論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	A study on discriminative training techniques for speaker verification
著者(和文)	ﾖﾊﾝﾛﾀﾞﾝ
Author(English)	Johan Rohdin
出典(和文)	学位:博士（学術）, 学位授与機関:東京工業大学, 報告番号:甲第10021号, 授与年月日:2015年11月30日, 学位の種別:課程博士, 審査員:篠田　浩一,徳永　健伸,小池　英樹,村田　剛志,藤井　敦
Citation(English)	Degree:, Conferring organization: Tokyo Institute of Technology, Report number:甲第10021号, Conferred date:2015/11/30, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

A study on discriminative training techniques for speaker verification

Johan Rohdin
09D54070
Supervised by Professor Koichi Shinoda

Department of Computer Science
Graduate School of Information Science and Engineering
Tokyo Institute of Technology

Dissertation submitted to the Tokyo Institute of Technology
for the degree of Doctor of Philosophy

2015

Abstract

One of the fundamental challenges in speaker verification is to separate characteristics of the speech signals that depend on the speaker's identity from characteristics that depend on other factors such as the speaker's emotions, the recording environment, or the transmission channel. This separation process is usually referred to as *session* or *channel* variability compensation.

Over the last decade, complex probabilistic generative models based on factor analysis have been shown to outperform other approaches for channel variability compensation. The state-of-the-art such a model is probabilistic linear discriminant analysis (PLDA). The PLDA parameters are usually optimized by generative training (GT) under the maximum likelihood (ML) criterion. Despite the success of PLDA based systems it is clear that the assumptions behind the PLDA model are inaccurate. This motivates the use of discriminative training (DT) as an alternative or complement to GT. Indeed, several studies have confirmed the effectiveness of DT for PLDA under certain conditions. However, many issues need to be addressed before PLDA based speaker verification can take advantage of the full potential of DT.

The work in this thesis improves DT of PLDA modeling in three aspects. First it proposes a technique for compensating the fact that the training data used in DT of the PLDA model is statistically dependent. Second, it proposes several *constrained* DT schemes in order to avoid the risk of overtraining. Third, it empirically evaluates several different loss functions (i.e., training objectives) as well as proposes a training strategy to deal with the non-convexity of some of the loss functions.

Declaration

I declare that the work presented in thesis is my own, except where otherwise indicated. The work that is my own has not been submitted for a degree anywhere else.

Acknowledgments

I sincerely thank my supervisor, Koichi Shinoda, for his fine guidance and kind support during my studies. I am also grateful for the advice from Sadaoki Furui, Takahiro Shinozaki and Koji Iwano. Needless to say, I would never have made this journey through without the great friendship, feedback and discussions from my dear labmates who cannot be thanked enough for being the amazing people they are. A special thank goes to Sangeeta Biswas who suffered through the speaker verification development and many related hardships with me.

I am thankful to Niko Brümmer, Sandro Cumani and Lukáš Burget, whose insightful works was a great inspiration for the work in this thesis. Further, I would like to thank Lukáš Burget as well as one anonymous reviewer of our Odyssey paper who suggested us to compensate for the statistical dependencies in the training data which, in my opinion, became the most interesting part of this thesis.

I also would like to thank my manager, Ed Whittaker, for giving me enough flexibility in my part-time job to manage my studies.

I have obtained huge amounts of energy from many great metal bands, in particular Iron maiden and Dream theater. Finally and most of all, I would like to thank my lovely family who have been a indispensable support during these tough years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Automatic speaker verification (ASV) refers to the process where a machine judges whether a voice sample is spoken by a claimed identity or not. For each speaker who should be verifiable by an ASV system, *enrollment speech data* must be supplied. In a *trial*, *authentication speech data* and an identity claim are supplied. The task of the ASV system is to compare the authentication data with the enrollment data of the claimed identity and provide a likelihood ratio (LLR) score for the hypotheses, *the identity claim is true* (target trial) and *the identity claim is false* (non-target trial). A decision to accept or reject the claim can then be made based on the LLR score, the prior probability of a true claim, and the cost of false acceptance and false rejection.

The main applications of ASV are access control, surveillance and forensic applications. In access control, ASV is used to authorize access to a resource such as a bank account or a building. In surveillance applications, it is for example used for detecting a wanted criminal in a collection of telephone recordings. In forensics, ASV is used for comparing a voice recording from a crime scene with the voice of a suspect or a victim. ASV can be either *text-dependent* or *text-independent*. In the former, the speakers are supposed to speak a fixed phrase, whereas in the latter they can speak freely. For access control, text-dependent ASV is usually employed due to its superior performance over text-independent ASV. Commercial products offer error rates less than 0.2% using three enrollment recordings and one authen-

tication recording of three seconds each (Agnitio, 2015). In surveillance and forensic applications, it is usually not possible to obtain enrollment and authentication utterances with the same lexical content. Therefore, text-independent ASV must be employed. In text-independent ASV, the error rates are much less impressive than in text-dependent ASV, even for long utterances. The challenges of text-independent ASV are the focus of most speaker verification research these days, including the work in this thesis. Research on text-independent ASV has also been much promoted by the National institute of standards and statistics (NIST) by their series of speaker recognition evaluations (SRE)s (e.g., NIST, 2006, 2008, 2010).

A typical ASV system involves the following steps:

1. **Pre-processing of speech data**: Voice activity detection and extraction of features.
2. **Modeling/classification**: Modeling of speaker and *channel* characteristics, and generating *raw* score for a trial (the higher the score is, the more likely is the trial a target trial).
3. **Post-processing of scores**: Normalization of score distributions and improvements of the score quality, so called *calibration*.

All of the above mentioned components are important for accurate speaker verification. The term *channel* refers broadly to characteristics in the speech signal that are not related to the speaker identity, such as the speaker's emotional state, the recording environment and the transmission channel. The process of separating speaker and channel characteristics in the speech signal is referred to as *channel compensation* and can be seen as the core task of speaker verification. Most efforts for channel compensation are made in the modeling/classification stage (e.g., Kenny et al., 2007) but channel compensation can also be done in the pre-processing (feature) stage (e.g., Hasan and Hansen, 2013) or in the post-processing stage (e.g., Reynolds et al., 2000). Regarding modeling, speaker verification is a complicated problem. Although our final target is a binary decision, we have access to multiple classes, i.e., speakers during the system development. Moreover, we have two kinds of data for model building. First, *training* data which is speech data from a large collection of speakers not involved in the trials. Second, *enrollment* data for each speaker who should be verifiable by the system. In most applications the enrollment data for each speaker is very

limited (usually just one utterance) so it is important to utilize the training data as much as possible. For example, if an authentication utterance of a target-trial is recorded with a different type of microphone than the corresponding enrollment utterance, the claim might be rejected unless the system has learned from the training data that the observed differences are typical for microphone mismatch.

Over the last decade, complex probabilistic generative models based on factor analysis have been shown to outperform other approaches for channel compensation. The first of them was Joint factor analysis (JFA) (Kenny, 2005; Kenny et al., 2007). More recently, probabilistic linear discriminant analysis (PLDA) (Ioffe, 2006; Kenny, 2010), has become the state-of-the-art model for channel compensation. Both JFA and PLDA assume additive speaker and channel components modeled by Gaussian distributions. The main difference between them lies in what kind of features they use. JFA models the mean vectors of Gaussian mixture models from which speech features are assumed to be generated whereas PLDA usually is applied to so-called i-vectors (Dehak et al., 2009a) which are features that represents whole utterances. JFA and PLDA are very effective also for small amounts of enrollment data but requires on the other hand large amounts of training data. Typically more than a thousand speakers with about 10 utterances each on average are used for training. The JFA and PLDA parameters are usually optimized by generative training (GT) under the maximum likelihood (ML) criterion using the speaker IDs as classes.

In order to make the optimal decision (*accept/reject*), it is important that the scores are accurate LLRs. For example, if a system on average gives too large LLRs scores, the scores need to be reduced. Adjusting the scores from the classifier to better serve as LLRs is known as *calibration*. Instead of tuning the decision threshold, calibration adjusts the scores so that the theoretically optimal threshold according to Bayes decision theory really will be optimal. This procedure has several advantages compared to tuning the threshold. Most importantly, it avoids overfitting of the decision threshold to the development data. The most popular approach to calibration is to apply a discriminatively trained (DT) affine transformation of the scores (AT-Cal) (Brümmer, 2010). This method results in substantial improvements for most ASV systems, including PLDA and JFA based ones and is a common component in most state-of-the-art systems.

The fact that AT-Cal improves the performance of a generative probabilistic model such as PLDA is an indication that the assumptions made in the model are not accurate. In fact, the standard pre-processing technique for i-vectors clearly violates the assumptions of the PLDA model. Additional mismatch between the model and the reality have been pointed out by, e.g., Bousquet et al. (2014). Whenever there is an mismatch between the model and the reality, DT may improve the performance. AT-Cal is, however, a very constrained DT scheme. In order to take full advantage of DT, it is preferable apply it on the model itself rather than on its scores. Therefore, a DT scheme that optimizes all the parameters of the PLDA LLR score function (Scr-UC[1]) was proposed by Burget et al. (2011) and Cumani et al. (2011). An important difference between this DT scheme and the previously very popular discriminative framework, GMM-SVM, (Campbell et al., 2006), is that PLDA treats the enrollment utterance and the authentication utterance symmetrically. Given two i-vectors, the PLDA model provides a LLR score for the hypotheses that *the two i-vectors are from the same speaker* and *the two i-vectors are from different speakers*. In GMM-SVM systems on the other hand, one model for each enrollment speaker is created, and an authentication segment is then scored against the model of the claimed identity. GMM-SVM systems therefore suffers severely from insufficient enrollment data. As discussed above, PLDA systems suffers less from this problem. However, DT of PLDA has several other problems.

1.2 Motivations and contributions

While GT of PLDA uses utterances as observations and the speaker IDs as classes, DT uses utterance pairs as observations and *same speaker* or *different speaker* as classes, i.e., it uses trials for training. The trials need to be constructed from the training set. Ideally, we should use all possible pairs of utterances that can be constructed from the training set. However, when a training utterance (or just the same speaker) is used in more than one trial, the trials will be statistically dependent. As a consequence, the *average loss* of the training trials that we use as training objective is no longer the best estimate of the *expected loss* of a test trial, which is what we would

[1] UC refers to *unconstrained*.

like to minimize. In order to compensate for this, we in this thesis, propose to adjust the weights of the training trials in order to obtain the *best linear unbiased estimator* (BLUE) of the *expected loss* of a test trial (Rohdin et al., 2016). This means that we can make better use of available training data.

In general, DT more easily overfits to the training data than GT (Ng and Jordan, 2002) so applying DT on a model that already requires a lot of data such as PLDA, might be risky. This has been confirmed in Cumani and Laface (2014) where Scr-UC was worse than GT as long as the number of training speakers were less than around 1600. Scr-UC easily overfits to the training data because it estimates all the PLDA parameters by DT. AT-Cal on the other hand, estimates only two parameters by DT so the risk of overfitting is small. In order to find the constraints that best avoid overfitting without constraining the model too much, we in this thesis, propose three discriminative training schemes (Rohdin et al., 2016) that are less constrained than Src-UC but more flexible than AT-Cal. The first is a transformation of the PLDA LLR score function having four parameters to be estimated. The second is a scaling of each element in the i-vectors. The third is a training scheme that, like Src-UC, estimates all the parameters of the PLDA LLR score function but preserves some properties of PLDA that are removed by Scr-UC (Rohdin et al., 2014a).

In DT it is desirable that the loss function used in the training objective corresponds to the relevant evaluation criteria of the application. In speaker verification applications, the evaluation criteria is typically given by one cost for false acceptance and one cost for false rejection, together with a prior probability that the claimed identity is true. The cost parameters and the prior naturally depends on the application. The standard loss function (the logistic regression loss) focuses on a broad range of cost parameters. Brümmer and Doddington (2013) proposed a framework for tailoring the loss function to a specific range of cost parameters and applied it to AT-Cal. In this thesis, we apply such *application-specific* loss functions to less constrained DT schemes (Rohdin et al., 2014b). This exploration is important because the merit of application-specific loss functions for less constrained DT schemes than AT-Cal is not clear. Moreover, we investigate how to deal with the non-convexity of the loss functions since this problem can be expected to be larger for models with more parameters.

1.3 Organization of the Thesis

The remainder of this thesis is organized as follows. Chapter 2 presents the technical background of speaker verification. Chapter 3 describes previous works on DT with a focus on PLDA. Chapter 4 describes our baseline system. Chapter 5 presents the proposed method for compensating for statistically dependent training data. Chapter 6 presents the proposed constrained DT schemes. Chapter 7 presents our evaluation of application specific loss functions. Chapter 8 presents experiments on combining our proposed methods. Finally, Chapter 9 concludes this thesis and suggests areas of future works.

Chapter 2

Speaker verification

In this chapter we give an overview of automatic speaker verification (ASV). We start by defining the task of an ASV system in Section 2.1. In Section 2.2, we then introduce the standard evaluation metrics. In Section 2.3, we describe two post-processing steps of the scores, calibration and normalization. In section 2.4, we give a short description of the pre-processing steps of the speech data. Finally, we describe the standard statistical modeling techniques for text-independent ASV with Gaussian mixture models (GMM)s in Section 2.5 and with probabilistic linear discriminant analysis (PLDA) in Section 2.6. Among the topics, the most import for understanding the work in this thesis are, calibration and the evaluation metrics, as these topics are closely related to discriminative training, and PLDA as we are work with discriminative training of different variants of this model.

2.1 Task description

Speaker verification is the process of judging whether a voice sample belongs to a claimed identity or not. The process is called *automatic* speaker verification if it is done solely by a machine, i.e., without human involvement. In this thesis we are only concerned with automatic speaker verification and and we refer to it simply as *speaker verification*.

For each speaker who should be verifiable by a speaker verification system, *enrollment speech data* must be supplied. In a *trial*, *authentication speech data* and an identity claim are supplied. The task of the system is to compare the authentication speech data with the enrollment speech data

and provide a likelihood ratio (LLR) score for the hypotheses, *the identity claim is true* (target trial) and *the identity claim is false* (non-target trial).

We denote the enrollment and the authentication speech data of a trial h as $\boldsymbol{x}_h$. Further, we denote the prior probability for a target trial, P(target), as P_{tar}, the posterior probability as $\text{P}(\text{target}|\boldsymbol{x}_h) = q_h$, and the log-likelihood ratio (LLR) as

$$s_h = \log \frac{\text{P}(\boldsymbol{x}_h|\text{target})}{\text{P}(\boldsymbol{x}_h|\text{non-target})}. \tag{2.1}$$

In speaker verification, we want the system to output LLRs rather than posterior probabilities. This is because

1. The operator of the system should be able to set the prior, P_{tar}.
2. Speaker verification is a binary classification problem.

The first point means that the prior probability of a target trial in the training data, i.e., the ratio of target trials, should not be included in the model. Together with the second point, it means that it is enough if the system outputs the LLR score. By setting P_{tar}, the operator can obtain q_h as

$$q_h = \Big(1 + \exp\Big(-s_h - \frac{\text{P}_{\text{tar}}}{1 - \text{P}_{\text{tar}}}\Big)\Big)^{-1}. \tag{2.2}$$

The data set terminology in speaker verification experiments is sometimes confusing. In this thesis we define the sets as follows:

- **Training set** Used for building speaker independent models.
- **Evaluation set** Used for evaluation. Does not contain any data from the training set. It has two subsets:
  - **Enrollment set** Used to enroll speakers to the system.
  - **Authentication set** Used for authentication.
- **Development set** Used for tuning parameters etc. Does not contain any data from the evaluation set. Ideally (and usually) does not contain any data from the training set. It has two subsets:
  - **Enrollment set** Used to enroll speakers to the system.
  - **Authentication set** Used for authentication.

2.2 Evaluation metrics

In this subsection we describe the evaluation metrics used in this thesis. These evaluation metrics are the most common to use in speaker verification. Understanding the evaluation metrics is important for understanding the training objectives used in discriminative training so we give a quite detailed presentation on this topic.

2.2.1 Detection cost function

When making a decision based on the score from a speaker verification system, it is typically desired to minimize the expected cost of the decision. This is reflected in the *detection cost function* (DCF) used in the NIST evaluations. When the test and enrollment utterances in a trial are from the same speaker, we refer to the trial as a *target trial*, otherwise we refer to it as a *non-target trial*. The DCF measures the cost for an application with a prior probability of a target trial, P_{tar}, and the costs C_{FR} and C_{FA} for false rejection (FR) and false acceptance (FA) respectively. We refer to this set of parameters as an *operating point* (OP). Let $P_{\text{FR}} = P(\text{error}|\text{target})$ and $P_{\text{FA}} = P(\text{error}|\text{non-target})$ be the empirical probabilities for FR and FA, respectively, estimated in the evaluation database. The DCF is then given by

$$\text{DCF} = P_{\text{tar}} C_{\text{FR}} P_{\text{FR}} + (1 - P_{\text{tar}}) C_{\text{FA}} P_{\text{FA}}, \tag{2.3}$$

For the purpose of ranking systems, a scaling of the DCF does not make any difference. Therefore, for system optimization it is equivalent to use

$$\text{DCF}' = P_{\text{eff}} P_{\text{FR}} + (1 - P_{\text{eff}}) P_{\text{FA}}, \tag{2.4}$$

where

$$P_{\text{eff}} = \frac{P_{\text{tar}} C_{\text{FR}}}{P_{\text{tar}} C_{\text{FR}} + (1 - P_{\text{tar}}) C_{\text{FA}}}, \tag{2.5}$$

is known as the *effective prior*. In order to minimize the DCF, the decision threshold for the LLR score should be set to

$$\begin{aligned} \tau &= -\left(\log \frac{P_{\text{tar}}}{1 - P_{\text{tar}}} + \log \frac{C_{\text{FR}}}{C_{\text{FA}}} \right) \\ &= -\log \frac{P_{\text{eff}}}{1 - P_{\text{eff}}}. \end{aligned} \tag{2.6}$$

Therefore, if the speaker verification system outputs scores that can be interpreted as LLRs, the threshold can easily be obtained for any P_{eff}. The cost

obtained by using τ as the decision threshold is called the *actual detection cost* (actDCF) and the cost obtained by the optimal threshold for the evaluation set, is called the *minimum detection cost* (minDCF). If actDCF and minDCF are similar, we say that the LLR scores are well *calibrated* for the particular P_{eff}. Obviously, for an unknown test set, we cannot know the optimal threshold so minDCF is a too optimistic evaluation metric. The actDCF is usually much higher than minDCF but can be improved by *calibrating* the scores. Compared to tuning the decision threshold, calibrating the scores has many advantages which are discussed in Section 2.3.1.

A few points are worth noting with the DCF. First, the DCF assigns the cost zero for correct decisions but in real applications, we may also have non-zero costs (typically negative costs, i.e., rewards) for correct decisions. For the purpose of ranking systems, this is, however, not a limitation. Let us denote the costs for correct decisions C_{TA} and C_{TR} where TA and TR stands for *True acceptance* and *False rejection*, respectively. If we add an offset to the costs of the target trials (C_{FR} and C_{TA}) and another offset to the non-target trials (C_{FA} and C_{TR}), the ranking of systems will not be affected (because these offsets adds the same cost to all systems regardless of their decisions). For system optimization, we can therefore select these offsets so that the costs for correct decisions are zero. Thus, for any binary cost function with parameters C_{FR}, C_{TA}, C_{FA}, C_{TR} and P_{tar}, there is an equivalent DCF with cost equal to 1 for wrong decision and the prior probability P_{eff} for a target trial. Here, *equivalent* means that the systems will be ranked in the same order by the equivalent DCF and the decision threshold will be the same. The actual values of the original binary cost function and the equivalent DCF, will be different, so any physical interpretation of the original binary cost function, e.g., money, is lost.

Second, minimizing the expected cost of a single trial may not be the main target if a set of trials is considered. In some scenarios, minimizing the risk for a very high total cost of all the trials might be more important then minimizing the expected total cost. In such cases decisions should better be made jointly for all trials.

2.2.2 Application independent cost, $\hat{C}_{\text{LLR}}$

The evaluation parameters, P_{tar}, C_{FR} and C_{FA} depends on the application and therefore a DCF can be said to be an *application dependent* evaluation metric. In Brümmer and du Preez (2006), a logarithmic cost function was proposed as an *application independent* evaluation metric. It is given by

$$\hat{C}_{\text{llr}} = \frac{1}{2\log 2} \sum_{t=-1,1} \frac{1}{N_t} \sum_{h:t_h=t} \log(1+\exp(-t_h s_h)), \tag{2.7}$$

where t_h and $\boldsymbol{s}_h$ are the label (1 for target and -1 for non-target) and score for trial h, respectively.

Interpretations

In Brümmer and du Preez (2006), several interpretations of $\hat{C}_{llr}$ that justifies its appellation *application independent* were given:

- **As an average of actDCFs**: With $\text{P}_{\text{tar}} = 0.5$, $C_{\text{FA}} = 1/(1-\zeta)$ and $C_{\text{FR}} = 1/\zeta$ all DCFs can be parameterized by ζ which ranges from 0 to 1 and gives the decision threshold for the posterior, $q = \text{P}(\text{target}|X)$.[1] The cost for a target trial *averaged* over all values of ζ is

$$\begin{aligned}\int_q^1 \frac{1}{\zeta}\text{d}\zeta &= -\log(q) \\ &= \log\Big(1+\exp\Big(-s+\underbrace{\log\frac{1-\text{P}_{\text{tar}}}{\text{P}_{\text{tar}}}}_{=0}\Big)\Big),\end{aligned} \tag{2.8}$$

  since the trial is falsely rejected if $q < \zeta$. Similarly, the cost for a non-target trial averaged over all ζ is

$$\int_0^q \frac{1}{1-\zeta}\text{d}\zeta = -\log(1-q) = \log\Big(1+\exp(s)\Big), \tag{2.9}$$

  since the trial is falsely accepted if $q > \zeta$. Thus $\hat{C}_{\text{llr}}$ is proportional to an average of actDCF at all possible OPs.

[1] $C_{\text{FA}} = 1/(1-\zeta)$ and $C_{\text{FR}} = 1/\zeta$ is not the only parameterization that results in the threshold ζ for the posterior. The main motivation for this particular parameterization is to make the cost function unbounded. This is desired in order to reflect that there is no bound on the possible values of C_{FA} and C_{FR} in applications. See Brümmer and du Preez (2006) for details.

- **Conditional log-likelihood**: If the target and non-target trials are balanced, it is proportional to the negative conditional log-likelihood of the data. Since using that $\mathrm{P}_{\text{tar}} = 0.5$, we have $-\log \prod_h \mathrm{P}(t_h|s_h) = \sum_h \log(1 - \exp(-t_h s_h)) \propto \hat{C}_{\text{llr}}$.

- **Information theoretic interpretation**: Let $\hat{C}_{\text{llr}}^{\text{min}} = \hat{C}_{\text{llr}} - \hat{C}_{\text{llr}}^{\text{cal}}$, where $\hat{C}_{\text{llr}}^{\text{cal}}$ is the cost due to bad calibration (calibration is explained in Section 2.3.1). Then the *empirical mutual information* of a label and a score, $\mathcal{I}(s,t)$ is

$$\mathcal{I}(s,t) = 1 - \hat{C}_{\text{llr}}^{\text{min}} = 1 - \hat{C}_{\text{llr}} + \hat{C}_{\text{llr}}^{\text{cal}}. \tag{2.10}$$

  $\hat{C}_{\text{llr}}^{\text{cal}}$ is the KL-divergence of the posteriors q_h, from posteriors with *perfect calibration*, $\mathring{q}_h$. In other words, $\hat{C}_{\text{llr}}^{\text{cal}}$, is the information lost by using q_h instead of $\mathring{q}_h$.

Minimum $\hat{C}_{\text{llr}}$

Here we give a more detailed explanation of $\hat{C}_{\text{llr}}^{\text{min}}$. As mentioned in Section 2.2.1, the raw scores from a speaker verification system usually have bad calibration, i.e., actDCF is larger than minDCF. Since $\hat{C}_{\text{llr}}$ is an average of actDCF at all possible OPs, it should also be affected by this issue. Intuitively, one could imagine to somehow using an average of minDCF instead of actDCF. This would tell us how good $\hat{C}_{\text{llr}}$ would be if the calibration had been perfect at all OPs. It turns out that this can elegantly be achieved by means of the *pool of adjacent violators* PAV algorithm (Brümmer, 2010). If the PAV algorithm is applied to a set of scores, actDCF will be equal to minDCF for these scores for any OP. Calculating $\hat{C}_{\text{llr}}^{\text{min}}$ on the transformed scores gives $\hat{C}_{\text{llr}}^{\text{min}}$ introduced above.

2.2.3 Equal error rate

Equal error rate (EER) is together with DCF probably the most common evaluation metric in speaker verification. It is defined as the error rate when the threshold is set so that $P_{\text{FR}} = P_{\text{FA}}$. This evaluation metric is less connected to applications than DCF and $\hat{C}_{\text{llr}}$ but we report results for it due to its popularity and ease of interpretation.

2.3 Post-processing of scores

Most speaker verification systems can gain substantial improvements by various post-processing steps of their scores. Two common processes are *score calibration*, where the scores are transformed to better serve as likelihood ratios, and *score normalization* where the score distributions of the enrollment speakers and/or test segments are normalized.

2.3.1 Score calibration

The concept of calibration have already been briefly mentioned in Sections 2.2.1 and 2.2.2 where we said that *minimum* versions of actDCF and $\hat{C}_{\text{llr}}$ is obtained when the scores are calibrated. In this section we will explain the meaning of score calibration more precisely.

Motivation

Score calibration is the process of converting the raw scores from a classifier so that they better serve as posterior probabilities, likelihoods or likelihood ratios. The advantage of having calibrated scores it that their values have an interpretation and that probabilistic rules such Bayes theorem applies to them. For example, with calibrated LLR scores, the theoretically optimal decision threshold, τ, will indeed be optimal so that the actDCF will be (approximately) equal to the minDCF. Score calibration has obtained a lot of attention in the speaker verification community over the last years. The latest NIST SREs have required the participants to submit real-valued scores instead of hard decisions and and the scores have been used as likelihood ratios in the evaluation, e.g., by the checking actDCF. However, even for systems that are designed to output LLR scores such a PLDA, the actDCF and the minDCF usually differs substantially, i.e., the scores have bad *calibration* This may be due to incorrect model assumptions and/or inadequate parameter estimation.

In order to explanation the importance of calibration, we here give an example with posterior probabilities since these are more intuitive than LLRs. Consider a speaker verification system that outputs raw scores, $0 \leq q \leq 1$, where the more the system believes a trial is a target trial, the higher value of q it outputs. However, The fact that this score is between 0 and 1 is

not enough for it serve well as the posterior probability for a target trial. To clarify this, assume that we test the system N times and in $N_{0.7}$ cases, we obtain $0.69 < q \leq 0.71$. If the accuracy for these $N_{0.7}$ cases is around 0.7, we can say that the system is well calibrated for this value of the score (the OP). That is, the scores q_h can serve well as the posterior probability for target trial. If, instead, the accuracy for these $N_{0.7}$ cases had been 0.9, the calibration would have been bad. As mentioned above, the problem with badly calibrated scores is that we do not know how to interpret them and therefore we cannot make good use them. As an example, consider an application were $C_{\text{FR}} = 4C_{\text{FA}}$. In this case, the optimal decision is to choose accept if P(target) > 0.8, otherwise reject. Therefore we would make the wrong decision if the system tells us P(target) $= 0.7$ instead of P(target) $= 0.9$.

The LLR, s_h and the posterior, q_h, are related as

$$s_h = \log \frac{q_h}{1 - q_h} - \log \frac{\text{P}_{\text{tar}}}{1 - \text{P}_{\text{tar}}}. \tag{2.11}$$

Therefore, if the prior is known, obtaining calibrated posteriors is, in principle, equivalent to obtaining calibrated LLRs.

Calibration techniques and proper scoring rules

Both actDCF and $\hat{C}_{\text{llr}}$ are so called *proper scoring rules*. These are cost functions that are sensitive to calibration. Formally, a (cost) function $C(q, t)$ are said to be a binary proper scoring rule if (Buja et al., 2005; Brümmer, 2010)

$$\left\langle C(q,t) \right\rangle_q \leq \left\langle \text{C}(q',t) \right\rangle_q, \tag{2.12}$$

where q and q' are two posterior probabilities and $\langle \cdot \rangle_q$ denotes the expectation using P$(t = 1) = q$. In other words, if a system whenever it outputs probability q' actually has the probability q of succeeding, it will obtain a higher cost than if it had output q. Therefore proper scoring rules penalizes badly *calibrated* probability estimates. To see that they encourages good discriminiation, i.e., target trials should obtain a high value of q and non-target trials a low, assume that at a classifier have perfect calibration, i.e., P(tar$|q) = q$. Then, if we have N trials for which the classifier assigns probability q, (approximately) Nq of these trials will be a target trial. If we then improve the classifier so that it among the N trials, now can distinguish

N_1 trials for which it assigns q_1, and N_2 trials for which it assigns q_2[2] (still having perfect calibration), the new cost is

$$\begin{aligned} N_1\big\langle C(q_1,t)\big\rangle_{q_1} + N_2\big\langle C(q_2,t)\big\rangle_{q_2} &\leq N_1\big\langle C(q,t)\big\rangle_{q_1} + N_2\big\langle C(q,t)\big\rangle_{q_2} \\ &= N_1\big[q_1C(q,t=1) + (1-q_1)C(q,t=-1)\big] \\ &\quad + N_2\big[q_2C(q,t=1) + (1-q_2)C(q,t=-1)\big] \\ &= (N_1+N_2)\big[qC(q,t=1) + (1-q)C(q,t=-1)\big] \\ &= N\big\langle C(q,t)\big\rangle_q, \end{aligned} \tag{2.13}$$

which shows that proper scoring rules encourages not only good calibration but also good discrimination.

The standard approach to score calibration in speaker verification applies a parametric transformation of the score, estimated by DT with proper scoring rules as loss functions (Brümmer et al., 2007; Brümmer, 2010; Brümmer and Doddington, 2013). This approach is described in Section 3 together with other similar techniques for DT in speaker verification.

Relation to tuning the decision threshold

An obvious way to improve DCF is to tune the decision threshold on a development set. This may help improving DCF but does solve the problem of uninterpretable scores. Moreover, this procedure easily overfits to the development set, i.e, the obtained threshold is far from optimal on the test test. Obviously, estimating a threshold for a particular OP is equivalent to estimating an optimal offset of the scores for this OP. Essentially, the PAV algorithm (discussed in Section 2.2.2) does this for all OPs, subject to the constraint that the score transformation should be monotonically increasing. It is therefore a *non-parametric* score calibration approach. While this approach gives *perfect* calibration for the development set, the calibration for other (test) sets is usually not nearly as good, i.e., the method suffers from overfitting. For the approach mentioned above based on a parametric transformations, overfitting is a smaller problem as long as the number of parameters are not too many.

[2] In the best case, we improve the classifier so that the Nq target-trials are assigned $q_1 = 1$ and the $N(1-q)$ non-target trials are assigned $q_2 = 0$. Notice that $Nq = N_1q_1 + N_2q_2$ and $N(1-q) = N_1(1-q_1) + N_2(1-q_2)$.

2.3.2 Score normalization

Score normalization is similar to score calibration in that it also applies an affine transformation to the scores. But in score normalization this transformation depending on the enrollment utterances and/or the test utterance. For this reason, calibration insensitive evaluation metrics such as minDCF may improve. In fact, score normalization is not explicitly designed tot improve calibration but may often do so. For systems based on i-vector + PLDA, it has been shown that score normalization is not effective as long as the i-vectors undergoes certain pre-processing (whitening and length-normalization see Section 2.6.1) (Garcia-Romero and Espy-Wilson, 2011).

Zero-normalization

In zero-normalization (Z-norm) (Li and Porter, 1988) applies a normalization that is specific for each enrollment speaker. In order to do find the normalization parameters, we need a set of utterances from speakers who are not in the the enrollment set. We then score each enrollment speaker against each such *normalization utterance*. Let s_{ij} be the score for enrollment speaker i and normalization utterance j, and let

$$\mu_i = \frac{1}{N}\sum_{j=1}^{N} s_{ij} \tag{2.14}$$

and

$$\sigma_i^2 = \frac{1}{N}\sum_{j=1}^{N}(s_{ij} - \mu_{ij})^2, \tag{2.15}$$

where N is the number of normalization utterances. Then the Z-norm score for speaker i against test utterance j is given by,

$$s_{ij}^z = \frac{s_{ij} - \mu_i}{\sigma_i}. \tag{2.16}$$

That is, affine transformation that is specific to each enrollment utterance. In other words, the aim of z-norm is to ensure that the non-target scores of each enrollment utterance have mean 0 and variance 1. Whether this is desirable depends on the properties of the original (raw) scores. If the distribution of the raw scores are different among the different enrollment

speaker due the speakers having different calibration error, then this procedure can be expected to be beneficial. But if differences in score distributions arise because some speakers really are more difficult to recognize than others, then this should be reflected in theirs scores and this kind of normalization can be harmful.

Test-normalization

Instead of ensure that the non-target scores of each enrollment speaker has mean 0 and variance 1, one can do the same for the test segments. This procedure is called Test-normalization (T-norm) (Auckenthaler et al., 2000). Naturally, Z-norm and T-norm can be combined but notice that the order in which they are applied makes a difference. Thus this procedure is called ZT-norm or TZ-norm depending on which normalization that is applied first.

Symmetric-normalization

While most speaker verification systems treats then enrollment utterance and test utterance differently, some recent methods that uses i-vectors as features (including PLDA) directly aims to answer whether two given i-vectors are from the same speaker or not with caring about which one utterance is from the enrollment and which utterance is from the test phase. With such symmetry between the two utterances, it would make very little sense to use a normalization technique that is not symmetric with respect to the two utterances. For this reason, a symmetric normalization (S-norm) technique was proposed in (Kenny, 2010). Let μ_i, μ_j, σ_i and σ_j for two utterances, i and j, be calculated as in Eqs (2.14) and (2.15). Then the S-norm score for these two utterances is given by

$$s_{ij}^s = \frac{s_{ij} - \mu_i}{\sigma_i} + \frac{s_{ij} - \mu_j}{\sigma_j}. \tag{2.17}$$

2.4 Pre-processing of speech data

2.4.1 Features

The choice of features in speaker verification follows similar considerations as in other pattern recognition problems. Specifically we would like features that (Kinnunen and Li, 2010)

- have large between speaker variability compared to the within speaker variability
- are robust against noise, speakers' emotions and other channel effects
- are frequent and easy to calculate.

Based on their physical interpretation, speech features can be categorized as (Kinnunen and Li, 2010)

- *Short-term spectral:* Captures spectral content in a short frame (typically around 30ms)
- *Voice-source:* For example fundamental frequency
- *Spectro-temporal:* Temporal behavior of spectral features
- *Prosodic:* Pitch contour and rhythm
- *High-level:* Word patterns

Short-term spectral features are the most common features to use in speaker verification systems. In this thesis we use one kind of such features, namely *Perceptual linear prediction* (PLP) features (Hermansky, 1990), along with log-energy. We further applied feature warping (Pelecanos and Sridharan, 2001) which is a normalization technique for speech enhancement. Finally, we the appended the first-order and second-order feature derivatives (Furui, 1981). Feature derivatives is one example of spectro-temporal features.

2.4.2 Voice activity detection

Like most speech processing applications, accurate speaker verification requires that only regions in the audio signal that contains speech are used. A simple approach is to use energy for *voice activity detection* (VAD). However, the energy of the signal can be high also in regions without speech due to noise. An powerful method for noise reduction of noise that is fairly stationary is spectral subtraction (Boll, 1979) which subtract the estimated noise in the frequency domain. Applying spectral subtraction before energy based VAD, has been proven effective on the NIST data (Mak and Yu, 2010). For this purpose, it does not matter much if the signal is distorted so spectral subtraction can be applied more aggressively than in speech enhancement applications.

2.5 Gaussian mixture models

Over the last two decades Gaussian mixture models (GMMs) have been an integral part in text-independent speaker recognition. Further back in time, simpler statistical classifiers, including single Gaussians, had been applied. Let $\mathcal{N}(\boldsymbol{f}|\boldsymbol{\mu}, \Sigma)$ denote the probability density function (PDF) for a multivariate Gaussian distribution. The PDF of a (multivariate) Gaussian mixture model is then given by

$$\mathrm{P}(\boldsymbol{f}) = \sum_{c=1}^{C} \pi_c \mathcal{N}(\boldsymbol{f}|\boldsymbol{\mu}_c, \Sigma_c), \tag{2.18}$$

where C is the number of Gaussian components in the mixture, $\boldsymbol{\mu}_c$ and Σ_c are the mean the covariance of the c-th Gaussian component respectively, and π_c are the so called *mixing coefficients*. In order for the GMM probability density to integrate to 1, it is necessary that

$$\sum_{c=1}^{K} \pi_c = 1. \tag{2.19}$$

In order to guarantee that $P(\boldsymbol{f})$ is positive every where, the mixing coefficients are required to be positive. Together with the constraint in Eq (2.19) this results in (Bishop, 2006, ch. 2.3.9)

$$0 \leq \pi_k \leq 1. \tag{2.20}$$

Gaussian mixture models can model more complex, multimodal, probability densities than single Gaussians. When applied to speech features, they have the intuitive interpretation that the single Gaussians represent different acoustic classes corresponding to specific sounds. Such an acoustic class could for example be a phoneme[3] but in text-independent speaker recognition we do not need to explicitly specify which kind of sound each Gaussian represents. This is in contrast to text-dependent speaker recognition as well speech recognition where a GMM models a part of a specific phoneme or word. In text-independent speaker recognition, we assume that the speech features from each frame are independently generated from a GMM according to Eq. (2.18). The number of Gaussians, C, is usually in the range 500 to 2000 which is substantially smaller than the number of Gaussians

[3] In fact, a phoneme needs more than one Gaussian to be modeled well.

used in a speech recognition system. Given N observations of the feature vector, $\boldsymbol{f}_1 \dots \boldsymbol{f}_N$, the parameters of a GMM with K Gaussian components, $\boldsymbol{\theta}_{GMM} = \{\boldsymbol{\mu}_1 \dots, \boldsymbol{\mu}_K, \Sigma_1 \dots, \Sigma_K, \pi_1 \dots, \pi_K\}$, can be estimated by the (generative) Maximum likelihood criterion, which under the independence assumption becomes

$$\hat{\boldsymbol{\theta}}_{\text{GMM}} = \arg\max_{\boldsymbol{\theta}_{\text{GMM}}} \prod_{i=1}^{N} \text{P}(\boldsymbol{f}_i|\boldsymbol{\theta}_{\text{GMM}}). \tag{2.21}$$

This maximum can be found with the EM algorithm (Dempster et al., 1977). See, e.g., (Bishop, 2006, ch. 9.2.2) for details. It should be noted that the likelihood for a GMM can go to infinity if one of the Gaussian components obtains a mean equal to one of the observations and a covariance matrix whose elements goes to zero. In order to avoid this, the elements of the covariance matrix are usually floored.

For speaker recognition, GMMs were initially applied to speaker identification (Rose and Reynolds, 1990; Reynolds and Rose, 1995). In these systems, a speaker specific GMM, $\boldsymbol{\theta}^s_{\text{GMM}}$, is estimated for each enrolled speaker s. Then, given the features from an utterance, $\boldsymbol{f}_1 \dots \boldsymbol{f}_n$, the speaker with the highest probability is selected, i.e.,

$$\begin{aligned}
\hat{s} &= \arg\max_{s \in \mathcal{S}} \text{P}(\boldsymbol{\theta}^s_{\text{GMM}}|\boldsymbol{f}_1 \dots \boldsymbol{f}_n) \\
&= \arg\max_{s \in \mathcal{S}} \frac{\text{P}(\boldsymbol{f}_1 \dots \boldsymbol{f}_n|\boldsymbol{\theta}^s_{\text{GMM}})\text{P}(s)}{\text{P}(\boldsymbol{f}_1 \dots \boldsymbol{f}_n)} \\
&= \arg\max_{s \in \mathcal{S}} \prod_{i=1}^{n} \text{P}(\boldsymbol{f}_i|\boldsymbol{\theta}^s_{\text{GMM}}),
\end{aligned} \tag{2.22}$$

where $\mathcal{S}$ is the set of enrolled speakers and assuming equal prior $\text{P}(s)$ for each speaker.

2.5.1 GMM-UBM

Despite the fact that speaker verification is a binary classification task, it is actually a more complex problem than speaker identification. This is because the likelihood for the non-target hypothesis, *the authentication sutterance is spoken by another person* is hard to estimate. One approach is to estimate the likelihood $P(\boldsymbol{f}_1 \dots \boldsymbol{f}_n|\mathcal{H}_0)$ by some function, e.g., the average, of the likelihoods from a set of *background speakers* (Reynolds, 1995). A

second approach is to create one model that represents all the speakers that can be expected in the application (except the target speaker) (Reynolds, 1997; Reynolds et al., 2000). Such a model is referred to a Universal background model (UBM). The UBM approach has been shown to outperform the first approach (Reynolds, 1997).

Adaptation

In order to reliably estimate the parameters of a complex probability distribution such as a GMM by ML, large amounts of data is needed. However, in speaker recognition applications, the available enrollment data is usually only between a few seconds and a few minutes which is far from sufficient. A solution to this problem is to, instead, use the data to *adapt* the UBM to the speaker. In other words, when estimating the speaker dependent GMM, some constraints that prevents it from being too different from the UBM are imposed.

The first such adaptation technique was maximum a posteriori (MAP) adaptation which was originally proposed for speech recognition in Gauvain and Lee (1994) and applied to speaker verification in (Reynolds, 1997). This approach relies on the Bayesian framework and uses parameters of the UBM be to design a prior probability distribution for the parameters of the speaker dependent GMM which then are estimated by the maximum a posteriori (MAP) criterion:

$$\hat{\boldsymbol{\theta}}_{\text{GMM}} = \arg\max_{\boldsymbol{\theta}_{\text{GMM}}} \text{P}(\boldsymbol{f}_1 \dots \boldsymbol{f}_N|\boldsymbol{\theta})\text{P}(\boldsymbol{\theta}_{\text{GMM}}) \tag{2.23}$$

$$= \arg\max_{\boldsymbol{\theta}_{\text{GMM}}} \prod_{i=1}^{N} \text{P}(\boldsymbol{f}_i|\boldsymbol{\theta}_{\text{GMM}})\text{P}(\boldsymbol{\theta}_{\text{GMM}}), \tag{2.24}$$

where $\text{P}(\boldsymbol{\theta}_{\text{GMM}})$ is the prior probability for the UBM parameters. For speaker verification, it has been shown that adapting other parameters than the mean vectors of the GMM is not helpful (Reynolds et al., 2000). Let the mean vectors $\boldsymbol{\mu}_c$ be of the GMM be collected in a vector $\boldsymbol{\mu}_{\text{GMM}} = [\boldsymbol{\mu}_1^T \dots \boldsymbol{\mu}_C^T]^T$. Such a vector is known as the *supervector*. When the variance is fixed, the conjugate prior for a Gaussian likelihood is also a Gaussian distribution. By

choosing the prior,

$$\mathrm{P}(\boldsymbol{\mu}_{\mathrm{GMM}}) = \mathcal{N}(\boldsymbol{\mu}_{\mathrm{UBM}}, \frac{1}{\tau}\Sigma), \quad \Sigma = \begin{bmatrix} \Sigma_1 & & \\ & \ddots & \\ & & \Sigma_C \end{bmatrix}, \tag{2.25}$$

we will get a particularly easy formula for MAP adaptation which for component c is given by

$$\hat{\boldsymbol{\mu}}_{\mathrm{MAP}}^{(c)} = \frac{\gamma^{(c)}\hat{\boldsymbol{\mu}}_{\mathrm{ML}}^{(c)} + \tau\boldsymbol{\mu}_{\mathrm{UBM}}^{(c)}}{\gamma^{(c)} + \tau}, \tag{2.26}$$

where $\hat{\boldsymbol{\mu}}_{\mathrm{ML}}^{(c)}$ is the ML estimate for the adaptation data, $\boldsymbol{\mu}_{\mathrm{UBM}}^{(c)}$ and $\gamma^{(c)}$ is the occupancy count for component c. The parameter τ is called the *relevance* factor and is usually tuned on a development set.

2.5.2 GMM-SVM

As explained in Subsection 2.2, the goal of a speaker verification system is to provide a likelihood ratio for the target and non-target hypotheses given the enrollment and authentication speech, $\boldsymbol{x}$. Therefore, if we can correctly estimate $P(\boldsymbol{x}|\text{target})$ and $P(\boldsymbol{x}|\text{non-target})$, there is nothing more to do. We cannot achieve better verification performance in any other way than to use the likelihood ratio. However, in order for the likelihoods to be correctly estimated, the model assumption must be correct (about both speaker and channel effects) and the model parameters must be correctly estimated. Neither of these criteria are fulfilled in reality. As an alternative to *generative classifiers* such as GMMs, it can therefore be worth examine *discriminative classifiers*, i.e., classifiers that directly aims to discriminate between the classes of interest.

One of the most successful discriminative classifiers, is support vector machines (SVM) (Vapnik, 1995). Given a set of feature vectors and a corresponding set of binary labels an SVM finds the hyperplane that best separates the two classes. In testing, features are simply classified based on which side on the hyperplane they are. For a concise explanation of SVMs, see Bishop (2006) and for a thorough tutorial see (Burges, 1998). An important property of SVMs is that they can be formulated in a way so that they only depend on the feature vectors in terms of dot products. During

training they utilizes the dot product of training pairs, and in testing the dot product of the test vector and a subset of the training vectors (the so called support vectors). Via the so called *kernel trick*, the dot product can be replaced with a suitable kernel. The kernel should be a suitable measure of similarity.

The difficulty in using SVMs in speaker recognition is to find a feature that represents whole utterances, which may be of different length, in way that captures speaker characteristics well. Or equivalently, to find a kernel that measures speaker similarity between utterances well. Kernels that compares sequences of features, such a features from each frame of a speech signal, are called sequence kernels. The first example of such a kernel was the Fisher kernel proposed in (Jaakkola and Haussler, 1998). For speaker verification several sequence kernels based on GMMS have been explored (Wan and Renals, 2005; Campbell et al., 2006). The most successful of them is the GMM supervector linear kernel proposed by Campbell et al. (2006) given by

$$K(utt_i, utt_j) = \sum_{c=1}^{C} \left(\sqrt{\pi_c}\Sigma_c^{1/2}\hat{\boldsymbol{\mu}}_{\text{MAP}}^{(c)}\right)^T \left(\sqrt{\pi_c}\Sigma_c^{1/2}\hat{\boldsymbol{\mu}}_{\text{MAP}}^{(c)}\right) \qquad (2.27)$$

2.5.3 Subspace based methods

Although relevance MAP adaptation is more robust to data insufficiency than ML estimation, it is still not particularly effective when the amount of adaptation data is very small i.e., a couple of seconds. In such case, the occupancy counts for many Gaussians will be very small (or even zero if hard counts are used). As can be seen in Eq (2.28), these Gaussians will then remain almost unchanged from the UBM. To overcome this, several adaptation methods that ties the Gaussians together during the adaptation phase have been proposed. In this way, even if a Gaussian has no counts in the adaptation data (or small soft counts) it can be adjusted based on how other Gaussians were adjusted. This can be done by constrained updates of the SI model estimated by ML (Leggetter and Woodland, 1995; Kuhn et al., 1998), or by using the Bayesian framework with a priors that enforce correlations between the Gaussian components (Zavaliagkos et al., 1995; Shinoda and Lee, 2001).

The ideas introduced in Zavaliagkos et al. (1995) and Kuhn et al. (1998)

can be said to have laid the ground for today's i-vector systems. Zavaliagkos et al. (1995) proposed to replace Σ in Eq (2.25) with full a covariance matrix. In other words, the covariance between the elements of the means of all Gaussian components are taken into account. This method is known as *extended* MAP (EMAP). A closed form solution for the MAP estimate is available (see Section 2.5.3). The paper did not detail how to estimate Σ.

Kuhn et al. (1998) proposed to reduce the dimension of the speaker dependent supervectors by PCA, i.e., finding the eigenvectors of Σ. The resulting low-dimensional vectors are referred to as *eigenvoices*. Let the eigenvoices corresponding to the n largest eigenvalues of Σ be the columns in a matrix $\boldsymbol{V}$. The adapted supervector is then given by

$$\hat{\boldsymbol{\mu}}_{\text{EV}} = \boldsymbol{\mu}_{\text{UBM}} + \boldsymbol{V}\boldsymbol{y}, \tag{2.28}$$

where $\boldsymbol{y}$ contains the coefficients for each eigenvoice and are to be estimated in the adaptation. By using only a few eigenvoices, over-fitting to the adaptation data can be avoided. Kuhn et al. (1998) proposed two methods to estimate $\boldsymbol{y}$. The first is to simply estimate $\boldsymbol{\mu}$ by ML without any constraints and then project it, i.e.,

$$\hat{\mathbf{y}}_{proj} = \mathbf{V}^T \left(\hat{\boldsymbol{\mu}}_{\text{ML}} - \boldsymbol{\mu}_{\text{UBM}}\right). \tag{2.29}$$

The second method was to estimate $\boldsymbol{y}$ by ML. It was found that ML estimation outperformed projection in speech recognition.

Eigenvoice MAP

A major problem with the EMAP and the eigenvoice approach to adaptation is how to estimate Σ. In order to estimate it by sample covariance, we need a large number of speaker dependent supervectors. Usually we do not have enough data for each speaker to estimate them reliably. This leads to a catch 22 situation where we need the speaker dependent supervectors in order to estimate Σ but where we also need Σ in order to estimate the supervectors. Moreover, the eigenvoice approach lacks the nice asymptotic properties that MAP adaptation has. The adapted model is restricted to rest in the subspace spanned by the n eigenvoices and will therefore never approach the ML solution when the amount of adaptation data increases. Also, this approach does not take the size of the eigenvalues of Σ into account. The eigenvoices

corresponding to the n largest eigenvalues utilized equally, whereas the remaining eigenvoices are completely ignored.

An elegant solution to these problems was proposed by Kenny et al. (2002, 2005). His approach extends the eigenvoice method in two ways. First, it estimates $\boldsymbol{y}$ by MAP instead of by ML. Therefore it is referred to as *eigenvoice MAP*. It is assumed that the elements of $\boldsymbol{y}$ are independent and that each of them follows a standard normal distribution. This means that the approach is equivalent to using $\Sigma = \boldsymbol{V}\boldsymbol{V}^T$ in EMAP. The length of the eigenvoices are different which means that contrary to the approach by Kuhn et al. (1998), they have different importance. The approach by Kuhn et al. (1998) can be seen as an extreme case of this method where the length of the discarded eigenvoices are zero and the length of the kept eigenvoices is going to infinity. In eigenvoice MAP, the lengths of the eigenvoices are estimated in training. In this way there is no need to reduce the number of eigenvoices (although they are limited by the number of training speakers).

The second extension is that it estimates the training supervectors and $\boldsymbol{V}$ jointly. Joint estimate was also suggested in Nguyen et al. (1999). The training procedure in Kenny et al. (2005) defines the likelihood by by marginalizing over the prior $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{y}|\boldsymbol{0}, \boldsymbol{I})$. The eigenvoices, $\boldsymbol{V}$ is then estimated by by ML, i.e.,

$$\begin{aligned} \hat{\boldsymbol{V}} &= \arg\max_{\boldsymbol{V}} \prod_s \int P(\boldsymbol{F}_{adp}^{(s)}|\boldsymbol{y})\mathcal{N}(\boldsymbol{y}|\boldsymbol{0}, \boldsymbol{I})d\boldsymbol{y} \\ &= \arg\max_{\boldsymbol{V}} \sum_s \log \int P(\boldsymbol{F}_{adp}^{(s)}|\boldsymbol{y})\mathcal{N}(\boldsymbol{y}|\boldsymbol{0}, \boldsymbol{I})d\boldsymbol{y}, \end{aligned} \tag{2.30}$$

where $\boldsymbol{X}_{adp}^{(s)}$ is the adaptation data for speaker s. A local optimum can be found by the EM-algorithm, see Kenny et al. (2005) for details.

Joint factor analysis

The number of eigenvoices in eigenvoice MAP is limited by the number of speakers in the training set which is usually smaller than the dimension of the supervector. In order to obtain the asymptotic properties of standard MAP adaptation one can add an additional term to the model:

$$\hat{\boldsymbol{\mu}}_{\text{VD}} = \boldsymbol{\mu}_{\text{UBM}} + \boldsymbol{V}\boldsymbol{y} + \boldsymbol{D}\boldsymbol{z}, \tag{2.31}$$

where the elements of $\boldsymbol{z}$ follows the standard normal distribution and $\boldsymbol{D}$ is diagonal. Notice that if $\boldsymbol{V} = \boldsymbol{0}$, this results in the standard MAP adaptation.

The various adaptation methods described above were initially developed for speech recognition. In speech recognition we want to adapt the model to fit the utterance of interest but it is irrelevant whether the adaptation compensates for the speaker effect or for channel effects in the utterance. In speaker verification on the other hand, it is crucial to exclude channel effects at some stage in the verification process. Joint factor analysis (JFA) (Kenny, 2005; Kenny et al., 2007) is an extension of the model in Eq (2.31) by letting the terms $\boldsymbol{Vy}$ and $\boldsymbol{Dz}$ be responsible for speaker effects and adding a third term, $\boldsymbol{Ux}$, that is responsible for channel effects. The model is given by

$$\hat{\boldsymbol{\mu}}_{\text{JFA}se} = \boldsymbol{\mu}_{\text{UBM}} + \boldsymbol{V}\boldsymbol{y}_s + \boldsymbol{U}\boldsymbol{x}_{se} + \boldsymbol{D}\boldsymbol{z}_s, \tag{2.32}$$

where the index s indicates speaker and the index e indicates session. Thus the channel effects is assumed to be unique for each session of speaker s and given by $\boldsymbol{Vy}$. Notice that according to this model, the between-speaker covariance is $\boldsymbol{V}\boldsymbol{V}^T + \boldsymbol{D}\boldsymbol{D}^T$ and the within-speaker covariance is $\boldsymbol{U}\boldsymbol{U}^T$. For details of training algorithms, see Kenny (2005); Kenny et al. (2008).

It is not practical to calculate the exact LLR score from JFA but several approximations have been proposed. See Glembek et al. (2009) for and comparison. As an alternative, SVMs can be applied either by using the GMM supervector linear kernel on the channel compensated supervectors, or by using the speaker factors directly with some suitable kernel (Dehak et al., 2009b).

i-Vector

Dehak (2009) showed that the channel factors of JFA contains information about the identity of the speaker. As an alternative approach, Dehak et al. (2009a, 2011) therefore proposed to use the factor analysis framework as a feature extractor. In this system, it is assumed that the GMM-*supervector*, $\boldsymbol{\mu}$, corresponding to an utterance can be modeled as

$$\boldsymbol{\mu} = \boldsymbol{\mu}_{\text{UBM}} + \boldsymbol{T}\boldsymbol{\phi}, \tag{2.33}$$

where $\boldsymbol{\phi}$ is a random vector, and $\boldsymbol{T}$ is the so called *total variability* matrix. Similarly to eigenvoice MAP, it is assumed that $\boldsymbol{\phi}$ follows a standard

normal distribution and its dimension, d, i.e., the rank of $\boldsymbol{T}$, is lower than dimension of $\boldsymbol{\mu}$. The only difference from eigenvoice MAP is that the factors, $\boldsymbol{\phi}$, of a given speaker are different from utterance to utterance whereas in eigenvoice MAP they are forced to be the same for all utterances of a given speaker. In other words, they are capturing both speaker and channel variability. This is taken into account also when estimating $\boldsymbol{T}$ hence the name total variability matrix. Given the speech features of an utterance, the *i-vector*, $\boldsymbol{\omega}$, is the MAP estimate of $\boldsymbol{\phi}$. The i-vector system differs from probabilistic PCA (PPCA) (Tipping and Bishop, 1999) in that $\boldsymbol{\mu}$ itself is not directly observed, but only *indirectly* observed via the features generated from the GMM. Since $\boldsymbol{\mu}$ is not observed, it can be forced to rest in the sub-space spanned by columns of $\boldsymbol{T}$, so the residual term of PPCA is not needed.

An i-vector contains information related to the speaker identity as well as irrelevant *channel* factors such as the speaker's emotions, transmission channels, language, and environmental noise. Channel factors should be removed in order to improve the accuracy of verification.

Currently, PLDA has become one of the state-of-the-art channel compensation techniques in i-vector based speaker verification (Kenny, 2010).

2.6 PLDA

2.6.1 Model

PLDA was originally proposed in image processing for object/face recognition (Ioffe, 2006; Prince and Elder, 2007). Kenny (2010) proposed to use it in speaker verification with i-vectors as features. In its most general form, PLDA assumes that the feature vectors (i-vectors), $\boldsymbol{\omega}$, are generated as:

$$\boldsymbol{\omega} = \boldsymbol{m} + \boldsymbol{V}\boldsymbol{y} + \boldsymbol{U}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{z}, \tag{2.34}$$

where $\boldsymbol{m}$ is the mean of $\boldsymbol{\omega}$, $\boldsymbol{y}$ is a random vector depending on the class, and, $\boldsymbol{x}$ and $\boldsymbol{z}$ are random vectors depending on the *channel*, i.e., they are different from session to session. Contrary to the GMM-supervector, the i-vector is observed, which means that $\boldsymbol{U}$ and $\boldsymbol{D}$ must together span the full i-vector space. Two different PLDA configurations are popular. The configuration suggested by Prince and Elder (2007) constrains both $\boldsymbol{V}$ and $\boldsymbol{U}$ to have a rank lower than d, and $\boldsymbol{D}$ to be diagonal. This configuration is suitable for

large d. This PLDA model is very similar to JFA. The configuration suggested by Ioffe (2006) skips $\boldsymbol{U}$ but puts no constraints on $\boldsymbol{D}$, i.e.,

$$\boldsymbol{\omega} = \boldsymbol{m} + \boldsymbol{V}\boldsymbol{y} + \boldsymbol{D}\boldsymbol{z}. \tag{2.35}$$

This is the most popular configuration in speaker verification (Kenny, 2010; Brümmer and de Villiers, 2010) and we will use it in this study. The speaker matrix, $\boldsymbol{V}$, may have a rank lower than d (Kenny, 2010), or equal to d (Brümmer and de Villiers, 2010) in which case the model is known as the *two-covariance model*.

The original PLDA model (Ioffe, 2006; Prince and Elder, 2007) assumes $\boldsymbol{y}$, $\boldsymbol{x}$ and $\boldsymbol{z}$ follow Gaussian distribution (G-PLDA). However, the elements of the i-vector are, in reality, more heavy-tailed than the Gaussian distribution. Therefore, an extension named heavy-tailed PLDA (HT-PLDA), based on t-distributions, has been proposed (Kenny, 2010). HT-PLDA has much better performance than G-PLDA but is much slower both in the training and the testing phase. Later, normalizing the i-vectors to unit length, was shown to greatly improve the Gaussianity of the i-vectors so that G-PLDA provides similar performance as HT-PLDA (Garcia-Romero and Espy-Wilson, 2011). From here on, we only consider G-PLDA and refer to it as PLDA.

2.6.2 LLR score

Given two i-vectors, $\boldsymbol{\omega_i}$ and $\boldsymbol{\omega_j}$, the LLR score is given by

$$s_{ij} = \log \frac{p\left(\boldsymbol{\omega_i}, \boldsymbol{\omega_j} | \mathcal{H}_s\right)}{p\left(\boldsymbol{\omega_i}, \boldsymbol{\omega_j} | \mathcal{H}_d\right)}, \tag{2.36}$$

where the hypotheses $\mathcal{H}_s$ and $\mathcal{H}_d$ are the following:

$\mathcal{H}_s$: $\boldsymbol{\omega_i}$ and $\boldsymbol{\omega_j}$ are from the same speaker.
$\mathcal{H}_d$: $\boldsymbol{\omega_i}$ and $\boldsymbol{\omega_j}$ are from different speakers.

According to Eq. (2.35), two i-vectors are generated by,

$$\begin{bmatrix} \boldsymbol{\omega}_i \\ \boldsymbol{\omega}_j \end{bmatrix} = \begin{bmatrix} \boldsymbol{m} \\ \boldsymbol{m} \end{bmatrix} + \begin{bmatrix} \boldsymbol{V} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{V} \end{bmatrix} \begin{bmatrix} \boldsymbol{y}_i \\ \boldsymbol{y}_j \end{bmatrix} + \begin{bmatrix} \boldsymbol{D} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{z}_i \\ \boldsymbol{z}_j \end{bmatrix}, \tag{2.37}$$

where the speaker factors, $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ are the same in a target trial but different in a non-target trial. Accordingly, $[\boldsymbol{\omega}_i^T \ \boldsymbol{\omega}_j^T]^T$ follows a multivariate normal distribution. Calculating the mean and covariance of an i-vector

pair in a target and a non-target trial based on Eq. (2.37) and plugging the resulting multivariate normal distributions into Eq. (2.36) results in a closed-form expression of the LLR given by

$$\begin{aligned} s_{ij} = &\boldsymbol{\omega}_i^T \boldsymbol{P} \boldsymbol{\omega}_j + \boldsymbol{\omega}_j^T \boldsymbol{P} \boldsymbol{\omega}_i + \boldsymbol{\omega}_i^T \boldsymbol{Q} \boldsymbol{\omega}_i + \boldsymbol{\omega}_j^T \boldsymbol{Q} \boldsymbol{\omega}_j \\ &+ (\boldsymbol{\omega}_i + \boldsymbol{\omega}_j)^T \boldsymbol{c} + k, \end{aligned} \tag{2.38}$$

where

$$\boldsymbol{P} = \frac{1}{2} \Sigma_{\text{tot}}^{-1} \Sigma_{\text{ac}} (\Sigma_{\text{tot}} - \Sigma_{\text{ac}} \Sigma_{\text{tot}}^{-1} \Sigma_{\text{ac}})^{-1}, \tag{2.39}$$

$$\boldsymbol{Q} = \frac{1}{2} \Sigma_{\text{tot}}^{-1} - (\Sigma_{\text{tot}} - \Sigma_{\text{ac}} \Sigma_{\text{tot}}^{-1} \Sigma_{\text{ac}})^{-1}, \tag{2.40}$$

$$\boldsymbol{c} = -2(\boldsymbol{P} + \boldsymbol{Q})\boldsymbol{m}, \tag{2.41}$$

$$\begin{aligned} k = &\frac{1}{2} (\log |\Sigma_{\text{tot}}| - \log |\Sigma_{\text{tot}} - \Sigma_{\text{ac}} \Sigma_{\text{tot}}^{-1} \Sigma_{\text{ac}}|) \\ &+ \boldsymbol{m}^T 2(\boldsymbol{P} + \boldsymbol{Q})\boldsymbol{m}, \end{aligned} \tag{2.42}$$

and $\Sigma_{\text{ac}} = \boldsymbol{V}\boldsymbol{V}^T$ and $\Sigma_{\text{tot}} = \boldsymbol{V}\boldsymbol{V}^T + \boldsymbol{D}\boldsymbol{D}^T$.

Let $\boldsymbol{\gamma} = [\text{vec}(\boldsymbol{P})^T, \text{vec}(\boldsymbol{Q})^T, \boldsymbol{c}^T, k]^T$, where $\text{vec}(\cdot)$ stacks the columns of a matrix into a column vector, and let

$$\varphi(\boldsymbol{\omega}_i, \boldsymbol{\omega}_j) = \begin{bmatrix} \text{vec}(\boldsymbol{\omega}_i \boldsymbol{\omega}_j^T + \boldsymbol{\omega}_j \boldsymbol{\omega}_i^T) \\ \text{vec}(\boldsymbol{\omega}_i \boldsymbol{\omega}_i^T + \boldsymbol{\omega}_j \boldsymbol{\omega}_j^T) \\ \boldsymbol{\omega}_i + \boldsymbol{\omega}_j \\ 1 \end{bmatrix}. \tag{2.43}$$

Then Eq. (2.38) can be rewritten as (Burget et al., 2011; Cumani et al., 2011)

$$s_{ij} = \boldsymbol{\gamma}^T \varphi(\boldsymbol{\omega}_i, \boldsymbol{\omega}_j). \tag{2.44}$$

In other words, the PLDA LLR score is a linear function of a non-linear feature expansion $\varphi(\boldsymbol{\omega}_i, \boldsymbol{\omega}_j)$ of the two i-vectors.

2.6.3 Parameter estimation by the generative ML criterion

Typically, the PLDA parameters are estimated by the generative maximum likelihood (ML) criterion:

$$[\hat{\boldsymbol{m}}, \hat{\boldsymbol{V}}, \hat{\boldsymbol{D}}] = \arg \max_{[\hat{\boldsymbol{m}}, \hat{\boldsymbol{V}}, \hat{\boldsymbol{D}}]} \prod_{s=1}^{S} \prod_{e=1}^{E_s} (\boldsymbol{\omega}_{se} | \boldsymbol{m}, \boldsymbol{V}, \boldsymbol{D}), \tag{2.45}$$

where index s indicates the speaker, index e indicates the session, S is the number of speakers and E_s is the number of sessions for speaker s. If all speakers have the same number of sessions, an analytic solution exists (Ioffe, 2006), otherwise the EM-algorithm (Brümmer, 2010) can be used. The EM-algorithm is described in Appendix A.1. The PLDA model needs large amounts of training data. Typically around 10k utterances from around 1k speakers are used.

2.6.4 Properties of P and Q

In this subsection we present some properties of the matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ and discusses their impact on the PLDA score function given by Eqs. (2.38) and (2.44). This discussion is important in order to understand the behavior of some the discriminative training schemes discussed later in the thesis. The matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ depends on the PLDA *between-class* covariance matrix, $\boldsymbol{V}\boldsymbol{V}^T$, and the *within-class* covariance matrix, $\boldsymbol{D}\boldsymbol{D}^T$, according to Eqs. (2.39) and (2.40). It is however not immediately apparent what constraints that follows on $\boldsymbol{\gamma}$. In this subsection, the constraints on $\boldsymbol{\gamma}$ are presented, as well as an analysis of their impact on the model.

The matrices, $\boldsymbol{P}$ and $\boldsymbol{Q}$, are symmetric and have the same rank as $\boldsymbol{V}$ (Garcia-Romero and Espy-Wilson, 2011). In addition, it can be shown based on Eq. (2.39) and (2.40), that the matrices, $\boldsymbol{P}$ and $\boldsymbol{Q}$, are constrained as follows:

1. $\boldsymbol{P}$ is positive-(semi)definite.
2. $\boldsymbol{Q}$ is negative-(semi)definite.
3. $\boldsymbol{P} + \boldsymbol{Q}$ is positive-(semi)definite.

For these constraints, *semi* applies when the rank of $\boldsymbol{V}$ is smaller than d. The proofs are given in A.2.

Scr-UC preserves the symmetry of $\boldsymbol{P}$ and $\boldsymbol{Q}$ but relaxes the definiteness constraints. In the remainder of this subsection, the effects of these constraints on the model are analyzed. In Section 6.2, the impact of the constraints is evaluated experimentally.

The first constraint leads to a *directional property*. Consider an i-vector, $\boldsymbol{\omega}$, scored against both $\alpha\boldsymbol{\omega}$ and $-\alpha\boldsymbol{\omega}$, where α is a positive constant. That is, in the first trial, $\boldsymbol{\omega}$ is scored against an i-vector pointing in same direction

and, in the second trial it is scored against an i-vectors pointing in the opposite direction. Let $s(\boldsymbol{\omega}_i, \boldsymbol{\omega}_j) = s_{ij}$ in Eq. (2.38). If the i-vectors are centered around $\boldsymbol{m}$, the difference between the scores of these two trials is

$$s(\boldsymbol{\omega}, \alpha\boldsymbol{\omega}) - s(\boldsymbol{\omega}, -\alpha\boldsymbol{\omega}) = 4\alpha\boldsymbol{\omega}^T\boldsymbol{P}\boldsymbol{\omega}. \tag{2.46}$$

In other words, the score of the *same direction* trial will be guaranteed to be larger than the score of the *different direction* trial if and only if $\boldsymbol{P}$ is positive definite.

The second constraint leads to a *length property*:

$$s(\boldsymbol{\omega}, \boldsymbol{\omega}) > s(\alpha\boldsymbol{\omega}, \frac{1}{\alpha}\boldsymbol{\omega}). \tag{2.47}$$

This property means that two i-vectors of equal length and direction will obtain a higher score than two i-vectors having just equal direction.

From the first and the second constraint, it follows directly that $\boldsymbol{P} - \boldsymbol{Q}$ is positive-definite. Together with the third constraint, this leads to the following properties:

$$s(\boldsymbol{\omega}, \boldsymbol{\omega}) \quad > s(\mathbf{0}, \mathbf{0}), \tag{2.48}$$

$$s(\boldsymbol{\omega}, -\boldsymbol{\omega}) < s(\mathbf{0}, \mathbf{0}). \tag{2.49}$$

This means that any two i-vectors pointing in the same direction obtain a higher score than any two i-vectors pointing in opposite direction. These two properties are therefore a stronger version of the directional property.

Chapter 3

Previous work on discriminative training

We have already introduced one approach to speaker verification that relies on discriminative training, namely the GMM-SVM approach discussed in Section 2.5.2. This approach trains one model for each enrolled speaker. However, In most applications only one, or at most few enrollment utterances with little channel variability are available for each speaker. Training a speaker-specific models therefore suffers severely from (enrollment) data insufficiency. The PLDA model discussed in Section 2.6 does not utilize speaker-specific models and is therefore not affected by this problem.

As mentioned in Section 2.6, normalizing the i-vectors to unit length improves their Gaussianity and substantially improves the performance of PLDA. However, even with length normalization, it is clear that there is still a mismatch between the model assumptions and the training data. Obviously a PLDA model cannot generate i-vectors of a fixed length. Further, it has been shown that for length-normalized i-vectors, the within-class covariance depends strongly on the speaker factors (Bousquet et al., 2014), whereas PLDA assumes the within-class covariance is independent of the speaker factors. Since the model assumptions are not accurate, we cannot expect the parameters obtained by GT to be optimal neither for discriminating between speakers nor for providing well-calibrated LLR scores. Therefore, it might be better to use a DT criterion that directly optimizes the model for providing accurate LLR scores.

In this chapter, we first present three previously proposed DT schemes

that do not utilize speaker-specific models. The first is calibration (and fusion) by means of an affine transformation of the score. The other two are based on the PLDA model. We then discuss the training objectives that have been used. Finally, we discuss three problems in previous approaches. These problems will then be addressed in Chapter 5, 6 and 7, respectively.

3.1 DT schemes

3.1.1 Calibration and fusion

Discriminative training has been proven very effective for score calibration and fusion (Brümmer et al., 2007; Brümmer, 2010) based on affine functions. Given the scores $s^{(1)}, \ldots, s^{(n)}$ from n different systems, the fused and/or calibrated score is given by

$$s(\boldsymbol{w}) = w_0 + w_1 s^{(1)} + \ldots + w_n s^{(n)}, \tag{3.1}$$

where the parameters $\boldsymbol{w} = [w_0, \ldots, w_n]$ need to be estimated. Let $t_h \in [-1, 1]$ be the label of trial h, i.e., it equals 1 if the two utterances are from the same speaker and -1 otherwise. Then $\boldsymbol{w}$ can be estimated by minimizing the objective, $\bar{l}(\boldsymbol{w})$,

$$\bar{l}(\boldsymbol{w}) = \sum_{h:t_h=1} \frac{P_{\text{eff}}}{N_1} l\left(t_h, s_h(\boldsymbol{w}), \tau\right) + \sum_{h:t_h=-1} \frac{1-P_{\text{eff}}}{N_{-1}} l\left(t_h, s_h(\boldsymbol{w}), \tau\right), \tag{3.2}$$

where N_1 and N_{-1} are the numbers of target and non-target trials respectively, and $l\left(t_h, s_h(\boldsymbol{w}), \tau\right)$ is a *loss function* for a trial. There are many possible choices of loss functions. In order order to obtain good calibration, the loss function should be a proper scoring rule. In speaker verification, the most popular choice of such a loss function is the logistic regression loss Brümmer et al. (2007); Brümmer (2010). See Section 3.2 below for details regarding the choice of loss function. By rebalancing the trials, the system is optimized for P_{eff} rather than the prior in the training data. With $n = 1$, we obtain an affine transformation of the scores which has become the standard approach for calibration (Brümmer et al., 2007; Brümmer, 2010). We refer to this method as AT-Cal. An affine transformation results in a very constrained update of the score function that cannot increase the system's ability to discriminate between target and non-target trials, i.e., to reduce

minDCF. On the other hand it can substantially improve calibration, even with quite small amounts of data.

3.1.2 PLDA

Burget et al. (2011) and Cumani et al. (2011) proposed to optimize the parameters $\boldsymbol{\gamma}$ in Eq. (2.44), by minimizing the loss in Eq. (3.2) where s_h is a function of $\boldsymbol{\gamma}$ instead of $\boldsymbol{w}$. Both the logistic regression loss in Eq. (3.3) and the SVM hinge loss were evaluated. We refer to this method as Scr-UC, where UC refers to *unconstrained*. Scr-UC is similar to a DT scheme for JFA, proposed by Burget et al. (2008). In Burget et al. (2011); Cumani et al. (2011), all possible i-vector pairs (typically some hundred millions) were used for training, and efficient calculations of the total loss and its gradient with respect to $\boldsymbol{\gamma}$ were presented. Even so, this method easily overfits to the training data due to the large number of parameters to be estimated. The studies in (Burget et al., 2011; Cumani et al., 2011) therefore added an L2 regularization term, $\rho\|\boldsymbol{\gamma} - \tilde{\boldsymbol{\gamma}}\|^2$, to the training objective.

Borgström and McCree (2013) considered discriminative PLDA training with multiple enrollment sessions. The proposed training scheme applies DT to the model parameters, $\boldsymbol{m}$, $\boldsymbol{V}$ and $\boldsymbol{D}$, rather than the parameters of the LLR score function. The training trials were from either single or multiple enrollment sessions. Only the eigenvalues (which were floored to be positive) or a scaling factor of the covariance matrices were updated by DT. The number of parameters to be estimated are therefore much fewer than in Scr-UC, which reduces the risk of over-training. However, the gradient calculations in this training scheme are only approximate and not as efficient as in Scr-UC where all possible training trials can be used.

3.2 Loss functions

In this section we discuss the choice of loss function more in detail. We will use $\boldsymbol{\theta}$ to denote any parameters to be estimated by DT (e.g., $\boldsymbol{w}$ in AT-Cal or $\boldsymbol{\gamma}$ in Scr-UC). The choice of loss function is an important issue in discriminative training. For fusion, Brümmer et al. (2007) proposed to use the logistic regression loss function given by

$$l\big(t_h, s_h(\boldsymbol{\theta}), \tau\big) = \log\Big(1 + \exp(-t_h\big(s_h(\boldsymbol{\theta}) - \tau\big)\Big). \tag{3.3}$$

As discussed in Section 2.2.2, this is the loss function used in $\hat{C}_{\text{llr}}$ and since it is a proper scoring rule it encourages good calibration. Compared to the standard logistic regression loss, the prior log-odds, $-\tau$, is included. Without it, $s(\boldsymbol{\theta})$ would be trained to be the posterior log-odds, $\log \frac{q_h}{1-q_h}$. In other words, it would include the prior log-odds learned from the data. By including $-\tau$ in the loss function, we therefore ensure that $s(\boldsymbol{\theta})$ is trained to be the LLR (compare Eq. 2.2).

To follow the true spirit of DT, one should use a loss function that is relevant for the intended application. As argued in Chapter 2, real applications require the scores to be calibrated LLRs. Using the logistic regression loss (Eq 3.3) in the training objective given by Eq (3.2) aims producing LLR scores that minimize shifted version of $\hat{C}_{\text{llr}}$. Recall that $\hat{C}_{\text{llr}}$ is an average of actDCFs for all possible OPs according to Eqs (2.8) and (2.8). However, in many application it might be desirable to optimize the system for a narrower range of OPs. In this thesis we refer to such loss functions as *application-specific*. By weighting the OPs differently, one can obtain cost functions that emphasizes on a certain range of OPs, i.e,

$$C_w(q,\ t=\ 1) = \int_q^1 \frac{1}{\zeta} w(\zeta) \mathrm{d}\zeta, \tag{3.4}$$

$$C_w(q, t=-1) = \int_0^q \frac{1}{1-\zeta} w(\zeta) \mathrm{d}\zeta, \tag{3.5}$$

where $C_w(q, t=1)$ and $C_w(q, t=-1)$ are the costs for target trials and non-target trials respectively. The cost functions given by Eq (3.4) results in a proper scoring rule for any normalizable $w(\zeta)$ (Buja et al., 2005). By using $w(\zeta) = 1$, the logistic regression loss ($\hat{C}_{\text{llr}}$) is obtained and by using $w(\zeta)$ equal to Dirac's impulse at τ, we obtain the actDCF for that OP. An elegant framework for choosing $w(\zeta)$ was proposed by Buja et al. (2005) and further developed and applied to AT-Cal in speaker verification by Brümmer and Doddington (2013). In the latter, the Brier loss which emphasizes more narrowly around the targeted OP than the logistic regression loss was very effective. It is given by

$$l_{\text{Brier}}(t_h, s_h, \tau) = \frac{1}{\Big(1 + \exp t_h\big(s_h(\boldsymbol{\theta}) - \tau\big)\Big)^2}. \tag{3.6}$$

The Brier loss corresponds to using $w(\zeta) = 6t(1-t)$.

3.3 Problems in previous approaches

3.3.1 Statistically dependent training data

In GT, each speaker is a class and each utterance is an observation of such a class. However, DT aims directly at improving the LLR score using trials as training data. The features of a trial can be, e.g., a pair of i-vector as in Scr-UC or simply the score as in AT-Cal. The labels of a trial is either *same* or *different* speaker, i.e., there are only two labels. The trials need to be constructed from the available training data, ideally over all possible trials. However, when a training utterance (or just the same speaker) is used in more than one trial, the trials will be statistically dependent which violates the assumptions in the training objective. To put it in a different way, unless the statistical dependencies are taken into account, we have ignored a lot of information about which speaker and utterance ID. In GT on the other hand, no information about the training data is ignored.

3.3.2 Over/under-fitting

In general, DT usually overfits to the training data more easily than GT (Ng and Jordan, 2002). This has been verified empirically also for PLDA. For example, in Cumani and Laface (2014), Scr-UC was worse than GT when the number of training speakers were less than around 1600. AT-Cal on the other hand is so constrained that it may not utilize the full potential of DT. Further, when the parameters γ are optimized directly, rather than the original PLDA model parameters, $\boldsymbol{m}$, $\boldsymbol{V}$ and $\boldsymbol{D}$, the properties of the PLDA model discussed in Section 2.6.4 are not preserved and the consequence of this is not obvious. The methods proposed by Borgström and McCree (2013) avoids these problems but optimizing the parameters $\boldsymbol{m}$, $\boldsymbol{V}$ and $\boldsymbol{D}$ is difficult due to the complex relation between these parameters and the LLR score. The proposed gradient calculations are only approximate and it was not addressed how to efficiently use all possible training trials.

3.3.3 The choice of loss function

The application-specific loss functions has only been applied to AT-Cal. In principle, the application-specific loss functions proposed for score calibration can be used for discriminative training of all the model parameters in

a speaker verification system (Brümmer and Doddington, 2013). However, when training a large number of model parameters, the non-convexity of the application-specific loss functions becomes a serious problem. In addition, when the training focuses only on a small range of operating points, i.e., the subset of the training trials whose score is close to the threshold of the operating point, the risk of over-training may increase. It is therefore necessary to explore if, and how, application-specific loss functions can benefit DT schemes with more parameters to be estimated than AT-Cal.

Chapter 4

Baseline experiments

4.1 i-Vector + PLDA baseline experiments

This section describes our baseline system. It consists a generatively trained PLDA model using i-vectors as features. The feature extraction was made by the HTK toolkit (Young et al., 2006). training of the i-vector extractor as well as i-vector extraction was done with the JFA cookbook (Glembek, 2008). PLDA training and scoring as well as score calibration were done with our own MATLAB implementations. Finally, evaluation was done with the Bosaris toolkit (Brümmer and de Villiers, 2011). We report the results all the evaluation metrics described in Section 2.2. Our main interest are in the calibration-sensitive evaluation metrics, actDCF and $\hat{C}_{\text{llr}}^{\text{min}}$ since this is what matters in applications. Also, it should be noticed that DT aims at reducing calibration-sensitive evaluation metrics. The calibration-insensitive evaluation metrics are only indirectly affected since they cannot be higher than the actual costs.

4.1.1 Experimental set-up

We conducted experiments on the male part of three sets, the NIST SRE 2006 core task (SRE06), NIST SRE 2008 core task condition-6 (SRE08) and NIST SRE 2010 core task condition-5 extended (SRE10). We used SRE06 as the development set for tuning the regularization parameter, ρ, and for the weight-adjustment parameter, α. For some experiments (see subsection 5.3.1), we also used SRE06 for calibration. SRE08 and SRE10 were used as the evaluation sets. A few trials in SRE06 and SRE08 were excluded because

of their inconsistent meta-data. The number of trials were 22123, 12356 and 179338 for SRE06, SRE08 and SRE10 respectively. It should be noted that SRE08 could be too small to give a reliable estimate of actDCF10. The evaluation metrics were calculated with the BOSARIS toolkit (Brümmer and de Villiers, 2011) which uses the PAV algorithm for calculating the minimum version of the evaluation metrics.

For training the UBM and the $\boldsymbol{T}$ matrix, we used NIST SRE 2004 (SRE04), NIST SRE 2005 (SRE05), Switchboard II Phase 1 (SB2P1), Switchboard II Phase 2 (SB2P2), Switchboard II Phase 3 (SB2P3), Switchboard Cellular Part 1 (SBCP1) and Switchboard Cellular Part 2 (SBCP2). For SRE04, we used speech files included in the training lists of one, three, eight and sixteen single-channel conversation sides and in the test list of one single-channel conversation side. For SRE05, we used speech files included in the training lists of one, three and eight two-channel conversation sides and in the test list of one single-channel conversation side. For the Switchboard datasets, we used all non-empty speech files.

For training PLDA models, we used the same data except SB2P1. In addition, from the Switchboard data, we excluded speech distorted by *echo* or *crosstalk* or *background noise* according to the meta-data in the databases. MIXER PIN and PIN were used as unique speaker IDs for NIST SRE and Switchboard datasets respectively. For the files whose MIXER PIN were missing, we used model IDs as speaker IDs. This gave 1153 speakers with in total 9152 utterances.

We used 15 PLP coefficients (Hermansky, 1990) along with log-energy and applied feature warping (Pelecanos and Sridharan, 2001). After that, we appended the first-order and second-order derivatives, resulting in 48 elements per frame. Non-speech parts were then removed by using a spectral subtraction-based voice activity detector (Mak and Yu, 2010). Our UBM had 2048 Gaussian components and d, i.e., the rank of $\boldsymbol{T}$, was set to 400. The i-vectors went through the process of centering, whitening, and length-normalization (Garcia-Romero and Espy-Wilson, 2011).

Generative PLDA training was performed with the EM algorithm (Brümmer, 2010). The number of columns of $\boldsymbol{V}$ was set to d.

Table 4.1: Results of GT in the calibration insensitive evaluation metrics. '%Spkr' is the percentage of the training speakers used for model training. 100% equals 1152 speakers.

Set	% Spkr	minDCF08	minDCF10	$\hat{C}_{\text{llr}}^{\text{min}}$	EER
SRE08	100	0.0250	0.000728	0.175	0.0480
	90	0.0254	0.000713	0.176	0.0497
SRE10	100	0.0101	0.000385	0.079	0.0198
	90	0.0103	0.000403	0.081	0.0201

4.1.2 Experimental results

Table 4.1 shows the results in the calibration insensitive evaluation metrics using 100% and 90% of the training speakers. Overall, reducing the number of training speakers by 10% did not result in any substantial degradation of the performance.

4.2 DT Experiments

In this section we present results for AT-Cal and Scr-UC which will serve as baselines for our later experiments.

4.2.1 Experimental set-up

We followed the experimental set-up given in Section 4.1.1. Additional details for the discriminative training were as follows. We implemented the methods in MATLAB. For optimization, we used the L-BGFS (Liu and Nocedal, 1989) implementation in Schmidt (2012). We used its default stopping criteria and in addition, we stopped the training if no change in minDCF08 had been observed on the development set for 20 iterations. As in Burget et al. (2011) and Cumani et al. (2011), we used all the trials that could be constructed from the training data, except that we excluded target trials where an utterance is scored against itself. The number of unique target trials in the training data was 52,709 and the number of unique non-target trials was 41,822,267. We used the effective prior of SRE08, $P_{\text{eff}} = 0.0917$, to balance target and non-target trials and for setting τ. For Scr-UC we applied L2 regularization. The regularization parameter, ρ, was

Table 4.2: Baseline results in calibration sensitive evaluation metrics. For Scr-UC we applied L2 regularization. The regularization parameter, ρ, was tuned to optimize for $\hat{C}_{\text{llr}}$ on the development set.

Set	Method	actDCF08	actDCF10	$\hat{C}_{\text{llr}}$
SRE08	AT-Cal	0.0256	0.00130	0.201
	Scr-UC	0.0334	0.000876	0.235
SRE10	AT-Cal	0.0143	0.000678	0.100
	Scr-UC	0.0304	0.000916	0.180

Table 4.3: Baseline results in calibration insensitive evaluation metrics. For Scr-UC we applied L2 regularization. The regularization parameter, ρ, was tuned to optimize for $\hat{C}_{\text{llr}}$ on the development set.

Set	Method	minDCF08	minDCF10	EER	$\hat{C}_{\text{llr}}^{\text{min}}$
SRE08	AT-Cal	0.0250	0.000728	0.0480	0.175
	Scr-UC	0.0304	0.000743	0.0564	0.212
SRE10	AT-Cal	0.0101	0.000385	0.0198	0.0788
	Scr-UC	0.0183	0.000598	0.0370	0.1368

optimized over the steps $10^{-3}, 10^{-2}, \ldots, 10^{4}$ on the development set, SRE06. The optimal value was 10^{2}.

4.2.2 Results

The results for the two baselines AT-Cal and Scr-UC in the calibration insensitive calibration metrics actDCF08, actDCF10 and $\hat{C}_{\text{llr}}$ are shown in Table 4.3. As could be expected, AT-Cal was clearly better for this amount of training data. The exception was actDCF10 for SRE08, but as discussed earlier, DCF08 may not be reliably estimated for SRE08. For reference, the results in the calibration insensitive evaluation metrics are shown in 4.3.

Chapter 5

Compensation for statistically dependent training data

In this chapter we address the problem of having statistically dependent training data described in Section 3.3.1. The chapter is divided in three sections. In Section 5.1 we discuss the effect of using statistically dependent training data and propose a compensation method for it. The method is not specific to speaker recognition nor to PLDA but requires knowledge of the pairwise correlations between the losses of all the training trails. In Section 5.2 we propose how to estimate these correlations for the specific statistical dependencies that arise in our speaker verification task, i.e., when the same speakers and utterances are used in more than one training trial. Finally, in Section 5.3, we experimentally evaluate the proposed methods on AT-Cal and Scr-UC.

5.1 The effect of statistically dependent training data

In this subsection we discuss how DT is affected by the use of statistically dependent trials. We argue that using an equal weight for all target trials and another equal weight for all non-target trials in the training objective is not optimal when the trials are statistically dependent. For example, consider the correlations due the same speakers being used in many training trials. If each trial has equal weight, speakers with many trials will influence the model more than speakers with few trials. In order to avoid this *speaker dependency* in the model and make it good for the general population, the

weights for speakers with many trials need to be reduced. The remaining discussion in this subsection does not consider the reason for the statistical dependencies. In section 5.2 we show how to apply the principles discussed in this subsection specifically to the statistical dependencies that arise when all possible training trials are used in DT of speaker verification systems.

A trial consists of a label $T \in [1, -1]$ and two i-vectors $\boldsymbol{\Omega}_i$ and $\boldsymbol{\Omega}_j$. Here, we use upper case letters to denote that we treat these variables as random variables. We collect the i-vector pair of a trial in a vector denoted $\boldsymbol{\Omega}^{(p)} = [\boldsymbol{\Omega}_i^T, \boldsymbol{\Omega}_j^T]^T$. The loss of a trial, $L(\boldsymbol{\theta}) = l\big(T, s(\boldsymbol{\theta}, \boldsymbol{\Omega}^{(p)})\big)$, is then also a random variable. The training trials are observations of these random variables. Analogously, we use $\bar{l}(\boldsymbol{\theta})$ to denote the average loss of an observed set of training trials as in Eq. (3.2), and $\bar{L}(\boldsymbol{\theta})$ to be the corresponding random variable, i.e., the average loss of a set of trials treated as random variables. The *expected loss* of a single trial is given by

$$\begin{aligned} \mathrm{E}_{T,\boldsymbol{\Omega}^{(p)}} L(\boldsymbol{\theta}) &= \mathrm{E}_{T,\boldsymbol{\Omega}^{(p)}} \big(l(\boldsymbol{\theta}, T, \boldsymbol{\Omega}^{(p)})\big) \\ &= \sum_{T=-1,1} \mathrm{P}(T) \int_{\boldsymbol{R}^{2d}} l\big(T, s(\boldsymbol{\theta}, \boldsymbol{\Omega}^{(p)})\big) \mathrm{P}(\boldsymbol{\Omega}^{(p)}|T) \mathrm{d}^{2d}\boldsymbol{\Omega}^{(p)} \end{aligned} \tag{5.1}$$

where $\mathrm{P}(T = 1) = P_{\text{eff}}$, $\mathrm{P}(T = -1) = 1 - P_{\text{eff}}$ and $P(\boldsymbol{\Omega}^{(p)}|T)$ is the probability density function for the i-vector pair conditioned on the trial label. Discriminative training aims to find the $\boldsymbol{\theta}$ that minimizes $\mathrm{E}_{T,\boldsymbol{\Omega}^{(p)}} \big(L(\boldsymbol{\theta})\big)$ by minimizing $\bar{l}(\boldsymbol{\theta})$. In order for this approach to be successful, $\bar{L}(\boldsymbol{\theta})$ must be a good estimator of $E_{T,\boldsymbol{\Omega}^{(p)}} \big(L(\boldsymbol{\theta})\big)$ for each value of $\boldsymbol{\theta}$.

Let us generalize the DT objective as

$$\hat{L}(\boldsymbol{\theta}) = \tilde{P}_{\text{eff}} \hat{L}_1(\boldsymbol{\theta}) + (1 - \tilde{P}_{\text{eff}}) \hat{L}_{-1}(\boldsymbol{\theta}), \tag{5.2}$$

where $0 \leq \tilde{P}_{\text{eff}} \leq 1$,

$$\hat{L}_1(\boldsymbol{\theta}) = \sum_{h:t_h=1} \beta_h l\left(t_h, \boldsymbol{\theta}, \boldsymbol{\Omega}_h^{(p)}\right), \tag{5.3}$$

$$\hat{L}_{-1}(\boldsymbol{\theta}) = \sum_{h:t_h=-1} \beta_h l\left(t_h, \boldsymbol{\theta}, \boldsymbol{\Omega}_h^{(p)}\right), \tag{5.4}$$

and

$$\sum_{h:t_h=1} \beta_h = \sum_{h:t_h=-1} \beta_h = 1. \tag{5.5}$$

Here $\hat{L}_1(\boldsymbol{\theta})$ and $\hat{L}_{-1}(\boldsymbol{\theta})$ are *estimators* of the expected loss of a target and non-target trial respectively, and β_h is the weight for trial h. The expected

loss of a trial with label t, $\mathrm{E}_{\mathbf{\Omega}^{(p)}|t}(L(\boldsymbol{\theta}))$, is not affected by the fact that the trials are statistically dependent. As long as Eq. (5.5) is fulfilled, $\tilde{P}_{\text{eff}} = P_{\text{eff}}$ therefore gives an unbiased estimate of the expected loss, $E_{T,\mathbf{\Omega}^{(p)}}\big(L(\boldsymbol{\theta})\big)$ (for any $\boldsymbol{\theta}$). In addition, we propose to adjust the trial weights, β_h, so that the variances of $\hat{L}_1(\boldsymbol{\theta})$ and $\hat{L}_{-1}(\boldsymbol{\theta})$ is minimized. This gives the best linear unbiased estimator (BLUE) (Kay, 1993, ch. 6) of the expected loss. From here on, we use $t \in [-1, 1]$ also as a suffix to indicate target or non-target trial. Let the i-vector pairs, $\mathbf{\Omega}_h^{(p)}$, of the training trials with label t be collected in a vector $\vec{\mathbf{\Omega}}_t \in \mathbb{R}^{2dN_t}$. Further, let the weights for the corresponding trials be collected in a vector $\boldsymbol{\beta}_t \in \mathbb{R}^{N_t}$, and let $\Sigma_t \in \mathbb{R}^{N_t \times N_t}$ be the covariance matrix for the losses of these trials.[1] Then, $\text{var}\left[\hat{L}_t\left(\boldsymbol{\theta}, \vec{\mathbf{\Omega}}_t\right)\right]$ is given by

$$\begin{aligned}
&\mathrm{E}_{\vec{\Omega}_t|\mathcal{I}_t}\left(\hat{L}_t(\boldsymbol{\theta}, \vec{\mathbf{\Omega}}_t) - \mathrm{E}_{\vec{\Omega}_t|\mathcal{I}_t}\hat{L}_t(\boldsymbol{\theta}, \vec{\mathbf{\Omega}}_t)\right)^2 \\
&\quad = \mathrm{E}_{\vec{\Omega}_t|\mathcal{I}_t}\left(\sum_{h:t_h=t} \beta_h l\left(t, \boldsymbol{\theta}, \mathbf{\Omega}_h^{(p)}\right) - \mathrm{E}_{\mathbf{\Omega}^{(p)}|t} L\left(\boldsymbol{\theta}\right)\right)^2 \\
&\quad = \boldsymbol{\beta}_t^T \Sigma_t \boldsymbol{\beta}_t, \qquad (5.6)
\end{aligned}$$

where $\mathcal{I}_t$ denotes any information about how $\vec{\mathbf{\Omega}}_t$ is generated that affects Σ_t.

Previous studies have set β_h to $1/N_1$ for the target trials and $1/N_{-1}$ for the non-target trials. From Eq. (5.6) it is clear that when Σ_t is diagonal whose all elements are equal, this choice of β_h is optimal and results in the well-known formula for the variance of the sample mean of IID variables. However, when the trials are correlated, this choice of β_h is not optimal. By using a Lagrange multiplier to enforce the constraint in Eq. (5.5) it can be shown that, as long as Σ_t is non-singular,[2] the minimizer is given by,

$$\boldsymbol{\beta}_t = \frac{\Sigma_t^{-1}\mathbf{1}}{\mathbf{1}^T\Sigma_t^{-1}\mathbf{1}}, \qquad (5.7)$$

where $\mathbf{1}$ is a column-vector of length N_t whose all elements equal 1.

According to Eq. (5.7), the optimal $\boldsymbol{\beta}_t$ is not affected by a scaling of the covariance matrices. Since all target/non-target trial losses have the

[1] For simplicity, we do not consider the statistical dependencies between a target trial and a non-target trial in this study.

[2] A singular Σ would mean that the losses of two trials have correlation 1 or that the variance of a trial loss is 0. These are not realistic scenarios.

same variance, we therefore only need to know the correlation between the trials. We denote the correlation matrices $\boldsymbol{R}_t = \Sigma_t / v_t$, where v_1 and v_{-1} are the variances for the target and non-target trial losses respectively. These correlation matrices have one entry per observation, In order to estimate them, it is therefore necessary to impose a structure on them so that they depend on a small number of parameters. In Section 5.2, we propose how to do this for the correlations that arise from using the same speakers and utterances in several training trials.

It should be noticed that same results can be obtained by regarding the trial losses $\boldsymbol{l}_t(\boldsymbol{\theta}) = [l_1(\boldsymbol{\theta}), \ldots, l_{N_t}(\boldsymbol{\theta})]^T$ as one multivariate observation following normal distribution with mean $\boldsymbol{\eta} = [\eta, \ldots, \eta]^T$ and covariance matrix Σ_t and then using the ML estimate of η as loss estimator, i.e.,

$$\begin{aligned}\hat{L}_t(\boldsymbol{\theta}) &= \arg\max_{\eta} \frac{1}{\sqrt{(2\pi)^{N_t}|\Sigma_t|}} \exp\left(-\frac{1}{2}(\boldsymbol{l}(\boldsymbol{\theta}) - \boldsymbol{\eta})^T \Sigma_t^{-1} (\boldsymbol{l}(\boldsymbol{\theta}) - \boldsymbol{\eta})\right) \\ &= \boldsymbol{l}_t(\boldsymbol{\theta})^T \frac{\Sigma_t^{-1}\mathbf{1}}{\mathbf{1}^T\Sigma_t^{-1}\mathbf{1}}. \end{aligned} \tag{5.8}$$

In fact, any elliptical density with covariance matrix Σ_t gives the above result. Among them only the normal distribution has the property that a diagonal covariance matrix is equivalent to the losses being statistically independent but the ML estimate ignores higher order dependencies than covariance.

5.2 Estimation of R_t

In Subsection 5.1, we showed that using an equal weight for all target trials and another equal weight for the non-target trials is typically not optimal when the trials are statistically dependent. We argued that it is preferable to adjust the weights of the trials to obtain the BLUE for the loss estimator, $\hat{L}(\boldsymbol{\theta})$, and showed that in order to do this, we need to know the correlation between the losses of the training trials. In this section, we propose practical methods for *weight-adjustment* of training trials that are statistically dependent due to each speaker and utterance being used in more than one trial.

Table 5.1: Different kinds of trial pairs and the notation of their correlation. Capital letters refer to speakers and their indices refer to utterances. 'Corr' is the notation of the correlation coefficient.

Set	Things in common	Trial pair example	Corr.
Target	1 utt.	(A_1, A_2) - (A_1, A_3)	c_a
	Spk.	(A_1, A_2) - (A_3, A_4)	c_b
	Nothing	(A_1, A_2) - (B_1, B_2)	0
Non-target	1 utt., 1 spk.	(A_1, B_1) - (A_1, B_2)	c_{-a}
	2 spk.	(A_1, B_1) - (A_2, B_2)	c_{-b}
	1 utt.	(A_1, B_1) - (A_1, C_1)	c_{-c}
	1 spk.	(A_1, B_1) - (A_2, C_1)	c_{-d}
	Nothing	(A_1, B_1) - (C_1, D_1)	0

5.2.1 Weight-adjustment formulas

Let capital letters denote different speakers in our training data. Let N_X be the number of utterances of speaker X, X_i be the i-th utterance of this speaker, and $l(X_i, Y_j, \boldsymbol{\theta})$ be the loss for the trial involving utterance X_i and Y_j. We would like to make some assumptions about Σ_1 and Σ_{-1} that allow us to calculate the optimal weights $\boldsymbol{\beta}_t$ by means of Eq. (5.7). Since the optimal weight vector only depends on the correlation between the trial losses, we can let the variances of the trial losses be functions of $\boldsymbol{\theta}$. However, if the correlation coefficients depend on $\boldsymbol{\theta}$, the optimal weights will also depend on $\boldsymbol{\theta}$. For simplicity, we therefore assume that the correlation coefficients do not depend on $\boldsymbol{\theta}$, but only on what the trials have in common. For example, we assume:

$$\text{corr}\Big(l(A_1, A_2, \boldsymbol{\theta}), l(A_1, A_3, \boldsymbol{\theta})\Big) = c_a, \tag{5.9}$$

where the correlation coefficient, c_a, is the same for all target trial pairs that share one utterance. All the possible relations between two trials as well as the notation for the correlation coefficients are given in Table 5.1. Notice that two trials that have nothing in common are statistically independent so that, e.g., $\text{corr}\left(l(A_1, A_2, \boldsymbol{\theta}), l(B_1, B_2, \boldsymbol{\theta})\right) = 0$.

In this study, we use all the trials that can be constructed from the training data except those where both the utterances are the same. Under the assumptions given in the previous paragraph, the optimal weight for each

target trial of speaker A is then given by (see A.3.1 for proof)

$$\beta_A = \frac{k_1}{1 + 2(N_A - 2)c_a + (N_A - 2)(N_A - 3)c_b/2}, \tag{5.10}$$

where k_1 is set so that the sum of the weights equals 1. Since we do not use target trials where both the utterances are the same, a speaker with only one utterance is never used for target trials, i.e., $N_A = 1$ is never used in the above formula. For a speaker with two utterances, there is only one unique target trial so the second and third term in the denominator will be 0. For a speaker with three utterances, we can construct two trials with one shared utterance but not two trials with no shared utterances. In this case, the third term in the denominator will be 0. Notice that if all correlations equals 0, or if each speaker has the same number of utterances, each trial will obtain the same weight. If all correlations equal 1, each speaker will obtain the same weight.

In order to derive the weights for the non-target trials, we do some approximations. The *approximately* optimal weight for each non-target trial of speaker A and B, is then

$$\beta_{AB} \approx \frac{k_{-1}}{W_{AB}} \tag{5.11}$$

where k_{-1} is set so that the sum of the weights equals 1 and,

$$\begin{aligned} W_{AB} = 1 &+ c_{-a}(N_A + N_B - 2) \\ &+ c_{-b}(N_A - 1)(N_B - 1) \\ &+ \big(2c_{-c} + c_{-d}(N_A + N_B - 2)\big) \sum_{I \neq A,B} N_I. \end{aligned} \tag{5.12}$$

The derivation including the approximations is given in A.3.2.

5.2.2 Estimation of correlation coefficients

Ideally, we would have knowledge about c_a, c_b, c_{-a}, c_{-b}, c_{-c} and c_{-d}. In this study we explore two ways to find their values. The first is to approximate them with functions that depend on one tunable parameter. The second is to estimate them based on sample correlations in the training data.

Estimation by a one-parameter model

Consider first the target trials. We assume that two target trials from the same speaker are correlated, and that two target trials where one utterance

is the same are more correlated, so that $0 \leq c_b \leq c_a \leq 1$. In order to obtain only one parameter to tune we set

$$\begin{aligned} c_a &= \alpha_1, \\ c_b &= \alpha_1^2, \end{aligned} \tag{5.13}$$

where $0 \leq \alpha_1 \leq 1$ will be tuned. For a numeric example of this formula, let us compare the target trial weights for a speaker, A, with 2 utterances and a speaker, B, with 10 utterances. Speaker A has 1 trial and speaker B has 45 trials. If $\alpha = 0.5$, the *total* weight for the trials of speaker A is k_1 and the total weight for the trials of speaker B is $k_1 45/23$. In other words, speaker A has 5 times as many utterances as speaker B, but will obtain approximately 2 times more weight. For further illustration, the trial weights for different values of α_1, are given in Fig. 5.1. Notice that, even for small values of α_1, the number of utterances of a speaker has large impact on the optimal weights for that speaker.

For the non-target trials, we can apply the same strategy, i.e., assume that a shared speaker makes the trials correlated and a shared utterance makes them more correlated. However, the relation between c_{-b} and c_{-c} is not clear. The former gives the correlation between non-target trial losses where both speakers are the same. The latter gives the correlation between non-target trial losses that has one common utterance. While it is clear that having one common utterance should give larger correlation than only having one common speaker, we cannot guess the relation between *one same utterance* and *two same speakers*. Therefore, we set

$$c_{-a} = \alpha_{-1},\ c_{-b} = \alpha_{-1}^2,\ c_{-c} = \alpha_{-1}^2,\ c_{-d} = \alpha_{-1}^3, \tag{5.14}$$

with $0 \leq \alpha_{-1} \leq 1$. In this study, we will make a further simplification and set, $\alpha_1 = \alpha_{-1}$, denoted α from here on. This parameter will be tuned on a development set which, similarly to DT, can compensate for inaccuracies introduced by our assumptions and approximations. In other words, the model is tuned for good performance, instead of for fitting the data.

Estimation by sample correlation

Let $\bar{l}_1(\boldsymbol{\theta})$ and $\bar{v}_1(\boldsymbol{\theta})$ be the sample mean and sample variance of the loss of the target trials for the parameter $\boldsymbol{\theta}$. Given N_a target trial pairs with one

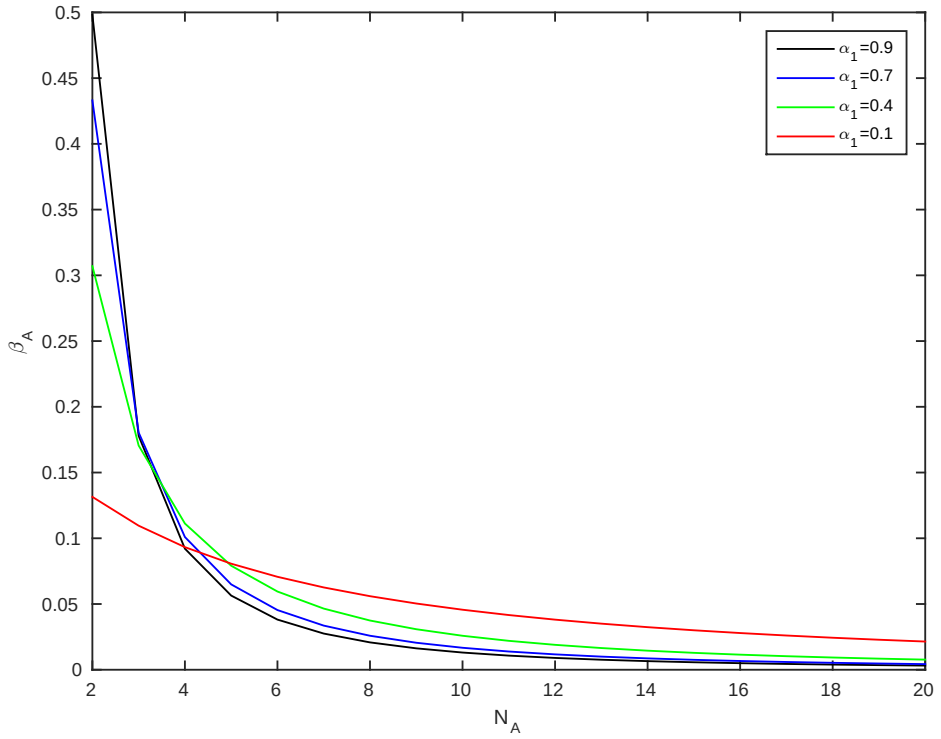

Figure 5.1: Optimal target trial weights for speaker A. 'N_A' is the number of utterances for the speaker and 'β_A' is the weight according to Eqs. (5.10) and (5.13). The normalization, k_1, is calculated assuming equally many trials of each N. Another distribution would change the relative position of the lines. Further, the scale of the y-axis depends on the total number of trials.

utterance in common (see Table 5.1), we calculate the sample correlation for those trials as

$$\bar{c}_a = \frac{1}{\bar{v}_1(\boldsymbol{\theta}) N_a} \sum_{h=1}^{N_a} \Big(l\big(\boldsymbol{\omega}_1^{(h)}, \boldsymbol{\omega}_2^{(h)}, \boldsymbol{\theta}\big) - \bar{l}_1(\boldsymbol{\theta}) \Big) \times \Big(l\big(\boldsymbol{\omega}_1^{(h)}, \boldsymbol{\omega}_3^{(h)}, \boldsymbol{\theta}\big) - \bar{l}_1(\boldsymbol{\theta}) \Big). \tag{5.15}$$

The sample correlations for the other correlation coefficients are calculated analogously. Compared to the one-parameter model, this method makes fewer assumptions about the data. It relies on the assumption that the correlation coefficients are independent of $\boldsymbol{\theta}$ and the same for each trial of the same kind (as defined in Table 5.1). On the other hand, since the correlation coefficients are not tuned for performance, but to fit the data, this method

may be more sensitive for incorrect model assumptions. For this method, we estimate the sample correlations based on trial losses calculated with the corresponding DT model without weight-adjustment.

5.3 Experiments

We followed the experimental set-up given in Section 4.1.1 and 4.2.1. The weight-adjustment parameter, α, was optimized over the steps $0, 0.1, \ldots, 1.0$. Sample correlations were estimated based on the losses of 10^6 trial pairs of each kind (sampled with replacement).

5.3.1 Results

Weight-adjustment for AT-Cal

For the initial exploration, we first evaluated the weight-adjustment with the one-parameter model for AT-Cal. We trained a PLDA model with the training data described in Subsection 5.3, and used data from the test set of SRE06 for calibration. We selected the calibration data in a way that the effect of the weight-adjustment should be easily observed, i.e., few speakers with large variation in their number of utterances. Specifically, we randomly selected 11, 14 or 21 speakers, and then for each of them, we randomly selected their number of utterances uniformly in the interval 1 to the number of available utterances (between 1 and 36, around 6 on average). Notice that this choice of calibration data was for demonstrating the effect of weight-adjustment. It is generally better to use all the available data. The actDCFs and $\hat{C}_{\text{llr}}$ for the α that was the optimal on the development set, as well as $\alpha = 0$ which gives the standard equal weight to each trial, are shown in Table 5.2. For reference, the result in the calibration-insensitive evaluation metrics are given in Table 4.1. In Figure 5.2, $\hat{C}_{\text{llr}}$ vs. α is shown. We observed a large improvement in $\hat{C}_{\text{llr}}$ for 11 calibration speakers. The optimal α on the development set was 0.5 but any value in between 0.1 and 0.6 gave similar results. For 14 and 21 speakers, the improvements were marginal. A general rule for the optimal value of α is therefore not possible to infer from this experiment.

The differences in actDCF08 were insignificant in most cases. Both actDCF08 and actDCF10 consider a small value of P_{eff} (0.0917 and 0.0010 re-

Table 5.2: Calibration results using SRE06 as calibration data. $\alpha = 0$ is the standard approach with equal weight to each trial. A '*' indicates that this value was optimal for $\hat{C}_{\text{llr}}$ on the development set.

Set	#Spkr	actDCF08	actDCF10	$\hat{C}_{\text{llr}}$	α
SRE08	11	0.0270	0.01525	3.660	0
		0.0271	0.01500	2.440	*0.5
	14	0.0343	0.00688	0.256	0
		0.0334	0.00552	0.245	*0.1
	21	0.0316	0.00266	0.215	0
		0.0291	0.00281	0.211	*1.0
SRE10	11	0.0113	0.001556	2.124	0
		0.0110	0.001518	1.363	*0.5
	14	0.0104	0.000468	0.087	0
		0.0104	0.000413	0.085	*0.1
	21	0.0103	0.000462	0.087	0
		0.0102	0.000452	0.082	*1.0

spectively). $\hat{C}_{\text{llr}}$ on the other hand, considers all values of P_{eff} (Brümmer and du Preez, 2006). For the training sets with 11, 14 and 21 speakers, the proportion of target-trials were 0.0774, 0.0736 and 0.0583 respectively. These values are close to P_{eff} of actDCF08. The expected loss is therefore likely to be better estimated for this value of P_{eff} than others, so that the benefit of an improved estimation procedure becomes smaller.

In the above experiment we used additional data for the calibration. Results using only the original training data for weight-adjustment both based on the one-parameter model and based on sample correlations are shown in Table 5.3. We considered the following training conditions:

1. Use the data from 90% of the training speakers for model training, and the data from the remaining training speakers for calibration.

2. Use all training data both for model training and for calibration.

3. As Condition 2, but in addition preserve the balance between the NIST SRE and the Switchboard corpora when weight-adjustment is utilized.

Condition 3 was motivated by the fact that the data is made-up by sev-

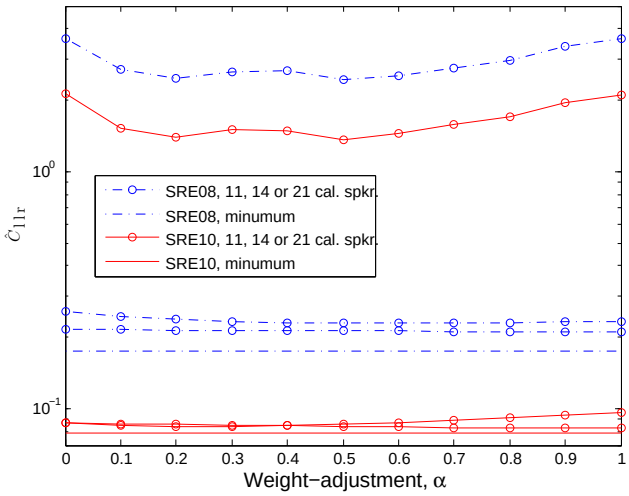

Figure 5.2: $\hat{C}_{\mathrm{llr}}$ vs. the weight-adjustment parameter α. The lines without circles denote $\hat{C}_{\mathrm{llr}}^{\mathrm{min}}$. Lines with circles denote $\hat{C}_{\mathrm{llr}}$ for 11 (upper), 14 (middle) and 21 (lower) calibration speakers, respectively.

eral different corpora and as a side-effect of weight-adjustment, the balance between these corpora may change. In fact, the Switchboard corpora has much fewer utterances per speaker than the NIST SRE corpora. Since the weight-adjustment increases the weights for speakers with fewer trials, the weight for Switchboard is increased. Again, we confirmed that the weight-adjustment is effective, although the effect was smaller than in our previous experiment. We did not see any significant difference between weight-adjustment based on the one-parameter model and weight-adjustment based on sample correlations. It was overall better to use all data both in model training and in calibration, than to split the data. Using calibration trials from i-vectors that have been used in PLDA training is not ideal since it does not resemble the test situation where the trials are from new i-vectors, while in our experiments, the benefit of having more data for training outweighed this problem. Preserving the balance between NIST SRE and Switchboard was useful, in which case we obtained an relative improvement in $\hat{C}_{\mathrm{llr}}$ of 5% by weight-adjustment on SRE10.

Weight-adjustment for Scr-UC

We have already confirmed that the weight-adjustment improves the performance of At-Cal. In this experiment we explore the effect of weight-

Table 5.3: Calibration results for three training/calibration conditions. The conditions are described in Subsubsection 5.3.1. 'W.-adj' refers to weight-adjustment, 'sp.' to sample correlation, and 'α' refers to the one-parameter model, tuned for $\hat{C}_{\text{llr}}$ on the development set.

Set	Cond.	actDCF08	actDCF10	$\hat{C}_{\text{llr}}$	W.-adj.
SRE08	1	0.0267	0.00151	0.286	no
		0.0278	0.00146	0.268	$\alpha = 0.2$
	2,3	0.0256	0.00130	0.201	no
	2	0.0251	0.00130	0.199	sp.
	2	0.0253	0.00131	0.197	$\alpha = 1.0$
	3	0.0251	0.00130	0.196	$\alpha = 1.0$
SRE10	1	0.0178	0.000623	0.168	no
		0.0182	0.000658	0.158	$\alpha = 0.2$
	2,3	0.0143	0.000678	0.100	no
	2	0.0141	0.000678	0.098	sp.
	2	0.0141	0.000688	0.098	$\alpha = 1.0$
	3	0.0135	0.000678	0.095	$\alpha = 1.0$

adjustment on Scr-UC. The former is important because this DT scheme performed the best on SRE10. Since Scr-UC is have more parameters to be estimated and therefore is more sensitive to over-fitting, it could be expected that weight-adjustment is more effective for this method than for AT-Cal. As in the baseline experiments, we applied regularization towards $\mathbf{0}$. We used the same regularization as for Scr-UC without weight-adjustment (which was tuned on the development set). The results are given in Table 5.4 and 5.5 in the calibration sensitive and insensitive evaluation metrics, respectively. As expected, the effect of weight-adjustment was larger than for AT-Cal, in particular actDCF08 where we observed improvements of around 5% and 8% for SRE08 and SRE10, respectively. The difference between the weight-adjustment based on the one-parameter model and the weight-adjustment based one sample correlations were as for AT-Cal small. In general, the minimum costs are much less effected by weight-adjustment than the actual costs. It should be noticed that the training objective aims at reducing actual costs. The minimum costs are only indirectly affected since they cannot be higher than the actual costs.

Table 5.4: Results of weight-adjustment for Scr-UC in the calibration sensitive evaluation metrics. The weight-adjustment parameter, α was tuned to optimize $\hat{C}_{\text{llr}}$ on the development set. L2 regularization towards $\mathbf{0}$ was applied.

Set	actDCF08	actDCF10	$\hat{C}_{\text{llr}}$	Weight-adj
SRE08	0.0334	0.000876	0.235	no
	0.0317	0.000851	0.231	$\alpha = 0.2$
	0.0314	0.000851	0.231	Sample corr.
SRE10	0.0304	0.000916	0.180	no
	0.0278	0.000888	0.173	$\alpha = 0.2$
	0.0275	0.000888	0.173	Sample corr.

Table 5.5: Results of weight-adjustment for Scr-UC in the calibration insensitive evaluation metrics. The weight-adjustment parameter, α was tuned to optimize $\hat{C}_{\text{llr}}$ on the development set. L2 regularization towards $\mathbf{0}$ was applied.

Set	minDCF08	minDCF10	EER	$\hat{C}_{\text{llr}}^{\text{min}}$	Weight-adj.
SRE08	0.0304	0.000743	0.0564	0.212	no
	0.0302	0.000739	0.0570	0.213	$\alpha = 0.2$
	0.0300	0.000740	0.0572	0.213	Sample corr.
SRE10	0.0183	0.000598	0.0370	0.137	no
	0.0182	0.000612	0.0369	0.137	$\alpha = 0.2$
	0.0183	0.000616	0.0368	0.138	Sample corr.

Reduced training data size

In the final experiment, we evaluated AT-Cal and SCR-UC using half smaller numbers of training speakers, with and without weight-adjustment. For simplicity, we did not preserve the balance between the NIST SRE and the Switchboard corpora. The same training data was used both in the GT step and the DT step. Since previous experiments showed very small differences between weight-adjustment based on the one-parameter model based on sample correlations, we use only the former in this experiment. In Table 5.6 the results using half of the training speakers are shown. Scr-UC benefited mostly from weight-adjustment where actDCF08 improved around 10% for

Table 5.6: Results for weight-adjustment using half of the training speakers in the calibration-sensitive evaluation metrics. The weight-adjustment parameter, α was tuned to optimize $\hat{C}_{\text{llr}}$ on the development set. For Scr-UC, L2 regularization towards **0** was applied .

Set	Method	actDCF08	actDCF10	$\hat{C}_{\text{llr}}$	α
SRE08	AT-Cal	0.0281	0.00092	0.201	0
		0.0286	0.00115	0.202	0.2
	Scr-UC	0.0591	0.001000	0.349	0
		0.0533	0.000997	0.328	0.4
SRE10	AT-Cal	0.0127	0.000743	0.101	0
		0.0125	0.000705	0.099	0.2
	Scr-UC	0.0646	0.001000	0.327	0
		0.0593	0.001000	0.302	0.4

Table 5.7: Results for weight-adjustment using half of the training speakers in the calibration-insensitive evaluation metrics. Notice that for these evaluation metrics, AT-Cal has no effect, i.e., the results are the same as if only GT had been used. The weight-adjustment parameter, α was tuned to optimize $\hat{C}_{\text{llr}}$ on the development set. For Scr-UC, L2 regularization towards **0** was applied .

Set	Method	minDCF08	minDCF10	EER	$\hat{C}_{\text{llr}}^{\text{min}}$	α
SRE08	AT-Cal	0.0263	0.000793	0.0523	0.187	-
	Scr-UC	0.0340	0.000790	0.0724	0.248	0
		0.0329	0.000819	0.0722	0.247	0.4
SRE10	AT-Cal	0.0111	0.000396	0.0229	0.090	-
	Scr-UC	0.0239	0.000695	0.0479	0.176	0
		0.0238	0.000737	0.0479	0.176	0.4

both SRE08 and SRE10, and $\hat{C}_{\text{llr}}$ improved around 6% for both SRE08 and SRE10. For reference, the results in the calibration insensitive evaluation metrics are shown in Table 5.7.

Table 5.8: Estimated correlations for AT-Cal, and Scr-UC. 'c_0' and 'c_{-0}' are the estimated correlations for trials that have nothing in common, and accordingly should be 0.

Set	Corr. coeff.	AT-Cal	Scr-UC
Target	c_a	0.378	0.364
	c_b	0.040	0.108
	c_0	4.79×10^{-4}	2.15×10^{-4}
Non-target	c_{-a}	0.768	0.770
	c_{-b}	0.534	0.507
	c_{-c}	0.010	0.006
	c_{-d}	7.75×10^{-4}	9.31×10^{-4}
	c_{-0}	-3.15×10^{-4}	3.20×10^{-4}

5.3.2 Analysis

In this subsection, we analyze how accurate the assumptions leading to the weight-adjustment formulas in Eqs. (5.10) and (5.12) are.

The weight-adjustment did not always work well. For example, in condition 1 in Table 5.3. The fact that the weight-adjustment did not always work may indicate that the assumptions behind it are not accurate enough. Let us recall the assumptions. First, we assumed that the correlation between the losses of two trials does not depend on the model parameters, $\boldsymbol{\theta}$, but only on what the trials have in common, e.g., one utterance might be the same in both trials. For the one-parameter model, we further assumed that all correlations are given by a parameter α which was tuned on the development set. The estimated sample correlations are shown in Table 5.8. We can see that there is a clear correlation between trials involving the same utterance or speaker. Moreover, the results for the three models are quite similar, which suggests the dependence on $\boldsymbol{\theta}$ may not be large. However, our assumptions about how the correlations depends on α are not that accurate. In particular, it is noticeable that using the same two speakers in both trials causes much more correlation than using only one same utterance, i.e., that $c_{-b} \gg c_{-c}$.

Chapter 6

Constrained discriminative PLDA training

In this chapter we address the problem of over- or underfitting described in Section 3.3.2 as well as examine the issues with direct optimization of the PLDA LLRs score function. As discussed in Section 3.3.2, AT-Cal is a very constrained DT scheme that is likely to underfit the training data. On the other hand, the other baseline, Scr-UC, easily overfits to the training data as showed in the experiments in Chapter 4.2. In those experiments, L2 regularization was crucial for decent performance. This was also concluded in Burget et al. (2011); Cumani et al. (2011).

Finding the right regularization is, however, difficult. Letting the regularization parameter be equal for all model parameters, as is typical, could be far from optimal. On the other hand, tuning many different regularization parameters is complicated. Here, we therefore propose two training schemes where a small number of parameters estimated by DT are used to adjust the score function of a PLDA model estimated by GT. This approach is in the spirit of AT-Cal but the training schemes we propose are less constrained. Based on the discussion in Subsection 2.6.4, we also propose a DT scheme that preserves the properties of $\boldsymbol{P}$ and $\boldsymbol{Q}$. In total, we propose three new DT schemes with varying degree of flexibility. As mentioned in Section 3.1.2, Borgström and McCree (2013) also reduced the number of parameters to be optimized. Contrary to that study we derive exact solutions for the gradient calculations of our proposed methods. These calculations are efficient enough for using all possible combinations of the training utter-

ances as training data. The gradients for the training schemes we propose can be derived based on the gradients for Scr-UC given in Cumani et al. (2011). The proposed DT schemes are presented in Section 6.1, the details of gradient calculations as well as the initializations are given in A.4 and experiments that compares the proposed DT schemes with the two baselines are given in Section 6.2.

6.1 Constrained DT schemes

In this section we present the three proposed constrained DT schemes for PLDA. The first two estimates fewer parameters than Scr-UC. The third one estimates as many parameters to as Scr-UC but restricts the values that these parameters can take.

The constrained DT schemes do not change the form of the PLDA score function, i.e., the function given in Eqs (2.38) and (2.44). This may seem to be a limitation but in fact, it has been shown that the second order Taylor expansion[1] of any (analytic) score function that is symmetric with respect to i-vector swapping, has this form (Cumani et al., 2013). Scr-UC can therefore be seen as discriminatively trained approximation of the best possible score function. By the constrained DT schemes we aim to make reasonable limitations of the score functions in order to avoid overfitting.

6.1.1 Reducing the number of parameters to be estimated

Using 4 parameters

As an option with $\mathcal{O}(1)$ parameters, we propose to scale each part of the PLDA LLR score function:

$$\begin{aligned} s_{ij} = a_{\mathrm{P}}\boldsymbol{\omega}_i^T\boldsymbol{P}\boldsymbol{\omega}_j + a_{\mathrm{P}}\boldsymbol{\omega}_j^T\boldsymbol{P}\boldsymbol{\omega}_i + a_{\mathrm{Q}}\boldsymbol{\omega}_i^T\boldsymbol{Q}\boldsymbol{\omega}_i + a_{\mathrm{Q}}\boldsymbol{\omega}_j^T\boldsymbol{Q}\boldsymbol{\omega}_j \\ +a_{\mathrm{c}}(\boldsymbol{\omega}_i + \boldsymbol{\omega}_j)^T\boldsymbol{c} + a_{\mathrm{k}}k, \end{aligned} \tag{6.1}$$

where a_{P}, a_{Q}, a_{c} and a_{k} are trained discriminatively, and $\boldsymbol{P}$, $\boldsymbol{Q}$ $\boldsymbol{c}$ and k are obtained by GT. In other words, we let the discriminative training adjust the weight of each *feature kind* in the original model parameters. The definiteness properties $\boldsymbol{P}$ and $\boldsymbol{Q}$ given in Subsection 2.6.4 were, in our experiments,

[1] The Taylor expansion needs to be done around a symmetric point such as $(\mathbf{0}, \mathbf{0})$ but this is reasonable since the mean both the i-vectors are $\mathbf{0}$.

almost always satisfied by itself (see Subsection 5.3.2), so we did not add any other constraints for this purpose. We refer to this method as Scr-4par. It should be noted that if $a_{\mathrm{P}} = a_{\mathrm{Q}} = a_{\mathrm{c}}$ in Eq. (6.1), we obtain AT-Cal.

Using $d + 1$ parameters

As an option with $\mathcal{O}(d)$ parameters, we propose to scale all the elements of the i-vector. Either one scaling for each of $\boldsymbol{P}$, $\boldsymbol{Q}$ and $\boldsymbol{c}$, or a common scaling can be used. In either case, we use a scaling of k. Accordingly this gives $3d + 1$ or $d + 1$ parameters to be estimated. In this study, we use the latter and refer to it as iV-elmnt. Another natural option with $\mathcal{O}(d)$ parameters is to scale the eigenvalues of $\boldsymbol{P}$ and $\boldsymbol{Q}$, but the advantage of iV-elmnt is that we do not have to consider whether to preserve the PLDA properties.

For generative ML training, letting $\mathrm{rank}(\boldsymbol{V}) = r < d$ has been reported to be beneficial (Matejka et al., 2011). As explained in Section 2.6.4, this reduces the rank of $\boldsymbol{P}$ and of $\boldsymbol{Q}$ from d to r as well. Based on results in (Bishop, 2006, p. 577) it follows that the number of parameters to be estimated will be reduced from $d^2 + 2d + 1 = \mathcal{O}(d^2)$ to $r(2d - r) + d + r + 1 = \mathcal{O}(dr)$. However, a large reduction of the number of parameters in this way limits the model too much. As an extreme example, if we want the same number of parameters as iV-elmnt, we have to set $r = 1$, which means that the i-vectors are projected into a one-dimensional space.

6.1.2 Preserve the properties of $\boldsymbol{P}$ and $\boldsymbol{Q}$

Scr-UC optimizes the parameters of LLR score, $[\mathrm{vec}(\boldsymbol{P})^T, \mathrm{vec}(\boldsymbol{Q})^T, \boldsymbol{c}^T, k]^T = \boldsymbol{\gamma}$, directly, whereas the DT scheme proposed in Borgström and McCree (2013) optimizes the parameters of the PLDA model, $\boldsymbol{m}$, $\boldsymbol{V}$ and $\boldsymbol{D}$. The discriminative training objective in Eq. (3.2) depends on the scores, s_h, of the training trials. Since the scores according to Eq (2.44) are given by a linear function of $\boldsymbol{\gamma}$, direct optimization of $\boldsymbol{\gamma}$ is most straight-forward. However, if no constraints are imposed on $\boldsymbol{\gamma}$, this may result in a model with different properties than a PLDA model.

As the least constrained option, we propose to just preserve the definiteness constraints of $\boldsymbol{P}$ and $\boldsymbol{Q}$. In this case the number of parameters to be estimated is not reduced, but the values they can take are limited. In this subsection, we propose reparameterizations of $\boldsymbol{P}$ and $\boldsymbol{Q}$ such that

when these parameters are optimized instead of $\boldsymbol{P}$ and $\boldsymbol{Q}$, the definiteness constraints will be preserved.

The matrix $\boldsymbol{P}$ is positive-semidefinite if

$$\boldsymbol{P} = \boldsymbol{P}_A \boldsymbol{P}_A^T, \tag{6.2}$$

where $\boldsymbol{P}_A$ is a $d \times r$ matrix with real elements. Accordingly, in order to keep $\boldsymbol{P}$ positive-semidefinite, we train $\boldsymbol{P}_A$ instead of $\boldsymbol{P}$. The rank of $\boldsymbol{P}$ is equal to r and can therefore be selected by selecting the number of columns in $\boldsymbol{P}_A$. If we wish to control the rank without keeping $\boldsymbol{P}$ positive definite, we can use $\boldsymbol{P} = \boldsymbol{P}_A \boldsymbol{P}_B$ where $\boldsymbol{P}_B \neq \boldsymbol{P}_A^T$.

In order to keep $\boldsymbol{Q}$ negative-semidefinite, we set

$$\boldsymbol{Q} = -\boldsymbol{Q}_A \boldsymbol{Q}_A^T, \tag{6.3}$$

and train $\boldsymbol{Q}_A$ instead of $\boldsymbol{Q}$.

In order to enforce the third constraint, we use $\boldsymbol{Q} = -\boldsymbol{Q}_A \boldsymbol{Q}_A^T$ but set

$$\boldsymbol{P} + \boldsymbol{Q} = \boldsymbol{R}_A \boldsymbol{R}_A^T, \tag{6.4}$$

instead of $\boldsymbol{P} = \boldsymbol{P}_A \boldsymbol{P}_A^T$. We then optimize $\boldsymbol{R}_A$ and $\boldsymbol{Q}_A$. In this study, we apply the three definiteness constraints of $\boldsymbol{P}$ and $\boldsymbol{Q}$ in this way without reducing their rank. We refer to this method as Scr-Def.

Regularization

As for Scr-UC, we apply L2 regularization to Scr-Def. That is, we add the term $\rho\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|^2$ to the training objective in Eq. (7.1), where $\|\cdot\|$ denotes the Euclidean norm and the regularization parameter, ρ, is tuned on a development set. This forces the parameter vector, $\boldsymbol{\theta}$, to be close to $\tilde{\boldsymbol{\theta}}$. For Scr-UC and Scr-Def, we use either $\mathbf{0}$ or the model from GT. For Scr-Def, we use regularization in terms of $\boldsymbol{P}$ and $\boldsymbol{Q}$ rather than $\boldsymbol{R}_A$ and $\boldsymbol{Q}_A$. For example, the contribution to the regularization term from $\boldsymbol{Q}$ is

$$\rho\|\mathrm{vec}(\boldsymbol{Q} - \tilde{\boldsymbol{Q}})\|^2 = \rho\|\mathrm{vec}(-\boldsymbol{Q}_A \boldsymbol{Q}_A^T - \tilde{\boldsymbol{Q}})\|^2, \tag{6.5}$$

rather than $\rho\|\mathrm{vec}(\boldsymbol{Q}_A - \tilde{\boldsymbol{Q}}_A)\|^2$, where $\tilde{\boldsymbol{Q}}$ and $\tilde{\boldsymbol{Q}}_A$ denotes either $\mathbf{0}$ or the parameters obtained by GT.

The optimal ρ depends on the size of the training data. Instead of tuning ρ for each training data size, we use a modified training objective given by

$$\hat{L}'(\boldsymbol{\theta}) = \kappa \hat{L}(\boldsymbol{\theta}) + \rho\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|^2 \tag{6.6}$$

where $\kappa = N_1 + N_{-1}$. We then tune ρ for the full training data and use this value also for smaller amounts of training data. This means that the influence of the regularization becomes larger for the smaller training data.

6.2 Experiments

In this section we evaluate the different DT schemes. Here we do not utilize the weight-adjustment proposed in Chapter 5. See Chapter 8 for experiments that combines the constrained DT schemes with weight-adjustment. We followed the experimental set-up given in Section 4.1.1 and 4.2.1.

Comparison of DT schemes

We evaluated the different discriminative PLDA training schemes without weight-adjustment. Since in the previous experiment, using all training data both for the GT step and the DT step was better in almost all cases, we continued to use this approach. The results are shown in Table 6.1 and 6.2 for the calibration sensitive and insensitive evaluation metrics, respectively. If we ignore the methods for which regularization towards the GT model was applied, there is a clear pattern in the results. Overall, one of the baselines, AT-Cal, performed best for SRE08 and Scr-4par performed best for SRE10, where the relative improvement over AT-Cal was 14%. The less constrained iV-elmnt performed worse than these two methods but better than Scr-UC which has no constraints other than regularization. One possible reason that the most constrained DT scheme, AT-Cal, was the best for SRE08 might be that this database has a larger mismatch with the training data than SRE10. In such case DT might be more risky unless it is very constrained. Compared to SRE10, SRE08 contains non-english speech and possibly also speech recorded with a microphone. The training data contains only a small ratio of such speech.

For Scr-Def and Scr-UC, we had to apply regularization in order to avoid overfitting. The regularization parameter, ρ was chosen to minimize $\hat{C}_{\text{llr}}$ on the development set, SRE06. In terms of $\hat{C}_{\text{llr}}$, these two methods performed worse than AT-Cal and Scr-4par. Applying regularization towards the GT model, gives mixed results. Both SCR-Def and SCR-UC performed well in actDCF08 but they did not perform well in actDCF10 and $\hat{C}_{\text{llr}}$. This indi-

Table 6.1: Results for different DT schemes in the calibration sensitive evaluation metrics. AT-Cal and Scr-UC R. **0** are baselines.'R. GT' and 'R. **0**' mean L2 regularization towards the model obtained by GT and regularization towards **0**, respectively. The regularization parameter, ρ, was tuned to optimize for $\hat{C}_{\text{llr}}$ on the development set.

Set	Method	actDCF08	actDCF10	$\hat{C}_{\text{llr}}$	ρ
SRE08	AT-Cal	0.0256	0.00130	0.201	-
	Scr-4par	0.0274	0.00257	0.202	-
	iV-elmnt	0.0269	0.00171	0.225	-
	Scr-Def. R. GT	0.0269	0.01278	1.415	10^2
	Scr-UC. R. GT	0.0268	0.01269	1.416	10^2
	Scr-UC. R. **0**	0.0334	0.000876	0.235	10^1
SRE10	AT-Cal	0.0143	0.000678	0.100	-
	Scr-4par	0.0117	0.000574	0.086	-
	iV-elmnt	0.0146	0.000563	0.119	-
	Scr-Def. R. GT	0.0110	0.001523	0.663	10^2
	Scr-UC. R. GT	0.0111	0.001495	0.664	10^2
	Scr-UC. R. **0**	0.0304	0.000916	0.180	10^1

cates that these systems are only good for the effective prior that has been specified in the training objective. The bad performance for the other effective priors is, however, surprising since the logistic regression loss function emphasizes on a broad range of effective priors (Brümmer and du Preez, 2006). Moreover, the optimal ρ was quite large and the effect on minDCF08 was minor. It should also be noted that the optimal ρ varies depending on which evaluation metric is considered. In particular, this was a problem for Scr-Def with regularization towards **0** so we did not include that result in the table. The problem of this method method might be because its objective function is non-convex.

All the results taken into account, AT-cal and Scr-4par seems to be the best methods for this amount of training data or smaller.

Reduced training data size

In the final experiment, we evaluated Scr-4par and the two baselines AT-Cal and SCR-UC using half the number of training speakers. The same training

Table 6.2: Results for different DT schemes in the calibration insensitive evaluation metrics. AT-Cal and Scr-UC R. **0** are baselines.'R. GT' and 'R. **0**' mean L2 regularization towards the model obtained by GT and regularization towards **0**, respectively. The regularization parameter, ρ, was tuned to optimize for $\hat{C}_{\text{llr}}$ on the development set.

Set	Method	minDCF08	minDCF10	EER	$\hat{C}_{\text{llr}}^{\text{min}}$	ρ
SRE08	AT-Cal	0.0250	0.000728	0.0480	0.175	-
	Scr-4par	0.0254	0.000802	0.0471	0.177	-
	iV-elmnt	0.0262	0.000669	0.0478	0.182	-
	Scr-Def. R. GT	0.0253	0.000809	0.0461	0.178	10^2
	Scr-UC. R. GT	0.0253	0.000809	0.0459	0.178	10^2
	Scr-UC. R. **0**	0.0304	0.000743	0.0564	0.212	10^1
SRE10	AT-Cal	0.0101	0.000385	0.0198	0.0788	-
	Scr-4par	0.0100	0.000375	0.0188	0.0744	-
	iV-elmnt	0.0121	0.000412	0.0253	0.0962	-
	Scr-Def. R. GT	0.0103	0.000377	0.0204	0.0805	10^2
	Scr-UC. R. GT	0.0103	0.000380	0.0203	0.0801	10^2
	Scr-UC. R. **0**	0.0183	0.000598	0.0370	0.1368	10^1

Table 6.3: Results for three DT schemes in the calibration insensitive evaluation metrics using half of the training speakers.. For Scr-UC, L2 regularization towards **0** was applied.

Set	Method	actDCF08	actDCF10	$\hat{C}_{\text{llr}}$
SRE08	AT-Cal	0.0281	0.00092	0.201
	Scr-4par	0.0322	0.00177	0.220
	Scr-UC	0.0591	0.001000	0.349
SRE10	AT-Cal	0.0127	0.000743	0.101
	Scr-4par	0.0116	0.000657	0.095
	Scr-UC	0.0646	0.001000	0.327

data was used both in the GT step and the DT step. The results are shown in Table 6.3 and 6.4 for the calibration sensitive and insensitive evaluation metrics respectively. Looking at the $\hat{C}_{\text{llr}}$ column we can see that reducing the amount of training data deteriorated the performance more, the less constrained the models were as can be expected.

Table 6.4: Results for three DT schemes in the calibration insensitive evaluation metrics using half of the training speakers. For Scr-UC, L2 regularization towards $\mathbf{0}$ was applied.

Set	Method	minDCF08	minDCF10	EER	$\hat{C}_{\text{llr}}^{\text{min}}$
SRE08	AT-Cal	0.0263	0.000793	0.0523	0.187
	Scr-4par	0.0274	0.000921	0.0515	0.190
	Scr-UC	0.0340	0.000790	0.0724	0.248
SRE10	AT-Cal	0.0111	0.000396	0.0229	0.090
	Scr-4par	0.0106	0.000394	0.0217	0.085
	Scr-UC	0.0239	0.000695	0.0479	0.176

6.3 Analysis

In this section, we first perform an error analysis to figure out why did not see any improvement over AT-Cal on SRE08. Second, we analyze whether the definiteness properties of $\boldsymbol{P}$ and $\boldsymbol{Q}$ discussed in Subsection 2.6.4 are important.

6.3.1 Error analysis

Our proposed DT scheme, Scr-4par, outperformed AT-Cal on SRE10 but not on SRE08. In order to understand the reason for the poor performance on SRE08, we need to consider the differences between SRE08 and SRE10 more in detail. Each NIST evaluation focuses on a variety of issues in speaker verification. For example, transmission channels (telephone, microphone), varying utterance lengths, multiple enrollment utterances, very short utterances, vocal effort (whispering, normal speech, screaming) etc. All such factors can affect the merit of a method. In our experiments though, we used the *core task condition-6* for SRE08 and the *(extended) core task condition-5* for SRE10. These two conditions excludes most of the above mentioned factors by involving only telephone speech in both the enrollment and the authentication data, having fairly matched utterance lengths of the enrollment and authentication utterances, only one enrollment utterance per target speaker and so on. Moreover, they are quite well matched with the properties of our training data (SRE04, SRE05 and the Switchboard corpora, see Section 4.1.1). This set-up is suitable when studying funda-

Table 6.5: Error analysis for SRE08. The error rates are calculated using the decision threshold for DCF08. The *FA cost* and *FR cost* are calculated using the effective prior for DCF08, $P_{\text{eff}} = 0.0917$, i.e., the FA cost equals the FA rate times $1 - P_{\text{eff}}$ and the FR cost equal the FR rate times P_{eff}. Accordingly, the *actDCF* is the sum of the cost for FA and FR.

Method	Data	FA rate	FR rate	FA cost	FR cost	actDCF	$\hat{C}_{\text{llr}}$
AT-Cal	All	0.0116	0.164	0.0151	0.0105	0.0256	0.201
	Eng.	0.0028	0.132	0.0025	0.0121	0.0146	0.119
	Other	0.0216	0.195	0.0196	0.0179	0.0375	0.288
Scr-	All	0.0183	0.117	0.0166	0.0108	0.0274	0.202
4par	Eng.	0.0042	0.085	0.0038	0.0078	0.0116	0.111
	Other	0.0343	0.149	0.0311	0.0137	0.0448	0.302

mental techniques rather then one of specific challenges mentioned above. However, there is one important difference between the SRE08 core task condition 6 (SRE08) and the SRE10 core task condition-5 (SRE10). Namely that SRE08 contains a large portion of non-English data whereas SRE10 contains only English data. The training data contains very little speech from languages other than English. Since a less constrained DT scheme more easily overfits to the training data, this kind of mismatch between the training and evaluation data can therefore be a reason for the bad performance of Scr-4par compared to AT-Cal. In order to check this hypothesis we evaluate the performance of the methods for the English and non-English trials separately. The results in actDCF08 and $\hat{C}_{\text{llr}}$ are shown in Table 6.5. For further analysis, we also check the FA rate and FR rate for actDCF08. As can be seen, for the English trials Scr-4par outperforms AT-Cal but for the non-English trials AT-Cal is clearly better. This analysis suggests that when there is a mismatch between the training and the test data, one need to use more constrained DT schemes. It is interesting to notice that for both English and non-English trials, the FA rate is lower for AT-Cal and the FR rate is lower for Scr-UC. For the English trials, the lower FR rate of Scr-4par outweighs its higher FA rate. Naturally, for other values of P_{eff}, the decision threshold would have been different and hence the error rates. The fact that $\hat{C}_{\text{llr}}$ improves (as well as the improvements in actDCF10 for SRE10) tells us that Scr-4par outperforms AT-Cal for other values of P_{eff} too.

6.3.2 Definiteness properties of $\boldsymbol{P}$ and $\boldsymbol{Q}$

As already revealed, for Scr-4par, the definiteness constraints on $\boldsymbol{P}$ and $\boldsymbol{Q}$ were almost always preserved by itself. $\boldsymbol{P}$ was always kept positive definite and $\boldsymbol{Q}$ was always kept negative definite. Out of the 8 training sizes, it happened once that the matrix $\boldsymbol{P}+\boldsymbol{Q}$ was not positive definite, but in this case, only one of its eigenvalues were negative. We did not see any difference in performance between Scr-Def and Scr-UC but an inspection of the eigenvalues reveals that the definiteness constraints were never fulfilled for Scr-UC, regardless of whether regularization was applied towards the model obtained by GT, or towards $\mathbf{0}$. However, we also investigated whether $\boldsymbol{\omega}^T\boldsymbol{P}\boldsymbol{\omega} > 0$, $\boldsymbol{\omega}^T\boldsymbol{Q}\boldsymbol{\omega} < 0$ and $\boldsymbol{\omega}^T(\boldsymbol{P}+\boldsymbol{Q})\boldsymbol{\omega} > 0$ for each i-vector, $\boldsymbol{\omega}$, in the PLDA training set and in the development set, SRE06. When regularization towards the GT model was applied, the number of *violations* was very few. This means that the constraints are, in some sense, practically fulfilled for i-vectors that are normally observed. This may be because the DT model remains close to the GT model and therefore keeps its properties. Interestingly though, when regularization towards $\mathbf{0}$ was applied, there were many violations against the second constraint, $\boldsymbol{\omega}^T\boldsymbol{Q}\boldsymbol{\omega} < 0$, but no violations against the other two constraints. Recall from Section 2.6.4 that the second constraint is related to a length property of the model, which is unlikely to be useful when the i-vectors are length-normalized. While this supports our analysis in Section 2.6.4, it also shows that, at least for the training sizes used in our experiment, the training procedure tends to learn the useful properties from the data.

Chapter 7

Application-specific loss functions

As discussed in Sections 3.2 and 3.3.3, the choice of loss function in DT is important but far from trivial. In order to obtain calibrated LLR scores, we should use proper scoring rules as loss functions. Among them $\hat{C}_{\text{llr}}$, emphasizes on a broad range of OPs. As opposed the logistic regression loss, we refer to loss function that focus on a narrower range of OPs as *application-specific*. In many applications the evaluation metric of interest is actDCF of one specific OP (and if several OPs are of interest, we could usually train one system for each of them). However, it is not certain DT with a loss function that focus only on the specific OP, i.e., a 0-1 loss, will result in the best performance due to the difficulty of optimization. In this chapter we explore this issue. Specifically, we aim to minimize actDCF08 for Scr-UC. To this end we evaluate the Brier loss and the (approximate) 0-1 loss as well as the tuning of p_{eff}. The contribution of this chapter lies in the experimental analysis rather than in any technical novelty. In section 7.1, we discuss the theoretical considerations and in Section 7.2 we present our experiments. Experiments that combines the application-specific loss function with the methods proposed in Chapters 5 and 6 are presented in Chapter 8.

7.1 Motivation

Recall that our training objective is to minimize

$$\bar{l}(\boldsymbol{\theta}) = \sum_{h:t_h=1} \frac{P_{\text{eff}}}{N_1} l\left(t_h, s_h(\boldsymbol{\theta}), \tau\right) + \sum_{h:t_h=-1} \frac{1-P_{\text{eff}}}{N_{-1}} l\left(t_h, s_h(\boldsymbol{\theta}), \tau\right), \quad (7.1)$$

where $l\left(t_h, s_h, \tau\right)$ is the loss function. In other words, we minimize the average of $l\left(t_h, s_h, \tau\right)$ of the training trials balanced with P_{eff}. By doing this we aim to minimize the expected loss of $l\left(t_h, s_h, \tau\right)$ of a test trial when the probability of a target trial is P_{eff} (see the beginning of Chapter 5 for details regarding the expected loss of test trial). In this thesis, we have up until now $l\left(t_h, s_h, \tau\right)$ employed the logistic regression loss. In many applications the relevant evaluation metric is actDCF for some OP. It may therefore tempting to use actDCF as a loss function, i.e., the 0-1 loss function. However, this has several problems. The optimization is difficult because the 0-1 loss is a non-differentiable and non-convex function. These problems becomes worse when there are many parameters to optimize as in Scr-UC. Furthermore, the 0-1 loss easily overfits to the training data since it only cares about getting the scores on the right side of the threshold. For AT-Cal, this would be equivalent to tuning the threshold which we argued is sensitive to overfitting in Section 2.3.1 and again, this problem will get worse when there are more parameters to optimize. The advantage of the 0-1 loss is that it is robust to outliers since the cost is bounded.

A compromise between the logistic regression loss and the 0-1 loss is the Brier loss. This loss function focus on a narrower range of operating points than the logistic regression loss but a not as extremely narrow (i.e., one OP) as the 0-1 loss. For AT-Cal, Brümmer and Doddington (2013) obtained better results with the Brier loss when the evaluation metric was C_{primary} of the NIST SRE 2012 (an average of actDCF with $P_{\text{eff}} = 0.01$ and actDCF with $P_{\text{eff}} = 0.001$).

Aside for the choice of loss function, the choice of P_{eff} needs to be considered. We have previously argued that in order to minimize the expected loss of a test trial, P_{eff} must be set according to Eq (2.5). However, this is true only when the loss function used in the training objective is the same as we want to minimize in the testing phase. If our aim is to minimize actDCF (the 0-1 loss) of the test trials, bu we in order to avoid overfitting, use logistic regression or the Brier loss in the training objective, there is not

guarantee that theP_{eff} given by Eq (2.5) is the optimal one. Furthermore, there are other, more practical reasons why the optimal value of P_{eff} can be different. One reason is that in the NIST test data, all target trials are from different telephone numbers. This is not the case in the training data where most target trials are from the same telephone number. In order for the *different-number* target trials to obtain appropriate influence over the model, P_{eff} may have to be adjusted.[1]

7.1.1 Loss functions

Here we give an summary of the key properties of the three loss functions. The shapes of the loss functions are shown in Figure 7.1 and the corresponding weight for different LLR thresholds are shown in Figure 7.2.

Logistic regression loss

Uses the weight function $w(\zeta) = 1 \Leftrightarrow w(\tau') = 1/(2 + \exp(x) + \exp(-x))$.

$$l_{\text{LR}}(t, s; \tau) = \log(1 + \exp(-t(s - \tau)), \tag{7.2}$$

- Focuses on a wide range of OPs
- Convex
- Can be sensitive to outliers

Brier loss

Uses the weight function $w(\zeta) = 6\zeta(1 - \zeta) \Leftrightarrow w(\tau') = 6/(2 + \exp(x) + \exp(-x))^2$.

$$l_{\text{Brier}}(t, s; \tau) = \frac{3}{(1 + \exp t(s - \tau))^2}, \tag{7.3}$$

- Focuses on a narrow range of OPs
- Non-convex

[1] We thank one anonymous reviewer of our Odyssey paper (Rohdin et al., 2014b) four pointing this out.

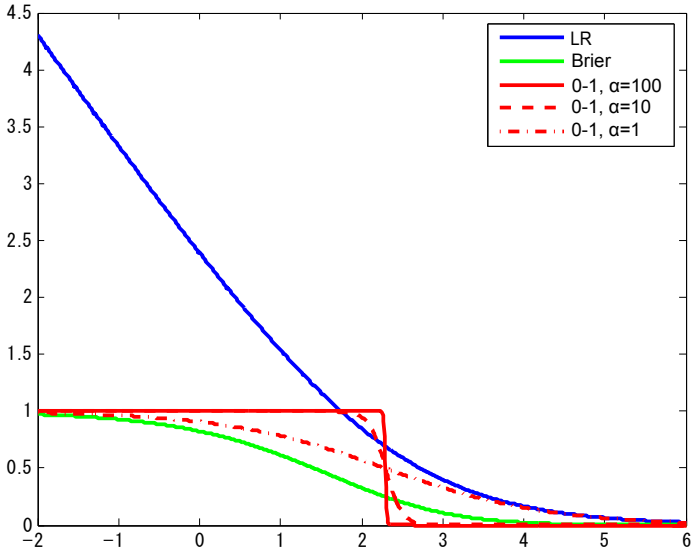

Figure 7.1: Comparison of loss functions. The Brier and the 0-1 loss functions have been normalized to have a maximum of 1.

0-1 loss

Uses the weight function $w(\zeta) = \delta(0) \Leftrightarrow w(\tau') = \delta(\tau' - \tau)$, where $\delta(\tau')$ is the Dirac impulse at τ'.

$$l_{0-1}(t, s; \tau) = \begin{cases} 1 & \text{if } t(s - \tau) < 0, \\ 0 & else. \end{cases} \tag{7.4}$$

- Focuses on one OP
- Non-convex
- Non-differentiable

To obtain a differentiable loss function we use a standard trick to approximate it with the sigmoid function Nguyen and Sanner (2013),

$$l_\sigma(t, s; \tau) = \frac{1}{1 + \exp(\alpha t(s - \tau))}. \tag{7.5}$$

This function is differentiable and can become arbitrary close to the 0-1 loss function if α is increased. We will refer to this as the approximate 0-1 loss. In order for it to be a good approximation to the 0-1 loss, we need to make α large enough. As can be seen in Figure 7.1, $\alpha = 100$ approximates the 0-1 well on the relevant scale.

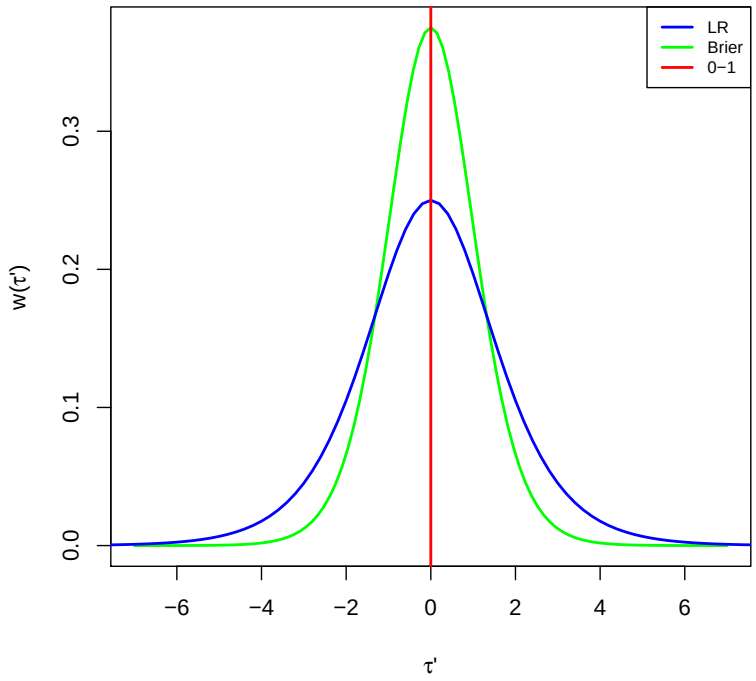

Figure 7.2: The LLR threshold weight for different loss functions functions. Here, we used $\tau = 0$.

7.1.2 Optimization procedure

Since the Brier and the approximate 0-1 loss are non-convex we may get stuck in a bad local minima during the optimization process. A simple approach to deal with the non-convexity of the approximate 0-1 loss was proposed in Nguyen and Sanner (2013). In this work they gradually increase the values of α during optimization. Although the loss function is non-convex for any choice of α, it was empirically shown that lower values of α results in fewer local minima. We will use this with $\alpha = [1, 10, 100]$. For the Brier loss we will use two steps, first the approximate 0-1 loss with $\alpha = 1$ and then the Brier loss. In both cases we will start from the LR model. It should be noted that the work in Nguyen and Sanner (2013) in addition to this strategy tried to escape local minima by systematically searching their neighborhood for lower points.

Table 7.1: Comparison of loss functions in the calibration sensitive evaluation metrics. L2 regularization towards $\mathbf{0}$ was applied. The regularization parameter, ρ, was tuned to optimize actDCF08 on the development set.

Set	Method	actDCF08	actDCF10	$\hat{C}_{\text{llr}}$	ρ
SRE08	Logistic regression	0.0334	0.000876	0.235	10
	Brier	0.0322	0.000938	0.285	10
	Approximate 0-1	0.0312	0.001000	1.626	100
SRE10	Logistic regression	0.0304	0.000916	0.180	10
	Brier	0.0287	0.000978	0.229	10
	Approximate 0-1	0.0453	0.001000	1.627	100

7.2 Experiments

We used the experimental set-up described in Chapter 4. But notice that aim of the experiments in this Chapter is to minimize one specific evacuation metric, namely actDCF08. Other evaluation metrics could therefore be expected to degrade. We conducted 2 experiments. In the Subsection 7.2.1, we compare the three loss functions for Scr-UC. In the Subsection 7.2.2, we evaluate the effect of varying P_{eff}. [2]

7.2.1 Comparison of loss functions

The results for the different loss functions in the calibration-insensitive evaluation metrics are given in in Table 7.1. The Approximate 0-1 loss performed well on SRE08 very bad on SRE10. The optimal regularization was also higher than for the other two loss function, which is reasonable since the 0-1 loss more easily overfits. The Brier loss performed better than the baseline on both SRE08 and SRE10 in actDCF08. In the other evaluation metrics, it performed worse but this is expected since our target was to optimize the system for actDCF08 only.

The results for the different loss functions in the calibration-insensitive

[2]Notice that the experimental set-up here is quite different from the one in Rohdin et al. (2014b). There we used rank($\boldsymbol{V}$) $= 250$ and a larger training data set. Further, we used regularization towards the ML models which our experiments in Chapter 6 showed is problematic. Also, when varying P_{eff}, τ was kept fixed at the original value. In the experiment in this thesis, we updated τ according to Eq (2.6).

Table 7.2: Comparison of loss functions in the calibration insensitive evaluation metrics. L2 regularization towards **0** was applied. The regularization parameter, ρ, was tuned to optimize actDCF08 on the development set.

Set	Method	minDCF08	minDCF10	EER	$\hat{C}_{\text{llr}}^{\text{min}}$	ρ
SRE08	Logistic regr.	0.0304	0.000743	0.0564	0.212	10
	Brier	0.0305	0.000743	0.0550	0.211	10
	Approx. 0-1	0.0304	0.000785	0.0667	0.231	100
SRE10	Logistic regr.	0.0183	0.000598	0.0370	0.1368	10
	Brier	0.0180	0.000594	0.0357	0.1325	10
	Approx. 0-1	0.0208	0.000630	0.0436	0.1593	100

evaluation metrics are given in in Table 7.2 As can be seen, the methods performed very similar in minDCF08. A possible explanation for this could be that model has not changed from much the logistic regression model that we used as starting point in the optimization. Although, it changed enough to affect the calibration sensitive evaluation metrics significantly. This issue deserves further analysis in the future.

7.2.2 Effect of the weight in the training objective

In order to see the effect of varying P_{eff}, we substituted P_{eff} with $P'_{\text{eff}} = \gamma P_{\text{eff}}/(\gamma P_{\text{eff}} + (1-\gamma)(1-P_{\text{eff}}))$. For $\gamma = 0.5$ this gives $P'_{\text{eff}} = P_{\text{eff}}$, for $\gamma = 1$, it gives $P'_{\text{eff}} = 1$ and for $\gamma = 0$, it gives $P'_{\text{eff}} = 0$. We then train an Scr-UC model with the Brier loss for γ between 0.1 and 0.9. The results shown in Figure 7.3.

The choice of β seemed to be important for actDCF but less important for minDCF.

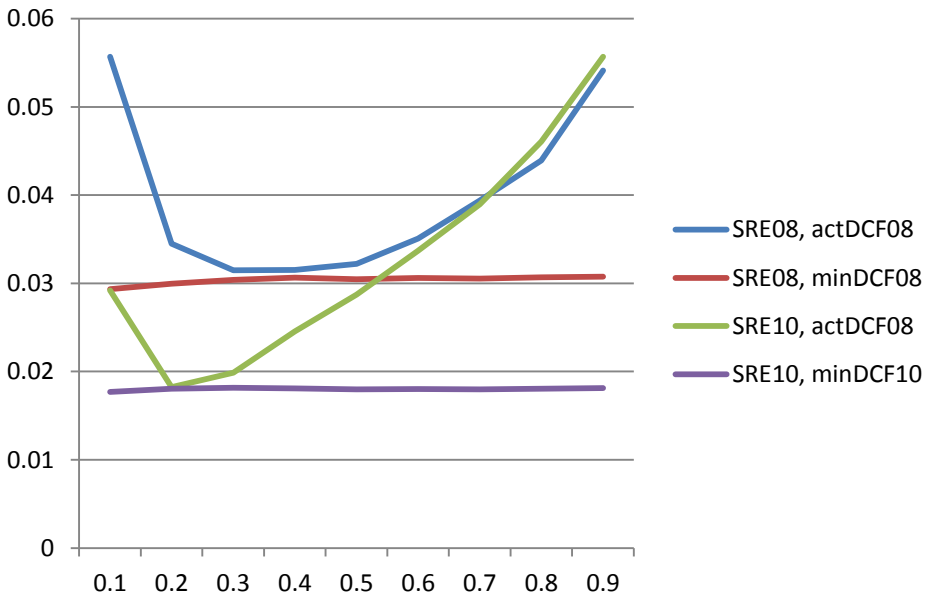

Figure 7.3: The effect of changing P_{eff}. The x-axis shows γ as defined in the text.

Chapter 8

Combining the proposed methods

In this chapter we analyze the combinations of the methods from Chapter 5, 6 and 7. It should be noted that while the methods of Chapters 5 and 6 aims at improving speaker verification in general, i.e., for all evaluation metrics, the application-specific loss functions of Chapter 7 aims at improving the performance for one or a few OPs on the expense of worsened performance for other OPs. This is not always desirable. The organization of this chapter is as follows. In Section 8.1 we hypothesize what can be expected when combining the methods, based on the discussions and experiments in the previous Chapters. In Section 8.2 we then present experiments with various combinations of the methods. We devote one subsection to each combination of two methods and one subsection to the combination of the three methods. Finally, we end this Chapter by a summary of the results as well as recommendations for how and when to use the different methods in Section 8.3.

8.1 Expectations

Based on the the discussions in Chapter 5, 6 and 7 we can expect the following:

- By reducing the variance of the training objective, weight adjustment has a similar effect as adding training data. Therefore it should be

more useful for less constrained DT schemes. In fact, this was confirmed in Chapter 5 were we showed that weight-adjustment was more effective for Scr-UC than the more constrained AT-Cal. Our best DT scheme was Scr-4par. This DT scheme is only slightly less constrained than AT-Cal, so the benefit of applying weight-adjustment to it can be expected to be similar to that of AT-Cal. We evaluate this in Section 8.2.1.

- Application specific loss functions are more sensitive to overfitting since it puts no or little emphasize on scores that are not close to the decision threshold, τ. Therefore they need more training data for robust parameter estimation and accordingly, weight-adjustment, should be more useful since it has a similar effect as increasing the training data. We evaluate this in Section 8.2.2.

- Since constrained DT schemes are not flexible enough to provide accurate LLRs for all OPs, application specific loss functions can be expected to be more useful for more constrained DT schemes. We evaluate this in Section 8.2.3.

8.2 Experiments

8.2.1 Weight-adjustment and constrained DT

In Chapter 5 we showed that the proposed weight-adjustment of the training trials improves the performance of the two baselines, Scr-UC with regularization toward $\mathbf{0}$, and At-Cal. In Chapter 6 we found the the proposed constrained DT scheme, Scr-4par outperformed AT-Cal on SRE10 but not on SRE08. In this experiment we explore the effect of weight-adjustment on Scr-4par. For simplicity, we did not preserve the balance between the NIST SRE and the Switchboard corpora. The same training data was used both in the GT step and the DT step.

Results using all training speakers

The results using all training data are given in Table 8.1 and 8.2 for the calibration-sensitive and calibration-insensitive evaluation metrics respectively. At-Cal which was the best baseline is included for comparison. For

Table 8.1: Combination of weight-adjustment and constrained DT. Results in calibration-sensitive evaluation metrics. The weight-adjustment parameter, α was tuned to optimize $\hat{C}_{\text{llr}}$ on the development set.

Set	Method.	actDCF08	actDCF10	$\hat{C}_{\text{llr}}$	Weight-adj
SRE08	At-Cal	0.0256	0.00130	0.201	no
		0.0253	0.00131	0.197	$\alpha = 1.0$
		0.0251	0.00130	0.199	sp.
	Scr-4par	0.0274	0.00257	0.202	no
		0.0276	0.00256	0.204	Sp.
SRE10	At-Cal	0.0143	0.000678	0.100	no
		0.0141	0.000688	0.098	$\alpha = 1.0$
		0.0141	0.000678	0.098	sp.
	Scr-4par	0.0117	0.000574	0.086	no
		0.0116	0.000569	0.085	Sp.

Table 8.2: Combination of weight-adjustment and constrained DT. Results in calibration-insensitive evaluation metrics. Notice that for these evaluation metrics, AT-Cal has no effect, i.e., the results are the same as if only GT had been used.

Set	Method.	minDCF08	minDCF10	EER	$\hat{C}_{\text{llr}}^{\text{min}}$	W.-adj
SRE08	At-Cal	0.0250	0.000728	0.0480	0.175	-
	Scr-4par	0.0254	0.000802	0.0471	0.177	no
		0.0254	0.000804	0.0476	0.178	Sp.
SRE10	At-Cal	0.0101	0.000385	0.0198	0.079	-
	Scr-4par	0.0100	0.000375	0.0188	0.0744	no
		0.0099	0.000375	0.0188	0.0742	Sp.

the one-parameter model, we did not obtain any improvement in $\hat{C}_{\text{llr}}$ on the development set. Therefore, only the $\alpha = 0$ is included. The effect of weight-adjustment based on sample correlations were small. In general, the minimum costs are much less effected by weight-adjustment than the actual costs.

Results using fewer training speakers

In this experiment, we evaluated Scr-4par and At-Cal for smaller numbers of training speakers, with and without weight-adjustment. Since previous experiments showed very small differences between weight-adjustment based on the one-parameter model based on sample correlations, we use only the former in this experiment. In Table 8.3 and 8.4 the results using half of the training speakers, for the calibration-sensitive and-insensitive evaluation metrics, respectively. In Fig. 8.1, $\hat{C}_{\text{llr}}$ vs. the number of training speakers is shown for SRE10. It is clear that Scr-4par gave better results than AT-cal and that the weight-adjustment in most cases improved the performance of both methods. The relative improvements of Scr-4par with weight-adjustment compared to the baseline, AT-Cal without weight-adjustment, ranged from 7% to 19% for the different training sizes. It is interesting that the gap between the two methods became larger when the amount of training data increased. This is reasonable since more training data is needed in order to take advantage of the extra flexibility of Scr-4par. However, in accordance with the experiment in Subsubsection 6.2, AT-Cal was better on SRE08 in most cases. As in the experiments with AT-Cal in Subsubsection 5.3.1, the effect of weight-adjustment disappears when the number of training speakers became large. For other evaluation metrics than $\hat{C}_{\text{llr}}$, the trend was less clear.

8.2.2 Weight-adjustment and application-specific loss functions

In this section, we apply weight-adjustment proposed in Chapter 5 and one of the application-specific loss functions presented in Chapter 7, the Brier loss, on Scr-UC. We choose the Brier loss since had a more stable behavior than the 0-1 loss and was effective for both SRE08 and SRE10. As in Chapter 7, we used the effective prior of DCF08, $P_{\text{eff}} = 0.0917$. We applied L2 regularization with the regularization parameter, ρ being tuned for $\alpha = 0$, i.e., no weight-adjustment. The results in the calibration sensitive evaluation metrics are shown in Table 8.5. It is clear that the benefits of weight-adjustment and the Brier loss are complementary for actDCF08. However, contrary to the expectation, the Brier loss did not benefit more than the logistic regression loss from weight-adjustment. This could perhaps partly be explained by the fact that the Brier loss gave a better actDCF08 and therefore is harder to

Table 8.3: Weight-adjustment for constrained DT using half of the training speakers. Results in calibration-sensitive evaluation metrics. The weight-adjustment parameter, α was tuned to optimize $\hat{C}_{\text{llr}}$ on the development set.

Set	Method	actDCF08	actDCF10	$\hat{C}_{\text{llr}}$	Weight-adj
SRE08	AT-Cal	0.0281	0.00092	0.201	no
		0.0286	0.00115	0.202	$\alpha = 0.2$
	Scr-4par	0.0322	0.00177	0.220	no
		0.0319	0.00217	0.219	$\alpha = 0.3$
SRE10	AT-Cal	0.0127	0.000743	0.101	no
		0.0125	0.000705	0.099	$\alpha = 0.2$
	Scr-4par	0.0116	0.000657	0.0950	no
		0.0114	0.000607	0.0934	$\alpha = 0.3$

Table 8.4: Weight-adjustment for constrained DT using half of the training speakers. Results in calibration-insensitive evaluation metrics. Notice that for these evaluation metrics, AT-Cal has no effect, i.e., the results are the same as if only GT had been used. The weight-adjustment parameter, α was tuned to optimize $\hat{C}_{\text{llr}}$ on the development set.

Set	Method	minDCF08	minDCF10	EER	$\hat{C}_{\text{llr}}^{\text{min}}$	Weight-adj
SRE08	AT-Cal	0.0263	0.000793	0.0523	0.187	-
	Scr-4par	0.0274	0.000921	0.0515	0.190	no
		0.0273	0.000910	0.0519	0.189	$\alpha = 0.3$
SRE10	AT-Cal	0.0111	0.000743	0.0229	0.090	-
	Scr-4par	0.0106	0.000657	0.0217	0.0848	no
		0.0106	0.000607	0.0219	0.0850	$\alpha = 0.3$

improve. As explained in Chapter 7, the application-specific loss functions improves the performance on one or a small range of OPs, on the expense of worsened performance on the other OPs. Since we target the OP corresponding to DCF08, it is not surprising that the performance is reduced for actDCF10 and $\hat{C}_{\text{llr}}$.

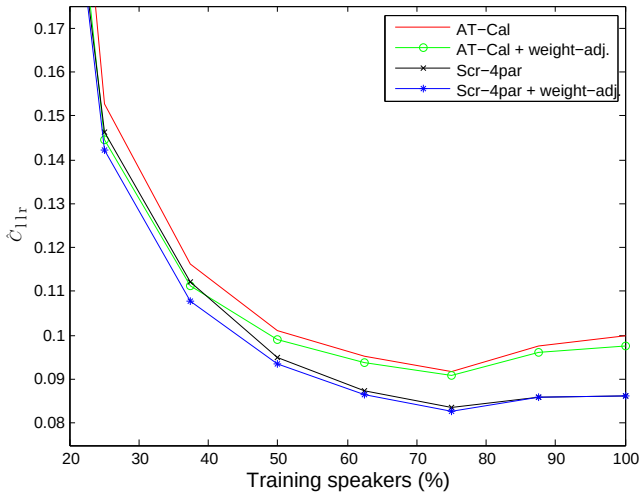

Figure 8.1: $\hat{C}_{\mathrm{llr}}$ for SRE10 vs. the percentage of training speakers. 100% equals 1152 speakers. The weight-adjustment parameter, α, was chosen to be optimal for the development set for each training-size.

Table 8.5: Combination of weight-adjustment and the Brier loss for Scr-UC. The results for the logistic regression loss are included for comparison. A '*' indicates that the value of the parameter was optimal on the development set, SRE06.

Set	Loss fcn.	Wght-adj.	actDCF08	actDCF10	$\hat{C}_{\mathrm{llr}}$	ρ
SRE08	Log. Reg.	$\alpha = 0$	0.0334	0.000876	0.235	*10
	Log. Reg.	$*\alpha = 2$	0.0317	0.000851	0.231	10
	Brier	$\alpha = 0$	0.0322	0.000938	0.285	*10
	Brier	$*\alpha = 3$	0.0309	0.000931	0.288	10
SRE10	Log. Reg.	$\alpha = 0$	0.0304	0.000916	0.180	*10
	Log. Reg.	$*\alpha = 2$	0.0278	0.000888	0.173	10
	Brier	$\alpha = 0$	0.0287	0.000978	0.229	*10
	Brier	$*\alpha = 3$	0.0270	0.000969	0.228	10

8.2.3 Constrained DT and application-specific loss functions

In this section we combine the best performing DT scheme of Chapter 6, Scr-4par, and the Brier loss presented in Chapter 7. As in previous experiments, we target actDCF08. The results are shown in Table 8.6. The most noticeable was the very bad results for other evaluation metrics than act-

Table 8.6: Scr-4par trained with the Brier loss. The results for Scr-4par and AT-Cal trained with the logistic regression loss are included for comparison.

Set	Loss fcn.	DT scheme	actDCF08	actDCF10	$\hat{C}_{\text{llr}}$
SRE08	Log. reg.	AT-Cal	0.0256	0.00130	0.201
	Log. reg.	Scr-4par	0.0274	0.00257	0.202
	Brier	Scr-4par	0.0257	0.01516	8633.739
SRE10	Log. reg.	AT-Cal	0.0143	0.000678	0.100
	Log. reg.	Scr-4par	0.0117	0.000574	0.086
	Brier	Scr-4par	0.0122	0.001482	5551.906

DCF08. An inspection of the estimated model parameters, a_{P}, a_{Q}, a_{c} and a_{k}, reveals that when the Brier loss is used for Scr-4par, they become very large in magnitude. Looking at Eq. (6.1), we can see that if all of these parameters equals one, the scores from the model obtained by GT are unchanged. On the other hand, if they have large magnitude, only a few scores corresponding to a small range of the *variables* $(\boldsymbol{\omega}_i^T \boldsymbol{P} \boldsymbol{\omega}_j + \boldsymbol{\omega}_j^T \boldsymbol{P} \boldsymbol{\omega}_i)$, $(\boldsymbol{\omega}_i^T \boldsymbol{Q} \boldsymbol{\omega}_i + \boldsymbol{\omega}_j^T \boldsymbol{Q} \boldsymbol{\omega}_j)$, $(\boldsymbol{\omega}_i + \boldsymbol{\omega}_j)^T \boldsymbol{c}$ and k will undergo a small change from the GT model. The rest of the scores will be changed largely. Assuming that the scores from the GT model were in reasonable range, this suggest that training procedure have cares almost only on the scores close to the threshold. Since the optimal decision threshold may vary slightly between the training set and evaluation set due to data mismatch, this behavior may not be beneficial even for the targeted OP which could explain the inconclusive results for actDCF08. It is worth noting that the problem with very large model parameters and accordingly, scores, did not occur for Scr-UC in the experiments in Section 7.2. The most likely explanation is that we used a regularization term for that method. Further, this problem will not occur for the logistic regression loss since the scores on the wrong side of the threshold would obtained too large costs due to the unboundedness of the logistic regression loss.

8.2.4 All three methods

In this section, we finally evaluate the combination of weight-adjustment proposed in Chapter 5, our best DT Scheme from Chapter 6, Scr-4par, and the Brier loss presented in Chapter 7. The results together with two base-

Table 8.7: Scr-4par trained with the Brier loss and weight-adjustment. The results for the two baselines are shown in the first and second row. The parameters α and ρ are the weight-adjustment and regularization parameter, respectively. $\alpha = 0$ means no weight-adjustment. A '*' indicates that the value of the parameter was optimal on the development set, SRE06.

Set	Loss	Scheme	α	actDCF08	actDCF10	$\hat{C}_{\text{llr}}$	ρ
SRE08	LR	AT-Cal	0	0.0256	0.00130	0.201	-
	LR	Scr-UC	0	0.0334	0.00088	0.235	*10
	Brier	Scr-4par	0	0.0257	0.01516	8633.739	-
	Brier	Scr-4par	*1	0.0266	0.01644	8384.825	-
SRE10	LR	AT-Cal	0	0.0143	0.000678	0.100	-
	LR	Scr-UC	0	0.0304	0.000916	0.180	*10
	Brier	Scr-4par	0	0.0122	0.001482	5551.906	-
	Brier	Scr-4par	*1	0.0117	0.001736	5272.156	-

lines, AT-Cal and Scr-UC trained with the logistic regression loss (LR) without weight-adjustment are shown in Table 8.7. Considering the previous results in this chapter, the result here are what could be expected. Applying the Brier loss to a DT scheme without regularization gives extreme values of the model parameters and bad performance for other OPs than the targeted one. The effect of weight-adjustment on Scr-4par for this training data size was marginal.

8.3 Summary and recommendations

In this section, we summarize the most important findings about weight-adjustment, constrained DT of PLDA, and the usage of application-specific loss functions as well as give recommendations for when to use the different methods and what to consider in such a case.

Weight-adjustment was almost always beneficial in our experiments. The effect of it is larger for smaller amounts of training data and less constrained DT schemes. The consequence of this method is that training speakers with many utterances obtains a lower weight per trial than training speakers with fewer utterances. This is intuitively sound since otherwise the model might become biased toward training speakers with many utter-

ances. However, as discussed in Chapter 6, there can be a bad side-effect of this when the training data is not sampled from a homogeneous population. For example, in our experiments the training data consists of several corpora and the average number of utterances per speaker differs for the different corpora. A corpora with many utterances per speaker will be down-weighted when weight-adjustment is applied which may not be good if this corpora is more similar to the evaluation set. Therefore, we recommend using weight-adjustment in general but if the training data inhomogeneous, one may have to compensate for it as we did in the experiments in Section 5.3.1.

Among the PLDA DT schemes, the baseline AT-Cal and our proposed Scr-4par were usually the best in our experiments. These are the two most constrained DT schemes. Further, our analysis in Section 6.3.1 showed that AT-Cal, which is most constrained, was better for utterances containing non-English speech but Scr-4par was better for utterances containing English speech. The training data contains very little non-English speech so this result is reasonable considering that less constrained DT schemes tends to overfit the training data more easily. There is therefore no DT scheme that is the best for any given situation. The general guideline is that the less training data that is available and the larger the mismatch between the training and the test data is, the more DT needs to be constrained.

The Brier loss proved useful for Scr-UC with L2 regularization for improving actDCF. For Scr-4par, the Brier loss gave strange results, most likely due to the lack of a regularization term. Scr-UC with Brier loss was not as good as Scr-4par with logistic regression loss for the training data sizes used in our experiments. However, if there is enough training data for Scr-UC to be the best DT scheme, combining it with the Brier loss and weight-adjustment seems promising for improving actDCF. Further, for smaller training data sizes, one could try, e.g., Scr-4par with the Brier loss and L2 regularization but more studies are needed before anything could be said for certain in this matter. The more extreme application-specific loss function, the 0-1 loss which focuses on only one OP, was quite unstable and is not recommended without further improvements.

Chapter 9

Conclusions and future work

9.1 Conclusions

In this thesis, we study discriminative training (DT) techniques in speaker verification. Recent speaker verification systems aims at directly answer the question whether two utterances are from the same speaker or not. Accordingly they do not require specific models for each enrolled speaker, which greatly reduces the amount of data needed to enroll a speaker. However, the model assumptions behind these systems are clearly inaccurate which motivates the use of DT. Indeed, previous work have shown that DT can improve these systems under certain conditions. Still, current approaches fail to take full advantage of DT. In this thesis we address three problems.

First, the training trials used in DT need to be constructed from the available training data. However, when a training utterance (or just the same speaker) is used in more than one trial, the trials will be statistically dependent. To solve this problem, we propose to adjust the weights of the trials in the training objective so that the training objective becomes a better estimator of the expected loss of unseen trials, i.e, test data. For a DT scheme with 2 parameters to be estimated, we observed relative improvements in $\hat{C}_{\text{llr}}$ of more than 30% on both SRE08 and SRE10 when using 11 training speakers. For around 1000 speakers, the effect of weight-adjustment was minor for this DT scheme but for a DT scheme with a much larger number of parameters to be estimated, the effect of weight adjustment was substantial.

Second, DT more easily overfits to the training data than generative training. Previously proposed DT schemes are either very constrained, or

hardly constrained at all. In order to find just the right constraints, we propose three new constrained DT schemes, and systematically compare them with existing training schemes. With one of these DT schemes, we obtained a relative improvement in $\hat{C}_{\text{llr}}$ of 14% for SRE10. However, one of the baselines, performed the best for SRE08. This was explained by the presence of non-English speech in SRE08. Excluding the non-English trials, our method improved the results for SRE08 as well. In combination with weight-adjustment, our proposed constrained DT scheme gave improvements in between 7% and 19% in $\hat{C}_{\text{llr}}$ on SRE10, depending on the training data size, compared to the best of our baselines.

Third, to follow the true spirit of DT, one shall use the loss function that is the relevant for the application. This is, however, not always the best in practice. Application-specific loss functions are non-convex and more vulnerable to overfitting. We evaluate several different loss functions as well as examines training strategies to deal with their non-convexity. Using application-specific loss functions, we obtained a reduction of actDCF08 of 3.6% on SRE08 and 5.6% on SRE10 compared to logistic regression loss.

9.2 Future work

Future directions are many. There are other phenomena that may cause the training trials to be statistically dependent than common utterances or speakers. For example, when the same microphone is used in more than one training utterance. It would be interesting to apply the weight-adjustment to deal with such dependencies. Although using the best linear unbiased estimator for the expected loss is well motivated and works well, is possible that the results could be improved by some other estimator than the BLUE. For example, a non-linear estimator or an estimator that considers higher moments than the variance. Our experiments indicate that the optimization of the non-convex application-specific loss functions is difficult. Future work will therefore include better optimization techniques. After a better optimization strategy have been found, it would be interesting to investigate more in detail what is the best loss function in various situations. In particular, to analyze more in detail if focusing on a broad range of OPs is more effective than other means of regularization. Another issue for future consideration is that there might be a mismatch between the properties of

the training trials and the properties of the test trials. Several studies in domain adaptation have shown that the Switchboard and the NIST SRE corpora have different properties (Garcia-Romero and McCree, 2014; Biswas et al., 2015). Furthermore, the target trials in the test sets of the NIST SRE are always from different telephone numbers whereas the majority of the target trials used for DT are from the same telephone number.

Chapter 10

Publications

Publications related to this thesis

International journal (peer-reviewed)

Johan Rohdin, Sangeeta Biswas and Koichi Shinoda, *Robust discriminative training against data insufficiency in PLDA-based speaker verification*, Computer Speech and Language, Vol. 35, pp. 32-57, Jan. 2016.

International conferences (peer-reviewed)

Johan Rohdin, Sangeeta Biswas and Koichi Shinoda, *Discriminative PLDA training with application-specific loss functions in speaker verification*, in *Proc. Odyssey*, ISCA, pp. 26-32, 2014.

Johan Rohdin, Sangeeta Biswas and Koichi Shinoda, *Constrained discriminative PLDA training for Speaker Verification*, in *Proc. ICASSP*, IEEE, pp. 1689-1693, 2014.

Domestic conference (not peer-reviewed)

Johan Rohdin, Sangeeta Biswas and Koichi Shinoda, *Robust 0-1 loss training for PLDA in Speaker Verification*, 2014 Spring Meeting of the Acoustical Society of Japan, 3-4-14, pp. 101-102, Mar. 12, 2014.

Johan Rohdin, Sangeeta Biswas and Koichi Shinoda, *Discriminatively Trained PLDA with Partially Preserved Model Assumptions in Speaker*

Verification, 2014 Spring Meeting of the Acoustical Society of Japan, 3-4-13, pp. 99-100, Mar. 12, 2014.

Other publications

International journal (peer-reviewed)

Sangeeta Biswas, **Johan Rohdin** and Koichi Shinoda, *Autonomous Selection of i-Vectors for PLDA modelling in Speaker verification*, Speech Communication, Vol. 72, pp. 32-46, Sep. 2015.

International conference (peer-reviewed)

Sangeeta Biswas, **Johan Rohdin** and Koichi Shinoda, *i-Vector Selection for Effective PLDA Modeling in Speaker Recognition*, in *Proc. Odyssey*, ISCA, pp. 100-105, 2014.

International meeting (not peer-reviewed)

Sangeeta Biswas, **Johan Rohdin** and Koichi Shinoda, *Tokyo Tech Speaker Recognition*, System description. NIST SRE 2012, Dec.11, 2012.

Domestic conferences (not peer-reviewed)

Sangeeta Biswas, **Johan Rohdin** and Koichi Shinoda, *Training Multiple PLDA models by Clustered I-Vectors for Speaker verification*, 2014 Spring Meeting of the Acoustical Society of Japan, 3-4-12, pp. 97-98, Mar. 12, 2014.

Johan Rohdin and Koichi Shinoda, *Speaker Adaptation for Dialog Act Recognition*, Conference Proceedings, 2012 Spring Meeting of the Acoustical Society of Japan, 3-7-12 pp. 111-112, Mar. 13, 2012.

Sangeeta Biswas, **Johan Rohdin**, Koichi Shinoda and Sadaoki Furui, *MAP Adaptation Using Multiple Priors for Speaker Verification*, Conference Proceedings, 2012 Spring Meeting of the Acoustical Society of Japan, 3-7-1 pp. 79-82, Mar. 13, 2012.

Sangeeta Biswas, **Johan Rohdin**, Koichi Shinoda and Sadaoki Furui, *Speaker Verification Using MMAP Adaptation*, Technical report, The Institute of Electronics, Information and Communication Engineers, No. SP2011-93, no. 23, Dec. 19, 2011.

Johan Rohdin, and Koichi Shinoda, *Speaker Adaptation for Dialogue Act Classification*, Technical report, Information Processing Society of Japan (SLP), vol.2011-SLP-87, no. 8, Jul. 21, 2011.

Appendix A

Derivations

A.1 The EM algorithm for PLDA

In this section we describe the EM algorithm with *minimum divergence* (MD) for PLDA with the configuration used in this thesis, i.e.,

$$\boldsymbol{\omega} = \boldsymbol{m} + \boldsymbol{V}\boldsymbol{y} + \boldsymbol{\epsilon}, \tag{A.1}$$

where $\epsilon \sim \mathcal{N}(0, \tilde{\boldsymbol{D}}^{-1})$. With $\boldsymbol{\epsilon} = \boldsymbol{D}\boldsymbol{z}$ and $\tilde{\boldsymbol{D}}^{-1} = \boldsymbol{D}\boldsymbol{D}^T$ (such a decomposition always exists) the model in Eq (A.1) equals the model in Eq (2.35). The presentation is based on Brümmer (2010) where the EM-algorithm for PLDA including a channel term $\boldsymbol{U}\boldsymbol{x}$ (as given by Eq 2.34) is presented.

The maximum likelihood criterion is given by:

$$[\hat{\boldsymbol{m}}, \hat{\boldsymbol{V}}, \hat{\tilde{\boldsymbol{D}}}] = \arg\max \prod_{s=1}^{S} \prod_{e=1}^{E_s} (\boldsymbol{\omega}_{se} | \boldsymbol{m}, \boldsymbol{V}, \tilde{\boldsymbol{D}}), \tag{A.2}$$

where index s indicates the speaker, index e indicates the session, S is the number of speakers and E_s is the number of sessions for speaker s. If the i-vectors of the training set are centered around their mean, the ML estimate of $\boldsymbol{m}$ is $\boldsymbol{0}$.

We will denote the parameters to be optimized $\boldsymbol{\zeta}$. Initially, $\boldsymbol{\zeta} = [\boldsymbol{V}, \tilde{\boldsymbol{D}}]$ but in MD step we will (temporarily) consider more parameters. Let $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_S]$, $\Psi_s = [\boldsymbol{\omega}_{s1}, \ldots, \boldsymbol{\omega}_{sE_s}]$, and $\boldsymbol{\Psi} = [\Psi_s, \ldots, \Psi_S]$. We will use the following notation for the expectation conditioned on the current model

parameters:

$$\mathbb{E}_{\boldsymbol{Y}|\boldsymbol{\Psi},\boldsymbol{\zeta}^{old}}\left[f(\boldsymbol{\zeta})\right] = \int_{\boldsymbol{y}} f(\boldsymbol{\zeta}) P(\boldsymbol{y}_s|\boldsymbol{\Psi}_s,\boldsymbol{\zeta}^{old})\mathrm{d}^r\boldsymbol{y}_s$$
$$= \Big\langle f(\boldsymbol{\zeta}) \Big\rangle \tag{A.3}$$

A.1.1 E-Step

The E-step is to calculate the auxiliary function:

$$Q(\boldsymbol{\zeta}|\boldsymbol{\zeta}^{old}) = \Big\langle \log P(\boldsymbol{\Psi},\boldsymbol{Y}|\boldsymbol{\zeta}) \Big\rangle$$
$$= \Big\langle \sum_{s=1}^{S} \log P(\boldsymbol{\Psi}_s,\boldsymbol{y}_s|\boldsymbol{\zeta}) \Big\rangle$$
$$= \Big\langle \sum_{s=1}^{S} \log P(\boldsymbol{\Psi}_s|\boldsymbol{y}_s,\boldsymbol{\zeta}) P(\boldsymbol{y}_s|\boldsymbol{\zeta}) \Big\rangle$$
$$= \Big\langle \sum_{s=1}^{S}\sum_{e=1}^{E_s} \log P(\boldsymbol{\omega}_e|\boldsymbol{y}_s,\boldsymbol{\zeta}) \Big\rangle \tag{A.4}$$
$$+ \Big\langle \sum_{s} \log P(\boldsymbol{y}_s|\boldsymbol{\zeta}) \Big\rangle. \tag{A.5}$$

We will refer to the term (A.4) as $Q_1(\boldsymbol{\zeta}|\boldsymbol{\zeta}^{old})$ and to the term (A.5) as $Q_2(\boldsymbol{\zeta}|\boldsymbol{\zeta}^{old})$. Since $Q_2(\boldsymbol{\zeta}|\boldsymbol{\zeta}^{old})$ does not depend on $\boldsymbol{V}$ or $\tilde{\boldsymbol{D}}$, it can be ignored. (It will considered in the MD step).

For $Q_1(\boldsymbol{\zeta}|\boldsymbol{\zeta}^{old})$ we have

$$
\begin{aligned}
Q_1(\boldsymbol{\zeta}|\boldsymbol{\zeta}^{old}) &= \Big\langle \sum_{s=1}^{S}\sum_{e=1}^{E_s} \log P(\boldsymbol{\Psi}_s, |\boldsymbol{y}_s, \boldsymbol{\zeta}) \Big\rangle \\
&= \frac{1}{2}\Big\langle \sum_{s=1}^{S}\sum_{e=1}^{E_s} \Big[-(\boldsymbol{\omega}_{se} - \boldsymbol{V}\boldsymbol{y_s})^T \tilde{\boldsymbol{D}}(\boldsymbol{\omega}_{se} - \boldsymbol{V}\boldsymbol{y_s}) + \log|\tilde{\boldsymbol{D}}| \\
&\qquad\qquad -r\log 2\pi \Big] \Big\rangle \\
&= \frac{1}{2}\Big\langle \sum_{s=1}^{S}\sum_{e=1}^{E_s} \Big[-\boldsymbol{\omega}_{se}^T\tilde{\boldsymbol{D}}\boldsymbol{\omega}_{se} - \boldsymbol{y}_s^T\boldsymbol{V}^T\tilde{\boldsymbol{D}}\boldsymbol{V}\boldsymbol{y}_s + \boldsymbol{\omega}_{se}^T\tilde{\boldsymbol{D}}\boldsymbol{V}\boldsymbol{y}_s \\
&\qquad\qquad +\boldsymbol{y}_s^T\boldsymbol{V}^T\tilde{\boldsymbol{D}}\boldsymbol{\omega}_{se} + \log|\tilde{\boldsymbol{D}}| \Big] \Big\rangle + const \\
&= \frac{1}{2}\Big\langle \sum_{s=1}^{S}\sum_{e=1}^{E_s} \Big[-\mathrm{tr}[\boldsymbol{\omega}_{se}\boldsymbol{\omega}_{se}^T\tilde{\boldsymbol{D}} - \boldsymbol{V}\boldsymbol{y}_s\boldsymbol{y}_s^T\boldsymbol{V}^T\tilde{\boldsymbol{D}} + 2\boldsymbol{V}\boldsymbol{y}_s\boldsymbol{\omega}_{se}^T\tilde{\boldsymbol{D}}] \\
&\qquad\qquad -\log|\tilde{\boldsymbol{D}}| \Big] \Big\rangle + const \\
&= \frac{N}{2}\log|\tilde{\boldsymbol{D}}| - \frac{1}{2}\mathrm{tr}\Big[\sum_{s=1}^{S}\sum_{e=1}^{E_s} \boldsymbol{\omega}_{se}\boldsymbol{\omega}_{se}^T\tilde{\boldsymbol{D}} - \sum_{s=1}^{S}\sum_{e=1}^{E_s} \Big\langle \boldsymbol{y}_s\boldsymbol{y}_s^T \Big\rangle \boldsymbol{V}^T\tilde{\boldsymbol{D}}\boldsymbol{V} \\
&\qquad\qquad +2\sum_{s=1}^{S}\sum_{e=1}^{E_s} \Big\langle \boldsymbol{y}_s \Big\rangle \boldsymbol{\omega}_{se}^T\tilde{\boldsymbol{D}}\boldsymbol{V} \Big] + const \\
&= \frac{N}{2}\log|\tilde{\boldsymbol{D}}| - \frac{1}{2}\mathrm{tr}\Big[\boldsymbol{S}\tilde{\boldsymbol{D}} + \boldsymbol{R}\boldsymbol{V}^T\tilde{\boldsymbol{D}}\boldsymbol{V} - 2\boldsymbol{T}\tilde{\boldsymbol{D}}\boldsymbol{V} \Big] + const, \quad \text{(A.6)}
\end{aligned}
$$

where $const$ refers to terms that does not contain $\boldsymbol{V}$ or $\boldsymbol{D}$, $\boldsymbol{S} = \sum_{se}\boldsymbol{\omega}_{se}\boldsymbol{\omega}_{se}^T$,

$$
\boldsymbol{R} = \sum_{s=1}^{S}\sum_{e=1}^{E_s} \Big\langle \boldsymbol{y}_s\boldsymbol{y}_s^T \Big\rangle \tag{A.7}
$$

and

$$
\boldsymbol{T} = \sum_{s=1}^{S}\sum_{e=1}^{E_s} \Big\langle \boldsymbol{y}_s \Big\rangle \boldsymbol{\omega}_{se}^T. \tag{A.8}
$$

In order to calculate these expectations, we need the posterior distribution

of $\boldsymbol{y}_s$,

$$\begin{aligned}
\Pr(\boldsymbol{y}_s|\Psi_s,\zeta^{old}) &\propto \Pr(\boldsymbol{y}_s|\zeta^{old})\Pr(\Psi_s|\boldsymbol{y}_s,\zeta^{old}) \\
&= \mathcal{N}(\boldsymbol{y}_s|\boldsymbol{0},\boldsymbol{I})\prod_{e=1}^{E_s}\mathcal{N}(\boldsymbol{\omega}_{se}|\boldsymbol{V}\boldsymbol{y}_s,\tilde{\boldsymbol{D}}) \\
&\propto \exp -\frac{1}{2}\Big(\boldsymbol{y}_s^T\boldsymbol{I}\boldsymbol{y}_s + \sum_{e=1}^{E_s}(\boldsymbol{\omega}_{se}-\boldsymbol{V}\boldsymbol{y}_s)^T\tilde{\boldsymbol{D}}(\boldsymbol{\omega}_{se}-\boldsymbol{V}\boldsymbol{y}_s)\Big) \\
&\propto \exp\Big(\sum_{e=1}^{E_s}\boldsymbol{y}_s^T\boldsymbol{V}^T\tilde{\boldsymbol{D}}\boldsymbol{\omega}_{se} - \frac{1}{2}\boldsymbol{y}_s^T\underbrace{(E_s\boldsymbol{V}^T\tilde{\boldsymbol{D}}\boldsymbol{V}+I)}_{\boldsymbol{P}_s}\boldsymbol{y}_s\Big) \\
&= \exp\Big(\boldsymbol{y}_s^T\boldsymbol{P}_s\underbrace{\boldsymbol{P}_s^{-1}\boldsymbol{V}^T\tilde{\boldsymbol{D}}\sum_{e=1}^{E_s}\boldsymbol{\omega}_{se}}_{\hat{\boldsymbol{y}}_s} - \frac{1}{2}\boldsymbol{y}_s^T\boldsymbol{P}_s\boldsymbol{y}_s\Big) \\
&\propto \mathcal{N}(\boldsymbol{y}_s|\hat{\boldsymbol{y}}_s,\boldsymbol{P}_s^{-1}).
\end{aligned} \tag{A.9}$$

This gives

$$\boldsymbol{R} = \sum_{s=1}^{S}\sum_{e=1}^{E_s}\Big\langle \boldsymbol{y}_s\boldsymbol{y}_s^T\Big\rangle = \sum_{s=1}^{S}E_s(\boldsymbol{P}^{-1}+\hat{\boldsymbol{y}}_s\hat{\boldsymbol{y}}_s{}^T) \tag{A.10}$$

and

$$\boldsymbol{T} = \sum_{s=1}^{S}\sum_{e=1}^{E_s}\Big\langle \boldsymbol{y}_s\Big\rangle\boldsymbol{\omega}_{se} = \sum_{s=1}^{S}\hat{\boldsymbol{y}}_s\sum_{e=1}^{E_s}\boldsymbol{\omega}_{se}^T. \tag{A.11}$$

A.1.2 M-Step

Using the matrix calculus results in Minka (2001) and the cyclic properties of the trace operator, the differential is then given by

$$\begin{aligned}
\mathrm{d}(Q_1) &= \mathrm{d}\left(\frac{N}{2}\log|\tilde{\boldsymbol{D}}| - \frac{1}{2}\mathrm{tr}\Big[\boldsymbol{S}\tilde{\boldsymbol{D}} + \boldsymbol{R}\boldsymbol{V}^T\tilde{\boldsymbol{D}}\boldsymbol{V} - 2\boldsymbol{T}\tilde{\boldsymbol{D}}\boldsymbol{V}\Big]\right) \\
&= -\frac{1}{2}\mathrm{tr}\Big[-N\tilde{\boldsymbol{D}}^{-1}\mathrm{d}\tilde{\boldsymbol{D}} + (\mathrm{d}\boldsymbol{S})\tilde{\boldsymbol{D}} + \boldsymbol{S}\mathrm{d}\tilde{\boldsymbol{D}} \\
&\qquad +(\mathrm{d}\boldsymbol{R})\boldsymbol{V}^T\tilde{\boldsymbol{D}}\boldsymbol{V} + \boldsymbol{R}(\mathrm{d}\boldsymbol{V}^T)\tilde{\boldsymbol{D}}\boldsymbol{V} + \boldsymbol{R}\boldsymbol{V}^T(\mathrm{d}\tilde{\boldsymbol{D}})\boldsymbol{V} + \boldsymbol{R}\boldsymbol{V}^T\tilde{\boldsymbol{D}}\mathrm{d}\boldsymbol{V} \\
&\qquad -2(\mathrm{d}\boldsymbol{T})\tilde{\boldsymbol{D}}\boldsymbol{V} - 2\boldsymbol{T}(\mathrm{d}\tilde{\boldsymbol{D}})\boldsymbol{V} - 2\boldsymbol{T}\tilde{\boldsymbol{D}}\mathrm{d}\boldsymbol{V}\Big] \\
&= -\frac{1}{2}\mathrm{tr}\Big[-N\tilde{\boldsymbol{D}}^{-1}\mathrm{d}\tilde{\boldsymbol{D}} + \tilde{\boldsymbol{D}}\mathrm{d}\boldsymbol{S} + \boldsymbol{S}\mathrm{d}\tilde{\boldsymbol{D}} \\
&\qquad +\boldsymbol{V}^T\tilde{\boldsymbol{D}}\boldsymbol{V}\mathrm{d}\boldsymbol{R} + \boldsymbol{R}\boldsymbol{V}^T\tilde{\boldsymbol{D}}\mathrm{d}\boldsymbol{V} + \boldsymbol{V}\boldsymbol{R}\boldsymbol{V}^T\mathrm{d}\tilde{\boldsymbol{D}} + \boldsymbol{R}\boldsymbol{V}^T\tilde{\boldsymbol{D}}\mathrm{d}\boldsymbol{V} \\
&\qquad -2\tilde{\boldsymbol{D}}\boldsymbol{V}\mathrm{d}\boldsymbol{T} - 2\boldsymbol{V}\boldsymbol{T}\mathrm{d}\tilde{\boldsymbol{D}} - 2\boldsymbol{T}\tilde{\boldsymbol{D}}\mathrm{d}\boldsymbol{V}\Big].
\end{aligned} \tag{A.12}$$

Reading off the coefficients for $\mathrm{d}\boldsymbol{V}$, we get

$$\frac{\mathrm{d}(Q_1)}{\mathrm{d}\boldsymbol{V}} = \boldsymbol{T}\tilde{\boldsymbol{D}} - \boldsymbol{R}\boldsymbol{V}^T\tilde{\boldsymbol{D}}. \tag{A.13}$$

Equating it to 0 gives

$$\boldsymbol{V}^T = \boldsymbol{R}^{-1}\boldsymbol{T}. \tag{A.14}$$

Reading off the coefficients for $\mathrm{d}\tilde{\boldsymbol{D}}$, we get

$$\frac{\mathrm{d}(Q_1)}{\mathrm{d}\tilde{\boldsymbol{D}}} = -\frac{1}{2}\Big[- N\tilde{\boldsymbol{D}}^{-1} + \boldsymbol{S} + \boldsymbol{V}\boldsymbol{R}\boldsymbol{V}^T - 2\boldsymbol{V}\boldsymbol{T}\Big]. \tag{A.15}$$

Equating it to 0 gives

$$\begin{aligned}\tilde{\boldsymbol{D}}^{-1} &= \frac{1}{N}\Big(\boldsymbol{S} - 2\boldsymbol{V}\boldsymbol{T} + \boldsymbol{V}\boldsymbol{R}\boldsymbol{V}^T\Big). \\ &= \frac{1}{N}\Big(\boldsymbol{S} - \boldsymbol{V}\boldsymbol{T}\Big)\end{aligned} \tag{A.16}$$

A.1.3 Minimum divergence

In the minimum divergence step we first use a more general prior for the $\boldsymbol{Y}$:

$$p(\boldsymbol{y}_s) + \mathcal{N}(\boldsymbol{y}_s|\boldsymbol{0}, \mathcal{Y}), \tag{A.17}$$

and estimate $\mathcal{Y}$. Q_2 but not Q_1 depends on $\mathcal{Y}$.

$$\begin{aligned}Q_2(\mathcal{Y}|\mathcal{Y}^{old}) &= \Big\langle \sum_s \log P(\boldsymbol{y}_s|\mathcal{Y}) P(\boldsymbol{y}_s|\boldsymbol{\Psi}_s, \mathcal{Y}^{old})\Big\rangle \\ &= -\frac{1}{2}\Big\langle \sum_s \log|\mathcal{Y}| + \boldsymbol{y}_s^T\mathcal{Y}^{-1}\boldsymbol{y}_s P(\boldsymbol{y}_s|\boldsymbol{\Psi}_s, \mathcal{Y}^{old})\Big\rangle + const \\ &= -\frac{1}{2}\Big\langle \sum_s \log|\mathcal{Y}| + \mathrm{tr}\Big(\boldsymbol{y}_s\boldsymbol{y}_s^T\mathcal{Y}^{-1}\Big) P(\boldsymbol{y}_s|\boldsymbol{\Psi}_s, \mathcal{Y}^{old})\Big\rangle + const\end{aligned} \tag{A.18}$$

The differential is given b

$$\begin{aligned}\frac{\mathrm{d}Q_2}{\mathrm{d}\mathcal{Y}} &= -\frac{1}{2}\mathrm{tr}\Big\langle \sum_s \Big(\mathcal{Y}^{-1}\mathrm{d}\mathcal{Y} + (\mathrm{d}\boldsymbol{y}_s\boldsymbol{y}_s^T)\mathcal{Y}^{-1} + \boldsymbol{y}_s\boldsymbol{y}_s^T\mathrm{d}\mathcal{Y}^{-1}\Big)\Big\rangle \\ &= -\frac{1}{2}\mathrm{tr}\Big\langle \sum_s \Big(\mathcal{Y}^{-1}\mathrm{d}\mathcal{Y} + (\mathrm{d}\boldsymbol{y}_s\boldsymbol{y}_s^T)\mathcal{Y}^{-1} - \boldsymbol{y}_s\boldsymbol{y}_s^T\mathcal{Y}^{-1}\mathrm{d}\mathcal{Y}\mathcal{Y}^{-1}\Big)\Big\rangle \\ &= -\frac{1}{2}\mathrm{tr}\Big\langle \sum_s \Big(\mathcal{Y}^{-1}\mathrm{d}\mathcal{Y} + (\mathrm{d}\boldsymbol{y}_s\boldsymbol{y}_s^T)\mathcal{Y}^{-1} - \mathcal{Y}^{-1}\boldsymbol{y}_s\boldsymbol{y}_s^T\mathcal{Y}^{-1}\mathrm{d}\mathcal{Y}\Big)\Big\rangle\end{aligned} \tag{A.19}$$

Equating the coefficients in front of $\mathrm{d}\mathcal{Y}$ to 0 gives:

$$\begin{aligned}S\mathcal{Y}^{-1} &= \sum_s \mathcal{Y}^{-1} < \boldsymbol{y}_s\boldsymbol{y}_s^T > \mathcal{Y}^{-1} \\ \mathcal{Y} &= \frac{1}{S}\sum_s < \boldsymbol{y}_s\boldsymbol{y}_s^T > \\ &= \boldsymbol{P}^{-1} + \hat{\boldsymbol{y}}\hat{\boldsymbol{y}}^T.\end{aligned} \tag{A.20}$$

A.2 Constraints on the PLDA LLR score function

In this section, we derive the constraints on $\boldsymbol{P}$ and $\boldsymbol{Q}$ mentioned in Subsection 2.6.4. In this paper, we use the term *definite* only for symmetric matrices. We use the term *semidefinite* when at least one eigenvalue of the matrix is zero, i.e., it does not have full rank, and the term *nonnegative-definite* for matrices which are either positive-definite or positive-semidefinite.

A.2.1 Rank of $\boldsymbol{P}$ and $\boldsymbol{Q}$

In this subsection, we show that both the rank of $\boldsymbol{P}$ and the rank of $\boldsymbol{Q}$ is equal to the rank of $\boldsymbol{V}$.

Let $\boldsymbol{S} = \Sigma_{\text{tot}} - \Sigma_{\text{ac}}\Sigma_{\text{tot}}^{-1}\Sigma_{\text{ac}}$. Then,

$$\text{rank}(\boldsymbol{P}) = \text{rank}(\Sigma_{\text{tot}}^{-1}\Sigma_{\text{ac}}\boldsymbol{S}^{-1}) \leq \text{rank}(\Sigma_{\text{ac}}), \tag{A.21}$$

$$\text{rank}(\Sigma_{\text{ac}}) = \text{rank}(\Sigma_{\text{tot}}\boldsymbol{P}\boldsymbol{S}) \leq \text{rank}(\boldsymbol{P}). \tag{A.22}$$

Hence, $\text{rank}(\boldsymbol{P}) = \text{rank}(\Sigma_{\text{ac}}) = \text{rank}(\boldsymbol{V})$.

Using that $\boldsymbol{S}$ is positive definite (Boyd and Vandenberghe, 2004, Ch. A.5.5) and the Woodbury identity (Petersen and Pedersen, 2012, Eq. (156)) we obtain

$$\begin{aligned}\boldsymbol{Q} &= -\Sigma_{\text{tot}}^{-1}\Sigma_{\text{ac}}\boldsymbol{S}^{-1}\Sigma_{\text{ac}}\Sigma_{\text{tot}}^{-1} \\ &= -\Sigma_{\text{tot}}^{-1}\Sigma_{\text{ac}}\boldsymbol{S}^{-1/2}(\Sigma_{\text{tot}}^{-1}\Sigma_{\text{ac}}\boldsymbol{S}^{-1/2})^T,\end{aligned} \tag{A.23}$$

where $\boldsymbol{S}^{-1/2}$ is the square root of $\boldsymbol{S}^{-1}$. Set $\boldsymbol{M} = \Sigma_{\text{tot}}^{-1}\Sigma_{\text{ac}}\boldsymbol{S}^{-1/2}$. Then,

$$\text{rank}(\boldsymbol{Q}) = \text{rank}(\boldsymbol{M}) \leq \text{rank}(\Sigma_{\text{ac}}), \tag{A.24}$$

$$\text{rank}(\Sigma_{\text{ac}}) = \text{rank}(\Sigma_{\text{tot}}\boldsymbol{M}\boldsymbol{S}^{1/2}) \leq \text{rank}(\boldsymbol{M}). \tag{A.25}$$

Hence, $\text{rank}(\boldsymbol{Q}) = \text{rank}(\Sigma_{\text{ac}}) = \text{rank}(\boldsymbol{V})$.

A.2.2 Definiteness of $\boldsymbol{P}$ and $\boldsymbol{Q}$

In this subsection, we derive the following constraints on $\boldsymbol{P}$ and $\boldsymbol{Q}$:

1. $\boldsymbol{Q}$ is negative-(semi)definite.
2. $\boldsymbol{P}$ is positive-(semi)definite.
3. $\boldsymbol{P} + \boldsymbol{Q}$ is positive-(semi)definite.

For these constraints, *semi* applies when rank$(V) < d$. Constraint 1 follows directly from Eq. (A.23) and the fact that rank$(\boldsymbol{Q}) =$ rank$(\boldsymbol{V})$. If $\boldsymbol{SPS}$ is positive-(semi)definite, then $\boldsymbol{S}^{-1}\boldsymbol{SPSS}^{-1} = \boldsymbol{P}$ is positive (semi)definite (Harville, 1997, Thm. 14.2.9). A bit of processing of $\boldsymbol{SPS}$ gives

$$\boldsymbol{SPS} = \Sigma_{\text{ac}} - \Sigma_{\text{ac}}\Sigma_{\text{tot}}^{-1}\Sigma_{\text{ac}} + \Sigma_{\text{ac}}\Sigma_{\text{tot}}^{-1}\Sigma_{\text{wc}}\Sigma_{\text{tot}}^{-1}\Sigma_{\text{ac}}, \tag{A.26}$$

where $\Sigma_{\text{wc}} = \Sigma_{\text{tot}} - \Sigma_{\text{ac}}$. From Eq. (A.26) it is clear that $\boldsymbol{P}$ is symmetric. The last term in Eq. (A.26) is nonnegative-definite. The term $\boldsymbol{Z} = \Sigma_{\text{ac}} - \Sigma_{\text{ac}}\Sigma_{\text{tot}}^{-1}\Sigma_{\text{ac}}$ is a Schur complement of Σ_{tot} in

$$\boldsymbol{M} = \begin{bmatrix} \Sigma_{\text{tot}} & \Sigma_{\text{ac}} \\ \Sigma_{\text{ac}} & \Sigma_{\text{ac}} \end{bmatrix}. \tag{A.27}$$

$\boldsymbol{Z}$ is positive-(semi)definite if $\boldsymbol{M}$ is positive-(semi)definite (Boyd and Vandenberghe, 2004, Ch. A.5.5). By expanding $[\boldsymbol{x}_1^T\ \boldsymbol{x}_2^T]\boldsymbol{M}[\boldsymbol{x}_1^T\ \boldsymbol{x}_2^T]^T$ for two real vectors, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, it can be verified that $\boldsymbol{M}$ is positive definite if rank$(\Sigma_{\text{ac}}) = d$, otherwise positive-semidefinite. Since, $\boldsymbol{SPS}$, is a sum of nonnegative-definite matrices, it is nonnegative definite. Since rank$(\boldsymbol{P}) =$ rank$(\boldsymbol{V})$, Constraint 2 follows.

$\boldsymbol{P} + \boldsymbol{Q}$ can be rewritten as

$$\begin{aligned} \boldsymbol{P} + \boldsymbol{Q} &= \Sigma_{\text{tot}}^{-1}\Sigma_{\text{ac}}\boldsymbol{S}^{-1}(I - \Sigma_{\text{ac}}\Sigma_{\text{tot}}^{-1}) \\ &= \boldsymbol{S}^{-1}\Sigma_{\text{ac}}\Sigma_{\text{tot}}^{-1}(I - \Sigma_{\text{ac}}\Sigma_{\text{tot}}^{-1}) \qquad \text{(A.28)} \\ &= \boldsymbol{S}^{-1}(\Sigma_{\text{ac}} - \Sigma_{\text{ac}}\Sigma_{\text{tot}}^{-1}\Sigma_{\text{ac}})\Sigma_{\text{tot}}^{-1}, \qquad \text{(A.29)} \end{aligned}$$

where (A.28) comes from the symmetry of $\boldsymbol{P} = \Sigma_{\text{tot}}^{-1}\Sigma_{\text{ac}}\boldsymbol{S}^{-1}$. If $\boldsymbol{S}^{1/2}(\boldsymbol{P} + \boldsymbol{Q})\boldsymbol{S}^{1/2}$ is positive (semi)definite, then $\boldsymbol{S}^{-1/2}\boldsymbol{S}^{1/2}(\boldsymbol{P}+\boldsymbol{Q})\boldsymbol{S}^{1/2}\boldsymbol{S}^{-1/2} = \boldsymbol{P} + \boldsymbol{Q}$ is positive (semi)definite. Since this matrix is symmetric, it is enough to show that its eigenvalues are larger than, or equal to zero. For two matrices, $\boldsymbol{A}$ and $\boldsymbol{B}$, the eigenvalues of $\boldsymbol{AB}$ and $\boldsymbol{BA}$ are the same (Harville, 1997, Thm. 21.10.1). Therefore, the eigenvalues of $\boldsymbol{S}^{1/2}(\boldsymbol{P} + \boldsymbol{Q})\boldsymbol{S}^{1/2}$ are the same as for

$$\boldsymbol{S}(\boldsymbol{P} + \boldsymbol{Q}) = (\Sigma_{\text{ac}} - \Sigma_{\text{ac}}\Sigma_{\text{tot}}^{-1}\Sigma_{\text{ac}})\Sigma_{\text{tot}}^{-1/2}\Sigma_{\text{tot}}^{-1/2}, \tag{A.30}$$

whose eigenvalues in turn are the same as for

$$\Sigma_{\text{tot}}^{-1/2}(\Sigma_{\text{ac}} - \Sigma_{\text{ac}}\Sigma_{\text{tot}}^{-1}\Sigma_{\text{ac}})\Sigma_{\text{tot}}^{-1/2}. \tag{A.31}$$

The middle part is, as pointed out earlier, positive-definite if rank$(\boldsymbol{V}) = d$, otherwise positive-semidefinite. Accordingly, Constraint 3 follows.

A.3 Derivation of formulas for weight-adjustment

In this section, we derive the formulas for the trial weights given in Eqs. (5.10) and (5.12).

A.3.1 Optimal weights for the target trials

From Eq. (5.7) we have

$$\Sigma_1 \boldsymbol{\beta} = \mathbf{1}/(\mathbf{1}^T \Sigma_1^{-1}(\boldsymbol{\theta})\mathbf{1}). \tag{A.32}$$

Consider the loss of one specific target trial of speaker A, $l(A_1, A_2)$, and its covariance with the loss of all other target trials. Let $\Sigma_1^{(A_1,A_2)}$ be the row in Σ_1, containing these covariances. Let the variance of the target trial losses be denoted $v_1(\boldsymbol{\theta})$. The covariances with the losses of the target trials from the other speakers are 0. The covariances with the losses of the other target trials of speaker A are either $v_1(\boldsymbol{\theta})c_a$ or $v_1(\boldsymbol{\theta})c_b$. Let the number of such trials be denoted n_a and n_b respectively. Assume that the weights for all target trials of speaker A are the same, β_A, then

$$\Sigma_1^{(A_1,A_2)} \boldsymbol{\beta} = (1 + n_a c_a + n_b c_b) v_1(\boldsymbol{\theta}) \beta_A. \tag{A.33}$$

Notice that the elements in $\boldsymbol{\beta}$ which are weights for another speaker than speaker A are always multiplied with elements in $\Sigma_1^{(A_1,A_2)}$ that are 0, and therefore they are not present on the right hand side of Eq. (A.33). Setting Eq. (A.33) equal to the corresponding row in Eq. (A.32) gives

$$\begin{aligned}(1 + n_a c_a + n_b c_b)\beta_A &= 1/\left(v_1(\boldsymbol{\theta})\mathbf{1}^T \Sigma_1^{-1}(\boldsymbol{\theta})\mathbf{1}\right) \\ &= 1/(\mathbf{1}^T \boldsymbol{R}_1^{-1}\mathbf{1}) \\ &= k_1,\end{aligned} \tag{A.34}$$

where $\boldsymbol{R}_1 = \Sigma_1(\boldsymbol{\theta})/v_1(\boldsymbol{\theta})$ is the correlation matrix which, according to our assumptions, does not depend on $\boldsymbol{\theta}$. Since we are using all possible target trials, the rows in Σ_1 corresponding to the other target trials of speaker A contains the same elements as $\Sigma_1^{(A_1,A_2)}$ but with a different order. These rows therefore also results in Eq. (A.34). Thus, an equal weight for all target trials of the same speaker gives a solution to Eq. (5.7) and since Σ_t is invertible, it is the only solution.

It remains to find n_a and n_b. There are $N_A(N_A-1)/2-1$ *unique* target trials of speaker A, excluding the trial (A_1, A_2). n_a is the number of trials that include either A_1 or A_2 but not both, in total $n_a = 2(N_A-2)$ trials. n_b is the remaining target trials of speaker A, except (A_1, A_2), in total

$$\begin{aligned} n_b &= N_A(N_A-1)/2 - 2(N_A-2) - 1 \\ &= (N_A-2)(N_A-3)/2. \end{aligned} \tag{A.35}$$

A.3.2 Approximately optimal weights for the non-target trials

Consider the loss of one specific trial, $l(A_1, B_1)$ and its covariance with the loss of other non-target trials. We use a similar approach as for the target trials. However, the non-target trials where one speaker is different from A and B will complicate matters. The number of non-target trials involving the same speakers, A and B, where one utterance is either, A_1 or B_1, is $n_{-a} = N_A + N_B - 2$. The number of non-target trials involving the same speakers but not the utterance A_1 or B_1, are $n_{-b} = (N_A-1)(N_B-1)$. Now, assume that each non-target trial involving speaker A and B has same weight, β_{AB}, and similarly for the other *speaker pairs*. (It can be verified that this gives a solution.) Then from Eq. (5.7), we have

$$\begin{aligned} 1/(1^T \boldsymbol{R}_{(-1)}^{-1}\mathbf{1}) &= (1 + n_{-a}c_{-a} + n_{-b}c_{-b})\beta_{AB} \\ &+ c_{-c} \sum_{X \neq A,B} (\beta_{AX} + \beta_{BX}) N_X \\ &+ c_{-d} \sum_{X \neq A,B} \left((N_A-1)\beta_{AX} + (N_B-1)\beta_{BX}\right) N_X, \end{aligned} \tag{A.36}$$

where $\boldsymbol{R}_{-1} = \Sigma_{-1}/v_{-1}(\boldsymbol{\theta})$ is the correlation matrix and $v_{-1}(\boldsymbol{\theta})$ is the variance for the non-target trial losses. The number of unknown variables in this equation is equal to the number of speaker pairs and we have one such equation per speaker pair. The number of speaker pairs is, however, very large. Instead of solving this system of equations, we use the approximations:

$$\sum_{X \neq A,B} (\beta_{AX} + \beta_{BX}) N_X \approx 2 \sum_{X \neq A,B} \beta_{AB} N_X, \tag{A.37}$$

and

$$\sum_{X\neq A,B}\left((N_A-1)\beta_{AX}+(N_B-1)\beta_{BX}\right)N_X \approx \sum_{X\neq A,B}\left((N_A+N_B-2)\beta_{AB}\right)N_X. \tag{A.38}$$

These approximations are quite reasonable since, e.g., a larger N_A results in a smaller values of both β_{AB} and β_{AX}. This results in

$$(1+n_{-a}c_{-a}+n_{-b}c_{-b}+n_{-c}c_{-c}+n_{-d}c_{-d})\beta_{AB}\approx k_{-1}, \tag{A.39}$$

where,

$$k_{-1}=1/(\mathbf{1}^T\boldsymbol{R}_{-1}^{-1}\mathbf{1}), \tag{A.40}$$

$$n_{-c}=2\sum_{X\neq B,A}N_X, \tag{A.41}$$

and

$$n_{-d}=(N_A+N_B-2)\sum_{X\neq B,A}N_X. \tag{A.42}$$

A.4 Initialization and calculation of gradients for constrained DT

A.4.1 states results given in previous studies. The gradients and initializations for Scr-4par, iV-elmnt and Scr-Def are then given in A.4.2, A.4.3 and A.4.4, respectively. In this section, $\mathbf{1}_{q\times r}$ denotes a matrix of dimension $q\times r$ whose all elements are equal to 1.

A.4.1 Results from previous studies

The results in this subsection are given in Cumani et al. (2011). Let the n training i-vectors be collected in a matrix, $\boldsymbol{\Psi}=[\boldsymbol{\omega}_1\ldots\boldsymbol{\omega}_n]$, and all the scores of the training data be collected in a matrix $\boldsymbol{S}$, i.e., $S_{ij}=s_{ij}$, where s_{ij} is given by Eq. (2.38). Then $\boldsymbol{S}=\boldsymbol{S}_{\mathrm{P}}+\boldsymbol{S}_{\mathrm{Q}}+\boldsymbol{S}_{\mathrm{c}}+\boldsymbol{S}_{\mathrm{k}}$, where

$$\begin{aligned}
\boldsymbol{S}_{\mathrm{P}}&=2\boldsymbol{\Psi}^T\boldsymbol{P}\boldsymbol{\Psi},\\
\boldsymbol{S}_{\mathrm{Q}}&=\mathrm{diag}(\boldsymbol{\Psi}^T\boldsymbol{Q}\boldsymbol{\Psi})\mathbf{1}_{1\times n}+(\mathrm{diag}(\boldsymbol{\Psi}^T\boldsymbol{Q}\boldsymbol{\Psi})\mathbf{1}_{1\times n})^T,\\
\boldsymbol{S}_{\mathrm{c}}&=\boldsymbol{\Psi}^T\boldsymbol{c}\mathbf{1}_{1\times n}+(\boldsymbol{\Psi}^T\boldsymbol{c}\mathbf{1}_{1\times n})^T,\\
\boldsymbol{S}_{\mathrm{k}}&=k\mathbf{1}_{n\times n}.
\end{aligned} \tag{A.43}$$

The gradient of $\hat{L}(\boldsymbol{\gamma})$ in Eq. (3.2) is given by,

$$\nabla\hat{L}(\boldsymbol{\gamma}) = \begin{bmatrix} \nabla_{\boldsymbol{P}}\hat{L}(\boldsymbol{\gamma}) \\ \nabla_{\boldsymbol{Q}}\hat{L}(\boldsymbol{\gamma}) \\ \nabla_{\boldsymbol{c}}\hat{L}(\boldsymbol{\gamma}) \\ \nabla_{k}\hat{L}(\boldsymbol{\gamma}) \end{bmatrix} = \begin{bmatrix} \text{vec}(\boldsymbol{P}') \\ \text{vec}(\boldsymbol{Q}') \\ \boldsymbol{c}' \\ k' \end{bmatrix}, \tag{A.44}$$

where

$$\boldsymbol{P}' = 2\boldsymbol{\Psi}\boldsymbol{G}\boldsymbol{\Psi}^T, \tag{A.45}$$

$$\boldsymbol{Q}' = 2\text{vec}([\boldsymbol{\Psi} \circ (\mathbf{1}_{d\times n}\boldsymbol{G})]\boldsymbol{\Psi}^T), \tag{A.46}$$

$$\boldsymbol{c}' = 2[\boldsymbol{\Psi} \circ (\mathbf{1}_{d\times n}\boldsymbol{G})\boldsymbol{\Psi}]\mathbf{1}_{n\times 1}, \tag{A.47}$$

$$k' = \mathbf{1}_{n\times 1}^T\boldsymbol{G}\mathbf{1}_{n\times 1}, \tag{A.48}$$

$$G_{ij} = \frac{\partial l_{ij}}{\partial s_{ij}}, \tag{A.49}$$

and

$$l_{ij} = l\left(t_{ij}, s_{ij}(\boldsymbol{\gamma}), \tau\right). \tag{A.50}$$

A.4.2 Scr-4Par

The derivative of $\hat{L}$ with respect to a_{P}, is

$$\begin{aligned} \frac{\partial\hat{L}}{\partial a_{\text{P}}} &= \sum_{ij}\frac{\partial l_{ij}}{\partial s_{ij}}\frac{\partial s_{ij}}{\partial a_{\text{P}}} = \sum_{ij} G_{ij}S_{\text{P}ij} \\ &= \mathbf{1}_{n\times 1}^T(\boldsymbol{G} \circ \boldsymbol{S}_{\text{P}})\mathbf{1}_{n\times 1}. \end{aligned} \tag{A.51}$$

The derivatives $\frac{\partial\hat{L}}{\partial a_{\text{Q}}}$, $\frac{\partial\hat{L}}{\partial a_{\text{c}}}$ and $\frac{\partial\hat{L}}{\partial a_{\text{k}}}$ are calculated in the same way. Each of a_{P}, a_{Q}, a_{c} and a_{k}, are initialized to 1.

A.4.3 iV-elmnt

We collect the scalings of the i-vector elements in a diagonal matrix, $\boldsymbol{D}$, so that $\boldsymbol{\omega}$ is replaced by $\boldsymbol{D\omega}$ in Eq. (2.38), i.e.,

$$\begin{aligned} s_{ij} = &\boldsymbol{\omega_i}^T\boldsymbol{DPD\omega_j} + \boldsymbol{\omega_j}^T\boldsymbol{DPD\omega_i} \\ &+\boldsymbol{\omega_i}^T\boldsymbol{DQD\omega_i} + \boldsymbol{\omega_j}^T\boldsymbol{DQD\omega_j} \\ &+(\boldsymbol{\omega_i} + \boldsymbol{\omega_j})^T\boldsymbol{Dc} + k. \end{aligned} \tag{A.52}$$

Let the contribution to the gradient from terms including $\boldsymbol{P}$ be denoted $\nabla^{(P)}_{\text{diag}(\boldsymbol{D})}\hat{L}$ and similarly for $\boldsymbol{Q}$ and $\boldsymbol{c}$. We will use matrix calculus (Minka,

2001) with the convention that the elements of the matrix derivative are laid out according to the transpose of the denominator. The contribution from $\boldsymbol{P}$ to the differential is

$$\mathrm{d}\hat{L} = \mathrm{tr}(\boldsymbol{P}'\mathrm{d}\boldsymbol{P}^T). \tag{A.53}$$

Replacing $\boldsymbol{P}$ with $\boldsymbol{DPD}$ we then get

$$\begin{aligned}\mathrm{d}\hat{L} &= \mathrm{tr}\left(\boldsymbol{P}'\mathrm{d}(\boldsymbol{DPD})^{\boldsymbol{T}}\right) \\ &= \mathrm{tr}\left(\boldsymbol{PDP}'\mathrm{d}\boldsymbol{D} + \boldsymbol{P}'\boldsymbol{DP}\mathrm{d}\boldsymbol{D} + \boldsymbol{DP}'\boldsymbol{D}\mathrm{d}\boldsymbol{P}\right),\end{aligned} \tag{A.54}$$

i.e.,

$$\nabla^{(P)}_{\mathrm{diag}(\boldsymbol{D})}\hat{L} = \mathrm{diag}(\boldsymbol{PDP}' + \boldsymbol{P}'\boldsymbol{DP})\,. \tag{A.55}$$

The contribution from the terms with $\boldsymbol{Q}$ is calculated in the same way. For $\boldsymbol{c}$, we get

$$\begin{aligned}\mathrm{d}\hat{L} &= \boldsymbol{c}'\mathrm{d}(\boldsymbol{Dc})^T \\ &= \boldsymbol{c}'((\mathrm{d}\boldsymbol{c}^T)\boldsymbol{D}^T + \boldsymbol{c}^T\mathrm{d}\boldsymbol{D}^T),\end{aligned} \tag{A.56}$$

i.e.,

$$\nabla^{(c)}_{\mathrm{diag}(\boldsymbol{D})}\hat{L} = \mathrm{diag}(\boldsymbol{c}'\boldsymbol{c}^T) = \boldsymbol{c}' \circ \boldsymbol{c}. \tag{A.57}$$

Finally,

$$\nabla_{\mathrm{diag}(\boldsymbol{D})}\hat{L} = \nabla^{(P)}_{\mathrm{diag}(\boldsymbol{D})}\hat{L} + \nabla^{(Q)}_{\mathrm{diag}(\boldsymbol{D})}\hat{L} + \nabla^{(c)}_{\mathrm{diag}(\boldsymbol{D})}\hat{L}. \tag{A.58}$$

The derivative for k is calculated as in A.4.2. The scalings of the i-vector elements and k are initialized to 1.

A.4.4 Scr-Def

The gradients for $\boldsymbol{c}$ and k in Eq. (A.44) are used without modification. The contribution from $\boldsymbol{P}$ and $\boldsymbol{Q}$ to the differential is

$$\begin{aligned}\mathrm{d}\hat{L} &= \mathrm{tr}\left(\boldsymbol{P}'\mathrm{d}\boldsymbol{P}^T + \boldsymbol{Q}'\mathrm{d}\boldsymbol{Q}^T\right) \\ &= \mathrm{tr}\left(\boldsymbol{P}'\mathrm{d}\boldsymbol{R}_A\boldsymbol{R}_A^T + \boldsymbol{P}'\mathrm{d}\boldsymbol{Q}_A\boldsymbol{Q}_A^T - \boldsymbol{Q}'\mathrm{d}\boldsymbol{Q}_A\boldsymbol{Q}_A^T\right).\end{aligned} \tag{A.59}$$

Using the fact that $\boldsymbol{P}'$ is symmetric, we get for the first term

$$\begin{aligned}\mathrm{tr}\left(\boldsymbol{P}'(\mathrm{d}\boldsymbol{R}_A\boldsymbol{R}_A^T)\right) &= \mathrm{tr}\left(\boldsymbol{P}'\left(\mathrm{d}\boldsymbol{R}_A\right)\boldsymbol{R}_A^T\right) + \mathrm{tr}\left(\boldsymbol{P}'\boldsymbol{R}_A\mathrm{d}\boldsymbol{R}_A^T\right) \\ &= \mathrm{tr}\left(\left(\left(\mathrm{d}\boldsymbol{R}_A\right)\boldsymbol{R}_A^T\right)^T\boldsymbol{P}'^T\right) + \mathrm{tr}\left(\boldsymbol{P}'\boldsymbol{R}_A\mathrm{d}\boldsymbol{R}_A^T\right) \\ &= \mathrm{tr}\left(2\boldsymbol{P}'\boldsymbol{R}_A\mathrm{d}\boldsymbol{R}_A^T\right).\end{aligned} \tag{A.60}$$

The other terms are treated analogously, resulting in

$$\begin{bmatrix} \nabla_{\boldsymbol{R}_A} \hat{L} \\ \nabla_{\boldsymbol{Q}_A} \hat{L} \end{bmatrix} = \begin{bmatrix} 2\text{vec}(\boldsymbol{P}'\boldsymbol{R}_A) \\ 2\text{vec}\left((\boldsymbol{P}' - \boldsymbol{Q}')\boldsymbol{Q}_A\right) \end{bmatrix}. \tag{A.61}$$

The regularization term is dealt with by adding $2(\boldsymbol{P}-\tilde{\boldsymbol{P}})$ to $\boldsymbol{P}'$ and $2(\boldsymbol{Q} - \tilde{\boldsymbol{Q}})$ to $\boldsymbol{Q}'$. For initialization, we use a model estimated by GT and calculate $\boldsymbol{R}_A$ and $\boldsymbol{Q}_A$ by means of eigendecomposition of $\boldsymbol{R}$ and $\boldsymbol{Q}$ respectively, e.g.,

$$\boldsymbol{Q}_A = \boldsymbol{E}(-\boldsymbol{Q})\boldsymbol{D}(-\boldsymbol{Q})^{\frac{1}{2}}, \tag{A.62}$$

where the columns of $\boldsymbol{E}(-\boldsymbol{Q})$ are the eigenvectors of $-\boldsymbol{Q}$ and $\boldsymbol{D}(-\boldsymbol{Q})$ is a diagonal matrix containing the corresponding eigenvalues.

Bibliography

Agnitio. Kivox 360 product data sheet. Website: http://www.agnitio-corp.com/sites/default/files/KIVOX360_DS_51915_FINAL_Hires.pdf, 2015.

R. Auckenthaler, M. Carey, and H. Lloyd-Thomas. Score normalization for text-independent speaker verification systems. *Digital Signal Processing*, 10(1âĂŞ3):42 – 54, 2000. ISSN 1051-2004. doi: http://dx.doi.org/10.1006/dspr.1999.0360. URL `http://www.sciencedirect.com/science/article/pii/S1051200499903603`.

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

S. Biswas, J. Rohdin, and K. Shinoda. Autonomous selection of i-vectors for {PLDA} modelling in speaker verification. *Speech Communication*, 72(0): 32 – 46, 2015. ISSN 0167-6393.

S. Boll. Suppression of acoustic noise in speech using spectral subtraction. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 27(2):113–120, Apr 1979. ISSN 0096-3518. doi: 10.1109/TASSP.1979.1163209.

B. J. Borgström and A. McCree. Discriminatively trained bayesian speaker comparison of i-vectors. In *ICASSP*, pages 7659–7662, 2013.

P.-M. Bousquet, J.-F. Bonastre, and D. Matrouf. Exploring some limits of gaussian plda modeling for i-vector distributions. In *Odyssey*, pages 41–47, 2014.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

N. Brümmer. EM for probabilistic LDA, Feb 2010. URL https://sites.google.com/site/nikobrummer.

N. Brümmer. *Measuring, refining and calibrating speaker and language information extracted from speech*. PhD thesis, Stellenbosch: University of Stellenbosch, 2010.

N. Brümmer and E. de Villiers. The speaker partitioning problem. In *Odyssey*, pages 194–201, 2010.

N. Brümmer and E. de Villiers. The bosaris toolkit user guide: Theory, algorithms and code for binary classifier score processing, Dec 2011. URL https://sites.google.com/site/bosaristoolkit/home/bosaristoolkit_userguide.pdf.

N. Brümmer and G. Doddington. Likelihood-ratio calibration using prior-weighted proper scoring rules. In *INTERSPEECH*, pages 1976–1980, 2013.

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

N. Brümmer, L. Burget, J. Černocký, O. Glembek, F. Grézl, M. Karafiát, D. A. van Leeuwen, P. Matějka, P. Schwarz, and A. Strasheim. Fusion of heterogeneous speaker recognition systems in the STBU submission for the NIST speaker recognition evaluation 2006. *IEEE Transactions on Audio, Speech & Language Processing*, 15(7):2072–2084, 2007.

A. Buja, W. Stuetzle, and Y. Shen. Loss functions for binary class probability estimation and classification: Structure and applications, 2005. URL http://www-stat.wharton.upenn.edu/~buja/PAPERS/paper-proper-scoring.pdf.

C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998. ISSN 1384-5810. doi: 10.1023/A:1009715923555. URL http://dx.doi.org/10.1023/A%3A1009715923555.

L. Burget, N. Brümmer, D. Reynolds, P. Kenny, J. Pelecanos, R. Vogt, F. Castaldo, N. Dehak, R. Dehak, O. Glembek, Z. N. Karam, John Noecker,

Jr., E. Na, C. C. Costin, V. Hubeika, S. Kajarekar, N. Scheffer, and J. Černocký. Robust speaker recognition over varying channels. Technical report, 2008. URL `http://www.fit.vutbr.cz/research/view_pub.php?id=8893`.

L. Burget, O. Plchot, S. Cumani, O. Glembek, P. Matějka, and N. Brümmer. Discriminatively trained probabilistic linear discriminant analysis for speaker verification. In *ICASSP*, pages 4832–4835, 2011.

W.M. Campbell, D.E. Sturim, and D.A. Reynolds. Support vector machines using gmm supervectors for speaker verification. *Signal Processing Letters, IEEE*, 13(5):308–311, May 2006. ISSN 1070-9908. doi: 10.1109/LSP.2006.870086.

S. Cumani and P. Laface. Training pairwise support vector machines with large scale datasets. In *ICASSP*, pages 1664–1668, 2014.

S. Cumani, N. Brümmer, L. Burget, and P. Laface. Fast discriminative speaker verification in the i-vector space. In *ICASSP*, pages 4852–4855, 2011.

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

N. Dehak. *Discriminative and Generative Approaches for Long- and Short-term Speaker Characteristics Modeling: Application to Speaker Verification*. PhD thesis, 2009. AAINR50490.

N. Dehak, R. Dehak, P. Kenny, N. Brümmer, P. Ouellet, and P. Dumouchel. Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification. In *INTERSPEECH*, pages 1559–1562, 2009a.

N. Dehak, P. Kenny, R. Dehak, O. Glembek, P. Dumouchel, L. Burget, V. Hubeika, and F. Castaldo. Support vector machines and joint factor analysis for speaker verification. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2009, 19-24 April 2009, Taipei, Taiwan*, pages 4237–4240, 2009b.

doi: 10.1109/ICASSP.2009.4960564. URL http://dx.doi.org/10.1109/ICASSP.2009.4960564.

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

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38, 1977.

S. Furui. Cepstral analysis technique for automatic speaker verification. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 29(2):254–272, Apr 1981. ISSN 0096-3518. doi: 10.1109/TASSP.1981.1163530.

D. Garcia-Romero and C.Y. Espy-Wilson. Analysis of i-vector length normalization in speaker recognition systems. In *INTERSPEECH*, pages 249–252, 2011.

D. Garcia-Romero and A. McCree. Supervised domain adaptation for i-vector based speaker recognition. In *ICASSP*, pages 4047–4051, May 2014.

J.-L. Gauvain and C.-H. Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing*, 2:291–298, 1994.

O. Glembek. Jfa cookbook, 2008. URL http://speech.fit.vutbr.cz/en/software/joint-factor-analysis-matlab-demo.

O. Glembek, L. Burget, N. Dehak, N. Brummer, and P. Kenny. Comparison of scoring methods used in speaker recognition with joint factor analysis. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 4057–4060, April 2009. doi: 10.1109/ICASSP.2009.4960519.

D. A. Harville. *Matrix Algebra From A Statistician's Perspective*. Springer-Verlag, 1997.

T. Hasan and J. H. L. Hansen. Acoustic factor analysis for robust speaker verification. *Audio, Speech, and Language Processing, IEEE Transactions on*,

21(4):842–853, April 2013. ISSN 1558-7916. doi: 10.1109/TASL.2012.2226161.

H. Hermansky. Perceptual linear predictive (PLP) analysis of speech. *J. Acoust. Soc. Am.*, 57(4):1738–1752, Apr 1990.

S. Ioffe. Probabilistic linear discriminant analysis. In *ECCV (4)*, pages 531–542, 2006.

T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *In Advances in Neural Information Processing Systems 11*, pages 487–493. MIT Press, 1998.

S. M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993. ISBN 0-13-345711-7.

P. Kenny. Joint factor analysis of speaker and session variability: Theory and algorithms, tech. report crim-06/08-13. 2005. URL `http://www.crim.ca/perso/patrick.kenny/`.

P. Kenny. Bayesian speaker verfication with heavy tailed priors. In *Odyssey*, 2010.

P. Kenny, G. Boulianne, and P. Dumouchel. Maximum likelihood estimation of eigenvoices and residual variances for large vocabulary speech recognition tasks. In *7th International Conference on Spoken Language Processing, ICSLP2002 - INTERSPEECH 2002, Denver, Colorado, USA, September 16-20, 2002*, 2002. URL `http://www.isca-speech.org/archive/icslp_2002/i02_0057.html`.

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

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

P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel. A study of interspeaker variability in speaker verification. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(5):980–988, July 2008. ISSN 1558-7916. doi: 10.1109/TASL.2008.925147.

T. Kinnunen and H. Li. An overview of text-independent speaker recognition: From features to supervectors. *Speech Communication*, 52(1):12 – 40, 2010. ISSN 0167-6393. doi: http://dx.doi.org/10.1016/j.specom.2009.08.009. URL `http://www.sciencedirect.com/science/article/pii/S0167639309001289`.

R. Kuhn, P. Nguyen, J.-C. Junqua, L. Goldwasser, N. Niedzielski, S. Fincke, K. Field, and M. Contolini. Eigenvoices for Speaker Adaptation. In *International Conference on Spoken Language Processing*, December 1998. URL `http://www.isca-speech.org/archive/icslp_1998/i98_0303.html`.

C. J. Leggetter and P. C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech & Language*, 9(2):171–185, 1995.

K.-P. Li and J.E. Porter. Normalizations and selection of speech segments for speaker recognition scoring. In *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, pages 595–598 vol.1, Apr 1988. doi: 10.1109/ICASSP.1988.196655.

D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.

M. W. Mak and H. B. Yu. Robust voice activity detection for interview speech in NIST speaker recognition evaluation. In *Proc. APSIPA ASC*, 2010.

P. Matejka, O. Glembek, F. Castaldo, M.J. Alam, O. Plchot, P. Kenny, L. Burget, and J. CÌŇernockyÌĄ. Full-covariance ubm and heavy-tailed PLDA in i-vector speaker verification. In *ICASSP*, pages 4828–4831, May 2011.

T. P. Minka. Old and new matrix algebra useful for statistics. Technical report, 2001. URL `http://research.microsoft.com/en-us/um/people/minka/papers/matrix/minka-matrix.pdf`.

A. Y. Ng and M. I. Jordan. On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes. In T.G. Dietterich,

S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 841–848. MIT Press, 2002.

P. Nguyen, C. Wellekens, and J.-C. Junqua. Maximum likelihood eigenspace and mllr for speech recognition in noisy environments. In *in Noisy EnvironmentsâĂİ, Eurospeech-99, V. 6*, pages 2519–2522, 1999.

T. Nguyen and S. Sanner. Algorithms for direct 0–1 loss optimization in binary classification. In *International Conference on Machine Learning (ICML)*, pages 1–9, Atlanta, USA, June 2013.

NIST. The NIST year 2006 speaker recognition evaluation plan. In *http://www.itl.nist.gov/iad/mig/tests/spk/2006/index.html*, 2006.

NIST. The NIST year 2008 speaker recognition evaluation plan. 2008. URL `http://www.itl.nist.gov/iad/mig/tests/spk/2008/index.html`.

NIST. The NIST year 2010 speaker recognition evaluation plan. 2010. URL `http://www.itl.nist.gov/iad/mig/tests/spk/2010/index.html`.

J. Pelecanos and S. Sridharan. Feature warping for robust speaker verification. In *Odyssey*, pages 213–218, 2001.

K. B. Petersen and M. S. Pedersen. The matrix cookbook, nov 2012. URL `http://www.imm.dtu.dk/pubdb/views/edoc_download.php/3274/pdf/imm3274.pdf`. Version 20121115.

S. J. D. Prince and J. H. Elder. Probabilistic linear discriminant analysis for inferences about identity. In *ICCV*, pages 1–8, 2007.

D. A. Reynolds. Speaker identification and verification using gaussian mixture speaker models. *Speech Commun.*, 17(1-2):91–108, August 1995. ISSN 0167-6393. doi: 10.1016/0167-6393(95)00009-D. URL `http://dx.doi.org/10.1016/0167-6393(95)00009-D`.

D. A. Reynolds. Comparison of background normalization methods for text-independent speaker verification. In *Proc. of 5th European Conf. on Speech Communication and Technology (Eurospeech)*, volume 2, pages 963–966, 1997.

D. A. Reynolds and R. C. Rose. Robust text-independent speaker identification using gaussian mixture speaker models. *Speech and Audio Processing, IEEE Transactions on*, 3(1):72–83, Jan 1995. ISSN 1063-6676. doi: 10.1109/89.365379.

D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10:19–41, 2000.

J. Rohdin, S. Biswas, and K. Shinoda. Constrained discriminative PLDA training for speaker verification. In *ICASSP*, pages 1689–1693, 2014a.

J. Rohdin, S. Biswas, and K. Shinoda. Discriminative PLDA training application-specific loss functions for speaker verification. In *Odyssey*, pages 26–32, 2014b.

J. Rohdin, S. Biswas, and K. Shinoda. Robust discriminative training against data insufficiency in plda-based speaker verification. *Computer Speech & Language*, 35(0):32 – 57, 2016. ISSN 0885-2308. doi: http://dx.doi.org/10.1016/j.csl.2015.06.003. URL `http://www.sciencedirect.com/science/article/pii/S0885230815000625`.

R. C. Rose and D. A. Reynolds. Text independent speaker identification using automatic acoustic segmentation. In *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, pages 293–296 vol.1, Apr 1990. doi: 10.1109/ICASSP.1990.115638.

M. Schmidt. minFunc.m, 2012. URL `http://www.di.ens.fr/~mschmidt/Software/minFunc.html`.

K. Shinoda and C.-H. Lee. A structural bayes approach to speaker adaptation. *Speech and Audio Processing, IEEE Transactions on*, 9(3):276–287, mar 2001.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622, 1999.

V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995. ISBN 0-387-94559-8.

V. Wan and S. Renals. Speaker verification using sequence discriminant support vector machines. *Speech and Audio Processing, IEEE Transactions on*, 13(2):203–210, March 2005. ISSN 1063-6676. doi: 10.1109/TSA.2004.841042.

S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. (A.) Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland. The htk book, Dec 2006. URL `http://htk.eng.cam.ac.uk`.

G. Zavaliagkos, R. Schwartz, and J. Makhoul. Batch, incremental and instantaneous adaptation techniques for speech recognition. In *ICASSP*, volume 1, pages 676–679, may 1995.