ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10095580

Toroidal Probabilistic Spherical Discriminant Analysis

Anna Silnova[1], *Niko Brümmer*[2], *Albert Swart*[3], *Lukáš Burget*[1]

[1]Brno University of Technology, Speech@FIT and IT4I Center of Excellence, Brno, Czechia
[2]Amazon Alexa, South Africa
[3]Speechly, Finland

isilnova@fit.vutbr.cz, niko.brummer@gmail.com, adswart@gmail.com

Abstract

In speaker recognition, where speech segments are mapped to embeddings on the unit hypersphere, two scoring back-ends are commonly used, namely cosine scoring and PLDA. We have recently proposed PSDA, an analog to PLDA that uses Von Mises-Fisher distributions instead of Gaussians. In this paper, we present toroidal PSDA (T-PSDA). It extends PSDA with the ability to model within and between-speaker variabilities in toroidal submanifolds of the hypersphere. Like PLDA and PSDA, the model allows closed-form scoring and closed-form EM updates for training. On VoxCeleb, we find T-PSDA accuracy on par with cosine scoring, while PLDA accuracy is inferior. On NIST SRE'21 we find that T-PSDA gives large accuracy gains compared to both cosine scoring and PLDA.[1]

Index Terms: speaker recognition, PSDA, Von Mises-Fisher

1. Introduction

Probabilistic *linear* discriminant analysis (PLDA) [1, 2], is a popular back-end for scoring speaker recognition embeddings (e.g., i-vectors [3] or x-vectors [4]) in $\mathbb{R}^D$, following [5, 6]. However, [7] showed that length-normalizing the embeddings onto the unit sphere, $\mathbb{S}^{D-1}$ has a Gaussianizing effect that improves speaker verification performance, and this has been the standard practice ever since. One disadvantage of the length-normalization is that within-speaker variability is squashed in the radial direction, making it *speaker-dependent*, in violation of the PLDA assumption of a constant within-class distribution. Moreover, given a flexible, discriminatively trained embedding extractor, it is often found that cosine scoring (dot products between embeddings on the hypersphere) outperforms PLDA, especially when the test data is in-domain, e.g., [8, 9].

In our previous work [10], we introduced *probabilistic spherical discriminant analysis* (PSDA), where the observed and hidden variables have Von Mises-Fisher (VMF) rather than Gaussian distributions. We found the performance of the PSDA model to be very similar to cosine scoring because it has very limited modeling capacity due to the low amount of trainable parameters. This paper presents an extended version of the PSDA model—*toroidal PSDA* (T-PSDA), where the observed data still live on the unit sphere and have VMF distributions, but now we have added a structured space (defined via a larger set of trainable parameters) where the hidden variables live. An open-source implementation of T-PSDA is available.[2]

2. Theory

2.1. The Von Mises-Fisher distribution

When embeddings in Euclidean space, $\mathbb{R}^D$ are length-normalized, they are projected onto the *unit hypersphere*:

$$\mathbb{S}^{D-1} = \{\mathbf{x} \in \mathbb{R}^D : \|\mathbf{x}\| = 1\}, \tag{1}$$

where we model them with the Von-Mises Fisher (VMF) distribution, which for $\mathbf{x} \in \mathbb{S}^{D-1}$ has the density [11]:

$$\mathcal{V}(\mathbf{x} \mid \boldsymbol{\mu}, \kappa) = \frac{C_\nu(\kappa)}{(2\pi)^{d/2}} e^{\kappa \boldsymbol{\mu}' \mathbf{x}}, \tag{2}$$

where $C_\nu(\kappa) = \frac{\kappa^\nu}{I_\nu(\kappa)}$ and $\nu = \frac{D}{2} - 1$ and $I_\nu(\kappa)$ is the modified Bessel function of the first kind. The parameters are the *mean direction*, $\boldsymbol{\mu} \in \mathbb{S}^{D-1}$ and the *concentration*, $\kappa \geq 0$. In terms of the *natural parameter*, $\mathbf{a} = \kappa\boldsymbol{\mu}$, the VMF density can alternatively be expressed as:

$$\mathcal{V}(\mathbf{x} \mid \mathbf{a}) = \frac{\bar{C}_\nu(\kappa)}{(2\pi^{d/2})} e^{\mathbf{a}'\mathbf{x} - \kappa}, \tag{3}$$

where $\kappa = \|\mathbf{a}\|$ and $\bar{C}_\nu(\kappa) = \frac{e^\kappa \kappa^\nu}{I_\nu(\kappa)}$.

2.2. T-PSDA model definition

T-PSDA is inspired by the original (full) PLDA model that has both a hidden speaker factor and a hidden within-speaker (channel) factor. We replicate this structure with T-PSDA. However, unlike in PLDA, the spherical geometry allows *multiple* speaker factors and *multiple* channel factors. For *each* speaker, let there be a set of $m \geq 1$ hidden *speaker factors*, $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^m$, where $\mathbf{z}_i \in \mathbb{S}^{d_i - 1}$, where all $d_i \geq 1$.[3] We represent $\mathbf{z}_i \in \mathbb{R}^{d_i}$, subject to $\mathbf{z}_i'\mathbf{z}_i = 1$. Let $n \geq m$ and with every observation t, let there be associated a set of $n - m$ hidden *within-speaker factors*, $\mathbf{Y}_t = \{\mathbf{y}_{ti}\}_{i=m+1}^n$, where $\mathbf{y}_{ti} \in \mathbb{S}^{d_i - 1}$ and all $d_i \geq 1$. (If $m = n$, it is understood that there are no within-speaker factors.) We define $D_s = \sum_{i=1}^n d_i$ and restrict $D_s \leq D$. For an observation, $\mathbf{x}_t \in \mathbb{S}^{D-1}$, we define the VMF likelihood:

$$P(\mathbf{x}_t \mid \mathbf{Z}, \mathbf{Y}_t) = \mathcal{V}(\mathbf{x}_t \mid \boldsymbol{\mu}_t, \kappa), \tag{4}$$

where $\kappa > 0$ represents the unstructured component of the within-speaker variability. Structured within and between-speaker variabilities are obtained by letting the mean direction, $\boldsymbol{\mu}_t \in \mathbb{S}^{D-1}$, be a linear combination of the hidden variables:[4]

$$\boldsymbol{\mu}_t = \sum_{i=1}^m w_i \mathbf{K}_i \mathbf{z}_i + \sum_{i=m+1}^n w_i \mathbf{K}_i \mathbf{y}_{ti}. \tag{5}$$

[1] The contributions of Niko Brümmer and Albert Swart to this paper were performed while they were employed by Phonexia and before they joined Amazon and Speechly, respectively.

[2] https://github.com/bsxfan/Toroidal-PSDA

[3] We include the degenerate case, $d_i = 1$, where $\mathbb{S}^0 = \{-1, 1\}$.

[4] Attention: i is *not* a speaker index. All of the variables $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^m$ represent a *single* speaker.

By restricting the *weights* as $\sum_{i=1}^n w_i^2 = 1$ and the *factor loading matrices*, $\mathbf{K}_i \in \mathbb{R}^{D\times d_i}$, as $\mathbf{K}_i'\mathbf{K}_i = \mathbf{I}_{d_i}$ and $\mathbf{K}_i'\mathbf{K}_{j\neq i} = \mathbf{0}$, we ensure $\boldsymbol{\mu}_t'\boldsymbol{\mu}_t = 1$. This linear combination spans a D_s-dimensional *linear subspace*, but since there are n restrictions of the forms $\mathbf{z}_i'\mathbf{z}_i = 1$ and $\mathbf{y}_{ti}'\mathbf{y}_{ti} = 1$, $\boldsymbol{\mu}_t$ is in fact restricted to submanifold of dimension $D_s - n$. In the special case that all $d_i = 2$, this manifold is known as the *Clifford torus*—in the general case, we use the term *toroidal manifold* and use it to name the model *toroidal PSDA* (T-PSDA).

For given observations, their dimensionality D is fixed, but we have some freedom to choose the hidden variable structure. We can choose the linear subspace dimension, D_s, and the number of factors, n, and their dimensions, d_i. We would choose $D_s < D$ if we believe the data lives close to a *linear* subspace of $\mathbb{R}^D$. Moreover, the more factors (n of them) we choose, the 'thinner' the toroidal manifold becomes because it is of dimension $D_s - n$. However, the VMF likelihood (4) allows the data to stray away from $\boldsymbol{\mu}_t$ and appear anywhere on $\mathbb{S}^{D-1}$. The VMF concentration, κ, controls how far the data can stray from the toroid where $\boldsymbol{\mu}_t$ lives.

If we set $n = m = 1$ and $d_1 = D$, then T-PSDA degenerates to the PSDA model of [10]. If additionally, we fix $\gamma_1 = 0$, the scoring with the resulting model is equivalent to cosine scoring.

2.3. Hidden variable prior and posterior

As in PLDA, speakers are independent. For a given speaker with observed data, $\mathbf{X} = \{\mathbf{x}_t\}_{t=1}^T$, the associated hidden variables are $\mathbf{Z}$ and $\mathbf{Y} = \{\mathbf{Y}_t\}_{t=1}^T$. The T-PSDA model is completed by specifying a *conjugate prior*:

$$P(\mathbf{Z},\mathbf{Y}) = \prod_{i=1}^m \mathcal{V}(\mathbf{z}_i \mid \mathbf{v}_i, \gamma_i) \prod_{t=1}^T \prod_{i=m+1}^n \mathcal{V}(\mathbf{y}_{ti} \mid \mathbf{v}_i, \gamma_i), \tag{6}$$

where $\mathbf{v}_i \in \mathbb{S}^{d_i-1}$ and $\gamma_i \geq 0$ are trainable parameters. The posterior is proportional to (and can be recovered from) the joint distribution:

$$\log P(\mathbf{X},\mathbf{Z},\mathbf{Y}) = \sum_{i=1}^m \mathbf{z}_i' \big[\gamma_i\mathbf{v}_i + \kappa w_i \mathbf{K}_i' \sum_{t=1}^T \mathbf{x}_t\big] + \sum_{i=m+1}^n \sum_{t=1}^T \mathbf{y}_{ti}' \big[\gamma_i \mathbf{v}_i + \kappa w_i \mathbf{K}_i'\mathbf{x}_t\big] + \text{const.}$$

This shows the posterior remains factorial—*all the hidden variables remain independent.*[5] The posterior factors are:

$$P(\mathbf{z}_i \mid \mathbf{X}) = \mathcal{V}(\mathbf{z}_i \mid \tilde{\mathbf{v}}_i), \quad \tilde{\mathbf{v}}_i = \gamma_i\mathbf{v}_i + \kappa w_i \mathbf{K}_i' \sum_{t=1}^T \mathbf{x}_t, \tag{7}$$

$$P(\mathbf{y}_{ti} \mid \mathbf{x}_t) = \mathcal{V}(\mathbf{y}_{ti} \mid \tilde{\mathbf{v}}_{ti}), \quad \tilde{\mathbf{v}}_{ti} = \gamma_i\mathbf{v}_i + \kappa w_i \mathbf{K}_i'\mathbf{x}_t, \tag{8}$$

where we have used the natural VMF parametrization of (3).

2.4. Scoring

Because of the conjugacy, the hidden variables can be integrated out using the candidate's trick [12]:

$$P(\mathbf{X}) = \frac{P(\mathbf{X} \mid \mathbf{Z}_0, \mathbf{Y}_0) P(\mathbf{Z}_0) P(\mathbf{Y}_0)}{P(\mathbf{Z}_0 \mid \mathbf{X}) P(\mathbf{Y}_0 \mid \mathbf{X})}, \tag{9}$$

where $\mathbf{Z}_0$ and $\mathbf{Y}_0$ are any convenient values for the hidden variables. Using this for two sets of observations, $\mathbf{E}$ and $\mathbf{T}$, the likelihood ratio (LR) between the *same-speaker* and *different-speakers* hypotheses can be expressed as:

$$\text{LR} = \frac{P(\mathbf{E},\mathbf{T})}{P(\mathbf{E})P(\mathbf{T})} = \frac{P(\mathbf{Z}_0 \mid \mathbf{E})P(\mathbf{Z}_0 \mid \mathbf{T})}{P(\mathbf{Z}_0 \mid \mathbf{E},\mathbf{T})P(\mathbf{Z}_0)}, \tag{10}$$

where factors involving $\mathbf{Y}_0$ cancel. By denoting $\mathbf{Z}_0 = \{\mathbf{z}_i\}_{i=1}^m$ and plugging in (7), the LR becomes:

$$\begin{aligned}\text{LR} &= \prod_{i=1}^m \frac{\mathcal{V}(\mathbf{z}_i \mid \gamma_i\mathbf{v}_i + \kappa w_i \mathbf{K}_i'\tilde{\mathbf{e}})\, \mathcal{V}(\mathbf{z}_i \mid \gamma_i\mathbf{v}_i + \kappa w_i \mathbf{K}_i'\tilde{\mathbf{t}})}{\mathcal{V}(\mathbf{z}_i \mid \gamma_i\mathbf{v}_i + \kappa w_i \mathbf{K}_i'(\tilde{\mathbf{e}}+\tilde{\mathbf{t}}))\, \mathcal{V}(\mathbf{z}_i \mid \gamma_i\mathbf{v}_i)} \\ &= \prod_{i=1}^m \frac{\mathcal{V}(\mathbf{z}_i \mid \boldsymbol{\ell}_i)\, \mathcal{V}(\mathbf{z}_i \mid \mathbf{r}_i)}{\mathcal{V}(\mathbf{z}_i \mid \mathbf{b}_i)\, \mathcal{V}(\mathbf{z}_i \mid \mathbf{n}_i)}.\end{aligned}$$

Here, $\tilde{\mathbf{e}}$ and $\tilde{\mathbf{t}}$ are the sums of the observations in respectively $\mathbf{E}$ and $\mathbf{T}$. We have introduced mnemonic short-hand notation for the parameters of the above VMF factors: $\boldsymbol{\ell}$ for left, $\mathbf{r}$ for right, $\mathbf{b}$ for both, and $\mathbf{n}$ for none. In the score, all factors of the form $e^{\mathbf{a}'\mathbf{z}_i}(2\pi)^{-d_i/2}$ cancel—see (3), so that the score simplifies to:

$$\text{LR} = \prod_{i=1}^m \frac{\bar{C}_{\nu_i}(\|\boldsymbol{\ell}_i\|)e^{-\|\boldsymbol{\ell}_i\|}\, \bar{C}_{\nu_i}(\|\mathbf{r}_i\|)e^{-\|\mathbf{r}_i\|}}{\bar{C}_{\nu_i}(\|\mathbf{b}_i\|)e^{-\|\mathbf{b}_i\|}\, \bar{C}_{\nu_i}(\|\mathbf{n}_i\|)e^{-\|\mathbf{n}_i\|}}. \tag{11}$$

To get an idea of what the score does, let us assume the ideal situation where the $\gamma_i = 0$, so that the speaker factors have uniform distributions. Noting that $\bar{C}_\nu(x)$ is almost constant compared to e^{-x} on the scale that x typically varies, the log LR can be *approximated* as:

$$\kappa \sum_{i=1}^m |w_i| \Big(\big\|\mathbf{K}_i'(\tilde{\mathbf{e}}+\tilde{\mathbf{t}})\big\| - \big\|\mathbf{K}_i'\tilde{\mathbf{e}}\big\| - \big\|\mathbf{K}_i'\tilde{\mathbf{t}}\big\| \Big) + \text{const.}$$

The score becomes more positive if $\tilde{\mathbf{e}}$ and $\tilde{\mathbf{t}}$ are aligned and more negative if they are not. If $\tilde{\mathbf{e}}$, or $\tilde{\mathbf{t}}$, or both are summed over multiple aligned inputs, then the score magnitude can increase. The score is essentially a linear fusion, weighted by the $|w_i|$. The score magnitude is also scaled by the observation VMF concentration, κ.

Finally, note that when scoring, we never use the w_i and the $\mathbf{K}_i$ for $i > m$, i.e., for the within-speaker subspaces. It may be asked what is the use of this part of the model? Yes, that part does not participate in scoring, but it *does* play a part when learning the model parameters. The three sources of variability—the unstructured VMF noise (parametrized by κ) and the structured within and between-speaker variabilities—*compete* to explain the total variability in the observed data.

2.5. Training

For a given T-PSDA architecture, the parameters are κ and $\{\mathbf{K}_i, w_i, \mathbf{v}_i, \gamma_i\}_{i=1}^n$ and they can be learned with maximum likelihood, using an EM algorithm. The E-step is to compute the *EM auxiliary*, i.e. the log-likelihood expectation w.r.t. the hidden variable posterior:

$$Q = \sum_{s=1}^S \big\langle \log P(\mathbf{X}_s \mid \mathbf{Z},\mathbf{Y}) + \log P(\mathbf{Z},\mathbf{Y}) \big\rangle_{P(\mathbf{Z},\mathbf{Y}\mid\mathbf{X}_s)},$$

where the data for speaker s is $\mathbf{X}_s = \{\mathbf{x}_{st}\}_{t=1}^{T_s}$, and S is the total number of speakers. Recall that the posterior is a product of VMF factors given by (7) and (8). Since the log-likelihood is

[5] Why do we not get explaining away? After all, n hidden variables are jointly responsible for each $\mathbf{x}_t$. This can be understood by the mutually orthogonal factor loading matrices, the $\mathbf{K}_i$. The hidden factor i is actually solely responsible for the projected observation $\mathbf{K}_i'\mathbf{x}_t$.

linear in the hidden variables, we can simply plug in the *posterior expectations*, $\bar{\mathbf{z}}_{si}$ and $\bar{\mathbf{y}}_{sti}$, in place of the hidden variables:

$$Q = Q^{(x)} + \sum_{i=1}^{m} Q_i^{(z)} + \sum_{i=m+1}^{n} Q_i^{(y)}, \tag{12}$$

where

$$\begin{aligned} Q^{(x)} &= \sum_{s=1}^{S}\sum_{t=1}^{T_s} \log \mathcal{V}(\mathbf{x}_{st} \mid \bar{\boldsymbol{\mu}}_{st}, \kappa), \\ Q_i^{(z)} &= \sum_{s=1}^{S} \log \mathcal{V}(\bar{\mathbf{z}}_{si} \mid \mathbf{v}_i, \gamma_i), \\ Q_i^{(y)} &= \sum_{s=1}^{S}\sum_{t=1}^{T_s} \log \mathcal{V}(\bar{\mathbf{y}}_{sti} \mid \mathbf{v}_i, \gamma_i), \end{aligned} \tag{13}$$

where we have defined:

$$\bar{\boldsymbol{\mu}}_{st} = \sum_{i=1}^{m} w_i \mathbf{K}_i \bar{\mathbf{z}}_{si} + \sum_{i=m+1}^{n} w_i \mathbf{K}_i \bar{\mathbf{y}}_{sti}. \tag{14}$$

The M-step maximizes Q w.r.t. all the parameters. We can maximize the above Q-terms independently. For the $Q_i^{(z)}$ and $Q_i^{(y)}$, these are standard VMF maximum likelihood problems:

$$\begin{aligned} \mathbf{v}_i, \gamma_i &\leftarrow \underset{\mathbf{v},\gamma}{\operatorname{argmax}} \prod_{s=1}^{S} \mathcal{V}(\bar{\mathbf{z}}_{si} \mid \mathbf{v}, \gamma), \quad i \leq m, \\ \mathbf{v}_i, \gamma_i &\leftarrow \underset{\mathbf{v},\gamma}{\operatorname{argmax}} \prod_{s=1}^{S}\prod_{t=1}^{T_s} \mathcal{V}(\bar{\mathbf{y}}_{sti} \mid \mathbf{v}, \gamma), \quad m < i \leq n. \end{aligned} \tag{15}$$

It remains to maximize $Q^{(x)}$. First, we can maximize it w.r.t. $\{\mathbf{K}_i, w_i\}$, independently of κ, which is equivalent to maximizing:

$$\begin{aligned} &\sum_{s=1}^{S}\sum_{t=1}^{T_s}\left[\sum_{i=1}^{m} w_i \mathbf{x}_{st}' \mathbf{K}_i \bar{\mathbf{z}}_{si} + \sum_{i=m+1}^{n} w_i \mathbf{x}_{st}' \mathbf{K}_i \bar{\mathbf{y}}_{sti}\right] \\ &= \sum_{i=1}^{m} w_i \operatorname{tr}\Big(\mathbf{K}_i \sum_{s} \bar{\mathbf{z}}_{si} \sum_{t} \mathbf{x}_{st}'\Big) + \sum_{i=m+1}^{n} w_i \operatorname{tr}\Big(\mathbf{K}_i \sum_{st} \bar{\mathbf{y}}_{sti} \mathbf{x}_{st}'\Big) \\ &= \sum_{i=1}^{n} w_i \operatorname{tr}(\mathbf{K}_i \mathbf{R}_i'), \end{aligned}$$

where $\mathbf{R}_i$ is defined by matching the final line to the one above. We do a few iterations of co-ordinate ascent, alternatively updating $\mathbf{w} = (w_1, \ldots, w_n)$ and $\mathbf{F} = \begin{bmatrix}\mathbf{K}_1 & \cdots & \mathbf{K}_n\end{bmatrix}$. When $\mathbf{F}$ is temporarily fixed, we define $\tilde{\mathbf{w}} = (\tilde{w}_1, \ldots, \tilde{w}_n)$, where $\tilde{w}_i = \operatorname{tr}(\mathbf{K}_i \mathbf{R}_i')$. Conversely, when $\mathbf{w}$ is temporarily fixed, we define $\tilde{\mathbf{F}} = \begin{bmatrix} w_1 \mathbf{R}_1 & \cdots & w_n \mathbf{R}_n\end{bmatrix}$. The maximizing updates, subject to the constraints $\mathbf{w}'\mathbf{w} = 1$ and $\mathbf{F}'\mathbf{F} = \mathbf{I}$, are:[6]

$$\mathbf{w} \leftarrow \frac{\tilde{\mathbf{w}}}{\|\tilde{\mathbf{w}}\|} \qquad \text{and} \qquad \mathbf{F} \leftarrow \tilde{\mathbf{F}}(\tilde{\mathbf{F}}'\tilde{\mathbf{F}})^{-\frac{1}{2}}. \tag{16}$$

Finally, once the optimal values for $\mathbf{w}$ and $\mathbf{F}$ have been obtained, we can fix them to find the optimal κ:

$$\begin{aligned} \kappa &\leftarrow \underset{\kappa}{\operatorname{argmax}} \sum_{st} \log \mathcal{V}(\mathbf{x}_{st} \mid \bar{\boldsymbol{\mu}}_{st}) \\ &= \underset{\kappa}{\operatorname{argmax}}\, T \log \frac{\kappa^{\nu}}{I_{\nu}(\kappa)} + \kappa \sum_{i=1}^{n} w_i \operatorname{tr}(\mathbf{K}_i \mathbf{R}_i'), \end{aligned} \tag{17}$$

where $T = \sum_s T_s$ and $\nu = \frac{D}{2} - 1$. This scalar optimization can be done with a general-purpose (typically derivative-free) numerical optimization algorithm.

3. Experiments

We perform the experiments with the T-PSDA model on two datasets: NIST SRE'21 [13] and VoxCeleb [14, 15]. In both cases, we compare the speaker verification performance of the proposed model with cosine scoring and PLDA. Experiments with the original PSDA are not repeated here, because it performs on par with cosine scoring [10].

3.1. NIST SRE'21

For the experiments on the NIST SRE'21 evaluation set, we use exactly the same ResNet152 embedding extractor as used in [16]. We used the version of the extractor that was fine-tuned on long (10 s) segments. This embedding extractor was used as input for a number of different scoring back-ends that we compare below.

The PLDA and T-PSDA models require training, while cosine scoring does not. All back-end models were trained on the full NIST CTS superset [17] unlike in [16] where only the English, Mandarin, and Cantonese subsets were used (hence, there are minor performance differences between the results of this paper and those presented in [16]). In all cases, the embeddings were centered and length-normalized. We optionally used linear discriminant analysis (LDA) to reduce the dimensionality of the embeddings from 256 to 100, because we have observed before [16] that roughly half of the embedding dimensions have almost no useful variability and that LDA improved some of the back-ends. The parameters for LDA and centering were obtained from the same training set.

Table 1: *Performance of the baseline and T-PSDA back-ends with and without score normalization on evaluation set of NIST SRE'21. The performance metrics are minimum cost* (min_C) *and equal-error-rate (EER, %) as computed by the NIST scoring tool.*

		no S-norm		S-norm	
		min_C	**EER**	**min_C**	**EER**
1	cos	0.490	9.18	0.520	8.62
2	cos + LDA	0.468	8.10	0.596	7.77
3	PLDA	0.444	7.88	0.653	7.55
4	PLDA + LDA	0.452	7.78	0.650	7.80
5	T-PSDA	**0.381**	**6.16**	**0.375**	**5.77**

Table 1 compares the proposed T-PSDA against the two baseline models (cosine scoring and PLDA), with and without LDA and with and without adaptive score normalization [18]. For PLDA we set the sizes of each of the speaker and channel subspaces to 100.[7] For score normalization, we used the 400 highest scores of the enrollment and test segments against 5000 randomly chosen embeddings from the training set. The results of the baseline models are shown in lines 1 to 4 of Table 1. Notice that score normalization leads to significant performance degradation in terms of minimum cost for all baseline back-ends, while in some cases, it improves equal error rate (EER).

[6] The $\mathbf{F}$ update requires the symmetric, positive-definite matrix square root.

[7] For PLDA after LDA down to 100 dimensions, the subspaces are therefore of full rank, so that PLDA degenerates to the *two-covariance* variant [6].

Table 2: *Performance of the baseline and T-PSDA back-ends. The performance is reported in terms of equal-error-rate (EER, %) and minimum Detection Cost Function computed at target probability* $p_{\mathrm{tar}} = 0.05$ *(*$\mathrm{minDCF}_{0.05}$*).*

		no S-norm				**S-norm**			
		Vox1-O		**Vox1-H**		**Vox1-O**		**Vox1-H**	
		$\mathbf{minDCF_{0.05}}$	**EER**	$\mathbf{minDCF_{0.05}}$	**EER**	$\mathbf{minDCF_{0.05}}$	**EER**	$\mathbf{minDCF_{0.05}}$	**EER**
1	cos	0.071	1.10	**0.126**	**2.13**	**0.061**	1.01	**0.118**	**2.03**
2	cos +LDA	0.116	1.54	0.263	4.15	0.100	1.38	0.160	2.62
3	PLDA	0.254	4.47	0.325	6.80	0.213	3.48	0.285	6.05
4	PLDA +LDA	0.113	1.55	0.198	3.44	0.106	1.40	0.152	2.96
5	T-PSDA	**0.069**	**1.07**	0.127	2.16	0.065	**0.97**	0.119	2.05

Also, let us notice that in this experiment (NIST'21), using LDA is beneficial for cosine scoring, but not for PLDA. For the other experiment (VoxCeleb, described below), we observe the *opposite*.

T-PSDA provides great freedom in selecting its parameters: the number and sizes of the hidden speaker and channel variables. For this reason, it is not feasible to consider all possible combinations of T-PSDA settings to select the best one. We used uniform hidden variable priors: we set all $\gamma_i = 0$ and did not learn the prior parameters. For the other parameters, we used the following strategy: we start from the simplest configuration with a single speaker variable and no channel variables. Gradually, one parameter at a time, we increase the complexity of the model. First, we optimize the dimensionality of the single speaker variable; then we experiment with having several speaker variables such that their summed dimensions are the same as what we found optimal in the first step. After this, having the speaker variable dimensions fixed, we introduce channel variability and similarly optimize the number and size of the channel variables. Following this approach, we arrive at the optimal architecture of T-PSDA model: we fix the number of hidden speaker variables to one ($m = 1$) and the dimensionality of this variable ($d_1 = 60$); also we use two 5-dimensional hidden channel variables ($n = 3,\ d_2 = d_3 = 5$). The performance of the final model is shown in line 5 of Table 1.

Comparing T-PSDA to the baselines, we observe significant performance improvement in both cases: when score normalization is performed or not. Also, notice that T-PSDA benefits from score normalization, unlike the baselines. However, it is important to mention that these results are achieved with a model found by a greedy search in the space of the T-PSDA configurations. T-PSDA performance greatly depends on the correct settings for the number and dimensionality of the hidden variables; in some experiments with different configurations of the model, the performance was considerably worse.

3.2. VoxCeleb

For the Audio from Video (AfV) data experiments, we replicate the experimental setup of [10] where a ResNet34 embedding extractor trained on the development part of the VoxCeleb2 dataset was used, and the performance was tested on the original test set of VoxCeleb1 (Vox1-O) and a set of "hard" trials constructed out of the whole VoxCeleb1 (Vox1-H). The performance is evaluated in terms of EER and minimum Detection Cost Function with the probability of the target trial set to 0.05.

The results of the baseline models are shown on lines 1 to 4 of Table 2. The results for cosine scoring and the PLDA model are shown with and without LDA reducing the dimensionality of the embeddings from 256 to 200. When the PLDA model is trained on the raw embeddings without dimensionality reduction, we set the sizes of each of the speaker and channel subspaces to 100. When LDA is applied, we train the PLDA with full-rank within and across-class covariances. Notice that LDA is critical for a good performance of the PLDA model, while for cosine distance scoring, it is rather detrimental. Also, for all baselines, we show the performance with adaptive score normalization. Score normalization is done the same way as for SRE experiments: we use the highest 400 scores of the trial sides against 5000 embeddings from the training set.

To find the appropriate architecture of the T-PSDA model, we followed an approach similar to the one used in NIST SRE'21 experiments: looking for one configuration parameter at a time and treating the others as fixed. In this way, we found the optimal architecture of the T-PSDA model which is having a single 120-dimensional speaker variable, i.e., $m = 1,\ d_1 = 120$, and five 1-dimensional hidden channel variables ($n = 6,\ d_2 = \ldots = d_6 = 1$). The performance of this model is shown on line 5 of Table 2. For this system, we provide the results with and without score normalization. As seen from the results, for AfV data, T-PSDA performs on par with the best-performing baseline method (cosine scoring), outperforming PLDA.

4. Conclusion

We have generalized the simple PSDA model of [10] to a new model called T-PSDA. The PSDA model is too simple: lacking trainable parameters, it was not able to outperform cosine scoring in a situation where a trainable PLDA model was able to outperform cosine scoring in a new domain (NIST SRE'21). The T-PSDA model has a set of trainable parameters of a size similar to PLDA. Like PLDA, T-PSDA can model within and between-speaker variabilities in subspaces, while T-PSDA has the advantage of using VMF distributions that better model length-normalized embeddings. These benefits gave a clear performance advantage for T-PSDA on the NIST data. In the VoxCeleb experiment, where the embedding extractor is trained on in-domain data, T-PSDA does not benefit from its domain adaptation capability and performs on par with cosine scoring. In contrast, PLDA performs worse on VoxCeleb.

5. Acknowledgements

The work was supported by Czech National Science Foundation (GACR) project NEUREM3 No. 19-26934X, and Horizon 2020 Marie Sklodowska-Curie grant ESPERANTO, No. 101007666. Computing on IT4I supercomputer was supported the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90140).

6. References

[1] S. Ioffe, “Probabilistic linear discriminant analysis,” in *9th European Conference on Computer Vision*, Graz, Austria, 2006.

[2] S. J. Prince and J. H. Elder, “Probabilistic linear discriminant analysis for inferences about identity,” in *IEEE 11th International Conference on Computer Vision*, 2007.

[3] 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*, vol. 19, no. 4, pp. 788–798, May 2011.

[4] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, “Deep neural network embeddings for text-independent speaker verification.” in *Interspeech*, 2017, pp. 999–1003.

[5] P. Kenny, “Bayesian speaker verification with heavy-tailed priors,” in *Odyssey Speaker and Language Recognition Workshop*, Brno, Czech Republic, June 2010, keynote presentation.

[6] N. Brümmer and E. de Villiers, “The speaker partitioning problem,” in *Odyssey 2010: The speaker and Language Recongnition Workshop, Brno*, 2010.

[7] D. Garcia-Romero and C. Y. Espy-Wilson, “Analysis of i-vector length normalization in speaker recognition systems,” in *Interspeech*, Florence, Italy, 2011.

[8] M. Zhao, Y. Ma, M. Liu, and M. Xu, “The SpeakIn system for VoxCeleb Speaker Recognition Challenge 2021,” *CoRR*, vol. abs/2109.01989, 2021. [Online]. Available: https://arxiv.org/abs/2109.01989

[9] H. Zeinali, S. Wang, A. Silnova, P. Matějka, and O. Plchot, “BUT system description to VoxCeleb Speaker Recognition Challenge 2019,” *arXiv preprint arXiv:1910.12592*, 2019.

[10] N. Brummer, A. Swart, L. Mosner, A. Silnova, O. Plchot, T. Stafylakis, and L. Burget, “Probabilistic Spherical Discriminant Analysis: An Alternative to PLDA for length-normalized embeddings,” in *Proc. Interspeech 2022*, 2022, pp. 1446–1450.

[11] K. V. Mardia and P. E. Jupp, *Directional Statistics*. Wiley, 2000.

[12] J. Besag, “A candidate’s formula: A curious result in bayesian prediction,” *Biometrika*, vol. 76, pp. 183–183, 1989.

[13] S. O. Sadjadi, C. Greenberg, E. Singer, L. Mason, and D. Reynolds, “The 2021 nist speaker recognition evaluation,” *arXiv preprint arXiv:2204.10242*, 2022.

[14] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, “Voxceleb: Large-scale speaker verification in the wild,” *Computer Speech and Language*, 2019.

[15] J. S. Chung, A. Nagrani, and A. Zisserman, “Voxceleb2: Deep speaker recognition,” *arXiv preprint arXiv:1806.05622*, 2018.

[16] A. Silnova, T. Stafylakis, L. Mošner, O. Plchot, J. Rohdin, P. Matějka, L. Burget, O. Glembek, and N. Brümmer, “Analyzing speaker verification embedding extractors and back-ends under language and channel mismatch,” in *Odyssey 2022: The speaker and Language Recongnition Workshop, Beijing*, 2022. [Online]. Available: https://arxiv.org/abs/2203.10300

[17] O. Sadjadi, “NIST SRE CTS Superset: A large-scale dataset for telephony speaker recognition,” 2021-08-16 04:08:00 2021. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=933116

[18] P. Matějka, O. Novotnỳ, O. Plchot, and L. Burget, “Analysis of score normalization in multilingual speaker recognition,” in *Interspeech, Stockholm*, 2017.