Journal of Machine Learning Research 20 (2019) 1-29 Submitted 3/18; Revised 11/18; Published 1/19

Joint PLDA for Simultaneous Modeling of Two Factors

Luciana Ferrer LFERRER@DC.UBA.AR
Instituto de Investigación en Ciencias de la Computación (ICC)
CONICET-Universidad de Buenos Aires
Pabellón I, Ciudad Universitaria, 1428, Ciudad Autónoma de Buenos Aires, Argentina

Mitchell McLaren MITCHELL.MCLAREN@SRI.COM
Speech Technology and Research Lab (StarLab)
SRI International
333 Ravenswood Ave, Menlo Park, 94025, United States

Editor: Barbara Engelhardt

Abstract

Probabilistic linear discriminant analysis (PLDA) is a method used for biometric problems like speaker or face recognition that models the variability of the samples using two latent variables, one that depends on the class of the sample and another one that is assumed independent across samples and models the within-class variability. In this work, we propose a generalization of PLDA that enables joint modeling of two sample-dependent factors: the class of interest and a nuisance condition. The approach does not change the basic form of PLDA but rather modifies the training procedure to consider the dependency across samples of the latent variable that models within-class variability. While the identity of the nuisance condition is needed during training, it is not needed during testing since we propose a scoring procedure that marginalizes over the corresponding latent variable. We show results on a multilingual speaker-verification task, where the language spoken is considered a nuisance condition. The proposed joint PLDA approach leads to significant performance gains in this task for two different data sets, in particular when the training data contains mostly or only monolingual speakers.

Keywords: Probabilistic linear discriminant analysis, speaker recognition, factor analysis, language variability, robustness to acoustic conditions

1. Introduction

PLDA was proposed by Prince (2007) for doing inferences about the identity of a person from an image of their face. A closely related model had been previously proposed by Ioffe (2006) and also tested on image processing tasks. The technique was later widely adopted by the speaker recognition community, becoming the state-of-the-art scoring technique for this task (Kenny, 2010; Burget et al., 2011; Brümmer, 2010a; Senoussaoui et al., 2011; Matejka et al., 2011). PLDA assumes that each sample[1] is represented by a feature vector of fixed dimension and that this vector is given by a sum of three terms: a term that depends on the class of the sample, a term that models the within-class variability and is assumed independent across samples, and a final term that models any remaining variability and is also independent across

1. Throughout this paper, the word sample is used to refer to an audio signal or recording containing speech.

samples. These assumptions imply that all samples from the same class are independent of each other and also independent of samples from other classes once the class is known.

In contrast with the assumptions made by PLDA, many training data sets consist of samples that come from a small set of distinct conditions. For example, many speaker recognition data sets contain only a few acoustic conditions (different microphones or noise conditions), speech styles (conversational, monologue, read), and languages. Samples corresponding to the same condition will most likely be statistically dependent.

The literature proposes a few approaches that generalize PLDA to consider metadata about the samples during training. The main motivation for these approaches, though, is not to relax the conditional independence assumption but rather to enable a more flexible model that can adapt to each of the available conditions rather than assuming that samples from all conditions can be modeled with the same linear model. Yet, a side effect of the proposed generalizations is the introduction of a dependency between samples from the same condition. The simplest approach of this family is to train a separate PLDA model for each condition, as proposed by Garcia-Romero et al. (2012). Nevertheless, in this paper, the authors show that pooling the data from all conditions, as proposed by Lei et al. (2012), leads to better performance than training separate models. This result is reasonable, since training separate PLDA models does not allow the overall model to learn how samples from the same class vary across conditions; only within-condition variation is learned.

The tied PLDA model proposed by Li et al. (2012) is designed to tackle this problem. In this approach, one PLDA model is trained for each condition, but these models are tied by forcing the latent variable corresponding to each class to be the same across all conditions. The approach was shown to outperform standard PLDA with pooled training data when each class in the training data is seen under both considered conditions, frontal and profile, in a face recognition task. A similar approach is proposed by Mak et al. (2016), but in this case, the mixture component is not given during training. Instead, the PLDA mixture components depend on a continuous metadata value, which is modeled with a mixture of Gaussians. The approach is tested by adding noise to the training data at different SNR levels. The resulting training data then contains samples for each speaker at different SNR levels. Under these conditions, the authors show gains from the proposed approach compared to pooling all the data to train a single PLDA model.

In this paper, we consider a scenario where each speaker in the training data is seen only under a small subset of the conditions present in the training set (potentially, only one). Further, we expect some conditions to have much less training data than others. Under this scenario, the tied PLDA approach does not work well, since it requires training a PLDA model of the same dimensions for each condition, which may be impossible or suboptimal for the conditions with less data. Further, the tied PLDA model can only learn how the nuisance conditions affect the classes of interest if it is provided samples for each class under different conditions during training.

We propose a novel generalization of the PLDA model that relaxes the conditional independence assumption without increasing the size of the parameter space, keeping the same functional form of the original PLDA model but modifying the training and scoring procedures to consider the dependency across samples originating from the sample's condition. In the proposed approach, which we call Joint PLDA (JPLDA), the condition is assumed to be known during training but not during testing. An expectation-maximization (EM) training

procedure is formulated that takes into account the condition of each sample. Scoring is performed, as in standard PLDA, by computing a likelihood ratio between the null hypothesis that the two sides of a trial belong to the same speaker versus the alternative hypothesis that the two sides belong to different speakers. The two likelihoods are computed by marginalizing over two hypotheses about the condition in both sides of a trial: that they are the same and that they are different. This way, we expect that the new model will be better at coping with same-condition versus different-condition trials than standard PLDA, since knowledge about the condition is used during training and implicitly considered during scoring. Further, we expect this model to behave better than tied PLDA under a training scenario where the number of samples is highly imbalanced across conditions and each speaker is seen only under one or a small subset of conditions.

The mathematical formulation for the proposed JPLDA approach was first published in arXiv (Ferrer, 2017), without results or analysis. A similar model with a significantly different formulation for the EM and scoring algorithms was later published, also in arXiv, by other authors (Shi et al., 2017). They show significant improvements from this approach on a text-dependent speaker verification task. In this paper, we show results and detailed analysis on two multilingual speaker recognition data sets, one composed of Mixer data (Cieri et al., 2007) from the speaker recognition evaluations organized by NIST and another that uses LASRS data (Beck et al., 2004). We evaluate two training scenarios, one using all available training data from the PRISM data set (Ferrer et al., 2011), which contains a small percentage of speakers speaking two different languages, and one where we subset the training data to contain only one language per speaker. We show that JPLDA significantly outperforms two standard PLDA approaches with different structures and tied PLDA, especially when the training data contains mostly or only a single language per speaker.

2. Standard PLDA Models

In this work, we adopt the nomenclature usually used by the speaker recognition community. Yet, the model proposed can be used for the original image processing task or any other task for which standard PLDA is used.

Standard PLDA (Prince, 2007) assumes that the vector m_i representing a certain sample from speaker s_i is given by

$$m_i = \mu + V y_{s_i} + U x_i + z_i, \tag{1}$$

where μ is a fixed bias; y_{s_i} is a vector of size R_y, the dimension of the speaker subspace; and x_i is a vector of size R_x, the dimension of the subspace corresponding to the nuisance condition or, as usually called in speaker recognition, the channel. The model assumes that

$$\begin{aligned} y_{s_i} &\sim N(0, I), \\ x_i &\sim N(0, I), \\ z_i &\sim N(0, D^{-1}), \end{aligned}$$

where the matrix D is assumed to be diagonal. All these latent variables are assumed independent: speaker variables are independent across speakers, and the nuisance variable x_i and noise variable z_i are independent across samples.

This model is equivalent (Sizov et al., 2014) to assuming the following distributions

$$m_i|\gamma_{s_i} \sim N(\gamma_{s_i}, UU^T + D^{-1}), \quad (2)$$

$$\gamma_{s_i} \sim N(\mu, VV^T). \quad (3)$$

where we can see that VV^T is the between-speaker and $UU^T + D^{-1}$ is the within-speaker covariance matrix of the m_i vectors. Note that, in the general case, the matrix VV^T could be singular. This fact, though, does not cause any theoretical or practical problems for the PLDA model.

The model described above corresponds to the original PLDA formulation, which we will call full PLDA (FPLDA). In speaker recognition, a simplified version of PLDA (SPLDA for the purpose of this paper) is more commonly used, where the nuisance term Ux_i is absorbed into the noise term, which is then assumed to have a full rather than diagonal covariance matrix. Sizov et al. (2014) gives a comprehensive explanation of the usual flavors of PLDA.

The training of PLDA parameters is done using an EM algorithm. The EM formulation for SPLDA and FPLDA can be found in two very detailed documents by Brümmer (2010a,b). For FPLDA, we use a minimum divergence (Kenny, 2010) step after every maximization step. This step is generally used to speed up convergence of the EM algorithm. For SPLDA, this step was not necessary for quick convergence since, as we will see, the smart initialization procedure described below already leads to an excellent model. We will not reproduce the EM formulas here, but we will describe the two initialization procedures we use, since they will be compared in the experimental section.

2.1. EM Initialization Procedure

The EM algorithm requires an initial model to start the iterations. This model can be generated randomly or, in the case of SPLDA, with a "smart" procedure that results in a much better initial model that, in turn, requires many fewer or no EM iterations to converge to the final parameters.

In our experiments, for random initialization, we set D to be an identity matrix, and V and U, when applicable, to be matrices with random elements drawn from a normal distribution with standard deviation 0.01 and mean 0.

For SPLDA, we also try a smart initialization approach that is directly motivated by Equations (2) and (3). Namely, we initialize V and D as follows

$$V = Q\Lambda^{-1/2},$$

$$D = W^{-1},$$

where W is the empirical within-speaker covariance matrix of the training data, and Q is a matrix with the eigenvectors corresponding to the R_y largest eigenvalues of the between-speaker covariance matrix of the training data and $\Lambda^{-1/2}$ is a diagonal matrix containing the square roots of those eigenvalues. As will be shown in the experiments, this initialization procedure works quite well, leading to models that are as good as those trained with many iterations of EM.

2.2. Scoring

In this work, we consider a verification task. Two sets of samples, an enrollment set E and a test set T, each corresponding to one or more samples from the same speaker, are compared

to decide whether the two speakers are the same or different. This comparison is usually called a *trial* in speaker verification. In some applications, a hard decision is needed; in others, a soft score is preferable. The PLDA paper (Prince, 2007) showed how to use this model to compute the likelihood ratio (LR) between the two hypotheses, which can be used directly as soft score or thresholded to make hard decisions if required. The LR is given by

$$LR = \frac{p(E,T|H_{SS})}{p(E,T|H_{DS})},$$

where H_{SS} is the hypothesis that the speakers in both sets are the same, while H_{DS} is the hypothesis that the speakers are different. This value can be computed using a closed form using the PLDA model. In our code we use the formulation derived by Cumani et al. (2014), Equation (34). Note, though, that the last term in that equation should not be there (this mistake was confirmed by one coauthor of the paper).

3. Tied PLDA Model

The tied PLDA model was introduced by Li et al. (2012). The model is a mixture of PLDA models where the latent variable corresponding to the speaker is tied across components. The vector representing sample i from speaker s_i is modeled as

$$m_i = \mu_{d_i} + V_{d_i} y_{s_i} + U_{d_i} x_i + z_i, \tag{4}$$

where d_i indicates the mixture component corresponding to sample i, and where

$$\begin{aligned} y_{s_i} &\sim N(0, I), \\ x_i &\sim N(0, I), \\ z_i &\sim N(0, D_{d_i}^{-1}). \end{aligned}$$

Hence, once the mixture component is given, the model reduces to a standard PLDA model. In this work, we assume that the mixture component is known both during training and during testing, as in the original work (Li et al., 2012), though the authors note that this is not a necessary condition. In the simplest case we could take the mixture component to be the nuisance condition of the sample but, as we will see, this might not be feasible if some conditions have too few training samples in which case grouping of samples from different conditions into the same component might be necessary. Note that the latent variable y_{s_i} does not depend on the component. Rather, this variable is tied for all samples from the same speaker across components. This enables the model to properly represent cross-component variability.

As for the original PLDA model, a simple PLDA model can be used instead of the full PLDA model for each component in the mixture. Further, the covariance matrix for the noise term can be either full or diagonal. In this work, each component is described by a SPLDA model for simplicity of implementation, since the difference between SPLDA and FPLDA is very small in practice.

The TPLDA model described by Li et al. (2012) and used here coincides with what Mak et al. (2016) calls SNR-dependent mixture PLDA model if we assume the SNR to be discrete rather than continuous so that the posterior probability for each component is fixed to 1 for the component corresponding to the sample, and to 0 otherwise. The training and scoring procedures for TPLDA can be found in the supplementary material for Mak et al. (2016).

4. Joint PLDA Model

In this work, we propose a generalization of the original PLDA model where the nuisance variable is no longer considered independent across samples, but potentially shared (tied) across samples that correspond to the same nuisance condition. This makes the model symmetric in the two latent variables corresponding to the speaker and the nuisance condition. To represent this dependency, we introduce a condition label for each sample, called c_i. Given this label, and the speaker label s_i, we propose to model vector m_i of dimension R_m for sample i as:

$$m_i = \mu + V y_{s_i} + U x_{c_i} + z_i, \tag{5}$$

where, as before, y_{s_i} is a vector of size R_y and x_{c_i} is a vector of size R_x, and

$$\begin{aligned} y_{s_i} &\sim N(0, I), \\ x_{c_i} &\sim N(0, I), \\ z_i &\sim N(0, D^{-1}). \end{aligned}$$

The model's parameters to estimate are $\lambda = \{\mu, V, U, D\}$, as in the standard PLDA formulation, but the input data for the training algorithm is now required to have a second set of labels indicating the nuisance condition of each sample.

The expectation-maximization equations for training this new model are significantly more involved than for the original PLDA model. This is due to the fact that each speaker cannot be treated separately from the others since samples from one speaker might be dependent on samples from a different speaker. This creates a potential dependency between all training samples, which greatly complicates the formulation, increasing the computation time by orders of magnitude for each EM iteration. Nevertheless, as we will see in the experiments, initializing the model in a smart way basically makes EM unnecessary in our experiments, reducing the training time of the model to the same order of what is required to train standard PLDA on the same data. A detailed derivation of the EM algorithm and scoring procedure for JPLDA is given by Ferrer (2017). In Appendix A we give a summary of the formulation, including all the equations needed to implement these algorithms. We have not implemented a minimum divergence step for this model. We plan to add this step in the future. Yet, given that, as we will see, the smart initialization procedure described below makes EM unnecessary in our experiments, we believe minimum divergence is unlikely to result in better models for this approach, at least for the current experiments.

Note that the matrix D in the JPLDA model can be full or diagonal. If we want D to be diagonal, we simply set D to be the diagonal part of the estimated value for D in each maximization step of the EM algorithm, as done for the standard PLDA EM algorithm (Brümmer, 2010a).

4.1. EM Initialization Procedure

The JPLDA model can be randomly initialized using the same procedure as for standard PLDA described in Section 2.1. We propose an alternative procedure to get the initial values for the PLDA model, U_0, V_0 and D_0. The procedure first estimates the matrix U_0 by training an SPLDA model with condition labels as targets, implicitly absorbing the effect of the speaker term into the noise term. This PLDA model is used to estimate the effect

of the condition in the samples, which is then subtracted to obtain condition-compensated samples. The new training data is then used to estimate another SPLDA model which should now model the effect of the speaker. This second SPLDA model is used obtain V_0 and D_0. Algorithm 1 gives the pseudocode for the proposed initialization algorithm. Note that, as usually done when training PLDA models, we assume an initial step where μ is set to the global mean of the data and subtracted from all training samples. The initialization step, as well as the EM iterations, use the resulting centered samples as input. As we will see this "smart" initialization leads to such a good starting point that EM iterations are unnecessary in our experiments.

Algorithm 1 Smart EM initialization approach for JPLDA. The matrix M, of size $N \times R_m$ contains the centered training vectors. L_c and L_s are vectors of size N containing the condition and speaker labels for the training data, respectively. Function SPLDAtrain returns the parameters of an SPLDA model (the subspace matrix and the noise precision matrix) after running 20 EM iterations, while function SPLDAlatent returns the estimated latent variables for each of the training samples. Note that all samples with the same label have the same latent variable.

procedure JPLDAInitialization(M, L_c, L_s, R_x, R_y)
  $U_0, D = \text{SPLDAtrain}(M, L_c, R_x)$
  $X = \text{SPLDAlatent}(M, L_c, U_0, D)$ ▷ X is a matrix of size $N \times R_x$
  $M_c = M - XU_0^T$
  $V_0, D_0 = \text{SPLDAtrain}(M_c, L_s, R_y)$
  return V_0, U_0, D_0

4.2. Scoring

As for standard PLDA, we define the score as the likelihood ratio between the two hypotheses: that the speakers are the same and that the speakers are different. Nevertheless, in this case we need to marginalize both likelihoods over two new hypotheses: that the nuisance conditions are the same and that they are different. This is because, in general, we cannot assume that the nuisance condition is known during testing. Hence, the LR is computed as follows:

$$LR = \frac{p(E,T|H_{SS},H_{SC})P(H_{SC}|H_{SS}) + p(E,T|H_{SS},H_{DC})P(H_{DC}|H_{SS})}{p(E,T|H_{DS},H_{SC})P(H_{SC}|H_{DS}) + p(E,T|H_{DS},H_{DC})P(H_{DC}|H_{DS})} \quad (6)$$

where, as before, H_{SS} is the hypothesis that the speakers for both sets are the same, and H_{DS} is the hypothesis that they are different, while H_{SC} is the hypothesis that the nuisance condition for both sets is the same, and H_{DC} is the hypothesis that they are different. This LR value can be computed using a closed form derived in Appendix A and, in more detail, in (Ferrer, 2017).

Note that here we assume that all samples from the enrollment set come from the same condition and all samples from the test set come from the same condition, which could be the same or different from the enrollment condition. This is trivially true when the sets are composed of a single sample, which is the case we consider in the experiments in this paper. The formulation would become more complex without this assumption since we would need to consider the possibility that each sample in each set could come from different conditions. In

(Ferrer, 2017), we also derive the scoring formula for a multi-enrollment single-test case where the enrollment conditions are known and different from the test condition. This formulation is used when applying JPLDA to language identification (LID). Experiments on LID will be the subject of another paper. Further, the generalization of the scoring formula to multiple enrollment or test samples with unknown conditions will be considered in future work.

Another implicit assumption made in Equation (6) is that the conditions in the test trials have not been seen during training. This assumption is also made with respect to the speakers, both in JPLDA and standard PLDA scoring. While, in many applications, the assumption that test speakers are unseen during training is appropriate, this may not always be the case for nuisance conditions. In particular, in our experiments, where the nuisance condition is given by the language spoken in the signal, this assumption does not hold for some trials, since some of the test languages are seen during training. In the future, we plan to relax this assumption, which may allow for further performance improvements.

The scoring formula above depends on two prior probabilities, the probability that the enrollment and test conditions are the same given that the speakers are the same, $P(H_{SC}|H_{SS})$, and the probability that the conditions are the same given that the speakers are different, $P(H_{SC}|H_{DS})$. The other two prior probabilities are dependent on these two since $P(H_{SC}|H_{SS}) + P(H_{DC}|H_{SS}) = 1$ and $P(H_{SC}|H_{DS}) + P(H_{DC}|H_{DS}) = 1$. These two independent prior probabilities are parameters that could be computed from the training data, tuned using a development set, or set to arbitrary values based on what is known about the test data. In some applications, the nuisance condition might be known also in testing. In that case, the same-condition priors can be set to 1.0 for same-condition trials and to 0 for different-condition trials.

5. Experimental Setup

In this section we describe the task, the performance metrics, the data used for the experiments and the procedure used to convert each audio sample to a fixed-length vector to be modeled by the different PLDA methods.

5.1. Multilanguage Speaker Verification

The task considered for our experiments is speaker verification, which consists of determining whether two sets of samples, an enrollment and a test set, belong to the same speaker or not. Here we consider the simplest case, where both enrollment and test sets each contain a single speech sample. A pair of enrollment and test samples is called a *trial*. A trial is a target trial if the enrollment and test speakers are the same and an impostor trial if the two speakers are different. In this paper we explore the problem of multilanguage speaker verification where test trials can be composed of two samples in the same language (same-language trials) or two samples in different languages (cross-language trials).

Most state-of-the-art speaker verification systems are inherently language-independent in the sense that they do not use information about the language spoken in order to generate the output score. Yet, this does not mean that they are robust to language variation. In fact, speaker verification performance is known to degrade significantly in cross-language trials as well as in same-language trials from languages not found in the training set (Auckenthaler et al., 2001; Misra and Hansen, 2014; Rozi et al., 2016).

Rozi et al. (2016) discusses a problem that occurs when training PLDA models with multilingual data: the distribution of the speaker factors becomes broader to cover the different languages that a speaker might speak, which could result in suboptimal performance on same-language trials. They propose to mitigate this problem by training a standard PLDA model using both language and speaker as targets (i.e., samples from the same speaker but different language are considered as different speakers). This language-aware PLDA model performs significantly better on same-language trials than the model trained with speaker targets, but degrades on cross-language trials, since it cannot model cross-language variation. JPLDA, on the other hand, can simultaneously model language and speaker factors, allowing the speaker factors to keep a sharper distribution, while still modeling the effect of language, resulting in improved performance both in same-language and cross-language trials with respect to standard PLDA.

5.2. Performance Metrics

We compute performance using the equal-error rate (EER) and detection error (DET) curves. The DET curves and the EER measure the performance of a system that uses the scores (in our case, the LRs) to make final decisions on the label of each sample by comparing these scores with a decision threshold. Samples whose scores are above the threshold are labeled as targets and samples whose scores are below the threshold are labeled as impostors. Two types of error are then possible: (1) misses, the true target trials that are labeled as impostors by the system, and (2) false alarms, the impostor trials that are labeled as targets by the system. The EER is given by the miss rate when the decision threshold is set such that the miss rate is equal to the false alarm rate.

DET curves (Martin et al., 1997) are a variation over the traditional receiver operating characteristic (ROC) curves that have been widely used for speaker verification for two decades. A DET curve is a plot of the false alarm rate versus the miss rate obtained while sweeping a decision threshold over a certain range where the axes are transformed to a probit scale. The probit transformation, the inverse of the cumulative distribution function of the standard normal distribution, converts the miss versus false alarm rate curve into a straight line if the score distribution for the two classes is Gaussian with the same standard deviation (Martin et al., 1997), which is a reasonable approximation for many speaker verification systems.

5.3. Training Data

We consider two training conditions, one that includes all our available training data (FULL) and a subset that keeps only one language for each speaker (MONOLING). The second condition is designed to help us analyze performance of the PLDA methods under this extremely challenging scenario where no explicit information is available in the training data of the effect that language has on the vectors representing the samples.

The FULL training set is composed of:

- Switchboard Cellular Part 1 (Graff et al., 2001) and Cellular Part 2 (Graff et al., 2004), consisting of English cellphone conversations
- Switchboard 2 Phase 2 (Graff et al., 1999) and Phase 3 (Graff et al., 2002) samples, consisting of English telephone conversations

- Mixer data (Cieri et al., 2007) from the 2004 to 2008 speaker recognition evaluations organized by the National Institute of Standards and Technology (NIST). This data contains English samples recorded both on telephone and microphone channels and non-English samples recorded on telephone channels. With very few exceptions, speakers that recorded non-English samples also recorded English samples. Only one speaker has data in two non-English languages and no data in English. Only a subset of this data is used for training, leaving some speakers out for testing (Section 5.4). We also discard data from languages for which only one or two speakers are available and samples where the language was unavailable or ambiguous (e.g., more than one language listed in the language key) in NIST's keys.

The MONOLING training set is created by randomly keeping the samples from only one of the languages spoken by each speaker from the FULL training set.

Finally, in Section 6.4, we analyze results when subsetting these two training sets to contain a more balanced representation of channels while keeping all data from bilingual speakers for the FULL training set. Specifically, we subset the FULL training set to discard all the telephone data from speakers that do not have data in both English and another language. This is data that is not adding much new information, since all non-English data is recorded over telephone line, and all speakers with non-English data also have telephone recordings in English. By subsetting the data this way, we achieve a more balanced representation of the telephone data with respect to the microphone data, while emphasizing the data from bilingual speakers, which is a very small minority on the original set including all the data. To create the subset for the MONOLING training set, we simply use the samples that appear in the subset from the FULL training set and also appear in the MONOLING set.

Table 1 shows statistics on the two training sets and their corresponding subsets. The languages included under "Other" are: Arabic (with 440 samples); Bengali (88); French (25); Chinese (868); Farsi (25); Hindi (143); Italian (11); Japanese (124); Korean (78); Russian (478); Spanish (170); Tagalog (26); Thai (185); Vietnamese (169); Chinese Wu (63); and Cantonese (216).

5.4. Test Data

We consider two testing conditions, one composed of Mixer data and used for development and one composed of LASRS data and used as held-out set for final evaluation of the selected methods.

The Mixer test data is composed of telephone samples from Mixer collections (Cieri et al., 2007) from the 2005 to 2010 NIST speaker recognition evaluations, from speakers not used for training. We include 119 samples in Arabic from 21 speakers; 200 samples in Russian from 47 speakers; 309 samples in Thai from 38 speakers; 827 samples in Chinese from 163 speakers; and 5755 samples in English from 701 speakers (including those that also speak one of the other languages). The trials are created by selecting the same number of target and impostor same-language and cross-language trials such that the final set of trials is a balanced union of both types of trials. Further, the same-language trials are created as a balanced union of English versus non-English trials. The final set of trials contains 11,522 target trials and 858,119 impostor trials.

The LASRS test data is composed of samples from a bilingual, multi-modal voice corpus (Beck et al., 2004). The corpus is composed of 100 bilingual speakers from each of

Name	Sel	Sample count				Speaker count			
		Eng Mic	Eng Phn	Other Phn	Total	MonoLing		BiLing	Total
						Eng	Other		
FULL	all	11017	38382	3109	52508	2764	34	495	3293
	subset	11017	3711	2733	17461	207	0	494	701
MONOLING	all	10797	36619	1688	49104	3026	267	0	3293
	subset	10797	1948	1332	14077	468	233	0	701

Table 1: Statistics for the four training sets considered in our experiments. "Eng" refers to English data while "Other" refers to any language other than English. "Phn" refers to telephone or cellphone data, and "Mic" refers to all other microphones in the training set. "Monoling" refers to speakers for which we only have samples in a single language, and "Biling" refers to speakers for which we have samples in two languages (English plus one other language in most cases).

three languages: Arabic, Korean and Spanish. In all cases, the other language of the speakers is English. Each speaker is asked to perform a series of tasks, including talking with a partner on the phone and reading various texts, in English and also in their native language. Each task is recorded using several recording devices and repeated in two separate sessions recorded on different days. The LASRS trials for this work are created by enrolling with data from the first recorded session and testing on the second recorded session in each of the two spoken languages. We use only the conversational data from each session. This results in the same number of same-language and cross-language trials for a total of 848 target trials and 100336 impostor trials for each of seven different microphones: a camcorder microphone (Cm); a Desktop microphone (Dm); a studio microphone (Sm); an omnidirectional microphone (Om); a local telephone microphone (Tm); a remote telephone microphone (Tk); and a telephone earpiece (Ts). For this study, we only consider same-microphone trials for simplicity of analysis. For more details on the collection protocol, see (Beck et al., 2004).

5.5. I-vector Extraction

For validation of the proposed approach, we use a traditional i-vector framework for speaker recognition (Dehak et al., 2011). I-vectors are fixed-dimensional vectors that attempt to represent, as fully as the assumptions allow, the characteristics of the speech in an audio recording. In this framework, each recording is first represented by a sequence of short-term feature vectors $x = \{x_1, \ldots, x_L\}$. The length L of this sequence is variable and depends on the duration of the recording. The i-vector approach then assumes each of these feature vectors x_j is independently drawn from an N-component Gaussian mixture model (GMM) with weights w_i, covariance matrices C_i and means $\mu_i + T_i\omega$, for $i \in \{1, \ldots, N\}$. The parameters of the i-vector model are the set of w_i, C_i, μ_i, and T_i. While weights and covariances are fixed for all recordings, the means vary in a subspace determined by the matrices T_i. Each recording is then modeled by a different GMM, determined by the vector ω. These vectors are assumed

to have a prior normal standard distribution. The mean of the posterior distribution of ω given the sequence of features x is the i-vector, used to represent the recording.

The posterior distribution of the i-vector model is intractable. It cannot be used in an EM algorithm to obtain the model's parameters or even to obtain the i-vectors for a new sample, given the model. A mean-field Variational Bayes (VB) EM approach can be used to estimate the model's parameters and the i-vectors, as described by Brümmer (2015). In this approach, the parameters are iteratively reestimated, along with the posterior distribution for ω and the responsibilities for each recording. The responsibilities are variational parameters that can be interpreted as soft assignments for the frames to each of the N components of the model.

In the classical i-vector approach, though, the parameters w_i, C_i and μ_i are fixed in an initial step. They are given by the weights, covariances and means of a GMM, called the universal background model (UBM), trained using recordings from many different speakers. An approximate posterior distribution of ω is then obtained using a simplifying assumption: the responsibilities are set to be the UBM state posteriors. With the responsibilities and the w_i, C_i and μ_i parameters fixed, the T_i matrices are estimated by maximizing the VB lower bound. Finally, once the parameters of the model have been estimated, i-vectors are extracted as the mean of the approximate posterior distribution of ω given the features for the recording, again fixing the responsibilities to be the UBM state posteriors.

In our experiments, the process for extracting an i-vector to represent a variable-length speech recording is as follows. The first 20 mel-frequency cepstral coefficients (MFCCs) are extracted from the audio signal using a 25ms window every 10ms. MFCCs are an acoustic feature vector that captures information regarding the amplitude of different frequencies in a similar manner to how sounds are perceived by the human ear (Davis and Mermelstein, 1980). The MFCCs are appended with deltas and double deltas to help capture the dynamics of speech over time (e.g., Gales and Young, 2008). This results in a feature vector of 60 dimensions, with 100 frames (of feature vector) per second.

Speech activity detection (SAD) is then applied to remove any frames that do not contain speech. For this purpose we use a deep neural network (DNN)-based model trained on telephone and microphone data from a combination of Fisher (Cieri et al., a,b), Switchboard (Graff et al., 2001, 2004, 1999, 2002) and Mixer data (Cieri et al., 2007), as well as a 30-minute long dual-tone multi-frequency (DTMF) signal without speech, and a set of 3740 signals where speech from the Fisher corpora was corrupted with non-vocal music at different SNR levels. The ground truth labels used to train the DNN were obtained using our previous SAD system which consisted of a speech/non-speech hidden Markov model (HMM) decoder and various duration constraints. This system performed very well but was slow, complex and hard to retrain given new data. The labels for the corrupted data were obtained from the clean signals. As input to the SAD system we use MFCC features, mean and variance normalized over each waveform, except for the C_0 coefficient, for which the maximum rather than the mean is subtracted before dividing by the standard deviation. The normalized features are concatenated over a window of 31 frames. The resulting 620-dimensional feature vector forms the input to a DNN that consists of two hidden layers of sizes 500 and 100. The output layer of the DNN consists of two nodes trained to predict the posteriors for the speech and non-speech classes. These posteriors are converted into likelihood ratios using Bayes rule (assuming a prior of 0.5), and a threshold of 0.5 is applied to obtain the final

speech regions. For a more detailed description of the DNN-based SAD approach, please see (Graciarena et al., 2016).

The UBM is then estimated with an EM algorithm using the speech frames from a random subset of 10,000 samples (full recordings) of the data used to estimate the T_i matrices described next. The subspaces T_i are then estimated using the FULL training data described in Section 5.3, except that samples from languages for which only one or two speakers are available or where the language was unavailable or ambiguous are not discarded for this purpose. Finally, once the model is trained, the i-vectors for any audio sample can be extracted using only the speech frames, as in training.

The i-vectors can then be used for determining speaker similarity between two utterances using PLDA. For the experiments, we process the i-vectors with multiclass linear discriminant analysis (LDA) trained on the same training data used for PLDA, after which we subtract the mean over the training data and perform length normalization (Garcia-Romero and Espy-Wilson, 2011). The length-normalization step serves to better satisfy the Gaussianity assumption behind PLDA.

6. Results

In this section we compare results for different EM initialization techniques, parameter settings and training data for the proposed and the baseline PLDA techniques described in previous sections.

The nuisance condition for JPLDA in these experiments is the language spoken in the sample. During training, this label is known; during scoring, the label is marginalized to compute the LR, unless otherwise indicated. For TPLDA, on the other hand, we cannot take the mixture component to be the language spoken in the sample. This is because we do not have enough training speakers for each language to train a good PLDA model for each component. Hence, we consider a two-component model with a component modeling all English data and another component modeling the non-English data. This, as we will see, turns out to be a good model when matched data is available for training both components. Note that our implementation of TPLDA assumes that the mixture component is given both in training and in scoring. This is possible in our experiments because we have the language spoken during testing.

For SPLDA, FPLDA and JPLDA, the LDA dimension is set to 400; no dimensionality reduction is done in these cases but the data is still transformed by the LDA matrix, centered and length normalized. For TPLDA, on the other hand, we use an LDA dimension of 200, because we found that this value gives significantly better performance than keeping the original dimension of 400.

The speaker and language ranks for all experiments in this section are fixed to 200 and 16, respectively. These values were chosen for being optimal or approximately optimal for all methods under study (FPLDA, SPLDA and JPLDA) when using all available training data. The language rank of 16 is the largest rank that can be used for JPLDA. This value turned out to be optimal for JPLDA. FPLDA is largely insensitive to this parameter, giving very similar performance for language ranks between 5 and 16. Unless otherwise stated, all JPLDA results are obtained using $P(H_{SC}|H_{SS}) = P(H_{SC}|H_{DS}) = 0.5$. For TPLDA we use a diagonal matrix for the covariance of the noise model which proved to be slightly better than a full covariance.

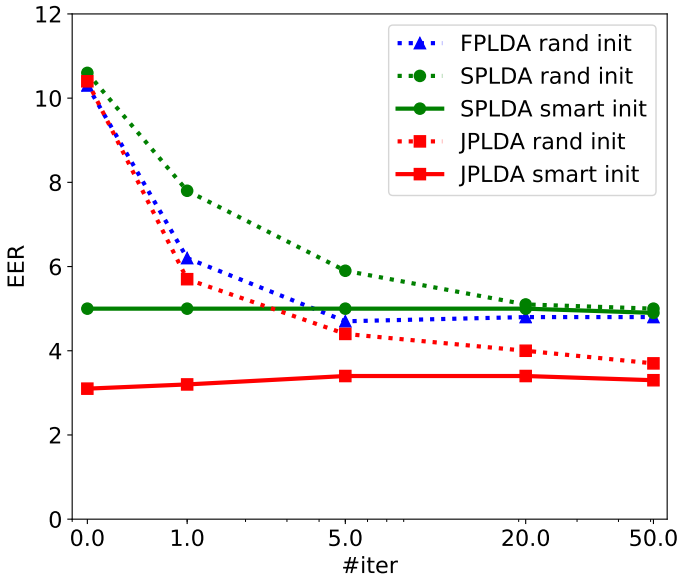


Figure 1: Comparison of performance as a function of the number of EM iterations on the Mixer test data using all available training data, for random and smart initialization for three different PLDA models. Note that a log scale is used for the x-axis.

All tuning decisions above were made based solely on the results on Mixer test data.

6.1. Initialization and Convergence of Training Procedure

We first show results for SPLDA, FPLDA and JPLDA as a function of the number of EM iterations run for the two initialization procedures, random and smart, explained in Sections 2.1 and 4.1. For FPLDA, no standard way exists of which we are aware to smartly initialize all parameters of the model. In this case, we only show results for random initialization. For this section, we use the FULL training data without subsetting and test on the Mixer data.

Results in Figure 1 show that EM iterations are essential when random initialization is used, leading to large gains over the initial random model as the iterations progress and converging to an approximately stable value when reaching 50 iterations. On the other hand, when smart initialization of SPLDA or JPLDA is used, EM iterations are not necessary on this data set. In fact, JPLDA performance with smart initialization slightly degrades for larger number of iterations, probably due to overfitting of the training data. For this reason, for the rest of the experiments we use only one iteration of EM for JPLDA, though zero iterations could also be safely used.

6.2. Prior Probability of Same Language in JPLDA

In this section we show JPLDA results on the Mixer development set when using all available training data as a function of the prior probabilities of same language, $P(H_{SC}|H_{SS})$ and $P(H_{SC}|H_{DS})$ (see Section 4.2). We fix these two parameters to the same value and sweep

this value between 0 and 1 at 0.1 steps. We show results on the full test data but also split the data into same-language and cross-language trials. We compare these results with those we would obtain by knowing the language of each sample a priori and using this knowledge during scoring to set the priors appropriately as explained in Section 4.2.

Figure 2 shows performance as a function of the probability of same language parameter. Values below 0.1 are optimal for the cross-language trials, while values above 0.1 and below 1.0 are optimal for same-language trials. The fact that a probability of same language of 1.0 is not optimal for same-language trials might be due to some samples including code-switching, making trials involving these samples not strictly same-language trials. Further, a value lower than 1.0 for same-language trials may better accommodate the variation in accent that takes place when people speak to different interlocutors. Once all trials are pooled together, values between 0.3 and 0.8 give almost identical performance. For this range of values, we can also see that performance is very close to what we would obtain if the language of the test files was known during scoring (the red dashed line in the plot). This performance is obtained by setting the probability of same language to 1.0 for same-language trials and to 0.0 for cross-language trials. For the remaining experiments, we use a probability of same language of 0.5.

Figure 2 shows that the same-language and different-language subsets have a significantly lower EER than the pooled set of trials. This is due to the fact that the scores for both sets of trials are misaligned with each other. That is, the EER threshold is different for both sets, leading to a larger EER than that for either set once the trials are pooled together. As we will see in Section 6.4, Figure 6, this effect is actually more salient in standard PLDA approaches, with JPLDA mitigating the problem, though not fully solving it.

6.3. Method Comparison

We now compare the performance of the four methods on all test sets from Mixer and LASRS divided by microphone type using the two training sets: FULL and MONOLING.

The top plot in Figure 3 shows that FPLDA gives slightly better performance than SPLDA for some channels (mostly the telephone ones) when the FULL training data is used. For this reason, for the remaining experiments in this paper, we use FPLDA as the baseline.

Comparing the two methods that consider language labels during training, TPLDA and JPLDA, on the top plot in Figure 3, we see that they both give significant gains over the baselines on Mixer data, where the channel is matched to the majority of the training data's channel. In this case, both approaches succeed in mitigating the effect of language variability. On the other hand, when the channel is not exactly the same as the one observed most in training, TPLDA fails to generalize, leading to consistently worse performance than JPLDA. This is reasonable: while alternative microphone data is observed for the English training data, only telephone data is observed for the non-English data. This implies that the PLDA mixture corresponding to non-English data in TPLDA was only learned with telephone data, resulting in the poorer performance observed on some of the LASRS channels. On the other hand, JPLDA can leverage the information about alternative microphones learned from English data for all languages, since the matrix that models this variability is shared across languages.

In the bottom plot in Figure 3, we see that when only a single language from each speaker is available for training (that is, the within speaker variation due to language is not

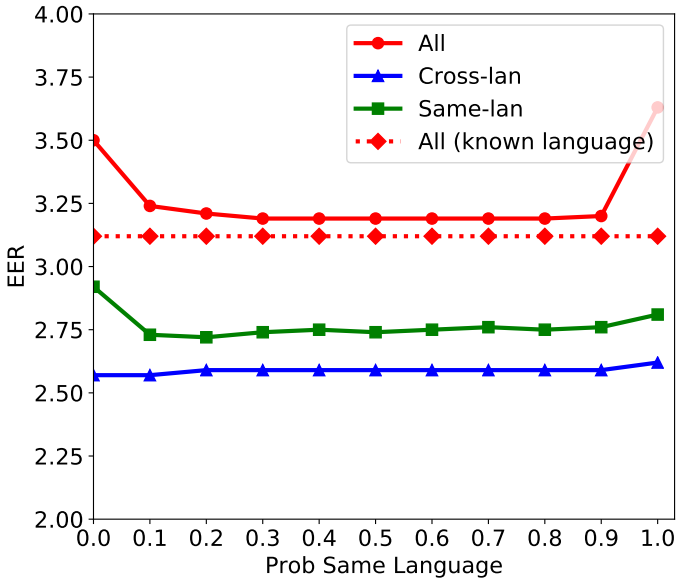


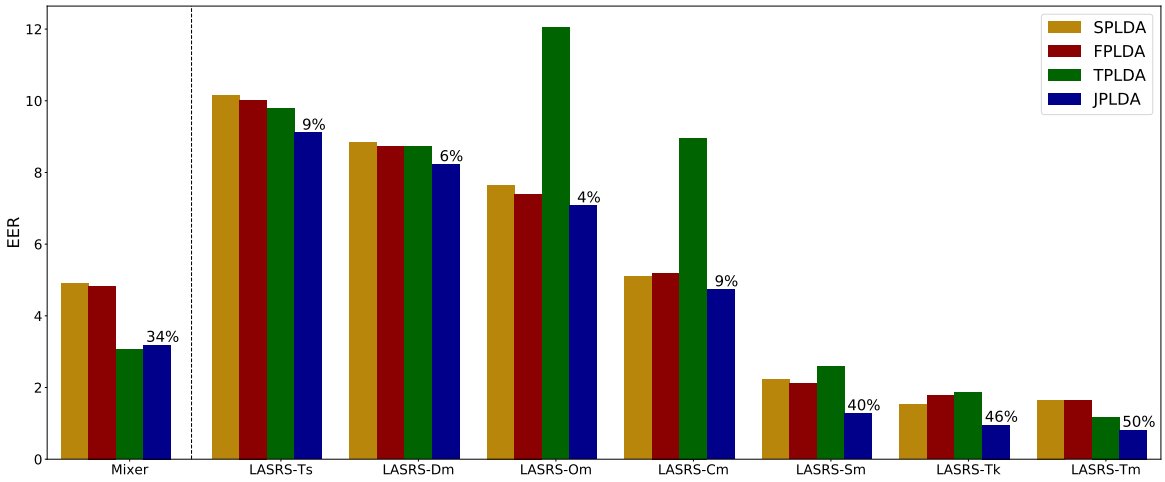
Figure 2: Comparison of JPLDA performance as a function of the prior probability of same language on the Mixer test data using the FULL training data. The known-language line corresponds to the performance on all trials when using the information about the test language during scoring.

observed in training), TPLDA leads to a large degradation over both baselines. Note that, as far as we know, TPLDA had not been tested under this challenging scenario. Rather, it was tested using training data where each class of interest (e.g., a face) was seen under all possible conditions (front and profile) (Li et al., 2012). When each class is seen under a single condition, the TPLDA model basically degenerates to separate (untied) PLDA models, each learned on the data from its own condition. This implies that the resulting mixture will be unable to model the cross-language variability, which results in extremely degraded performance on the cross-language trials. Indeed, our results indicate that the same-language trials get reasonable TPLDA performance (results not shown here), it is the degradation on the cross-language trials that affects the overall performance as observed in the plot.

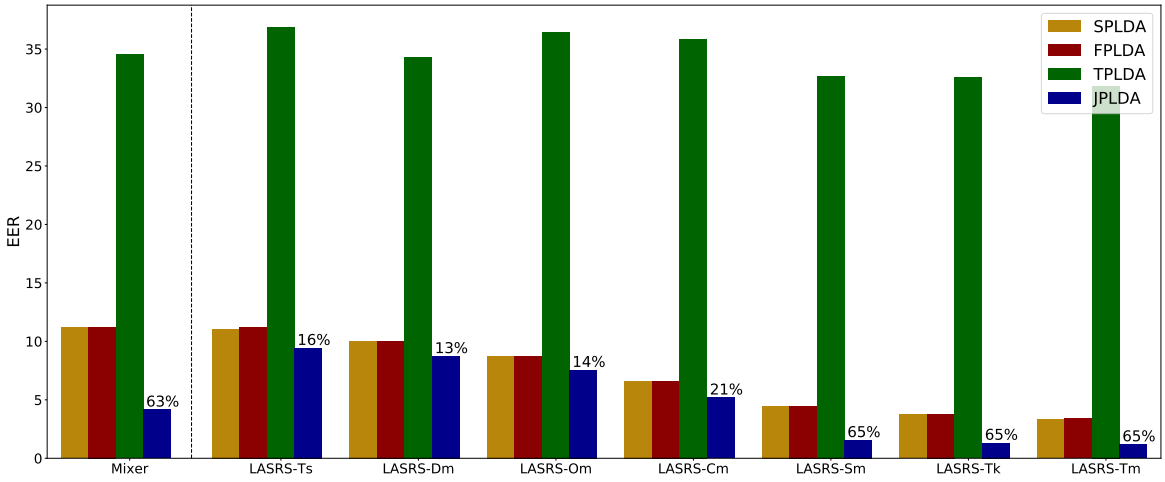
Finally, focusing on JPLDA, we see that significant gains are observed compared to both baselines using both training sets, with larger relative gains when the training data contains only a single language per speaker, in which case we find gains from 13% of up to 65% relative to the FPLDA baseline.

6.4. Training Data Comparison

Finally, in this section we compare the FPLDA baseline and JPLDA using the two training sets defined in Section 5.3 and their subsets, where we discard telephone samples from speakers that only have English samples in an attempt to achieve a better balance between English and non-English samples and telephone and microphone samples.



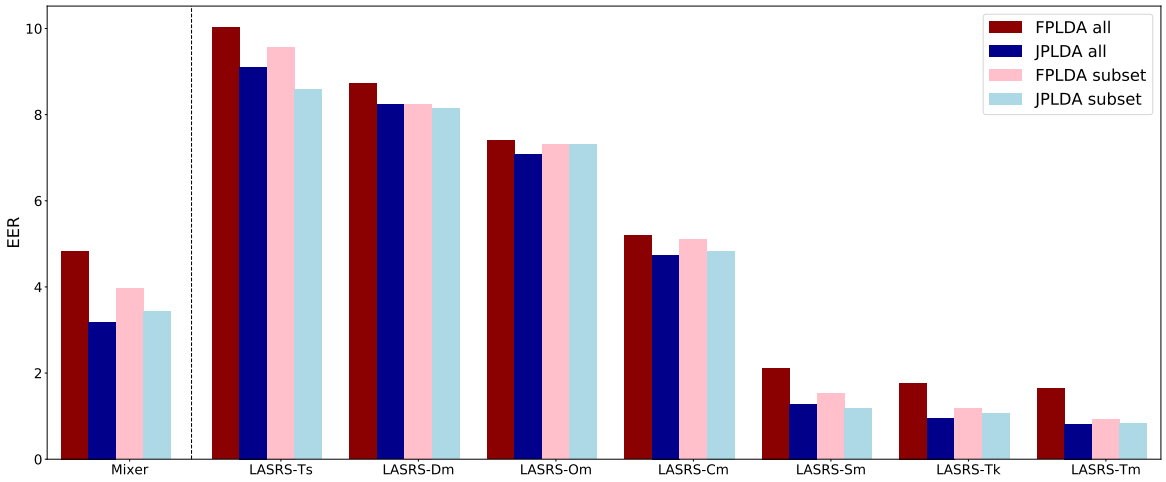
(a) Training data: FULL



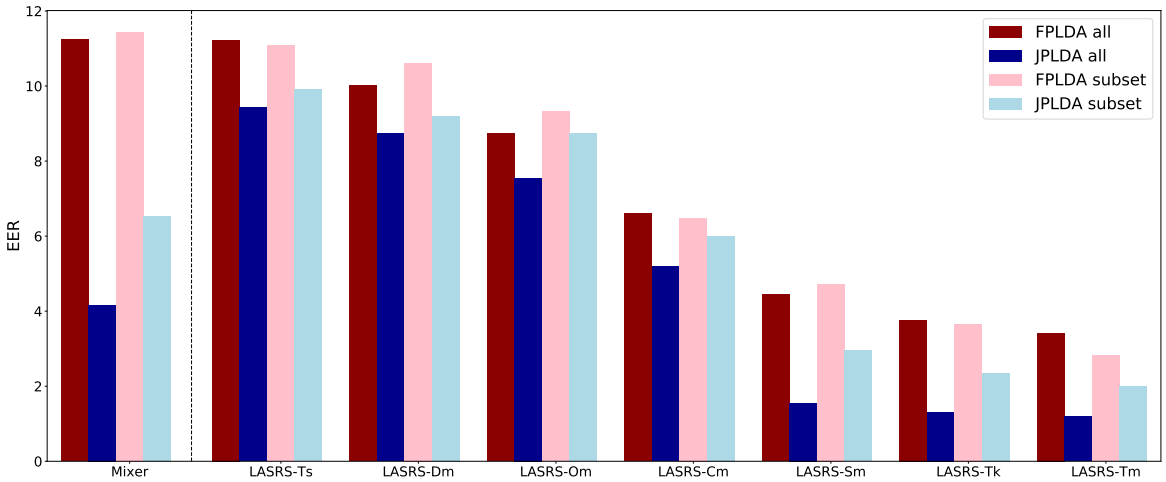
(b) Training data: MONOLING

Figure 3: Comparison of performance for four PLDA methods on all test sets using both training sets, FULL and MONOLING. The numbers on top of the JPLDA bars show the relative gain of JPLDA relative to FPLDA.

Figure 4 shows that, for FPLDA, using the subset is significantly better than using the full training set for both training conditions, FULL and MONOLING, for most test conditions. That is, FPLDA benefits from having a more balanced distribution of conditions within the training data. This is because, in standard PLDA, the samples from all speakers are assumed to follow the same distribution, regardless of whether these samples are all in English, or both in English and some other language. Hence, if a large proportion of speakers only have English samples, the parameters in the PLDA model will be mostly determined by what is optimal for these speakers, degrading the performance on non-English and cross-language trials.



(a) Training data: FULL



(b) Training data: MONOLING

Figure 4: Comparison of performance for FPLDA and JPLDA on all test sets using the two training sets, FULL and MONOLING. For each case, we compare using the full training set and a subset where we discard telephone samples from speakers that only have English samples in the FULL training set.

On the other hand, JPLDA does not seem to require subsetting the data[2]. In fact, for the FULL training condition, JPLDA leads to similar or better performance (using either the full training set or the subset) than FPLDA using the subset. For the MONOLING condition, the advantage of JPLDA over FPLDA is much larger than for the FULL training set, consistently showing significant gains over the best FPLDA result. Further, for this training condition we

2. Note that the EER on the better performing test sets (LASRS-Sm, LASRS-Tk and LASRS-Tm) corresponds to very few misses, making that metric somewhat unreliable on those sets. However, DET curves shown later in the section complement the EER results, supporting the overall conclusions made based on EERs.

see a consistent trend showing that JPLDA benefits from using the full training set, which indicates that, contrary to PLDA, JPLDA can handle the imbalance in the full set of data, successfully leveraging the additional samples missing from the subset.

To complement the EER results in the bar plots, Figure 5 shows the DET curves for all test sets. We show these curves for the more challenging training condition, MONOLING, where JPLDA gives the biggest advantage over FPLDA. The plots show that the gains are not specific to the EER operating point. Rather, JPLDA gives a significant gain over FPLDA over a very wide range of operating points corresponding to miss and false alarms rates between 0.01% to 40%. Further, we also see the advantage of using all the available training data rather than just the subset when using JPLDA, while the opposite is true for FPLDA, as already observed in the EER bar plots.

Finally, Figure 6 shows EER results on Mixer test data using the two training sets and their subsets for all trials (as in previous bar plots) as well as for same-language and cross-language trials. The performance on all trials is the same as in Figure 4. These plots show that: (1) Both same-language and cross-language trials benefit from using JPLDA, particularly for the MONOLING training conditions. (2) The JPLDA benefit from using the complete training sets holds for both same-language and cross-language subsets of trials. (3) The FPLDA benefit from using the subset only holds on the same-language trials; cross-language trial performance is degraded or unchanged by subsetting the training data. And (4) the relative gain from JPLDA is larger once same-language and cross-language trials are pooled together. This last observation indicates that JPLDA is not only improving discrimination for each type of trial (same-language and cross-language), but it is also aligning the distributions of these two types of trials such that when they are pooled together, the relative gain from using JPLDA is emphasized. Yet, as we can see, JPLDA does not appear to fully solve the problem, since the pooled performance is still somewhat worse than that of the subsets. We plan to study the source of the remaining misalignment in the near future.

7. Conclusions

We have proposed a generalization of PLDA where within-class variability factors are no longer considered independent across samples. The method assumes that the identity of a nuisance condition is known during training and ties the latent variable corresponding to the within-class variability across all samples with the same nuisance condition label. During scoring, a likelihood ratio is computed as for standard PLDA by marginalizing over the nuisance condition. Hence, the identity of the nuisance condition can be unknown during testing.

We show results on a multilingual speaker recognition task comparing the proposed method with two types of standard PLDA models as well as to a tied PLDA model where the nuisance condition is used to determine the component in a mixture of PLDA models. Our results show that large relative gains are obtained from using JPLDA when the training data contains few or no speakers with data in more than one language. That is, the JPLDA model is able to extrapolate the effect of language from a small proportion or even zero training speakers with data from more than one language. Standard PLDA models are only able to mitigate the effect of language when exposed to a significant proportion of training speakers with data in more than one language.

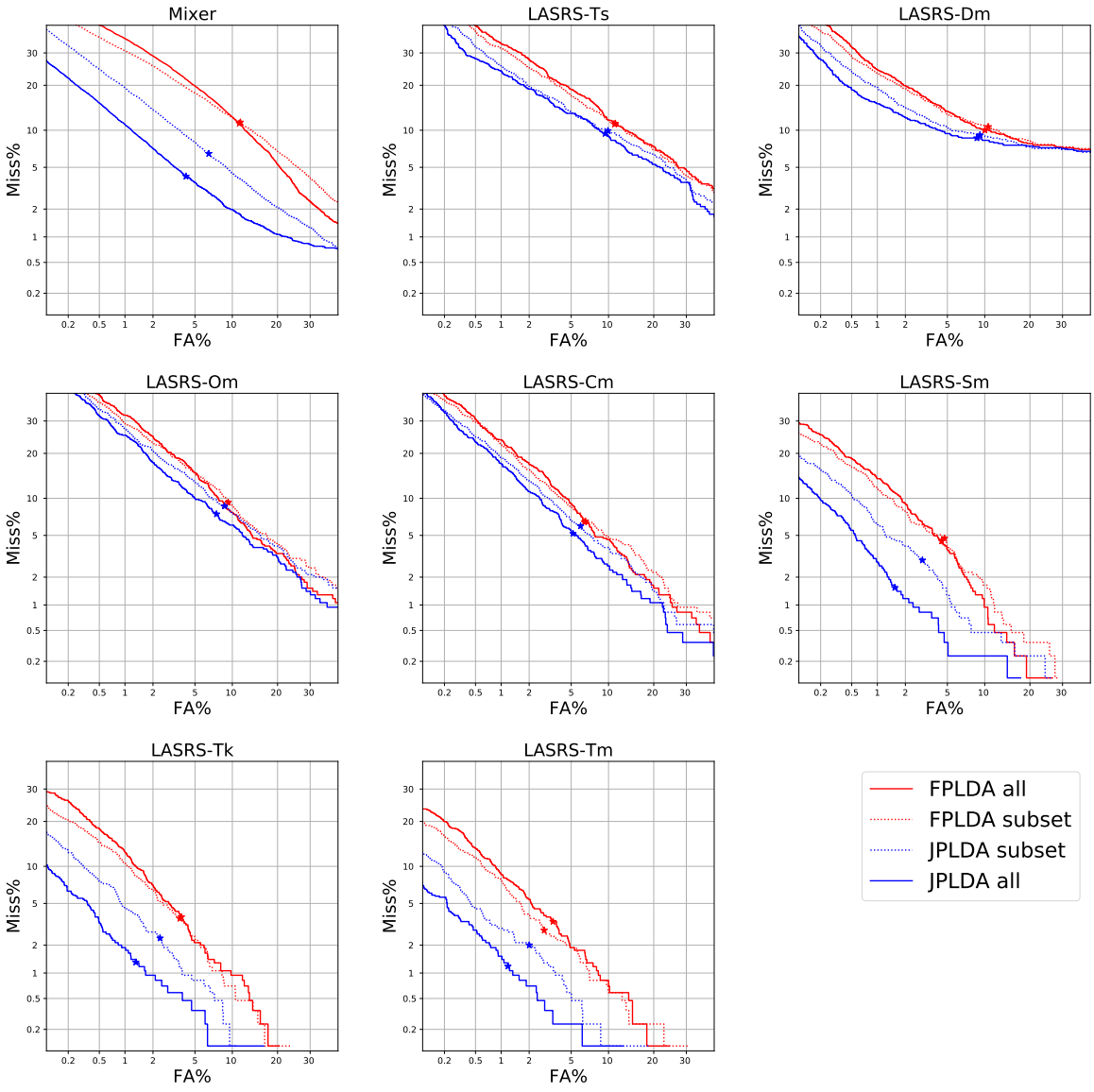
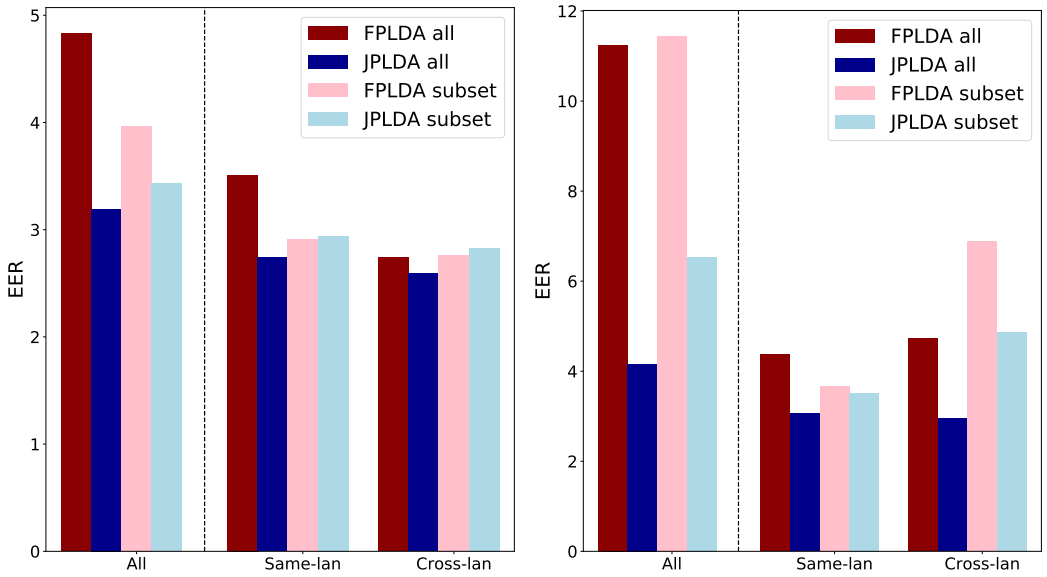


Figure 5: DET curves for FPLDA and JPLDA on all test sets using the MONOLING training set and its subset. The marker over each curve corresponds to the EER point for that system.

The proposed JPLDA method can be used for any task for which standard PLDA is used whenever a discrete nuisance condition is known during training. Examples include speaker recognition using channel, speaking style or language labels, among others, as the sample-dependent nuisance condition, and face recognition using pose as sample-dependent nuisance condition. The strength of JPLDA lies in its ability to extrapolate the effect that



(a) Training data: FULL (b) Training data: MONOLING

Figure 6: Comparison of performance for FPLDA and JPLDA on the Mixer test set on all trials as well as on same-language and cross-language subsets, using the two training sets, FULL and MONOLING, and their subsets.

the nuisance condition has on the samples based on few or even no classes (speakers or faces) seen under several nuisance conditions.

The proposed approach introduces the additional requirement with respect to the original PLDA approach that the identity of the nuisance condition be known during training. In future work, we will explore the possibility of automatically detecting the nuisance conditions, using classifiers trained on data for which the factors are known or using clustering with distance metrics designed to reflect the nuisance of interest. Finally, an interesting generalization of the proposed approach would be to allow for more than one sample-dependent nuisance condition. These are directions we plan to explore in the near future.

Acknowledgments

This material is based upon work supported partly by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-15-C-0037. The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. Distribution Statement A: Approved for Public Release, Distribution Unlimited.

Appendix A. JPLDA Formulas

In this appendix we derive the probabilities that are needed during training with the EM algorithm and during scoring with the likelihood ratio for the proposed JPLDA method. The derivations closely follow the ones for standard PLDA in (Brümmer, 2010a) with one main difference: in the new model, most probabilities cannot be formulated by speaker and then multiplied to get the total probabilities, as is usually done for standard PLDA, since the condition introduces dependencies across samples from different speakers. Instead, we formulate all probabilities over all samples.

In the following, we take

$$\begin{aligned} Y &= \{y_1, \ldots, y_S\} \\ X &= \{x_1, \ldots, x_C\} \\ M &= \{m_1, \ldots, m_N\} \end{aligned}$$

where S, C and N are the total number of speakers, conditions, and samples, respectively. Further, we assume that $\mu = 0$. In the general case, as done for standard PLDA, this parameter is set to the global mean of the training data and subtracted from all samples before running EM or scoring.

A.1. Probability Distributions

The joint prior for the hidden variables for all the data is given by

$$p(Y, X) = p(X)p(Y) \propto \exp(-\frac{1}{2}\sum_s y_s^T y_s - \frac{1}{2}\sum_c x_c^T x_c), \tag{7}$$

and the full data likelihood is given by

$$\begin{aligned} p(M|Y, X, \lambda) &= \prod_i N(m_i|Vy_{s_i} + Ux_{c_i}, D^{-1}) \\ &\propto \exp\sum_i \left(-\frac{1}{2}(m_i - Vy_{s_i} - Ux_{c_i})^T D(m_i - Vy_{s_i} - Ux_{c_i}) + \frac{1}{2}\log|D|\right). \end{aligned} \tag{8}$$

The joint probability is proportional, as a function of M, X and Y, to the product of the likelihood and the prior,

$$\begin{aligned} p(M, Y, X|\lambda) \propto \exp\Bigg[&\sum_i \left(-\frac{1}{2}m_i^T D m_i + m_i^T DV y_{s_i} + m_i^T DU x_{c_i} - x_{c_i}^T J y_{s_i}\right) \\ &-\frac{1}{2}\sum_s y_s^T L_s y_s - \frac{1}{2}\sum_c x_c^T K_c x_c\Bigg] \end{aligned}$$

where

$$\begin{aligned} J &= U^T DV \\ K_c &= n_c U^T DU + I \\ L_s &= n_s V^T DV + I \end{aligned}$$

where n_c and n_s are the number of samples for condition c and speaker s, respectively.

We can now compute the posterior from two factors:

$$p(Y, X|M, \lambda) = p(Y|X, M, \lambda)p(X|M, \lambda)$$

The outer posterior is proportional, as a function of Y, to the joint probability. Keeping only the terms in the joint probability that depend on Y we get

$$p(Y|X, M, \lambda) \propto \exp\left[\sum_i \left(m_i^T DV y_{s_i} - x_{c_i}^T J y_{s_i}\right) - \frac{1}{2}\sum_s y_s^T L_s y_s\right] \propto \prod_s N(y_s|\hat{y}_s, L_s^{-1}) \quad (9)$$

where

$$\begin{aligned}
\hat{y}_s &= \tilde{y}_s - L_s^{-1} J^T \bar{x}_s \\
\tilde{y}_s &= L_s^{-1} V^T D f_s \\
\bar{x}_s &= \sum_{i|s_i=s} x_{c_i} \\
f_s &= \sum_{i|s_i=s} m_i
\end{aligned}$$

Marginalizing that distribution we can extract the posterior for a single latent variable y_s:

$$p(y_s|X, M, \lambda) = N(y_s|\hat{y}_s, L_s^{-1}) \quad (10)$$

The inner posterior is proportional to the joint probability of X and M:

$$\begin{aligned}
p(X|M, \lambda) &\propto p(M, X|\lambda) = \left.\frac{p(Y, X, M|\lambda)}{p(Y|X, M, \lambda)}\right|_{Y=0} \\
&\propto \exp\left(\sum_c g_c^T DU x_c - \frac{1}{2}\sum_c x_c^T K_c x_c - \sum_c x_c^T J\bar{\tilde{y}}_c + \frac{1}{2}\sum_s \bar{x}_s^T J L_s^{-1} J^T \bar{x}_s\right)
\end{aligned}$$

where we have used the *candidate's formula* (Besag, 1989) to obtain the joint probability and where

$$\begin{aligned}
g_c &= \sum_{i|c_i=c} m_i \\
\bar{\tilde{y}}_c &= \sum_{i|c_i=c} \tilde{y}_{s_i} = \sum_{i|c_i=c} L_{s_i}^{-1} V^T D f_{s_i}
\end{aligned}$$

We now define vectors which are the concatenation of all individual latent vectors:

$$\begin{aligned}
\boldsymbol{X} &= [x_1^T \dots x_C^T]^T \\
\boldsymbol{Y} &= [y_1^T \dots y_S^T]^T
\end{aligned}$$

and, similarly, for all other vectors. Converting the sums into matrix form, we get

$$p(\boldsymbol{X}|M, \lambda) \propto \exp\left(\boldsymbol{X}^T \Phi - \frac{1}{2}\boldsymbol{X}(T_2 - H^T T_4 H)\boldsymbol{X}\right) \propto N(\boldsymbol{X}|\hat{\boldsymbol{X}}, \Sigma) \quad (11)$$

where

$$\begin{aligned}
\Sigma &= (T_2 - H^T T_4 H)^{-1} \\
\hat{\boldsymbol{X}} &= \Sigma\Phi \\
\Phi &= T_1\boldsymbol{G} - T_3\bar{\bar{\boldsymbol{Y}}} \\
T_1 &= \text{diagn}(U^T D, C) \\
T_2 &= \text{diag}(K_1, \ldots, K_C) \\
T_3 &= \text{diagn}(J, C) \\
T_4 &= \text{diag}(JL_1^{-1}J^T, \ldots, JL_S^{-1}J^T)
\end{aligned} \tag{12}$$

where $\text{diagn}(M, N)$ is a block diagonal matrix with matrix M in each of N blocks and $\text{diag}(T_1, \ldots, T_N)$ is a block diagonal matrix with blocks given by matrices T_i. The matrix H is of size SR_xxCR_x, where block $H_{s,c}$ (R_x rows and columns starting at position (sR_x, cR_x) in H) is given by:

$$H_{s,c} = n_{s,c}I$$

where I is the identity matrix of size R_x and $n_{s,c}$ is the number of times that condition c occurs for speaker s, which could be zero.

As for the outer posterior, we can marginalize the distribution in Equation (11) to get the distribution for an individual x_c

$$p(x_c|M, \lambda) = N(x_c|\hat{x}_c, \Sigma_c) \tag{13}$$

where Σ_c and $\hat{x}_c$ are the blocks corresponding to latent variable c in Σ and $\hat{X}$.

A.2. EM Algorithm

The EM auxiliary function is given by the expected value of the log-likelihood with respect to the posterior probability of the hidden variables given the data and the previously estimated model parameters, λ_{k-1}.

$$\begin{aligned}
Q(\lambda_k|\lambda_{k-1}) &= E_{X,Y|M,\lambda_{k-1}}\left[\log p(M, Y, X|\lambda_k)\right] \\
&= \frac{N}{2}\log|D| - \frac{1}{2}\text{tr}(SD) - \frac{1}{2}\text{tr}(RW^T DW) + \text{tr}(TDW) + \text{const}
\end{aligned}$$

where

$$\begin{aligned}
W &= [UV] \\
S &= \sum_i m_i m_i^T \\
R &= \sum_i \langle z_i z_i^T \rangle \\
T &= \sum_i \langle z_i \rangle m_i^T \\
z_i &= [x_{c_i}^T y_{s_i}^T]^T
\end{aligned}$$

where the $\langle$ and $\rangle$ symbols stand for the expectation with respect to the distribution of z_i given the data M and the previous parameters λ_{k-1}.

A.2.1. M-Step

Now, differentiating Q with respect to D and W and setting to zero, we get that

$$\begin{aligned} D^{-1} &= \frac{1}{N}(S - WT) \\ W^T &= R^{-1}T \end{aligned}$$

So, the matrices are estimated exactly the same way as in the standard PLDA approach (Brümmer, 2010a). The additional complexity of JPLDA lies in the forms that R and T take.

A.2.2. E-Step

To find T and R we use the equations we have obtained for the posterior distributions of x_c and y_s (Equations 10 and 13). The two components of T are given by:

$$\begin{aligned} T_x &= \sum_i \langle x_{c_i} \rangle m_i^T = \sum_c \hat{x}_c g_c^T \\ T_y &= \sum_s \langle y_{s_i} \rangle m_i^T = \sum_s L_s^{-1}(V^T D f_s - J^T \bar{\hat{x}}_s) f_s^T \end{aligned}$$

where we use the law of total expectations to get the expectation of y_s from its conditional expectation and where

$$\bar{\hat{x}}_s = \sum_{i|s_i=s} \hat{x}_{c_i}$$

Finally, we can get the components of R as follows:

$$\begin{aligned} R_{xx} &= \sum_i \langle x_{c_i} x_{c_i}^T \rangle = \sum_c n_c(\Sigma_c + \hat{x}_c \hat{x}_c^T) \\ R_{yx} &= \sum_i \langle y_{s_i} x_{c_i}^T \rangle = \sum_s \left[\tilde{y}_s \bar{\hat{x}}_s^T - \tilde{J}^T \sum_{i|s_i=s} \sum_{j|s_j=s} \left[\hat{x}_{c_i} \hat{x}_{c_j}^T + \Sigma_{c_j,c_i} \right] \right] \\ R_{yy} &= \sum_i \langle y_{s_i} y_{s_i}^T \rangle = \sum_s n_s \left[L_s^{-1} + \tilde{y}_s \tilde{y}_s^T - \tilde{y}_s \bar{\hat{x}}_s^T \tilde{J} - \tilde{J}^T \bar{\hat{x}}_s \tilde{y}_s^T + \tilde{J}^T \langle \bar{x}_s \bar{x}_s^T \rangle \tilde{J} \right] \end{aligned}$$

where $\tilde{J} = JL_s^{-1}$ and Σ_{c_j,c_i} is the block in matrix Σ (Equation 12) corresponding to latent variables c_i and c_j.

A.3. Scoring

In this paper we assume all trials are composed of a single enrollment and a single test sample. That is, the sets E and T in Equation (6) are each composed of a single vector, m_E and m_T, respectively. M is then given by $\{m_E, m_T\}$. We can now use the formulas derived above to obtain the LR in Equation (6) where we need to compute four probabilities for the data given different hypotheses. The probabilities can be obtained using the candidate's formula (Besag, 1989):

$$p(M|h_s, h_c) = \left. \frac{p(M|X_{h_c}, Y_{h_s}) p(X_{h_c}) p(Y_{h_s})}{p(Y_{h_s}|X_{h_c}, M) p(X_{h_c}|M, h_s)} \right|_{X_{h_c}=0, Y_{h_s}=0} \tag{14}$$

where $h_s \in \{H_{SS}, H_{DS}\}$ and $h_c \in \{H_{SC}, H_{DC}\}$, and where

$$X_{h_c} = \begin{cases} \{x\}, & \text{if } h_c = H_{SC} \\ \{x_E, x_T\}, & \text{if } h_c = H_{DC} \end{cases}$$

$$Y_{h_s} = \begin{cases} \{y\}, & \text{if } h_s = H_{SS} \\ \{y_E, y_T\}, & \text{if } h_s = H_{DS} \end{cases}$$

That is, the latent variables are two independent vectors for the different-condition and different-speaker hypotheses and a single vector for the same-condition and same-speaker hypotheses.

The likelihood in the numerator of Equation (14) is the same for all four combination of hypotheses since, regardless of whether the latent variables are tied or not, Equation (8) has the same form. Hence, that term cancels out in the computation of the LR. The priors, on the other hand, will have one factor for the same-speaker or same-condition case and two identical factors, once evaluated at 0, for the different-speaker or different-condition case. All that is left to do is compute the inner and outer posteriors in the denominator of Equation (14) and evaluate them at 0.

The outer posterior $p(Y_{h_s}|X_{h_c}, M)$, given by Equation (9), takes the same value for both values of h_c when the latent variables are set to 0. Its logarithm is given by

$$\log p(Y_{h_s}|X_{h_c}, M)|_0 = \begin{cases} \frac{1}{2}k + \frac{1}{2}\log|L_1| - \frac{1}{2}(\tilde{m}_E + \tilde{m}_T)^T L_1^{-1}(\tilde{m}_E + \tilde{m}_T), & \text{if } h_s = H_{SS} \\ k + \log|L_2| - \frac{1}{2}\tilde{m}_E^T L_2^{-1}\tilde{m}_E - \frac{1}{2}\tilde{m}_T^T L_2^{-1}\tilde{m}_T, & \text{if } h_s = H_{DS} \end{cases}$$

where $L_2 = V^T D V + I$, $L_1 = 2V^T D V + I$, $\tilde{m}_E = V^T D m_E$, $\tilde{m}_T = V^T D m_T$ and $k = -R_y \log(2\pi)$.

The inner posterior is given in Equation (11). For the scoring scenario, its logarithm is given by

$$\log p(X_{h_c}|M, h_s) = \begin{cases} -\frac{1}{2}R_x k - Q_{h_c,h_s}, & \text{if } h_c = H_{SC} \\ -R_x k - Q_{h_c,h_s}, & \text{if } h_c = H_{DC} \end{cases}$$

where $Q_{h_c,h_s} = \frac{1}{2}\log|\Sigma_{h_c,h_s}| + \frac{1}{2}\Phi_{h_c,h_s}^T \Sigma_{h_c,h_s} \Phi_{h_c,h_s}$ with

$$\begin{aligned}
\Sigma_{H_{SC},H_{SS}} &= \left[2U^T D U + I - 4JL_S^{-1}J^T\right]^{-1} \\
\Sigma_{H_{SC},H_{DS}} &= \left[2U^T D U + I - 2JL_D^{-1}J^T\right]^{-1} \\
\Sigma_{H_{DC},H_{SS}} &= \left[\text{diagn}(K_D, 2) - [II]^T JL_S^{-1}J^T[II]\right]^{-1} \\
\Sigma_{H_{DC},H_{DS}} &= \left[\text{diagn}(K_D, 2) - \text{diagn}(JL_D^{-1}J^T, 2)\right]^{-1} \\
\Phi_{H_{SC},H_{SS}} &= \left((\hat{m}_E + \hat{m}_T) - 2JL_S^{-1}(\tilde{m}_E + \tilde{m}_T)\right) \\
\Phi_{H_{SC},H_{DS}} &= \left((\hat{m}_E + \hat{m}_T) - JL_D^{-1}(\tilde{m}_E + \tilde{m}_T)\right) \\
\Phi_{H_{DC},H_{SS}} &= \begin{bmatrix} \hat{m}_E - JL_S^{-1}(\tilde{m}_E + \tilde{m}_T) \\ \hat{m}_T - JL_S^{-1}(\tilde{m}_E + \tilde{m}_T) \end{bmatrix} \\
\Phi_{H_{DC},H_{DS}} &= \begin{bmatrix} \hat{m}_E - JL_D^{-1}\tilde{m}_E \\ \hat{m}_T - JL_D^{-1}\tilde{m}_T \end{bmatrix}
\end{aligned}$$

where $\hat{m}_E = U^T D m_E$ and $\hat{m}_T = U^T D m_T$.

Finally, since the outer posterior is independent of the condition hypothesis, the logarithm of the LR (LLR) can be written as a sum of terms involving the outer posterior and the inner posteriors

$$\text{LLR} = \text{LLR}_o + \text{LLR}_i$$

where

$$\begin{aligned}
\text{LLR}_o &= \log \frac{p(Y_{H_{DS}}|X_{h_c}, M)}{p(y)p(Y_{H_{SS}}|X_{h_c}, M)}\bigg|_0 \\
&= \log|L_2| - \frac{1}{2}\log|L_1| + \frac{1}{2}\tilde{m}_E^T(L_1^{-1} - L_2^{-1})\tilde{m}_E + \frac{1}{2}\tilde{m}_T^T(L_1^{-1} - L_2^{-1})\tilde{m}_T + \tilde{m}_T^T L_1^{-1}\tilde{m}_E \\
\text{LLR}_i &= \log \frac{p(x)p(X_{H_{SC}}|M, H_{SS})^{-1}P_{SS} + p(x)^2 p(X_{H_{DC}}|M, H_{SS})^{-1}P_{DS}}{p(x)p(X_{H_{SC}}|M, H_{DS})^{-1}P_{SD} + p(x)^2 p(X_{H_{DC}}|M, H_{DS})^{-1}P_{DD}}\bigg|_0 \\
&= \log\left(\exp(Q_{H_{SC},H_{SS}})P_{SS} + \exp(Q_{H_{DC},H_{SS}})P_{DS}\right) - \\
&\quad \log\left(\exp(Q_{H_{SC},H_{DS}})P_{SD} + \exp(Q_{H_{DC},H_{DS}})P_{DD}\right)
\end{aligned}$$

where we use the fact that $\log p(x)|_0 = -\frac{1}{2}R_x k$ and $\log p(y)|_0 = -\frac{1}{2}R_y k$, and where $P_{SS} = P(H_{SC}|H_{SS})$, $P_{SD} = P(H_{SC}|H_{DS})$, $P_{DS} = P(H_{DC}|H_{SS})$, and $P_{DD} = P(H_{DC}|H_{DS})$.

References

R. Auckenthaler, M. J. Carey, and J. S. D. Mason. Language dependency in text-independent speaker verification. In *Proc. ICASSP*, Salt Lake City, May 2001.

S. D. Beck, R. Schwartz, and H. Nakasone. A bilingual multi-modal voice corpus for language and speaker recognition (LASR) services. In *Proc. Odyssey-04*, Toledo, Spain, May 2004.

J. Besag. A candidate's formula: A curious result in bayesian prediction. *Biometrika*, 76(1): 183–183, 1989.

N. Brümmer. EM for probabilistic LDA. Technical report, Available at https://sites.google.com/site/nikobrummer/EMforPLDA.pdf, 2010a.

N. Brümmer. EM for simplified PLDA. Technical report, Available at https://sites.google.com/site/nikobrummer/EMforSPLDA.pdf, 2010b.

N. Brümmer. Vb calibration to improve the interface between phone recognizer and i-vector extractor. *arXiv:1510.03203*, 2015.

L. Burget, O. Plchot, S. Cumani, O. Glembek, P. Matejka, and N. Brümmer. Discriminatively trained probabilistic linear discriminant analysis for speaker verification. In *Proc. ICASSP*, Prague, May 2011.

C. Cieri, D. Graff, O. Kimball, D. Miller, and K. Walker. Fisher english training speech part 1 speech ldc2004s13, a.

C. Cieri, D. Graff, O. Kimball, D. Miller, and K. Walker. Fisher english training speech part 2 speech ldc2005s13, b.

C. Cieri, L. Corson, D. Graff, and K. Walker. Resources for new research directions in speaker recognition: The Mixer 3, 4 and 5 corpora. In *Proc. Interspeech*, Antwerp, Belgium, August 2007.

S. Cumani, O. Plchot, and P. Laface. On the use of i-vector posterior distributions in probabilistic linear discriminant analysis. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 22(4):846–857, 2014.

S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, 1980.

N. Dehak, P.J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. Front-end factor analysis for speaker verification. *IEEE Trans. Audio, Speech, and Lang. Process.*, 19(4):788–798, May 2011.

L. Ferrer. Joint probabilistic linear discriminant analysis. *arXiv:1704.02346v2*, 2017.

L. Ferrer, H. Bratt, L. Burget, H. Cernocky, O. Glembek, M. Graciarena, A. Lawson, Y. Lei, P. Matejka, O. Plchot, and N. Scheffer. Promoting robustness for speaker modeling in the community: The PRISM evaluation set. In *Proceedings of SRE11 Analysis Workshop*, Atlanta, USA, December 2011.

M. Gales and S. Young. The application of hidden Markov models in speech recognition. *Foundations and trends in signal processing*, 1(3):195–304, 2008.

D. Garcia-Romero and C.Y. Espy-Wilson. Analysis of i-vector length normalization in speaker recognition systems. In *Proc. Interspeech*, Florence, Italy, August 2011.

D. Garcia-Romero, X. Zhou, and C. Y. Espy-Wilson. Multicondition training of gaussian PLDA models in i-vector space for noise and reverberation robust speaker recognition. In *Proc. ICASSP*, pages 4257–4260, Kyoto, March 2012. IEEE.

M. Graciarena, L. Ferrer, and V. Mitra. The SRI system for the NIST OpenSAD 2015 speech activity detection evaluation. In *Proc. Interspeech*, San Francisco, September 2016.

D. Graff, K. Walker, and A. Canavan. Switchboard-2 phase II LDC99S79, 1999. URL `https://catalog.ldc.upenn.edu/LDC99S79`.

D. Graff, K. Walker, and D. Miller. Switchboard cellular part 1 audio LDC2001S13, 2001. URL `https://catalog.ldc.upenn.edu/LDC2001S13`.

D. Graff, D. Miller, and K. Walker. Switchboard-2 phase III LDC2002S06, 2002. URL `https://catalog.ldc.upenn.edu/LDC2002S06`.

D. Graff, K. Walker, and D. Miller. Switchboard cellular part 2 audio LDC2004S07, 2004. URL `https://catalog.ldc.upenn.edu/LDC2004S07`.

S. Ioffe. Probabilistic linear discriminant analysis. In *Proc. of the 9th European Conference on Computer Vision*, Graz, Austria, 2006.

P. Kenny. Bayesian speaker verification with heavy-tailed priors. In *Proc. Odyssey-10*, Brno, Czech Republic, June 2010. Keynote presentation.

Y. Lei, L. Burget, L. Ferrer, M. Graciarena, and N. Scheffer. Towards noise robust speaker recognition using probabilistic linear discriminant analysis. In *Proc. ICASSP*, Kyoto, March 2012.

P. Li, Y. Fu, U. Mohammed, J. Elder, and S. Prince. Probabilistic models for inference about identity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1):144–157, 2012.

M.-W. Mak, X. Pang, and J.-T. Chien. Mixture of plda for noise robust i-vector speaker verification. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 24(1):130–142, 2016.

A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. The DET curve in assessment of detection task performance. In *Proc. Eurospeech*, Rhodes, Greece, September 1997.

P. Matejka, O. Glembek, F. Castaldo, J. Alam, O. Plchot, P. Kenny, L. Burget, and J. Cernocky. Full-covariance UBM and heavy-tailed PLDA in i-Vector speaker verification. In *Proc. Interspeech*, Florence, Italy, August 2011.

A. Misra and J. H. Hansen. Spoken language mismatch in speaker verification: An investigation with NIST-SRE and CRSS Bi-ling corpora. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 372–377. IEEE, 2014.

S. Prince. Probabilistic linear discriminant analysis for inferences about identity. In *Proceedings of the International Conference on Computer Vision*, 2007.

A. Rozi, D. Wang, L. Li, and T. F. Zheng. Language-aware plda for multilingual speaker recognition. In *Coordination and Standardization of Speech Databases and Assessment Techniques (O-COCOSDA), 2016 Conference of The Oriental Chapter of International Committee for*, pages 161–165. IEEE, 2016.

M. Senoussaoui, P. Kenny, N. Brümmer, N.mmer, E. De Villiers, and P. Dumouchel. Mixture of PLDA models in i-vector space for gender-independent speaker recognition. In *Proc. Interspeech*, pages 25–28, Florence, Italy, August 2011.

Z. Shi, L. Liu, and R. Liu. Multi-view (joint) probability linear discrimination analysis for multi-view feature verification. *arXiv:1704.06061*, 2017.

A. Sizov, K. A. Lee, and T. Kinnunen. Unifying probabilistic linear discriminant analysis variants in biometric authentication. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 464–475. Springer, 2014.