A Scalable Formulation of Probabilistic Linear Discriminant Analysis: Applied to Face Recognition

Laurent El Shafey, Chris McCool, Roy Wallace, and Sébastien Marcel

***Abstract*—In this paper, we present a scalable and exact solution for probabilistic linear discriminant analysis (PLDA). PLDA is a probabilistic model that has been shown to provide state-of-the-art performance for both face and speaker recognition. However, it has one major drawback: At training time estimating the latent variables requires the inversion and storage of a matrix whose size grows quadratically with the number of samples for the identity (class). To date, two approaches have been taken to deal with this problem, to 1) use an exact solution that calculates this large matrix and is obviously not scalable with the number of samples or 2) derive a variational approximation to the problem.**

We present a scalable derivation which is theoretically equivalent to the previous nonscalable solution and thus obviates the need for a variational approximation. Experimentally, we demonstrate the efficacy of our approach in two ways. First, on Labeled Faces in the Wild (LFW), we illustrate the equivalence of our scalable implementation with previously published work. Second, on the large Multi-PIE database, we illustrate the gain in performance when using more training samples per identity (class), which is made possible by the proposed scalable formulation of PLDA.

***Index Terms*—PLDA, probablistic model, expectation maximization, face verification**

I. Introduction

In recent years, a technique known as probabilistic linear discriminant analysis (PLDA) has been proposed [1] for face recognition. This probabilistic technique has since been applied to both face [2] and speaker recognition [3] and achieved state-of-the-art performance in both fields.

Despite PLDA's state-of-the-art performance in multiple domains and for multiple tasks, it has a major limiting factor for its application to large datasets. The calculation of the posteriors, for training the model, implies the inversion of a matrix whose size grows quadratically with the number of samples per class, J_i. In [3], a variational approximation was proposed to address this issue. However, the quality of such an approximation is subject to the validity of the questionable assumption (see [3, Section 3]) that the posterior variables are independent. A similar problem occurs when calculating the likelihood. For this, Kenny [3] proposed a further approximation by using a lower bound of the likelihood and noting that a change of variables could be performed to diagonalize the covariance matrix. To do this efficiently, it was noted that the matrix needed to perform the change of variable could be computed offline. However, it had to be loaded and used whenever the likelihood was calculated.

In this paper, we show that, for PLDA with a Gaussian prior, the calculation of the posteriors (for training) and the calculation of the likelihood have exact closed form solutions that are scalable. This exact closed form and scalable solution, which has thus far not been presented anywhere else, obviates the need for the approximations proposed by [3] and is the main contribution of this paper. This contribution provides three significant advantages over prior work. First, we provide a scalable way to derive the posteriors and thus a scalable way to perform training that does not involve any approximations. Second, we detail an exact and scalable way to calculate the joint probability of the samples which has never been presented before, neither in [2] nor [3]. Third, and finally, we show that enrollment of a client (storing the information for a model) reduces to calculating a single low-dimensional feature vector rather than retaining multiple high-dimensional feature vectors. This further improves the efficiency and scalability of the model.

We briefly present PLDA and the origin of the problems for scalability for both training and computing the likelihood in Section II. In Section III, we describe how, by introducing a well-defined change of variable, it is possible to derive exact closed form solutions for the posteriors and the likelihood that are scalable. Section IV discusses the complexity of previous and proposed approaches. Finally, in Section V, we demonstrate experimentally that, for the task of face authentication, it is advantageous to train the PLDA system with many samples per class, which is only made possible by the proposed scalable formulation.

II. Background

The PLDA model proposed by Prince and Elder [1] has been shown to provide state-of-the-art performance for both face recognition [4], [2] and speaker recognition [3]. However, the model has one major problem. It cannot be easily applied to large datasets. This was highlighted by Kenny [3], for the task of speaker recognition.

The PLDA model [1] assumes that the jth image of the ith client can be described by the following process:

$$\boldsymbol{x}_{i,j} = \boldsymbol{\mu} + \boldsymbol{F}\boldsymbol{h}_i + \boldsymbol{G}\boldsymbol{w}_{i,j} + \boldsymbol{\epsilon}_{i,j}. \quad (1)$$

In this process, the input signal $\boldsymbol{x}_{i,j}$, of dimensionality D_x, is considered to consist of: 1) the identity part given by $\boldsymbol{\mu}+\boldsymbol{F}\boldsymbol{h}_i$ and 2) the noise component given by $\boldsymbol{G}\boldsymbol{w}_{i,j} + \boldsymbol{\epsilon}_{i,j}$. The

L. El Shafey is with Idiap Research Institute and Ecole Polytechnique Fédérale de Lausanne, Switzerland e-mail: laurent.el-shafey@idiap.ch

C. McCool, R. Wallace and S. Marcel are with Idiap Research Institute, Martigny, Switzerland e-mail: {christopher.mccool,roy.wallace,sebastien.marcel}@idiap.ch

The research leading to these results has received funding from the European Communitys Seventh Framework Programme (FP7) under grant agreements 238803 (BBfor2) and 257289 (TABULA RASA).

matrices $\boldsymbol{F}$ and $\boldsymbol{G}$ are subspaces that contain the bases for the between-class variation and within-class variation respectively, these subspaces are of size (D_x, D_F) and (D_x, D_G), respectively. Correspondingly, $\boldsymbol{h}_i$ and $\boldsymbol{w}_{i,j}$ represent the position in these subspaces for $\boldsymbol{x}_{i,j}$ and are of size D_F and D_G, respectively. Finally, the residual $\boldsymbol{\epsilon}_{i,j}$, is defined to be Gaussian with zero mean and diagonal covariance $\boldsymbol{\Sigma}$.

The process above can be described in terms of a conditional probability

$$Pr\left(\boldsymbol{x}_{i,j}|\boldsymbol{h}_i, \boldsymbol{w}_{i,j}, \boldsymbol{\Theta}\right) = \mathcal{N}\left[\boldsymbol{\mu} + \boldsymbol{F}\boldsymbol{h}_i + \boldsymbol{G}\boldsymbol{w}_{i,j}, \boldsymbol{\Sigma}\right], \quad (2)$$

and prior probabilities ($\boldsymbol{I}$ being the identity matrix)

$$Pr\left(\boldsymbol{h}_i\right) = \mathcal{N}\left[0, \boldsymbol{I}\right], \quad (3)$$

$$Pr\left(\boldsymbol{w}_{i,j}\right) = \mathcal{N}\left[0, \boldsymbol{I}\right], \quad (4)$$

where the parameters of the model are $\boldsymbol{\Theta} = [\boldsymbol{\mu}, \boldsymbol{F}, \boldsymbol{G}, \boldsymbol{\Sigma}]$. Equations (3) and (4) define the priors on the latent variables, $\boldsymbol{h}_i$ and $\boldsymbol{w}_{i,j}$, to be Gaussian. The equations above can be written in a more compact form by setting $\boldsymbol{A} = [\boldsymbol{F}, \boldsymbol{G}]$ and

$$\boldsymbol{y}_{i,j} = [\boldsymbol{h}_i^T, \boldsymbol{w}_{i,j}^T]^T. \quad (5)$$

This would give us

$$\boldsymbol{x}_{i,j} = \boldsymbol{\mu} + \boldsymbol{A}\boldsymbol{y}_{i,j} + \boldsymbol{\epsilon}_{i,j}, \quad (6)$$

and

$$Pr\left(\boldsymbol{x}_{i,j}|\boldsymbol{y}_{i,j}, \boldsymbol{\Theta}\right) = \mathcal{N}\left[\boldsymbol{\mu} + \boldsymbol{A}\boldsymbol{y}_{i,j}, \boldsymbol{\Sigma}\right], \quad (7)$$

$$Pr\left(\boldsymbol{y}_{i,j}\right) = \mathcal{N}\left[0, \boldsymbol{I}\right]. \quad (8)$$

We can extend the above formulation to handle multiple observations. For instance, if we are given $J_i = 2$ observations for identity i we would set

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{F}, \boldsymbol{G}, 0 \\ \boldsymbol{F}, 0, \boldsymbol{G} \end{bmatrix}. \quad (9)$$

Consequently, we would write that $\tilde{\boldsymbol{x}}_i = [\bar{\boldsymbol{x}}_{i,1}^T, \bar{\boldsymbol{x}}_{i,2}^T]^T$, $\tilde{\boldsymbol{\epsilon}}_i = [\boldsymbol{\epsilon}_{i,1}^T, \boldsymbol{\epsilon}_{i,2}^T]^T$, $\tilde{\boldsymbol{w}}_i = [\boldsymbol{w}_{i,1}^T, \boldsymbol{w}_{i,2}^T]^T$, $\tilde{\boldsymbol{y}}_i = [\boldsymbol{h}_i^T, \boldsymbol{w}_{i,1}^T, \boldsymbol{w}_{i,2}^T]^T$,

$$\tilde{\boldsymbol{\Sigma}} = \begin{bmatrix} \boldsymbol{\Sigma}, 0 \\ 0, \boldsymbol{\Sigma} \end{bmatrix}, \quad (10)$$

where

$$\bar{\boldsymbol{x}}_{i,j} = \boldsymbol{x}_{i,j} - \boldsymbol{\mu}, \quad (11)$$

the tilde symbol ˜ indicating that the size of a variable (or a matrix) depends on the number of samples J_i for the class i.

This notation makes it explicit that we tie all of the J_i observations for identity i to have the same latent identity variable $\boldsymbol{h}_i$ but to have different latent session, or noise, variables $\boldsymbol{w}_{i,j}$; with such a formulation we would drop the reference to $\boldsymbol{\mu}$, the mean of the data, in the Gaussian as it has already been subtracted from the samples in (11).

For the general case of a class i with J_i samples, and keeping the same notation, the model could be rewritten

$$\tilde{\boldsymbol{x}}_i = \tilde{\boldsymbol{A}}\tilde{\boldsymbol{y}}_i + \tilde{\boldsymbol{\epsilon}}_i. \quad (12)$$

There are two main tasks that we wish to accomplish with this probabilistic model. The first is to train the model and this is achieved using an expectation-maximization (EM) algorithm. The second is to use the trained model to perform recognition and it was shown [1] that central to this problem is to calculate the likelihood that a set of observations, $[\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \ldots, \boldsymbol{x}_{i,J_i}]$, share the same latent identity variable $\boldsymbol{h}_i$. Below we provide some more details on each of these tasks.

A. Training the PLDA Model

To train the PLDA model an EM algorithm is used [1]. All of the M-Steps are provided on a per sample basis once the latent variables have been estimated. It is this estimation of the latent variables, corresponding to the E-Step, that presents the difficulties in making PLDA scalable. In the E-Step, we need to calculate the first-order and second-order moments of the latent variables, we reproduce the equations as follows:

$$E\left[\tilde{\boldsymbol{y}}_i|\tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right] = \left(\tilde{\boldsymbol{I}} + \tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{A}}\right)^{-1}\tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{\Sigma}}^{-1}\left(\tilde{\boldsymbol{x}}_i\right), \quad (13)$$

$$E\left[\tilde{\boldsymbol{y}}_i\tilde{\boldsymbol{y}}_i^T|\tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right] = \left(\tilde{\boldsymbol{I}} + \tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{A}}\right)^{-1} + E\left[\tilde{\boldsymbol{y}}_i|\tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right]E\left[\tilde{\boldsymbol{y}}_i|\tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right]^T. \quad (14)$$

From the above equations, it is obvious that the problem for the E-Step is how to cope with the matrix $\left(\tilde{\boldsymbol{I}} + \tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{A}}\right)^{-1}$ efficiently as it has to be recomputed for each iteration of EM. This matrix is indeed of size $(D_F + J_iD_G, D_F + J_iD_G)$, and has to be used in calculations as well as stored. A solution to this problem was proposed by Kenny [3] when applying PLDA to speaker recognition. Kenny's solution was to apply a variational approximation for this inference problem; however, this approximation relies on a factorization which assumes that the posterior variables are independent and whose quality with respect to the exact solution has not been demonstrated. Once the PLDA model has been trained, it can be used to perform various tasks, which all rely on likelihood calculations.

B. Likelihood for PLDA

It was shown in [2] that the likelihood can be used to solve several problems ranging from identification and authentication through to registration. One wishes to calculate the likelihood that a set of samples, $\tilde{\boldsymbol{x}}_i$, share the same latent identity variable. This can be calculated in a probabilistic way by integrating out over all of the latent variables. To do this, we tie together the latent identity variable, $\boldsymbol{h}_i$, of the samples that we consider to have the same identity and then consider each observation, $\boldsymbol{x}_{i,j}$, to have a separate latent session variable, $\boldsymbol{w}_{i,j}$. We would then integrate over $\boldsymbol{h}_i$ and all of the individual $\boldsymbol{w}_{i,j}$s. For the case of $J_i = 2$ this equates to

$$Pr\left(\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}\right) = \int\left[\int Pr\left(\boldsymbol{x}_{i,1}, |\boldsymbol{h}_i, \boldsymbol{w}_{i,1}\right)Pr\left(\boldsymbol{w}_{i,1}\right)d\boldsymbol{w}_{i,1} \int Pr\left(\boldsymbol{x}_{i,2}, |\boldsymbol{h}_i, \boldsymbol{w}_{i,2}\right)Pr\left(\boldsymbol{w}_{i,2}\right)d\boldsymbol{w}_{i,2}\right]Pr\left(\boldsymbol{h}_i\right)d\boldsymbol{h}_i, \quad (15)$$

where for brevity we have dropped the reference to $\boldsymbol{\Theta}$.

The above problem can be written as [1][1],

$$Pr\left(\tilde{\boldsymbol{x}}_i|\boldsymbol{\Theta}\right) = \mathcal{N}\left[\boldsymbol{0}, \tilde{\boldsymbol{\Sigma}} + \tilde{\boldsymbol{A}}\tilde{\boldsymbol{A}}^T\right]. \tag{16}$$

Equivalently, we can calculate the log-likelihood and write

$$\begin{aligned}\ln\left[Pr\left(\tilde{\boldsymbol{x}}_i|\boldsymbol{\Theta}\right)\right] = &-\frac{J_i D_x}{2}\ln\left[2\pi\right] \\ &-\frac{1}{2}\ln\left[\det\left(\tilde{\boldsymbol{\Sigma}} + \tilde{\boldsymbol{A}}\tilde{\boldsymbol{A}}^T\right)\right] \\ &-\frac{1}{2}\tilde{\boldsymbol{x}}_i^T\left(\tilde{\boldsymbol{\Sigma}} + \tilde{\boldsymbol{A}}\tilde{\boldsymbol{A}}^T\right)^{-1}\tilde{\boldsymbol{x}}_i. \end{aligned}\tag{17}$$

This has three terms with various computational complexity, the first term being a simple constant factor depending upon the number of samples J_i. In contrast, the second term, $\ln\left[\det\left(\tilde{\boldsymbol{\Sigma}} + \tilde{\boldsymbol{A}}\tilde{\boldsymbol{A}}^T\right)\right]$, requires the computation of the determinant of a matrix which grows quadratically with the number of samples J_i[2]. Finally, the third term, $\tilde{\boldsymbol{x}}_i^T\left(\tilde{\boldsymbol{\Sigma}} + \tilde{\boldsymbol{A}}\tilde{\boldsymbol{A}}^T\right)^{-1}\tilde{\boldsymbol{x}}_i$, requires the inversion of the same large matrix and is also computationally demanding. An optimization to this quadratic term was suggested in [4] (see Section 3.2.3) by reformulating it as follows:

$$\tilde{\boldsymbol{x}}_i^T\left(\tilde{\boldsymbol{\Sigma}} + \tilde{\boldsymbol{A}}\tilde{\boldsymbol{A}}^T\right)^{-1}\tilde{\boldsymbol{x}}_i = \tilde{\boldsymbol{x}}_i^T\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{x}}_i^T\tilde{\boldsymbol{\Gamma}}\tilde{\boldsymbol{\Gamma}}^T\tilde{\boldsymbol{x}}_i, \tag{18}$$

where $\tilde{\boldsymbol{\Gamma}} = \tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{A}}\left(\tilde{\boldsymbol{I}} + \tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{A}}\right)^{-\frac{1}{2}}$. The first part of this new problem is easy to compute as $\tilde{\boldsymbol{\Sigma}}$ is diagonal. However, the second term requires us to calculate the square root of the large inverted matrix $\left(\tilde{\boldsymbol{I}} + \tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{A}}\right)^{-1}$; although its computation might be performed offline, this matrix grows quadratically in size with the number of samples.

Two alternatives have been proposed to efficiently calculate the likelihood. Li *et al.* [2] suggested using the predictive distribution rather than the joint distribution (as originally proposed in [1]); however, their solution does not explicitly solve the issue of multiple probe and enrollment samples. In [3], another formulation for computing the likelihood was given where the lower bound of $\ln\left[Pr\left(\tilde{\boldsymbol{x}}_i|\boldsymbol{\Theta}\right)\right]$ is used. In this paper, we will show that, for the case of a Gaussian prior, a direct, exact, and scalable solution can be derived and thus obviates the need for storing the large inverted matrix, using predictive distributions or using the lower bound.

Before we continue we note that central to providing a scalable solution for this problem and the problem of training, is finding an efficient way to use the matrix $\left(\tilde{\boldsymbol{I}} + \tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{A}}\right)^{-1}$. This matrix is of size $(D_F + J_i D_G, D_F + J_i D_G)$ and grows quadratically with the number of samples, thus limiting scalability. However, this matrix has a well defined structure which can be exploited. In the next section, we will show how an exact and scalable solution can be found for these problems.

[1]We have dropped the reference to $\boldsymbol{\mu}$ in this Gaussian model because we have subtracted it from the observations in (11).

[2]Li *et al.*'s [4], [2] implementation suggests that this can be decomposed into a series of simpler determinants by exploiting the structure of the PLDA model. In fact, we will show that it is possible to exploit this structure to reduce the complexity of most of the computationally demanding parts of both the training and the likelihood computation.

III. Proposed Solution

The overall idea of our approach consists of exploiting the structure of this probabilistic model by diagonalizing the model and then replacing full matrix inversions by a set of block matrix inversions which are less computationally demanding. This scalable version of training is in contrast to previous solutions such as the nonscalable solution outlined in [2] and the nonexact solution presented in [3]. Also, the calculation of the likelihood is quite different to the solutions proposed in [2] and [3].

Considering the training procedure, the solution given in [2] suggests that the computation of the moments of the tied latent variables $\tilde{\boldsymbol{y}}_i$ requires the matrix inversion $\left(\tilde{\boldsymbol{I}} + \tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{A}}\right)^{-1}$. However, the following properties hold for the PLDA model and are described by the structure of the $\tilde{\boldsymbol{A}}$ matrix. First, the samples $\boldsymbol{x}_{i,j}$s of a given class i share a common term $\boldsymbol{F}\boldsymbol{h}_i$, but have separate confounding factors $\boldsymbol{G}\boldsymbol{w}_{i,j}$s. Hence, the sum of the $\boldsymbol{x}_{i,j}$s for class i should intuitively be sufficient to estimate the latent variable $\boldsymbol{h}_i$. Second, each sample $\boldsymbol{x}_{i,j}$ is associated with a separate latent variable $\boldsymbol{w}_{i,j}$. In addition, the $\boldsymbol{w}_{i,j}$s are independent Gaussian samples. This implies that their independence is preserved if we consider any orthogonal mixtures of them.

In the following, we show that a simple change of variable allows us to diagonalize the PLDA model. In combination with simple linear algebra operations, this leads to a scalable formulation for both the training and the likelihood computation.

A. Change of variable

Let $\tilde{\boldsymbol{U}}$ be any $J_i \times J_i$ orthogonal matrix whose first row $\tilde{\boldsymbol{u}}_0$ is $[1, \cdots, 1]/\sqrt{J_i}$. The remaining rows can be anything with the constraint of $\tilde{\boldsymbol{U}}$ to be orthogonal. For instance, let the remaining rows $\tilde{\boldsymbol{u}}_j$ of $\tilde{\boldsymbol{U}}$ be (for $j \in \{1, \cdots, J_i - 1\}$)

$$\tilde{\boldsymbol{u}}_j = \left[\underbrace{\frac{1}{\sqrt{j(j+1)}}, \cdots, \frac{1}{\sqrt{j(j+1)}}}_{j \text{ identical positive terms}}, \frac{-j}{\sqrt{j(j+1)}}, 0, \cdots, 0\right], \tag{19}$$

such that the row $\tilde{\boldsymbol{u}}_j$ has j identical positive terms followed by one negative term, sums to zero, and is of unit length.

Then, tensoring $\tilde{\boldsymbol{U}}$ with the identity matrix in the $\boldsymbol{x}_{i,j}$-space (also known as the Kronecker product) and multiplying by $\tilde{\boldsymbol{x}}_i$ leads to new variables $\mathring{\tilde{\boldsymbol{x}}}_i = \left(\tilde{\boldsymbol{U}} \otimes \boldsymbol{I}_{D_x}\right)\tilde{\boldsymbol{x}}_i$ (In the following, the circle symbol $^\circ$ is used for variables transformed by such a change of variable.) In particular, the upper part of $\mathring{\tilde{\boldsymbol{x}}}_i$ corresponds to a normalized sum of the $\bar{\boldsymbol{x}}_{i,j}$s, which is useful to estimate the identity factor $\boldsymbol{h}_i$, as highlighted at the beginning of this section III. This transform could also be applied to the $\tilde{\boldsymbol{\epsilon}}_i$ variables leading to $\mathring{\tilde{\boldsymbol{\epsilon}}}_i$, and the following PLDA model:

$$\mathring{\tilde{\boldsymbol{x}}}_i = \left(\tilde{\boldsymbol{U}} \otimes \boldsymbol{I}_{D_x}\right)\tilde{\boldsymbol{A}}\tilde{\boldsymbol{y}}_i + \mathring{\tilde{\boldsymbol{\epsilon}}}_i. \tag{20}$$

In this case, the independence of the new variables $\mathring{\tilde{\boldsymbol{\epsilon}}}_i$ is preserved as the mixing matrix $\tilde{\boldsymbol{U}}$ is orthogonal.

Similarly, the change of variable could be applied to $\boldsymbol{w}_{i,j}$ by tensoring $\tilde{\boldsymbol{U}}$ with the identity matrix in the $\boldsymbol{w}_{i,j}$-space. Assuming that $\boldsymbol{h}_i$ is not updated by this change of variable, the transform leading to $\mathring{\tilde{\boldsymbol{y}}}_i$ would be as follows:

$$\mathring{\tilde{\boldsymbol{y}}}_i = \begin{bmatrix} \boldsymbol{I}_{D_F} & 0 \\ 0 & \tilde{\boldsymbol{U}} \otimes \boldsymbol{I}_{D_G} \end{bmatrix} \tilde{\boldsymbol{y}}_i = \tilde{\boldsymbol{V}} \tilde{\boldsymbol{y}}_i. \tag{21}$$

It is then straightforward to show that the PLDA model can be rewritten with the previously introduced variables as

$$\mathring{\tilde{\boldsymbol{x}}}_i = \mathring{\tilde{\boldsymbol{A}}} \mathring{\tilde{\boldsymbol{y}}}_i + \mathring{\tilde{\boldsymbol{\epsilon}}}_i, \tag{22}$$

with $\mathring{\tilde{\boldsymbol{A}}} = \left(\tilde{\boldsymbol{U}} \otimes \boldsymbol{I}_{D_x}\right) \tilde{\boldsymbol{A}} \tilde{\boldsymbol{V}}^{-1}$ and $\tilde{\boldsymbol{V}}$ being easy to invert as

$$\tilde{\boldsymbol{V}}^{-1} = \begin{bmatrix} \boldsymbol{I}_{D_F} & 0 \\ 0 & \tilde{\boldsymbol{U}}^T \otimes \boldsymbol{I}_{D_G} \end{bmatrix}. \tag{23}$$

Interestingly, $\mathring{\tilde{\boldsymbol{A}}}$ is sparser than the original matrix $\tilde{\boldsymbol{A}}$ and is block diagonal, which justifies the choice of the previous change of variable

$$\mathring{\tilde{\boldsymbol{A}}} = \begin{bmatrix} \sqrt{J_i}\boldsymbol{F} & \boldsymbol{G} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \boldsymbol{G} \end{bmatrix}. \tag{24}$$

In the following, we will show that this will indeed lead, for both the training and the likelihood computation, to matrix operations such as determinant computation and matrix inversion, which can be performed very efficiently on a block basis, significantly reducing the complexity of this probabilistic approach.

Finally, the variables of our proposed solution have the following Gaussian distributions:

$$Pr\left(\mathring{\tilde{\boldsymbol{x}}}_i | \mathring{\tilde{\boldsymbol{y}}}_i, \boldsymbol{\Theta}\right) = \mathcal{N}\left[\mathring{\tilde{\boldsymbol{A}}} \mathring{\tilde{\boldsymbol{y}}}_i, \tilde{\boldsymbol{\Sigma}}\right], \tag{25}$$

$$Pr\left(\mathring{\tilde{\boldsymbol{y}}}_i\right) = \mathcal{N}\left[0, \tilde{\boldsymbol{I}}\right] \text{ and } Pr\left(\mathring{\tilde{\boldsymbol{\epsilon}}}_i\right) = \mathcal{N}\left[0, \tilde{\boldsymbol{\Sigma}}\right]. \tag{26}$$

B. Scalable training

Our aim is to be able to train the PLDA model so that it is scalable with respect to the number of training samples. To train the PLDA model, an EM algorithm is used; however, the E-Step of this algorithm has a bottleneck. With the original PLDA formulation, this leads to the computation of the matrix $\left(\tilde{\boldsymbol{I}} + \tilde{\boldsymbol{A}}^T \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{A}}\right)^{-1}$, which grows quadratically with the number of samples, J_i. In contrast, the proposed diagonalization leads to a scalable E-Step formulation. Using Bayes rule, the probability distribution of the latent variables of the transformed PLDA model could be written

$$Pr\left(\mathring{\tilde{\boldsymbol{y}}}_i | \mathring{\tilde{\boldsymbol{x}}}_i, \boldsymbol{\Theta}\right) = \mathcal{N}\left[\mathring{\tilde{\boldsymbol{\mathcal{P}}}}^{-1}\left(\mathring{\tilde{\boldsymbol{A}}}^T \tilde{\boldsymbol{\Sigma}}^{-1} \mathring{\tilde{\boldsymbol{x}}}_i\right), \mathring{\tilde{\boldsymbol{\mathcal{P}}}}^{-1}\right], \tag{27}$$

with $\mathring{\tilde{\boldsymbol{\mathcal{P}}}} = \tilde{\boldsymbol{I}} + \mathring{\tilde{\boldsymbol{A}}}^T \tilde{\boldsymbol{\Sigma}}^{-1} \mathring{\tilde{\boldsymbol{A}}}$ being block diagonal. The upper left block being

$$\boldsymbol{\mathcal{P}}_0 = \begin{bmatrix} \boldsymbol{I}_{D_F} + J_i \boldsymbol{F}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{F} & \sqrt{J_i} \boldsymbol{F}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{G} \\ \sqrt{J_i} \boldsymbol{G}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{F} & \boldsymbol{I}_{D_G} + \boldsymbol{G}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{G} \end{bmatrix}, \tag{28}$$

and the remaining $J_i - 2$ blocks being equal to

$$\boldsymbol{\mathcal{P}}_1 = \boldsymbol{\mathcal{G}}^{-1}, \tag{29}$$

with the (symmetric) matrix $\boldsymbol{\mathcal{G}}$ being defined by

$$\boldsymbol{\mathcal{G}} = \left(\boldsymbol{I}_{D_G} + \boldsymbol{G}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{G}\right)^{-1}. \tag{30}$$

a) Estimating the first-order moment of the latent variables: This is given by the mean of the Gaussian distribution in (27), which is $\mathring{\tilde{\boldsymbol{\mathcal{P}}}}^{-1}\left(\mathring{\tilde{\boldsymbol{A}}}^T \tilde{\boldsymbol{\Sigma}}^{-1} \mathring{\tilde{\boldsymbol{x}}}_i\right)$. $\mathring{\tilde{\boldsymbol{\mathcal{P}}}}$ is diagonal by blocks and can be efficiently inverted. Then, as $\boldsymbol{h}_i$ is not affected by the change of variable (21), it corresponds to the upper sub-vector of $\mathring{\tilde{\boldsymbol{y}}}_i$. Furthermore, the first-order moment is directly obtained from the first rows, such that

$$E[\boldsymbol{h}_i | \tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}] = \boldsymbol{\mathcal{F}}_{J_i} \sum_j \boldsymbol{F}^T \boldsymbol{\mathcal{S}} \bar{\boldsymbol{x}}_{i,j}, \tag{31}$$

where for readability the following (symmetric) matrices have been defined:

$$\boldsymbol{\mathcal{S}} = \left(\boldsymbol{\Sigma} + \boldsymbol{G}\boldsymbol{G}^T\right)^{-1} = \boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1} \boldsymbol{G} \boldsymbol{\mathcal{G}} \boldsymbol{G}^T \boldsymbol{\Sigma}^{-1}, \tag{32}$$

$$\boldsymbol{\mathcal{F}}_{J_i} = \left(\boldsymbol{I}_{D_F} + J_i \boldsymbol{F}^T \boldsymbol{\mathcal{S}} \boldsymbol{F}\right)^{-1}. \tag{33}$$

In addition, and after reverting the change of variable, the $\boldsymbol{w}_{i,j}$s might be expressed as

$$E[\boldsymbol{w}_{i,j} | \tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}] = \boldsymbol{\mathcal{G}} \boldsymbol{G}^T \boldsymbol{\Sigma}^{-1} \left(\bar{\boldsymbol{x}}_{i,j} - \boldsymbol{F} E[\boldsymbol{h}_i | \tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}]\right). \tag{34}$$

b) Estimating the second-order moment of the latent variables: Calculating the expected value of the second-order moment of the latent variables, (14), in a scalable manner is more difficult.

This second order-moment can be expressed as the sum of both the square of the mean and the variance. The mean has already been expressed in (31) and (34). On the other hand, after reverting the change of variable, the variance is given by

$$\tilde{\boldsymbol{\mathfrak{P}}}^{-1} = \tilde{\boldsymbol{V}}^{-1} \mathring{\tilde{\boldsymbol{\mathcal{P}}}}^{-1} \tilde{\boldsymbol{V}}. \tag{35}$$

However, the variance of the latent variables is only ever used on a per sample basis. This can be seen by examining (28) of [1]. We can therefore simplify the problem into computing the quantity $E\left[\boldsymbol{y}_{i,j} \boldsymbol{y}_{i,j}^T | \tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right]$ (using the variable $\boldsymbol{y}_{i,j}$ defined by (5)), which consists of a few blocks of the above matrix $\tilde{\boldsymbol{\mathfrak{P}}}^{-1}$.

Furthermore, the variance of the latent variables on a per sample basis is given by

$$Var\left[\boldsymbol{y}_{i,j} | \tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right] = \begin{bmatrix} \boldsymbol{\mathcal{F}}_{J_i} & \boldsymbol{\mathcal{H}}^T \\ \boldsymbol{\mathcal{H}} & \left(\boldsymbol{I}_{D_G} - \boldsymbol{\mathcal{H}} \boldsymbol{F}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{G}\right) \boldsymbol{\mathcal{G}} \end{bmatrix}, \tag{36}$$

with

$$\boldsymbol{\mathcal{H}} = -\boldsymbol{\mathcal{G}} \boldsymbol{G}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{F} \boldsymbol{\mathcal{F}}_{J_i}. \tag{37}$$

C. Scalable likelihood

Similarly to (16), the likelihood of the proposed solution is obtained by integrating out the latent variable $\mathring{\tilde{\boldsymbol{y}}}_i$:

$$Pr\left(\mathring{\tilde{\boldsymbol{x}}}_i|\boldsymbol{\Theta}\right) = \mathcal{N}\left[0, \tilde{\boldsymbol{\Sigma}} + \mathring{\tilde{\boldsymbol{A}}}\mathring{\tilde{\boldsymbol{A}}}^T\right], \tag{38}$$

which can be split into three terms (like in (17)), the last two of them being difficult to evaluate, as they require us to compute, respectively, the determinant and the inverse of the following large matrix:

$$\left(\tilde{\boldsymbol{\Sigma}} + \mathring{\tilde{\boldsymbol{A}}}\mathring{\tilde{\boldsymbol{A}}}^T\right). \tag{39}$$

However, with our proposed solution, $\tilde{\boldsymbol{\Sigma}} + \mathring{\tilde{\boldsymbol{A}}}\mathring{\tilde{\boldsymbol{A}}}^T$ is a block diagonal matrix, the upper left block being $\boldsymbol{\Sigma} + J_i\boldsymbol{F}\boldsymbol{F}^T + \boldsymbol{G}\boldsymbol{G}^T$ and the $J_i - 1$ other ones being equal to $\boldsymbol{\Sigma} + \boldsymbol{G}\boldsymbol{G}^T$. Therefore, some simplifications are possible to improve the efficiency of the approach.

First, the second term of the likelihood in (17) which involves the computation of the determinant of this large matrix can be efficiently computed using the following decomposition:

$$\det\left(\tilde{\boldsymbol{\Sigma}} + \mathring{\tilde{\boldsymbol{A}}}\mathring{\tilde{\boldsymbol{A}}}^T\right) = \det\left(\tilde{\boldsymbol{\Sigma}}\right)\det\left(\tilde{\boldsymbol{I}} + \mathring{\tilde{\boldsymbol{A}}}^T\tilde{\boldsymbol{\Sigma}}^{-1}\mathring{\tilde{\boldsymbol{A}}}\right), \tag{40}$$

and then using the block decomposition of $\tilde{\boldsymbol{I}} + \mathring{\tilde{\boldsymbol{A}}}^T\tilde{\boldsymbol{\Sigma}}^{-1}\mathring{\tilde{\boldsymbol{A}}}$ given by (28) and (29) leading to

$$\det\left(\tilde{\boldsymbol{\Sigma}} + \mathring{\tilde{\boldsymbol{A}}}\mathring{\tilde{\boldsymbol{A}}}^T\right) = \det\left(\boldsymbol{\Sigma}\right)^{J_i}\det\left(\boldsymbol{\mathcal{G}}^{-1}\right)^{J_i}\det\left(\boldsymbol{\mathcal{F}}_{J_i}^{-1}\right). \tag{41}$$

Second, the third term of the likelihood can be computed by performing a matrix inversion by blocks. Combining this approach with the Woodbury matrix identity for each of these blocks, it can be shown that

$$\begin{aligned}\mathring{\tilde{\boldsymbol{x}}}_i^T\left(\tilde{\boldsymbol{\Sigma}} + \mathring{\tilde{\boldsymbol{A}}}\mathring{\tilde{\boldsymbol{A}}}^T\right)^{-1}\mathring{\tilde{\boldsymbol{x}}}_i &= \sum_{j=1}^{J_i}\bar{\boldsymbol{x}}_{i,j}^T\boldsymbol{\Sigma}^{-1}\bar{\boldsymbol{x}}_{i,j} \\ &- \left(\sum_{j=1}^{J_i}\bar{\boldsymbol{x}}_{i,j}^T\boldsymbol{\mathcal{S}}^T\boldsymbol{F}\right)\boldsymbol{\mathcal{F}}_{J_i}\left(\sum_{j=1}^{J_i}\boldsymbol{F}^T\boldsymbol{\mathcal{S}}\bar{\boldsymbol{x}}_{i,j}\right) \\ &- \sum_{j=1}^{J_i}\bar{\boldsymbol{x}}_{i,j}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{G}\boldsymbol{\mathcal{G}}\boldsymbol{G}^T\boldsymbol{\Sigma}^{-1}\bar{\boldsymbol{x}}_{i,j}.\end{aligned} \tag{42}$$

If the likelihood computation involves the same number of samples J_i several times, the above expression might be further optimized using the same trick as proposed in [2]. The square root of the matrices $\boldsymbol{\mathcal{F}}_{J_i}$ and $\boldsymbol{\mathcal{G}}$ could be precomputed and then used to calculate one half of each the last two terms before taking their magnitude. It is important to notice that the change of variable does not affect the PLDA model, and hence, the above formulae remain valid to compute the likelihood given by (17).

Finally, the above equations show that storing the information for a model (enrollment of a given class) reduces to a single low-dimensional feature vector such as $\left(\sum_{j=1}^{J_i}\boldsymbol{F}^T\boldsymbol{\mathcal{S}}\bar{\boldsymbol{x}}_{i,j}\right)$ and a scalar (corresponding to the other terms involved in the likelihood computation), which further emphasizes the efficiency and scalability of the proposed approach.

Complexity		Li *et al.* [2]	Our approach
Likelihood computation	Memory	$\mathcal{O}\left(J_i^2\right)$	$\mathcal{O}\left(1\right)$
	Time	$\mathcal{O}\left(J_i^2\right)$	$\mathcal{O}\left(J_i\right)$
Training	Memory	$\mathcal{O}\left(J_i^2\right)$	$\mathcal{O}\left(1\right)$
	Time	$\mathcal{O}\left(J_i^3\right)$	$\mathcal{O}\left(J_i\right)$

TABLE I: Complexity with respect to the number of samples J_i for the class, for both the likelihood computation and the training, assuming that matrices have been precomputed whenever possible. The likelihood complexity of [2] is the one for the joint distribution approach (see [2, Section 3.2]).

IV. Complexity

In this section, we analyze the complexity of our proposed solution against the solution proposed by Li *et al.* [2]. We will start by first examining the complexity of training the PLDA model and then finish with some analysis and comments on the complexity of computing the likelihood. We will refer to time and memory complexity to express respectively the time and the memory required to run the algorithm. A quick summary of the below analysis is provided in Table I.

A. Training

As mentioned previously, one of the most computationally demanding parts of the PLDA approach occurs during the E-Step of the training algorithm. The E-Step update rules given by Li *et al.* [2] rely on the usage of the matrix $\left(\tilde{\boldsymbol{I}} + \tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{A}}\right)$, which grows quadratically with the number of samples. This suggests that the training procedure is demanding as, for each iteration of EM, this matrix has to be inverted and multiplied with the similarly large matrix $\tilde{\boldsymbol{A}}$. The memory complexity is then given by $\mathcal{O}\left(J_i^2\right)$, whereas the time complexity is $\mathcal{O}\left(J_i^3\right)$[3]. In contrast, the update rules of our proposed approach provided in section III-B show that the use of the structure of the PLDA model leads to a much more efficient training procedure.

Considering the computation of the first-order moment of the latent variables, this is in contrast to [2] performed on a per sample basis. Therefore, the time complexity is linear with the number of samples J_i, as opposed to cubic (inversion of a large matrix which grows quadratically). In addition, the matrices involved in (31) and (34) are common to all the training samples for the class. Therefore, they can be precomputed and the memory requirement is constant, irrespective of the number of samples. This leads to a memory complexity of $\mathcal{O}\left(1\right)$. And, as for the likelihood computation, if the number of training samples for the class J_i has changed, only the matrix $\boldsymbol{\mathcal{F}}_{J_i}$ needs to be recomputed again.

Finally, the second-order moment of the latent variables is only ever used on a per sample basis, as already depicted in previous work [1], [2]. Furthermore, this has linear time complexity with the number of samples J_i, and once again, many matrices can be precomputed by examining (36) to

[3]For readability, we consider that the time complexity of inverting a square matrix of size (N, N) is given by $\mathcal{O}\left(N^3\right)$. Using a non-naive approach such as the popular Strassen algorithm, it is true that this can be improved to $\mathcal{O}\left(N^\alpha\right)$ with $\alpha \approx 2.81$.

further optimize the training procedure. In addition, only the sum of the second-order moments is required for the maximization step, which does not affect the memory complexity for training, which is $\mathcal{O}(J_i)$.

B. Likelihood computation

Comparing the naive way to compute the likelihood given by (17) with our proposed solution (given by (41) and (42)), several conclusions can be drawn.

Considering memory usage, the matrix $\left(\tilde{\boldsymbol{I}} + \tilde{\boldsymbol{A}}^T \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{A}}\right)$ involved in the approach of Li *et al.* [2] is of size $(D_F + J_i D_G, D_F + J_i D_G)$, and hence the memory complexity is $\mathcal{O}\left(J_i^2\right)$. In contrast, the proposed solution exploits the structure of the problem by using several smaller matrices, such as $\boldsymbol{\mathcal{G}}$, $\boldsymbol{\mathcal{S}}$ or $\boldsymbol{\mathcal{F}}_{J_i}$, which are of constant size, irrespective of the number of samples. This implies that the memory complexity is $\mathcal{O}(1)$.

In addition, many of these matrices can be precomputed. For instance, $\boldsymbol{\mathcal{G}}$ or $\boldsymbol{\mathcal{S}}$ are required to compute the likelihood of any set of samples. Besides, the inverted matrix $\boldsymbol{\mathcal{F}}_{J_i}$ can be precomputed and used to compute the likelihood of any set of J_i samples. If the number of samples J_i involved in the likelihood computation has changed, the only new and required matrix inversion would be for the $\boldsymbol{\mathcal{F}}_{J_i}$ matrix which is of size (D_F, D_F). In contrast, with the naive solution, the inverted matrix $\left(\tilde{\boldsymbol{\Sigma}} + \tilde{\boldsymbol{A}}\tilde{\boldsymbol{A}}^T\right)^{-1}$ would have to be recomputed. Furthermore the time complexity considering matrix inversion and with respect to the number of samples J_i has in this case become $\mathcal{O}(1)$ instead of $\mathcal{O}\left(J_i^3\right)$.

Finally, assuming that all the previous matrices involved in the likelihood computation have been precomputed, the likelihood of J_i samples given the PLDA model requires the calculation of products $\tilde{\boldsymbol{\Gamma}}^T \tilde{\tilde{\boldsymbol{x}}}_i$. The time complexity with respect to the number of samples for the class J_i is then quadratic, which is $\mathcal{O}\left(J_i^2\right)$. With the proposed solution and using the square root of the matrices $\boldsymbol{\mathcal{G}}$ and $\boldsymbol{\mathcal{F}}_{J_i}$, this involves several matrix-vector multiplications such as $\left(\boldsymbol{\mathcal{G}}^{\frac{1}{2}} \boldsymbol{G}^T \boldsymbol{\Sigma}^{-1}\right) \bar{\boldsymbol{x}}_{i,j}$, and the time complexity is given by $\mathcal{O}(J_i)$.

V. Experimental Results

PLDA has already been shown to provide state-of-the-art performance for both face recognition [2] and speaker recognition [3]. Given the prior work on PLDA, our aim here is not to prove the efficacy of PLDA in general. Hence, we will not provide an exhaustive experimental study on most of the well-known databases, such as FRGC, for instance, nor against multiple baselines, as it will be out of the scope of this paper. Rather, we aim to demonstrate the usefulness of providing a scalable and exact solution for training PLDA[4]. We use the task of face authentication to demonstrate this. First, we demonstrate that this scalable implementation provides similar state-of-the-art performance to [2] already published on LFW: In the preceding section, we have already demonstrated the theoretical equivalence. Second, we provide evidence that having more training data per identity gives us a better PLDA model and at the same time highlight the difference between our scalable solution and the other nonscalable solution [2]. We do this by performing experiments on Multi-PIE and deriving several models using a varying number of training samples per identity (class). This is only possible on Multi-PIE since it contains a significant number of images per identity (up to 76) unlike FRGC. Furthermore, FRGC is not applicable as it does not include separate development and evaluation sets[5].

We test our scalable system on the task of face authentication. In this scenario, someone claims to be identity i and presents their signal (face or speech) as a probe, $\boldsymbol{x}_p$, to the system. The system already has enrolled the identity by having a known sample (or samples) of them, $\boldsymbol{x}_{i,1}$. So the system calculates if: 1) "they are the client i," which equates to tying together the latent identity variable for $\boldsymbol{x}_{i,1}$ and $\boldsymbol{x}_p$, which is $Pr(\boldsymbol{x}_{i,1}, \boldsymbol{x}_p)$ or 2) "they are not the client i," which equates to having separate latent identity variables, which is $Pr(\boldsymbol{x}_{i,1})\, Pr(\boldsymbol{x}_p)$. These joint distributions are given by (15) and the information is combined using Bayes rule to form a log-likelihood ratio:

$$s = \ln\left(Pr(\boldsymbol{x}_{i,1}, \boldsymbol{x}_p)\right) - \ln\left(Pr(\boldsymbol{x}_{i,1})\, Pr(\boldsymbol{x}_p)\right). \quad (43)$$

A threshold is then applied to this score s to decide if the probe image is of the identity they claim to be. We use the scalable version of computing the likelihood for all of our experiments.

For all of the experiments, we have used the following initialization procedure: The subspaces $\boldsymbol{F}$ and $\boldsymbol{G}$ are initialized based on the result of singular value decomposition (SVD) on the between-class and within-class scatter of the training data, respectively; each basis vector is normalized by the eigenvalue to ensure that the latent variables are unit variance. We initialize the variance $\boldsymbol{\Sigma}$ to be the variance of the training data. We always perform 200 rounds of EM training as preliminary work showed that this was a reasonable value.

A. LFW Experiments

We provide results on the LFW database [5] using the same SIFT features as used in [2] and [6], which are extracted on nine landmark points on the face at three different scales. These features are used because they are publicly available and produced competitive performance on LFW. Furthermore, SIFT descriptors have previously demonstrated appealing performance in the face recognition field [7]. As per [2], after concatenating the keypoint descriptors, we reduce the dimensionality of these feature vectors using PCA and retain the top 200 dimensions. This is done independently for View1 and for each fold of View2.

In [2] the optimal PLDA hyperparameters using these SIFT features were stated to be $D_F = D_G = 48$. Therefore, we used the same hyperparameters and found that our performance was comparable to the PLDA model of [2]; see Table II.

[4] An implementation of the proposed solution can be found with the Bob project http://idiap.github.com/bob.

[5] In the FRGC database, 153 clients occur in both the training set as well as the evaluation set, and there is no publicly-available development set.

Method	Accuracy
SIFT PLDA, funneled (u)	0.863 ± 0.005
SIFT PLDA, funneled (u) [2]	0.862 ± 0.012
SIFT LDML, funneled (u) [6]	0.832 ± 0.004

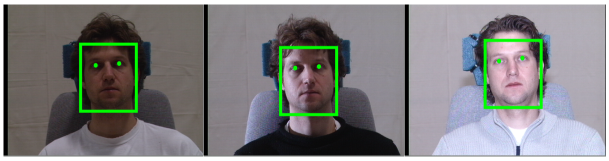
TABLE II: Above are the results on LFW View2 for our system and a prior PLDA system and the LDML system [6] using the same SIFT features. Accuracy is presented in terms of mean classification accuracy and the standard error of the mean [5].

We attribute the minor differences in performance to potential differences in initialization of the PLDA model and the number of EM iterations. Given the theoretical equivalence and the result of these experiments, we conclude that our PLDA model implementation is equivalent to that of [2], which achieves state-of-the-art performance for several face recognition tasks.

B. Multi-PIE Experiments

In this section we present experiments on Multi-PIE [8] that highlight the scalability of our derivation. The aim is to demonstrate that using many samples per identity improves face recognition accuracy, and thereby demonstrate the need for the proposed scalable solution. We use the Multi-PIE database as it has 337 identities (classes) and up to 76 images for each identity to train with. To ensure the validity of our approach we have provided a protocol with independent training, development, and evaluation sets and provided a comparison against three well-known techniques from literature using exactly the same features.

The protocol for the Multi-PIE database consists of three independent sets: training, development, and evaluation. We use the annotations for the illumination varying part of Multi-PIE for camera 05_1 which are frontal images with 19 images per session[6]. Some example images are provided in Fig. 1.



(a) (b) (c)

Fig. 1: Above are three example images for one identity in Multi-PIE along with a corresponding bounding box for cropping the face image in green.

The three sets are independent in terms of identity. The training set consists of 208 identities who have a variable number of images ranging from 19 to 76; in total we have $9,785$ images for training the parameters of our models. The development set has 64 identities and we use one image from the first session to enroll each identity (to make a model of what each identity looks like). Tests are performed using the remaining three sessions; impostor tests are done in an exhaustive manner where the remaining 63 identities are used. This gives us $4,864$ true access and $306,432$ false access samples. The evaluation set has 65 identities and we use this set in the same manner as the development set. In total, we have $4,940$ true access and $316,160$ false access samples.

[6]These annotations are available at: www.idiap.ch/resource/biometric. Also, there are 20 images per session, but two of these are almost exactly the same and so we only retain one of them, resulting in 19 images per session.

System	Evaluation HTER (%)
PCA [12]	10.34%
LDA [13]	8.22%
LBP Histogram (χ^2) [10]	8.02%
PLDA	6.77%

TABLE III: The results in this table are for the HTER on the Evaluation set for Multi-PIE for the four systems.

1) System Description: Our system consists of the following steps: We initially crop and resize the image, using the annotated eye positions, to be 64×80 pixels in size and apply a standard image normalization algorithm [9]. Following this, we obtain local LBP features, similar to [10], by applying a 10×10 window across the face and from each window we obtain a uniform LBP histogram using a radius of 2 and 8 equally spaced points [11][7]. We concatenate all of these histograms to form a single feature vector and then dimensionally reduce this to 500 dimensions using PCA, a similar approach with LBP histograms was used for previous PLDA experiments in [2]. We then use these features as input to the PLDA system.

We provide three reference systems. These reference systems were chosen because they are well known, they are 1) a PCA system [12] using the same PCA reduced features as we use for PLDA, 2) a linear discriminant analysis (LDA) system [13] trained on the same dimensionality reduced features as we use for PLDA, and 3) a system which directly compares each local LBP histogram [10] using a chi-squared distance.

2) Results: We present results on Multi-PIE by quoting the half total error rate (HTER), similar to the results produced for BANCA [14]. The HTER is a performance measure that obtains a threshold τ at the equal error rate (EER) point on an independent development set that is then applied to an evaluation set. The HTER is then obtained by taking the average of the false acceptance rate and the false rejection rate on the evaluation set at this threshold τ.

The results for PLDA and the three reference systems can be found in Table III. These results make use of the full training set, $9,785$ images, where there were at most 76 images per identity and as few as 19 images per identity. It can be seen from these results that PLDA consistently outperforms every other system.

Using the previous result, we took the optimal hyperparameters of PLDA, $D_F = 128$ and $D_G = 64$, and varied the number of training images per identity to show the impact on performance [8]. We varied the maximum number of training

[7]We have an overlap margin of 4 pixels between consecutive blocks.

[8]For the optimal parameters, the PLDA system took approximately 35 minutes to train using $9,785$ images and 45 minutes to produce the scores (including enrollment) for the $311,296$ development probes. These experiments were run on an Intel Quad Core CPU 2.50GHz computer using a C++ implementation without multithreading.

samples per identity from 2 to 76 and plotted the relative size of the matrices needed to compute the latent variables (on a log-scale) versus the number of training samples; the results are plotted in Fig. 2. The matrices used to compute the latent variables for 1) the scalable derivation were $\boldsymbol{F}^T\boldsymbol{\mathcal{S}}$, $\left(\boldsymbol{I}_{D_G} + \boldsymbol{G}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{G}\right)^{-1}$ and $\left(\boldsymbol{I}_{D_F} + J_i\boldsymbol{F}^T\boldsymbol{\mathcal{S}}\boldsymbol{F}\right)^{-1}$ and 2) the nonscalable derivation was $\left(\tilde{\boldsymbol{I}} + \tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{A}}\right)^{-1}$.

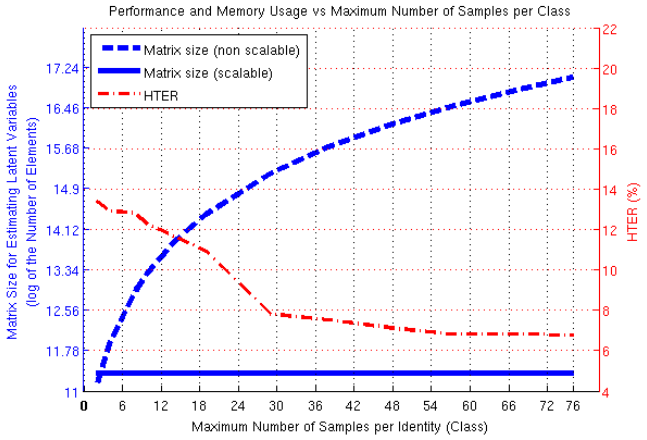


Fig. 2: We plot the HTER achieved using the PLDA system on the Multi-PIE database, showing that the error rate decreases as we use more samples per identity. Matrix size is also plotted to contrast the corresponding memory usage of the nonscalable and scalable solutions in terms of the log number of elements in the matrices used to compute the latent variables.

In Fig. 2, we also provide a plot of the HTER versus the number of training samples. It can be seen that, for these experiments, adding more training samples always improves performance. This clearly shows the applicability of a scalable form of PLDA because, as the number of training samples increases, it quickly becomes infeasible to use a nonscalable derivation.

VI. Conclusions

We have presented a novel and scalable derivation for training PLDA. This derivation is theoretically equivalent to the previous nonscalable solution [4] and so obviates the need for a variational approximation [3]. Aside from the theoretical proof, we further demonstrate the efficacy of this technique by performing experiments on two well-known face databases. First, on LFW we illustrate the equivalence of our model with that of [2], which achieves state-of-the-art performance on several face recognition problems. Second, on the large Multi-PIE database, we illustrate the gain in performance when using more training samples per identity (class) and so demonstrate the need for a scalable solution. Our second contribution has been to present a novel and scalable derivation for computing the likelihood of PLDA, which we have used in all of our experiments. Finally, we show that enrollment of a client (storing the information for a model) reduces to calculating a single low-dimensional feature vector rather than retaining multiple high-dimensional feature vectors.

References

[1] S. J. D. Prince and J. H. Elder, "Probabilistic Linear Discriminant Analysis for Inferences about Identity," in *IEEE International Conference on Computer Vision*, 2007, pp. 1–8. 1, 2, 3, 4, 5

[2] P. Li, Y. Fu, U. Mohammed, J. H. Elder, and S. J. D. Prince, "Probabilistic models for inference about identity," *Pattern Analysis and Machine Intelligence*, vol. 34, pp. 144 – 157, 2012. 1, 2, 3, 5, 6, 7, 8

[3] P. Kenny, "Bayesian Speaker Verification with Heavy-Tailed Priors," in *Odyssey: The Speaker and Language Recognition Workshop*, 2010. 1, 2, 3, 6, 8

[4] P. Li and S. J. D. Prince, *Advance in Face Image Analysis: Techniques and Technologies*. Idea Group Publishing, 2010, ch. Probabilistic Methods for Face Registration and Recognition (In Press). 1, 3, 8

[5] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments," University of Massachusetts, Amherst, Tech. Rep., 2007. [Online]. Available: http://vis-www.cs.umass.edu/lfw/ 6, 7

[6] M. Guillaumin, J. Verbeek, and C. Schmid, "Is that you? Metric Learning Approaches for Face Identification," in *International Conference on Computer Vision*, 2009. 6, 7

[7] D. R. Kisky, A. Rattani, E. Gross, and M. Tistarelli, "Face Identification by SIFT-based Complete Graph Topology," in *5th IEEE International Workshop on Automatic Identification Advanced Technologies (AUTOID 2007)*, 2007, pp. 63–68. 6

[8] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-PIE," *Image and Vision Computing*, vol. 28, pp. 807–813, 2010. 7

[9] X. Tan and B. Triggs, "Enhanced Local Texture Feature Sets for Face Recognition under Difficult Lighting Conditions," in *AMFG*, 2007. 7

[10] T. Ahonen, A. Hadid, and M. Pietikainen, "Face Description with Local Binary Patterns: application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Inteligence*, vol. 28, pp. 2037–2041, 2006. 7

[11] T. Ojala, M. Pietikinen, and T. Maenpaa, "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, pp. 971–987, 2002. 7

[12] M. Turk and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, vol. 3, pp. 71–86, 1991. 7

[13] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 711–720, 1997. 7

[14] E. Bailly-Bailliere, S. Bengio, F. Bimbo, M. Hamouz, J. Kittler, J. Mariethoz, J. Matas, K. Messer, V. Popovici, F. Poree, B. Ruiz, and J.-P. Thiran, "The BANCA database and evaluation protocol," *Lecture Notes in Computer Science*, pp. 625–638, 2003. 7

Laurent El Shafey graduated with a Master in Electrical Engineering from Supelec, France, and a Master in Computer Science from the TU Darmstadt, Germany. He is currently working towards his PhD at Ecole Polytechnique Federale de Lausanne and Idiap Research Institute, Switzerland. His research interest is in biometrics, computer vision and machine learning.

Christopher McCool received his PhD with the Speech, Audio, Image and Video Technologies (SAIVT) group, Queensland University of Technology (QUT), Australia in 2007. He is currently a postdoctoral researcher in Biometrics at the Idiap Research Institute, he has a particular interest in 2D and 3D face authentication, face detection and computer vision.

Roy Wallace received his BEng(Hon) in 2006 and PhD in Engineering in 2010 with the Speech, Audio, Image and Video Technologies (SAIVT) group, Queensland University of Technology (QUT), Australia. He is now a postdoctoral researcher in biometrics and machine learning at Idiap Research Institute, Switzerland, with a particular interest in the use of biometrics for forensics.

Sébastien Marcel received the PhD degree in signal processing from Université de Rennes I in France (2000) at CNET, the research center of France Telecom (now Orange Labs). He is a senior research scientist at the Idiap Research Institute (CH), where he leads a research team and conducts research on face recognition, speaker recognition and spoofing attacks detection.

A Scalable Formulation of Probabilistic Linear Discriminant Analysis: Applied to Face Recognition

Laurent El Shafey, Chris McCool, Roy Wallace, and Sébastien Marcel

APPENDIX A
MATHEMATICAL DERIVATIONS

The goal of the following section is to provide more detailed proofs of the formulae given in the article for both training and computing the likelihood.

The following proofs make use of a formulation of the inverse of a block matrix that uses the Schur complement. The corresponding identity can be found in [1] (Equations 1.11 and 1.10)

$$\begin{bmatrix} \boldsymbol{L} & \boldsymbol{M} \\ \boldsymbol{N} & \boldsymbol{O} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{R}, & -\boldsymbol{R}\boldsymbol{M}\boldsymbol{O}^{-1} \\ -\boldsymbol{O}^{-1}\boldsymbol{N}\boldsymbol{R}, & \boldsymbol{O}^{-1} + \boldsymbol{O}^{-1}\boldsymbol{N}\boldsymbol{R}\boldsymbol{M}\boldsymbol{O}^{-1} \end{bmatrix}, \tag{51}$$

where we have substituted $\boldsymbol{R} = \left(\boldsymbol{L} - \boldsymbol{M}\boldsymbol{O}^{-1}\boldsymbol{N}\right)^{-1}$.

Another related expression is the Woodbury matrix identity (Equation $C.7$ of [2]), which states that

$$\begin{aligned}&(\boldsymbol{L} + \boldsymbol{M}\boldsymbol{O}\boldsymbol{N})^{-1} = \\ &\quad \boldsymbol{L}^{-1} - \boldsymbol{L}^{-1}\boldsymbol{M}\left(\boldsymbol{O}^{-1} + \boldsymbol{N}\boldsymbol{L}^{-1}\boldsymbol{M}\right)^{-1}\boldsymbol{N}\boldsymbol{L}^{-1}.\end{aligned} \tag{52}$$

A. Scalable training

The bottleneck of the training procedure is the expectation step (E-Step) of the Expectation-Maximization algorithm. This E-Step requires the computation of the first- and second-order moments of the latent variables.

1) Estimating the first-order moment of the latent variables: The most computationally expensive part when estimating the latent variables is the inversion of the matrix $\mathring{\tilde{\boldsymbol{\mathcal{P}}}}$ (Equation (27)). This matrix is block diagonal, the two blocks being $\boldsymbol{\mathcal{P}}_0$ (Equation (28)) and (a repetition of) $\boldsymbol{\mathcal{P}}_1$ (Equation (29)),

$$\mathring{\tilde{\boldsymbol{\mathcal{P}}}} = \begin{bmatrix} \boldsymbol{\mathcal{P}}_0 & 0 & \cdots & 0 \\ 0 & \boldsymbol{\mathcal{P}}_1 & \ddots & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \boldsymbol{\mathcal{P}}_1 \end{bmatrix}. \tag{53}$$

The inverse of $\boldsymbol{\mathcal{P}}_1$ is equal to the matrix $\boldsymbol{\mathcal{G}}$, defined by (30). This matrix is of constant size ($D_G \times D_G$), irrespective of the number of training samples for the class. In addition, the inversion of $\boldsymbol{\mathcal{P}}_0$ can be further optimized using the block matrix inversion identity introduced at the beginning of this section, leading to

$$\boldsymbol{\mathcal{P}}_0^{-1} = \begin{bmatrix} \boldsymbol{\mathcal{F}}_{J_i} & \sqrt{J_i}\boldsymbol{\mathcal{H}}^T \\ \sqrt{J_i}\boldsymbol{\mathcal{H}} & \left(\boldsymbol{I}_{D_G} - J_i\boldsymbol{\mathcal{H}}\boldsymbol{F}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{G}\right)\boldsymbol{\mathcal{G}} \end{bmatrix}, \tag{54}$$

where $\boldsymbol{\mathcal{F}}_{J_i}$ is defined by (33) and $\boldsymbol{\mathcal{H}}$ by (37).

Then, the computation of $\mathring{\tilde{\boldsymbol{\mathcal{P}}}}^{-1}\mathring{\tilde{\boldsymbol{A}}}^T\tilde{\boldsymbol{\Sigma}}^{-1}$ gives a block diagonal matrix, the first block being

$$\begin{bmatrix} \sqrt{J_i}\boldsymbol{\mathcal{F}}_{J_i}\boldsymbol{F}^T\boldsymbol{\mathcal{S}} \\ \boldsymbol{\mathcal{G}}\boldsymbol{G}^T\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{I}_{D_x} - J_i\boldsymbol{F}\boldsymbol{\mathcal{F}}_{J_i}\boldsymbol{F}^T\boldsymbol{\mathcal{S}}\right) \end{bmatrix},$$

and the other ones being equal to $\boldsymbol{\mathcal{G}}\boldsymbol{G}^T\boldsymbol{\Sigma}^{-1}$.

As explained in section III.B.a of the article, $\boldsymbol{h}_i$ corresponds to the upper subvector of $\mathring{\tilde{\boldsymbol{y}}}_i$ and is not affected by the change of variable, as depicted in (21). Therefore, the first-order moment of $\boldsymbol{h}_i$ is directly obtained by multiplying the first block-rows of the matrix $\mathring{\tilde{\boldsymbol{\mathcal{P}}}}^{-1}\mathring{\tilde{\boldsymbol{A}}}^T\tilde{\boldsymbol{\Sigma}}^{-1}$ with $\mathring{\tilde{\boldsymbol{x}}}_i$, which gives (31).

Considering only the $\mathring{\tilde{\boldsymbol{w}}}_i$ (lower) subvector of $\mathring{\tilde{\boldsymbol{y}}}_i$, the corresponding (lower) part $\mathring{\tilde{\boldsymbol{\mathcal{B}}}}$ of the matrix $\mathring{\tilde{\boldsymbol{\mathcal{P}}}}^{-1}\mathring{\tilde{\boldsymbol{A}}}^T\tilde{\boldsymbol{\Sigma}}^{-1}$ can be decomposed into a sum of two matrices, the first one being sparse with a single non-zero block (upper left) equal to $\boldsymbol{\mathcal{B}}_0 = -J_i\boldsymbol{\mathcal{G}}\boldsymbol{G}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{F}\boldsymbol{\mathcal{F}}_{J_i}\boldsymbol{F}^T\boldsymbol{\mathcal{S}}$, and the second one being diagonal by blocks with identical blocks $\boldsymbol{\mathcal{B}}_1 = \boldsymbol{\mathcal{G}}\boldsymbol{G}^T\boldsymbol{\Sigma}^{-1}$,

$$\mathring{\tilde{\boldsymbol{\mathcal{B}}}} = \begin{bmatrix} \boldsymbol{\mathcal{B}}_0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} \boldsymbol{\mathcal{B}}_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \boldsymbol{\mathcal{B}}_1 \end{bmatrix}. \tag{55}$$

Furthermore, the first-order moment of the variables $\tilde{\boldsymbol{w}}_i$ is given by

$$\begin{aligned} E\left[\tilde{\boldsymbol{w}}_i|\tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right] = &\left(\tilde{\boldsymbol{U}}^T \otimes \boldsymbol{I}_{D_G}\right)\begin{bmatrix} \boldsymbol{\mathcal{B}}_0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\mathring{\tilde{\boldsymbol{x}}}_i \\ &+ \left(\tilde{\boldsymbol{U}}^T \otimes \boldsymbol{I}_{D_G}\right)\begin{bmatrix} \boldsymbol{\mathcal{B}}_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \boldsymbol{\mathcal{B}}_1 \end{bmatrix}\left(\tilde{\boldsymbol{U}} \otimes \boldsymbol{I}_{D_x}\right)\tilde{\boldsymbol{x}}_i. \end{aligned} \tag{56}$$

The previous decomposition greatly simplifies the computation, and leads to the following expression for each $\boldsymbol{w}_{i,j}$

$$\begin{aligned} E\left[\boldsymbol{w}_{i,j}|\tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right] = &\boldsymbol{\mathcal{G}}\boldsymbol{G}^T\boldsymbol{\Sigma}^{-1}\bar{\boldsymbol{x}}_{i,j} \\ &- \boldsymbol{\mathcal{G}}\boldsymbol{G}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{F}\boldsymbol{\mathcal{F}}_{J_i}\boldsymbol{F}^T\boldsymbol{\mathcal{S}}\sum_j \bar{\boldsymbol{x}}_{i,j} \end{aligned} \tag{57}$$

L. El Shafey is with Idiap Research Institute and Ecole Polytechnique Fédérale de Lausanne, Switzerland e-mail: laurent.el-shafey@idiap.ch

C. McCool, R. Wallace and S. Marcel are with Idiap Research Institute, Martigny, Switzerland e-mail: {christopher.mccool,roy.wallace,sebastien.marcel}@idiap.ch

The research leading to these results has received funding from the European Communitys Seventh Framework Programme (FP7) under grant agreements 238803 (BBfor2) and 257289 (TABULA RASA).

which, after grouping the common factors, finally provides (34).

2) Estimating the second-order moment of the latent variables: The computation of the second-order moment of the latent variables is performed using the expression of $\mathring{\tilde{\boldsymbol{\mathcal{P}}}}^{-1}$ computed before, and reverting the change of variable. This leads to the computation of the following variance

$$Var\left[\tilde{\boldsymbol{y}}_i|\tilde{\boldsymbol{x}}_i,\boldsymbol{\Theta}\right] = \tilde{\boldsymbol{\mathfrak{P}}}^{-1} = \tilde{\boldsymbol{V}}^{-1}\mathring{\tilde{\boldsymbol{\mathcal{P}}}}^{-1}\tilde{\boldsymbol{V}}, \tag{58}$$

where $\tilde{\boldsymbol{V}}^{-1}$ is efficiently computed as

$$\tilde{\boldsymbol{V}}^{-1} = \begin{bmatrix} \boldsymbol{I}_{D_F} & 0 \\ 0 & \tilde{\boldsymbol{U}}^T \otimes \boldsymbol{I}_{D_G} \end{bmatrix}. \tag{59}$$

In addition, the variance of the latent variables is only ever used on a per sample basis, which implies that only the elements of the first row, first column and diagonal of $\tilde{\boldsymbol{\mathfrak{P}}}^{-1}$ are necessary. A direct computation provides a matrix which is as follows:

$$\begin{bmatrix} \boldsymbol{\mathcal{F}}_{J_i} & \boldsymbol{\mathcal{H}}^T & \cdots & \boldsymbol{\mathcal{H}}^T \\ \boldsymbol{\mathcal{H}} & \boldsymbol{\mathcal{D}} & & \\ \vdots & & \ddots & \\ \boldsymbol{\mathcal{H}} & & & \boldsymbol{\mathcal{D}} \end{bmatrix},$$

with $\boldsymbol{\mathcal{D}} = \left(\boldsymbol{I}_{D_G} - \boldsymbol{\mathcal{H}}\boldsymbol{F}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{G}\right)\boldsymbol{\mathcal{G}}$. Exploiting the structure of this matrix leads to (36).

Finally the corresponding term involved in the E-Step update rules is given by

$$\begin{aligned} E\left[\tilde{\boldsymbol{y}}_i\tilde{\boldsymbol{y}}_i^T|\tilde{\boldsymbol{x}}_i,\boldsymbol{\Theta}\right] &= Var\left[\tilde{\boldsymbol{y}}_i|\tilde{\boldsymbol{x}}_i,\boldsymbol{\Theta}\right] \\ &\quad + E\left[\tilde{\boldsymbol{y}}_i|\tilde{\boldsymbol{x}}_i,\boldsymbol{\Theta}\right]E\left[\tilde{\boldsymbol{y}}_i|\tilde{\boldsymbol{x}}_i,\boldsymbol{\Theta}\right]^T. \end{aligned} \tag{60}$$

B. Likelihood computation

As described in section II.B. of the article, the likelihood of the PLDA model can be expressed as a sum of three terms, which is given by (17). In addition, the change of variable, which gives the proposed transformed PLDA model, leads to efficient computations but has no impact on the final value of the likelihood.

Firstly, the first term is obvious to compute and is not affected by the change of variable. Secondly, the calculation of the second term has already been well described in the article by (40) and (41), where the following block determinant is used for $\boldsymbol{\mathcal{P}}_0$,

$$\begin{aligned} \det\left[\boldsymbol{\mathcal{P}}_0\right] &= \det\begin{bmatrix} \boldsymbol{I}_{D_F} + J_i\boldsymbol{F}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{F} & \sqrt{J_i}\boldsymbol{F}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{G} \\ \sqrt{J_i}\boldsymbol{G}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{F} & \boldsymbol{I}_{D_G} + \boldsymbol{G}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{G} \end{bmatrix} \\ &= \det\left[\boldsymbol{I}_{D_G} + \boldsymbol{G}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{G}\right].\det\left[\boldsymbol{I}_{D_F} + J_i\boldsymbol{F}^T\boldsymbol{\mathcal{S}}\boldsymbol{F}\right], \end{aligned}$$

$\boldsymbol{\mathcal{S}}$ being defined by (32). In particular, the change of variable implies to multiply the left and right sides by two matrices which are orthogonal, and hence the product of their determinant is equal to one.

Finally, the third term involved in the computation of the likelihood relies on the inversion of the matrix $\tilde{\boldsymbol{\Sigma}} + \mathring{\tilde{\boldsymbol{A}}}\mathring{\tilde{\boldsymbol{A}}}^T$. This matrix is block diagonal, the upper left block being $\boldsymbol{\mathcal{T}}_0 = \boldsymbol{\Sigma} + J_i\boldsymbol{F}\boldsymbol{F}^T + \boldsymbol{G}\boldsymbol{G}^T$ and the $J_i - 1$ other ones being equal to $\boldsymbol{\mathcal{T}}_1 = \boldsymbol{\Sigma} + \boldsymbol{G}\boldsymbol{G}^T$. Therefore, this matrix inversion could be efficiently performed on a block basis.

In addition, $\boldsymbol{G}$ is of dimension (D_x, D_G), usually with $D_G \ll D_x$ and hence, $\boldsymbol{G}\boldsymbol{G}^T$ is a potentially low rank matrix. Furthermore, it is indeed possible to further reduce the complexity of each block inversion $\boldsymbol{\mathcal{T}}_0^{-1}$ and $\boldsymbol{\mathcal{T}}_1^{-1}$, using the Woodbury matrix identity introduced at the beginning of this section.

Furthermore, applying this identity to compute the inverse of the block $\boldsymbol{\mathcal{T}}_1$ leads to,

$$\boldsymbol{\mathcal{T}}_1^{-1} = \boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1}\boldsymbol{G}\boldsymbol{\mathcal{G}}\boldsymbol{G}^T\boldsymbol{\Sigma}^{-1} = \boldsymbol{\mathcal{S}} \tag{61}$$

Similarly, this identity could consecutively be applied twice to compute the inverse of the other block $\boldsymbol{\mathcal{T}}_0$, which gives

$$\boldsymbol{\mathcal{T}}_0^{-1} = \boldsymbol{\mathcal{S}} - J_i\boldsymbol{\mathcal{S}}^T\boldsymbol{F}\boldsymbol{\mathcal{F}}_{J_i}\boldsymbol{F}^T\boldsymbol{\mathcal{S}}, \tag{62}$$

$\boldsymbol{\mathcal{F}}_{J_i}$ being defined by (33).

Finally, the likelihood is expressed as a function of the transformed input $\mathring{\tilde{\boldsymbol{x}}}_i$. However, it is easy to notice that the change of variable could be easily reverted without increasing the complexity of the likelihood computation. $\left(\tilde{\boldsymbol{\Sigma}} + \mathring{\tilde{\boldsymbol{A}}}\mathring{\tilde{\boldsymbol{A}}}^T\right)^{-1}$ can indeed be expressed as a sum of two matrices, the first one being diagonal by blocks with identical blocks $\boldsymbol{\mathcal{S}}$, and the second one being sparse with a single non-zero block (upper left) equal to $J_i\boldsymbol{\mathcal{S}}^T\boldsymbol{F}\boldsymbol{\mathcal{F}}_{J_i}\boldsymbol{F}^T\boldsymbol{\mathcal{S}}$. This provides a formulation similar to the right hand side of (55). Next, to revert the change of variable, the left hand side of these matrices are multiplied by $\left(\tilde{\boldsymbol{U}} \otimes \boldsymbol{I}_{D_x}\right)^T$ whereas the right hand side is multiplied by $\left(\tilde{\boldsymbol{U}} \otimes \boldsymbol{I}_{D_x}\right)$. For the first matrix, all the blocks are the same, and hence the mixing matrices have no influence, such that it provides the first and third terms of (42). For the second matrix, it gives the second term of (42), which uses the sum of the input samples for the class.

APPENDIX B
SCALABLE (GAUSSIAN) PLDA IN A NUTSHELL

A. PLDA Model

$$\boldsymbol{x}_{i,j} = \boldsymbol{\mu} + \boldsymbol{F}\boldsymbol{h}_i + \boldsymbol{G}\boldsymbol{w}_{i,j} + \boldsymbol{\epsilon}_{i,j}. \tag{63}$$

$\boldsymbol{x}_{i,j}$: input signal, dimensionality D_x, i: class, j: sample
$\boldsymbol{F}$: subspace for the between-class variation
$\boldsymbol{G}$: subspace for the within-class variation
$\boldsymbol{h}_i$: position in the $\boldsymbol{F}$ subspace for $\boldsymbol{x}_{i,j}$ (size D_F)
$\boldsymbol{w}_{i,j}$: position in the $\boldsymbol{G}$ subspace for $\boldsymbol{x}_{i,j}$ (size D_G)
$\boldsymbol{\mu}$: mean of the data ($\boldsymbol{x}_{i,j}$)
$\boldsymbol{\epsilon}_{i,j}$: residual, zero mean and diagonal covariance $\boldsymbol{\Sigma}$
$\boldsymbol{N}$: number of classes
J_i: number of samples for class i

B. Common terms

$$\bar{\boldsymbol{x}}_{i,j} = \boldsymbol{x}_{i,j} - \boldsymbol{\mu} \tag{64}$$

$$\tilde{\boldsymbol{x}}_i = [\bar{\boldsymbol{x}}_{i,1}^T, \bar{\boldsymbol{x}}_{i,2}^T, \cdots, \bar{\boldsymbol{x}}_{i,J_i}^T]^T \tag{65}$$

$$\boldsymbol{y}_{i,j} = [\boldsymbol{h}_i^T, \boldsymbol{w}_{i,j}^T]^T \tag{66}$$

$$\tilde{\boldsymbol{y}}_i = [\boldsymbol{h}_i^T, \boldsymbol{w}_{i,1}^T, \boldsymbol{w}_{i,2}^T, \cdots, \boldsymbol{w}_{i,J_i}^T]^T \tag{67}$$

$$\boldsymbol{A} = [\boldsymbol{F}, \boldsymbol{G}] \tag{68}$$

$$\boldsymbol{\mathcal{G}} = \left(\boldsymbol{I}_{D_G} + \boldsymbol{G}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{G}\right)^{-1} \tag{69}$$

$$\boldsymbol{\mathcal{S}} = \left(\boldsymbol{\Sigma} + \boldsymbol{G}\boldsymbol{G}^T\right)^{-1} = \boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1}\boldsymbol{G}\boldsymbol{\mathcal{G}}\boldsymbol{G}^T\boldsymbol{\Sigma}^{-1} \tag{70}$$

$$\boldsymbol{\mathcal{F}}_{J_i} = \left(\boldsymbol{I}_{D_F} + J_i\boldsymbol{F}^T\boldsymbol{\mathcal{S}}\boldsymbol{F}\right)^{-1} \tag{71}$$

C. Scalable Training

1) E-Step:

First-order moment of the latent variables

$$E[\boldsymbol{h}_i|\tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}] = \boldsymbol{\mathcal{F}}_{J_i}\sum_j \boldsymbol{F}^T\boldsymbol{\mathcal{S}}\bar{\boldsymbol{x}}_{i,j} \tag{72}$$

$$E[\boldsymbol{w}_{i,j}|\tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}] = \boldsymbol{\mathcal{G}}\boldsymbol{G}^T\boldsymbol{\Sigma}^{-1}\left(\bar{\boldsymbol{x}}_{i,j} - \boldsymbol{F}E[\boldsymbol{h}_i|\tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}]\right) \tag{73}$$

Second-order moment of the latent variables
(This is only ever used on a per sample basis)

$$\boldsymbol{\mathcal{H}} = -\boldsymbol{\mathcal{G}}\boldsymbol{G}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{F}\boldsymbol{\mathcal{F}}_{J_i} \tag{74}$$

$$Var\left[\boldsymbol{y}_{i,j}|\tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right] = \begin{bmatrix} \boldsymbol{\mathcal{F}}_{J_i} & \boldsymbol{\mathcal{H}}^T \\ \boldsymbol{\mathcal{H}} & \left(\boldsymbol{I}_{D_G} - \boldsymbol{\mathcal{H}}\boldsymbol{F}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{G}\right)\boldsymbol{\mathcal{G}} \end{bmatrix} \tag{75}$$

$$E\left[\boldsymbol{y}_{i,j}\boldsymbol{y}_{i,j}^T|\tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right] = Var\left[\boldsymbol{y}_{i,j}|\tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right] + E\left[\boldsymbol{y}_{i,j}|\tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right]E\left[\boldsymbol{y}_{i,j}|\tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right]^T \tag{76}$$

2) M-Step:

$$\boldsymbol{\mu} = \frac{1}{\boldsymbol{N}J_i}\sum_{i,j}\boldsymbol{x}_{i,j} \tag{77}$$

$$\boldsymbol{A} = \left(\sum_{i,j}\bar{\boldsymbol{x}}_{i,j}E[\boldsymbol{y}_{i,j}|\tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}]^T\right)\left(\sum_{i,j}E[\boldsymbol{y}_{i,j}\boldsymbol{y}_{i,j}^T|\tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}]\right)^{-1} \tag{78}$$

$$\boldsymbol{\Sigma} = \frac{1}{\boldsymbol{N}J_i}\sum_{i,j}\text{Diag}\left[\bar{\boldsymbol{x}}_{i,j}\bar{\boldsymbol{x}}_{i,j}^T - \boldsymbol{A}E[\boldsymbol{y}_{i,j}|\tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}]\bar{\boldsymbol{x}}_{i,j}^T\right] \tag{79}$$

D. Scalable Likelihood

$$\begin{aligned}\ln\left[Pr\left(\tilde{\boldsymbol{x}}_i|\boldsymbol{\Theta}\right)\right] = &-\frac{J_iD_x}{2}\ln\left[2\pi\right] \\ &-\frac{1}{2}\ln\left[\det\left(\boldsymbol{\Sigma}\right)^{J_i}\det\left(\boldsymbol{\mathcal{G}}^{-1}\right)^{J_i}\det\left(\boldsymbol{\mathcal{F}}_{J_i}^{-1}\right)\right] \\ &-\frac{1}{2}\sum_{j=1}^{J_i}\bar{\boldsymbol{x}}_{i,j}^T\boldsymbol{\mathcal{S}}\bar{\boldsymbol{x}}_{i,j} \\ &+\frac{1}{2}\left(\sum_{j=1}^{J_i}\bar{\boldsymbol{x}}_{i,j}^T\boldsymbol{\mathcal{S}}^T\boldsymbol{F}\right)\boldsymbol{\mathcal{F}}_{J_i}\left(\sum_{j=1}^{J_i}\boldsymbol{F}^T\boldsymbol{\mathcal{S}}\bar{\boldsymbol{x}}_{i,j}\right)\end{aligned} \tag{80}$$

REFERENCES

[1] D. V. Ouellette, "Schur Complements and Statistics," *Linear Algebra and its Applications*, vol. 36, pp. 187–295, 1981.
[2] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, 1st ed. Springer, oct 2007.