DISCRIMINATIVE PLDA FOR SPEAKER VERIFICATION WITH X-VECTORS

Bengt J. Borgström

MIT Lincoln Laboratory, Lexington, MA

ABSTRACT

This paper proposes a novel approach to discriminative training of probabilistic linear discriminant analysis (PLDA) for speaker verification with x-vectors. The Newton Method is used to discriminatively train the PLDA model by minimizing the log loss of verification trials. By diagonalizing the across-class and within-class covariance matrices as a pre-processing step, the PLDA model can be trained without relying on approximations, and while maintaining important properties of the underlying covariance matrices. The training procedure is extended to allow for efficient domain adaptation. When applied to the Speakers in the Wild and SRE16 tasks, the proposed approach provides significant performance improvements relative to conventional PLDA.

Index Terms— Speaker Verification, Probabilistic Linear Discriminant Analysis, Discriminative Training, X-vectors

1. INTRODUCTION

Probabilistic linear discriminant analysis (PLDA) is a likelihood ratio test between same-class and different-class hypotheses in a verification task, and has become the standard practice for state-of-the-art speaker verification [1, 2]. By separately modeling across-class and within-class variability, PLDA emphasizes important speaker-specific information while de-emphasizing confusable information such as the acoustic channel. For many years, PLDA scoring was successfully used in combination with i-vectors [3]. Recently, however, x-vectors have been proposed as an alternative form of speaker embedding, and have shown impressive performance particularly in difficult acoustic channels [4].

Typically PLDA is trained as a generative model, using e.g. the expectation-maximization (EM) algorithm [5]. However, PLDA can alternatively be trained to directly optimize a cost function which may be more relevant to the desired application. Several studies have explored discriminative training of PLDA (D-PLDA) for speaker verification [2, 6, 7, 8, 9, 10]. Some of these approaches reformulate PLDA scoring as logistic regression with a non-linear basis function whose form is derived from the PLDA log-likelihood ratio (LLR). While showing promise [2, 7, 8], such techniques discard the two-covariance structure of PLDA, and instead optimize intermediate parameters. It may therefore be difficult to guarantee important properties of the underlying covariance matrices, such as being symmetric and non-singular. Other approaches to D-PLDA have reduced the number of trainable parameters, making optimization less prone to over-fitting [6, 9, 10]. Such techniques, however, may limit the potential effectiveness of discriminative training.

This work is sponsored by the Department of Defense under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

In this paper, we propose a novel approach to discriminative training of PLDA. By first diagonalizing the across-class and within-class covariance matrices, the method is able to directly optimize PLDA parameters using the Newton Method without relying on approximations. In this way, important properties of the underlying PLDA covariance matrices are easily guaranteed. Additionally, the approach allows for statistically meaningful regularization and efficient domain adaptation. The proposed technique achieves significant performance improvements when used in combination with recently proposed x-vectors [4], when applied to the Speakers in the Wild (SITW) [11] and SRE16 [12] tasks.

This paper is organized as follows. The statistical framework of PLDA is reviewed in Sec. 2. Sec. 3 discusses the proposed D-PLDA model, and Sec. 4 describes the associated discriminative training method. Experimental results are provided in Sec. 5, and Sec. 6 includes conclusions and future work.

2. PROBABILISTIC LINEAR DISCRIMINANT ANALYSIS

In this section, the statistical framework for PLDA is reviewed in the context of speaker verification with x-vectors. The additive model is assumed

$$\mathbf{x} = \mathbf{s} + \mathbf{c}, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^D$ is the observed x-vector, and $\mathbf{s}$ and $\mathbf{c}$ are the underlying speaker and channel components. Speaker and channel components are drawn from Gaussian distributions:

$$p(\mathbf{s}) = \mathcal{N}(\mathbf{s}; \boldsymbol{\mu}, \boldsymbol{\Sigma}_a), \tag{2}$$

$$p(\mathbf{x} \mid \mathbf{s}) = \mathcal{N}(\mathbf{x}; \mathbf{s}, \boldsymbol{\Sigma}_w). \tag{3}$$

Given two x-vectors, $\mathbf{x}_1$ and $\mathbf{x}_2$, PLDA provides the log-likelihood ratio (LLR) between the same-speaker and different-speaker hypotheses, $\mathcal{H}_1$ and $\mathcal{H}_0$. The PLDA LLR is given by:

$$\begin{aligned}\mathcal{L}(\mathbf{x}_1, \mathbf{x}_2) &= \log \frac{p(\mathbf{x}_1, \mathbf{x}_2 \mid \mathcal{H}_1)}{p(\mathbf{x}_1, \mathbf{x}_2 \mid \mathcal{H}_0)} \\ &= \log \frac{\int \mathcal{N}(\mathbf{x}_1; \mathbf{s}, \boldsymbol{\Sigma}_w)\, \mathcal{N}(\mathbf{x}_2; \mathbf{s}, \boldsymbol{\Sigma}_w)\, \mathcal{N}(\mathbf{s}; \boldsymbol{\mu}, \boldsymbol{\Sigma}_a)\, d\mathbf{s}}{\mathcal{N}(\mathbf{x}_1; \boldsymbol{\mu}, \boldsymbol{\Sigma}_w + \boldsymbol{\Sigma}_a)\, \mathcal{N}(\mathbf{x}_2; \boldsymbol{\mu}, \boldsymbol{\Sigma}_w + \boldsymbol{\Sigma}_a)},\end{aligned} \tag{4}$$

where the PLDA model is defined by the hyperparameters $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}_a$, and $\boldsymbol{\Sigma}_w$. Given the LLR, the posterior probability of the same-speaker hypothesis is expressed as

$$P(\mathcal{H}_1 \mid \mathbf{x}_1, \mathbf{x}_2) = (1 + \exp(-\theta - \mathcal{L}(\mathbf{x}_1, \mathbf{x}_2)))^{-1}, \tag{5}$$

where θ is the log odds ratio of priors, θ=$\log P(\mathcal{H}_1) - \log P(\mathcal{H}_0)$. It was shown in [1] that the solution from (4) can be expressed equivalently as:

$$\begin{aligned}\mathcal{L}(\mathbf{x}_1, \mathbf{x}_2) = &-\frac{1}{2}\log f + \frac{1}{2}(\mathbf{x}_1 - \boldsymbol{\mu})^T \mathbf{Q}(\mathbf{x}_1 - \boldsymbol{\mu}) \\ &+ \frac{1}{2}(\mathbf{x}_2 - \boldsymbol{\mu})^T \mathbf{Q}(\mathbf{x}_2 - \boldsymbol{\mu}) + (\mathbf{x}_1 - \boldsymbol{\mu})^T \mathbf{P}(\mathbf{x}_2 - \boldsymbol{\mu}),\end{aligned} \tag{6}$$

where:
$$f = \frac{|\boldsymbol{\Sigma}_w|\,|\boldsymbol{\Sigma}_w + 2\boldsymbol{\Sigma}_a|}{|\boldsymbol{\Sigma}_w + \boldsymbol{\Sigma}_a|^2}, \tag{7}$$
$$\mathbf{Q} = (\boldsymbol{\Sigma}_w + \boldsymbol{\Sigma}_a)^{-1} - \left(\boldsymbol{\Sigma}_w + \boldsymbol{\Sigma}_a - \boldsymbol{\Sigma}_a\left(\boldsymbol{\Sigma}_w + \boldsymbol{\Sigma}_a\right)^{-1}\boldsymbol{\Sigma}_a\right)^{-1},$$
$$\mathbf{P} = (\boldsymbol{\Sigma}_w + \boldsymbol{\Sigma}_a)^{-1}\boldsymbol{\Sigma}_a\left(\boldsymbol{\Sigma}_w + \boldsymbol{\Sigma}_a - \boldsymbol{\Sigma}_a\left(\boldsymbol{\Sigma}_w + \boldsymbol{\Sigma}_a\right)^{-1}\boldsymbol{\Sigma}_a\right)^{-1}.$$

The set $\{\boldsymbol{\mu}, \boldsymbol{\Sigma}_a, \boldsymbol{\Sigma}_w\}$ parameterizes PLDA, and can be trained as a generative model using the expectation-maximization (EM) algorithm [5].

3. DISCRIMINATIVE PROBABILISTIC LINEAR DISCRIMINANT ANALYSIS

Discriminative training typically relies on gradient descent methods. Therefore, discriminative training of the PLDA model discussed in Sec. 2 requires differentiation of (6) with respect to the set $\{\boldsymbol{\mu}, \boldsymbol{\Sigma}_a, \boldsymbol{\Sigma}_w\}$, which generally is intractable. We propose to preprocess x-vectors so that the matrices $\boldsymbol{\Sigma}_a$ and $\boldsymbol{\Sigma}_w$ are diagonalized, yielding a simplified form of (6), and allowing for direct differentiation. Note that similar ideas were proposed in [9] and [13].

3.1. Simplifying the PLDA log-likelihood ratio

The expression in (6) can be simplified if the x-vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ are first transformed so that their within-class and across-class covariances are jointly diagonalized. Assume there exists a matrix $\mathbf{U}$ which diagonalizes both $\boldsymbol{\Sigma}_w$ and $\boldsymbol{\Sigma}_a$, so that $\mathbf{U}^T\boldsymbol{\Sigma}_a\mathbf{U}$=$\mathbf{A}$ and $\mathbf{U}^T\boldsymbol{\Sigma}_w\mathbf{U}$=$\mathbf{W}$, where $\mathbf{W}$=diag$\{\mathbf{w}\}$ and $\mathbf{A}$=diag$\{\mathbf{a}\}$ (see Sec. 3.2 for a derivation of $\mathbf{U}$). By pre-processing x-vectors according to

$$\mathbf{y}_i = \mathbf{U}^T\left(\mathbf{x}_i - \boldsymbol{\mu}\right), \tag{8}$$

the matrices $\mathbf{Q}$ and $\mathbf{P}$ are diagonalized with diagonal vectors $\mathbf{q}$ and $\mathbf{p}$, respectively. The PLDA solution from (6) then reduces to:

$$\mathcal{L}\left(\mathbf{x}_1, \mathbf{x}_2\right) = \tag{9}$$
$$-\frac{1}{2}\log f + \frac{1}{2}\sum_{d=1}^{D}\left(q_d\mathbf{y}_1^2\left(d\right) + q_d\mathbf{y}_2^2\left(d\right) + 2p_d\mathbf{y}_1\left(d\right)\mathbf{y}_2\left(d\right)\right),$$

where:
$$f = \prod_{d=1}^{D}\frac{w_d\left(w_d + 2a_d\right)}{\left(w_d + a_d\right)^2}, \tag{10}$$
$$q_d = \frac{-a_d^2}{w_d\left(w_d + a_d\right)\left(w_d + 2a_d\right)},$$
$$p_d = \frac{a_d}{w_d\left(w_d + 2a_d\right)},$$

and where subscripts are used to index elements within vectors. In this way, the LLR is expressed solely in terms of scalar operations.

3.2. Deriving the U matrix

The matrix $\mathbf{U}$ introduced in the previous section can be derived according to the following steps, as in [13]:

1. Perform an eigendecomposition of $\boldsymbol{\Sigma}_w$, so that $\boldsymbol{\Sigma}_w$=$\mathbf{E}_1\mathbf{V}_1\mathbf{E}_1^T$
2. Define the matrix $\mathbf{H}$=$\mathbf{E}_1\mathbf{V}_1^{-1/2}$, so that $\mathbf{H}^T\boldsymbol{\Sigma}_w\mathbf{H}$=$\mathbf{I}$
3. Perform an eigendecomposition of $\mathbf{H}^T\boldsymbol{\Sigma}_a\mathbf{H}$, so that $\mathbf{H}^T\boldsymbol{\Sigma}_a\mathbf{H}$=$\mathbf{E}_2\mathbf{V}_2\mathbf{E}_2^T$
4. Define $\mathbf{U}$=$\mathbf{H}\mathbf{E}_2$=$\mathbf{E}_1\mathbf{V}_1^{-1/2}\mathbf{E}_2$, so that $\mathbf{U}^T\boldsymbol{\Sigma}_w\mathbf{U}$=$\mathbf{I}$ and $\mathbf{U}^T\boldsymbol{\Sigma}_a\mathbf{U}$=$\mathbf{V}_2$

In this way, the matrix $\mathbf{U}$ jointly diagonalizes $\boldsymbol{\Sigma}_w$ and $\boldsymbol{\Sigma}_a$, and the simplified expression for the PLDA LLR in (9) is obtained.

3.3. Training

From Sec. 3.1 it can be observed that the D-PLDA model is parameterized by the set $\{\mathbf{U}, \mathbf{a}, \mathbf{w}, \boldsymbol{\mu}\}$ (in fact it is overparameterized, and could be fully parameterized by a subset, e.g. $\{\mathbf{U}, \mathbf{a}, \boldsymbol{\mu}\}$). It is interesting to note that the original PLDA statistical framework can be extracted from the D-PLDA hyperparameters using:

$$\boldsymbol{\Sigma}_w = \mathbf{U}^{-T}\mathbf{W}\mathbf{U}^{-1}, \tag{11}$$
$$\boldsymbol{\Sigma}_a = \mathbf{U}^{-T}\mathbf{A}\mathbf{U}^{-1},$$

which is not possible in other approaches to D-PLDA such as [2, 7, 8]. In this way, important properties of the PLDA covariance matrices can be guaranteed. For example, it is clear from (11) that $\boldsymbol{\Sigma}_a$ and $\boldsymbol{\Sigma}_w$ are symmetric. Furthermore, $\boldsymbol{\Sigma}_w$ and $\boldsymbol{\Sigma}_a$ are guaranteed to be non-singular if $\mathbf{w}$ and $\mathbf{a}$ are constrained to be positive. Alternatively, $\mathbf{a}$ can be constrained to be non-negative, which is consistent with Simplified PLDA [14].

In Sec. 4, optimization of the D-PLDA model using the Newton Method will be discussed. Due to the simple form of (10), the LLR can be differentiated with respect to the set $\{\mathbf{U}, \mathbf{a}, \mathbf{w}, \boldsymbol{\mu}\}$, and the D-PLDA model can be trained without relying on approximations. In our experimentation, however, we found optimization of $\mathbf{U}$ to be prone to over-fitting, and optimization of $\boldsymbol{\mu}$ to offer little benefit. Therefore, in this paper, we focus on training the set $\{\mathbf{a}, \mathbf{w}\}$.

3.4. Relationship to Linear Discriminant Analysis (LDA)

It is interesting to note the similarities between the D-PLDA model presented in Sec. 3.1 and linear discriminant analysis. LDA is commonly used in speaker verification to reduce dimensionality of speaker embeddings prior to scoring. The LDA projection is trained by solving the eigendecomposition of $\boldsymbol{\Sigma}_w^{-1}\boldsymbol{\Sigma}_a$, and discarding the dimensions corresponding to the smallest eigenvalues. By referencing the steps outlined in Sec. 3.2, it can be shown that $\boldsymbol{\Sigma}_w^{-1}\boldsymbol{\Sigma}_a\mathbf{U}$=$\mathbf{U}\mathbf{A}$, implying that the matrix $\mathbf{U}$ is a valid LDA transform with eigenvalues diag$\{\mathbf{A}\}$. D-PLDA training that leads to a_d=0 for any d corresponds to discarding the associated LDA basis. Therefore, the proposed D-PLDA approach can be interpreted as, in part, optimizing the basis selection during LDA dimension reduction.

4. D-PLDA TRAINING

4.1. The Generalized Cost Function

To optimize the D-PLDA model, we can minimize some discriminative cost function over a training set of x-vectors. The general form of the discriminative cost function is given by:

$$C = \sum_{m=0}^{1}\sum_{(i,j)\in\mathcal{H}_m} l_m\left(\mathcal{L}\left(\mathbf{x}_i, \mathbf{x}_j\right)\right), \tag{12}$$

where $l_0\left(\mathcal{L}\left(\mathbf{x}_i, \mathbf{x}_j\right)\right)$ and $l_1\left(\mathcal{L}\left(\mathbf{x}_i, \mathbf{x}_j\right)\right)$ represent the losses associated with $\mathcal{L}\left(\mathbf{x}_i, \mathbf{x}_j\right)$ for labels 0 and 1, respectively. Additionally, the notation (i, j) denotes a verification trial with inputs $\mathbf{x}_i$ and $\mathbf{x}_j$. The expression in (12) can be modified to balance the effect of trials from the two hypotheses via appropriate normalization, but we omit this for the sake of clarity. We propose to train the D-PLDA model using the Newton Method, which for a_d is given by:

$$a_d \Leftarrow a_d - \gamma\left(\frac{\partial^2 C}{\partial a_d^2} + \lambda\right)^{-1}\frac{\partial C}{\partial a_d}, \tag{13}$$

with an analogous expression for w_d, and where λ is a regularization constant and γ is the step-size.

The update rule in (13) requires the first and second derivatives of the cost function, C, which can be expressed with respect to e.g. the parameter a_d using the chain rule:

$$\frac{\partial C}{\partial a_d} = \sum_{m=0}^{1} \sum_{(i,j)\in\mathcal{H}_m} \frac{\partial l_m\left(\mathcal{L}\left(\mathbf{x}_i,\mathbf{x}_j\right)\right)}{\partial \mathcal{L}\left(\mathbf{x}_i,\mathbf{x}_j\right)} \frac{\partial \mathcal{L}\left(\mathbf{x}_i,\mathbf{x}_j\right)}{\partial a_d}, \tag{14}$$

$$\frac{\partial^2 C}{\partial a_d^2} = \sum_{m=0}^{1} \sum_{(i,j)\in\mathcal{H}_m} \left(\frac{\partial^2 l_m\left(\mathcal{L}\left(\mathbf{x}_i,\mathbf{x}_j\right)\right)}{\partial \mathcal{L}\left(\mathbf{x}_i,\mathbf{x}_j\right)^2} \left(\frac{\partial \mathcal{L}\left(\mathbf{x}_i,\mathbf{x}_j\right)}{\partial a_d}\right)^2 + \frac{\partial l_m\left(\mathcal{L}\left(\mathbf{x}_i,\mathbf{x}_j\right)\right)}{\partial \mathcal{L}\left(\mathbf{x}_i,\mathbf{x}_j\right)} \frac{\partial^2 \mathcal{L}\left(\mathbf{x}_i,\mathbf{x}_j\right)}{\partial a_d^2} \right).$$

Evaluation of (14) requires partial derivatives of the loss functions and log-likelihood ratios, which are derived in Sec. 4.2 and Sec. 4.3, respectively. The expressions in (14) can be expressed as functions of statistics accumulated across the training trials. In our implementation, D-PLDA training first accumulates these statistics, and then applies the Newton Method update.

4.2. Differentiating the loss function

There exists a variety of loss functions that can be used in the generalized cost in (12), and the choice of l_m can be made based on the intended application. In this paper the log loss is used, but the proposed framework allows for other functions, such as the Brier or hinge loss [8]. The log loss is defined as:

$$l_m\left(\mathcal{L}\left(\mathbf{x}_i,\mathbf{x}_j\right)\right) = -\log\left(P\left(\mathcal{H}_m \mid \mathbf{x}_i,\mathbf{x}_j\right)\right). \tag{15}$$

Using the Newton Method for training D-PLDA involves evaluation of (14). This requires the first and second derivatives of the log loss, which can be expressed as [15]:

$$\frac{\partial l_m\left(\mathcal{L}\left(\mathbf{x}_i,\mathbf{x}_j\right)\right)}{\partial \mathcal{L}\left(\mathbf{x}_i,\mathbf{x}_j\right)} = P\left(\mathcal{H}_1 \mid \mathbf{x}_i,\mathbf{x}_j\right) - m, \tag{16}$$

and

$$\frac{\partial^2 l_m\left(\mathcal{L}\left(\mathbf{x}_i,\mathbf{x}_j\right)\right)}{\partial \mathcal{L}\left(\mathbf{x}_i,\mathbf{x}_j\right)^2} = P\left(\mathcal{H}_1 \mid \mathbf{x}_i,\mathbf{x}_j\right)\left(1 - P\left(\mathcal{H}_1 \mid \mathbf{x}_i,\mathbf{x}_j\right)\right). \tag{17}$$

4.3. Differentiating the log-likelihood ratio

The derivatives of the LLR with respect to the parameters a_d can be derived from (9) and (10) as

$$\frac{\partial \mathcal{L}\left(\mathbf{x}_i,\mathbf{x}_j\right)}{\partial a_d} = -\frac{1}{2}\left(\frac{1}{f}\frac{\partial f}{\partial a_d} - \frac{\partial q_d}{\partial a_d}\mathbf{y}_i^2\left(d\right) - \frac{\partial q_d}{\partial a_d}\mathbf{y}_j^2\left(d\right) - 2\frac{\partial p_d}{\partial a_d}\mathbf{y}_i\left(d\right)\mathbf{y}_j\left(d\right)\right), \tag{18}$$

and

$$\frac{\partial^2 \mathcal{L}\left(\mathbf{x}_i,\mathbf{x}_j\right)}{\partial a_d^2} = -\frac{1}{2}\left(\frac{1}{f}\frac{\partial^2 f}{\partial a_d^2} - \left(\frac{1}{f}\frac{\partial f}{\partial a_d}\right)^2 - \frac{\partial^2 q_d}{\partial a_d^2}\mathbf{y}_i^2\left(d\right) - \frac{\partial^2 q_d}{\partial a_d^2}\mathbf{y}_j^2\left(d\right) - 2\frac{\partial^2 p_d}{\partial a_d^2}\mathbf{y}_i\left(d\right)\mathbf{y}_j\left(d\right)\right), \tag{19}$$

with analogous expressions for w_d, and where

$$\begin{aligned}
\frac{\partial f}{\partial a_d} &= \frac{-2a_d f}{\left(w_d + a_d\right)\left(w_d + 2a_d\right)}, \\
\frac{\partial q_d}{\partial a_d} &= \frac{-a_d\left(2w_d + 3a_d\right)}{\left(w_d + a_d\right)^2\left(w_d + 2a_d\right)^2}, \\
\frac{\partial p_d}{\partial a_d} &= \frac{1}{\left(w_d + 2a_d\right)^2}, \\
\frac{\partial f}{\partial w_d} &= \frac{2a_d^2 f}{w_d\left(w_d + a_d\right)\left(w_d + 2a_d\right)}, \\
\frac{\partial q_d}{\partial w_d} &= \frac{a_d^2\left(3w_d^2 + 6w_d a_d + 2a_d^2\right)}{w_d^2\left(w_d + a_d\right)^2\left(w_d + 2a_d\right)^2}, \\
\frac{\partial p_d}{\partial w_d} &= \frac{-2a_d\left(w_d + a_d\right)}{w_d^2\left(w_d + 2a_d\right)^2}.
\end{aligned} \tag{20}$$

and

$$\begin{aligned}
\frac{\partial^2 f}{\partial a_d^2} &= \frac{-2\left(w_d - 2a_d\right) f}{\left(w_d + a_d\right)^2\left(w_d + 2a_d\right)}, \\
\frac{\partial^2 q_d}{\partial a_d^2} &= \frac{-2\left(w_d^3 - 6w_d a_d^2 - 6a_d^3\right)}{\left(w_d + a_d\right)^3\left(w_d + 2a_d\right)^3}, \\
\frac{\partial^2 p_d}{\partial a_d^2} &= \frac{-4}{\left(w_d + 2a_d\right)^3}, \\
\frac{\partial^2 f}{\partial w_d^2} &= \frac{-6a_d^2 f}{w_d\left(w_d + a_d\right)^2\left(w_d + 2a_d\right)}, \\
\frac{\partial^2 q_d}{\partial w_d^2} &= \frac{6a_d^2}{w_d^2\left(w_d + a_d\right)\left(w_d + 2a_d\right)^2}, \\
&\quad - \frac{2a_d^2\left(3w_d^2 + 6w_d a_d + 2a_d^2\right)^2}{w_d^3\left(w_d + a_d\right)^3\left(w_d + 2a_d\right)^3} \\
\frac{\partial^2 p_d}{\partial w_d^2} &= \frac{-2a_d\left(w_d + a_d\right)}{w_d^2\left(w_d + 2a_d\right)^2}.
\end{aligned} \tag{21}$$

The terms in (18)-(21) are required by (14) when performing the Newton Method during D-PLDA training.

4.4. Maximum Likelihood Regularization

In order to avoid over-fitting, the D-PLDA training process can include regularization. In the context of learning a D-PLDA model, conventional regularization methods such as the l_2-norm may not be appropriate, since they do not take into account the statistical distribution of x-vectors. Instead, we use maximum likelihood (ML) regularization which offers some probabilistic intuition. The cost function in (12) is updated to include the ML regularization term:

$$\begin{aligned}
C_{ml} &= -\frac{\eta}{N}\sum_{i=1}^{N}\log p\left(\mathbf{y}_i\right) \\
&= -\frac{\eta}{N}\sum_{i=1}^{N}\log \mathcal{N}\left(\mathbf{y}_i; \mathbf{0}, \mathbf{W} + \mathbf{A}\right) \\
&= \frac{\eta}{2}\sum_{d=1}^{D}\left(\log\left(w_d + a_d\right) + \frac{\sigma_y^2\left(d\right)}{w_d + a_d}\right) + \text{const.}
\end{aligned} \tag{22}$$

where $\sigma_y^2\left(d\right)$ is the variance of the d^{th} component of $\mathbf{y}_i$ and η controls the balance between minimizing the discriminative cost and the regularization term. Using C_{ml} with the Newton Method requires

first and second derivatives, which are given by:

$$\frac{\partial C_{ml}}{\partial a_d} = \frac{\eta}{2}\left(\frac{a_d + w_d - \sigma_y^2(d)}{(a_d + w_d)^2}\right), \qquad (23)$$
$$\frac{\partial^2 C_{ml}}{\partial a_d^2} = -\frac{\eta}{2}\left(\frac{a_d + w_d - 2\sigma_y^2(d)}{(a_d + w_d)^3}\right),$$

and which are equivalent when taken with respect to w_d. Using ML regularization during training ensures that the D-PLDA model reflects the true distribution of x-vectors.

4.5. Domain Adaptation

When PLDA-based speaker verification systems are tested on unseen types of data, performance can be expected to degrade. To combat this, domain adaptation can be used to adapt the PLDA model to a typically small set of in-domain data [14, 16, 17]. The proposed D-PLDA framework can be extended to perform domain adaptation by adjusting the cost function. Let C_{ood} denote the cost function from (12) when applied to a rich out-of-domain data set, and let C_{id} denote the cost when applied to an in-domain data set. Domain adaptation can then be performed by utilizing a total cost function of the form C_{total}=$(1-\alpha)\,C_{ood} + \alpha C_{id}$. Here, α controls the balance between the data sets during D-PLDA training.

5. RESULTS

This section presents experimental results for the proposed D-PLDA model. The speaker verification system used x-vectors generated according to [4]. Global whitening and length normalization [1] were performed, followed by an optional LDA dimension reduction. Two baseline systems were used: the first used original 512-dimensional x-vectors, and the second used LDA dimension reduction to 150 dimensions[1].The baseline systems used conventional PLDA for scoring, which was trained using the EM algorithm. The D-PLDA system was applied to 512-dimensional x-vectors, and was initialized using the EM algorithm. All possible combinations of x-vectors in the training set were used when generating D-PLDA training trials. The Newton Method was applied for 3 iterations, and used a step size of γ=0.4. The Newton Method regularization constant was λ=10^{-3} and the ML regularization constant was η=10^{-4}. Results are provided in terms of equal error rate (EER) and the minimum decision cost function (mindcf) with $P(\mathcal{H}_1)$=10^{-2}.

5.1. Speaker Verification on the Speakers in the Wild Core Task

D-PLDA was applied to the Speakers in the Wild (SITW) Core Task, and results are provided in Table 1. The LDA dimension reduction used by the baseline systems is specified in parentheses. In the first set of experiments, the PLDA and D-PLDA training set includes data from the NIST SRE04-SRE10 along with Mixer 6, totaling 63k cuts (denoted 'SRE' in the Table). In the case of D-PLDA, this results in 3.9B training trials. In the second set of experiments, the training set is extended to use data augmentation with noise and reverberation from [18], according to [4], resulting in 151k cuts (denoted 'SRE + aug.' in the table). For D-PLDA, this results in 22.8B training trials. In the last set of experiments, domain adaptation is performed with the development set from SITW, which includes 823 cuts from 119 speakers. For PLDA, Bayesian adaptation was applied [16], and for D-PLDA, domain adaptation was applied according to Sec. 4.5. In either case, the whitening matrix was adapted to the in-domain data.

Since D-PLDA was applied to the original 512 dimensional x-vectors, its effect can be observed in Table 1 by comparing it to the PLDA (512) baseline. D-PLDA yields significant performance improvements in terms of both EER and mindcf for each set of experiments. Specifically, D-PLDA provides 15%-21% and 18%-28% relative improvement in EER and mindcf, respectively, compared to the PLDA (512) system. On the other hand, PLDA (150) can be considered similar to the state-of-the-art system in [4]. Compared to PLDA (150), D-PLDA provides 5%-13% and 8%-18% relative improvement in EER and mindcf, respectively.

Table 1. Speaker Verification Results for the SITW Core Task

Training Data	Adaptation Data	Model	EER (%)	mindcf
SRE	-	PLDA (512)	10.61	0.959
		PLDA (150)	9.10	0.815
		D-PLDA	8.42	0.700
SRE + aug.	-	PLDA (512)	7.71	0.678
		PLDA (150)	6.89	0.606
		D-PLDA	6.31	0.548
SRE + aug.	SITW	PLDA (512)	6.01	0.611
		PLDA (150)	5.90	0.588
		D-PLDA	5.25	0.493

5.2. Speaker Verification on the SRE16 Fixed Task

The same experimental design was applied to the NIST SRE16 Fixed Task, and the results are provided in Table 2. In the case of domain adaptation, the SRE16 *major unlabeled* set was used with speaker labels, totaling 2272 cuts from 1164 speakers. From Table 2, it can again be observed that D-PLDA provides significant performance improvements over the baseline systems in many cases. However, D-PLDA fails to outperform the PLDA (512) baseline in the case of domain adaptation, which may be due to the small number of cuts per speaker in the SRE16 in-domain set.

Table 2. Speaker Verification Results for the SRE16 Fixed Task

Training Data	Adaptation Data	Model	EER (%)	mindcf
SRE	-	PLDA (512)	23.19	1.000
		PLDA (150)	19.30	0.961
		D-PLDA	18.24	0.950
SRE + aug.	-	PLDA (512)	21.82	1.000
		PLDA (150)	18.03	0.953
		D-PLDA	15.85	0.882
SRE + aug.	SRE16	PLDA (512)	8.48	0.613
		PLDA (150)	9.59	0.635
		D-PLDA	9.52	0.615

6. CONCLUSIONS

This paper proposed a novel approach to discriminative PLDA for speaker verification with x-vectors. Pre-processing data to diagonalize within-class and across-class covariance matrices allows the PLDA model to be trained with the Newton Method without relying on approximations. The proposed method provides significant performance improvements on the Speakers in the Wild and SRE16 tasks, relative to using conventional PLDA. Although introduced in the context of x-vectors, the proposed D-PLDA method can likely be applied to many other types of speaker embeddings.

[1]The baseline systems apply length normalization to raw x-vectors. Alternatively, applying LDA prior to length normalization may provide improved performance with x-vectors for certain tasks. Future work includes integration of the proposed D-PLDA method with the latter system configuration.

7. REFERENCES

[1] Daniel Garcia-Romero and Carol Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Interspeech*, 2011, pp. 249–252.

[2] Lucas Burget, Oldrich Pichot, Sandro Cumani, Ondrej Glembek, Patel Matejka, and Niko Brummer, "Discriminatively trained probabilistic linear discriminat analysis analysis for speaker verification," in *ICASSP*, 2011, pp. 4832–4835.

[3] Najim Dehak, Patrick Kenny, Rema Dehak, Pierre Ouellet, and Pierre Dumouchel, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 788–798, 2011.

[4] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," *ICASSP*, 2018.

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

[6] Bengt J. Borgström and Alan McCree, "Discriminatively trained bayesian speaker comparison of i-vectors," in *ICASSP*, 2013, pp. 7659–7662.

[7] Johan Rohdin, Sangeeta Biswas, and Koichi Shinoda, "Constrained discriminative PLDA training for speaker verification," in *ICASSP*, 2014, pp. 1670–1674.

[8] Johan Rohdin, Sangeeta Biswas, and Koichi Shinoda, "Discriminative PLDA training with application-specific loss functions for speaker verification," in *Odyssey*, 2014.

[9] Pierre-Michel Bousquet and Jean-Francois Bonastre, "Constrained discriminative speaker verification specific to normalized i-vectors," in *Odyssey*, 2016, pp. 53–59.

[10] Johan Rohdin, Sangeeta Biswas, and Koichi Shinoda, "Robust discriminative training against data insufficiency in PLDA-based speaker verification," *Computer Speech & Language*, vol. 35, pp. 32–57, 2016.

[11] Mitchell McLaren, Luciana Ferrer, Diego Castan, and Aaron Lawson, "The 2016 speakers in the wild speaker recognition evaluation.," in *INTERSPEECH*, 2016, pp. 823–827.

[12] Seyed Omid Sadjadi, Timothée Kheyrkhah, Audrey Tong, Craig Greenberg, Douglas Reynolds, Elliot Singer, Lisa Mason, and Jaime Hernandez-Cordero, "The 2016 NIST speaker recognition evaluation," 2017.

[13] Alan McCree, "Multiclass discriminative training of i-vector language recognition," in *Odyssey*, 2014, pp. 166–172.

[14] Daniel Garcia-Romero and Alan McCree, "Supervised domain adaptation for i-vector based speaker recognition," in *ICASSP*, 2014, pp. 4047–4051.

[15] Thomas P Minka, "A comparison of numerical optimizers for logistic regression," *Unpublished draft*, pp. 1–18, 2003.

[16] Jesus Villalba and Eduardo Lleida, "Bayesian adaptation of PLDA based speaker recognition to domains with scarce development data," in *Odyssey*, 2012.

[17] Bengt J. Borgstrom, Douglas A. Reynolds, Elliot Singer, and Omid Sadjadi, "Improving the effectiveness of speaker verification domain adaptation with inadequate in-domain data," *Interspeech*, pp. 1557–1561, 2017.

[18] David Snyder, Guoguo Chen, and Daniel Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.