Factor analysis based channel compensation in speaker verification

Charles-alban Deledalle

Technical Report n^o0705, May 2007
revision 1255

This report describes a powerful channel compensation method for the text-independent speaker verification task. This powerful method is developed in the LRDE Speaker Verification framework. The purpose of a text-independent speaker verification system is to check whether a hypothesised speaker is really the author of a speech utterance. The channel compensation problem arises when training data and test data come from different channels. This report first focuses on the current state-of-the-art system based on a probabilistic approach of the acoustic event distribution with Gaussian mixture models. Unfortunately, this baseline system does not cope with channel effects. Then, the method developed in the LRDE Speaker Verification framework is introduced. This one is based on a factor analysis which enables to deal with the channel compensation problem by taking advantage of the limits of the state-of-the-art system. The Factor Analysis model considers the variability of the Gaussian mixture model as a linear combination of the variabilities of the speaker and channel unobservable components. This decomposition is based under the classical MAP and the eigenchannel MAP assumptions. Finally, the results obtained on the NIST-2006 Speaker Recognition Evaluation (female trials) will be presented.

Keywords
Speaker verification, Cepstral vector, Gaussian mixture model, Supervector, Channel compensation, Factor analysis, Eigenchannel, Eigenvoice

Laboratoire de Recherche et Développement de l'Epita
14-16, rue Voltaire – F-94276 Le Kremlin-Bicêtre cedex – France
Tél. +33 1 53 14 59 47 – Fax. +33 1 53 14 59 22
deledalle@lrde.epita.fr – http://www.lrde.epita.fr/

Copying this document

Contents

Introduction

Nowadays, in a lot of secure applications, people have to prove their own identity to go into a critical system. Most of these applications are based on password authentications as in bank cash points or in the access of some protected buildings. However in such systems, a robber can easily find the password and use it to gain access to the critical area. For this reason, more and more authentication systems are based on biometrical features like fingerprints, iris or voice which present more inviolable features. Indeed, the study of these biometrical features is an expanding research field. Furthermore, it is known that fingerprints have been used extensively in criminal investigations for a long time. Today, voice recognition systems are beginning to have a legal status in some countries as a proof to authenticate a speaker on a tape recording. In this report, we consider the problem of voice authentication, generally called the speaker verification problem. More precisely, we focus on the text-independent speaker verification i.e we do not consider the uttered text.

In our speaker verification task, speakers are first modelled from enrolment data coming from phone recordings. During the verification task, these models enable to check whether a segment of speech is uttered by a hypothesised speaker. For a few years, the community has been more and more interested in resolving this verification problem even if the enrolment conditions are different during the training step and the testing step. In this report, we are particularly interested in dealing with the problem of channel compensation, that is to cope with channel variations between training and testing step. Typically, imagine the problem arising when a speaker is enrolled on a cellular phone while the test utterance comes from a landline phone. Actually, the channel adds its own noise in the speaker models, and thus it is very hard to authenticate the speaker of an utterance enrolled on a different channel. Cancelling the channel effects also seems to be a good way to improve the verification system.

This report describes the factor analysis model proposed by Kenny et al. (2005) and developed in the LRDE Speaker Verification framework in order to enhance the current LRDE system and to have a channel compensation system. In the first chapter, we discover that the state-of-the-art system presents some limits. It constitutes a motivation to develop a new system that will be presented in a second time. This model is based on a factor analysis which is able to deal with the channel effects. Finally, we will introduce the results obtained by the experiments which compare the baseline system and the system obtained by the factor analysis based LRDE Speaker Verification tool.

Chapter 1

The limits of the probabilistic approach

The state-of-the-art system in speaker verification is built on three main parts as described on Figure 1.1. The first one aims at building a model of the speaker population, commonly called the world model. This one represents the speech behaviour of the studied population in order to model the non-target speakers (or impostors). Typically, the world model is trained from a large amount of speech data which presents speaker and channel variabilities in order to consider all acoustic events and their variations. Moreover, the world model is also used in the enrolment step. Actually, given a short training speech utterance coming from a speaker, a client model is obtained by adaptation of the world model. This adaptation is performed in order to deal with the few amount of available data and with the unseen events. Finally, during the testing step, given a testing speech utterance and a target speaker, a decision score is calculated with respect to the target and the non-target model.

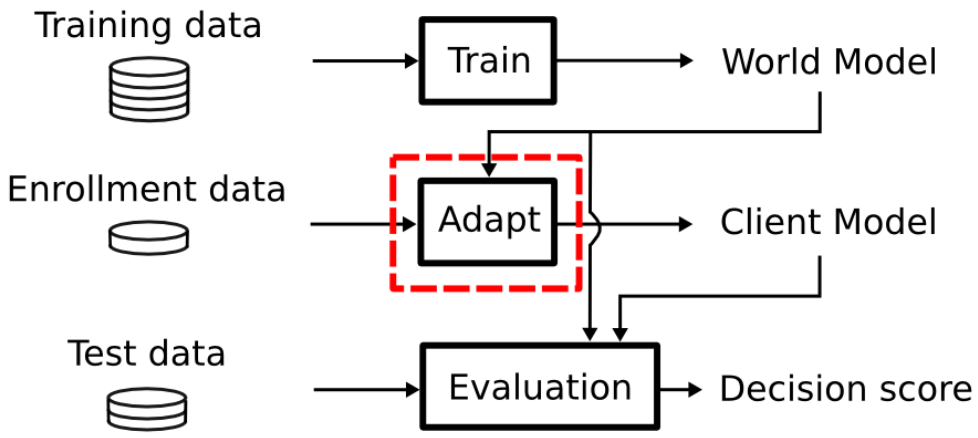

Figure 1.1: Baseline system

As illustrated on Figure 1.1, this chapter focuses on the enrolment step of the client speaker and more precisely on the adaptation procedure used in the baseline system. First of all, the probabilistic models fitting the distribution of the acoustic events will be defined. Then, the Maximum A-Posteriori (MAP) adaptation will be described with the common modification used in speaker verification. Finally, we will discuss about the limits of the MAP adaptation in order to introduce a new model based on a factor analysis.

1.1 Speaker models based on Gaussian Mixture Models (GMM)

In speaker verification, a Gaussian Mixture Model (GMM) is used to characterise the target and the non-target speakers. A GMM is a probability density function (p.d.f) fitting the distribution of the acoustic events. In this report, we assume that the acoustic events are represented by cepstral vectors x_t of size F and their distributions are modelled by GMMs. These GMMs are assumed to have C mixture components which are multivariate Gaussian distributions. Then, a GMM λ is characterised by a set of C triplets

$$\lambda = (\omega_c, \mu_c, \sigma_c)_{1 \leq c \leq C} \tag{1.1}$$

where ω_c is a weight associated to the c-th components, and μ_c and σ_c are respectively a F-dimensional vector and a $F \times F$ diagonal matrix representing the mean and the covariance matrix associated to the c-th multivariate Gaussian distribution. In order to be a p.d.f, the weights of a GMM have to respect the constraint $\sum_{c=1}^{C} \omega_c = 1$. Finally, given a cepstral vector x_t, the likelihood to be uttered by a speaker λ is given by

$$p(x_t|\lambda) = \sum_{c=1}^{C} \omega_c \mathcal{N}(x_t, \mu_c, \sigma_c) \tag{1.2}$$

where $\mathcal{N}(x, \mu, \sigma)$ denotes the multivariate Gaussian distribution having the mean μ and the covariance matrix σ.

1.2 Maximum A Posterior (MAP) adaptation

Generally, in the speaker verification task, there is a few amount of data to train a client speaker. According to the different tasks, the speech enrolments last between 10 seconds and 5 minutes. Given the high degree of freedom of GMMs (C and F are quite big) and the few amount of training data, a speaker model based on GMM cannot be directly estimated with the Expectation Maximisation (EM) algorithm (Dempster et al., 1977). For this reason, the world model is used as a prior knowledge of a speaker in the Maximum A Posteriori (MAP) adaptation. In this section we present the MAP adaptation used in speaker verification which estimates a client model from the world model and a short training utterance (Reynolds et al., 2000).

In MAP adaptation, the client model is derived from the world model by considering a short training utterance. As a variant of the EM algorithm, the MAP adaptation iteratively updates the parameters of the GMM $\lambda = (\omega_c, \mu_c, \sigma_c)_{1 \leq c \leq C}$ such that the total likelihood for an enrolment utterance $x_1, \ldots, x_T$ is maximised

$$\prod_{t=1}^{T} p(x_t|\lambda^{new}) \geq \prod_{t=1}^{T} p(x_t|\lambda) \tag{1.3}$$

The mixture components are updated by a tradeoff between fitting the corresponding acoustic events and the prior knowledge given by the world model. The problem arising from here, and also from the EM algorithm, is to determine which cepstral vectors correspond to a given mixture component. This latent information could be given by estimating the alignment statistics (Baum-Welch or Viterbi statistics) as described in Appendix B.1. The alignment statistics tell us how a set of cepstral vectors reacts on each mixture component. Figure 1.2 describes the Baum-Welch statistics where the colours of the cepstral vectors indicate the alignment on each Gaussians. These statistics are used to assess whether a component is responding or not. If a component is highly responding, then it could be adapted on the corresponding acoustic events. In another way, if a component is not responding, this one corresponds to an unseen event, and the information will be given by the world model. Finally if the component is fairly responding, the component will be updated by a weighted sum between the corresponding acoustic events and the world model. The convergence speed is determined by a relevance factor τ.

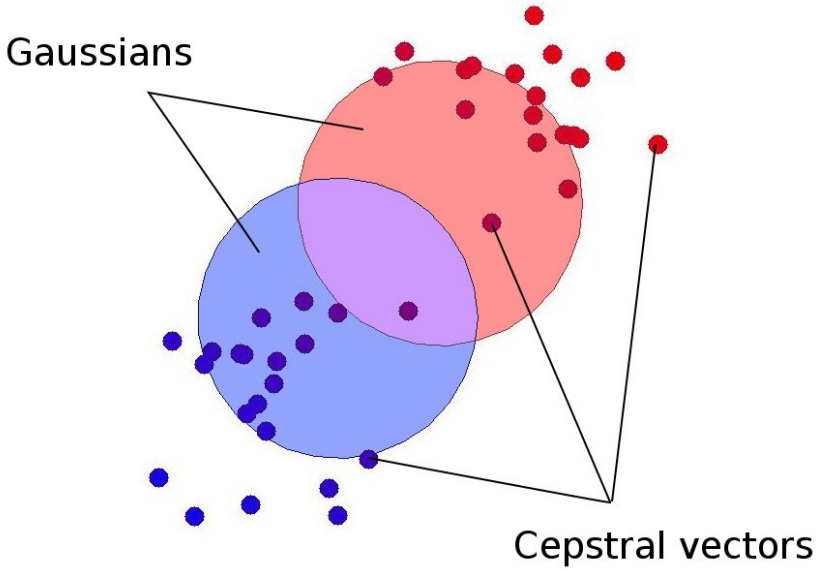

Figure 1.2: Baum-Welch statistics

In speaker verification, Reynolds et al. (2000) present a different version of the MAP adaptation to build the speaker models. Actually, they proved experimentally that best performances are obtained by only updating the mean vectors μ_c and by fixing the weights ω_c and covariance matrices σ_c associated to each Gaussian. As just mean vectors are updated during MAP adaptation, the weights and the covariance matrices could be shared between all speaker models. Therefore, the difference between the speaker models only lies in the mean vectors. Thus, a speaker model could be reduced to a CF-dimensional vector built on the concatenation of the F-dimensional mean vectors coming from the C mixture components. This CF-dimensional vector is commonly named the supervector of a speaker. Now, the MAP adaptation can be expressed as the estimation of a supervector $\mathbf{M}$ by considering a short speech enrolment and a prior supervector extracted from the world model.

1.3 The limits of the MAP adaptation

In the Maximum A Posteriori (MAP) adaptation, an estimate of the speaker supervector is obtained from a short speech enrolment and a prior supervector. Working on the text-independent speaker verification task, we want to build speaker supervectors which are independents from the uttered text and all the enrolment conditions. However, in MAP adaptation it is not clear to have the same estimate from different utterances for a given speaker. There are different ways to explain this source of variability. First, the speech of an individual is continuously changing whether it is during a short time (as a day) or a long time (as a year) that is called the speaker variability. However, the main source of variation is for sure the session variability, that is to say the whole phenomena which are speaker-independent and impair the speaker supervector estimates. A well-known source of session variability occurs when different channels are used, this is precisely the channel variability.

The limits of the MAP adaptation originate from this uncertainty to produce a well-estimated supervector. For this reason, the information about uncertainty has to be considered in the speaker representation in order to have a more robust model and to cope with speaker and channel variabilities. For instance, instead of modelling a speaker by using a supervector, the uncertainty should be estimated by modelling a speaker-dependent and channel-dependent utterance with a random supervector $\mathbf{M}$ (Kenny et al., 2005). In fact, the variance of its random supervector $\mathbf{M}$ should indicate the uncertainty during the MAP adaptation. That is the main idea of the factor analysis decomposition presented in the next chapter.

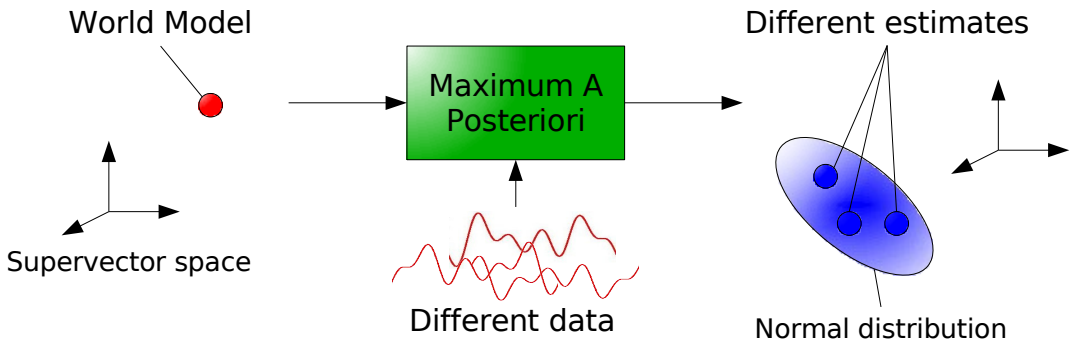

Figure 1.3: Uncertainty during MAP adaptation

The scheme in Figure 1.3 summarises the limits of the MAP adaptation. It is shown that the world model is used as a prior knowledge of the speaker supervector. This model is represented as a supervector which lies in the supervector space. Then, given different short speech enrolments, different speaker supervectors are estimated. Finally, it is clear that an utterance could be represented as a random supervector $\mathbf{M}$ where its variance is assumed to follow a normal distribution.

Chapter 2

A factor analysis modeling for channel compensation

This chapter describes how to perform channel compensation by modelling a speaker with a factor analysis decomposition of the speaker-dependent and channel-dependent random supervector $\mathbf{M}$ (Kenny et al., 2005). A first reason of using a such model is presented in the Chapter 1. Moreover, this new model is also used to deal with channel effects by taking advantage of the available information of channel variability. First, the factor analysis based speaker models will be described. Then, we will explain how to compute the decision score which tells whether a speech utterance is really uttered by a hypothesised speaker. Finally, the different algorithms used to estimate the factor analysis parameters will be explained. Note that this approach is implemented in the LRDE Speaker Verification framework as described in the Appendix A.

2.1 Factor analysis based speaker and channel models

As mentioned in Chapter 1, a speaker-dependent and channel-dependent model is characterised by a CF-dimensional random supervector $\mathbf{M}$ in which the variation indicates the uncertainty during the estimation procedure. We first assumed that this uncertainty comes from both speaker and channel variabilities. Then, the random supervector $\mathbf{M}$ could be expressed as a factor analysis of a speaker-dependent component $\mathbf{s}$ and a channel-dependent component $\mathbf{c}$ given by

$$\mathbf{M} = \mathbf{s} + \mathbf{c} \tag{2.1}$$

where $\mathbf{s}$ and $\mathbf{c}$ are CF-dimensional latent vectors. Observe that a factor analysis is used instead of a Principal Component Analysis (PCA) since the random vector $\mathbf{M}$ is not observable or hidden, there is no analytic solution to estimate $\mathbf{s}$ and $\mathbf{c}$. In the Equation 2.1, the fact that the channel component has an additive effect is a common assumption in speaker verification. This additive effect comes from the extraction procedure of the cepstral vectors performed on the input speech signal to design the acoustic events. Moreover, it is also assumed that the speaker and channel effects lie in different and orthogonal subspaces of the supervector space. Actually, without a such assumption, it would be impracticable to perform a such factor analysis. The subspaces spanned by the speaker component $\mathbf{s}$ and the channel component $\mathbf{c}$ are respectively called the speaker space and the channel space.

Figure 2.1 illustrates the factor analysis decomposition of the random supervector $\mathbf{M}$. There are a speaker component $\mathbf{s}$ lying in the speaker space, and a channel component $\mathbf{c}$ lying in the channel space. In this figure the channel space and the speaker space are two-dimensional and together they span the three-dimensional supervector space. Thus, any supervector $\mathbf{M}$ could be decomposed as a sum of a latent speaker component $\mathbf{s}$ and a latent channel component $\mathbf{c}$.

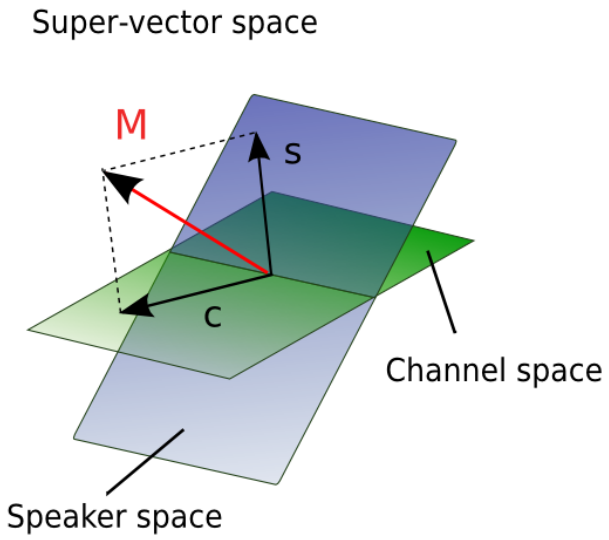

Figure 2.1: Speaker and channel subspaces

In this report, we focus on the factor analysis based on the classical MAP and the eigenchannel MAP assumptions (Kenny et al., 2007). Under the classical MAP assumption, the speaker component $\mathbf{s}$ lies in the supervector space. The classical MAP assumption is opposed to the eigenvoice MAP assumption under which the speaker space is a low dimensional subspace of the supervector space. Using classical MAP instead of eigenvoice MAP has some restrictions in practice. In classical MAP, if the number C of components is large, the speaker space is a high dimensional space and therefore it presents a high degree of freedom. Thus, in order to estimate properly the speaker space, a large amount of speech is required. Unlikely, in speaker verification just a few amount of data is available to train client speaker models. On the contrary, in the eigenvoice MAP the speaker is a low dimensional subspace and can be well-estimated with a small amount of data. However there are several motivations to use classical MAP instead of using eigenvoice MAP. First, in the classical MAP, the estimation of the speaker space is less computationally expensive and is guaranteed to converge to a correct solution. In another way, Shou-Chun (2006) proposes a progressive speaker adaptation in which test data could be used to reestimate the speaker space. That enables to deal with the problem of the few number of enrolment data. Finally, the classical MAP assumption was also selected because it is easier to implement for the first version of the factor analysis based LRDE Speaker Verification tool.

As in eigenvoice MAP, in eigenchannel MAP the channel space is assumed to be a low R_c-dimensional subspace of the CF-dimensional supervector space where $R_c \ll CF$. This assumption is based on the design of the supervectors. Indeed, in speaker verification, GMMs are built with respect to the speaker information. The aim is to model the speaker features only. Thus, a supervector gives very few pieces of information about the channel component. That is the main reason to assume that the channel space dimension is very low.

Now, on the basis of the classical MAP and the eigenvoice MAP, the form of the factor analysis could be introduced as

$$\mathbf{M} = \underbrace{\mathbf{m} + \mathbf{dz}}_{\mathbf{s}} + \underbrace{\mathbf{ux}}_{\mathbf{c}} \tag{2.2}$$

where

$\mathbf{m}$	is a CF-dimensional supervector
$\mathbf{d}$	is a $CF \times CF$ diagonal matrix
$\mathbf{z}$	is a CF-dimensional latent vector
$\mathbf{u}$	is a $CF \times R_c$ full rectangular matrix
$\mathbf{x}$	is a R_c-dimensional latent vector

and $\mathbf{z}$ and $\mathbf{x}$ are both assumed to follow a normal distribution (Kenny et al., 2007). As mentioned in the Equation 2.2 and according to the Equation 2.1, the speaker component $\mathbf{s}$ is equal to $\mathbf{m} + \mathbf{dz}$ and the channel component $\mathbf{c}$ is $\mathbf{ux}$. The components of $\mathbf{z}$ are speaker factors and the components of $\mathbf{x}$ are channel factors. A factor analysis model of a speaker and channel dependent random supervector $\mathbf{M}$ is illustrated in Figure 2.2. It is important to notice that the latent random vector $\mathbf{z}$ and $\mathbf{x}$ are represented as centred on zero. Thus, the random supervector $\mathbf{M}$ is distributed around the supervector $\mathbf{m}$. The matrix $\mathbf{d}$ defined the orientation of the speaker space. This matrix is supposed to be diagonal since all components of the supervector space are assumed to be independents. The latent random vector $\mathbf{z}$ corresponds to the distribution of the speaker component in the speaker space. The matrix $\mathbf{u}$ defines the channel space. This matrix performs a mapping between a low dimensional vector of the channel space to the supervector space. Finally, the latent random vector $\mathbf{x}$ corresponds to the distribution of the channel component in the channel space.

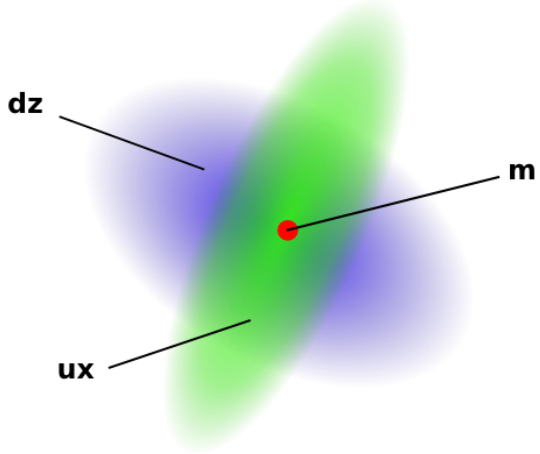

Figure 2.2: Factor analysis decomposition

In our approach, given a set of training utterances, we want to model the specific speaker space and the specific channel space by the set of hyperparameters $\mathbf{\Lambda} = (\mathbf{m}, \mathbf{d}, \mathbf{u})$. The vector $\mathbf{m}$ and the matrix $\mathbf{d}$ are speaker-dependent and channel-independent hyperparameters. The matrix $\mathbf{u}$ is a speaker-independent and channel-dependent hyperparameter. Modelling speaker

and channel space only by hyperparameters holds by fitting the distributions of the latent variables $\mathbf{z}$ and $\mathbf{x}$ on a standard normal distribution $\mathcal{N}(0,1)$. Such a representation enables to estimate the likelihood of an utterance during the speaker verification task and indeed, evaluate the decision score. Actually, given a test speech segment, the more the estimates of the latent variables $\mathbf{x}$ and $\mathbf{z}$ have a standard normal distribution, the more the speech segment utterance is likely uttered by the hypothesised speaker.

The factor analysis model considers both speaker variability and channel variability in a set of speech utterances. This variabilities are reflected by the latent random variables $\mathbf{z}$ and $\mathbf{x}$. The form of the factor analysis as a sum of the speaker component $\mathbf{s}$ and the channel component $\mathbf{c}$ seems to be an appropriate way to deal with the channel effects. That enables to perform channel compensation to produce better decision scores during the verification task. In the next section, the way to estimate the decision score and to cope with the channel effects is described.

2.2 Decision score estimation

In this section, we will explain how to answer to the verification task in the case of using the factor analysis model. The goal is to tell whether a hypothesised speaker, modelled by the hyperparameter set $\mathbf{\Lambda}$, is really the author of a test utterance χ. Generally, in speaker verification, this decision is given by a score θ. Then, the final answer is obtained by comparing the decision score θ with a threshold. As in the baseline speaker verification system, the decision score is defined by the following likelihood ratio

$$\theta = \frac{P(\chi|\mathbf{\Lambda})}{P(\chi|\mathbf{\Lambda}_0)} \tag{2.3}$$

where χ is the set of the observations, $\mathbf{\Lambda}$ is the target speaker model and $\mathbf{\Lambda}_0$ is the non-target speaker model. At this point, it is important to notice that channel effects are dealt by using the likelihood ratio. As mentioned in the 2.3.3, the estimate of the channel space is the same for the target model $\mathbf{\Lambda}$ and the non-target model $\mathbf{\Lambda}_0$ since the channel space is speaker independent. Thus, the channel component is compensated in the likelihood ratio. Therefore, the decision score is more relevant than the likelihood ratio used in the baseline system.

In order to calculate the decision score θ, the likelihood $P(\chi|\mathbf{\Lambda})$ of an observation χ given a speaker model $\mathbf{\Lambda}$ has to be defined. In our approach, the hyperparameter set $\mathbf{\Lambda}$ models the speaker and channel spaces. As mentioned in 2.1, that holds by assuming that the distributions of the latent variables $\mathbf{z}$ and $\mathbf{x}$ in Equation 2.2 have a standard normal distribution $\mathcal{N}(0,1)$ for utterances coming from same speaker and same channel. Then, the likelihood $P(\chi|\mathbf{\Lambda})$ could be defined by evaluating how far this distribution from the standard normal distribution is. Kenny (2005) defines it by writing the likelihood $P(\chi|\mathbf{\Lambda})$ as a sum of conditional probabilities according to the Bayesian formula

$$P(\chi|\mathbf{\Lambda}) = \int_{\mathbf{z},\mathbf{x}} P(\chi|\mathbf{\Lambda},\mathbf{z},\mathbf{x})P(\mathbf{z},\mathbf{x})\, d\mathbf{z}d\mathbf{x} \tag{2.4}$$

where the marginal probability $P(\mathbf{z},\mathbf{x}) = \mathcal{N}(\mathbf{z},\mathbf{x}|0,I) = \mathcal{N}(\mathbf{z}|0,I)\ \mathcal{N}(\mathbf{x}|0,I)$. The function $\mathcal{N}(.|0,I)$ refers to the Gaussian kernel function which measures the proximity to the standard normal distribution. In practice, this integral calculus cannot be evaluated directly. To avoid this problem, the likelihood function is expressed with a different form by using some algebraic

transformations. Then, the likelihood function is given from the estimate of the distribution of the latent variables $\mathbf{z}$ and $\mathbf{x}$ of the observation χ on the hyperparameter set $\mathbf{\Lambda}$. The details of the likelihood calculus are given in the Appendix B.3.

Given the hyperparameters of the target model, the hyperparameters of the non-target model, and a test utterance, the decision score for speaker verification task could be calculated from the estimate of the distributions of the latent variables $\mathbf{z}$ and $\mathbf{x}$. Now, the estimation of the distributions of the latent variables, of the independent-speaker hyperparameters (non-target speaker model) and of the dependent-speaker hyperparameters (target speaker model) have to be introduced. That is the purpose of the next section where the different algorithms to estimate properly the parameters of the factor analysis model are described.

2.3 Factor analysis parameter estimation

This section describes the methods to estimate the different parameters of the factor analysis approach. First, the estimation of the distributions of the latent variables $\mathbf{z}$ and $\mathbf{x}$ is presented. Then, the speaker-independent hyperparameter estimation is described. Finally, we show how to estimate the speaker-dependent hyperparameters.

2.3.1 Distribution estimation of the latent variables

In the Equation 2.1, the factor analysis is formed by a sum of a speaker-dependent and channel-independent component $\mathbf{s}$ and a speaker-independent and channel-dependent component $\mathbf{c}$. In this section, the estimation of the distributions of the latent variables $\mathbf{s}$ and $\mathbf{c}$ defined in (Kenny et al., 2007) is presented. This one has to be performed on acoustic event observations that is to say cepstral vector observations. Unfortunately, in cepstral vectors the information about speaker and channel effects is hidden, so it is not possible to measure $\mathbf{s}$ and $\mathbf{c}$ directly. Moreover the channel effects cannot be ignored in estimating $\mathbf{s}$ and $\mathbf{s}$ is not known in estimating $\mathbf{c}$. Thus, the distributions of the latent variables $\mathbf{s}$ and $\mathbf{c}$ have to be jointly estimated.

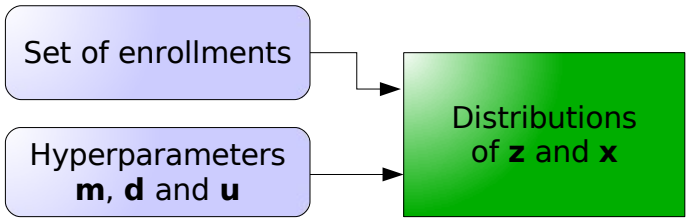

Figure 2.3: Distribution estimation

In order to estimate the joint distribution of the latent variables $\mathbf{s}$ and $\mathbf{c}$, the prior estimate of the speaker and channel space are given by the hyperparameter set $\mathbf{\Lambda} = (\mathbf{m}, \mathbf{d}, \mathbf{u})$. As mentioned above, the hyperparameter set $\mathbf{\Lambda}$ describes the speaker and channel spaces by assuming that the latent variables $\mathbf{z}$ and $\mathbf{x}$ have a standard normal distribution in the Equation 2.2. Now, given this prior estimate and a set of cepstral vectors, the purpose is to jointly reestimate the distribution of the latent variables $\mathbf{z}$ and $\mathbf{x}$ as illustrated on Figure 2.3. That is performed by first calculating the alignment statistics of the cepstral vectors on the GMM centred on $\mathbf{m}$ as

described in Appendix B.1. These alignment statistics enable to study the behaviour of the utterances on the current estimate of the hyperparameter set $\mathbf{\Lambda}$. Then, the joint distributions of the latent variables $\mathbf{z}$ and $\mathbf{x}$ could be estimated as described in Appendix B.2. Since the distributions of $\mathbf{z}$ and $\mathbf{x}$ are assumed to be normally distributed, only the estimations of $E[\mathbf{z}]$, $E[\mathbf{x}]$, $Cov(\mathbf{z}, \mathbf{z})$ and $Cov(\mathbf{x}, \mathbf{x})$ are required.

2.3.2 Speaker-independent hyperparameter estimation

In this section, we want to estimate a speaker-independent factor analysis model from a large set of speech in which many utterances of different speakers are available and in which each speaker is enrolled on several different channels (Kenny, 2005). This speaker-independent model is designed in order to model the non-target speakers (or impostors) and also to have a prior knowledge of the client speakers for the same reasons as mentioned in 1.2. In the factor analysis model, the hyperparameter set $\mathbf{\Lambda}$ characterised the prior knowledge of the speaker and channel dependent random supervector $\mathbf{M}$. That holds by assuming that the latent variables $\mathbf{z}$ and $\mathbf{x}$ have a standard normal distribution. In the previous section, it is shown that given a set of hyperparameters $\mathbf{\Lambda}$ and a set of observations, the distributions of the latent variables $\mathbf{z}$ and $\mathbf{x}$ could be jointly reestimated. In this section, a method is defined to iteratively update the hyperparameter set $\mathbf{\Lambda}$. As described in Figure 2.4, in each iteration the new estimate of $\mathbf{\Lambda}$ is obtained by considering the joint distributions of $\mathbf{z}$ and $\mathbf{x}$. More precisely, the new hyperparameter set $\mathbf{\Lambda}$ is obtained by fitting the new distribution of $\mathbf{z}$ and $\mathbf{x}$ on a standard normal distribution. That is a likelihood maximisation according to the definition given in 2.2 since the distributions of the latent variables $\mathbf{z}$ and $\mathbf{x}$ catch up with a standard normal distribution.

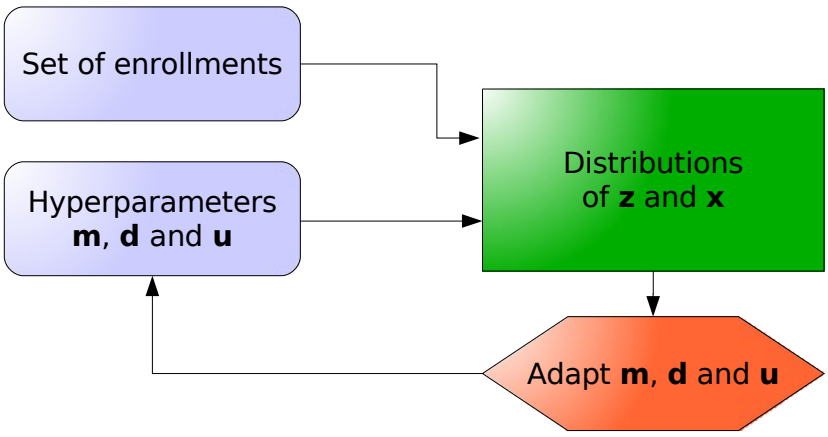

Figure 2.4: Speaker-independent hyperparameter estimation

In the estimation task of speaker-independent hyperparameters, given a hyperparameter set $\mathbf{\Lambda}_0 = (\mathbf{m}_0, \mathbf{d}_0, \mathbf{u}_0)$ and a set of speech utterances χ_s, a new hyperparameter set $\mathbf{\Lambda} = (\mathbf{m}, \mathbf{d}, \mathbf{u})$ has to be estimated. This estimation has to be done such that the total likelihood of the speaker population observations χ_s, on the hyperparameter set is maximised

$$\prod_s P(\chi_s|\mathbf{\Lambda}) > \prod_s P(\chi_s|\mathbf{\Lambda}_0) \tag{2.5}$$

where the likelihood function P is defined in Section 2.2. Such an algorithm could be given by an Expectation-Maximisation (EM) algorithm. An EM algorithm maximises iteratively the

likelihood in two steps, the Expectation Step (E-Step) and the Maximisation Step (M-Step). The E-Step is used to estimate the hidden information of the observations while the M-Step updates the model parameters by considering this hidden information. In our case the E-Step corresponds to the reestimation of the distributions of the latent variables $\mathbf{x}$ and $\mathbf{z}$. The M-Step corresponds to the method which infers the hyperparameter set $\mathbf{\Lambda}$ from the new distributions of the latent variables and therefore maximises the likelihood presented in 2.2. Kenny (2005) proposes two algorithms to perform this transformation known as Maximum Likelihood (ML) estimation and Minimum Divergence (MD) estimation. The details of these algorithms are described in the Appendices B.4 and B.5. According to Kenny (2005), the ML algorithm tends to converge very slowly while the MD algorithm converges much more rapidly. However, the MD algorithm as the property to keep the orientations of the speaker and the channel spaces. Because of this inconvenient property, the MD algorithm converges prematurely to a local maxima solution. In the LRDE Speaker Verification framework, the both ML and MD algorithms are implemented. Then, we have also tried to compare these two algorithms on a 2 minutes speech utterance and with $\mathbf{u} = 0$. Figure 2.5 presents the increase of the likelihood over 50 iterations. As expected, the ML algorithm gives better results, however it is not clear that the MD algorithm converges faster. There are several ways to explain these results. First of all, 2 minutes of speech is not enough to draw any conclusion. In another way, $\mathbf{u}$ was set to zero since there is no channel variability in a single training utterance, indeed our experiment is not in agreement with the experiment of Kenny (2005).

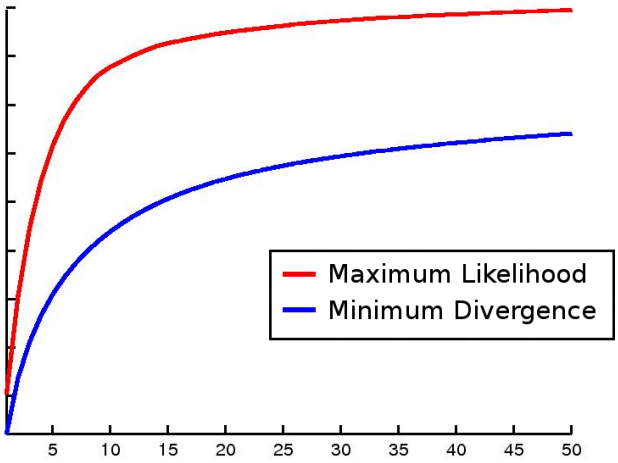

Figure 2.5: Comparison between the Maximum Likelihood estimation, and the Minimum Divergence estimation

An Expectation-Maximisation (EM) algorithm enables to estimate iteratively a set of hyperparameters which describes the speaker and channel spaces. However, the EM algorithms tend to converge to local optima. To avoid such a behaviour, a well-tuned initialisation has to be performed before the application of the EM algorithm. First of all, observe that $E[\mathbf{s}] = \mathbf{m}$ and $Cov(\mathbf{s}, \mathbf{s}) = \mathbf{d}^2$ assuming that $\mathbf{z}$ follows a standard normal distribution. Moreover, the GMM based world model represents the distribution of the acoustic events with a set of mean and covariance matrices. Thus, the initial vector $\mathbf{m}$ is naturally chosen to be the CF-dimensional supervector extracted form the world model. In the same way, the matrix $\mathbf{d}$ is built from the $CF \times CF$ diagonal block matrix $\mathbf{d}^2$ in which each $F \times F$ block matrix is a diagonal covariance matrix of the world model. The matrix $\mathbf{d}^2$ is diagonal and so $\mathbf{d}$ is obtained by taking the square

root of the diagonal elements. Now, observe that $E[\mathbf{c}] = 0$ and $Cov(\mathbf{c}, \mathbf{c}) = \mathbf{u}\mathbf{u}^*$ by assuming that $\mathbf{x}$ follows a standard normal distribution. Then, the $CF \times R_c$ matrix $\mathbf{u}$ is given by the R_c eigenvectors associated to the R_c higher eigenvalues of the $CF \times CF$ channel covariance matrix $\mathbf{u}\mathbf{u}^*$. The matrix $\mathbf{u}\mathbf{u}^*$ is estimated as described in Appendix B.7. Indeed, this provides a good estimate of the matrix $\mathbf{u}$ since these eigenvectors are assumed to span the channel space. More precisely, we assume that these eigenvectors reflect the major variation axes in the supervector space. This assumption holds since the supervectors are built to design speaker models and not to design channel effects. Then, by enrolling same speakers on different channels, the major supervector variabilities come from channel effects and are represented by eigenvectors associated to the higher eigenvalues. This assumption could be easily verified by experiment. In Figure 2.6, the first 100 eigenvalues of a 16896×16896 matrix $\mathbf{u}\mathbf{u}^*$ are plotted on decreasing order. The curve of eigenvalues obtained seems to decrease exponentially to zero. That means that there are few axis of variations which confirms the assumption that $Rc \ll CF$. Moreover, that means that these axis reflect high variations which are caused by channel effects. That is why these axis are used to span the channel space and therefore to construct the matrix $\mathbf{u}$.

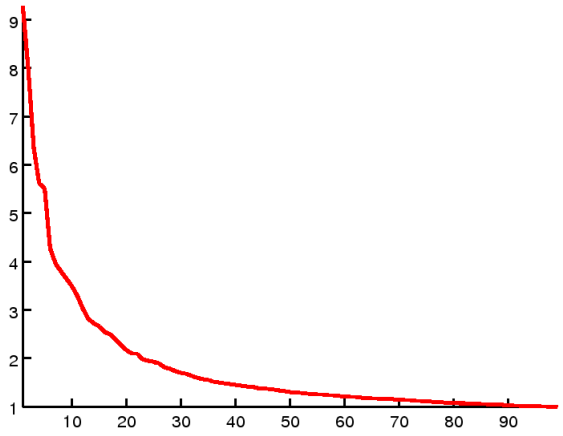

Figure 2.6: Eigenvalues of the channel covariance matrix

The speaker-independent hyperparameter estimation is done by Expectation-Maximisation algorithm from a well-tuned initialisation. The speaker-independent hyperparameters enable to estimate the non-target speaker model. Moreover, speaker-independent hyperparameters give also a prior knowledge of the client models for the speaker-dependent hyperparameter estimation presented in the next section.

2.3.3 Speaker-dependent hyperparameter estimation

In this section, the purpose is to estimate the speaker-dependent hyperparameters in order to model client speaker. Because of the few amount of enrolment data, the speaker-independent hyperparameters are used as prior knowledge of the client speakers. Thus, the method consists in estimating iteratively the speaker-dependent hyperparameter set $\mathbf{\Lambda}$ from the speaker-independent hyperparameter set. However, there is a notable difference here compared to the speaker-independent hyperparameter estimation. Actually, client speaker models are built to

characterise the speaker variability which is independent of the channel variability. Then, there is no reason for changing the channel space and then to reestimate the channel-dependent hyperparameter $\mathbf{u}$. Figure 2.7 summarises the method. As in speaker-independent hyperparameter estimation, during each iteration, the current hyperparameter set $\mathbf{\Lambda}$ and the set of observations are used to estimate the joint distribution of the latent variables $\mathbf{z}$ and $\mathbf{x}$. Then, the hyperparameter $\mathbf{m}$ and $\mathbf{d}$ could be updated in order to fit the speaker space on the observations while the hyperparameter $\mathbf{u}$ is fixed.

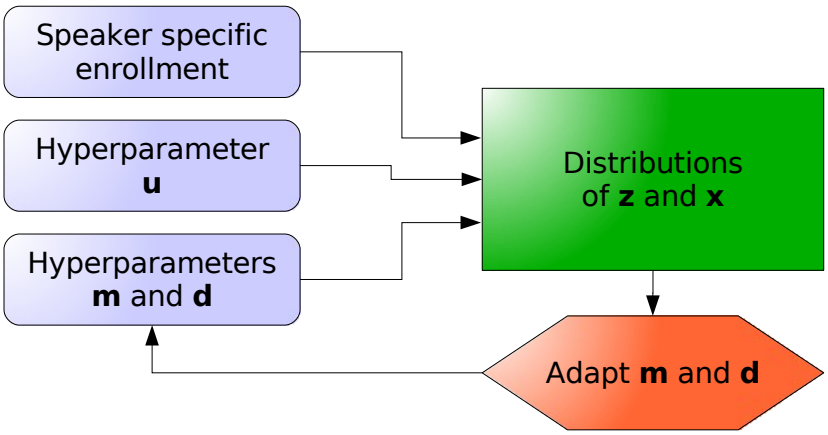

Figure 2.7: Speaker-dependent hyperparameter estimation

In the estimation task of speaker-dependent hyperparameters, given a hyperparameter set $\mathbf{\Lambda}_0 = (\mathbf{m}_0, \mathbf{d}_0, \mathbf{u}_0)$ and a speech enrolment χ, a new hyperparameter set $\mathbf{\Lambda} = (\mathbf{m}, \mathbf{d}, \mathbf{u}_0)$ has to be estimated. This estimation has to be done such that the likelihood of the observation χ on the hyperparameter set is maximised

$$P(\chi|\mathbf{\Lambda}) > P(\chi|\mathbf{\Lambda}_0) \tag{2.6}$$

As the speaker-dependent hyperparameter estimation, such an algorithm could be given by an Expectation-Maximisation (EM) algorithm. However, in this case, Maximum Likelihood (ML) estimation cannot be applied. Actually, the ML algorithm iteratively updates the speaker and channel space orientations. Unfortunately, this change of orientations is operated jointly on speaker and channel spaces. Thus, it is not possible to change the speaker space orientation and to keep the channel space orientation. If such an operation was performed, the two spaces would be defined by two different basis and the Equation 2.1 would have no sense. Thus, since the channel space is fixed in speaker-dependent hyperparameter estimation, ML algorithm cannot be applied.That is why the Minimum Divergence (MD) algorithm have to be used (Kenny, 2005). The details of MD algorithm for speaker-dependent estimation are given in Appendix B.6.

A factor analysis model of speaker-dependent and channel-dependent components enables to consider speaker and channel variabilities. The non-target and target speaker models could be estimated by using Maximum Likelihood estimation and Minimum Divergence estimation. Then, given non-target and target speaker models, given a test utterance, the decision score of the speaker verification task is estimated from the estimate of the joint distribution of the latent variables $\mathbf{z}$ and $\mathbf{x}$.

Chapter 3

Experiments and results

This chapter describes the obtained results given by the experiments realized to test our factor analysis approach. First, the whole step of parametrisation, the used tools and the used corpus to performed the NIST 2006 SRE campaign are described. Then the results are presented and compared to the results obtained from the baseline system.

3.1 Experiments

This section describes the experiments realized to compare the baseline system and the channel compensation system based on a factor analysis model under the classical MAP and eigenchannel MAP assumptions. These experiments are performed on the female trials of the NIST 2006 SRE campaign which consists in training 462 clients and performing 30674 tests.

Front-end processing

The feature extraction is processed by the ALIZE module of the ELISA Consortium (Magrin-Chagnolleau et al., 2001). First, this one sampled the speech signal every 10ms on a 20ms sliding window. From this window, 16 Linear Frequency Cepstral Coefficients (LFCC) are extracted. Then, the 16 first order deltas are computed. A 33-dimensional cepstral vector is obtained by taking the 16 LFCCs, the 16 first order deltas and the delta energy. Finally, normalisation is performed such that the 33 features are centred on zero and scaled to an unitary variance. Furthermore, cepstral vectors corresponding to the silence are skipped by a Bi-Gaussian model of the speech energy.

Probabilistic modelling

The BECARS module of the ENST (Blouet et al., 2004) is used to build probabilistic speaker models which are large GMMs of 512 Gaussians with diagonal covariance matrices. The world model is trained from the Fisher English database Part 1 and 2 and NIST 2003 by applying LBG based Expectation-Maximisation algorithm. The speaker models are estimated by Maximum A Posteriori adaptation from the world model by updating the Gaussian mean vectors only.

Baseline system experiment

In the baseline experiment, the ALIZE module is used to compute the decision score. Each of the 30674 female test utterances is confronted to one target model among the 462 GMM based female client models. At the same time, every test utterance is confronted to the non-target model that is to say the world model. Then, the log likelihood ratio is produced by simply using the probability density functions defined by each GMM.

Channel compensation system experiment

In the channel compensation experiment, the factor analysis models are trained by our factor analysis based LRDE Speaker Verification tool. The implementation details of this tool are given in Appendix A. The speaker-independent hyperparameters are trained from the NIST 2004 SRE database which presents several enrolments for each speaker on different channels. This corpus is used to estimate the channel covariance matrix and also to update iteratively the whole speaker-independent hyperparameters by Maximum Likelihood estimation. There are 100 eigenvectors selected in the channel covariance matrix which span a 100-dimensional channel space. Then, for the 462 female clients, the speaker-dependent hyperparameters are estimated from the speaker-independent hyperparameters by using Minimum Divergence estimation. Finally, our LRDE Factor Analysis tool is used to compute the decision score for the 30674 female trials. The decision score is computed by the log likelihood ratio defined in 2.2 between the target speaker and the non-target speaker.

Step	Input data	Time
Training	7h of speech	10 h
Enrolment	462 clients	20 h
Testing	30674 tests	11 days

Table 3.1: Time requirement

The calculation performances of the LRDE Factor Analysis tool are presented in the Table 3.1. This table shows the results obtained for the three main steps of this speaker verification experiment i.e the training, the enrolment and the testing steps. For each one of these steps, we measured the execution time of our applications on the amount of data given in this table. These execution times are obtained on a 64 bits Dual-Core AMD Opteron 2.2 GHz Processor with 8 GB Memory. We can show that the testing step is the task which required most of the time. Each test is computed in about 31 seconds. Thus, this task has to be done offline. Note that it could be an inconvenience for online speaker verification task.

3.2 Results

This section describes and compares the results obtained by the baseline system based on Gaussian mixture models and the new system based on a factor analysis under the classical MAP and the eigenchannel MAP assumptions.

Evaluation criterions

The results are presented using Decision Error Tradeoff (DET) curves. These enable to compare different speaker verification systems. The performances are represented by plotting the false-alarm probability $P_{FalseAlarm}$ as a function of the miss error probability P_{Miss}. The probability $P_{FalseAlarm}$ corresponds to the probability to reject a target speaker while the probability P_{Miss} corresponds to the probability to accept a non-target speaker (or impostor). Unfortunately, it is generally not possible to minimise $P_{FalseAlarm}$ and P_{Miss} together. Thus, the DET curves underline the tradeoff between the miss errors and the false-alarm errors. Given a false-alarm error rate, we can infer the corresponding miss error rate and vice-versa. So a system is better than another if its DET curve is closer to the origin point than the other DET curve.

Another criterion to compare speaker verification systems is to study specific points of the DET curves. Generally, systems are compared by studying the point where the two probabilities $P_{FalseAlarm}$ and P_{Miss} are equal. This point is characterised by the equal probability known as the Equal Error Rate (EER). However, in the speaker verification task, it is more preferable to reject clients than to accept impostors to intrude in the system. That is given by studying the point which minimised a Decision Cost Function (DCF). The DCF is a function which assigns a cost for each kind of errors. Actually, the DCF promotes the false-alarm errors and penalises the miss errors. Thus the DCF is defined as a weighted sum of miss and false-alarm error probabilities given by

$$C_{DCF} = C_{Miss} \times P_{Miss} \times P_{Target} + C_{FalseAlarm} \times P_{FalseAlarm} \times (1 - P_{Target}) \tag{3.1}$$

where C_{Miss} and $C_{FalseAlarm}$ are respectively the associated costs to a miss and a false-alarm error. The P_{Target} is the probability to have a true access. The costs for the DCF function computation used in the framework of the NIST 2006 SRE are set in Table 3.2.

C_{Miss}	$C_{FalseAlarm}$	P_{target}
10	1	0.01

Table 3.2: Decision cost function parameters of the NIST 2006 SRE

Comparison of the results

The results of our experiments are plotted on Figure 3.1. There are two DET curves representing each of the two experiments, namely the baseline system experiment and the channel compensation system experiment. Note that the baseline curve is closer to the origin point than the channel compensation curve. Unfortunately, that means that the channel compensation system based on factor analysis gives poor results comparing to the baseline system. Actually the baseline system presents an EER of 10.8% while the channel compensation system presents an EER of 22.6%.

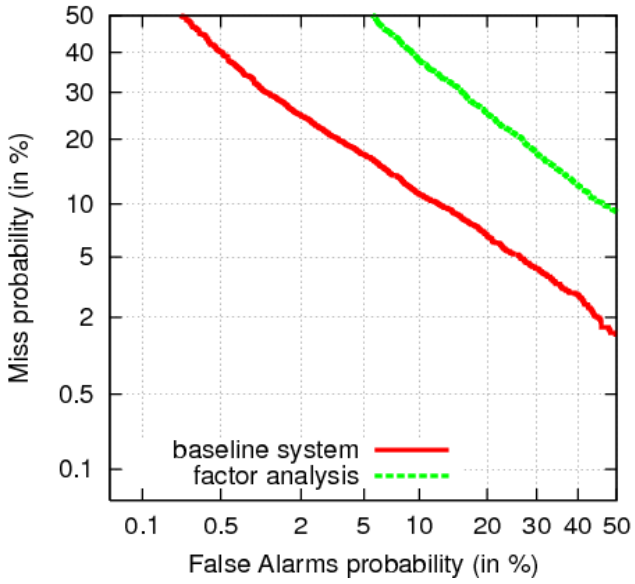

Figure 3.1: DET curves comparing the baseline system and the factor analysis based system

There are several ways to explain these unexpected results. First, notice that today, the factor analysis approaches give the best performances to cope with channel effects. Actually, Shou-Chun (2006) realized an experiment close to ours, with 25 channel factors by applying z-norm score normalisation (Auckenthaler et al., 2000). He obtained an EER of 7.5% on the NIST 2005 SRE campaign. Knowing that this method gives good results, we could assume that the problem arises either from the implementation or from the experiment. As in all numerical implementation, it is difficult to claim that there is no implementation errors since the code is very error prone and that it is hard to check the obtained results. Moreover, in such applications, the calculations easily produce some capacity overflow problems or important accumulations of precision errors because of the internal representation of the real numbers. Yet, Figure 2.5 shows that the likelihood during the Maximum Likelihood (ML) and the Minimum Divergence (MD) estimations increases iteratively. That is a proof that these two estimation algorithms behave correctly. The likelihood calculation seems also consistent. And then the joint distributions of the latent variables appears to be well-estimated. Regarding the experiment, we can notice that is the first experiment done with this new tool. No parameter has been tuned such as the channel space dimension or the stop criterion of the ML and MD algorithms.

Conclusion

This report describes the factor analysis model developed in the LRDE Speaker Verification framework. First, it was shown that the baseline system is not designed to cope with the channel effects. Moreover, the Maximum A Posteriori adaptation does not consider the uncertainty coming from the speaker and the channel variabilities. Then, an approach based on a factor analysis of GMM based supervectors was introduced. This one was developed in order to cope with channel effects and therefore to improve the results of the baseline system. In this approach, the speaker and the channel components are split by assuming that they come from different subspaces namely the speaker and the channel spaces. This assumption is based on the classical MAP and the eigenchannel MAP assumptions. The eigenchannel MAP assumption provides a dimensional reduction of the channel space while the classical MAP does not. In this report, speaker models are based on the hyperparameters of the factor analysis which enable to define a likelihood objective function. Then, the Maximum Likelihood (ML) and the Minimum Divergence (MD) estimations are introduced in order to model target and non-target speakers by optimising this likelihood function. Finally, the decision score is given by the likelihood ratio which deals with the channel effects and indeed performs channel compensation.

Our factor analysis tool is used on the NIST 2006 SRE campaign (female trials). It is compared to a baseline system based on Gaussian mixture models. The obtained results present some unexpected annoyances. Actually, the channel compensation system gives worse results than the baseline system. Our channel compensation gives 20% of Equal Error Rate (ERR) while the baseline system gives 10%. However, today the factor analysis model is known as the most powerful method to perform channel compensation. Then, our future work will consist in studying why we obtained such bad results.

Once the results will go far the 10% of EER, the system will be considered as well tuned and so we will develop new features for the factor analysis model. First, it would be interesting to develop progressive speaker adaptation in order to produce a better model under the classical MAP assumption (Shou-Chun, 2006). Otherwise, we would consider the eigenvoice MAP assumption which reduces the speaker space. Moreover, it would be interesting to implement the likelihood function approximation described by Kenny et al. (2007) which enables to perform online the speaker verification task. Finally, we would like to develop a system mixing factor analysis and support vector machines which know how to discriminate target and non-target speakers by using a supervised learning.

Appendix A

Implementation

The LRDE Speaker Verification Framework (LRDE-SVF) is a C++ framework. It is developed in order to realize speech processing for speaker verification task. I have implemented the factor analysis of the GMM based supervectors as a module of the LRDE-SVF. The LRDE Factor Analysis Tool (LRDE-FAT) directly uses the available and efficient features of the LRDE-SVF i.e the GMM manipulation tools and the powerful feature manager. We have shown that the different algorithms for factor analysis are essentially a matter of matrix manipulation. For this task, I have implemented a light C++ library. This one interfaces with the Fortran BLAS library (Lawson et al., 1979) and the Fortran LAPACK library (Anderson et al., 1999). The purpose of this interface is to give a simple way to manipulate different kinds of matrices with C++ objects and with total performances.

In the LRDE-FAT, there are three main programs named FaTrain, FaAdapt and FaLlr. The FaTrain program estimates the speaker-independent hyperparameter set by using the Maximum Likelihood (ML) estimation. The FaAdapt program enables to adapt the speaker-independent hyperparameters for a given client. Finally, the FaLlr program computes the log likelihood ratio with respect to the likelihood function defined in 2.2. These three programs used the same functionalities given by the LRDE-FAT module. The Figure A.1 describes the structure of the LRDE-FAT module on an UML class diagram. This one shows the different interactions between the main classes of the LRDE-FAT and the LRDE-SVF. We can notice the presence of the Statistics class which compute the Baum-Welch or Viterbi alignment statistics of the cepstral vectors on a GMM. Here, the cepstral vectors and the GMM are given respectively by the FeatureManager class and the GMM class. The Factor class estimates the distributions of the latent variables by considering the alignment statistics and the hyperparameters, given by the Hyperparameters class. The Hyperparameters class represents the hyperparameter set. This one has two methods to perform hyperparameters estimation, which implement respectively the Maximum Likelihood (ML) estimation and the Minimum Divergence (MD) estimation. These two estimation algorithms used the suitable accumulators, given by the class hierarchy of the Accumulators class. Observe also the presence of the likelihood method which estimate the likelihood objective function.

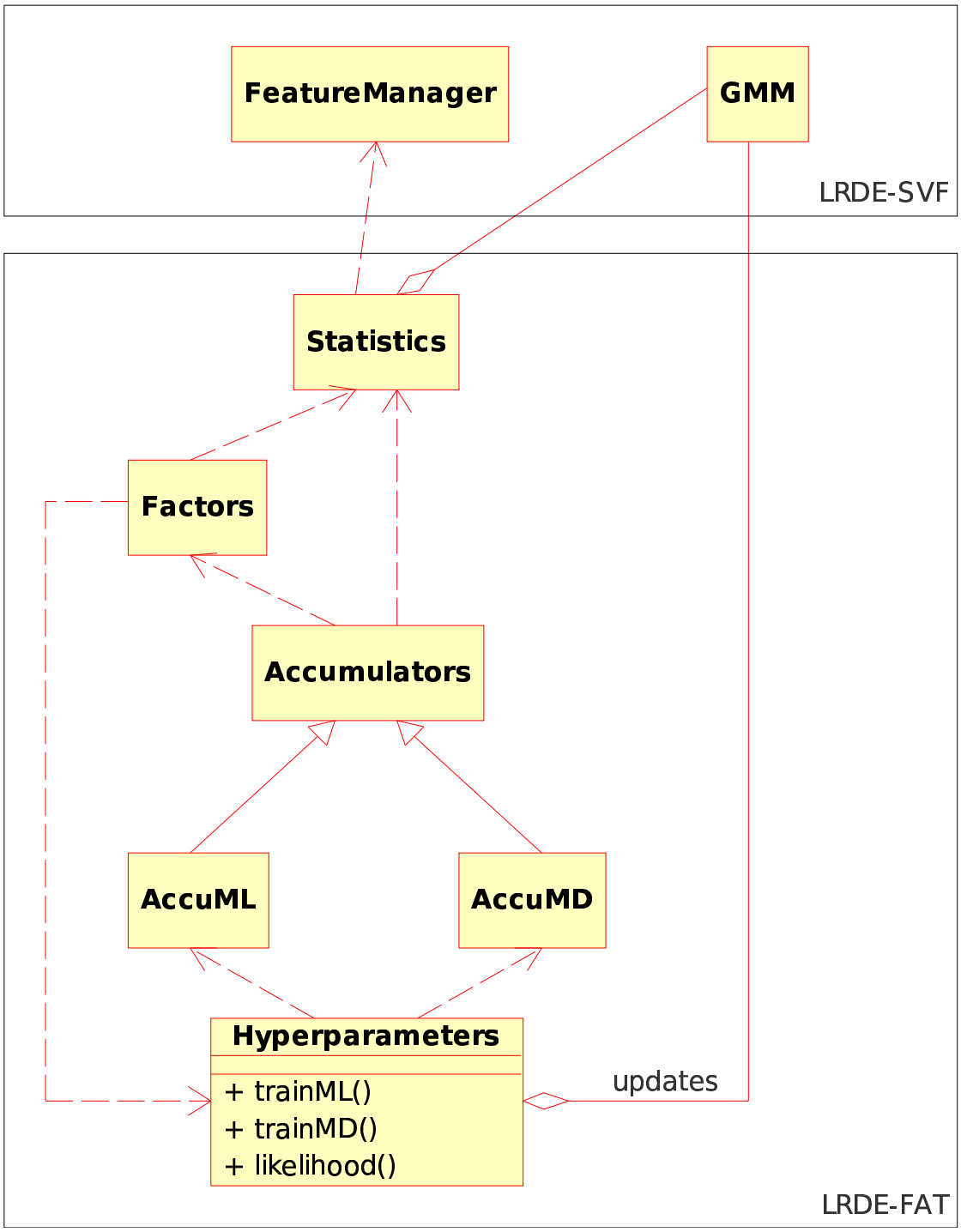

Figure A.1: UML diagram of the LRDE-FAT and its interactions with the LRDE-SVF

Appendix B

Algorithms

This Appendix technically describes the different algorithms used to estimate the parameters of the factor analysis model. Note that most of the following algorithms are extracted from (Kenny, 2005; Kenny et al., 2007). In the following section, the hyperparameter set $\mathbf{\Lambda}$ refers to the extended hyperparameter set $\mathbf{\Lambda} = (\mathbf{m}, \mathbf{d}, \mathbf{u}, \mathbf{\Sigma})$. The $CF \times CF$ diagonal matrix $\mathbf{\Sigma}$ is used to capture the uncertainty independent to $\mathbf{s}$ and $\mathbf{c}$ that is the distribution of the supervectors when there is no speaker and channel variabilities. In practice, the matrix $\mathbf{\Sigma}$ is initially the diagonal matrix whose diagonal block matrices are the covariance matrices of the GMM based world model.

B.1 Alignment statistic estimation

Given a Gaussian Mixture Model with C components $\lambda = (\omega_c, \mu_c, \sigma_c)_{1 \leq c \leq C}$, and given a set of observations $(x_t)_{1 \leq t \leq T}$ which are cepstral vectors, the purpose is to compute the alignment statistics. There are several ways to compute these statistics. Here, the Baum-Welch and the Viterbi statistics are presented. The alignment statistics on a mixture component c are defined by the null N_c, the first F_c and the second S_c order statistics given by

$$N_c = \sum_{t=1}^{T} \rho_t \tag{B.1}$$

$$F_c = \sum_{t=1}^{T} \rho_t \, (x_t - \mu_c) \tag{B.2}$$

$$S_c = diag \left(\sum_{t=1}^{T} \rho_t \, (x_t - \mu_c)(x_t - \mu_c)^* \right) \tag{B.3}$$

$$\tag{B.4}$$

where, in the case of Baum-Welch statistics

$$\rho_t = P(c|x_t, \lambda) = \frac{\omega_c p_c(x_t)}{\sum_{c=1}^{C} \omega_c p_c(x_t)} \tag{B.5}$$

or, in the case of Viterbi statistics

$$\rho_t = \begin{cases} 1 & \text{if } c = \arg\max_{1 \leq c \leq C} \omega_c p_c(x_t) \\ 0 & \text{otherwise} \end{cases} \tag{B.6}$$

Finally, we defined the alignment statistics for the supervector representation by $\mathbf{N}$, $\mathbf{F}$ and $\mathbf{S}$. The $CF \times CF$ diagonal matrix $\mathbf{N}$ is the diagonal matrix where the $F \times F$ diagonal blocks are $N_1 I, ..., N_C I$. The CF-dimensional vector $\mathbf{F}$ is the concatenation of the C vectors F_c. The $CF \times CF$ diagonal matrix $\mathbf{S}$ is the diagonal block matrix where the c-th diagonal block is the matrix S_c.

B.2 Joint distribution estimation

Given an extended hyperparameter set $\mathbf{\Lambda} = (\mathbf{m}, \mathbf{d}, \mathbf{u}, \mathbf{\Sigma})$ and the alignment statistics $\mathbf{N}$ and $\mathbf{F}$ obtained from a set of speech, the purpose is to estimate the joint distribution of the hidden random variables $\mathbf{x}$ and $\mathbf{z}$ that is $E[\mathbf{x}]$, $E[\mathbf{z}]$, $Cov(\mathbf{x}, \mathbf{x})$, $Cov(\mathbf{z}, \mathbf{z})$ and also $Cov(\mathbf{x}, \mathbf{z})$

In order to have an easier representation of the factor analysis decomposition, the random vector $\mathbf{M}$ is expressed as follow

$$\mathbf{M} = \mathbf{m} + \mathbf{U}\mathbf{X}$$

where

$$\mathbf{U} = \begin{pmatrix} \mathbf{u} & \mathbf{d} \end{pmatrix} \quad \text{and} \quad \mathbf{X} = \begin{pmatrix} \mathbf{x} \\ \mathbf{z} \end{pmatrix}$$

Estimate the distribution of the hidden variable $\mathbf{X}$ is a matter of inverting the $R_c + CF$-dimensional matrix $\mathbf{L} = I + \mathbf{U}^* \mathbf{\Sigma}^{-1} \mathbf{N} \mathbf{U}$. Actually, it is proved in (Kenny, 2005) that

$$E[\mathbf{X}] = \mathbf{L}^{-1} \mathbf{U} \mathbf{\Sigma}^{-1} \mathbf{F} \tag{B.7}$$
$$Cov(\mathbf{X}, \mathbf{X}) = \mathbf{L}^{-1} \tag{B.8}$$

The matrix $\mathbf{L}$ can be written as

$$\mathbf{L} = \begin{pmatrix} \alpha & \beta \\ \beta^* & \gamma \end{pmatrix} \tag{B.9}$$

where

$$\alpha = I + \mathbf{u}^* \mathbf{\Sigma}^{-1} \mathbf{N} \mathbf{u} \tag{B.10}$$
$$\beta = \mathbf{u}^* \mathbf{\Sigma}^{-1} \mathbf{N} \mathbf{d} \tag{B.11}$$
$$\gamma = I + \mathbf{\Sigma}^{-1} \mathbf{N} \mathbf{d}^2 \tag{B.12}$$
$$\tag{B.13}$$

So, $\mathbf{L}^{-1}$ can be calculated by the following identity

$$\mathbf{L}^{-1} = \begin{pmatrix} \alpha & \beta \\ \beta^* & \gamma \end{pmatrix}^{-1} = \begin{pmatrix} \zeta^{-1} & -\zeta^{-1} \beta \gamma^{-1} \\ -\gamma^{-1} \beta^* \zeta^{-1} & \gamma^{-1} + \gamma^{-1} \beta^* \zeta^{-1} \beta \gamma^{-1} \end{pmatrix} \tag{B.14}$$

where

$$\zeta = \alpha - \beta \gamma^{-1} \beta^* \tag{B.15}$$

At this step, it is convenient to calculate $|L|$ for future treatments, which could be obtained as:

$$|L| = \begin{vmatrix} \alpha & \beta \\ \beta^* & \gamma \end{vmatrix} = |\zeta||\gamma| \tag{B.16}$$

B.3 Likelihood estimation

Given an extended hyperparameter set $\mathbf{\Lambda} = (\mathbf{m}, \mathbf{d}, \mathbf{u}, \mathbf{\Sigma})$, the alignment statistics N_c, $\mathbf{F}$ and $\mathbf{S}$ obtained from a speech utterance χ, the distributions of the latent variables $\mathbf{z}$ and $\mathbf{x}$ and $|L|$ defined in B.2, the aim is to compute the likelihood $P(\chi|\mathbf{\Lambda})$. In practice it is easier to compute the log likelihood $\log P(\chi|\mathbf{\Lambda})$. In order to have an suitable representation of the factor analysis decomposition, the random vector $\mathbf{M}$ is expressed as follow:

$$\mathbf{M} = \mathbf{m} + \mathbf{U}\mathbf{X}$$

where

$$\mathbf{U} = \begin{pmatrix} \mathbf{u} & \mathbf{d} \end{pmatrix} \quad \text{and} \quad \mathbf{X} = \begin{pmatrix} \mathbf{x} \\ \mathbf{z} \end{pmatrix}$$

Then the log likelihood is defined by Kenny (2005) as

$$\begin{aligned} \log P(\chi|\ \mathbf{\Lambda}) \quad = \quad & \sum_{c=1}^{C} N_c log \frac{1}{(2\pi)^{F/2}|\mathbf{\Sigma}_c|^{1/2}} \\ & -\frac{1}{2} tr(\mathbf{\Sigma}^{-1}\mathbf{S}) \\ & -\frac{1}{2} log|\mathbf{L}| \\ & +\frac{1}{2} E[\mathbf{X}^*]\mathbf{U}^*\mathbf{\Sigma}^{-1}\mathbf{F} \end{aligned}$$

B.4 Maximum Likelihood estimation

This section describes the Maximum Likelihood (ML) estimation algorithm (Kenny, 2005). Given for each channel $a = 1, ..., A$, the alignments statistics $N_c(a)$, $\mathbf{N}(a)$, $\mathbf{F}(a)$ and $\mathbf{S}(a)$ obtained from the training data of the channel a and the distributions of the latent variables $\mathbf{z}(a)$ and $\mathbf{x}(a)$, ML estimates the new extended hyperparameter set $\mathbf{\Lambda} = (\mathbf{m}, \mathbf{d}, \mathbf{u}, \mathbf{\Sigma})$ which maximises the total likelihood.

In order to have an easier representation of the factor analysis decomposition, the random vector $\mathbf{M}(a)$ is expressed as follow:

$$\mathbf{M}(a) = \mathbf{U}\mathbf{X}(a) + \mathbf{d}\mathbf{z}(a)$$

where

$$\mathbf{U} = \begin{pmatrix} \mathbf{u} & \mathbf{m} \end{pmatrix} \quad \text{and} \quad \mathbf{X} = \begin{pmatrix} \mathbf{x}(a) \\ 1 \end{pmatrix}$$

The first step in the ML algorithm is to compute the following accumulators which consider the alignment statistics and the distributions of the latent variables $\mathbf{X}(a)$ and $\mathbf{z}(a)$ over the different channels $a = 1, ..., A$.

$$N_c = \sum_a N_c(a) \tag{B.17}$$

$$\mathfrak{A}_c = \sum_{a=1}^{A} N_c(a) E[\mathbf{X}(a)\mathbf{X}^*(a)] \tag{B.18}$$

$$\mathfrak{B} = \sum_{a=1}^{A} \mathbf{N}(a) E[\mathbf{z}(a)\mathbf{X}^*(a)] \tag{B.19}$$

$$\mathfrak{C} = \sum_{a=1}^{A} \mathbf{F}(a) E[\mathbf{X}^*(a)] \tag{B.20}$$

$$\mathfrak{a} = \sum_{a=1}^{A} diag(\mathbf{N}(a) E[\mathbf{z}(a)\mathbf{z}^*(a)]) \tag{B.21}$$

$$\mathfrak{b} = \sum_{a=1}^{A} diag(\mathbf{F}(a) E[\mathbf{X}^*(a)]) \tag{B.22}$$

The new extended hyperparameter set $\mathbf{\Lambda} = (\mathbf{m}, \mathbf{d}, \mathbf{u}, \mathbf{\Sigma})$ is estimated in two steps. The first step computes the hyperparameters $\mathbf{m}, \mathbf{d}, \mathbf{u}$ which enable to compute $\mathbf{\Sigma}$ in the second step.

- for each mixture components $c = 1, ..., C$ and for each feature components $f = 1, ..., F$, set $i = (c-1)F + f$ and let U_i denote the i-th row of $\mathbf{U}$ and d_i the i-th entry of $\mathbf{d}$. Then U_i and d_i are defined by the equation

$$\begin{pmatrix} U_i & d_i \end{pmatrix} \begin{pmatrix} \mathfrak{A}_c & \mathfrak{B}_i^* \\ \mathfrak{B}_i & \mathfrak{a}_i \end{pmatrix} = \begin{pmatrix} \mathfrak{C}_i & \mathfrak{b}_i \end{pmatrix} \tag{B.23}$$

  where $\mathfrak{B}_i$ is the i-th row of $\mathfrak{B}$, $\mathfrak{a}_i$ is the i-th entry of $\mathfrak{a}$, $\mathfrak{C}_i$ is the i-th row of $\mathfrak{C}$ and $\mathfrak{b}_i$ is the i-th entry of $\mathfrak{b}$. This equation could be resolved by Cholesky factorisation of the left hand-side matrix.

- Let $\mathfrak{M}$ be the $CF \times CF$ diagonal matrix given by

$$\mathfrak{M} = diag(\mathfrak{C}\mathbf{U}^* + \mathfrak{b}\mathbf{d})$$

  Then

$$\mathbf{\Sigma} = \mathbf{N}^{-1} \left(\sum_a \mathbf{S}(a) - \mathfrak{M} \right) \tag{B.24}$$

B.5 Minimum Divergence estimation

This section describes the Minimum Divergence (MD) estimation algorithm (Kenny, 2005). Given for each channel $a = 1, ..., A$, the alignments statistics $\mathbf{N}(a)$, $\mathbf{F}(a)$ and $\mathbf{S}(a)$ obtained from the training data of the channel a and the distributions of the latent variables $\mathbf{z}(a)$ and $\mathbf{x}(a)$, MD estimates the new extended hyperparameter set $\mathbf{\Lambda} = (\mathbf{m}, \mathbf{d}, \mathbf{u}, \mathbf{\Sigma})$ which minimises the divergence.

The first step in the MD algorithm is to compute the following accumulators which consider the alignment statistics and the distributions of the latent variables $\mathbf{z}$ and $\mathbf{x}$ over the different channels $a = 1, ..., A$.

$$\mu_{\mathbf{z}} = \frac{1}{A} \sum_{a=1}^{A} E[\mathbf{z}(a)] \tag{B.25}$$

$$\mu_{\mathbf{xx}} = \frac{1}{A}\sum_{a=1}^{A} E[\mathbf{x}(a)\mathbf{x}^*(a)] \quad \text{(B.26)}$$

$$\mu_{\mathbf{zz}} = \frac{1}{A}\sum_{a=1}^{A} E[\mathbf{z}(a)\mathbf{z}^*(a)] \quad \text{(B.27)}$$

$$S = \sum_{a=1}^{A} \mathbf{S}(a) \quad \text{(B.28)}$$

$$\mathfrak{D} = \sum_{a=1}^{A} diag(\mathbf{F}(a)E[\mathbf{O}^*(a)]) \quad \text{(B.29)}$$

$$\mathfrak{E} = \sum_{a=1}^{A} diag(E[\mathbf{O}(a)\mathbf{O}^*(a)]\mathbf{N}(a)) \quad \text{(B.30)}$$

where $\mathbf{O}(a) = \mathbf{d}_0\mathbf{z}(a) + \mathbf{u}_0\mathbf{x}(a)$

Then, the new extended hyperparameter set $\mathbf{\Lambda} = (\mathbf{m}, \mathbf{d}, \mathbf{u}, \mathbf{\Sigma})$ is estimated as follow:

$$\mathbf{m} = \mathbf{m}_0 + \mathbf{d}_0\mu_z \quad \text{(B.31)}$$
$$\mathbf{d} = \mathbf{d}_0\mathbf{K}_{\mathbf{zz}}^{\mathbf{1/2}} \quad \text{(B.32)}$$
$$\mathbf{u} = \mathbf{u}_0\mathbf{K}_{\mathbf{xx}}^{\mathbf{1/2}} \quad \text{(B.33)}$$
$$\mathbf{\Sigma} = \mathbf{N}^{-1}\left(S - 2\mathfrak{D} + \mathfrak{E}\right) \quad \text{(B.34)}$$

where

$$\mathbf{K}_{\mathbf{zz}} = diag\left(\mu_{zz} - \mu_z\mu_z^*\right) \quad \text{(B.35)}$$
$$\mathbf{K}_{\mathbf{xx}} = \mu_{xx} \quad \text{(B.36)}$$

B.6 Special case for Minimum Divergence estimation

This section describes a special case of Minimum Divergence (MD) estimation algorithm where the channel space is fixed and where there is just one short speech enrolment χ (Kenny, 2005). Given the distributions of the latent variables $\mathbf{z}$ for the speech enrolment χ obtained from the current estimate of the extended hyperparameter set $\mathbf{\Lambda}_0 = (\mathbf{m}_0, \mathbf{d}_0, \mathbf{u}_0, \mathbf{\Sigma}_0)$, MD estimates the new extended hyperparameter set $\mathbf{\Lambda} = (\mathbf{m}, \mathbf{d}, \mathbf{u}_0, \mathbf{\Sigma}_0)$ which minimises the divergence.

The new extended hyperparameter set $\mathbf{\Lambda} = (\mathbf{m}, \mathbf{d}, \mathbf{u}_0, \mathbf{\Sigma}_0)$ is estimated as follow:

$$\mathbf{m} = \mathbf{m}_0 + \mathbf{d}_0\mu_z \quad \text{(B.37)}$$
$$\mathbf{d} = \mathbf{d}_0\mathbf{K}_{\mathbf{zz}}^{\mathbf{1/2}} \quad \text{(B.38)}$$

where

$$\mu_{\mathbf{z}} = E[\mathbf{z}] \quad \text{(B.39)}$$
$$\mathbf{K}_{\mathbf{zz}} = diag\left(Cov(\mathbf{z}, \mathbf{z})\right) \quad \text{(B.40)}$$

B.7 Channel covariance matrix estimation

The channel covariance matrix $\mathbf{uu}^*$ has to be estimated from a database where each of the S speakers has several enrolments coming from A different channels. For each utterance of the speaker s enrolled on the channel a, a GMM based supervector $\mathbf{v}_s^a$ is estimated. Then, the $\mathbf{uu}^*$ is defined by

$$\begin{aligned} \mathbf{uu}^* &= \frac{1}{S}\sum_{s=1}^{S}\frac{1}{A}\sum_{a=1}^{A}(\mathbf{v}_s^a - \mu^a)(\mathbf{v}_s^a - \mu^a)^* \\ \text{where} \quad \mu^a &= \frac{1}{S}\sum_{s=1}^{S}\mathbf{v}_s^a \end{aligned}$$

Appendix C

Bibliography

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., and Sorensen, D. (1999). *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition.

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

Blouet, R., Mokbel, C., Mokbel, H., Sanchez, E., Chollet, G., and Greige, H. (2004). Becars: A free software for speaker verification. *Proc. Odyssey (2004)*.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Kenny, P. (2005). Joint factor analysis of speaker and session variability: Theory and algorithms. Technical report, Montreal, CRIM.

Kenny, P., Boulianne, G., and Dumouchel, P. (2005). Eigenvoice modeling with sparse training data. *IEEE Trans. Speech Audio Processing*, 13.

Kenny, P., Boulianne, G., Ouellet, P., and Dumouchel, P. (2007). Joint factor analysis versus eigenchannels in speaker recognition. *IEEE Trans. Audio, Speech, and Language Processing*.

Lawson, C. L., Hanson, R. J., Kincaid, D. R., and Krogh, F. T. (1979). Basic Linear Algebra Subprograms for Fortran usage. *ACM Transactions on Mathematical Software*, 5(3):308–323.

Magrin-Chagnolleau, I., Gravier, G., and Blouet, R. (2001). Overview of the 2000-2001 ELISA consortium research activities. *Speaker Odyssey Workshop (2001)*.

Reynolds, D., Quatieri, T., and Dunn, R. (2000). Speaker Verification Using Adapted Gaussian Mixture Models. *Digital Signal Processing*, 10(1):3.

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

Shou-Chun, Y. (2006). Speaker adaptation in joint factor analysis based text independent speaker verification. Technical report, Department of Electrical and Computer Engineering, McGill University.