsymbol{\Sigma} \rangle = \left\langle \begin{bmatrix} w^{(1)} \\ \vdots \\ w^{(C)} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\mu}^{(1)} \\ \vdots \\ \boldsymbol{\mu}^{(C)} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}^{(1)} & 0 & \cdots & 0 \\ 0 & \boldsymbol{\Sigma}^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{\Sigma}^{(C)} \end{bmatrix} \right\rangle \tag{2.4}$$

$$. \tag{2.5}$$

It should be noted, that the covariance matrices can be full rank or constrained to be diagonal. Sometimes, the parameters can be also shared among the Gaussian components. In general, the configuration with full covariance matrices needs more training data to properly estimate all the parameters. Often the GMM with larger amount of components with diagonal covariance matrices is used instead of the configuration with full rank covariance matrices.

For evaluating the GMM model given the data, and therefore also for estimating its parameters, it is necessary to define the quantities associated with individual GMM components. Having observed the data point $\mathbf{x}_i$, posterior probabilities $p(c|\mathbf{x}_i)$—also referred to as *occupation probabilities* and shortly denoted as $\gamma_i^{(c)}$—can be computed using the Bayes rule:

$$\gamma_i^{(c)} = \frac{w^{(c)}\mathcal{N}\left(\mathbf{x}_i; \boldsymbol{\mu}^{(c)}, \boldsymbol{\Sigma}^{(c)}\right)}{\sum_{c=1}^{C} w^{(c)}\mathcal{N}\left(\mathbf{x}_i; \boldsymbol{\mu}^{(c)}, \boldsymbol{\Sigma}^{(c)}\right)}. \tag{2.6}$$

The configuration of the posterior probabilities for each feature vector is referred to as the *alignment* of the data to the mixture components. In this text, we will always assume, that the alignment of the feature vectors to Gaussian components is always based on UBM.

It is also convenient to define Baum-Welch statistics. Having our speech segment $\mathcal{X}$ which consists of $i = 1 \ldots \tau$ feature vectors of dimensionality F and the alignment of each feature vector $\mathbf{x}_i$ defined by (2.6), the Baum-Welsch [Kenny et al., 2007] statistics are defined as

$$N^{(c)} = \sum_{i=1}^{\tau} \gamma_i^{(c)} \tag{2.7}$$

$$\mathbf{f}^{(c)} = \sum_{i=1}^{\tau} \gamma_i^{(c)} \mathbf{x}_i \tag{2.8}$$

$$\mathbf{S}^{(c)} = \sum_{i=1}^{\tau} \gamma_i^{(c)} \mathbf{x}_i \mathbf{x}_i^{\mathrm{T}} \ . \tag{2.9}$$

We refer to these as the zero-, the first-, and the second-order statistics (or cumulants) respectively. For the simplification of the derivations, often the statistics centered around the UBM mean are defined as

$$\tilde{\mathbf{f}}^{(c)} = \mathbf{f}^{(c)} - N^{(c)}\boldsymbol{\mu}^{(c)} \tag{2.10}$$

$$\tilde{\mathbf{S}}^{(c)} = \mathbf{S}^{(c)} - \mathbf{f}^{(c)}\boldsymbol{\mu}^{(c)\mathrm{T}} - \boldsymbol{\mu}^{(c)}\mathbf{f}^{(c)\mathrm{T}} + N^{(c)}\boldsymbol{\mu}^{(c)}\boldsymbol{\mu}^{(c)\mathrm{T}} \tag{2.11}$$

For further simplification, the statistics can be stacked into the form of supervector and matrices as:

$$\mathbf{N} = \begin{bmatrix} N^{(1)}\mathbf{I} & 0 & \cdots & 0 \\ 0 & N^{(2)}\mathbf{I} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & N^{(C)}\mathbf{I} \end{bmatrix}$$

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}^{(1)} \\ \vdots \\ \mathbf{f}^{(C)} \end{bmatrix} \tag{2.12}$$

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}^{(1)} & 0 & \cdots & 0 \\ 0 & \mathbf{S}^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{S}^{(C)} \end{bmatrix},$$

where the identity matrices in (2.12) have the same dimensionality as the feature vector. Stacked centered statistics $\tilde{\mathbf{f}}$ and $\tilde{\mathbf{S}}$ are created according to the same scheme as their non-centered version.

2.1 Maximum Likelihood Estimate of Parameters

Given enough training data and some initial GMM configuration $\lambda^{(0)}$, we want to estimate the new parameters, which best matches the underlying distribution of the data. A possible approach is to perform a Maximum-Likelihood (ML) estimate [Reynolds and Rose, 1995, Bishop, 2006] and search for the solution of

$$\lambda_{ML} = \arg\max_{\lambda} p(\mathcal{X}|\lambda) \tag{2.13}$$

Assuming that the statistical independence of the frames/feature vectors, the likelihood of the data $\mathcal{X}$, given the model parameters λ, is given as

$$p(\mathcal{X}|\lambda) = \prod_{i=1}^{\tau} \mathcal{G}(\mathbf{x}_i; \lambda) \; . \tag{2.14}$$

Usually, the logarithm of the likelihood is required for evaluating the model and estimating the parameters. Its basic form is given as

$$\log p(\mathcal{X}|\lambda) = \sum_{i=1}^{\tau} \log \sum_{c=1}^{C} w^{(c)} \mathcal{N}\left(\mathbf{x}_i; \boldsymbol{\mu}^{(c)}, \boldsymbol{\Sigma}^{(c)}\right) . \tag{2.15}$$

For any choice of distributions $q_i(c)$ over the Gaussian components, we can rewrite this likelihood as

$$\begin{aligned}\log p(\mathcal{X}|\lambda) &= \sum_{i=1}^{\tau} \log p(\mathbf{x}_i|\lambda) = \sum_{i=1}^{\tau} \underbrace{\sum_{c=1}^{C} q_i(c)}_{1} \log \frac{p(\mathbf{x}_i, c|\lambda)}{p(c|\mathbf{x}_i, \lambda)} \frac{q_i(c)}{q_i(c)} \\ &= \sum_{i=1}^{\tau} \left[\sum_{c=1}^{C} q_i(c) \log \left(w^{(c)} \mathcal{N}\left(\mathbf{x}_i; \boldsymbol{\mu}^{(c)}, \boldsymbol{\Sigma}^{(c)}\right)\right) \right. \\ &\left. - \sum_{c=1}^{C} q_i(c) \log q_i(c) + \sum_{c=1}^{C} q_i(c) \log \frac{q_i(c)}{\gamma_i^{(c)}} \right] ,\end{aligned} \tag{2.16}$$

where the last term

$$\sum_{c=1}^{C} q_i(c) \log \frac{q_i(c)}{\gamma_i^{(c)}} = \mathrm{D}_{\mathrm{KL}}(q_i(c) \| \gamma_i^{(c)}) \tag{2.17}$$

corresponds to the Kullback-Leibler (KL) divergence between $q_i(c)$ and the posterior distribution $p(c|\mathbf{x}_i, \lambda) = \gamma_i^{(c)}$. Hence, if we set $q_i(c)$ to the true posterior $\gamma_i^{(c)}$, the KL divergence vanishes and the likelihood can be expressed as

$$\log p(\mathcal{X}|\lambda) = \sum_{i=1}^{\tau} \left[\sum_{c=1}^{C} \gamma_i^{(c)} \log \left(w^{(c)} \mathcal{N}\left(\mathbf{x}_i; \boldsymbol{\mu}^{(c)}, \boldsymbol{\Sigma}^{(c)}\right)\right) - \sum_{c=1}^{C} \gamma_i^{(c)} \log \gamma_i^{(c)} \right] . \tag{2.18}$$

Using the Baum-Welch statistics, we can further rewrite the log-likelihood [Kenny et al., 2004] and get

$$\begin{aligned}\log p(\mathcal{X}|\lambda) = \sum_{c=1}^{C} & \left[N^{(c)} \log \frac{1}{(2\pi)^{F/2} |\boldsymbol{\Sigma}^{(c)}|^{1/2}} \right. \\ & \left. - \frac{1}{2} \mathrm{tr} \left(\boldsymbol{\Sigma}^{(c)^{-1}} \left(\mathbf{S}^{(c)} - \mathbf{f}^{(c)} \boldsymbol{\mu}^{(c)^{\mathrm{T}}} - \boldsymbol{\mu}^{(c)} \mathbf{f}^{(c)^{\mathrm{T}}} + N^{(c)} \boldsymbol{\mu}^{(c)} \boldsymbol{\mu}^{(c)^{\mathrm{T}}} \right) \right) \right] \\ & - \sum_{i=1}^{\tau} \sum_{c=1}^{C} \gamma_i^{(c)} \log \frac{\gamma_i^{(c)}}{w^{(c)}},\end{aligned} \tag{2.19}$$

which is the correct likelihood, if the statistics were collected with the true posterior distribution $\gamma_i^{(c)}$. If the true posterior distribution is not available and is provided via different model, e.g. Universal Background Model (UBM), then this function serves as an approximation and a lower-bound of the correct likelihood, since the omitted KL divergence is always non-negative.

Unfortunately, direct optimization of the parameters given the data is analytically intractable. However, ML estimates of the parameters can be obtained iteratively by means of EM algorithm [Dempster et al., 1977, Bishop, 2006].

For the E-step of the EM algorithm, the auxiliary function can be constructed from (2.19) as

$$\begin{aligned}\mathcal{Q}_{\mathrm{GMM}}(\lambda, \lambda^{(0)}) = \sum_{c=1}^{C} & \left[N_{\lambda_0}^{(c)} \log \frac{1}{(2\pi)^{F/2} |\boldsymbol{\Sigma}^{(c)}|^{1/2}} \right. \\ & \left. - \frac{1}{2} \mathrm{tr} \left(\boldsymbol{\Sigma}^{(c)^{-1}} \left(\mathbf{S}_{\lambda_0}^{(c)} - \mathbf{f}_{\lambda_0}^{(c)} \boldsymbol{\mu}^{(c)^{\mathrm{T}}} - \boldsymbol{\mu}^{(c)} \mathbf{f}_{\lambda_0}^{(c)^{\mathrm{T}}} + N_{\lambda_0}^{(c)} \boldsymbol{\mu}^{(c)} \boldsymbol{\mu}^{(c)^{\mathrm{T}}} \right) \right) \right] \\ & + \sum_{c=1}^{C} \log w^{(c)}.\end{aligned} \tag{2.20}$$

By fixing the alignment of the data using the current model estimate $\lambda^{(0)}$, we obtain $\gamma_{i}{}_{\lambda_0}^{(c)}$ and collect the statistics $\{N_{\lambda_0}^{(c)}, \mathbf{f}_{\lambda_0}^{(c)}, \mathbf{S}_{\lambda_0}^{(c)}\}$. In the M-step of the algorithm, the new ML estimate of parameters is then computed as

$$\theta_{\mathrm{ML}} = \arg\max_{\lambda} \mathcal{Q}_{\mathrm{GMM}}(\lambda, \lambda^{(0)}) \tag{2.21}$$

for which the update formulas are given as:

$$\begin{aligned} \boldsymbol{\mu}_{\mathrm{ML}}^{(c)} &= \frac{1}{N^{(c)}}\mathbf{f}^{(c)} \\ \boldsymbol{\Sigma}_{\mathrm{ML}}^{(c)} &= \frac{1}{N^{(c)}}\mathbf{S}^{(c)} - \boldsymbol{\mu}_{\mathrm{ML}}^{(c)}\boldsymbol{\mu}_{\mathrm{ML}}^{(c)\,\mathrm{T}} \\ w_{\mathrm{ML}}^{(c)} &= \frac{N^{(c)}}{\tau}\ . \end{aligned} \tag{2.22}$$

Repeating the E and M steps guarantees not to decrease the likelihood and iterating is usually stopped when the likelihood increase in two consecutive iterations is smaller than some convergence threshold. For more detailed derivations following roughly our notation, we refer the kind reader to [Glembek, 2012].

2.2 Latent Variable Models for Speaker Recognition

In this Section, we will describe essential techniques based on Factor Analysis [Bishop, 2006]. These techniques build upon the MAP estimate of the speaker-dependent GMM, while taking into account either inter- or intra-session variability or both of them at the same time. To study the problematic in detail, we refer the reader to the following publications [Kenny, 2005, Kenny et al., 2007, Kenny et al., 2005].

Let us begin with a brief description of MAP adaptation in terms of hidden variable models by following [Kenny, 2005]. Continuing with the notation of GMM from previous section, we will define the a speaker-dependent supervector $\mathbf{g(s)}$ as a latent variable model for speaker s as

$$\mathbf{g}(s) = \boldsymbol{\mu} + \mathbf{Dz}(s). \tag{2.23}$$

The speaker-dependent supervector is distributed according to $\mathbf{g} \sim (\boldsymbol{\mu}, \mathbf{DD}^{\mathrm{T}})$ and a $CF \times S$ matrix $\mathbf{D}$ acts as a prior on the UBM mean supervector $\boldsymbol{\mu}$. Latent variable $\mathbf{z}_s$ is a S-dimensional speaker-dependent hidden vector distributed according to the standard normal distribution, $\mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$. The S in the dimensionalities of the variabilities denotes an arbitrary positive number and will be discussed later in the end of Section 2.2.1.

The log-likelihood of data and hidden variable is based on the general GMM log-likelihood function as defined in Section 2.1. We will assume fixed data alignment [Kenny, 2005] and represent the log-likelihood by the means of the Baum-Welch statistics collected using UBM. As already discussed in the previous section, this is an approximated log-likelihood acting as a lower-bound to the real log likelihood. Using the Universal Background Model to collect the statistics for all observations $\mathcal{X} = \{\mathbf{x}_1\mathbf{x}_2\ldots\mathbf{x}_\tau\}$ corresponding to the speaker s, we get

$$\begin{aligned} \log p(\mathcal{X}|\mathbf{D}, \mathbf{z}) &= G + H(\mathbf{z}) \\ G &= \sum_{c=1}^{C}\left(N_{\mathcal{X}}^{(c)} \log \frac{1}{(2\pi)^{F/2}|\boldsymbol{\Sigma}^{(c)}|^{1/2}}\right) - \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\tilde{\mathbf{S}}_{\mathcal{X}}\right) \\ H(\mathbf{z}) &= \mathbf{z}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\tilde{\mathbf{f}}_{\mathcal{X}} - \frac{1}{2}\mathbf{z}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{N}_{\mathcal{X}}\mathbf{Dz}, \end{aligned} \tag{2.24}$$

where $\mathbf{\Sigma}$ is a block diagonal covariance matrix of the UBM composed as in (2.3), $\mathbf{N}_{\mathcal{X}}$, $\tilde{\mathbf{f}}_{\mathcal{X}}$ and $\tilde{\mathbf{S}}_{\mathcal{X}}$ are stacked zero-, first- and second-order centered statistics collected with the UBM according to (2.8), (2.10) and (2.7).

The joint log-likelihood of the observed data $\mathcal{X}$ and the hiden variable is given by

$$\begin{aligned}\log p(\mathcal{X}, \mathbf{z}|\mathbf{D}) &= \log p(\mathcal{X}|\mathbf{D}, \mathbf{z})p(\mathbf{z}) \\ &= K_{\mathbf{\Sigma}} + (\mathbf{z}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\mathbf{\Sigma}^{-1}\tilde{\mathbf{f}}_{\mathcal{X}} - \frac{1}{2}\mathbf{z}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\mathbf{\Sigma}^{-1}\mathbf{N}_{\mathcal{X}}\mathbf{D}\mathbf{z} - \frac{1}{2}\mathbf{z}^{\mathrm{T}}\mathbf{z}),\end{aligned} \tag{2.25}$$

where the term $K_{\mathbf{\Sigma}}$ is a constant (also referred to as a normalization term), which does not depend on $\mathbf{z}$ and $\mathbf{D}$. Leaving out the $K_{\mathbf{\Sigma}}$, the posterior of the hidden variable $\mathbf{z}$, given the data $\mathcal{X}$ observed for speaker s, is given as:

$$\log p(\mathbf{z}|\mathcal{X}) \propto \log p(\mathcal{X}, \mathbf{z}) \propto (\mathbf{z}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\mathbf{\Sigma}^{-1}\tilde{\mathbf{f}}_{\mathcal{X}} - \frac{1}{2}\mathbf{z}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\mathbf{\Sigma}^{-1}\mathbf{N}_{\mathcal{X}}\mathbf{D}\mathbf{z} - \frac{1}{2}\mathbf{z}^{\mathrm{T}}\mathbf{z}). \tag{2.26}$$

By completion of squares, the posterior for $\mathbf{z}$ is also Gaussian

$$p(\mathbf{z}|\mathcal{X}) \sim \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_{\mathbf{z}}, \Gamma_{\mathbf{z}}^{-1}) \tag{2.27}$$

with precision matrix and mean given by

$$\Gamma_{\mathbf{z}} = (\mathbf{D}^{\mathrm{T}}\mathbf{\Sigma}^{-1}\mathbf{N}_{\mathcal{X}}\mathbf{D} + \mathbf{I}) \tag{2.28}$$

$$\boldsymbol{\mu}_{\mathbf{z}} = \Gamma_{\mathbf{z}}^{-1}\mathbf{D}^{\mathrm{T}}\mathbf{\Sigma}^{-1}\tilde{\mathbf{f}}_{\mathcal{X}} \tag{2.29}$$

The mean of supervector posterior $p(\mathbf{g}|\mathcal{X})$ (i.e. its MAP estimate) is the given as

$$\begin{aligned}\hat{\mathbf{g}} &= \boldsymbol{\mu} + \mathbf{D}\boldsymbol{\mu}_{\mathbf{z}} \\ &= \boldsymbol{\mu} + \mathbf{D}(\mathbf{D}^{\mathrm{T}}\mathbf{\Sigma}^{-1}\mathbf{N}_{\mathcal{X}}\mathbf{D} + \mathbf{I})^{-1}\mathbf{D}^{\mathrm{T}}\mathbf{\Sigma}^{-1}\tilde{\mathbf{f}}_{\mathcal{X}} \\ &= \boldsymbol{\mu} + (\mathbf{N}_{\mathcal{X}} + \mathbf{\Sigma}(\mathbf{D}\mathbf{D}^{\mathrm{T}})^{-1})^{-1}\tilde{\mathbf{f}}_{\mathcal{X}}\end{aligned} \tag{2.30}$$

2.2.1 Training Prior Hyper-Parameters

In the previous section, we discussed how to artificially supply a prior by means of another model (UBM). Now, we will describe how to train it from the data in a ML fashion. The training objective is to maximize the likelihood of the training data $p(\mathcal{X}|\mathbf{D}, \mathbf{z})$. Similarly to the GMM training, the ML estimate of the parameters can be obtained by means of EM algorithm [Brümmer, 2009]. While the other parameters $\{\boldsymbol{\mu}, \mathbf{\Sigma}, \mathbf{w}\}$ could be also re-estimated, here we will consider, re-estimating only the matrix $\mathbf{D}$. Taking the $\mathbf{z}$ as a hidden variable, the EM auxiliary function is then constructed as

$$\mathcal{Q}(\mathbf{D}, \mathbf{D}_0) = \sum_s \langle \log p(\mathcal{X}_s, \mathbf{z}|\mathbf{D}_0)\rangle_{\mathbf{z}|\mathcal{X}_s, \mathbf{w}|\mathbf{D}_0}, \tag{2.31}$$

where $p(\mathcal{X}_s, \mathbf{z}|\mathbf{D}_0)$ is the joint probability of the observations $\mathcal{X}_s$ for speaker s. Considering that

$$p(\mathcal{X}_s, \mathbf{z}|\mathbf{D}) = \log p(\mathcal{X}_s|\mathbf{D}, \mathbf{z}) + \log p(\mathbf{z}) \tag{2.32}$$

and $p(\mathbf{z})$ being set to a standard normal distribution and kept fixed, there is no need to re-estimate parameters of $p(\mathbf{z})$, as any changes in the prior distribution can be equivalently accomplished by appropriately changing $\boldsymbol{\mu}$ and $\mathbf{D}$. Therefore, we can simplify the auxiliary function as

$$\mathcal{Q}(\mathbf{D}, \mathbf{D}_0) = \sum_s \langle \log p(\mathcal{X}_s|\mathbf{z}, \mathbf{D}_0)\rangle_{\mathbf{z}|\mathcal{X}_s, \mathbf{D}_0}. \tag{2.33}$$

By looking at the expression for the joint likelihood (2.25) and realizing that $K_{\mathbf{\Sigma}}$ does not depend on $\mathbf{D}$, we can further express the auxiliary function as

$$\begin{aligned}\mathcal{Q}(\mathbf{D},\mathbf{D}_0) &= \sum_s \left\langle \mathbf{z}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\mathbf{\Sigma}^{-1}\tilde{\mathbf{f}}_{\mathcal{X}_s} - \frac{1}{2}\mathbf{z}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\mathbf{\Sigma}^{-1}\mathbf{N}_{\mathcal{X}_s}\mathbf{D}\mathbf{z} \right\rangle_{\mathbf{z}|\mathcal{X}_s,\mathbf{D}_0} \\ &= \sum_s \mathrm{tr}\left[\mathbf{\Sigma}^{-1}\left(\tilde{\mathbf{f}}_{\mathcal{X}_s}\langle\mathbf{z}\rangle\mathbf{D}^{\mathrm{T}} - \frac{1}{2}\mathbf{N}_{\mathcal{X}_s}\mathbf{D}\langle\mathbf{z}\mathbf{z}^{\mathrm{T}}\rangle\mathbf{D}^{\mathrm{T}}\right)\right],\end{aligned} \tag{2.34}$$

where the expectations are taken over $\mathbf{z}|\mathcal{X}_s,\mathbf{D}_0$. Now, in order to minimize the auxiliary function, we can take its derivative with respect to $\mathbf{D}$ and set it to zero:

$$\frac{\partial}{\partial\mathbf{D}}\sum_s \mathrm{tr}\left[\mathbf{\Sigma}^{-1}\left(\tilde{\mathbf{f}}_{\mathcal{X}_s}\langle\mathbf{z}\rangle\mathbf{D}^{\mathrm{T}} - \frac{1}{2}\mathbf{N}_{\mathcal{X}_s}\mathbf{D}\langle\mathbf{z}\mathbf{z}^{\mathrm{T}}\rangle\mathbf{D}^{\mathrm{T}}\right)\right] = \mathbf{0}, \tag{2.35}$$

which gives

$$\sum_s \mathbf{\Sigma}^{-1}\left(\tilde{\mathbf{f}}_{\mathcal{X}_s}\langle\mathbf{z}\rangle - \mathbf{N}_{\mathcal{X}_s}\mathbf{D}\langle\mathbf{z}\mathbf{z}^{\mathrm{T}}\rangle\right) = \mathbf{0}. \tag{2.36}$$

We need to solve the linear system

$$\mathbf{D}^{(c)}\sum_s N_{\mathcal{X}_s}^{(c)}\langle\mathbf{z}_{\mathcal{X}_s}\mathbf{z}_{\mathcal{X}_s}^{\mathrm{T}}\rangle = \sum_s \tilde{\mathbf{f}}_{\mathcal{X}_s}^{(c)}\langle z_{\mathcal{X}_s}^{\mathrm{T}}\rangle, \tag{2.37}$$

where c is spanning the rows of the matrices corresponding to individual UBM components. The expectation over the hidden variable $\langle\mathbf{z}\rangle$ is given as a mean of the posterior distribution of $\mathbf{z}$ given the $\mathbf{D}_0$ (see (2.29)) and $\langle\mathbf{z}\mathbf{z}^{\mathrm{T}}\rangle = \langle\mathbf{z}\rangle\langle\mathbf{z}^{\mathrm{T}}\rangle + \Gamma_{\mathbf{z}}^{(-1)}$, where $\Gamma_{\mathbf{z}}^{(-1)}$ is the covariance matrix (see (2.28)) of the posterior of $\mathbf{z}$ given $\mathbf{D}_0$. Finally, the closed-form solution for computing the hyper-parameters is :

$$\mathbf{D}^c = \sum_s \left[\tilde{\mathbf{f}}_{\mathcal{X}_s}^{(c)}\boldsymbol{\mu}_{\mathbf{z}\mathcal{X}_s}(N_{\mathcal{X}_s}^{(c)}(\boldsymbol{\mu}_{\mathbf{z}\mathcal{X}_s}\boldsymbol{\mu}_{\mathbf{z}\mathcal{X}_s}^{\mathrm{T}} + \Gamma_{\mathbf{z}\mathcal{X}_s}^{(-1)}))^{-1}\right]. \tag{2.38}$$

The framework described in this section allows for setting different dimensionalities and constraints for $\mathbf{D}$. In theory, we could take $\mathbf{D}$ as a full $CF \times CF$ matrix. This would be impractical, since the amount of parameters to train would be very large. For this reason, $\mathbf{D}$ is often constrained to be diagonal or low rank. Taking $\mathbf{D}$ as a low-rank $CF \times S$ matrix constraints the speaker-dependent supervector to lie in a S-dimensional subspace, which is a widely used approach. The use of the subspace modeling will be shown in the following sections.

Chapter 3

Probabilistic Linear Discriminant Analysis and i–vectors

In the last four years, SRE systems based on the i–vectors and Probabilistic Linear Discriminant Analysis (PLDA) became state-of-the art. In PLDA model, an i–vector $\boldsymbol{\phi}$ is considered to be a realization of a random variable $\boldsymbol{\Phi}$, whose generation process can be described in terms of a set of latent variables. Different PLDA models exist, which use different numbers of hidden variables as well as different priors. The two favourite models are heavy-tailed PLDA (HTPLDA)[Kenny, 2010], where Student's t-distribution is imposed on the latent variables and the PLDA [Prince and Elder, 2007], which assumes Gaussian priors.

3.1 I–vector approach

The main idea behind the i–vector model is to transform the large utterance specific GMM supervector $\mathbf{s}$ into a small subspace, while retaining most of the important variability. From the perspective of speaker recognition, the supervector $\mathbf{s}$ contains both the speaker and inter-session characteristics of a given speech segment and is modeled according to:

$$\mathbf{s} = \mathbf{u} + \mathbf{T}\mathbf{w} \ , \tag{3.1}$$

where $\mathbf{u}$ is the UBM GMM mean supervector, composed of C GMM components of dimension F. $\mathbf{T}$ is a low-rank rectangular matrix representing M bases spanning the sub–space including important inter and intra–speaker variability in the supervector space. The subspace defined by the matrix $\mathbf{T}$ is often referred to as "i–vector subspace" or "total variability subspace". Vector $\mathbf{w}$ is a realization of a latent variable $\mathbf{W}$, of size M, having a standard normal prior distribution

$$\mathbf{W} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{3.2}$$

The principle of i–vectors resides in tying the latent variable to every utterance, independent of speaker. The same steps as already described for the subspace modeling in Section 2.2 will apply also to i–vectors.

Ultimately, the aim is to estimate the parameters of the posterior distribution of the latent variable $\mathbf{W}$ for each set of τ input features extracted from the given speech segment $\mathcal{X} = \{\mathbf{x}_1\mathbf{x}_2 \ldots \mathbf{x}_\tau\}$. Assuming the standard normal prior for $\mathbf{W}$, the posterior distribution is also Gaussian:

$$\mathbf{W}|\mathcal{X} \sim \mathcal{N}(\boldsymbol{\phi}_{\mathcal{X}}, \boldsymbol{\Gamma}_{\mathcal{X}}^{-1}). \tag{3.3}$$

with mean vector and precision matrix as in (2.28 and 2.29):

$$\begin{aligned}\boldsymbol{\phi}_{\mathcal{X}} &= \boldsymbol{\Gamma}_{\mathcal{X}}^{-1}\mathbf{T}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\tilde{\mathbf{f}}_{\mathcal{X}} \\ \boldsymbol{\Gamma}_{\mathcal{X}} &= \mathbf{I} + \sum_{c=1}^{C} N_{\mathcal{X}}^{(c)}\mathbf{T}^{(c)^{\mathrm{T}}}\boldsymbol{\Sigma}^{(c)^{-1}}\mathbf{T}^{(c)} \ , \end{aligned} \tag{3.4}$$

respectively. As in chapter 2, in these equations, $N_{\mathcal{X}}^{(c)}$ (2.7) are the zero–order statistics collected with the UBM for the set of feature vectors in $\mathcal{X}$, $\mathbf{T}^{(c)}$ is the $F \times M$ sub-matrix of $\mathbf{T}$ corresponding to the c–th mixture component such that $\mathbf{T} = \left(\mathbf{T}^{(1)\mathrm{T}}, \ldots, \mathbf{T}^{(C)\mathrm{T}}\right)^{\mathrm{T}}$, and $\tilde{\mathbf{f}}_{\mathcal{X}}$ is the supervector stacking the first–order statistics $\tilde{\mathbf{f}}_{\mathcal{X}}^{(c)}$, centered (see (2.10)) around the corresponding UBM means, $\boldsymbol{\Sigma}^{(c)}$ is the UBM c–th covariance matrix, $\boldsymbol{\Sigma}$ is a block diagonal matrix composed of matrices $\boldsymbol{\Sigma}^{(c)}$, and $\gamma_t^{(c)}$ is the occupation probability of feature vector $\mathbf{x}_t$ for the c-th Gaussian component.

The i-vector $\boldsymbol{\phi}$ – a low dimensional fixed-length vector, which represents the segment $\mathcal{X}$ of a variable length, is then computed as the MAP point estimate of the variable $\mathbf{W}$, i.e., the mean of the posterior distribution $P_{\mathbf{W}|\mathcal{X}}(\mathbf{w})$.

A Maximum-Likelihood estimate of matrix $\mathbf{T}$ can be obtained by following the steps from Section 2.2.1. Each submatrix $\mathbf{T}^c$ can be re-estimated as in (2.38):

$$\mathbf{T}^c = \sum_{\mathcal{X}} \left[\tilde{\mathbf{f}}_{\mathcal{X}}^{(c)}\boldsymbol{\phi}_{\mathcal{X}}(N_{\mathcal{X}}^{(c)}(\boldsymbol{\phi}_{\mathcal{X}}\boldsymbol{\phi}_{\mathcal{X}}^{T} + \boldsymbol{\Gamma}_{\mathcal{X}}^{-1})^{-1}\right] . \tag{3.5}$$

Note that the we do not require any speaker labels and the $\mathbf{T}$ matrix is trained in an unsupervised way. The GMM subspace framework is then used as a feature extractor of the low-dimensional vectors containing most of the relevant variability from the original data – both useful and harmful for the target classification task. The presence of the unwanted variability in the i–vectors has to be dealt with when using i–vectors as features for classifiers or when using i–vectors directly for scoring.

3.2 Probabilistic Linear Discriminant Analysis

All PLDA models for speaker recognition [Kenny, 2010, Brümmer and de Villiers, 2010] represent the speaker identity in terms of a latent variable $\mathbf{Y}$ which is assumed to be tied across all segments of the same speaker. Usually, inter–speaker variability for a speech segment $\mathcal{X}_i$ is represented by hidden variable $\mathbf{X}_i$. The hidden variables $\mathbf{X}_i$ are assumed to be i.i.d. with respect to the speech segments.

In the most common PLDA model, an i–vector $\boldsymbol{\phi}$ is the sum of multiple terms [Kenny, 2010]:

$$\boldsymbol{\phi} = \mathbf{m} + \mathbf{U}\mathbf{y} + \mathbf{V}\mathbf{x} + \mathbf{e} \tag{3.6}$$

where $\mathbf{m}$ is the i–vector mean, $\mathbf{y}$ is a realization of the speaker identity variable $\mathbf{Y}$, $\mathbf{x}$ is the realization of channel variable $\mathbf{X}$ and $\mathbf{e}$ is the realization of the residual noise $\mathbf{E}$.

The role of matrices $\mathbf{U}$ and $\mathbf{V}$ is to constrain the dimension of the sub–spaces for $\mathbf{y}$ and $\mathbf{x}$, providing the bases for a speaker subspace, often called "eigenvoices“ and bases for a channel subspace, usually called "eigenchannels". In this work, we will assume standard normal priors for the speaker identity variable $\mathbf{Y}$ and channel variable $\mathbf{X}$. The noise $\mathbf{E}$ is assumed to be Gaussian distributed with the diagonal covariance matrix of the residual data variability $\mathbf{D}^{-1}$:

$$\mathbf{Y} \sim \mathcal{N}(0, \mathbf{I}) \tag{3.7}$$

$$\mathbf{X} \sim \mathcal{N}(0, \mathbf{I}) \tag{3.8}$$

$$\mathbf{E} \sim \mathcal{N}(0, \mathbf{D}^{-1}). \tag{3.9}$$

In case of this PLDA model, an across-class covariance matrix is defined as $\mathbf{\Sigma}_{ac} = \mathbf{U}^T\mathbf{U}$, which is often low rank and limits the speaker variability to live in a subspace spanned by the columns of the reduced rank matrix $\mathbf{U}$. Similarly, a within-class covariance matrix is defined as $\mathbf{\Sigma}_{wc} = \mathbf{V}^T\mathbf{V} + \mathbf{D}^{-1}$.

3.3 Trial scoring

Given the sets of enrollment and test segments forming a speaker verification trial, we obtain a speaker verification score. In this section, we will define the score as a log-likelihood ratio between the hypotheses that all of the segments were generated by the same speaker and that each set of segments was generated independently by a different speaker.

Since i–vectors are assumed independent given the hidden variables, the likelihood that a set of n speech segments $\mathcal{X}_1 \dots \mathcal{X}_n$ belongs to the same speaker (hypothesis H_s) can be evaluated as:

$$\begin{aligned} l\left(\mathcal{X}_1 \dots \mathcal{X}_n | H_s\right) &= P_{\mathbf{\Phi}_1 \dots \mathbf{\Phi}_n}(\boldsymbol{\phi}_1 \dots \boldsymbol{\phi}_n | H_s) \\ &= \int_{\mathbf{y}} \int_{\mathbf{x}_1} \cdots \int_{\mathbf{x}_n} \prod_{i=1}^{n} \left[P_{\mathbf{\Phi}_i|\mathbf{Y},\mathbf{X}_i}\left(\boldsymbol{\phi}_i|\mathbf{y},\mathbf{x}_i\right) P_{\mathbf{X}_i}\left(\mathbf{x}_i\right) d\mathbf{x}_i \right] \cdot P_{\mathbf{Y}}(\mathbf{y}) \mathrm{d}\mathbf{y} \ , \end{aligned} \tag{3.10}$$

where $\boldsymbol{\phi}_i$ is the i–vector extracted from segment $\mathcal{X}_i$, $P_{\mathbf{\Phi}_1 \dots \mathbf{\Phi}_n|H_s}(\boldsymbol{\phi}_1 \dots \boldsymbol{\phi}_n)$ is the joint probability of the i–vectors given the same speaker hypothesis H_s, $P_{\mathbf{X}}(\mathbf{x})$ and $P_{\mathbf{Y}}(\mathbf{y})$ are the prior distributions for $\mathbf{X}$ and $\mathbf{Y}$, respectively. $P_{\mathbf{\Phi}|\mathbf{Y},\mathbf{X}}\left(\boldsymbol{\phi}|\mathbf{y},\mathbf{x}\right)$ is the conditional distribution of an i–vector given the hidden variables. It is related to the distribution $P_{\mathbf{E}}(\mathbf{e})$ of the noise term by $P_{\mathbf{\Phi}|\mathbf{Y},\mathbf{X}}\left(\boldsymbol{\phi}|\mathbf{y},\mathbf{x}\right) = P_{\mathbf{E}}(\boldsymbol{\phi} - \mathbf{m} - \mathbf{U}\mathbf{y} - \mathbf{V}\mathbf{x})$.

In order to obtain an inference about the speaker identity, we ask the question, whether a set of n enrollment segments $\mathcal{X}_{e_1} \dots \mathcal{X}_{e_n}$ for a known (target) speaker and a set of m test segments of a single unknown speaker $\mathcal{X}_{t_1} \dots \mathcal{X}_{t_m}$ belong to the same speaker or not. Specifically, we want to compute the log-likelihood ratio of the segments being observed under the same speaker and different speaker hypotheses

$$s = \log \frac{l\left(\mathcal{X}_{e_1} \dots \mathcal{X}_{e_n}, \mathcal{X}_{t_1} \dots \mathcal{X}_{t_m} | H_s\right)}{l\left(\mathcal{X}_{e_1} \dots \mathcal{X}_{e_n}, \mathcal{X}_{t_1} \dots \mathcal{X}_{t_m} | H_d\right)} \ . \tag{3.11}$$

Since speaker factors are assumed independent, the speaker verification log–likelihood ratio s can be formulated as:

$$s = \log \frac{l\left(\mathcal{X}_{e_1} \dots \mathcal{X}_{e_n}, \mathcal{X}_{t_1} \dots \mathcal{X}_{t_m} | H_s\right)}{l\left(\mathcal{X}_{e_1} \dots \mathcal{X}_{e_n} | H_s\right) l\left(\mathcal{X}_{t_1} \dots \mathcal{X}_{t_m} | H_s\right)} \ . \tag{3.12}$$

It is worth noting, that the log-likelihood ratio calculated in this way is symmetric in terms of swapping the enroll and test sets. Also note that standard i–vector, which is extracted by MAP point estimate of the posterior distribution of $\mathbf{W}$ given $\mathcal{X}$, and classified by PLDA, does not embed the intrinsic uncertainty of its estimate. We will address this fact in the next chapter, where we will extend the PLDA model and no longer consider the segment $\mathcal{X}$ being represented by a single i–vector, but to the i–vector distribution $\mathbf{W}|\mathcal{X}$.

3.4 Simplified PLDA Model

It is convenient to assume that the noise term $\mathbf{E}$ has a full covariance matrix, so that the terms $\mathbf{Vx}$ and $\mathbf{e}$ in (3.6) can be merged. Therefore, in our approach a distribution of i–vector $\boldsymbol{\phi}$ is modeled as:

$$\boldsymbol{\phi} = \mathbf{m} + \mathbf{Uy} + \mathbf{e} \ . \tag{3.13}$$

In this model, we restrict only the speaker variability to reside in the subspace spanned by the reduced rank matrix $\mathbf{U}$. The across class covariance matrix is again defined as $\boldsymbol{\Sigma}_{ac} = \mathbf{U^T U}$. Channel variability is then modeled by a full rank within class covariance matrix $\boldsymbol{\Sigma}_{wc} = \boldsymbol{\Lambda}^{-1}$. Speaker factors and the residual noise priors are assumed to be Gaussian, i.e.:

$$\mathbf{Y} \sim \mathcal{N}(0, \mathbf{I}) \ , \quad \mathbf{E} \sim \mathcal{N}(0, \boldsymbol{\Lambda}^{-1}) \ , \tag{3.14}$$

where $\boldsymbol{\Lambda}$ is the precision matrix of noise $\mathbf{E}$. According to (3.13) and (3.14), the conditional distribution of an i–vector random variable $\boldsymbol{\Phi}$ given a value $\mathbf{y}$ for the speaker identity $\mathbf{Y}$ is:

$$\boldsymbol{\Phi} | \left(\mathbf{Y} = \mathbf{y}\right) \sim \mathcal{N}(\mathbf{m} + \mathbf{Uy}, \boldsymbol{\Lambda}^{-1}) \ . \tag{3.15}$$

Omitting the channel factors, which in our model are now embedded in the noise term, the likelihood that the n speech segments $\mathcal{X}_1 \dots \mathcal{X}_n$ belong to the same speaker can be computed by means of a simplified expression of (3.10) as:

$$\begin{aligned} l(\mathcal{X}_1 \dots \mathcal{X}_n | H_s) &= P_{\boldsymbol{\Phi}_1 \dots \boldsymbol{\Phi}_n}(\boldsymbol{\phi}_1 \dots \boldsymbol{\phi}_n | H_s) \\ &= \int_{\mathbf{y}} \prod_{i=1}^{n} P_{\boldsymbol{\Phi}_i | \mathbf{Y}}(\boldsymbol{\phi}_i | \mathbf{y}) P_{\mathbf{Y}}(\mathbf{y}) \mathrm{d}\mathbf{y} \ . \end{aligned} \tag{3.16}$$

3.4.1 Closed-Form Solution for Scoring

In order to compute the likelihood of a set of n i–vectors $\boldsymbol{\phi}_1 \dots \boldsymbol{\phi}_n$ (or corresponding speech segments $\mathcal{X}_1 \dots \mathcal{X}_n$, we observe that the joint log-likelihood of the i–vectors and the hidden variables is:

$$\begin{aligned} &\log P_{\boldsymbol{\Phi}_1 \dots \boldsymbol{\Phi}_n, \mathbf{Y}}(\boldsymbol{\phi}_1 \dots \boldsymbol{\phi}_n, \mathbf{y} | H_s) = \sum_{i=1}^{n} \log P_{\boldsymbol{\Phi} | \mathbf{Y}}(\boldsymbol{\phi}_i | \mathbf{y}) + \log P_{\mathbf{Y}}(\mathbf{y}) \\ &= \sum_{i=1}^{n} \left[-\frac{1}{2} (\boldsymbol{\phi}_i - \mathbf{m} - \mathbf{Uy})^{\mathrm{T}} \boldsymbol{\Lambda} \left(\boldsymbol{\phi}_i - \mathbf{m} - \mathbf{Uy}\right) \right] + \frac{1}{2} \mathbf{y}^{\mathrm{T}} \mathbf{y} + k \ , \end{aligned} \tag{3.17}$$

where k is a constant collecting the terms that do not depend on speaker identity $\mathbf{y}$. Since equation 3.17 is a quadratic function, using “completion of squares”, we can observe that the posterior distribution of $\mathbf{Y}$ given a set of i–vectors is Gaussian

$$\mathbf{Y} | \boldsymbol{\Phi}_1 \dots \boldsymbol{\Phi}_n \sim \mathcal{N}(\hat{\mathbf{y}}, \mathbf{P}^{-1}), \tag{3.18}$$

with precision matrix and mean:

$$\begin{aligned} \mathbf{P} &= \mathbf{I} + \mathbf{U}^{\mathrm{T}} \boldsymbol{\Lambda} \mathbf{U} \\ \hat{\mathbf{y}} &= \mathbf{P}^{-1} \mathbf{U}^{\mathrm{T}} \sum_{i=1}^{n} \boldsymbol{\Lambda} \left(\boldsymbol{\phi}_i - \mathbf{m}\right) \ . \end{aligned} \tag{3.19}$$

The likelihood that a set of segments belongs to the same speaker can be written as:

$$P_{\mathbf{\Phi}_1 \dots \mathbf{\Phi}_n}(\boldsymbol{\phi}_1 \dots \boldsymbol{\phi}_n | H_s) = \frac{P(\boldsymbol{\phi}_1 \dots \boldsymbol{\phi}_n | \mathbf{y}_0) P(\mathbf{y}_0)}{P(\mathbf{y}_0 | \boldsymbol{\phi}_1 \dots \boldsymbol{\phi}_n)} \ , \tag{3.20}$$

where $\mathbf{y}_0$ is an arbitrary vector, which does not cause the denominator to be zero. For the convenience, we can set the $\mathbf{y}_0 = \mathbf{0}$, so that $\mathbf{U}\mathbf{y}_0 = 0$ and derive a closed form solution for the same speaker hypothesis [Brümmer and de Villiers, 2010]:

$$\begin{aligned} \log P_{\mathbf{\Phi}_1 \dots \mathbf{\Phi}_n}(\boldsymbol{\phi}_1 \dots \boldsymbol{\phi}_n | H_s) = & \sum_{i=1}^{n} \left[\frac{1}{2} \log |\mathbf{\Lambda}| - \frac{M}{2} \log 2\pi - \frac{1}{2} (\boldsymbol{\phi}_i - \mathbf{m})^{\mathrm{T}} \mathbf{\Lambda} (\boldsymbol{\phi}_i - \mathbf{m}) \right] \\ & - \frac{1}{2} \log |\mathbf{P}| + \frac{1}{2} \hat{\mathbf{y}}^{\mathrm{T}} \mathbf{P} \hat{\mathbf{y}} - \frac{S}{2} \log 2\pi \ , \end{aligned} \tag{3.21}$$

where M is the i–vector dimension, and S is the speaker factor dimension.

Chapter 4

Full Posterior Distribution PLDA Model

In this chapter, we will demonstrate, how to extend the standard PLDA model, where we considered the utterance to be sufficiently well represented by a single i–vector. We will show that the simple and effective PLDA framework can still be used even if a speech segment is no more represented by a single i–vector but by its posterior distribution. In particular, we will derive the formulation of likelihood for a standard Gaussian PLDA model based on the i–vector posterior distribution, and propose a new PLDA model where the inter–speaker variability is assumed to have an utterance–dependent distribution. We will show that it is possible to rely on the standard PLDA framework simply replacing the PLDA likelihood definition.

It is well known, that the goodness of the i–vector estimate depends mainly on the covariance of the distribution, which accounts for the "uncertainty" of the i–vector extraction process. This ucertainity of the i–vector estimate is however not exploited by many standard and popular classifiers based on i–vectors, such as the ones based on cosine distance scoring [Dehak et al., 2010], PLDA [Kenny, 2010], discriminative PLDA [Burget et al., 2011] or SVMs [Cumani et al., 2013].

The i–vector covariance depends on the zero–order statistics estimated using a UBM for the set of observed features (see equation (3.4) in Chapter 3.1). These statistics are affected by several factors such as the noise level, the channel characteristics, and the acoustic content of the observed features, but the predominant factor is the number of the observed feature frames – duration of a given utterance. Shorter utterances tend to produce larger covariances, so that i–vector estimates become less reliable.

4.1 Incorporating the I–vector Posterior Distribution into PLDA

The standard i–vector, which is extracted by MAP point estimate of the posterior distribution of $\mathbf{W}$ given $\mathcal{X}$ does not embed the intrinsic uncertainty of its estimate. Remembering the likelihood computation for the standard PLDA (see 3.10), we can extend this model by considering all possible i–vectors, which correspond to the speech segments $\mathcal{X}_1 \ldots \mathcal{X}_n$.

We refer to this new model as the PLDA based on the "Full Posterior Distribution" (FPD–PLDA) of $\mathbf{W}$ given $\mathcal{X}$. As previously mentioned, we now assume that every segment $\mathcal{X}$ is no represented by a single i–vector corresponding to the most likely value of the latent variable $\mathbf{w}$ in the i–vector model (3.1). Instead segment $\mathcal{X}$ will be represented by the i–vector extractor distribution $\mathbf{W}|\mathcal{X}$ (see (3.3)). Therefore, the uncertainty in i–vector estimate will be taken into account. In the following text, we will refer to the posterior distribution $\mathbf{W}|\mathcal{X}$ simply as to i–vector posterior distribution.

The PLDA model allows computing the likelihood of a speech segment given a realization $\mathbf{w}$ of the random variable $\mathbf{W}|\mathcal{X}$. The likelihood of a set of segments $\mathcal{X}_1 \dots \mathcal{X}_n$, thus, can be evaluated by integrating the PLDA likelihood (see equations 3.10 and 3.17) over all possible realizations following the posterior distribution $\mathbf{W}|\mathcal{X}_1 \dots \mathcal{X}_n$.

$$l\left(\mathcal{X}_1 \dots \mathcal{X}_n | H_s\right) = \int_{\mathbf{w}_1} \cdots \int_{\mathbf{w}_n} P_{\mathbf{W}_1 \dots \mathbf{W}_n}\left(\mathbf{w}_1 \dots \mathbf{w}_n | \, H_s\right) \prod_{i=1}^{n} \left[P_{\mathbf{W}_i|\mathcal{X}_i}(\mathbf{w}_i) \mathrm{d}\mathbf{w}_i \right], \tag{4.1}$$

where the first factor is the likelihood of the segments according to the original PLDA model given realizations $\mathbf{w}_1, \dots, \mathbf{w}_n$ of the i–vector posterior random variables, computed as in (3.10), and the second factor is the posterior probability of realizations $\mathbf{w}_1, \dots, \mathbf{w}_n$ representing segments $\mathcal{X}_1 \dots \mathcal{X}_n$ according to the i–vector extractor model. Using the form of (3.10) in (4.1), the likelihood can be rewritten as:

$$\begin{aligned} l\left(\mathcal{X}_1 \dots \mathcal{X}_n | H_s\right) = \int_{\mathbf{w}_1} \cdots \int_{\mathbf{w}_n} \int_{\mathbf{y}} \int_{\mathbf{x}_1} \cdots \int_{\mathbf{x}_n} \prod_{i=1}^{n} \Big[& P_{\mathbf{W}_i|\mathbf{Y},\mathbf{X}_i}\left(\mathbf{w}_i | \mathbf{y}, \mathbf{x}_i\right) \\ & \cdot P_{\mathbf{X}_i}(\mathbf{x}_i) P_{\mathbf{W}_i|\mathcal{X}_i}\left(\mathbf{w}_i\right) \mathrm{d}\mathbf{x}_i \mathrm{d}\mathbf{w}_i \Big] P_{\mathbf{Y}}(\mathbf{y}) \mathrm{d}\mathbf{y} \; . \end{aligned} \tag{4.2}$$

It is worth noting that, if the posterior for $\mathbf{W}|\mathcal{X}$ is replaced by a delta distribution centered in the posterior mean $\delta(\boldsymbol{\phi}_{\mathcal{X}})$, the likelihood of the original PLDA model using MAP–estimated i–vectors, given by (3.10), is obtained.

4.2 Extending the Classical Simplified PLDA

We will continue with the derivations using the simplified PLDA model introduced in the previous chapter (3.4). Starting from the point where we introduced the likelihood of a set of segments given the same speaker hypothesis in 3.16, we introduce the full i–vector posterior into the equation and we get:

$$\begin{aligned} l\left(\mathcal{X}_1 \dots \mathcal{X}_n | H_s\right) &= \int_{\mathbf{w}_i} \cdots \int_{\mathbf{w}_n} \int_{\mathbf{y}} P_{\mathbf{Y}}(\mathbf{y}) \cdot \prod_{i=1}^{n} \left[P_{\mathbf{W}_i|\mathbf{Y}}(\mathbf{w}_i|\mathbf{y}) P_{\mathbf{W}_i|\mathcal{X}_i}(\mathbf{w}_i) \mathrm{d}\mathbf{w}_i \right] \mathrm{d}\mathbf{y} \\ &= \int_{\mathbf{y}} P_{\mathbf{Y}}(\mathbf{y}) \prod_{i=1}^{n} \left[\int_{\mathbf{w}_i} P_{\mathbf{W}_i|\mathbf{Y}}(\mathbf{w}_i|\mathbf{y}) P_{\mathbf{W}_i|\mathcal{X}_i}(\mathbf{w}_i) \mathrm{d}\mathbf{w}_i \right] \mathrm{d}\mathbf{y} \; , \end{aligned} \tag{4.3}$$

According to the Gaussian assumptions given in (3.3) and (3.14), the inner integral can be computed as

$$\begin{aligned} & \int_{\mathbf{w}_i} P_{\mathbf{W}_i|\mathbf{Y}}(\mathbf{w}_i|\mathbf{y}) P_{\mathbf{W}_i|\mathcal{X}_i}(\mathbf{w}_i) \mathrm{d}\mathbf{w}_i = \\ & \int_{\mathbf{w}_i} \frac{1}{(2\pi)^{\frac{M}{2}} \left|\boldsymbol{\Lambda}^{-1}\right|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{w}_i - \mathbf{m} - \mathbf{U}\mathbf{y})^{\mathrm{T}} \boldsymbol{\Lambda} (\mathbf{w}_i - \mathbf{m} - \mathbf{U}\mathbf{y})} \\ & \cdot \frac{1}{(2\pi)^{\frac{M}{2}} \left|\boldsymbol{\Gamma}_i^{-1}\right|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{w}_i - \boldsymbol{\phi}_i)^{\mathrm{T}} \boldsymbol{\Gamma}_i (\mathbf{w}_i - \boldsymbol{\phi}_i)} \mathrm{d}\mathbf{w}_i \; , \end{aligned} \tag{4.4}$$

where $\boldsymbol{\phi}_i$ and $\boldsymbol{\Gamma}_i$ are the mean and precision matrix of $\mathbf{W}_i|\mathcal{X}_i$ computed as in (3.4). Integral (4.4) can be interpreted as the convolution of two Gaussian distributions, leading to:

$$l(\mathcal{X}_1 \ldots \mathcal{X}_n|\mathbf{Y}=\mathbf{y}) = \prod_{i=1}^{n} \frac{1}{(2\pi)^{\frac{M}{2}} \left|\boldsymbol{\Lambda}^{-1} + \boldsymbol{\Gamma}_i^{-1}\right|^{\frac{1}{2}}} \cdot e^{(\boldsymbol{\phi}_i - \mathbf{m} - \mathbf{U}\mathbf{y})^{\mathrm{T}} \left(\boldsymbol{\Lambda}^{-1} + \boldsymbol{\Gamma}_i^{-1}\right)^{-1} (\boldsymbol{\phi}_i - \mathbf{m} - \mathbf{U}\mathbf{y})} . \tag{4.5}$$

Comparing (4.5) and (3.17), we can see that now the covariance matrix of noise becomes segment-dependent as $[\boldsymbol{\Lambda}^{-1} + \boldsymbol{\Gamma}^{-1}]$. Considering the similarity of both models, we can say that the FPD-PLDA can be equivalently represented (likelihood calculation can be “similated”) by the standard PLDA modeling the usual i–vectors (i.e. i–vector posterior means), while assuming modified utterance dependent prior imposed on residual noise

$$\overline{\mathbf{E}}_i \sim \mathcal{N}\left(\mathbf{0}, \left[\boldsymbol{\Lambda}^{-1} + \boldsymbol{\Gamma}_i^{-1}\right]\right) . \tag{4.6}$$

4.3 Scoring with FPD-PLDA

The log–likelihood that a set of segments belongs to the same speaker can be obtained by means of the same steps followed for the standard Gaussian PLDA model, just using the modified likelihood in (4.5). The new PLDA model can be described as:

$$\boldsymbol{\phi} = \mathbf{m} + \mathbf{U}\mathbf{y} + \overline{\mathbf{e}} \ , \tag{4.7}$$

as in (3.13), but with an segment–dependent distribution of the residual noise $\overline{\mathbf{E}}$. The i–vector associated to speech segment $\mathcal{X}_i$ is again the mean $\boldsymbol{\phi}_i$ of the i–vector posterior $\mathbf{W}_i|\mathcal{X}_i$, but the priors of the PLDA parameters are given by:

$$\overline{\mathbf{E}}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda}^{-1} + \boldsymbol{\Gamma}_i^{-1}) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda}_{eq,i}^{-1}) \ , \mathbf{Y} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \ , \tag{4.8}$$

where

$$\boldsymbol{\Lambda}_{eq,i} = \left(\boldsymbol{\Lambda}^{-1} + \boldsymbol{\Gamma}_i^{-1}\right)^{-1} . \tag{4.9}$$

In the following text, to simplify the notation, we will refer to distributions without explicitly naming the corresponding hidden variable, e.g., we will write $P(\mathbf{y})$ rather than $P_{\mathbf{Y}}(\mathbf{y})$.

To compute the likelihood of a set of n i–vectors $\boldsymbol{\phi}_1 \ldots \boldsymbol{\phi}_n$ (i.e., of the set of speech segments $\mathcal{X}_1 \ldots \mathcal{X}_n$), we follow the same steps as in the previous section on the standard PLDA. Similarly to 3.17, we observe that the joint log–likelihood of the i–vectors and the hidden variables is:

$$\begin{aligned} \log P(\boldsymbol{\phi}_1 \ldots \boldsymbol{\phi}_n, \mathbf{y}|H_s) &= \sum_{i=1}^{n} \log P(\boldsymbol{\phi}_i|\mathbf{y}) + \log P(\mathbf{y}) \\ &= \sum_{i=1}^{n} \left[-\frac{1}{2} (\boldsymbol{\phi}_i - \mathbf{m} - \mathbf{U}\mathbf{y})^{\mathrm{T}} \boldsymbol{\Lambda}_{eq,i} \left(\boldsymbol{\phi}_i - \mathbf{m} - \mathbf{U}\mathbf{y}\right) \right] \\ &+ \frac{1}{2} \mathbf{y}^{\mathrm{T}} \mathbf{y} + k \ , \end{aligned} \tag{4.10}$$

The posterior distribution of $\mathbf{y}$ given a set of i–vectors is again Gaussian:

$$\mathbf{y}|\boldsymbol{\phi}_1 \ldots \boldsymbol{\phi}_n \sim \mathcal{N}(\hat{\mathbf{y}}, \mathbf{P}^{-1}), \tag{4.11}$$

with parameters:

$$\mathbf{P} = \mathbf{I} + \sum_{i=1}^{n} \mathbf{U}^{\mathrm{T}} \boldsymbol{\Lambda}_{eq,i} \mathbf{U} \tag{4.12}$$

$$\hat{\mathbf{y}} = \mathbf{P}^{-1} \mathbf{U}^{\mathrm{T}} \sum_{i=1}^{n} \boldsymbol{\Lambda}_{eq,i} \left(\boldsymbol{\phi}_i - \mathbf{m} \right) \ . \tag{4.13}$$

The likelihood of a set of segments belonging to the same speaker can be written as

$$P(\boldsymbol{\phi}_1 \dots \boldsymbol{\phi}_n | H_s) = \frac{P(\boldsymbol{\phi}_1 \dots \boldsymbol{\phi}_n | \mathbf{y}_0) P(\mathbf{y}_0)}{P(\mathbf{y}_0 | \boldsymbol{\phi}_1 \dots \boldsymbol{\phi}_n)} \ , \tag{4.14}$$

which is the same form as in the original PLDA and setting $\mathbf{y}_0 = \mathbf{0}$ for the convenience will produce the similar equation to (3.21). Using (4.11), and (4.5) we finally get

$$\begin{aligned} &\log P(\boldsymbol{\phi}_1 \dots \boldsymbol{\phi}_n | H_s) = \\ &\sum_{i=1}^{n} \left[\frac{1}{2} \log |\boldsymbol{\Lambda}_{eq,i}| - \frac{M}{2} \log 2\pi - \frac{1}{2} (\boldsymbol{\phi}_i - \mathbf{m})^{\mathrm{T}} \boldsymbol{\Lambda}_{eq,i} (\boldsymbol{\phi}_i - \mathbf{m}) \right] \\ &- \frac{1}{2} \log |\mathbf{P}| + \frac{1}{2} \hat{\mathbf{y}}^{\mathrm{T}} \mathbf{P} \hat{\mathbf{y}} - \frac{S}{2} \log 2\pi \ , \end{aligned} \tag{4.15}$$

where M is the i–vector dimension, and S is the speaker factor dimension. Again that the difference to the standard PLDA lies in the segment-based $\Lambda_{eq,i}$, which greatly affect the computational complexity of scoring.

4.4 Parameter Estimation

The model presented in (4.7) allows obtaining a simple expression for computing the log–likelihood ratio of a speaker recognition trial. However, it does not allow the update formulas to be easily derived. An equivalent expression of (4.7), where the contributions of the i–vector posterior covariance and of the residual noise are decoupled, is more suitable for the estimation of model parameters [Kenny et al., 2013]. To this extent, the segment–dependent residual term $\overline{\mathbf{E}}_i$ can be written as:

$$\overline{\mathbf{E}}_i = \mathbf{C}_i \mathbf{X}_i + \mathbf{E} \ , \tag{4.16}$$

where $\mathbf{C}_i$ is is given by the Cholesky decomposition $\mathbf{C}_i \mathbf{C}_i^{\mathrm{T}} = \boldsymbol{\Gamma}_i^{-1}$, $\mathbf{X}_i$ is a standard Gaussian distributed random variable, $\mathbf{X}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and $\mathbf{E}$ is the PLDA residual term introduced in (3.14). The corresponding PLDA model is then given by:

$$\boldsymbol{\phi}_i = \mathbf{m} + \mathbf{U}\mathbf{y} + \mathbf{C}_i \mathbf{x}_i + \mathbf{e}_i \ , \tag{4.17}$$

where $\mathbf{x}_i$ is a realization of $\mathbf{X}_i$. It is worth noting that (4.17) formally corresponds to the PLDA model in (3.6) with the channel sub–space matrix $\mathbf{V}$ replaced by a segment–dependent matrix $\mathbf{C}_i$. The same steps to derive the EM algorithm for the PLDA model (3.6) can be easily modified to estimate the parameters of the FPD–PLDA model. The details of the derivation of the EM algorithm can be found in [Kenny et al., 2013] or [Brümmer, 2010] with modifications related to this model.

4.5 I–vector Pre-Processing

We assume that i–vectors are standard-normal distributed and both speaker and channel effects modeled by the Gaussian PLDA are additive, statistically independent and normally distributed. In [Kenny, 2010], Patrick Kenny clearly demonstrated that these assumptions are not satisfied, which leads to a sub-optimal performance of the model. Additionally the score normalization was needed (s-norm) to obtain better results contradicting the intuition that a good generative model should produce well calibrated likelihood ratios which do not need to be further normalized.

A simple method of normalizing i–vectors to suit the Gaussian PLDA model was introduced in [Garcia-Romero, 2011]. The normalization generally consists of two steps: data whitening and length normalization. Whitening is the process where we enforce the total covariance matrix of i–vectors to be identity. The whitening can be performed as

$$\boldsymbol{\phi}_{\text{wht}} = \mathbf{D}^{1/2}\mathbf{E}^{\mathrm{T}}\boldsymbol{\phi}, \tag{4.18}$$

where $\mathbf{E}$ and $\mathbf{D}$ are the orthogonal matrix of eigenvectors (in columns of $\mathbf{E}$) and diagonal matrix of eigenvalues of the total covariance matrix estimated on training i–vectors, respectively. Length normalization is a nonlinear transformation where we divide each i–vector by its norm and transform it to a vector of unit length:

$$\boldsymbol{\phi}_{\text{norm}} = \frac{\boldsymbol{\phi}}{\|\boldsymbol{\phi}\|}. \tag{4.19}$$

4.5.1 Length Normalization

Performing a transformation of the data into the unit length indeed again violates the Gaussian assumptions as the samples drawn from the high-dimensional standard normal Gaussians lie far away from the unit sphere. In fact, the samples are mostly present in a thin shell of a multidimensional sphere, of which distance from the origin is increasing with the dimensionality of data. If we are considering 600−dimensional i–vectors and knowing that the distribution of lengths of standard-normal distributed i–vectors follows Chi distribution, inner radius would be approximately 24 (see the mode of the Chi distribution in Figure 4.1).

When comparing the actual lengths of the i–vectors extracted from the training data and held out evaluation data, we observe completely different distributions of the lengths. In Figure 4.1, we present a situation of the i–vectors extracted for the Domain Adaptation Challenge [MITLL, 2103]. There are three different datasets (training, adaptation and evaluation set) used in the Adaptation Challenge coming from various LDC data collections. The training set consists of all telephone calls from the all speakers taken from Switchboard-I and Switchboard-II (all phases) corporas. The adaptation set is composed of all telephone calls from all speakers taken from the NIST SRE data collections from years 2004, 2005, 2006 and 2008. Finally, the evaluation set is the telephone data from NIST SRE 2010 evaluations.

Not only we can observe a considerable shift in the lengths distributions of the individual databases, but all distributions have a longer right tail. The PDF of Chi distribution with 600 degrees of freedom representing the distribution of 600 dimensional standard normal distributed vectors is depicted in black color. As the i–vector extractor was trained on the training data, the i–vector length distribution of this dataset is closest to the expected distribution.

These shifts between datasets indeed lead to problems. As pointed out in [Garcia-Romero, 2011], the shift in the i–vector lengths would introduce a global scaling in the obtained scores (see equations 3.21 or 5.10). Scaling could be partly recovered by means of the linear calibration. However, especially in the cases, when the evaluation data are composed

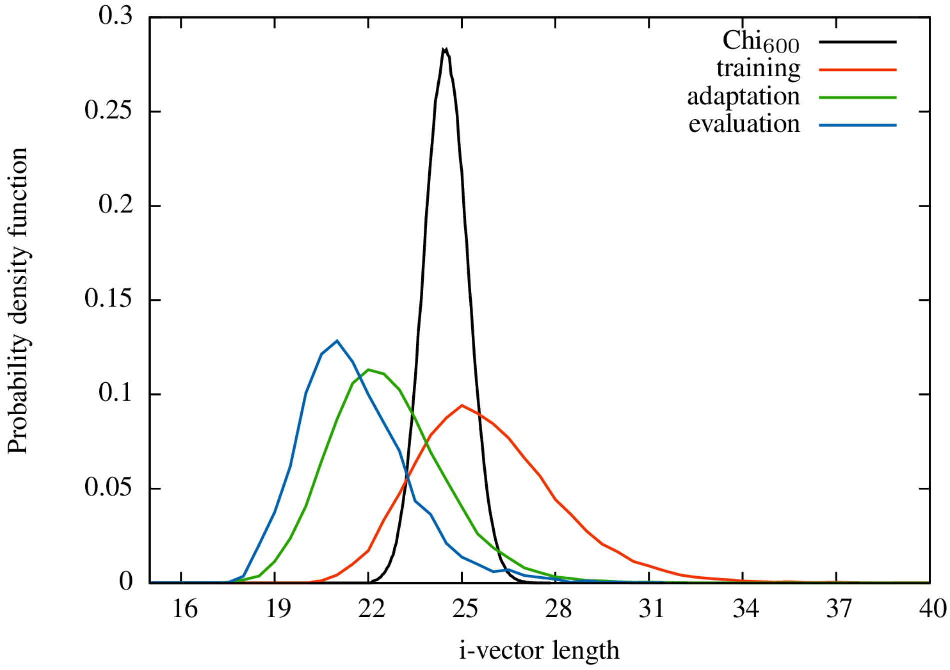

Figure 4.1: Histograms of the i–vector length distributions of three sets of Domain Adaptation Challenge. A probability density function of Chi distribution with 600 degrees of freedom depicted in black represents the distribution of 600 dimensional standard normal distributed vectors.

of recordings coming from different sources, there would be more such scalings and one global calibration would not be sufficient to overcome this problem.

By performing normalization to unit length, we place all i–vectors on a surface of a common unit sphere and effectively greatly compress all distances between them. Also we replace a distribution of their lengths by a constant. With a proper scaling, the constant could be even set into the mode of the Chi distribution, which in the end is not necessary. This way, we made the distribution of the i–vector lengths closer to the distribution of lengths of the i–vectors following standard-normal distribution. We also avoided problems with the score scaling. It is important to note, that before actual length normalization, we must ensure that the i–vectors are normalized to zero mean. Although zero mean of the i–vectors is also assumed by the i–vector extraction model, it is often not the case for i–vectors extracted from some held out data. After all of these transformations, the PLDA is trained on normalized i–vectors. Alternatively the cosine scoring can be directly performed.

4.5.2 Application to Full Posterior Distribution

This section presents the length normalization applied to the i–vector posterior distribution. A straightforward approach is to replace the i–vector distribution $\mathbf{W}|\mathcal{X}$ by $\widehat{\mathbf{W}} = \frac{\mathbf{W}|\mathcal{X}}{\|\mathbf{W}|\mathcal{X}\|}$, which forces all realizations of $\widehat{\mathbf{W}}$ to lie on the unit sphere. However, since the resulting random variable $\widehat{\mathbf{W}}$ would not be Gaussian distributed, it would not be possible to rely on the simple derivations of Section 3.4, and to avoid the higher complexity introduced by the use of a non Gaussian distribution.

Alternatively, the length normalization can be seen as a non–linear transformation $F(\boldsymbol{\phi}_0)$ of the observed i–vector $\boldsymbol{\phi}_0$, which can be approximated by its first order Taylor expansion around the i–vector itself. The expansion is given by:

$$F(\boldsymbol{\phi}) = F(\boldsymbol{\phi}_0) + J_F(\boldsymbol{\phi}_0)(\boldsymbol{\phi} - \boldsymbol{\phi}_0) + o(\|\boldsymbol{\phi} - \boldsymbol{\phi}_0\|) \ , \tag{4.20}$$

where $J_F(\boldsymbol{\phi}_0)$ is the Jacobian of F computed at $\boldsymbol{\phi}_0$ and F is the function $F(\mathbf{x}) = \frac{\mathbf{x}}{\|\mathbf{x}\|}$. The linear transformation which approximates the length normalization function around the i–vector is then:

$$\widehat{F}(\boldsymbol{\phi}) = F(\boldsymbol{\phi}_0) + J_F(\boldsymbol{\phi}_0)(\boldsymbol{\phi} - \boldsymbol{\phi}_0) = \mathbf{v} + \frac{(\mathbf{I} - \mathbf{v}\mathbf{v}^{\mathrm{T}})}{\|\boldsymbol{\phi}_0\|}\boldsymbol{\phi} \tag{4.21}$$

where $\mathbf{v} = \frac{\boldsymbol{\phi}_0}{\|\boldsymbol{\phi}_0\|}$ and $\mathbf{I}$ is the identity matrix.

The extension to the full i–vector posterior consists in computing the first order Taylor expansion of F centered at the posterior distribution mean $\boldsymbol{\phi}_{\mathcal{X}}$, and applying the resulting linear transformation to the i–vector posterior $\mathbf{W}|\mathcal{X} \sim \mathcal{N}(\boldsymbol{\phi}_{\mathcal{X}}, \boldsymbol{\Gamma}_{\mathcal{X}}^{-1})$. The expansion of F around $\boldsymbol{\phi}_{\mathcal{X}}$ is:

$$\widehat{F}(\boldsymbol{\phi}_{\mathcal{X}}) = \mathbf{v}_{\mathcal{X}} + \frac{(\mathbf{I} - \mathbf{v}_{\mathcal{X}}\mathbf{v}_{\mathcal{X}}^{\mathrm{T}})}{\|\boldsymbol{\phi}_{\mathcal{X}}\|}\boldsymbol{\phi}_{\mathcal{X}} = \mathbf{v}_{\mathcal{X}} + \mathbf{A}\boldsymbol{\phi}_{\mathcal{X}} \ , \tag{4.22}$$

where $\mathbf{v}_{\mathcal{X}} = \frac{\boldsymbol{\phi}_{\mathcal{X}}}{\|\boldsymbol{\phi}_{\mathcal{X}}\|}$ and $\mathbf{A} = \frac{(\mathbf{I} - \mathbf{v}_{\mathcal{X}}\mathbf{v}_{\mathcal{X}}^{\mathrm{T}})}{\|\boldsymbol{\phi}_{\mathcal{X}}\|}$. Thus, the transformed distribution is given by:

$$\begin{aligned} \widehat{\mathbf{W}} &\sim \mathcal{N}\left(\widehat{F}(\boldsymbol{\phi}_{\mathcal{X}}), \mathbf{A}\boldsymbol{\Gamma}_{\mathcal{X}}^{-1}\mathbf{A}^{\mathrm{T}}\right) \\ &\sim \mathcal{N}\left(\frac{\boldsymbol{\phi}_{\mathcal{X}}}{\|\boldsymbol{\phi}_{\mathcal{X}}\|}, \frac{1}{\|\boldsymbol{\phi}_{\mathcal{X}}\|^2}(\mathbf{I} - \mathbf{v}_{\mathcal{X}}\mathbf{v}_{\mathcal{X}}^{\mathrm{T}})\boldsymbol{\Gamma}_{\mathcal{X}}^{-1}(\mathbf{I} - \mathbf{v}_{\mathcal{X}}\mathbf{v}_{\mathcal{X}}^{\mathrm{T}})\right) , \end{aligned} \tag{4.23}$$

Expression (4.23) can be further approximated as:

$$\overline{\mathbf{W}} \sim \mathcal{N}\left(\frac{\boldsymbol{\phi}_{\mathcal{X}}}{\|\boldsymbol{\phi}_{\mathcal{X}}\|}, \frac{\boldsymbol{\Gamma}_{\mathcal{X}}^{-1}}{\|\boldsymbol{\phi}_{\mathcal{X}}\|^2}\right) \ . \tag{4.24}$$

In the experimental section, we show that these linearizations of the length normalization are effective. In particular, the approximation (4.24) allows a simplification of (4.23) without incurring in any performance degradation. We will refer to (4.23) as “Projected Length Normalization” (FPD1), and to (4.24) as “Length Normalization” (FPD2).

Chapter 5

Discriminative Training of PLDA

In this chapter, we propose to estimate verification scores using a *discriminative model* rather than a generative PLDA model. More specifically, the speaker verification score for a pair of i-vectors is computed using a function having the functional form derived from the standard PLDA model. The parameters of the function, however, are estimated using a discriminative training criterion. We use an objective function that directly addresses the speaker verification task, i.e. the discrimination between "same-speaker" and "different-speaker" trials. In other words, a binary classifier that takes a pair of i-vectors as an input, is trained to answer the question of whether or not the two i-vectors come from the same speaker. We show that the functional form derived from PLDA can be interpreted as a binary linear classifier in a non-linearly expanded space of i-vector pairs. We have experimented with two discriminative linear classifiers, namely linear support vector machines (SVM) and logistic regression. The advantage of logistic regression is its probabilistic interpretation: the linear output of this classifier can be directly interpreted as the desired log-likelihood ratio verification score. We will concentrate more on training with logistic regression and we will use the abbreviation DPLDA (Discrminative PLDA) for such systems later in Chapter 6.

5.1 Original Model

In order to effectively deploy the discriminative approach to speaker recognition, we need to derive an efficient scheme for obtaining scores for the training examples. We will build our model on previously presented LDA principles and consider a special form of PLDA, a *two-covariance model*, where the simplification is obtained by merging together the residual noise and inter-session components. In this model, both speaker and inter-session variabilities are modeled using across-class and within-class full covariance matrices $\mathbf{\Sigma}_{ac}$ and $\mathbf{\Sigma}_{wc}$. The two-covariance model is a generative linear-Gaussian model, where latent vectors $\mathbf{y}$ representing speakers (or more generally classes) are assumed to be distributed according to prior distribution

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}, \mathbf{\Sigma}_{ac}). \tag{5.1}$$

For a given speaker represented by a vector $\hat{\mathbf{y}}$, the distribution of i-vectors is assumed to be

$$p(\boldsymbol{\phi}|\hat{\mathbf{y}}) = \mathcal{N}(\boldsymbol{\phi}; \hat{\mathbf{y}}, \mathbf{\Sigma}_{wc}). \tag{5.2}$$

The maximum likelihood estimates of the model parameters, $\boldsymbol{\mu}$, $\mathbf{\Sigma}_{ac}$, and $\mathbf{\Sigma}_{wc}$, can be obtained by means of EM algorithm similar to the previous sections. Alternatively, if we want to only obtain a reasonable initialization of the parameters for the discriminative training, the parameters can be directly estimated on the training data as for standard LDA. The training data (i-vectors) come from a database comprising recordings of many speakers (to capture across-class variability), each recorded in several sessions (to capture within-class variability).

5.2 Verification Score of a Trial

To obtain an effective way of scoring, we will consider a trial to be composed only by two i–vectors ($\boldsymbol{\phi}_1$, $\boldsymbol{\phi}_2$). Note, that multi-session scoring, when more i–vectors are available for enroll or test or both, can be easily achieved by averaging the corresponding i–vectors and using the resulting means as single i–vectors. The averaging of i–vectors does not cause any significant problems or deterioration of the performance [Villalba et al., 2013] and in fact is widely used in the community.

We will follow the same steps as in 3.4.1 but with the constraint of a single i–vector per enroll and test part of the evaluation trial. In the case of a same-speaker trial (hypothesis H_s), a single vector $\hat{\mathbf{y}}$ representing a particular speaker is generated from the prior $p(\mathbf{y})$, for which both $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$ are generated from $p(\boldsymbol{\phi}|\hat{\mathbf{y}})$. For a different-speaker trial (hypothesis H_d), two vectors $\hat{\mathbf{y_1}}, \hat{\mathbf{y_2}})$ representing two different speakers are independently generated from $p(\mathbf{y})$. For each, one of the i-vectors $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$ is generated. The speaker verification score can be again calculated as a log-likelihood ratio between the two hypotheses H_s and H_d as

$$s = \log \frac{p(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2 | H_s)}{p(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2 | H_d)}. \tag{5.3}$$

The joint likelihood of the two independent i–vectors being generated from a particular speaker factor $\hat{\mathbf{y}}$ is the product of two likelihoods:

$$p(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2 | \hat{\mathbf{y}}) = p(\boldsymbol{\phi}_1 | \hat{\mathbf{y}})\, p(\boldsymbol{\phi}_2 | \hat{\mathbf{y}}). \tag{5.4}$$

Considering the hypothesis H_s that these two i–vectors can be generated by any speaker common for both of them, we marginalize over all possible speakers:

$$p(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2 | H_s) = \int p(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2 | \mathbf{y})\, p(\mathbf{y}) \mathrm{d}\mathbf{y}. \tag{5.5}$$

For the different speaker hypothesis H_d we again marginalize over all possible speakers and compute the likelihood of the i–vectors being generated independently by any two speakers:

$$\begin{aligned} p(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2 | H_d) &= \int p(\boldsymbol{\phi}_1 | \mathbf{y_1})\, p(\mathbf{y}_1)\, \mathrm{d}\mathbf{y}_1 \int p(\boldsymbol{\phi}_2 | \mathbf{y_2})\, p(\mathbf{y}_2)\, \mathrm{d}\mathbf{y}_2, \\ &= p(\boldsymbol{\phi}_1)\, p(\boldsymbol{\phi}_2). \end{aligned} \tag{5.6}$$

Plugging the conditional likelihoods (5.5) and (5.6) into the log-likelihood ration (5.3) we obtain

$$\begin{aligned} s &= \log \frac{p(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2 | H_s)}{p(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2 | H_d)} && (5.7) \\ &= \log \frac{\int p(\boldsymbol{\phi}_1 | \mathbf{y}) p(\boldsymbol{\phi}_2 | \mathbf{y}) p(\mathbf{y}) d\mathbf{y}}{p(\boldsymbol{\phi}_1) p(\boldsymbol{\phi}_2)}. && (5.8) \end{aligned}$$

The integrals, which can be interpreted as convolutions of Gaussians, can be evaluated analytically giving

$$s = \log \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\phi}_1 \\ \boldsymbol{\phi}_2 \end{bmatrix} ; \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu} \end{bmatrix} , \begin{bmatrix} \boldsymbol{\Sigma}_{tot} & \boldsymbol{\Sigma}_{ac} \\ \boldsymbol{\Sigma}_{ac} & \boldsymbol{\Sigma}_{tot} \end{bmatrix} \right)$$

$$- \quad \log \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\phi}_1 \\ \boldsymbol{\phi}_2 \end{bmatrix}; \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{tot} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{tot} \end{bmatrix} \right), \tag{5.9}$$

where the total covariance matrix is given as $\boldsymbol{\Sigma}_{tot} = \boldsymbol{\Sigma}_{ac} + \boldsymbol{\Sigma}_{wc}$. By expanding the log of Gaussian distributions and simplifying the final expression, we obtain

$$\begin{aligned} s &= \boldsymbol{\phi}_1^T \boldsymbol{\Lambda} \boldsymbol{\phi}_2 + \boldsymbol{\phi}_2^T \boldsymbol{\Lambda} \boldsymbol{\phi}_1 + \boldsymbol{\phi}_1^T \boldsymbol{\Gamma} \boldsymbol{\phi}_1 + \boldsymbol{\phi}_2^T \boldsymbol{\Gamma} \boldsymbol{\phi}_2 \\ &+ (\boldsymbol{\phi}_1 + \boldsymbol{\phi}_2)^T \mathbf{c} + k, \end{aligned} \tag{5.10}$$

where

$$\begin{aligned} \boldsymbol{\Gamma} &= -\frac{1}{4}(\boldsymbol{\Sigma}_{wc} + 2\boldsymbol{\Sigma}_{ac})^{-1} - \frac{1}{4}\boldsymbol{\Sigma}_{wc}^{-1} + \frac{1}{2}\boldsymbol{\Sigma}_{tot}^{-1} \\ \boldsymbol{\Lambda} &= -\frac{1}{4}(\boldsymbol{\Sigma}_{wc} + 2\boldsymbol{\Sigma}_{ac})^{-1} + \frac{1}{4}\boldsymbol{\Sigma}_{wc}^{-1} \\ \mathbf{c} &= ((\boldsymbol{\Sigma}_{wc} + 2\boldsymbol{\Sigma}_{ac})^{-1} - \boldsymbol{\Sigma}_{tot}^{-1})\boldsymbol{\mu} \\ k &= \log|\boldsymbol{\Sigma}_{tot}| - \frac{1}{2}\log|\boldsymbol{\Sigma}_{wc} + 2\boldsymbol{\Sigma}_{ac}| - \frac{1}{2}\log|\boldsymbol{\Sigma}_{wc}| \\ &\quad + \boldsymbol{\mu}^T(\boldsymbol{\Sigma}_{tot}^{-1} - (\boldsymbol{\Sigma}_{wc} + 2\boldsymbol{\Sigma}_{ac})^{-1})\boldsymbol{\mu}. \end{aligned} \tag{5.11}$$

We recall that the computation of a bilinear form $\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{y}$ can be expressed in terms of the Frobenius inner product as $\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{y} = \langle \mathbf{A}, \mathbf{x}\mathbf{y}^{\mathrm{T}} \rangle = \mathrm{vec}(\mathbf{A})^{\mathrm{T}}\mathrm{vec}(\mathbf{x}\mathbf{y}^{\mathrm{T}})$, where $\mathrm{vec}(\cdot)$ stacks the columns of a matrix into a vector. Therefore, the log-likelihood ratio score can be written as a dot product of a vector of weights $\mathbf{w}^{\mathrm{T}}$, and an expanded vector $\boldsymbol{\varphi}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)$ representing a trial:

$$\begin{aligned} s &= \mathbf{w}^{\mathrm{T}} \boldsymbol{\varphi}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) \\ &= \begin{bmatrix} \mathrm{vec}(\boldsymbol{\Lambda}) \\ \mathrm{vec}(\boldsymbol{\Gamma}) \\ \mathbf{c} \\ k \end{bmatrix}^T \begin{bmatrix} \mathrm{vec}(\boldsymbol{\phi}_1\boldsymbol{\phi}_2^{\mathrm{T}} + \boldsymbol{\phi}_2\boldsymbol{\phi}_1^{\mathrm{T}}) \\ \mathrm{vec}(\boldsymbol{\phi}_1\boldsymbol{\phi}_1^{\mathrm{T}} + \boldsymbol{\phi}_2\boldsymbol{\phi}_2^{\mathrm{T}}) \\ \boldsymbol{\phi}_1 + \boldsymbol{\phi}_2 \\ 1 \end{bmatrix}. \end{aligned} \tag{5.12}$$

Hence, we have obtained a generative generalized linear classifier [Bishop, 2006], where the probability for a same-speaker trial can be computed from the log-likelihood ratio score using the sigmoid activation function as

$$p(H_s|\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) = \sigma\left(\log \frac{p(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2|H_s)}{1 - p(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2|H_s)} + \log \frac{p(Hs)}{1 - p(H_s)} \right) = \sigma(s + \mathrm{logit}(p(H_s))). \tag{5.13}$$

Adding the $\mathrm{logit}(p(H_s))$ score, which adjusts the constant k in the vector of weights, allows for setting different priors for both hypotheses.

5.3 Discriminative classifier

In this section, we describe how we train the weights $\mathbf{w}$ directly, in order to discriminate between same-speaker and different-speaker trials, without having to explicitly model the distributions of i-vectors. To represent a trial, we keep the same expansion $\boldsymbol{\varphi}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)$ as defined in (5.12). Hence, we reuse the functional form for computing verification scores that provided excellent results with generative PLDA.

5.3.1 Logistic Regression

The set of training examples $\mathbf{r}_1 \ldots \mathbf{r}_{|\mathcal{T}|} \in \mathcal{T}$, which we continue referring to as training trials, comprises both different-speaker and same-speaker trials. By trial $\mathbf{r}$ we understand a combination of two i–vectors $\mathbf{r} = (\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)$. By introducing the variable for trial, our score for a particular trial becomes $s_\mathbf{r} = \mathbf{w}^\mathrm{T}\boldsymbol{\varphi}(\mathbf{r}) = \mathbf{w}^\mathrm{T}\boldsymbol{\varphi}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)$. Let us also define the coding scheme $t \in \{-1, 1\}$ to represent labels for the different-speaker, and same-speaker trials, respectively. Assigning each trial a log-likelihood ratio $s_\mathbf{r}$ and the correct label $t_\mathbf{r}$, the log probability of recognizing the trial correctly can be expressed as

$$\log p(t_\mathbf{r}|\mathbf{r}) = -\log(1 + \exp(-s_\mathbf{r} t_\mathbf{r})). \tag{5.14}$$

This is easy to see from equation (5.13) and recalling that $\sigma(-s) = 1 - \sigma(s)$. In the case of logistic regression, the objective function to maximize with respect to the optimized parameters $\mathbf{w}$ is the log posterior probability of correct labeling of all training examples, i.e. the sum of expressions (5.14) evaluated for all training trials.

$$\mathcal{Q} = \sum_{\mathbf{r}\in\mathcal{T}} \log p(t_\mathbf{r}|s_\mathbf{r}(\mathbf{w})) \tag{5.15}$$

$$= \sum_{\mathbf{r}\in\mathcal{T}} -\log\left(1 + \exp(-t_\mathbf{r} s_\mathbf{r}(\mathbf{w}))\right) \tag{5.16}$$

Equivalently, this can be expressed by minimizing the cross-entropy error function, which is a sum over all training trials

$$E(\mathbf{w}) = \sum_{\mathbf{r}\in\mathcal{T}} \alpha_\mathbf{r} E_{LR}(t_\mathbf{r} s_\mathbf{r}) \tag{5.17}$$

where the logistic regression loss function

$$E_{LR}(t_\mathbf{r} s_\mathbf{r}) = \log(1 + \exp(-t_\mathbf{r} s_\mathbf{r})) \tag{5.18}$$

is simply the negative log probability (5.14) of correctly recognizing a trial.

To control over-fitting to training data and to keep the optimized parameters from reaching large values, we can introduce a regularization by adding a penalty term to the error function. The simplest form of the regularization penalty is the sum of squares of all parameters, leading to a modified error function

$$\tilde{E}(\mathbf{w}) = \sum_{\mathbf{r}\in\mathcal{T}} \alpha_\mathbf{r} E_{LR}(t_\mathbf{r} s_\mathbf{r}) + \frac{\lambda}{2}\|\mathbf{w}\|^2, \tag{5.19}$$

where $\|\mathbf{w}\|^2 = \mathbf{w}^\mathrm{T}\mathbf{w}$ and the coefficient λ is a constant controlling the tradeoff between the error function and the regularizer. This L_2 regularizer can be extended by incorporating a prior knowledge of the parameters $\mathbf{w}$ and therefore allow it to limit the distance of the optimized parameters from some particular offset (for example the parameters estimated from the generative model). The error function then takes the form of

$$\tilde{E}(\mathbf{w}) = \sum_{\mathbf{r}\in\mathcal{T}} \alpha_\mathbf{r} E_{LR}(t_\mathbf{r} s_\mathbf{r}) + \frac{\lambda}{2}\|\mathbf{w} - \hat{\mathbf{w}}\|^2. \tag{5.20}$$

This regularization can be seen as imposing an isotropic Gaussian prior on the parameters [Bishop, 2006]. The $\hat{\mathbf{w}}$ defines the mean of the isotropic Gaussian prior and the regularization constant λ can be seen as a parameter to control the variance of this prior.

The coefficients α_n allow us to weight individual trials. When set to zero, it can be used to "turn off" some unwanted trials – for example same i–vector trials or cross-gender trials. We use these coefficients also to assign different weights to same-speaker and different-speaker trials. This allows us to select a particular operating point, around which we want to optimize the performance of our system without relying on the proportion of same- and different-speaker trials in the training set. The advantage of using the cross-entropy objective for training is that it reflects performance of the system over a wide range of operating points (around the selected one). We can show that by setting the α coefficients proportional to the number of same- ($|\mathcal{T}_1|$) and different-speaker trials ($|\mathcal{T}_2|$) as $\frac{1}{2\log(2)|\mathcal{T}_1|}$ and $\frac{1}{2\log(2)|\mathcal{T}_2|}$, our error function without regularization becomes

$$\begin{aligned} E_{\mathcal{T}}(\mathbf{w}) &= \frac{1}{2\log(2)} \left(\frac{1}{|\mathcal{T}_1|} \sum_{\mathbf{r}\in\mathcal{T}} \log(1+\exp(s_{\mathbf{r}}(\mathbf{w}))) + \frac{1}{|\mathcal{T}_2|} \sum_{\mathbf{r}\in\mathcal{T}} \log(1+\exp(s_{\mathbf{r}}(\mathbf{w}))) \right) \qquad (5.21) \\ &= C_{llr,\mathbf{w}}(\mathcal{T}), \end{aligned}$$

which is the C_{llr} performance measure for the speaker verification task as defined in [Brümmer and du Preez, 2006]. This probabilistic behavior of the logistic regression classifier is one of its advantages against the SVM as it trains the weights so that the score $s_{\mathbf{r}} = \mathbf{w}^{\mathrm{T}}\boldsymbol{\varphi}(\mathbf{r}) = \mathbf{w}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)$ can be interpreted as the log-likelihood ratio between hypotheses H_s and H_d, and therefore, the calibration step is not so necessary.

Chapter 6

Experimental Results

This chapter will present results obtained with the presented techniques on various datasets. First, to put the presented techniques into the historical context, we will present a short description and performance comparison of the past state-of-the-art techniques on a common SRE 2010 dataset.In Section 6.2, we will take the standard PLDA without any i–vector normalization as a baseline and present (still on SRE 2010 dataset) the effects of discriminatively trained PLDA and i–vector length normalization.Finally, we will compare all presented PLDA techniques on NIST SRE 2012 dataset in Section 6.4. The superiority of the full-posterior PLDA for short segments, where the uncertainty of extracted i–vectors is high, will be demonstrated on modified NIST SRE 2010 datasets.

6.1 Comparison of Techniques on NIST SRE 2010

In Figure 6.1, we can observe the evolution of the SRE systems. Clearly, the introduction of the channel adaptation has dramatically increased the performance, especially when the system was evaluated on data coming from different collection or simply containing channel effects not present during the UBM training.

JFA was another milestone, which greatly improved the performance at the time when it was introduced. Surprisingly, the effect is not so big on the NIST SRE 2010. However this technique led to the introduction of i–vectors and we can observe another substantial gain in the performance with the cosine distance scoring of i–vectors.

If we compare PLDA with the cosine distance scoring, we do not see much of a difference between the two systems. In fact the cosine distance scoring is better on the low miss-rate region of the DET curve. However, this situation has changed in favor of PLDA after applying length normalization.

6.2 Evolution of the PLDA

After the NIST SRE 2010 evaluation, PLDA was in the center of the interest of the research community. Shortly after the NIST workshop and Odyssey 2010 conference in Brno, we have introduced a discriminative way of training the PLDA parameters. It was the BOSARIS workshop in Brno, where both the training using SVM [Cumani et al., 2013] and logistic regression [Burget et al., 2011] were developed.

In Figure 6.2, we can observe the effect of both discriminatively trained PLDA, length normalization and additional condition-dependent mean normalization (mean of the training

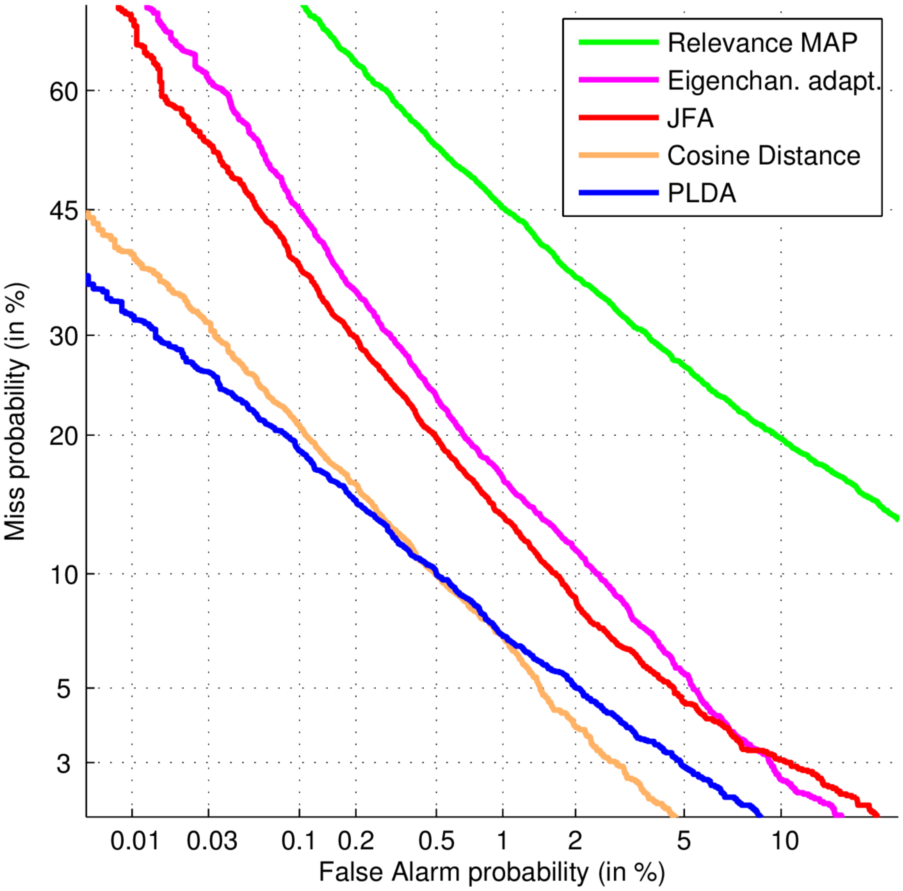

Figure 6.1: Comparison of SRE techniques on female subset of NIST SRE 2010 condition 5

i–vectors coming from the telephone data was removed from the evaluation data). All of the PLDA systems are trained on the same dataset as described in the previous section. The baseline PLDA system represented by the blue DET curve is taken from the previous section, the red DET curve represents the discriminatively trained PLDA system, with no length normalization or other transformation of i–vectors. DPLDA was trained with all of the parameters initialized as matrices of zeros. The target prior probability was set to 0.001 to reflect the NIST SRE 2010 primary metric. The regularization was performed by means of early stopping during this experiment. It took approximately 30 iterations for the algorithm to converge.

The Magenta line represents the system with length normalization that was tuned to get the best overall results for all NIST SRE 2010 conditions. In this system, i–vectors were first reduced into 150 dimensions and then the PLDA with both full rank matrices representing speaker and channel subspaces was trained. The last system represented by the black DET curve is a modification of the magenta system which consists only in the condition dependent mean normalization. This has further improved the PLDA system on the telephone condition. It should be noted, that this approach was specific to the particular training list used during these experiments. During our other experiments with the PLDA, we have extended our training list with the additional telephone and microphone data and the positive effect of this condition-dependent mean normalization was reduced.

The discriminative training can apparently deal with the non-Gaussian behavior of the i–vectors and produce significantly better results than the baseline PLDA. However, the discrim-

inative PLDA was not superior to the generative trained version for very long time. Shortly after this approach was developed, a length normalization was introduced, and standard PLDA with the i–vector pre-processing as described in chapter 4.5 has reached the performance of the
heavy-ta d on
i–vectors urve
for the I dard
PLDA w

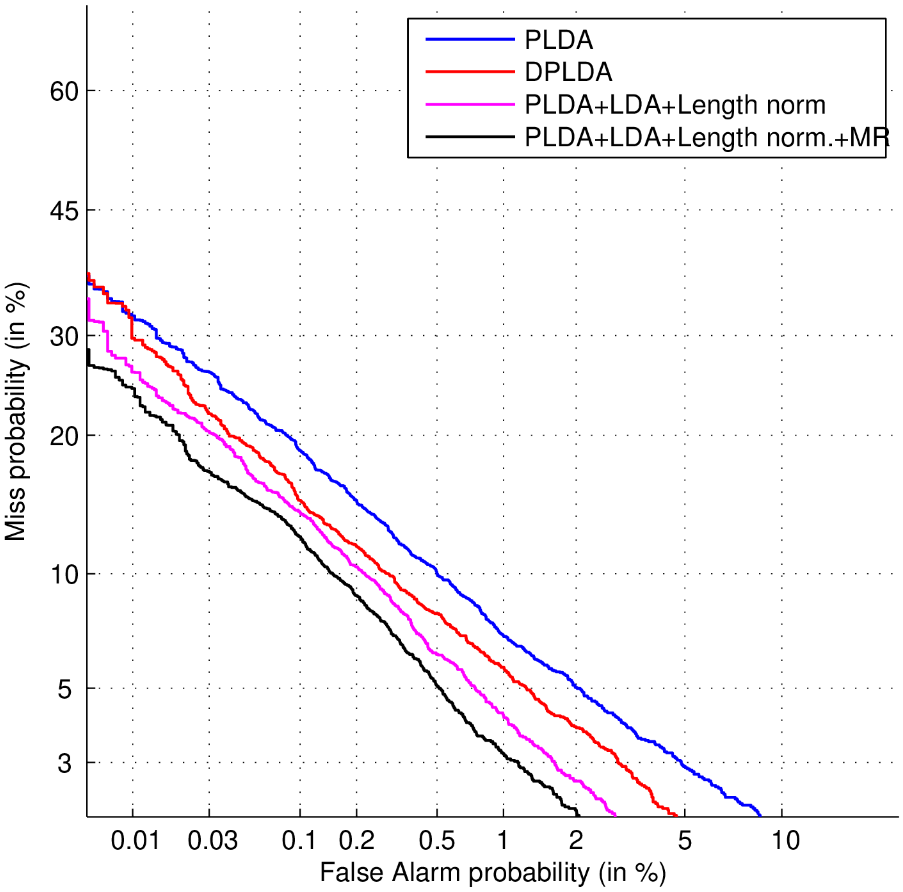

Figure 6.2: Comparison of PLDA systems on female subset of NIST SRE 2010 condition 5: Blue system is a standard PLDA without length normalization, red DET curve represents discriminatively trained PLDA (DPLDA), magenta and black corresponds to the standard PLDA system with length normalization and additional condition dependent mean normalization.

6.3 Analysis of PLDA and DPLDA on RATS Data

Evaluating SRE performance on the RATS data poses many more challenges than simply taking the state-of-the-art system and running it on the data. This extremely noisy data has brought a lot of attention to developing different variants of robust acoustic features and voice activity detection. It would be out of the scope of this work to discuss the RATS-specific techniques and we refer the reader to a general system description [Plchot et al., 2013] of our submission for the RATS evaluation in 2013, from which we derive our baseline system.

It is important to mention the composition of the training set for PLDA. After tuning the

Table 6.1: Comparison of the PLDA and DPLDA systems trained on all data, 10s segments or 30 s segmets. Results are given on the RATS Patrol development sets. 30s-30 s and 10s-10 s correspond to the duration of the enrollment and test utterances. The metrics are FA_10, which correspond to the false alarm rate at miss rate 10% and MISS_2.5 is a miss rate at false alarm rate 2.5%. EER stands for equal-error rate.

	30 s – 30 s			10 s – 10 s		
System	FA_10	MISS_2.5	EER	FA_10	MISS_2.5	EER
PLDA all	3.53	13.36	6.21	10.04	27.01	10.04
DPLDA all	3.68	13.89	6.30	10.03	28.11	10.02
PLDA 30 s	3.32	12.62	6.06	9.99	26.41	9.99
DPLDA 30 s	**3.12**	**12.09**	**5.81**	9.29	26.43	9.66
PLDA 10 s	3.54	13.21	6.17	9.29	**25.75**	9.65
DPLDA 10 s	3.48	13.24	6.08	**9.01**	25.94	**9.49**

composition of our training data, the general consensus was to use as many short cuts from the segments as possible along with the original long segments. The reason for this composition is greatly influenced by the evaluations, where the emphasis is put on the performance obtained on the 30 s and 10 s cuts. There is also a 3 s and 120 s test condition in RATS SRE evaluation protocol. The 120 s condition is getting less attention as the program goals for this test were mostly achieved. The 3 s condition was considered too hard especially in the first two phases of the RATS project and we did not focus on tuning for these durations.

The final training list for our baseline PLDA system was a compromise between the performance on the short duration segments and a reasonable amount of data for training the DPLDA system. In total, it contained 210 thousand segments, out of which 70 thousand were randomly selected 30 s cuts and another 70 thousand were randomly selected 10 s cuts. The training of PLDA followed the same recipe as previously described, with LDA dimensionality reduction to 200 dimensions and length normalization. Corresponding DPLDA systems were trained using parameters initialized to zeros. As the trials in the RATS SRE evaluations are defined as multi-session (6 enrollment segments versus one test), our development test sets also follow this scheme. In order to obtain the scores with the DPLDA system, we used i–vector averaging to represent the multi-session trial as a standard one-to-one i–vector trial. We performed the multi-session scoring with standard PLDA, but it should be noted that doing the averaging does not significantly change the results.

Results of the experiments reported on the metrics of the RATS program are summarized in table 6.1. We report only results obtained on the RATS Patrol team development test sets as the key for the official evaluation set of the program was not available at the time of writing this text.

It can be seen that training both systems on the whole dataset yields slightly worse performance than training a duration-dependent system. Also the DPLDA system is performing slightly worse than the PLDA system when trained on all data. The situation has finally turned in favor of DPLDA when training duration-dependent systems. In these scenarios, the DPLDA outperformed PLDA on almost all metrics.

6.4 Full Posterior Distributions PLDA

The proposed PLDA model aims at addressing the uncertainty in i–vector estimates. Thus, a dataset was defined that consists of speech segments, from NIST SRE10 extended core condition, which were cut, after Voice Activity Detection, to obtain segments of variable duration in the range 3–30, 10–30, 3-60, and 10–60 seconds, respectively. These sets of segments have been scored according to the official NIST SRE 2010 conditions 1–5 [NIST, 2010].

All experiments were performed using i–vector posteriors with dimensionality 400. The PLDA was trained with a speaker variability sub–space of dimensionality 120, and full channel variability sub–space. Although both female and male speaker tests were performed, we report more detailed results on the female datasets only, because the NIST SRE 2010 core test on female speakers is known to be more difficult, thus more often compared in the literature.

Table 6.2 summarizes the results of the tests performed on the NIST SRE 2010 female extended conditions, including the core condition (condition 5), in terms of percent Equal Error Rate and normalized minimum Detection Cost Function ($\mathrm{DCF}_{\mathrm{old}}$ and $\mathrm{DCF}_{\mathrm{new}}$) as defined by NIST for SRE08 and SRE10 evaluations [NIST, 2010]. In this table, the PLDA and FPD–PLDA systems are compared using the original interview data, or telephone conversations, without any cut. Labels "tel" and "tel+mic" refer to the datasets used for training the PLDA parameters, including telephone data only, or additional microphone data. Labels "Std" and "FPD" refer to the standard and the Full Posterior Distribution PLDA, respectively. The first two rows give the baseline results, obtained using standard i–vectors trained on telephone data only, for the five NIST 2010 conditions. It can be observed that the matched conditions 5 and 1 — tel–tel and int–int, respectively, achieve the best results, whereas the difficulty of the task decreases from condition 2 to condition 4. The same behavior is confirmed for the other experimental conditions, shown in the remaining lines, and for the other tests using variable duration segments. The new model not only keeps the accuracy of the standard model, as expected for long segments, but also shows a slight relative improvement in three conditions (2,3,4). The third row describes the effect of using the i–vector covariance also in training. As expected, since the training segments have long durations, the results are similar to the ones reported in the second row. The last three rows show the effect of adding microphone data in training the PLDA parameters: sensible performance improvement is obtained, excluding, as expected, the matched tel–tel condition 5.

Since the system trained with the "tel" list performs worse than the one trained with the "tel+mic" list, all the remaining experiment on the NIST 2010 data, whenever not mentioned, have been performed with the latter. Table 6.3 compares, in its first three rows, the performance of the PLDA and FPD–PLDA classifiers using the two length–normalization methods described in Chapter 4.5 on the 3–60 seconds cuts. The results of the last row show that there is no advantage in using the full i–vector posterior in training the PLDA models. The effect of the two length–normalization approaches is comparable, thus in the following we will present only the results obtained with the Projected Length Normalization (FPD2) (4.24).

6.4.1 Comparison on NIST 2012

Pooled results for female and male speakers are reported in Table 6.4 for the NIST 2012 SRE . The i–vector dimension was increased to 600. Moreover, Linear Discriminant Analysis was performed to reduce the i–vector dimensionality to 200, before applying i–vector whitening and length normalization. Since the resulting i–vectors are already small, no dimensionality reduction was applied for the speaker sub–space, i.e. the speaker sub–space was set to 200.

The results comparing standard PLDA and FPD–PLDA are given in Table 6.4 in terms of minimum and actual $C_{primary}$. Note, that in contrast to min-DCF, there is no analytic version of the "minimum" $C_{primary}$. By "minimum", we mean a $C_{primary}$ as defined by NIST, but with calibration performed on the evaluation data

These results show that the Asymmetric FPD–PLDA is almost equivalent to the standard PLDA. For minimum $C_{primary}$, it gains for conditions 2 and 5, which include short and variable duration segments. An excellent result have been obtained with discriminatively trained PLDA in terns of the actual $C_{primary}$, where the calibration loss for DPLDA system is low compared to the other two techniques. These results confirm that DPLDA is a technique with a built-in calibration, which is a very useful property for a real use scenario.

Table 6.2: Results for the core extended NIST SRE2010 female tests in terms of % EER, minDCF$_{\text{old}}$ × 1000 and minDCF$_{\text{new}}$ × 1000 using different training lists and PLDA models. Label “tel” and “tel+mic” refer to the datasets used for training the PLDA, including or not microphone data. “Std” and “FPD” labels refer to standard PLDA and FPD–PLDA, respectively. I–vector posterior length–normalization is performed by means of (4.24).

List	Train	Test	condition 2			condition 3			condition 4			condition 1			condition 5		
			EER	DCF$_{\text{old}}$	DCF$_{\text{new}}$	EER	DCF$_{\text{old}}$	DCF$_{\text{new}}$	EER	DCF$_{\text{old}}$	DCF$_{\text{new}}$	EER	DCF$_{\text{old}}$	DCF$_{\text{new}}$	EER	DCF$_{\text{old}}$	DCF$_{\text{new}}$
tel	Std	Std	4.2	224	641	2.5	113	445	1.7	102	411	2.0	84	346	2.0	100	339
tel	Std	FPD	3.9	214	638	2.3	111	462	1.6	101	419	1.7	81	346	2.0	100	346
tel	FPD	FPD	3.9	214	635	2.4	110	450	1.6	99	415	1.8	79	345	2.0	98	336
tel+mic	Std	Std	2.6	124	460	2.2	103	405	1.1	65	303	1.8	68	258	1.9	105	335
tel+mic	Std	FPD	2.3	114	455	2.1	103	402	1.0	60	296	1.7	63	254	2.0	103	347
tel+mic	FPD	FPD	2.3	112	455	2.0	100	396	1.0	59	288	1.6	60	253	2.0	101	344

Table 6.3: Results for cuts of 3–60 second test data, using different length–normalization approaches. The PLDA parameters are trained using both microphone and telephone data. Labels “Std” and “FPD” refer to standard PLDA and FPD–PLDA, respectively, and the numeric suffix of FPD corresponds to the i–vector posterior length–normalization method.

Train	Test	condition 2			condition 3			condition 4			condition 1			condition 5		
		EER	DCF$_{\text{old}}$	DCF$_{\text{new}}$	EER	DCF$_{\text{old}}$	DCF$_{\text{new}}$	EER	DCF$_{\text{old}}$	DCF$_{\text{new}}$	EER	DCF$_{\text{old}}$	DCF$_{\text{new}}$	EER	DCF$_{\text{old}}$	DCF$_{\text{new}}$
Std	Std	9.1	384	812	7.8	368	832	7.3	312	695	7.0	273	630	6.7	337	729
Std	FPD1 (eq. 4.23)	6.7	327	791	6.1	343	838	5.2	259	676	4.8	232	603	6.2	322	722
Std	FPD2 (eq. 4.24)	6.7	328	791	6.2	343	838	5.2	259	676	4.7	232	603	6.2	323	722
FPD2	FPD2	6.5	327	796	6.3	355	837	5.0	255	676	4.6	229	601	6.3	328	731

Table 6.4: NIST SRE 2012 core-extended test: comparison of DPLDA, PLDA and Asymmetric FPD–PLDA on minimum and actual $C_{primary}$. The numbers associated to the conditions refer to the mean duration of the segments, after voice activity detection, and to the corresponding standard deviation.

System	Condition 1	Condition 2	Condition 3	Condition 4	Condition 5
	interview	phone call	interview	phone call	phone call
	without added noise	without added noise	with added noise	phone with added noise	from a noisy environment
	45s – 41	56s – 48	75s – 37	110s – 56	57s – 48
DPLDA (min)	0.230	0.261	0.206	0.287	0.249
PLDA (min)	0.255	0.206	0.244	0.265	0.222
FPD–PLDA (min)	0.253	0.193	0.241	0.264	0.211
DPLDA (act)	0.250	0.300	0.215	0.339	0.333
PLDA (act)	0.336	0.292	0.294	0.370	0.342
FPD–PLDA (act)	0.336	0.292	0.293	0.389	0.344

Chapter 7

Conclusions

This work proposes two variants of the Probabilistic Discriminant Analysis, which, in its standard form, is currently considered as the state-of-the art technique in the text-independent speaker recognition. Preceding state-of-the art techniques have been put into the context with the standard PLDA, which also serves as a baseline for the proposed modifications. The performed comparison of all techniques on the NIST SRE 2010 dataset presents a historical progress in the SRE technology. In Figure 6.1, we can identify two milestones in the SRE technology. It is an introduction of the channel compensation techniques and using i–vectors as low-dimensional, information-rich features for modeling.

Discriminative PLDA

The functional form of the standard PLDA model for evaluating the speaker verification trial has been used as a basis for designing the discriminative approach to training of the PLDA parameters. A single discriminative model then directly addresses the symmetric speaker verification task: a discrimination between the same- and different-speaker trial formed by two i–vectors. Although the discriminative training was initially bringing substantial improvements with respect to original PLDA, after the application of the length normalization of i–vectors, the standard PLDA model achieves slightly better performance in the minimum DCF and EER metrics.

The performed comparative study of PLDA and DPLDA in various acoustic environments has also shown slightly better overall performance of the standard generative PLDA in terms of minimum DCF and EER evaluation metrics. In the domain of highly degraded RATS data, the discriminative approach has shown minor improvements in the duration-dependent systems with respect to generative baseline. These experiments, however, show a theoretical best possible performance not taking into account any calibration loss.

Minimizing the cross-entropy error function as an objective for discriminative training of DPLDA forces the system to output scores in form of calibrated log-likelihood ratios. The possibility of weighting individual trials allows for setting the desired operating point of the system already during training, which makes the consecutive calibration step less necessary. The quality of the calibration of the DPLDA scores has been confirmed by the experiments where the calibration loss on an unseen evaluation set is lower than for the other PLDA variants. This behavior is a desirable property in a real use scenario, where the actual error rates matter much more than the theoretical minimum error rates.

Full Posterior Distribution PLDA

In generative approach, a PLDA model which exploits the uncertainty of the i-vector extraction process has been presented. The basic idea lies in the formulation of the PLDA likelihood, which has been derived for a Gaussian PLDA model based on the i–vector posterior distribution. The new formulation of likelihood evaluation defines a new PLDA model, where the inter–speaker variability is assumed to have a segment–dependent distribution.

Taking into account the posterior distribution of all i–vectors representing an utterace also leads to the need of normalize this distribution in line with the already established length normalization of i–vectors. Two i–vector pre–processing techniques complying with the new PLDA model have been proposed and their effects were compared on the system accuracy. It was shown that an approximate version of a linearized length normalization is sufficiently accurate.

The complexity of the PLDA and FPD–PLDA implementations has been analyzed and an Asymmetric FPD–PLDA approach has been proposed. The asymmetric approach allows for a substantial complexity reduction in a practical detection scenario with known target speakers.

The results obtained both on the extended core tests and on short cuts of different duration of the NIST SRE 2010, and on the extended tests of NIST SRE 2012, confirm that the FPD–PLDA outperforms PLDA mostly for short variable duration test segments. No loss in the performance has been observed for the standard tests containing long test segments. It was also experimentally demonstrated that for the scenarios when sufficiently long utterances are available for training the PLDA model, we can use the standard PLDA for training and FPD-PLDA for scoring. Therefore in most real use cases, there is no need to perform more expensive FPD-PLDA training.

Future Work

The FPD-PLDA can clearly outperform the baseline when testing on short utterances and DPLDA excels at producing well-calibrated scores. Therefore both techniques present a viable option for a real use and should be evaluated in production systems. In my opinion, there are more unknowns in the discriminative approach to be explored. A possible direction for future research could be to address the problem of overtraining the model on the training data and propose more sophisticated ways of regularization. Also an automatic forming of all possible trials in the discriminative training by taking all possible i–vector pairs does not correspond to the real test and could be redesigned. For example forming the trials out of the same utterance, just recorded over different microphone introduces many artificial positive examples, which should be avoided. From the perspective of the functional form for scoring, other blocks can be added to simulate the i–vector pre-processing or condition-dependent calibration.

Bibliography

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[Brümmer, 2009] Brümmer, N. (2009). EM for JFA: Technical report, Agnitio Research, South Africa. `https://sites.google.com/site/nikobrummer/EMforJFA.pdf`.

[Brümmer, 2010] Brümmer, N. (2010). EM for PLDA: Technical report, Agnitio Research, South Africa. `https://sites.google.com/site/nikobrummer/EMforPLDA.pdf`.

[Brümmer and de Villiers, 2010] Brümmer, N. and de Villiers, E. (2010). The speaker partitioning problem. In *Proc. of Odyssey 2010*, Brno, CZ.

[Brümmer and du Preez, 2006] Brümmer, N. and du Preez, J. (2006). Application-independent evaluation of speaker detection. *Computer Speech & Language*, 20(2-3):230–275.

[Burget et al., 2011] Burget, L., Plchot, O., Cumani, S., Glembek, O., Matějka, P., and Brümmer, N. (2011). Discriminatively trained probabilistic linear discriminant analysis for speaker verification. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Prague, CZ.

[Cumani et al., 2011] Cumani, S., Brümmer, N., Burget, L., and Laface, P. (2011). Fast discriminative speaker verification in the i-vector space. In *Proc. of ICASSP , 2011*, pages 4852 –4855, Prague, CZ.

[Cumani et al., 2013] Cumani, S., Brümmer, N., Burget, L., Laface, P., Plchot, O., and Vasilakis, V. (2013). Pairwise discriminative speaker verification in the i–vector space. *IEEE Transactions on Audio, Speech and Language Processing*, 21(6):1217–1227.

[Cumani et al., 2014] Cumani, S., Plchot, O., and Laface, P. (2014). On the use of i-vector posterior distributions in probabilistic linear discriminant analysis. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 22(4):846–857.

[Dehak et al., 2010] Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., and Ouellet, P. (2010). Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech and Language Processing*, pages 1 –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, SERIES B*, 39(1):1–38.

[Ferrer et al., 2011] Ferrer, L., Bratt, H., Burget, L., Cernocky, H., Glembek, O., 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 *Proceedings of SRE11 analysis workshop*, Atlanta.

[Ferrer et al., 2012] Ferrer, L., Burget, L., Plchot, O., and Scheffer, N. (2012). A unified approach for audio characterization and its application to speaker recognition. In *Proceedings of Odyssey 2012*, pages 317–323. International Speech Communication Association.

[Garcia-Romero, 2011] Garcia-Romero, D. (2011). Analysis of i-vector length normalization in Gaussian-PLDA speaker recognition systems. In *Proc. of the International Conference on Spoken Language Processing (ICSLP)*.

[Glembek, 2012] Glembek, O. (2012). *Optimization of Gaussian Mixture Subspace Models and related scoring algorithms in speaker verification*. PhD thesis, Brno University of Technology.

[Kenny, 2005] Kenny, P. (2005). Joint factor analysis of speaker and session variability: Theory and algorithms - technical report CRIM-06/08-13. Montreal, CRIM, 2005.

[Kenny, 2010] Kenny, P. (2010). Bayesian speaker verification with heavy–tailed priors. In *Proc. of Odyssey 2010*, Brno, Czech Republic. http://www.crim.ca/perso/patrick.kenny, keynote presentation.

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

[Kenny et al., 2004] Kenny, P., Boulianne, G., Ouellet, P., and Dumouchel, P. (2004). Speaker adaptation using an eigenphone basis. *IEEE Transactions on Speech and Audio Processing*, 12(6):579–589.

[Kenny et al., 2007] Kenny, P., Boulianne, G., Oullet, P., and Dumouchel, P. (2007). Joint factor analysis versus eigenchannels in speaker recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 15(7):2072–2084.

[Kenny et al., 2013] Kenny, P., Stafylakis, T., Ouellet, P., and Dumouchel, P. (2013). PLDA for speaker verification with utterances of arbitraty duration. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 7649 – 7653.

[MITLL, 2103] MITLL (2103). Domain adaptation challenge 2013. `http://www.clsp.jhu.edu/user_uploads/workshops/ws13/DAC_description_v2.pdf`.

[NIST, 2010] NIST (2010). The NIST year 2010 speaker recognition evaluation plan. `http://www.itl.nist.gov/iad/mig//tests/sre/2010`.

[Plchot et al., 2013] Plchot, O., Matsoukas, S., Matějka, P., Dehak, N., Ma, J., Cumani, S., Glembek, O., Heřmanský, H., Mesgarani, N., Soufifar, M. M., Thomas, S., Zhang, B., and Zhou, X. (2013). Developing a speaker identification system for the darpa rats project. In *Proceedings of ICASSP 2013*, pages 6768–6772. IEEE Signal Processing Society.

[Prince and Elder, 2007] Prince, S. J. D. and Elder, J. H. (2007). Probabilistic linear discriminant analysis for inferences about identity. In *11th International Conference on Computer Vision*.

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

[Villalba et al., 2013] Villalba, J., Diez, M., Varona, A., and Lleida, E. (2013). Handling recordings acquired simultaneously over multiple channels with plda. In *Proceedings of Interspeech 2013*, Lyon, France.