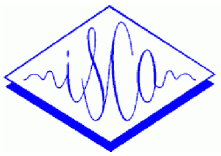
ISCA Archive
http://www.isca-speech.org/archive



INTERSPEECH 2012
ISCA's 13th Annual Conference
Portland, OR, USA
September 9-13, 2012

Bilinear Factor Analysis for iVector Based Speaker Verification

Yun Lei, Lukáš Burget, Nicolas Scheffer

Speech Technology and Research Laboratory, SRI International, California, USA
{yunlei,burget,scheffer}@speech.sri.com

Abstract

The combination of iVector extraction and Probabilistic Linear Discriminant Analysis (PLDA) model forms a basis of the current state of the art speaker verification. The PLDA model makes an assumption that the within-speaker (or inter-session) variability in the iVector space is independent of speaker identity. In this work we propose a new model, which can be seen as an extension of PLDA, relaxing this assumption and allowing the within-speaker variability to be different for different locations of speakers in the iVector space. The potential of the proposed model is demonstrated in preliminary experiments.

10.21437/Interspeech.2012-343

1. Introduction

We propose a new model for speaker verification, which can be seen as an extension of the state-of-the-art Probabilistic Linear Discriminant Analysis (PLDA), relaxing certain assumptions made by the standard PLDA model. More specifically, our model does not strictly assume that inter-session variability in the feature space (e.g. iVector space) is independent of speaker identity.

Current state-of-the-art cepstral speaker verification systems are based on a combination of two techniques: iVector extraction and PLDA-based verification. iVector extraction [1] is a process where a sequence of conventional speech features (e.g., MFCC) is converted into a single low-dimensional feature vector representing important information about the speaker (and channel) of a given speech segment. Once iVectors are extracted from speech segments, the task in speaker verification is to decide whether or not two iVectors come from the same speaker. Simple methods such as using cosine distance to compare iVectors were proposed for this purpose [1]. However, currently the most successful techniques for calculating verification scores are based on the PLDA model [2, 3]. The basic "Gaussian" form of PLDA is a generative model making LDA-like assumptions: For each speaker, iVectors are assumed to be Gaussian distributed with a speaker-specific mean and within-class covariance matrix shared across all speakers. Here, the within-class covariance matrix represents the inter-session (or channel) variability in the iVector space. The speaker mean, which is a hidden variable in the model, is assumed to be Gaussian distributed with the speaker variability described by an across-class covariance matrix. Typically, maximum likelihood (ML) estimates of PLDA parameters describing the within- and across-speaker distributions are estimated using the expectation maximization (EM) algorithm [2, 3]. Given a pair of iVectors, verification score can be evaluated using the PLDA model as a ratio between likelihoods of two hypotheses: the iVectors were generated from the same speaker or independently. When calculating the likelihoods, we need to integrate over the hidden variable, which can be done very efficiently in the case of the basic Gaussian PLDA, where the integral can be solved analytically [4, 5].

Several extensions of the PLDA model were proposed. To improve the iVector distribution fit, so-called Heavy-tailed PLDA [3] models within- and across-speaker variability used the more expressive Student's t-distribution. Bayesian PLDA [6] was proposed to cope with over-fitting the across-speaker variability, which is learned from a limited number of training speaker examples. In these PLDA extensions, however, calculation of the posterior distribution of hidden variables needed by the EM algorithm becomes intractable and one has to resort to an approximate probabilistic inference such as variational Bayes. This makes model training, and especially verification score evaluation, much more costly. Moreover, although the PLDA extensions were reported to bring significant gains in verification performance, it was found that similar gains can be obtained with the basic PLDA model applied to iVectors post-processed by LDA dimensionality reduction and length normalization [7].

Like Gaussian PLDA, the model proposed in this paper assumes within- and across-speaker distributions to be Gaussian. However, the within-class covariance matrix is not constant for different speakers. Instead, it is assumed to be a function of the speaker mean. In other words, depending on where the speaker is in the iVector space, inter-session variability can result in different distributions of iVectors around the speaker mean. As in the case of the aforementioned PLDA extensions, variational Bayes inference is used for training and evaluation of our model.

2. Bilinear Factor Analysis model

Our model is derived from the standard PLDA model [2, 3], where the distribution of iVectors is described using a factor analysis formula

$$\mathbf{m} = \mathbf{a} + \mathbf{V}\mathbf{y} + \mathbf{U}\mathbf{x} + \boldsymbol{\epsilon}, \qquad (1)$$

where $\mathbf{m}$ is a random vector representing an iVector, $\mathbf{a}$ is the mean of the iVector distribution, $\mathbf{V}$ and $\mathbf{U}$ are matrices describing across- and within-speaker variability, and $\mathbf{y}$ and $\mathbf{x}$ are standard normally distributed latent vectors representing speaker and channel components of the iVector. The random variable $\epsilon = \mathcal{N}(0, \mathbf{D}^{-1})$ with diagonal precision matrix $\mathbf{D}$ describes the residual within-speaker variability not covered by subspace $\mathbf{U}$. In this formula, the term $\mathbf{a} + \mathbf{V}\mathbf{y}$ represents the across-speaker Gaussian distribution with mean $\mathbf{a}$ and across-speaker covariance matrix $\mathbf{V}\mathbf{V}^T$. The term $\mathbf{U}\mathbf{x} + \boldsymbol{\epsilon}$ represents the within-

This material is based on work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. D10PC20024. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. "A" (Approved for Public Release, Distribution Unlimited)

speaker distribution, where the within-speaker covariance matrix is $\mathbf{U}\mathbf{U}^T + \mathbf{D}^{-1}$.

For convenience, let us rewrite the formula for individual dimensions of iVector $\mathbf{m}$:

$$m_k = a_k + \mathbf{v}_k\mathbf{y} + \mathbf{u}_k\mathbf{x} + \epsilon_k, \tag{2}$$

where $\mathbf{v}_k$ and $\mathbf{u}_k$ are the k-th rows of the matrix $\mathbf{V}$ and $\mathbf{U}$, respectively, and $\epsilon_k = \mathcal{N}(0, d_k^{-1})$. As was already mentioned, our goal is to make the within-speaker distribution dependent on speaker identity represented by the latent variable $\mathbf{y}$. For this purpose, we introduce an additional term into the formula (2) which is a bilinear form of the latent variables $\mathbf{y}$ and $\mathbf{x}$:

$$\begin{aligned} m_k &= a_k + \mathbf{v}_k\mathbf{y} + \mathbf{u}_k\mathbf{x} + \mathbf{y}^T\mathbf{A}_k\mathbf{x} + \epsilon_k \\ &= \mathbf{y}^{+T}\mathbf{A}_k^+\mathbf{x}^+ + \epsilon_k, \end{aligned} \tag{3}$$

where

$$\mathbf{y}^{+T} = \begin{bmatrix} 1 & \mathbf{y}^T \end{bmatrix} \tag{4}$$

$$\mathbf{x}^{+T} = \begin{bmatrix} 1 & \mathbf{x}^T \end{bmatrix} \tag{5}$$

$$\mathbf{A}_k{}^+ = \begin{bmatrix} a_k & \mathbf{u}_k^T \\ \mathbf{v}_k & \mathbf{A}_k \end{bmatrix}. \tag{6}$$

Note that the last line of equation (3) resembles a standard factor analysis formula [8], with the usual linear term describing the variability of data in a subspace replaced by a bilinear form. This is why we have chosen the name Bilinear Factor Analysis (BLFA) for our model. Note also that, for a fixed speaker (i.e., fixed variable $\mathbf{y}$), the within-speaker distribution is still Gaussian. However, the covariance matrix of this distribution is $\mathbf{W}\mathbf{W}^T + \mathbf{D}^{-1}$, where the k-th row of matrix $\mathbf{W}$ is $\mathbf{w}_k = (\mathbf{u}_k + \mathbf{y}^T\mathbf{A}_k)$. Therefore, it is a function of the speaker latent variable $\mathbf{y}$. As a consequence, the overall distribution of iVectors described by this model does not have to be Gaussian.

One obvious problem of the proposed extension is the massive increase in the number of trainable model parameters, which are constants a_k, vectors $\mathbf{v}_k$ and $\mathbf{u}_k$, matrices $\mathbf{A}_k$ and precisions d_k, one for each iVector dimension k. The newly introduced matrices $\mathbf{A}_k$, with the number of rows and columns corresponding to dimensionalities of latent variables $\mathbf{y}$ and $\mathbf{x}$ (typically around 200 each), now represents the majority of the parameters. To avoid over-fitting, it will be desirable to control the number of nonzero elements in the matrices $\mathbf{A}_k$ as will be discussed in Section 4. Note that by setting the matrices $\mathbf{A}_k$ to zero, we simply recover the original PLDA model.

2.1. Estimation of model parameters

The likelihood of a training set of iVectors $\mathbf{M} = \{\mathbf{m}_{ij}\}$ can be evaluated as

$$\begin{aligned} p(\mathbf{M}) &= \prod_i \int d\mathbf{y} p(\mathbf{y}) \prod_j \int d\mathbf{x} p(\mathbf{x}) p(\mathbf{m}_{ij}|\mathbf{y},\mathbf{x}) \\ &= \prod_i \int d\mathbf{y} p(\mathbf{y}) \prod_j \int d\mathbf{x} p(\mathbf{x}) \\ &\quad \prod_k p(m_{ijk}|\mathbf{y}^{+T}\mathbf{A}_k^+\mathbf{x}^+, d_k^{-1}), \end{aligned} \tag{7}$$

where index i represents individual speakers in the training data and for each speaker j refers to individual recordings of the same speaker. When evaluating the likelihood, note the marginalization over the standard normal priors $p(\mathbf{y})$ and $p(\mathbf{x})$, which takes into account that recordings of the same speaker are generated using the same value of latent variable $\mathbf{y}$.

The parameters of our model can be estimated to maximize the likelihood of training data. The standard EM algorithm requires evaluation of joint posterior distributions $p(\mathbf{y}_i, \mathbf{x}_{i1}, \mathbf{x}_{i1}, \ldots, \mathbf{x}_{iJ}|\mathbf{M})$ for each speaker i. Unfortunately, for our model, these posterior distributions are not Gaussian as in the case of standard PLDA models and their evaluation is intractable. Therefore, we use approximate variational inference to estimate the parameters, where the usual lower bound on log-likelihood is maximized instead of the true log-likelihood function:

$$\begin{aligned} \ln P(\mathbf{M}) &\geq \sum_{ij} \iint d\mathbf{y}_i d\mathbf{x}_{ij} q(\mathbf{y}_i) q(\mathbf{x}_{ij}) \\ &\quad \sum_k \ln \mathcal{N}(m_{ijk}|\mathbf{y}_i^{+T}\mathbf{A}_k^+\mathbf{x}_{ij}^+, d_k^{-1}) \\ &\quad + \sum_i \int q(\mathbf{y}_i) \ln \frac{p(\mathbf{y}_i)}{q(\mathbf{y}_i)} d\mathbf{y}_i \\ &\quad + \sum_{i,j} \int q(\mathbf{x}_{ij}) \ln \frac{p(\mathbf{x}_{ij})}{q(\mathbf{x}_{ij})} d\mathbf{x}_{ij} \\ &= \mathcal{L} \end{aligned} \tag{8}$$

It can be shown that this expression is a lower bound on the true likelihood function for any valid probability density functions $\{q(\mathbf{y}_i), q(\mathbf{x}_{ij})\}$, which we have chosen to be Gaussian densities[1] and which can be interpreted as approximations to the posterior distribution of all hidden variables. The closer the functions are to the true posteriors, the closer the lower bound is to the true log-likelihood. As we already mentioned, however, the true posteriors are not Gaussian. Furthermore, speaker and channel latent variables are not independent in the true joint posterior $p(\mathbf{y}_i, \mathbf{x}_{i1}, \mathbf{x}_{i1}, \ldots, \mathbf{x}_{iJ}|\mathbf{M})$, while we have used the factorization into independent distributions $q(\mathbf{y}_i)$ (one for each speaker) and $q(\mathbf{x}_{ij})$ (for each recording) as is the usual approximation used for variational Bayes.

Now, our task is to find probability density functions $\{q(\mathbf{y}_i), q(\mathbf{x}_{ij})\}$ and model parameters $\{\mathbf{A}_k^+\}$ and d_k that maximizes the lower bound (8). This is done by iteratively optimizing the parameters and the individual density functions one at a time with the remaining functions and parameters fixed. To find a new estimate of a density function $q(\mathbf{y}_i)$, we take a variational derivative

$$\begin{aligned} \frac{\partial L}{\partial q(\mathbf{y}_i)} &= \sum_j \sum_k \langle \ln \mathcal{N}(m_{ijk}|\mathbf{y}_i^{+T}\mathbf{A}_k^+\mathbf{x}_{ij}^+, d_k^{-1}) \rangle_{q(\mathbf{x}_{ij})} \\ &\quad + \ln p(\mathbf{y}_i) - \ln q(\mathbf{y}_i), \end{aligned} \tag{9}$$

where $\langle\cdot\rangle_{q(\mathbf{x}_{ij})}$ reads as an expectation w.r.t. the current estimate of $q(\mathbf{x}_{ij})$. Setting this derivative equal to zero and solving for $q(\mathbf{y}_i)$ gives us a new estimate of $q(\mathbf{y}_i)$, which is a Gaussian distribution with covariance matrix and the mean vector

$$\begin{aligned} \boldsymbol{\Sigma}_{\mathbf{y}_i} &= \left[\mathbf{I} + \sum_j \sum_k \langle (\mathbf{v_k} + \mathbf{A_k}\mathbf{x}_{ij}) d_k (\mathbf{v_k} + \mathbf{A_k}\mathbf{x}_{ij})^T \rangle_{q(\mathbf{x}_{ij})} \right]^{-1} \\ &= \Big[\mathbf{I} + \sum_j \sum_k d_k (\mathbf{v_k}\mathbf{v_k}^T + \mathbf{v_k}\langle \mathbf{x}_{ij}^T \rangle \mathbf{A_k}^T + \mathbf{A_k}\langle \mathbf{x}_{ij} \rangle \mathbf{v_k}^T \\ &\quad + \mathbf{A_k}\langle \mathbf{x}_{ij}\mathbf{x}_{ij}^T \rangle \mathbf{A_k}^T) \Big]^{-1} \end{aligned} \tag{10}$$

[1] More precisely, the Gaussianity follows from the assumption, which we make about the independency of these distributions.

$$\langle \mathbf{y}_i \rangle = \boldsymbol{\Sigma}_{\mathbf{y}_i} \sum_j \sum_k \langle (m_{ijk} - a_k - \mathbf{u_k}^T \mathbf{x}_{ij}) d_k (\mathbf{v_k} + \mathbf{A_k} \mathbf{x}_{ij}) \rangle_{q(\mathbf{x}_{ij})}$$
$$= \boldsymbol{\Sigma}_{\mathbf{y}_i} \sum_j \sum_k d_k \Big[(m_{ijk} - a_k)(\mathbf{v_k} + \mathbf{A_k} \langle \mathbf{x}_{ij} \rangle) - \mathbf{u_k}^T \langle \mathbf{x}_{ij} \rangle \mathbf{v_k} - \mathbf{A_k} \langle \mathbf{x}_{ij} \mathbf{x}_{ij}^T \rangle \mathbf{u_k} \Big]. \quad (11)$$

Symmetrically, for a density function $q(\mathbf{x}_{ij})$, the updates for the covariance matrix and the mean vector are

$$\boldsymbol{\Sigma}_{\mathbf{x}_{ij}} = \left[\mathbf{I} + \sum_k \langle (\mathbf{u_k}^T + \mathbf{y}_i^T \mathbf{A_k})^T d_k (\mathbf{u_k}^T + \mathbf{y}_i^T \mathbf{A_k}) \rangle_{q(\mathbf{y}_i)} \right]^{-1}$$
$$= \Big[I + \sum_k d_k (\mathbf{u_k} \mathbf{u_k}^T + \mathbf{u_k} \langle \mathbf{y}_i^T \rangle \mathbf{A_k} + \mathbf{A_k}^T \langle \mathbf{y}_i \rangle \mathbf{u_k}^T + \mathbf{A_k}^T \langle \mathbf{y}_i \mathbf{y}_i^T \rangle \mathbf{A_k}) \Big]^{-1} \quad (12)$$

$$\langle \mathbf{x}_{ij} \rangle = \boldsymbol{\Sigma}_{\mathbf{x}_{ij}} \sum_k \langle (m_{ijk} - a_k - \mathbf{y}_i^T \mathbf{v_k}) d_k (\mathbf{u_k}^T + \mathbf{y}_i^T \mathbf{A_k})^T \rangle_{q(\mathbf{y}_i)}$$
$$= \boldsymbol{\Sigma}_{\mathbf{x}_{ij}} \sum_k d_k \Big[(m_{ijk} - a_k)(\mathbf{u_k} + \mathbf{A_k}^T \langle \mathbf{y}_i \rangle) - \langle \mathbf{y}_i^T \rangle \mathbf{v_k} \mathbf{u_k}^T - \mathbf{A_k}^T \langle \mathbf{y}_i \mathbf{y}_i^T \rangle \mathbf{v_k} \Big]. \quad (13)$$

In the above equations, $\langle \mathbf{y}_i \mathbf{y}_i^T \rangle = \boldsymbol{\Sigma}_{\mathbf{y}_i} - \langle \mathbf{y}_i \rangle \langle \mathbf{y}_i^T \rangle$ and similarly for $\langle \mathbf{x}_{ij} \mathbf{x}_{ij}^T \rangle$. Finally, we take derivatives of the lower bound with respect to the model parameters $\mathbf{A}_\mathbf{k}^+$

$$\frac{\partial L}{\partial \mathbf{A}_\mathbf{k}^+} = \frac{\partial}{\partial \mathbf{A}_\mathbf{k}^+} \sum_{i,j} \langle (m_{ijk} - \mathbf{y}_i^{+T} \mathbf{A}_\mathbf{k}^+ \mathbf{x}_{ij}^+)^T \quad (14)$$
$$d_k (m_{ijk} - \mathbf{y}_i^{+T} A_k^+ \mathbf{x}_{ij}^+) \rangle_{q(\mathbf{x}_{ij}), q(\mathbf{y}_i)}$$
$$= \sum_{i,j} \langle \mathbf{y}_i^+ \rangle m_{ijk} \langle \mathbf{x}_{ij}^{+T} \rangle - \sum_{i,j} \langle \mathbf{y}_i^+ \mathbf{y}_i^{+T} \rangle \mathbf{A}_\mathbf{k}^+ \langle \mathbf{x}_{ij}^+ \mathbf{x}_{ij}^{+T} \rangle,$$

where

$$\langle \mathbf{x}_{ij}^{+T} \rangle = \begin{bmatrix} 1 & \langle \mathbf{x}_{ij}^T \rangle \end{bmatrix} \quad (15)$$
$$\langle \mathbf{x}_{ij}^+ \mathbf{x}_{ij}^{+T} \rangle = \begin{bmatrix} 1 & \langle \mathbf{x}_{ij}^T \rangle \\ \langle \mathbf{x}_{ij} \rangle & \langle \mathbf{x}_{ij} \mathbf{x}_{ij}^T \rangle \end{bmatrix} \quad (16)$$
$$(17)$$

and likewise for $\langle \mathbf{y}_i^{+T} \rangle$ and $\langle \mathbf{y}_i^+ \mathbf{y}_i^{+T} \rangle$. Setting the derivative equal to zero and solving for $\mathbf{A}_k^+$ gives us the update formula

$$\mathrm{vec}(\mathbf{A}_k) = \left(\sum_{i,j} \langle \mathbf{x}_{ij}^+ \mathbf{x}_{ij}^{+T} \rangle \otimes \langle \mathbf{y}_i^+ \mathbf{y}_i^{+T} \rangle \right)^{-1} \mathrm{vec} \left(\sum_{i,j} \langle \mathbf{y}_i^+ \rangle m_{ijk} \langle \mathbf{x}_{ij}^{+T} \rangle \right) \quad (18)$$

where $\otimes$ is the Kronecker product and vec is an operator which creates a column vector from a matrix by stacking its columns. Similarly, the update formula for precision d_k can be shown to be

$$d_k^{-1} = \frac{1}{N} \sum_{i,j} \langle (m_{ijk} - \mathbf{y}_i^{+T} \mathbf{A}_\mathbf{K}^+ \mathbf{x}_{ij}^+)^2 \rangle_{q(\mathbf{x}_{ij}), q(\mathbf{y}_i)}$$
$$= \frac{1}{N} \sum_{ij} \Big[m_{ijk}^2 - 2 m_{ijk} \langle \mathbf{y}_i^{+T} \rangle \mathbf{A}_\mathbf{K}^+ \langle \mathbf{x}_{ij}^+ \rangle + \mathrm{tr}(\mathbf{A}_\mathbf{k}^+ \langle \mathbf{x}_{ij}^+ \mathbf{x}_{ij}^{+T} \rangle \mathbf{A}_\mathbf{k}^{+\mathbf{T}} \langle \mathbf{y}_i^+ \mathbf{y}_i^{+T} \rangle) \Big] \quad (19)$$

2.2. Log-likelihood lower bound evaluation

To check for convergence of the training algorithm, the value of the lower bound (8) can be evaluated as

$$\mathcal{L} = -\frac{1}{2} NK \ln 2\pi + \frac{1}{2} N \sum_k \ln d_k$$
$$-\frac{1}{2} \sum_{i,j} \sum_k d_k \Big[m_{ijk}^2 - 2 m_{ijk} \langle \mathbf{y}_i^{+T} \rangle \mathbf{A}_\mathbf{K}^+ \langle \mathbf{x}_{ij}^+ \rangle + \mathrm{tr}(\mathbf{A}_\mathbf{k}^+ \langle \mathbf{x}_{ij}^+ \mathbf{x}_{ij}^{+T} \rangle \mathbf{A}_\mathbf{k}^{+\mathbf{T}} \langle \mathbf{y}_i^+ \mathbf{y}_i^{+T} \rangle) \Big]$$
$$-\frac{1}{2} \sum_{i,j} \mathrm{tr}(\langle \mathbf{x}_{ij} \mathbf{x}_{ij}^T \rangle) - \ln \det \boldsymbol{\Sigma}_{\mathbf{x}_{ij}}$$
$$-\frac{1}{2} \sum_i \mathrm{tr}(\langle \mathbf{y}_i \mathbf{y}_i^T \rangle) - \ln \det \boldsymbol{\Sigma}_{\mathbf{y}_i}, \quad (20)$$

where we have analytically solved the integrals in (8) for the current estimates of functions $\{q(\mathbf{y}_i), q(\mathbf{x}_{ij})\}$.

2.3. Verification score evaluation

As a verification score for a pair of iVectors, we want to evaluate the log-likelihood ratio between two hypotheses: $\mathcal{H}_1$, where both iVectors were generated from the same speaker, and $\mathcal{H}_0$, where the iVectors were generated independently. Since it is intractable to evaluate the exact log-likelihood using our model, we approximate the log-likelihood ratios as $\mathcal{L}_{\mathcal{H}_1} - \mathcal{L}_{\mathcal{H}_0}$, where $\mathcal{L}_{\mathcal{H}_1}$ and $\mathcal{L}_{\mathcal{H}_0}$ are the lower bound approximations to the log-likelihoods for the two hypotheses. Using well-trained model parameters, $\mathcal{L}_{\mathcal{H}_1}$ can be evaluated using equation (20) in the same way as we did for training data, where the two iVectors from the verification trial are treated as if they were the only two training examples coming from the same speaker (i.e., they would be given speaker and recording indices $\mathbf{m}_{11}$ and $\mathbf{m}_{12}$). On the contrary, to evaluate $\mathcal{L}_{\mathcal{H}_0}$, the iVectors will be treated as if they were from two different speakers (i.e., the indices would be $\mathbf{m}_{11}$ and $\mathbf{m}_{21}$). Similarly, the likelihoods can be constructed for a trial consisting of more than two iVectors (e.g., multi-session training).

To obtain a good approximation to the true likelihoods, the lower bounds must still be maximized with respect to functions $\{q(\mathbf{y}_i), q(\mathbf{x}_{ij})\}$. Therefore, for each verification trial, several iterations of updates (10)-(13) must be performed before evaluating each lower bound $\mathcal{L}_{\mathcal{H}_1}$ and $\mathcal{L}_{\mathcal{H}_0}$.

3. Experimental Setup

The speech features used in our experiments are 19 MFCCs and energy, augmented with deltas and double deltas. A universal background model (UBM) with 2048 diagonal covariance Gaussian components is trained in a gender-independent fashion on NIST SRE 04 and 05 telephone data. Starting from this UBM, an iVector extractor is trained on NIST SRE 04,05,06, Switchboard, and Fisher data for iVector dimensionality of 400. The dimensionality of iVectors is further reduced using LDA transformation, which is estimated on NIST SRE 04, 05, 06 data. LDA reduction to only 50 dimensions is used in these preliminary experiments (see the discussion in Section 5).

The PLDA model and the proposed BLFA models are trained on iVectors extracted from NIST SRE 04, 05, 06 data and also from Switchboard and Fisher data, where multiple sessions are available per speaker. For the total of 3296 speakers,

53233 sessions from telephone and microphone channels were used for PLDA or BLFA training.

4. Results

Table 1 presents results for various configurations of the BLFA model. The verification performance for different models and configurations is reported in terms of detection cost function (DCF) and equal error rate (EER) on NIST SRE10 extended conditions 1, 3 and 5 (int-int same mic., int-tel and tel-tel) [9]. As we have already noted, the problem with the BLFA model is the large number of parameters in the matrices $\mathbf{A}_k$. To control the number of trainable parameters in the model, we retrain only c first columns of each matrix and force the remaining columns to be zero. Note that if the matrices $\mathbf{A}_k$ are set to zero (i.e., c=0), we recover the standard PLDA model. Further, to avoid overfitting, we have simplified the training procedure as follows: At first, only PLDA model (i.e. c=0) is trained using the described variational Bayes update formulae. The matrix $\mathbf{V}$ and the distributions $q(\mathbf{y}_i)$ for all training speakers are fixed and only the remaining parameters and $q(\mathbf{x}_{ij})$ distributions are updated in the following BLFA model training. We found this strategy to be necessary for obtaining good verification performance.

Table 1 shows results for different settings of c. As can be seen, very good performance is obtained for $c = 10$. With this configuration, the BLFA model always performs better than the PLDA (BLFA $c = 0$) model, which confirms the increased modeling power of the proposed model. The improvement is especially significant at the DCF operating point important in NIST evaluations.

Although the BLFA model becomes equivalent to the PLDA model for $c = 0$, we still use the approximate variational inference to train the model parameters and to evaluate the log-likelihood verification score. Therefore, we also include results obtained with the PLDA model trained using the usual EM algorithm, where exact inference is used for both model training and score evaluation. As expected, in many cases, EM-trained PLDA (the last line in the table) outperforms PLDA trained using the approximate inference (the first line). Nevertheless, BLFA with $c = 10$ also always outperforms the EM-trained PLDA.

	sre10c01		sre10c03		sre10c05	
Model	nDCF	EER	nDCF	EER	nDCF	EER
BLFA $c = 0$	0.62	2.37	0.87	4.87	0.72	3.82
BLFA $c = 10$	**0.44**	**2.28**	**0.67**	**4.81**	**0.55**	3.48
BLFA $c = 20$	**0.44**	2.32	0.71	4.82	0.58	3.40
BLFA $c = 50$	**0.44**	2.42	0.70	4.89	0.58	**3.36**
EM PLDA	0.46	2.32	0.73	5.26	0.61	3.92

Table 1: *Comparison of the proposed model BLFA model and PLDA model on three SRE10 NIST evaluation extended conditions. Different values of c corresponds to the number of nonzero columns in the matrices $\mathbf{A}_k$. Note that BLFA with $c = 0$ corresponds to the standard PLDA, which is, however, trained using variational Bayes rather than EM algorithm.*

5. Discussion and Future work

In this study, we have proposed and tested a new extension of the PLDA model, where within-class (channel) variability is modeled as a function of the class (speaker) location in the feature (iVector) space. Because of the mathematical tractability, we have considered only a simple bilinear relation between latent variables describing speaker and channel. However, in the future, we may consider more complicated but possibly more compactly represented relations.

We have shown the potential of the proposed model on verification experiments, where the new model outperforms the state-of-the-art PLDA model. However, the presented results should by no means be interpreted as the state-of-the-art results obtained on this task. In these initial experiments, we used iVectors reduced by LDA to only 50 dimensions, while optimal performance is usually obtained with higher dimensionality (around 200). This configuration was used to allow fast turn-around of the initial experiments, and also to overcome the memory limitations given by our current implementation. Currently, the bottleneck is in the implementation of the update formula (18), where we simply construct and invert the matrix given by the Kronecker product, which becomes unfeasibly large for even reasonable dimensionalities of the latent variables. In future, we plan to replace this formula by an iterative and less memory-intensive update. Also, for the presented results, no iVector length normalization [7] was applied, which is known to significantly improve performance in the case of the standard PLDA model. Nevertheless, we believe that even this preliminary work can inspire other researchers and that it can initiate interesting discussions in the speaker recognition community.

We are currently testing the method for more state-of-the-art-like configurations, where we have already seen first encouraging results still outperforming the standard PLDA model. It also will be interesting to fuse scores produced by both PLDA and the BLFA model. The additional parameters introduced into the BFFA model should increase its modeling power compared to PLDA. On the other hand, it uses an approximate variational inference, while exact inference can be used in the case of PLDA. Therefore, we can expect a certain degree of complementarity of BLFA and PLDA models.

6. References

[1] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front–end factor analysis for speaker verification," in *IEEE Trans. on Audio, Speech and Lang. Process.*, 2010.

[2] S. J. D. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *11th International Conference on Computer Vision*, 2007, pp. 1–8.

[3] P. Kenny, "Bayesian speaker verification with heavy–tailed priors," in *keynote presentation, Odyssey 2010*, 2010.

[4] N. Brümmer and E. de Villiers, "The speaker partitioning problem," in *Proc. of Odyssey 2010*, 2010, pp. 194–201.

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

[6] J. Villalba and N. Brummer, "Towards fully Bayesian speaker recognition: Integrating out the between-speaker covariance," in *Proc. of Interspeech 2011*, August 2011.

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

[8] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society, Series B*, vol. 61, pp. 611–622, 1999.

[9] NIST, "The NIST year 2010 speaker recognition evaluation plans," http://www.itl.nist.gov/iad/mig/tests/sre.