This is the Pre-Published Version.
The following publication Y. Jiang and H. F. Frank Leung, "The Scalable Version of Probabilistic Linear Discriminant Analysis and Its Potential as A Classifier for Audio Signal Classification," 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 2018 is available at https://doi.org/10.1109/IJCNN.2018.8488995.

The Scalable Version of Probabilistic Linear Discriminant Analysis and Its Potential as A Classifier for Audio Signal Classification

Yuechi Jiang
Electronic and Information Engineering
The Hong Kong Polytechnic University
Hong Kong, China
yuechi.jiang@connect.polyu.hk

Frank H. F. Leung
Electronic and Information Engineering
The Hong Kong Polytechnic University
Hong Kong, China
frank-h-f.leung@polyu.edu.hk

***Abstract*—Probabilistic Linear Discriminant Analysis (PLDA) has exhibited good performance in face recognition and speaker verification. However, it is not widely used as a general-purpose classifier. The major limitation of PLDA lies in that, in the original formulation, the modeling part and the prediction part require the inversion of large matrices, whose sizes are proportional to the number of training vectors in a class. The original formulation of PLDA is not scalable if there are many training vectors, because the matrices will become too large to be inverted. In the literature, some scalable versions for the modeling part have been proposed. In this paper, we propose the scalable version for the prediction part, which completes the scalable version of PLDA. This makes PLDA able to handle a large number of training data, enabling PLDA to be used as a general-purpose classifier for different classification tasks. We then apply PLDA as the classifier to three different audio signal classification tasks, and compare its performance with Support Vector Machine (SVM), which is a widely used general-purpose classifier. Experimental results show that PLDA performs very well and can be even better than SVM, in terms of classification accuracy.**

Keywords—probabilistic linear discriminant analysis, audio signal classification

I. Introduction

Probabilistic Linear Discriminant Analysis (PLDA) was first proposed in [1] for face recognition. In the PLDA model, a feature is supposed to be generated by a between-class latent variable and a within-class latent variable. The between-class latent variable reflects the difference between different classes, and therefore is the same for the same class. The within-class latent variable reflects the difference between the features within the same class, and therefore can be different even for the same class. PLDA has been shown to outperform many state-of-the-art methods in face recognition [1][2]. Later, the usage of PLDA was extended to speaker verification as an alternative to Support Vector Machine (SVM) [3]. In this paper, we extend the application of PLDA to more audio signal classification tasks, including microphone identification, telephone session identification, and speaker identification. To enable PLDA to be used as a general-purpose classifier, we propose the scalable version of PLDA. We then compare the performance of PLDA and SVM in the aforementioned three audio signal classification tasks.

In the literature, SVM has been applied to many different audio signal classification tasks, such as speaker verification [3], mobile phone identification [4][5], and microphone identification [6][7]; however, PLDA is not as widely used as SVM. The limitation of PLDA is that, during modeling and prediction, finding the inverse of large matrices is needed, and the sizes of the matrices are proportional to the dimensionality and the number of training vectors. This matrix inversion operation is difficult or even infeasible to be performed if there are many training vectors. Therefore, the application of PLDA is limited to small-scale classification tasks, such as face recognition and speaker verification, where the number of training data is small.

PLDA consists of a modeling part and a prediction part. In the literature, a scalable version of the modeling part for PLDA was given in [8], which made the operation of matrix inversion more efficient. In [9], a variable changing scheme was adopted to derive a scalable version of the modeling part. In this paper, we give the scalable version of the prediction part for PLDA, which completes the scalable version of PLDA. The scalable version of PLDA is able to handle a large number of training data, and therefore has the potential to be used as a general-purpose classifier for different classification tasks. We then compare the performance of the scalable version PLDA and SVM as the classifier in doing microphone identification, telephone session identification, and speaker identification. Microphone identification aims at identifying which microphone is used to record a speech recording; telephone session identification aims at identifying when the speech is recorded; speaker identification aims at identifying who gives the speech. We use Gaussian Supervector (GSV) [6][10] and i-vector [11] as the feature vector, which are widely used in audio signal classification tasks.

This paper is organized as follows. In Section II, we give the original formulation of the modeling part and the prediction part for PLDA. In Section III, we give the scalable formulation of the modeling part and the prediction part for the scalable version of PLDA, and briefly justify the scalability. In Section IV, we briefly describe the datasets used in our experiments.

The work described in this paper was substantially supported by a grant from The Hong Kong Polytechnic University (Project Account Code: RUG7).

©2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

In Section V, we compare the performance of the scalable PLDA and SVM as the classifier in different audio signal classification tasks, and give some comments. A conclusion will be drawn in Section VI.

II. Original Formulation of PLDA

Suppose we have a set of training feature vectors denoted as $\{\boldsymbol{x}_{11}, \boldsymbol{x}_{12} \ldots \boldsymbol{x}_{1J}, \boldsymbol{x}_{21}, \boldsymbol{x}_{22} \ldots \boldsymbol{x}_{2J} \ \ldots \ \boldsymbol{x}_{K1}, \boldsymbol{x}_{K2} \ldots \boldsymbol{x}_{KJ}\}$, where $\boldsymbol{x}_{kj}$ denotes the j-th training vector in the k-th class, K denotes the total number of classes, and J denotes the number of training feature vectors in a class. Then in the PLDA model, $\boldsymbol{x}_{kj}$ is expressed as in (1) below, where $\boldsymbol{\mu}$ is the global mean, $\boldsymbol{F}$ and $\boldsymbol{G}$ are two factor loading matrices, $\boldsymbol{h}_k$ is the between-class latent variable, $\boldsymbol{w}_{kj}$ is the within-class latent variable, and $\boldsymbol{\varepsilon}_{kj}$ is a noise term [1]. The between-class latent variable $\boldsymbol{h}_k$ is only class dependent, meaning that all the feature vectors in the same class share the same between-class latent variable, while the within-class latent variable $\boldsymbol{w}_{kj}$ is sample dependent, meaning that different feature vectors have different within-class latent variables even if they are in the same class.

$$x_{kj} = \mu + Fh_k + Gw_{kj} + \varepsilon_{kj} \tag{1}$$

PLDA model is a special type of factor analysis model, and thus it complies with the assumptions in factor analysis, namely, $\boldsymbol{h}_k$, $\boldsymbol{w}_{kj}$ and $\boldsymbol{\varepsilon}_{kj}$ follow normal distribution with zero mean and diagonal covariance, as given in (2) ~ (4) below, where $\boldsymbol{I}$ is an identity matrix, and $\boldsymbol{\Sigma}$ is a diagonal matrix. Since $\boldsymbol{h}_k$, $\boldsymbol{w}_{kj}$ and $\boldsymbol{\varepsilon}_{kj}$ are assumed to be independent, $\boldsymbol{x}_{kj}$ also follows a normal distribution as given in (5).

$$p(h_k) = N(h_k \mid 0, I) \tag{2}$$

$$p(w_{kj}) = N(w_{kj} \mid 0, I) \tag{3}$$

$$p(\varepsilon_{kj}) = N(\varepsilon_{kj} \mid 0, \Sigma) \tag{4}$$

$$p(x_{kj}) = N(x_{kj} \mid 0, FF^T + GG^T + \Sigma) \tag{5}$$

As all the feature vectors in class k $\{\boldsymbol{x}_{k1}, \boldsymbol{x}_{k2} \ldots \boldsymbol{x}_{kJ}\}$ share the same $\boldsymbol{h}_k$, they are combined to form a super vector and considered as a whole, as given in (6) [1]. Eq. (6) can be reformulated into a compact form as given by (7), where $\boldsymbol{X}_k$ is the concatenation of $\{\boldsymbol{x}_{k1}, \boldsymbol{x}_{k2} \ldots \boldsymbol{x}_{kJ}\}$ in a column-wise manner, $\boldsymbol{U}$ is the concatenation of J's $\boldsymbol{\mu}$ in a column-wise manner, $\boldsymbol{R}$ consists of J's $\boldsymbol{F}$ and J's $\boldsymbol{G}$, $\boldsymbol{Y}_k$ is the concatenation of $\boldsymbol{h}_k$ and $\{\boldsymbol{w}_{k1}, \boldsymbol{w}_{k2} \ldots \boldsymbol{w}_{kJ}\}$ in a column-wise manner, and $\boldsymbol{\zeta}_k$ is the concatenation of $\{\boldsymbol{\varepsilon}_{k1}, \boldsymbol{\varepsilon}_{k2} \ldots \boldsymbol{\varepsilon}_{kJ}\}$ in a column-wise manner.

$$\begin{bmatrix} x_{k1} \\ x_{k2} \\ \vdots \\ x_{kJ} \end{bmatrix} = \begin{bmatrix} \mu \\ \mu \\ \vdots \\ \mu \end{bmatrix} + \begin{bmatrix} F & G & 0 & \cdots & 0 \\ F & 0 & G & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ F & 0 & 0 & \cdots & G \end{bmatrix} \begin{bmatrix} h_k \\ w_{k1} \\ w_{k2} \\ \vdots \\ w_{kJ} \end{bmatrix} + \begin{bmatrix} \varepsilon_{k1} \\ \varepsilon_{k2} \\ \vdots \\ \varepsilon_{kJ} \end{bmatrix} \tag{6}$$

$$X_k = U + RY_k + \zeta_k \tag{7}$$

Since $\boldsymbol{x}_{kj}$ follows the normal distribution as given in (5), the joint distribution of $\{\boldsymbol{x}_{k1}, \boldsymbol{x}_{k2} \ldots \boldsymbol{x}_{kJ}\}$ is also a normal distribution as given in (8) [2], where $\boldsymbol{\Phi}$ is the covariance of $\boldsymbol{\zeta}_k$, which consists of J's $\boldsymbol{\Sigma}$, as given in (9).

$$p(x_{k1}, x_{k2} \ldots x_{kJ}) = p(X_k) = N(X_k \mid U, RR^T + \Phi) \tag{8}$$

where

$$\Phi = \begin{bmatrix} \Sigma & 0 & \cdots & 0 \\ 0 & \Sigma & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Sigma \end{bmatrix} \tag{9}$$

A. Modeling Part of PLDA (Training)

Eq. (7) is the standard form of factor analysis, and the model parameters $\boldsymbol{U}$, $\boldsymbol{R}$ and $\boldsymbol{\Phi}$ can be estimated using the Expectation-Maximization (EM) algorithm [1][12]. In the E-step, we calculate the expected mean and the expected covariance of $\boldsymbol{Y}_k$, both conditioned on $\boldsymbol{X}_k$, as given in (10) and (11) respectively [1].

$$E[Y_k] = (I + R^T \Phi^{-1} R)^{-1} R^T \Phi^{-1} (X_k - U) \tag{10}$$

$$E[Y_k Y_k^T] = (I + R^T \Phi^{-1} R)^{-1} + E[Y_k] E[Y_k]^T \tag{11}$$

Note that (1) can also be reformulated as the standard form of factor analysis as in (12) below, where $\boldsymbol{V}$ consists of $\boldsymbol{F}$ and $\boldsymbol{G}$, and $\boldsymbol{z}_{kj}$ is the concatenation of $\boldsymbol{h}_k$ and $\boldsymbol{w}_{kj}$.

$$x_{kj} = \mu + \begin{bmatrix} F & G \end{bmatrix} \begin{bmatrix} h_k \\ w_{kj} \end{bmatrix} + \varepsilon_{kj} = \mu + Vz_{kj} + \varepsilon_{kj} \tag{12}$$

Originally, we need to estimate $\boldsymbol{U}$, $\boldsymbol{R}$ and $\boldsymbol{\Phi}$ in the factor analysis model in (7). However, noting the fact that $\boldsymbol{U}$ is made up of $\boldsymbol{\mu}$, $\boldsymbol{R}$ is made up of $\boldsymbol{F}$ and $\boldsymbol{G}$, $\boldsymbol{\Phi}$ is made up of $\boldsymbol{\Sigma}$, we can then estimate $\boldsymbol{\mu}$, $\boldsymbol{V}$ and $\boldsymbol{\Sigma}$ for the factor analysis model in (12). Then in the M-step, we calculate $\boldsymbol{\mu}$, $\boldsymbol{V}$ and $\boldsymbol{\Sigma}$, using (13) ~ (15) below, where K is the total number of classes and J is the number of training feature vectors in a class, and *diag*(.) is an

operation that sets all the non-diagonal elements to be zero. The expected mean $E[z_{kj}]$ and the expected covariance $E[z_{kj}z_{kj}^T]$, both conditioned on $\boldsymbol{x}_{kj}$, can be obtained from $E[\boldsymbol{Y}_k]$ and $E[\boldsymbol{Y}_k\boldsymbol{Y}_k^T]$ by considering the equivalence of (6) and (7) [1].

$$\mu = \frac{1}{KJ}\sum_{k=1}^{K}\sum_{j=1}^{J} x_{kj} \tag{13}$$

$$V = \left(\sum_{k=1}^{K}\sum_{j=1}^{J}(x_{kj}-\mu)E[z_{kj}]^T\right)\left(\sum_{k=1}^{K}\sum_{j=1}^{J}E[z_{kj}z_{kj}^T]\right)^{-1} \tag{14}$$

$$\Sigma = \frac{1}{KJ}\sum_{k=1}^{K}\sum_{j=1}^{J} diag\left((x_{kj}-\mu)(x_{kj}-\mu)^T - VE[z_{kj}](x_{kj}-\mu)^T\right) \tag{15}$$

Each EM iteration includes one E-step followed by one M-step. In the E-step, we calculate the expectations using (10) and (11), while in the M-step, we re-estimate the model parameters using (13) ~ (15). In the modeling stage of PLDA, the EM algorithm is usually executed for a predefined number of iterations.

B. Prediction Part of PLDA (Classification)

After the estimation of model parameters $\theta=\{\boldsymbol{\mu}, \boldsymbol{F}, \boldsymbol{G}, \boldsymbol{\Sigma}\}$ using the training feature vectors, for a given test feature vector $\boldsymbol{x}_t$, the joint distribution of $\boldsymbol{x}_t$ and all the training vectors $\{\boldsymbol{x}_{k1}, \boldsymbol{x}_{k2} \ldots \boldsymbol{x}_{kJ}\}$ in class k is supposed to be a normal distribution [2], given by (16), where $\boldsymbol{U}'$, $\boldsymbol{R}'$ and $\boldsymbol{\Phi}'$ are given by (17).

$$\begin{aligned} &p(x_t, x_{k1}, x_{k2}...x_{kJ}) = p(x_t, X_k) \\ &= N(x_t, X_k \mid U', R'R'^T + \Phi') \end{aligned} \tag{16}$$

where

$$U' = \begin{bmatrix} U \\ \mu \end{bmatrix}, \quad R' = \begin{bmatrix} \begin{bmatrix} & R & \end{bmatrix} & \begin{matrix} 0 \\ 0 \\ \vdots \end{matrix} \\ F \quad 0 \quad \cdots \quad G \end{bmatrix}, \quad \Phi' = \begin{bmatrix} \Phi & 0 \\ 0 & \Sigma \end{bmatrix} \tag{17}$$

In (16), the covariance $\boldsymbol{R}'\boldsymbol{R}'^T+\boldsymbol{\Phi}'$ and the mean $\boldsymbol{U}'$ can also be expressed as in (18), where $\boldsymbol{U}_a$ and $\boldsymbol{\Sigma}_{aa}$ are the mean and covariance of $\boldsymbol{x}_t$, and $\boldsymbol{U}_b$ and $\boldsymbol{\Sigma}_{bb}$ are the mean and covariance of $\boldsymbol{X}_k$, which are given in (19).

$$U' = \begin{bmatrix} U_a \\ U_b \end{bmatrix}, \quad R'R'^T + \Phi' = \begin{bmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{bmatrix} \tag{18}$$

where

$$\begin{aligned} &U_a = \mu, \quad U_b = U \\ &\Sigma_{aa} = FF^T + GG^T + \Sigma, \quad \Sigma_{ab} = \Sigma_{ba}^T = \begin{bmatrix} FF^T & \cdots & FF^T \end{bmatrix} \\ &\Sigma_{bb} = \begin{bmatrix} \Sigma_{aa} & FF^T & FF^T & \cdots \\ FF^T & \Sigma_{aa} & FF^T & \cdots \\ FF^T & FF^T & \Sigma_{aa} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \end{aligned} \tag{19}$$

Then the conditional probability of $\boldsymbol{x}_t$ with respect to class k is given by (20), where $\boldsymbol{U}_{a|b}$ is the conditional mean and $\boldsymbol{\Sigma}_{a|b}$ is the conditional covariance [13]. After calculating the conditional probabilities with respect to all the K classes, $\boldsymbol{x}_t$ can then be classified to the class with the highest conditional probability [2].

$$\begin{aligned} &p(x_t \mid X_k) = N(x_t \mid U_{a|b}, \Sigma_{a|b}) \\ &= N(x_t \mid U_a + \Sigma_{ab}\Sigma_{bb}^{-1}(X_k - U_b), \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba}) \end{aligned} \tag{20}$$

III. The Scalable Version of PLDA

A. Modeling Part of Scalable PLDA

In the E-step of the modeling part for the original PLDA (i.e. (10) and (11)), the inverse of a large matrix $(\boldsymbol{I}+\boldsymbol{R}^T\boldsymbol{\Phi}^{-1}\boldsymbol{R})$ has to be found, whose dimensionality is proportional to the number of training vectors in a class. In [8], a scalable version of the modeling part of PLDA has been given, which makes the computation with a large number of training vectors in a class more efficient. In [8], the formulae of E-step are given by (21) ~ (23), while the M-step (i.e. (13) ~ (15)) is unchanged.

$$E[z_{kj}] = \begin{bmatrix} E[h_k] \\ E[w_{kj}] \end{bmatrix} = \begin{bmatrix} (MF^T\Sigma^{-1} - M\Lambda^T G^T\Sigma^{-1})\sum_{j=1}^{J}(x_{kj}-\mu) \\ L^{-1}G^T\Sigma^{-1}(x_{kj}-\mu) - \Lambda E[h_k] \end{bmatrix} \tag{21}$$

$$E[z_{kj}z_{kj}^T] = \begin{bmatrix} M & -M\Lambda^T \\ -\Lambda M & L^{-1} + \Lambda M\Lambda^T \end{bmatrix} + E[z_{kj}]E[z_{kj}]^T \tag{22}$$

where

$$\begin{aligned} &L = I + G^T\Sigma^{-1}G \\ &\Lambda = L^{-1}G^T\Sigma^{-1}F \\ &M = \left(I + \sum_{j=1}^{J} F^T\Sigma^{-1}(F - G\Lambda)\right)^{-1} \end{aligned} \tag{23}$$

B. Prediction Part of Scalable PLDA

To calculate the conditional probability in (20), the inverse of a large matrix $\boldsymbol{\Sigma}_{bb}$ is needed to be found, whose dimensionality is proportional to the number of training vectors in a class.

Noting that $\boldsymbol{\Sigma}_{bb}$ is a symmetric matrix, thus the inverse of $\boldsymbol{\Sigma}_{bb}$ should also have a symmetric form as given by (24), where $\boldsymbol{P}$ and $\boldsymbol{Q}$ are symmetric matrices. Using (24), instead of finding the inverse of the large matrix $\boldsymbol{\Sigma}_{bb}$ directly, we can now find two small matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$. By multiplying $\boldsymbol{\Sigma}_{bb}$ to its inverse and equating the result to an identity matrix, as given by (25), $\boldsymbol{P}$ and $\boldsymbol{Q}$ can then be solved.

$$\Sigma_{bb}^{-1} = \begin{bmatrix} \Sigma_{aa} & FF^T & FF^T & \cdots \\ FF^T & \Sigma_{aa} & FF^T & \cdots \\ FF^T & FF^T & \Sigma_{aa} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}^{-1} = \begin{bmatrix} P & Q & Q & \cdots \\ Q & P & Q & \cdots \\ Q & Q & P & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{24}$$

$$\begin{bmatrix} \Sigma_{aa} & FF^T & FF^T & \cdots \\ FF^T & \Sigma_{aa} & FF^T & \cdots \\ FF^T & FF^T & \Sigma_{aa} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} P & Q & Q & \cdots \\ Q & P & Q & \cdots \\ Q & Q & P & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots \\ 0 & 1 & 0 & \cdots \\ 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{25}$$

By equating the block on the principal diagonal on both sides of (25), we have (26). By equating the blocks not on the principal diagonal (e.g. the block on the first row and second column) on both sides of (25), we have (27).

$$\begin{aligned} & \Sigma_{aa} P + \sum_{j=1}^{J-1} FF^T Q = I \\ & \Leftrightarrow \\ & (FF^T + GG^T + \Sigma) P + \sum_{j=1}^{J-1} FF^T Q = I \end{aligned} \tag{26}$$

$$\begin{aligned} & \Sigma_{aa} Q + FF^T P + \sum_{j=1}^{J-2} FF^T Q = 0 \\ & \Leftrightarrow \\ & (FF^T + GG^T + \Sigma) Q + FF^T P + \sum_{j=1}^{J-2} FF^T Q = 0 \end{aligned} \tag{27}$$

Then $\boldsymbol{P}$ and $\boldsymbol{Q}$ can be solved easily by considering (26) and (27) together, as given in (28).

$$\begin{cases} P = \left(FF^T + GG^T + \Sigma - \sum_{j=1}^{J-1} FF^T W^{-1} FF^T \right)^{-1} \\ Q = -W^{-1} FF^T P \end{cases} \tag{28}$$

where $W = GG^T + \Sigma + \sum_{j=1}^{J-1} FF^T$

The conditional mean $\boldsymbol{U}_{a|b}$ and conditional covariance $\boldsymbol{\Sigma}_{a|b}$ used in (20) for prediction can be obtained using (29) and (30) respectively. Having obtained $\boldsymbol{U}_{a|b}$ and $\boldsymbol{\Sigma}_{a|b}$, we can then efficiently calculate the conditional probability using (20).

$$\begin{aligned} U_{a|b} &= U_a + \Sigma_{ab} \Sigma_{bb}^{-1} (X_k - U_b) \\ &= \mu + \begin{bmatrix} FF^T & \cdots & FF^T \end{bmatrix} \begin{bmatrix} P & Q & \cdots \\ Q & P & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} x_{k1} - \mu \\ \vdots \\ x_{kJ} - \mu \end{bmatrix} \\ &= \mu + \left(FF^T P + \sum_{j=1}^{J-1} FF^T Q \right) \sum_{j=1}^{J} (x_{kj} - \mu) \end{aligned} \tag{29}$$

$$\begin{aligned} \Sigma_{a|b} &= \Sigma_{aa} - \Sigma_{ab} \Sigma_{bb}^{-1} \Sigma_{ba} \\ &= \Sigma_{aa} - \begin{bmatrix} FF^T & \cdots & FF^T \end{bmatrix} \begin{bmatrix} P & Q & \cdots \\ Q & P & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} FF^T \\ \vdots \\ FF^T \end{bmatrix} \\ &= \left(FF^T + GG^T + \Sigma \right) - \left(FF^T P + \sum_{j=1}^{J-1} FF^T Q \right) \sum_{j=1}^{J} FF^T \end{aligned} \tag{30}$$

C. Brief Comparison between Original PLDA and Scalable PLDA

Regarding the modeling part, in the E-step, for the original PLDA, it can be seen from (10) and (11) that, the inverse of a large matrix ($\boldsymbol{I}+\boldsymbol{R}^T\boldsymbol{\Phi}^{-1}\boldsymbol{R}$) has to be calculated. Suppose the dimensionality of $\boldsymbol{F}$ and $\boldsymbol{G}$ are $D\times D$, then according to (6) and (7), the dimensionality of ($\boldsymbol{I}+\boldsymbol{R}^T\boldsymbol{\Phi}^{-1}\boldsymbol{R}$) will be $(J+1)D\times(J+1)D$, where J is the number of training vectors in a class. If J is too large, meaning that there are a large number of training vectors in a class, the inversion operation may be even infeasible if the size of the matrix exceeds the memory of the computer. On the contrary, for the scalable PLDA, as can be seen from (21) ~ (23), only $\boldsymbol{F}$, $\boldsymbol{G}$ and $\boldsymbol{\Sigma}$ are involved in calculation, whose sizes are independent of J, meaning that even the number of training vectors is very large, the calculation is still feasible.

Regarding the prediction part, for the original PLDA, the probability that a feature vector is classified to a class is given by (20). As can be seen from (20), the inverse of a large matrix $\boldsymbol{\Sigma}_{bb}$ has to be calculated. Suppose the dimensionality of $\boldsymbol{\Sigma}$ is $D\times D$, then according to (19), the dimensionality of $\boldsymbol{\Sigma}_{bb}$ will be $JD\times JD$, where J is the number of training vectors in a class. Similar to the modeling part, if J is too large, it may be even infeasible to calculate the inverse of $\boldsymbol{\Sigma}_{bb}$. Instead of involving $\boldsymbol{\Sigma}_{bb}$ in the calculation of the conditional probability, it is also possible to directly use $\boldsymbol{U}_{a|b}$ and $\boldsymbol{\Sigma}_{a|b}$ to calculate the conditional probability, according to (20). Then for the scalable PLDA, $\boldsymbol{U}_{a|b}$ and $\boldsymbol{\Sigma}_{a|b}$ can be calculated efficiently using (29) and (30), where only $\boldsymbol{F}$, $\boldsymbol{G}$, $\boldsymbol{\Sigma}$, $\boldsymbol{P}$ and $\boldsymbol{Q}$ are involved in calculation, whose sizes are independent of J. This means that even J is very large, it is still feasible to calculate the conditional probability and make prediction.

TABLE I. MICROPHONE IDENTIFICATION DATASET

Notation	Microphone Model	Number of Speech Recordings	
		Training	*Testing*
M1	AKG C410B Head Mounted	240	260
M2	AKH D80S Desktop	240	260
M3	SONY ECM 66B Lapel	240	260
M4	TARGET Lapel	240	260
UBM	All the models	599	

TABLE II. TELEPHONE SESSION IDENTIFICATION DATASET

Notation	Telephone Session	Number of Speech Recordings	
		Training	*Testing*
T1	Session 1	240	259
T2	Session 2	240	260
T3	Session 3	240	260
UBM	All the sessions	300	

IV. DATASETS

In this paper, we consider three audio classification tasks, which are 1) microphone identification aiming at identifying which device is used to record the speech signal, and 2) telephone session identification aiming at identifying when the speech is recorded, and 3) speaker identification aiming at identifying who gives the speech. The datasets are obtained from AHUMADA [14]. In microphone identification dataset, there are 4 different microphones to be identified; in telephone session identification dataset, there are 3 different sessions to be identified; in speaker identification dataset, there are 25 different speakers to be identified.

Each dataset is divided into a training set, a testing set, and a Universal Background Model (UBM) set. The UBM set is used to calculate GSV and i-vector. For microphone identification, 599 microphone speeches are used for UBM, 960 microphone speeches are used for training, and 1040 microphone speeches are used for testing. For telephone session identification, 300 telephone speeches are used for UBM, 720 telephone speeches are used for training, and 779 telephone speeches are used for testing. For speaker identification, 421 telephone speeches are used for UBM, 990 telephone speeches are used for training, and 988 telephone speeches are used for testing. Details are shown in Tables I ~ III.

V. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this paper, we use GSV [6][7] and i-vector [11] as the feature vector, and employ scalable PLDA and linear SVM as the classifier. GSV and i-vector are obtained based on a 32-mixture Universal Background Model (UBM). The UBM is constructed using the mixture splitting technique [15] and the EM algorithm [16]. For GSV, several different relevance factors r are investigated; for i-vector, the dimensionality is set to be half of that of GSV. SVM is implemented using LIBSVM [17] with default parameters. When estimating the PLDA model parameters $\theta=\{\boldsymbol{\mu}, \boldsymbol{F}, \boldsymbol{G}, \boldsymbol{\Sigma}\}$ using the EM algorithm, the columns of $\boldsymbol{F}$ and $\boldsymbol{G}$ are initialized to be the result of Singular Value Decomposition (SVD) of the between-class scatter matrix $\boldsymbol{S}_B$ and the within-class scatter matrix $\boldsymbol{S}_W$ of the training vectors, respectively; $\boldsymbol{\Sigma}$ is initialized to be the diagonalized covariance matrix of the training vectors [9]. Microphone identification results, telephone session identification results and speaker identification results are shown in Figs. 1 ~ 3 respectively.

TABLE III. SPEAKER IDENTIFICATION DATASET

Notation	Speaker Index	Number of Speech Recordings	
		Training	*Testing*
L001	Speaker 1	40	40
L002	Speaker 2	40	40
L003	Speaker 3	40	40
L004	Speaker 4	40	40
L005	Speaker 5	40	40
L006	Speaker 6	40	40
L007	Speaker 7	30	30
L008	Speaker 8	40	39
L009	Speaker 9	40	40
L010	Speaker 10	40	40
L011	Speaker 11	40	40
L012	Speaker 12	40	40
L013	Speaker 13	40	40
L014	Speaker 14	40	40
L015	Speaker 15	40	40
L016	Speaker 16	40	40
L017	Speaker 17	40	40
L018	Speaker 18	40	40
L019	Speaker 19	40	40
L020	Speaker 20	40	40
L021	Speaker 21	40	40
L022	Speaker 22	40	39
L023	Speaker 23	40	40
L024	Speaker 24	40	40
L025	Speaker 25	40	40
UBM	All the speakers	421	

From Fig. 1, it can be seen that, on using GSV as the feature vector, PLDA can outperform SVM, but the performance of PLDA degrades with the increase of the number of EM iterations during modeling. While on using i-vector as the feature vector, PLDA also works better than SVM when the number of EM iterations is small, but the performance of PLDA drops rapidly if there are too many EM iterations. On employing SVM as the classifier, GSV and i-vector gives similar performances. While on employing PLDA as the classifier, when there are only a few EM iterations during modeling, GSV outperforms i-vector; when there are more EM iterations, GSV tends to perform worse than i-vector. However, GSV tends to be more stable than i-vector with different numbers of EM iterations.

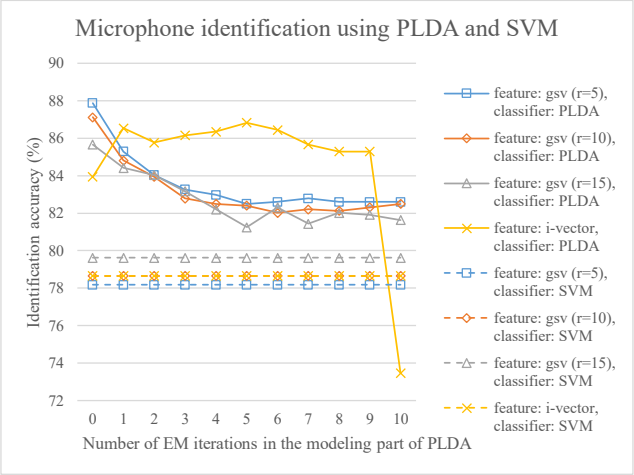

Fig. 1. Microphone identification results.

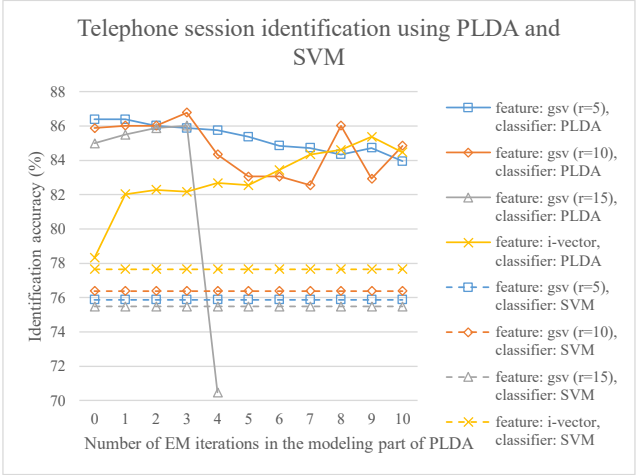

Fig. 2. Telephone session identification results.

From Fig. 2, generally speaking, no matter the feature vector is GSV or i-vector, PLDA can work better than SVM. On using GSV as the feature vector, the performance of PLDA tends to degrade with the increase of the number of EM iterations during modeling. While on the contrary, on using i-vector as the feature vector, the performance of PLDA tends to be improved with the increase of the number of EM iterations during modeling.

From Fig. 1 and Fig. 2, it can be seen that, for different applications (i.e. microphone identification or telephone session identification), the performances of GSV and i-vector differ (sometimes GSV outperforms i-vector while sometimes i-vector works better). In fact, although GSV and i-vector are both calculated based on a UBM, they are intrinsically rather different. GSV is obtained by adapting the mean vectors in the UBM, while i-vector is obtained by applying factor analysis on the UBM. As shown before, PLDA is essentially a factor analysis model, so when combining i-vector and PLDA, factor analysis is performed twice. This causes the different behaviors of GSV and i-vector on employing PLDA as the classifier. On the contrary, the performances of GSV and i-vector are quite similar on employing SVM. It is also noticed that there is a sudden drop in performance on using PLDA (for i-vector in Fig. 1 and GSV (r=15) in Fig. 2). As can be seen from (1) ~ (5), in the PLDA model, the latent variables hidden in the feature vector is assumed to follow normal distributions, however, this assumption may not be always true for different types of feature vectors in different applications. Therefore, the usage and effectiveness of PLDA may also be dependent on the choice of feature vectors as well as the type of applications.

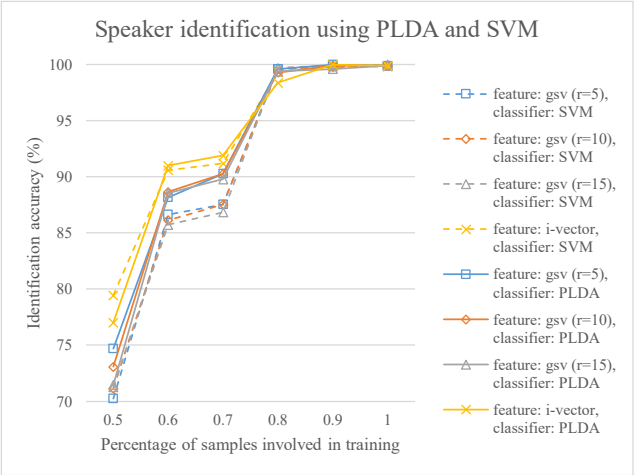

Fig. 3. Speaker identification results.

Generally speaking, on using different feature vectors, PLDA tends to give a better performance than SVM. This may be owing to the intrinsic characteristics of PLDA. As can be seen from (1), essentially, PLDA is a factor analysis model, but the between-class latent variable and the within-class latent variable provide similar functionalities to the between-class scatter matrix $\boldsymbol{S}_B$ and the within-class scatter matrix $\boldsymbol{S}_W$ used in Linear Discriminant Analysis (LDA). From this point of view, PLDA is the fusion of factor analysis and LDA. Factor analysis is an unsupervised modeling technique, while LDA is a supervised modeling technique. From this perspective, we see that PLDA is actually a combination of supervised and unsupervised modeling techniques.

The correlation between the performance of PLDA and the number of EM iterations during modeling of PLDA can be explained as follows. In the initialization stage, $\boldsymbol{F}$ and $\boldsymbol{G}$ are initialized to be the result of the SVD of $\boldsymbol{S}_B$ and $\boldsymbol{S}_W$, respectively. In this stage, PLDA is more like LDA which aims to find the eigenvectors of $S_W^{-1}S_B$ [18], or the Regularized Fisher Discriminant Analysis (RFDA) which aims to find the eigenvectors of $S_B - \alpha S_W$ [7]. After initialization, as the number of EM iterations increases, PLDA tends to be more like factor analysis. So, there is a trade-off between the supervised modeling ability and the unsupervised modeling ability of PLDA, and this balance is influenced by the number of EM iterations during modeling.

In Fig. 3, the performances of SVM and PLDA in doing speaker identification are compared, with different numbers of training data. The modeling part of PLDA only involves 1 EM iteration. It can be seen that, the performances of SVM and

PLDA are improved with the increase of the number of training data, which is reasonable. With all the training data involved in modeling, both SVM and PLDA work very well. On using GSV as the feature vector, PLDA tends to work better than SVM if the number of training data is small. On using i-vector as the feature vector, the performances of PLDA and SVM tend to be similar with different numbers of training data. In addition, it seems i-vector tends to work better than GSV if the number of training data is small, nevertheless, the performance is application dependent.

In terms of the speed, the modeling part of PLDA can be slow, as there may be many EM iterations. However, surprisingly, from the experimental results, we can see that only 1 EM iteration can be good enough. In this way, the modeling of PLDA (i.e. training) can be fast. Regarding the prediction part of PLDA, according to (20), (29) and (30), for a given testing feature vector, we only need to calculate a conditional Gaussian probability as given in (20), and the parameters of the conditional distribution can be pre-computed efficiently using (29) and (30). In this way, the prediction of PLDA (i.e. testing) can also be fast.

VI. Conclusion

In this paper, we give the formulation of the scalable version PLDA. In the literature, the scalable version of the modeling part of PLDA has been given. In this paper, we give the scalable version of the prediction part of PLDA, which completes the scalable version of PLDA. The scalable PLDA is able to handle a large number of training data, enabling it to be able to be used as a general-purpose classifier. We also compare the performance of the scalable PLDA and SVM as the classifier for three different audio signal classification tasks. In terms of classification accuracy, experimental results demonstrate that PLDA can outperform SVM in most cases. In terms of classification speed, according to the formulation of the prediction part of PLDA, the prediction process can be very fast; according to the experimental results, the modeling of PLDA can also be fast, as only several EM iterations are enough. In addition, on using PLDA to do classification, we notice an interesting phenomenon in training the PLDA model: the performance of PLDA is highly affected by the number of EM iterations during modeling. This phenomenon is owing to the fact that PLDA is the fusion of LDA and factor analysis. The fewer the EM iterations, the more PLDA will be like LDA; the more the EM iterations, the more PLDA will be like factor analysis.

References

[1] S. J. D. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2007, pp. 1-8.

[2] P. Li, Y. Fu, U. Mohammed, J. H. Elder, and S. J. D. Prince, "Probabilistic models for inference about identity," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 1, pp. 144-157, 2012.

[3] J. H. L. Hansen and T. Hasan, "Speaker recognition by machines and humans: a tutorial review," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 74-99, 2015.

[4] C. L. Kotropoulis, "Source phone identification using sketches of features," *IET Biometrics*, vol. 3, no. 2, pp. 75-83, 2014.

[5] Y. Jiang and F. H. F. Leung, "Mobile phone identification from speech recordings using weighted support vector machine," in *Proc. IEEE 42nd Annual Conf. on Industrial Electronics (IECON)*, 2016, pp. 963-968.

[6] D. G. Romero and C. Y. E. Wilson, "Automatic acquisition device identification from speech recordings," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2010, pp. 1806-1809.

[7] Y. Jiang and F. H. F. Leung, "Using regularized Fisher discriminant analysis to improve the performance of Gaussian supervector in session and device identification," in *Proc. IEEE Int. Joint Conf. on Neural Networks (IJCNN)*, 2017, pp. 705-712.

[8] Y. Jiang *et al.*, "PLDA modeling in i-vector and supervector space for speaker verification," in *Proc. 13th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2012.

[9] L. E. Shafey, C. McCool, R. Wallace, and S. Marcel, "A scalable formulation of probabilistic linear discriminant analysis: applied to face recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1788-1794, 2013.

[10] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 308-311, 2006.

[11] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788-798, 2011.

[12] C. M. Bishop, "Continuous latent variables," in *Pattern Recognition and Machine Learning*, Springer, 2006, ch. 12, pp. 559-603.

[13] C. M. Bishop, "Probability distributions," in *Pattern Recognition and Machine Learning*, Springer, 2006, ch. 2, pp. 67-136.

[14] J. O. Garcia, J. G. Rodriguez, and V. M. Aguiar, "AHUMADA: a large speech corpus in Spanish for speaker characterization and identification," *Speech Communication*, vol. 31, no. 2, pp. 255-264, 2000.

[15] S. Young *et al.*, *The HTK book (v3.4)*, Cambridge: Cambridge University Press, 2006, pp. 156-157.

[16] D. Reynolds, "Gaussian mixture models," in *Encyclopedia of Biometrics*, 2015, pp. 827-832.

[17] C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1-27, 2011.

[18] C. M. Bishop, "Linear models for classification," in *Pattern Recognition and Machine Learning*, Springer, 2006, ch. 4, pp. 179-224.