Odyssey 2014:
The Speaker and Language Recognition Workshop
16-19 June 2014, Joensuu, Finland

Discriminative PLDA training with application-specific loss functions for speaker verification

Johan Rohdin, Sangeeta Biswas and Koichi Shinoda

Tokyo Institute of Technology, Japan
[johan@ks,sangeeta@ks,shinoda@]cs.titech.ac.jp

10.21437/Odyssey.2014-5

Abstract

Speaker verification systems are usually evaluated by a weighted average of its false acceptance (FA) rate and false rejection (FR) rate. The weights are known as the *operating point* (OP) and depend on the applications. Recent researches suggest that, for the purpose of score calibration of speaker verification systems, it is beneficial to let discriminative training emphasize on the operating points of interest, i.e., use *application-specific* loss functions. In score calibration, a transformation is applied to the scores in order to make them better represent likelihood ratios. The same application-specific training objective can be used in discriminative training of all parameters of a speaker verification system. In this study, we apply application-specific loss functions in discriminative PLDA training. We observe an improvement in the minimun detection cost function (minDCF) for the male trials of the NIST SRE10 telephone for the targeted operating point compared to the baseline, discriminative PLDA training with logistic regression loss.

1. Introduction

When making a decision based on the output from a speaker verification system, we would typically like to minimize the expected cost of the decision. This is reflected in the *detection cost function* (DCF) used as the evaluation metric in the NIST speaker recognition evaluations [1]. DCF assigns one cost to false acceptance (FA) and one cost to false rejection (FR). Together with a prior probability for a trial being a target trial, i.e., the claimed identity is true, the two costs constitutes the DCF evaluation parameters, known as an *operating point* (OP). The optimal OP depends on the application. For example, in forensic applications, the prior probability of a target trial is usually low, whereas in access control it is expected to be high. Moreover, in both forensic and access control applications, the costs of FA and FR may vary depending on outer factors. For example, in an access control application, the cost of FA may depend on resources requested. In such a case it is desirable that the system performs well at a variety of operating points. However, it is rare that an application needs the system to perform well at all possible operating points. Therefore we need *application-specific* speaker verification systems optimized for the OPs on which it will be used.

The current state-of-the-art speaker verification is based on probabilistic linear discriminant analysis (PLDA) [2, 3], with i-vectors [4] as features. For such a system, several studies have shown that discriminative training of the PLDA model parameters is effective [5, 6]. In these studies, the logistic regression (LR) loss function and its approximation, the hinge (SVM) loss functions, were proposed. These loss functions have the benefit of being convex and thus easy to be optimized. However, they are not application-specific.

For the purpose of score calibration and fusion, discriminative training has been well developed [7, 8]. Typically an affine transformation of the log-likelihood ration (llr) score is optimized. Application-dependent score calibration, by means of loss functions that emphasize on a certain range of operating points, has been successfully used in [9]. However, score calibration cannot increase the ability of the system to discriminate between target and non-target trials, i.e., it cannot reduce the minimum detection cost function (minDCF), since it will not change the order of the scores in a set of trials. In order for discriminative training to increase the systems ability to discriminate between trial labels, it must be applied to the earlier stages in the speaker verification process.

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 [9]. 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 not certain that application-specific loss functions are beneficial in discriminative PLDA training.

In this study we focus on application-dependent discriminative training of PLDA-based speaker verification systems with i-vectors as features. In order to obtain a good baseline, we first experimentally compare different regularization and feature normalization options. These experiments show that regularization towards the generative maximum likelihood model and normalizing the i-vectors with their total covariance give the best discrimination but but WCCN and regularization towards $\mathbf{0}$ give a more well-calibrated system. We then compare two application-specific specific loss function, the Brier loss which focuses on a narrow range of OPs compared to the logistic regression loss and the 0-1 loss which targets only one specific OP. We observe small improvements in minDCF and EER for the application-specific loss functions over logistic regression on the male trials of NIST SRE 2010 coreext-coreext condition-5.

The remainder of this paper is organized as follows. Section 2 introduces the DCF. Section 3 introduces the i-vector + PLDA system. Section 4 introduces discriminative PLDA training. Section 5, introduces the loss functions. Section 6 discusses the optimization strategy. Section 7, presents our experiments. Finally, Section 8 concludes the paper.

2. Detection cost function

When making a decision based on the output from a speaker verification system, we would typically like to minimize the expected cost of the decision[1]. The DCF measures the cost for a specific application with a prior probability of a target trial, P_{tar}, and the costs C_{FR} and C_{FA} for FR and FA respectively.

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

where $P_{\text{FR}} = P(\text{error}|\text{tar})$ and $P_{\text{FA}} = P(\text{error}|\text{non-tar})$ are the empirical probability for FR and FA respectively estimated in the evaluation data-base. Since the Bayes decision is not affected by an equal scaling of both costs, we can rewrite it as

$$\text{DCF} = P_{\text{eff}} P_{\text{FR}} + (1 - P_{\text{eff}}) P_{\text{FA}}, \tag{2}$$

where

$$P_{\text{eff}} = \frac{P_{\text{tar}} C_{\text{FR}}}{P_{\text{tar}} C_{\text{FR}} + (1 - P_{\text{tar}}) C_{\text{FA}}}, \tag{3}$$

is known as the *effective prior*. The decision threshold for the log-likelihood ratio (llr) score is given by

$$\tau = -\log \frac{P_{\text{eff}}}{1 - P_{\text{eff}}} \tag{4}$$

$$= -\left(\log \frac{P_{\text{tar}}}{1 - P_{\text{tar}}} + \log \frac{C_{\text{FR}}}{C_{\text{FA}}} \right), \tag{5}$$

assuming that the llr scores are calibrated [8]. When building a speaker verification system, we should optimize it for the minimum expected cost in the application.

3. I-vector and PLDA based Speaker Verification

3.1. i-vector

In the i-vector system [4], it is assumed that a *Gaussian Mixture Model* (GMM) *-supervector*, $\boldsymbol{\mu}$, corresponding to an utterance can be modeled as

$$\boldsymbol{\mu} = \bar{\boldsymbol{\mu}} + \boldsymbol{T}\boldsymbol{\omega}, \tag{6}$$

where $\boldsymbol{\omega}$ is a random vector known as the *i-vector*, $\boldsymbol{T}$ is a basis matrix for the *total variability space*, i.e., for both speaker and channel variability, of $\boldsymbol{\mu}$, and $\bar{\boldsymbol{\mu}}$ is the mean of $\boldsymbol{\mu}$. It is assumed that $\boldsymbol{\omega}$ follows standard normal distribution and that its dimension, d, i.e., the rank of $\boldsymbol{T}$, is lower than the dimension of $\bar{\boldsymbol{\mu}}$. The i-vector is the MAP point estimate of ω.

3.2. PLDA

In [3], it was proposed to use PLDA in speaker verification with i-vectors as features. In that study, a modification of the original PLDA model, suitable for low-dimensional features were suggested. It models i-vectors, $\boldsymbol{\omega}$, as

$$\boldsymbol{\omega} = \boldsymbol{m} + \boldsymbol{V}\boldsymbol{y} + \boldsymbol{D}\boldsymbol{z}, \tag{7}$$

where, $\boldsymbol{m}$ is the mean of the i-vectors, $\boldsymbol{y}$ and $\boldsymbol{z}$ are random vectors depending on speakers and sessions respectively. The speaker variability is given by $\boldsymbol{V}$ and the channel variability is given by $\boldsymbol{D}$. The elements of $\boldsymbol{y}$ and $\boldsymbol{z}$ are assumed to be independent and each follows a standard normal distribution. Usually, $\text{rank}(\boldsymbol{V}) < d$ but $\text{rank}(\boldsymbol{D}) = d$. For the special case when $\text{rank}(\boldsymbol{V}) = d$, the model is referred to as the *two covariance* model [10].

For scoring two i-vectors, $\boldsymbol{\omega_i}$ and $\boldsymbol{\omega_j}$, we need to calculate the llr score of the hypothesis $\mathcal{H}_s$ that the two i-vectors are from the same speaker and the hypothesis $\mathcal{H}_d$ that they are from different speakers, i.e.,

$$s_{ij} = \log \frac{p(\boldsymbol{\omega_i}, \boldsymbol{\omega_j}|\mathcal{H}_s)}{p(\boldsymbol{\omega_i}, \boldsymbol{\omega_j}|\mathcal{H}_d)} = \log \frac{\int P(\boldsymbol{\omega_i}|\boldsymbol{y}) P(\boldsymbol{\omega_j}|\boldsymbol{y}) P(\boldsymbol{y}) d\boldsymbol{y}}{\int\int P(\boldsymbol{\omega_i}|\boldsymbol{y}_1) P(\boldsymbol{\omega_j}|\boldsymbol{y}_2) P(\boldsymbol{y}_1) P(\boldsymbol{y}_2) d\boldsymbol{y}_1 d\boldsymbol{y}_2}, \tag{8}$$

since the speaker factors, $\boldsymbol{y}$, are the same if the two i-vectors are from the same speaker. Eq. (8) has a closed form solution. It is given by:

$$s(\boldsymbol{\omega}_i, \boldsymbol{\omega}_j) = \boldsymbol{\omega}_i^T \boldsymbol{P} \boldsymbol{\omega}_j + \boldsymbol{\omega}_j^T \boldsymbol{P} \boldsymbol{\omega}_i + \boldsymbol{\omega}_i^T \boldsymbol{Q} \boldsymbol{\omega}_i + \boldsymbol{\omega}_j^T \boldsymbol{Q} \boldsymbol{\omega}_j + (\boldsymbol{\omega_i} + \boldsymbol{\omega_j})^T \boldsymbol{c} + k, \tag{9}$$

where

$$\boldsymbol{P} = \frac{1}{2} \Sigma_{\text{tot}}^{-1} \Sigma_{\text{ac}} (\Sigma_{\text{tot}} - \Sigma_{\text{ac}} \Sigma_{\text{tot}}^{-1} \Sigma_{\text{ac}})^{-1}, \tag{10}$$

$$\boldsymbol{Q} = \frac{1}{2} \Sigma_{\text{tot}}^{-1} - (\Sigma_{\text{tot}} - \Sigma_{\text{ac}} \Sigma_{\text{tot}}^{-1} \Sigma_{\text{ac}})^{-1}, \tag{11}$$

$$\boldsymbol{c} = -2(\boldsymbol{P} + \boldsymbol{Q})\boldsymbol{m}, \tag{12}$$

$$k = \frac{1}{2} (\log |\Sigma_{\text{tot}}| - \log |\Sigma_{\text{tot}} - \Sigma_{\text{ac}} \Sigma_{\text{tot}}^{-1} \Sigma_{\text{ac}}|) + \boldsymbol{m}^T 2(\boldsymbol{P} + \boldsymbol{Q})\boldsymbol{m}, \tag{13}$$

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

In [2], the parameters $\boldsymbol{m}$, $\boldsymbol{V}$ and $\boldsymbol{D}$ were trained to maximize likelihood (ML).

$$[\hat{\boldsymbol{m}}, \hat{\boldsymbol{V}}, \hat{\Sigma}] = \arg \max_{\boldsymbol{m}\boldsymbol{V}, \Sigma} \prod_{k=1}^{K} \prod_{l=1}^{L_k} P(\boldsymbol{\omega}_{kl}|\boldsymbol{m}, \boldsymbol{V}, \Sigma), \tag{14}$$

where index k denotes speaker, index l denotes i-vector, K is the number of speakers and L_k is the number of training i-vectors for speaker k. This is typically done by the EM-algorithm as described in [11].

4. Discriminative PLDA Training

4.1. Training procedure

The *discriminative training* method [5, 6] trains the parameters, $\boldsymbol{P}$, $\boldsymbol{Q}$, $\boldsymbol{c}$ and k, of the scoring function in Eq. (9) directly instead of the parameters, $\boldsymbol{m}$, $\boldsymbol{V}$ and $\boldsymbol{D}$ of the PLDA model in Eq. (7). Let t_{ij} equal to 1 if $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_j$ are from the same speaker and -1 if they are from different speakers. Further let $\boldsymbol{\theta} = \text{vec}([\boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{c}, k])$. Then $\boldsymbol{\theta}$ can be trained discriminatively by minimizing the *total loss*:

$$E(\boldsymbol{\theta}) = \sum_{i=1}^{N} \sum_{j=1}^{N} \beta_{ij} l\left(t_{ij}, s_{ij}(\boldsymbol{\theta})\right) + \lambda \|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\|^2, \tag{15}$$

where $l(t, s)$ is a *loss function* for a trial, β_{ij} is a weight, N is the number of i-vectors in the training set, and $\lambda\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\|^2$ is a L2 regularization term. In our experiments, $\bar{\boldsymbol{\theta}}$, is either $\boldsymbol{0}$ or the ML trained parameter vector. The weight β compensates for the fact that P_{eff} in the intended application is typically different for the training database. For this purpose it is set to $P_{\text{eff}}/N_{\text{tar}}$ for the target trials and $(1 - P_{\text{eff}})/N_{\text{non-tar}}$ for the non-target (impostor)

[1] This is however not the only possibility if we consider a set of trials. For example, in some scenarios minimizing the risk for a very high total cost of several trials might be more important than minimizing the expected cost of the set of trial.

trials, where N_{tar} and $N_{\text{non-tar}}$ are the number of target and non-target trials respectively, in the training database.

The gradient of in $E(\boldsymbol{\theta})$ in (15) is given by, ([6]),

$$\begin{aligned}\nabla E(\boldsymbol{\theta}) &= \begin{bmatrix} \nabla_{\boldsymbol{P}} E(\boldsymbol{\theta}) \\ \nabla_{\boldsymbol{Q}} E(\boldsymbol{\theta}) \\ \nabla_{c} E(\boldsymbol{\theta}) \\ \nabla_{k} E(\boldsymbol{\theta}) \end{bmatrix} \\ &= \begin{bmatrix} 2\text{vec}(\boldsymbol{\Omega}\boldsymbol{G}\boldsymbol{\Omega}^T) \\ 2\text{vec}([\boldsymbol{\Omega} \circ (\mathbf{1}_A \boldsymbol{G})]\boldsymbol{\Omega}^T) \\ 2[\boldsymbol{\Omega} \circ (\mathbf{1}_A \boldsymbol{G})\boldsymbol{\Omega}]\mathbf{1}_B \\ \mathbf{1}_B^T \boldsymbol{G} \mathbf{1}_B \end{bmatrix}, \end{aligned} \tag{16}$$

where $\mathbf{1}_A$ is a $d \times n$ matrix of ones and $\mathbf{1}_B$ is a $n \times 1$ matrix of ones, $\circ$ denotes the element wise multiplication of two matrices and

$$G_{ij} = \beta_{ij} \frac{\partial l(t_{ij}, s_{ij})}{\partial s_{i,j}}, \tag{17}$$

An alternative approach for discriminative PLDA training was presented in [12]. In that study, a generalization of the PLDA model was considered in order to deal with multiple enrollment sessions. In the case of only one enrollment session, the method corresponds to standard PLDA. In their discriminative training scheme, only the eigenvalues of the covariance matrices of the PLDA model, or, a scaling factor of them was trained discriminatively while remaining parameters were obtained from ML training. We do not consider that method in this study.

4.2. Regularization and normalization

In order to obtain good performance regularization as well a normalization of the i-vectors are needed. In [13], L2 regularization and *within class covariance normalization* (WCCN) were applied. In order to find a good baseline, we will compare two regularization options and two normalization options. For regularization we compare L2 regularization towards either 0 or the generative ML model. For normalization, we compare WCCN with *total covariance normalization* (TCN).

5. Loss functions

5.1. Logistic regression loss

A modification of the logistic regression (LR) loss function suitable for training log-likelihood ratios was proposed in [7]. It is given by,

$$l_{\text{LR}}(t, s; \tau) = \log(1 + \exp(-t(s + \tau)), \tag{18}$$

As explained in [14] and [8], the logistic regression loss has the property that it focuses on a wide range of OPs. Therefore it was proposed as application-independent evaluation metric in speaker verification in those works. It is also the negative conditional likelihood of the labels in the training set given the i-vectors. We will refer to it as logistic regression (LR) loss in this paper and use let ML refer to the generative maximum likelihood in Eq. (14). The LR loss was used for discriminative PLDA training in [5] and will be our baseline.

5.2. Application dependent loss functions

In [15] and [9] loss functions that emphasize on more narrow range of OPs than the logistic regression were suggested for evaluation and calibration. In this study, we will use one of them, the Brier loss,

$$l_{\text{Brier}}(t, s; \tau) = \frac{1}{(1 + \exp t(s + \tau))^2}, \tag{19}$$

In order to target one specific operating point we should optimize the 0-1 loss,

$$l_{0-1}(t, s; \tau) = \begin{cases} 1 & \text{if } ts < t\tau, \\ 0 & else. \end{cases} \tag{20}$$

In order to use the discriminative training method [5, 6] we need to calculate the derivative in Eq. 17. A standard trick to make the 0-1 loss differentiable is to approximate it with the sigmoid function [16],

$$l_{\sigma}(t, s; \tau) = \frac{1}{1 + \exp(\alpha t(s + \tau))}. \tag{21}$$

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. Since these loss functions are bounded, they are also more robust to outliers than the LR loss. We will target the evaluation parameters of NIST SRE08, $(P_{\text{tar}}, C_{\text{FR}}, C_{\text{FA}}) = (0.01, 10, 1)$, i.e., we will use these parameters for calculating τ and P_{eff} that are used in the loss function. Notice that if no regularization is used, i.e., $\lambda = 0$, then the value of τ will not affect minDCF and EER.

6. Optimization process

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 [16]. 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 [16] in addition to this strategy tried to escape local minima by systematically searching their neighborhood for lower points. We will apply such a strategy in future work.

7. Experiments

We performed four experiments. Different regularization and normalization techniques are compared in Subsection 7.2. In Subsection 7.3, we compare the different loss functions. In Subcection 7.4, we compare the training methods when a portion of the training data used for the calibration. Finally in Subsection 7.5, we investigate, how the choice of the weight, β, in Eq. (15) affects the performance.

7.1. Experimental setup

We used the male trials of NIST SRE 2006 core task (SRE06), as the development set and the male trials of NIST SRE 2008 core condition-6 (SRE08) and NIST SRE 2010 coreext-coreext condition-5 (SRE10) as the evaluation sets. The development set was used to select the regularization parameter, λ, that minimized the detection cost function (DCF).

Voice activity detection using spectral subtraction [17] was

Table 1: Comparison of regularization and normalization techniques on the development set, SRE06. The results are for the optimal regularization parameter λ. EER is in (%).

Training	Norm.	Reg.	actDCF	minDCF	EER
ML	WCCN	-	0.0248	0.0116	2.33
ML	TCN	-	0.0270	0.0117	2.24
LR	WCCN	$\mathbf{0}$	**0.0177**	0.0176	3.49
LR	WCCN	ML	0.0220	**0.0110**	2.37
LR	TCN	$\mathbf{0}$	0.0221	0.0177	3.89
LR	TCN	ML	0.0216	**0.0110**	**2.13**

used for removing non speech. For features we used 15 PLP coefficients and log-energy plus their first-order and second-order derivatives. We applied feature warping [18] before applying VAD. We used gender-dependent systems. For training the UBM and i-vector extractor, we used NIST SRE 2004 and 2005, Switchboard II-Phase 1, 2 and 3, Switchboard Cellular -Part 1 and 2. The dimension of the i-vector, d, was set to 400. For PLDA training we used the same sets except Switchboard II-Phase 1. Also, we excluded in the Switchboard corpora labeled as *noisy* or *cross-talk*. The number of i-vectors in the training data was 9152 corresponding to 61861 target trials (including same segment trials), and 41822267 non-target trials. After the optimal regularization parameter was found, we added NIST SRE06 to the PLDA training data and evaluated SRE08 and SRE10 using the same regularization. Including SRE06, the number of i-vectors in the training data was 11102 corresponding to 72182 target trials (including same segment trials), and 61560571 non-target trials. In order to make the choice of λ robust to the size of the training set, we scale the weights β, with the size of the training data. For WCCN, NIST SRE 2004 and NIST SRE 2005 were used as this was argued to be good in [4]. The i-vectors were length-normalized [19] in all experiments. The rank of $\boldsymbol{V}$ was set to 250.

For optimization, we used the L-BGFS method by [20]. We used its default stopping criteria and in addition we stopped the training if no change in minDCF had been observed on the development set for 20 iterations.

We reported the actual detection cost (actDCF), i.e., using the decision threshold, τ, in Eq. 4, minDCF and equal error rate (EER). Since the training aims to optimize the detection cost of NIST SRE08, we will report this cost at this OP.

7.2. Baseline regularization and normalization results

We first compare the two regularization options and the two normalization options. The result are shown in Table 1. As expected, LR had better calibration than ML. The best minDCF and EER were obtained with TCN and regularization towards the ML model but WCCN and regularization towards $\mathbf{0}$ gave the best actDCF. However, notice that the regularization parameter, λ was chosen to minimize minDCF. In the remaining experiments we used the combination that gave the best discrimination, i.e., TCN with regularization towards ML.

7.3. Training objective results

The results for the different training objectives are summarized in Table 2.

We can see that actDCF is better for all discriminative objectives compared to the generative ML model. However the ML model is competitive in minDCF and EER. The application-specific training objective gives better minDCF and EER than LR for SRE10 but not for SRE08. The fact that the Brier and the approximate 0-1 loss performs very similar suggests that they may have found similar minima.

7.4. Calibration

The discriminative training methods were better in terms of actDCF compared to the ML trained model. However, in terms of the calibration insensitive evaluation metrics, minDCF and EER, the ML trained model is very competitive. The actDCF can be improved with calibration. However, for this we need to use a portion of the training data which could deteriorate the model. Figure 1 shows the actDCF for the various systems when some of the training data is used for calibration. Calibration was done with the Bosaris toolkit [21], by applying an affine transformation to the score, estimated with the CLLR loss with the same value of P_{eff} as for the discriminative training. We used the same regularization parameter λ as in the previous experiment. SRE06 was not included in the training data for this experiment.

Three things are noticeable. First, ML training with calibration was better than discriminative training without calibration. Second, the discriminative training objectives are also benefited from calibration. Third, using 75-90% of the data for PLDA training and the rest for calibration is the optimal, which seems to be quite a lot considering that the PLDA model models is much more complex than the calibration model.

Without regularization, the discriminative training objectives used in this study should not need calibration. Regularization seems therefore to destroy the nice properties of these objective functions to a quite large extent.

7.5. Effect of the weight in the training objective

The choice of $P_{\text{eff}}/N_{\text{tar}}$ for the target trials and $(1-P_{\text{eff}})/N_{\text{non-tar}}$ for the non-target trials is optimal assuming that the trials in the training data are statistically independent and that the evaluation data is similar to the training data. However, this assumption does not hold since every speech segment was used in many trials. Thus the optimal β might be a different one. In order to investigate this we substitute P_{eff} with $P'_{\text{eff}} = \gamma P_{\text{eff}}/(\gamma P_{\text{eff}} + (1-\gamma)(1-P_{\text{eff}}))$, and vary γ between 0 and 1. 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 used the Brier loss in this experiment. SRE06 was not included in the training data for this experiment. The results shown in Figure 2.

The choice of β seemed to be important for actDCF but less important for minDCF. It is very noticeable that minDCF for SRE10 was slightly lower for $\gamma = 1$ (no weight for non-target trials) or $\gamma = 0$ (no weight for target trials) than for other values of γ. The reason why this can give a good result could be that we are doing regularization towards the ML model and the discriminative training with γ equal to 0 or 1 may have stopped quickly, i.e., without deviating much from the ML model. For this reason we show the results for training with regularization towards $\mathbf{0}$. In this case, $\gamma = 0$ or $\gamma = 1$ should not work well. This was confirmed in Figure 2.

8. Conclusion and future work

We have evaluated application-specific loss function in discriminative PLDA training. We observed a reduction of minDCF with around 8% on the male trials of NIST SRE10 coreext-

Table 2: The results obtained by the different training objectives.

	SRE10			SRE08		
Training	actDCF	minDCF	EER (%)	actDCF	minDCF	EER (%)
ML	0.0272	**0.0089**	1.80	0.0323	0.0207	**4.17**
Log. Reg.	**0.0221**	0.0098	1.91	**0.0282**	**0.0199**	4.26
Appr. 0-1	0.0244	0.0091	**1.78**	0.0291	0.0207	4.24
Brier	0.0260	0.0090	1.80	0.0296	0.0204	4.42

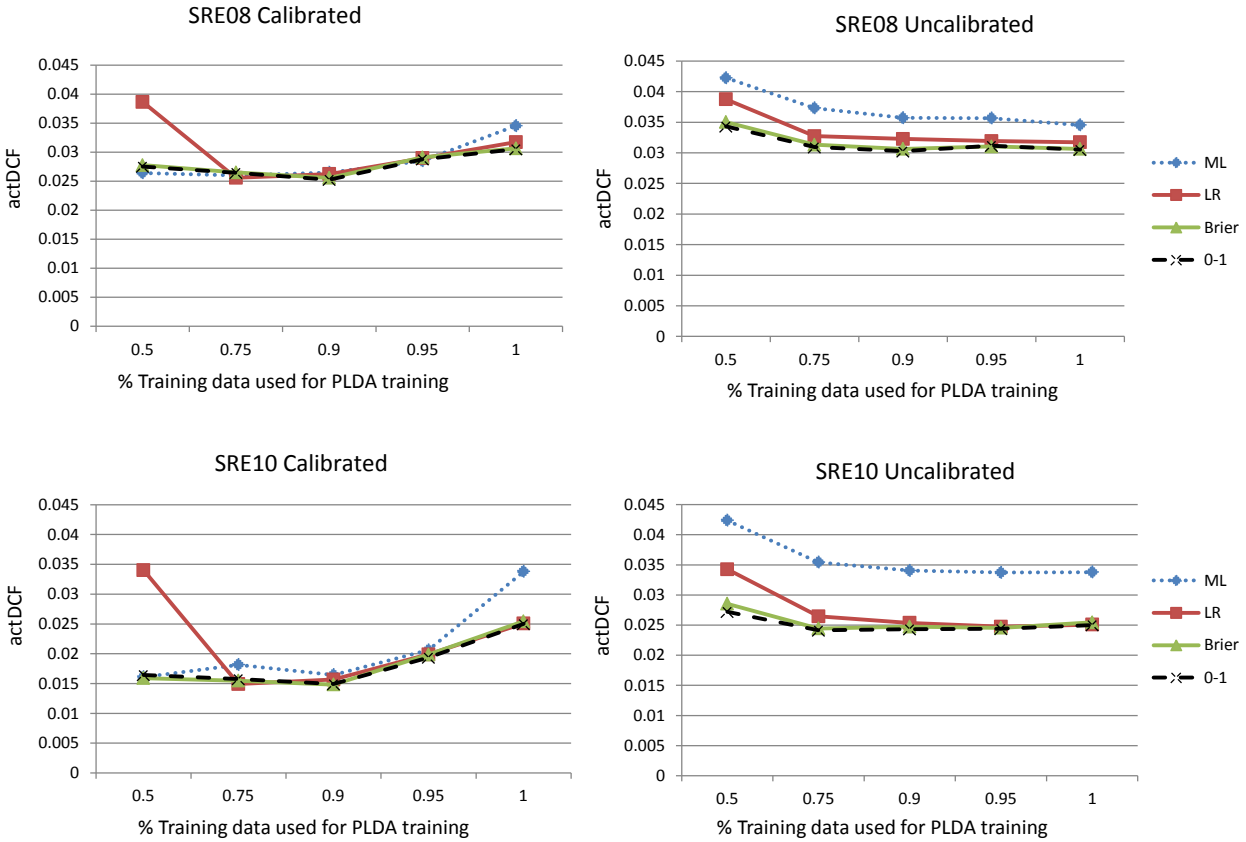

Figure 1: The effect of calibration using a portion of the training data. The x-axis indicates the percentage of the training data used for training the PLDA model. For the calibrated results, the remaining training data was used for calibration.

coreext condition-5 compared to logistic regression loss. However, on NIST SRE08 core condition-6, we did not see any improvement. 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, if focusing on a broad range of OPs is effective as regularization.

The regularization as well as the use of TCN was chosen because it minimize DCF. However, WCCN and regularization towards $\mathbf{0}$ gave better actDCF. It would be worth investigating if this configuration gives better actDCF also when calibration is applied.

9. Acknowledgment

This paper was greatly improved by the comments of the reviewers. In particular, the experimental protocol for calibration as well as to evaluate of the effect of changing β were suggested by the reviewers.

10. References

[1] NIST, "NIST speaker recognition evaluation plans," Website: `http://www.itl.nist.gov/iad/mig//tests/sre/`.

[2] S. J. D. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *ICCV*, pp. 1–8, 2007.

[3] P. Kenny, "Bayesian speaker verfication with heavy tailed priors," in *Odyssey*. ISCA, 2010.

[4] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and

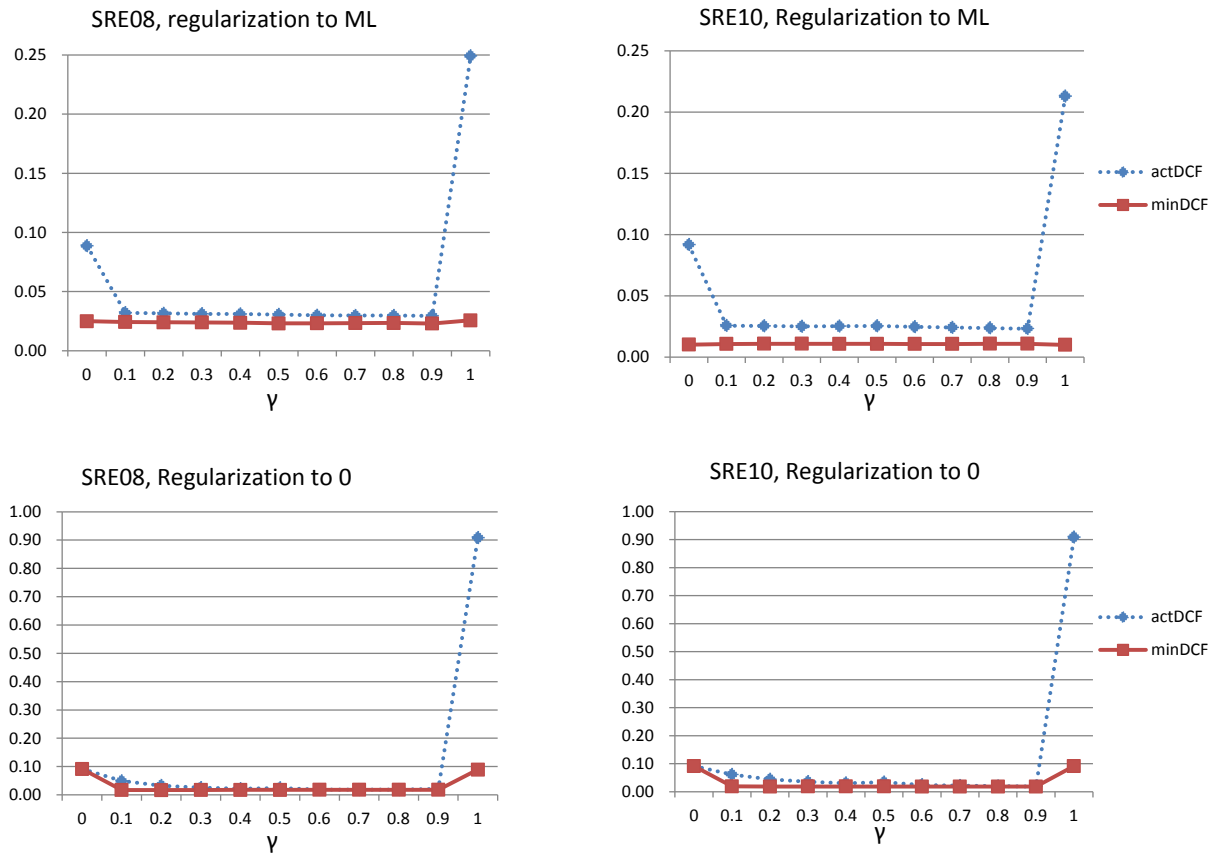

Figure 2: The effect of changing β. In the calculation of β we substitute P_{eff} with P'_{eff}. For $\gamma = 0.5$, $P'_{\text{eff}} = P_{\text{eff}}$, for $\gamma = 1$, $P'_{\text{eff}} = 1$ and for $\gamma = 0$, $P'_{\text{eff}} = 0$.

P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788 –798, 2011.

[5] L. Burget, O. Plchot, S. Cumani, O. Glembek P., Matejka N., and Brümmer, "Discriminatively trained probabilistic linear discriminant analysis for speaker verification," in *Proc. ICASSP*, pp. 4832–4835, 2011.

[6] S. Cumani, N. Brümmer, L. Burget, and P. Laface, "Fast discriminative speaker verification in the i-vector space," in *Proc. ICASSP*, pp. 4852–4855, 2011.

[7] N. Brümmer, L. Burget, J. Cernocký, O. Glembek, F. Grézl, M. Karafiát, D. A. van Leeuwen, P. Matejka, 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*, vol. 15, no. 7, pp. 2072–2084, 2007.

[8] N. Brümmer, *Measuring, refining and calibrating speaker and language information extracted from speech*, Ph.D. thesis, University of Stellenbosch, Dec. 2010.

[9] N. Brümmer and G. R. Doddington, "Likelihood-ratio calibration using prior-weighted proper scoring rules," in *INTERSPEECH*, pp. 1976–1980, 2013.

[10] N. Brümmer and E. de Villiers, "The speaker partitioning problem," in *Odessey*. pp. 194–201, ISCA, 2010.

[11] N. Brümmer, "Em for probabilistic lda," `https://sites.google.com/site/nikobrummer`, Feb 2010.

[12] B. J. Borgström and A. McCree, "Discriminatively trained bayesian speaker comparison of i-vectors," in *Proc. ICASSP*, 2013.

[13] 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*, vol. 21, no. 6, pp. 1217–1227, 2013.

[14] N. Brümmer and J. A. du Preez, "Application-independent evaluation of speaker detection," *Computer Speech & Language*, vol. 20, no. 2-3, pp. 230–275, 2006.

[15] G. R. Doddington, "The role of score calibration in speaker recognition.," in *INTERSPEECH*. ISCA, 2012.

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

[17] M. W. Mak and H. B. Yu, "Robust voice activity detection for interview speech in nist speaker recognition evaluation.," in *Proc. APSIPA ASC*, 2010.

[18] J. Pelecanos and S. Sridharan, "Feature warping for ro-

bust speaker verification," in *Proc. Odyssey*, pp. 213–218, 2001.

[19] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *INTERSPEECH*, pp. 249–252, 2011.

[20] "minFunc.m," Website: `http://www.di.ens.fr/~mschmidt/Software/minFunc.html`.

[21] "Bosaris toolkit," Website: `https://sites.google.com/site/bosaristoolkit/`.