Direct Optimization of the Detection Cost for I-vector based Spoken Language Recognition

Aleksandr Sizov, Kong Aik Lee, *Senior Member, IEEE,* and Tomi Kinnunen, *Member, IEEE*

***Abstract*—We explore a method to boost discriminative capabilities of Probabilistic Linear Discriminant Analysis (PLDA) model without losing its generative advantages. We show a sequential projection and training steps leading to a classifier that operates in the original i-vector space but is discriminatively trained in a low-dimensional PLDA latent subspace. We use extended Baum-Welch technique to optimize the model with respect to two objective functions for discriminative training. One of them is the well-known Maximum Mutual Information (MMI) objective, while the other one is a new objective that we propose to approximate the language detection cost. We evaluate the performance on NIST Language Recognition Evaluation (LRE) 2015 and our development dataset comprised of the utterances from previous LREs. We improve the detection cost by 10% and 6% relative compared to our fine-tuned generative and discriminative baselines, and by 10% over the best of our previously reported results. The proposed approximation method of the cost function and PLDA subspace training are applicable for a broad range of tasks.**

***Index Terms*—discriminative training, language detection, factor analysis, language identification, PLDA**

I. Introduction

SPOKEN language recognition is a task to determine the language spoken in a given speech utterance [1]. From here on we will omit the term "spoken" and refer to it as just language recognition. It is important to distinguish two related subtasks, *language detection* and *language identification*. The former is the task to verify whether a given speech utterance contains the hypothesized language, while language identification aims at recognizing a language from a predefined set. If there is a possibility to encounter an unknown language, it is called an *open-set* task, otherwise it is a *closed-set* task.

Language recognition relies on extraction and modeling of language cues [1]. The cues can be subdivided into high-level *phonotactic* and low-level *acoustic-phonetic* (spectral) ones. In the phonotactic approach, a phone recognizer transcribes input utterances to phone sequences. This is typically followed by an n-gram model [2], where the frequency and order of phones specific to each language are modeled. Over the recent years, spectral features have been established as a basis for state-of-the-art language recognition. Their modeling framework has also evolved considerably: from GMM-UBM systems [3] to GMM-SVM [4] and GMM-MMI [5], followed by state-of-the-art *total variability* (TV) approach [6].

The TV approach has stimulated the development of a series of a back-end classifiers that use low-dimensional i-vectors [6] as an input. The i-vector is a fixed-length representation of a speech utterance of arbitrary duration. Because it contains both speaker, language and channel information, it requires an appropriate classifier to extract and emphasize the relevant factors, while suppressing the irrelevant ones. One of the most useful classifiers for such purpose is *probabilistic linear discriminant analysis* (PLDA) classifier [7], [8]. PLDA combines an intuitive formulation with an efficient learning algorithm and shows state-of-the-art results for speaker recognition. Of course, the choice of i-vector back-ends for language recognition is not limited to PLDA. Simple Gaussian classifier [9], [10], logistic regression [11], cosine scoring [12] and support vector machines [13] have all been successfully applied during recent *language recognition evaluations* (LRE) [14], [15], especially for LRE'11 [14].

A lot of the i-vector based methods were initially developed for speaker recognition and then adopted for language recognition. The key difference between these two is that speaker recognition, used mainly in security, surveillance, and forensics, is almost always an open-set task with a dynamically changing user base. Language recognition tasks, in contrast, involve usually a small number of languages that are known in advance. An example would be telephone-based call service agents to improve customer service. Such closed-set constraint enables language recognition applications to benefit from discriminative methods [1], [16], as they optimize the model parameters using the data from both target and competing classes [17]. Motivated by its success in speech recognition [18], it was first shown in [5] that a Gaussian mixture model trained with maximum likelihood criterion could be refined using MMI discriminative criterion. And more recently, in [10] MMI training was shown to be effective for modeling i-vectors.

In this study, we formulate PLDA scoring as a *language prior estimation problem* and unify conventional by-the-book solution [7], i-vector averaging [19] and minimum-divergence estimation [20] under a common framework. Our initial experiments indicated that direct discriminative optimization in the original i-vector space results in both severe over-fitting and slow optimization. Motivated to tackle this problem, in this study we show that the over-fitting problem can be addressed by performing MMI training on the language priors in the low-dimensional latent subspace of the PLDA, and subsequently

Aleksandr Sizov is with School of Computing, University of Eastern Finland, Finland and the Institute for Infocomm Research, Singapore, 138632. (email: aleksandr.sizov.work@gmail.com)

Tomi Kinnunen is with School of Computing, University of Eastern Finland, Finland. (email: tkinnu@cs.joensuu.fi).

Kong Aik Lee is with the Institute for Infocomm Research, Singapore, 138632. (e-mail: kalee@i2r.a-star.edu.sg).

This work was funded by A*STAR Research Attachment Programme, Singapore and Academy of Finland (grant no. 253120, 283256 and 288558).

"lifting" the parameters back to the original i-vector space. Further, as an alternative to the MMI cost, we propose a new objective function that directly minimizes an approximated version of the primary language detection cost. Like the MMI objective function, it uses similar optimization approach, based on *extended Baum-Welch* equations [17], [21]. Unlike the MMI objective function, however, it takes into account only the misclassified samples and confusable samples close to the decision boundary. To validate our theoretical contribution, we carry out experiments on the latest NIST LRE'15 dataset.

We are aware of an earlier attempt to discriminatively optimize detection cost function for language recognition [22]. That method is based on phonotactic *parallel phoneme recognizer vector space modeling* (PPR-VSM) paradigm [16] and operates in pair-wise SVM score space. The most important difference is that the method in [22] forms detection *log-likelihood ratios* (LLR) by pair-wise GMMs explicitly modeling the target and non-target hypotheses for each language. We do not need the latter models because after our classifiers provide a likelihood for each test sample and each language, LLR is computed in a straightforward and mathematically strict manner (2).

A preliminary report of our study appears in [8]. The present article presents a more unified and self-contained formulation of a discriminative cost optimization and two different approaches using i-vector duration information in the system. Furthermore, we consider different adaptation scopes and covariance matrix tying scopes for all classifiers and extend experiments to an additional dataset.

II. Language detection

We first give a brief introduction to i-vector focusing on its use for language detection task. We then introduce a general formulation of language detection cost function as the performance metric suitable for multi-cluster and open-set language detection task.

A. Language detection in i-vector space

I-vector is a way to represent a speech utterance, regardless of its duration, as a single, fixed-dimensional vector [6]. Specifically, it is a *Maximum a Posteriori* (MAP) estimate of a latent variable in a multi-Gaussian factor analysis model based on a special *Gaussian mixture model* (GMM), known as the *universal background model* (UBM). Formally, an i-vector $\boldsymbol{\varphi}$ is inferred as follows:

$$\boldsymbol{\varphi} = \arg\max_{\mathbf{x}} \left[\prod_{j=1}^{J} \prod_{h=1}^{H_j} \mathcal{N}\left(\mathbf{o}_h \middle| \boldsymbol{\mu}_j + \mathbf{T}_j\mathbf{x}, \boldsymbol{\Sigma}_j\right) \right] \mathcal{N}\left(\mathbf{x}|\mathbf{0}, \mathbf{I}\right) , \tag{1}$$

where $\{\mathbf{o}_h\}_{h=1}^{H_j}$ is a set of acoustic feature vectors for a given utterance aligned to the j-th mixture component, $\{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}_{j=1}^{J}$ are the mean vectors and covariance matrices of the UBM with J components, and $\{\mathbf{T}_j\}_{j=1}^{J}$ are blocks of the *total variability* matrix $\mathbf{T}$. Different from speaker recognition, it is important to use speech utterances from multiple languages to train both the UBM and the total variability matrix.

I-vector can be viewed as a compressed representation of the GMM mean supervector. The low dimensionality of i-vectors allows simple and effective models to be used. Indeed, it was found in [9] that each language could simply be modeled as a Gaussian in the i-vector space. For language detection, we evaluate the log-likelihood ratio score of language $\mathcal{L}_i$ ($i \in \mathcal{C}$) for a given test i-vector $\boldsymbol{\varphi}_n$:

$$\mathrm{llr}(\mathcal{L}_i|\boldsymbol{\varphi}_n) = \log \frac{p(\boldsymbol{\varphi}_n|\mathcal{L}_i)}{\frac{1}{|\mathcal{C}|-1}\sum\limits_{\substack{j\in\mathcal{C}\\ j\neq i}} p(\boldsymbol{\varphi}_n|\mathcal{L}_j)} . \tag{2}$$

Here, $\mathcal{C}$ consists of all languages considered in the detection task. In a closed-set scenario, $\mathcal{C} = \{\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_M\}$ represents M explicitly specified languages. In an open-set scenario, out-of-set languages are treated as one "none-of-the-above" language in the cluster $\mathcal{C}$.

B. Language detection cost

Let M be the number of languages and let $\Phi_i = \{\boldsymbol{\varphi}_n\}_{n=1}^{N_i}$ denote the collection of the training i-vectors of language $\mathcal{L}_i$, and $N = \sum_{i=1}^{M} N_i$ be the total number of i-vectors. Considering a general case, where closely related languages are grouped into clusters (e.g., different dialects of Arabic), we introduce the notation $\mathcal{C}_k$ to indicate the k-th language cluster and $\{\mathcal{C}_k\}_{k=1}^{K}$ as a collection of language clusters. In particular, the set $\mathcal{C}_k$ contains the indices of languages that belong to the k-th cluster. For $K = 1$, the formulation reduces to the ordinary language detection task.

The primary objective of language detection [1], as defined by NIST [15], is to minimize the *average detection cost function*, $\mathrm{DCF}^{\mathrm{avg}}$, which has the following form for each cluster $\mathcal{C}_k$:

$$\mathrm{DCF}_k^{\mathrm{avg}} = \frac{1}{2|\mathcal{C}_k|}\left(\sum_{i\in\mathcal{C}_k} \mathrm{FRR}(\mathcal{L}_i) + \frac{1}{|\mathcal{C}_k|-1} \sum_{\substack{(i,j)\in\mathcal{C}_k\times\mathcal{C}_k\\ i\neq j}} \mathrm{FAR}(\mathcal{L}_i, \mathcal{L}_j) \right) . \tag{3}$$

Here, $\mathrm{FRR}(\bullet)$ and $\mathrm{FAR}(\bullet)$ are the false rejection and false acceptance rates, respectively. To compute them, we compare the language detection scores against a threshold θ and apply an indicator function $I(\bullet)$, as follows:

$$\mathrm{FRR}(\mathcal{L}_i) = \frac{1}{N_i} \sum_{\boldsymbol{\varphi}_n\in\Phi_i} I\{\mathrm{llr}(\mathcal{L}_i|\boldsymbol{\varphi}_n) < \theta\}, \tag{4}$$

$$\mathrm{FAR}(\mathcal{L}_i, \mathcal{L}_j) = \frac{1}{N_i} \sum_{\boldsymbol{\varphi}_n\in\Phi_i} I\{\mathrm{llr}(\mathcal{L}_j|\boldsymbol{\varphi}_n) \geqslant \theta\}. \tag{5}$$

Note that in (3), the costs for making both types of errors are assumed equal. We also assume that a target class has an equal probability with a non-target class. With this, the detection threshold θ can be set to 0 in (4) and (5).

III. Using PLDA for language detection

In this section, we provide details for PLDA model [7] and present a unified approach to three different scoring methods, that we will later use in Section IV for discriminative training.

A. Probabilistic LDA

Probabilistic LDA [7] is a Gaussian model with a structured covariance to split apart channel and language variability. In this study, we advocate using simplified PLDA [23]. The reader may refer to [24] for further discussion of other variants of PLDA. The simplified PLDA, takes the following form:

$$\boldsymbol{\varphi}_n = \boldsymbol{\mu} + \mathbf{V}\mathbf{y}_i + \boldsymbol{\varepsilon}_n \,, \tag{6}$$

where $\boldsymbol{\varphi}_n$ is an i-vector from language $\mathcal{L}_i$, $\boldsymbol{\mu}$ is the global mean vector, $\mathbf{V}$ is the factor loading matrix, $\mathbf{y}_i$ is the language-dependent latent variable, and $\boldsymbol{\varepsilon}_n \sim \mathcal{N}\left(\boldsymbol{\varepsilon}_n \middle| \mathbf{0}, \boldsymbol{\Lambda}^{-1}\right)$ models the residual.

The prior imposed on the latent variable $\mathbf{y}_i$ determines the resulting PLDA distribution. In this paper, we constrain it to be a Gaussian:

$$\mathbf{y}_i \sim \mathcal{N}\left(\mathbf{y}_i | \mathbf{m}_i, \mathbf{G}_i\right) \,. \tag{7}$$

Using (7) in (6) and integrating out the latent variable $\mathbf{y}_i$, we arrive at

$$\begin{aligned} p(\boldsymbol{\varphi}_n|\mathcal{L}_i) &= \int \mathcal{N}\left(\boldsymbol{\varphi}_n \middle| \boldsymbol{\mu} + \mathbf{V}\mathbf{y}_i, \boldsymbol{\Lambda}^{-1}\right) \mathcal{N}\left(\mathbf{y}_i | \mathbf{m}_i, \mathbf{G}_i\right) \mathbf{d}\mathbf{y}_i \\ &= \mathcal{N}\left(\boldsymbol{\varphi}_n \middle| \boldsymbol{\mu} + \mathbf{V}\mathbf{m}_i, \mathbf{V}\mathbf{G}_i^{-1}\mathbf{V}^\top + \boldsymbol{\Lambda}^{-1}\right) \,, \end{aligned} \tag{8}$$

which is a Gaussian distribution with mean $\boldsymbol{\mu} + \mathbf{V}\mathbf{m}_i$ and covariance $\mathbf{V}\mathbf{G}_i^{-1}\mathbf{V}^\top + \boldsymbol{\Lambda}^{-1}$, where $\mathbf{m}_i$ and $\mathbf{G}_i$ depend on the language $\mathcal{L}_i$, while the parameters $\{\boldsymbol{\mu}, \mathbf{V}, \boldsymbol{\Lambda}\}$ are shared across the languages. Assuming a standard Gaussian prior on $\mathbf{y}_i$, where $\mathbf{m}_i = \mathbf{0}$ and $\mathbf{G}_i = \mathbf{I}$, (7) reduces to the standard PLDA

$$p(\boldsymbol{\varphi}_n) = \mathcal{N}\left(\boldsymbol{\varphi}_n \middle| \boldsymbol{\mu}, \mathbf{V}\mathbf{V}^\top + \boldsymbol{\Lambda}^{-1}\right) \,. \tag{9}$$

It is computationally more efficient [25], [20] to evaluate the likelihood ratio

$$l(\boldsymbol{\varphi}_n|\mathcal{L}_i) = \frac{p(\boldsymbol{\varphi}_n|\mathcal{L}_i)}{p(\boldsymbol{\varphi}_n)} \tag{10}$$

between language-specific and default PLDA whereby common terms will be canceled out. Notice that using either (8) or (10) for $p(\boldsymbol{\varphi}_n|\mathcal{L}_i)$ in (2) results in the same detection score.

B. Estimating language-specific prior

Equation (8) involves two language-specific terms, $\mathbf{m}_i$ and $\mathbf{G}_i^{-1}$, introduced through the prior imposed on $\mathbf{y}_i$. We consider three different approaches to estimating them from a given set $\Phi_i = \{\boldsymbol{\varphi}_n\}_{n=1}^{N_i}$ of training i-vectors for each language. For the sake of notational simplicity, we use the same notations $\{\mathbf{m}_i, \mathbf{G}_i^{-1}\}$ for all three approaches.

(i) **By-the-book scoring***:* We treat all i-vectors in Φ_i as independent sessions. The parameters $\mathbf{m}_i$ and $\mathbf{G}_i^{-1}$ are taken as the posterior mean and covariance of the latent variable $\mathbf{y}_i$ given Φ_i. This leads to the following mean vector and covariance estimates:

$$\mathbf{m}_i = \mathbf{G}_i^{-1}\mathbf{V}^\top\boldsymbol{\Lambda} \sum_{\boldsymbol{\varphi}_n \in \Phi_i} \left(\boldsymbol{\varphi}_n - \boldsymbol{\mu}\right) , \tag{11}$$

$$\mathbf{G}_i^{-1} = \left(\mathbf{I} + N_i\mathbf{V}^\top\boldsymbol{\Lambda}\mathbf{V}\right)^{-1} . \tag{12}$$

Although this approach gives us a proper posterior estimate, it generally results in highly peaked distribution for large N_i in (12) and therefore an overconfident prediction. Using (11) and (12) in (10), we obtain the so-called by-the-book PLDA scoring [7].

(ii) **I-vector averaging***:* We first take the average of all i-vectors in Φ_i. The parameters $\mathbf{m}_i$ and $\mathbf{G}_i^{-1}$ are then estimated from the posterior distribution of the latent variable $\mathbf{y}_i$ given the *average i-vector*, as follows:

$$\mathbf{m}_i = \mathbf{G}_i^{-1}\mathbf{V}^\top\boldsymbol{\Lambda} \left(\frac{1}{N_i} \sum_{\boldsymbol{\varphi}_n \in \Phi_i} \boldsymbol{\varphi}_n - \boldsymbol{\mu} \right) , \tag{13}$$

$$\mathbf{G}_i^{-1} = \left(\mathbf{I} + \mathbf{V}^\top\boldsymbol{\Lambda}\mathbf{V}\right)^{-1} . \tag{14}$$

Due to its simplicity, this approach has been the most widely used one. Another reason for its success is that the posterior covariance $\mathbf{G}_i^{-1}$ is independent of N_i, thus avoiding the covariance shrinking issue of the first approach.

(iii) **Minimum divergence estimation***:* Different from that in (ii), we first infer the posterior distribution $p(\mathbf{y}_{ij}|\boldsymbol{\varphi}_n)$ for each i-vector in Φ_i. The parameters $\mathbf{m}_i$ and $\mathbf{G}_i^{-1}$ are obtained as the mean vector and covariance matrix of a Gaussian distribution that gives the minimum sum of KL divergence from all $p(\mathbf{y}_{ij}|\boldsymbol{\varphi}_n)$. As shown in [20], this is given by

$$\mathbf{m}_i = \left(\mathbf{I} + \mathbf{V}^\top\boldsymbol{\Lambda}\mathbf{V}\right)^{-1}\mathbf{V}^\top\boldsymbol{\Lambda} \left(\frac{1}{N_i} \sum_{\boldsymbol{\varphi}_n \in \Phi_i} \boldsymbol{\varphi}_n - \boldsymbol{\mu} \right) , \tag{15}$$

$$\mathbf{G}_i^{-1} = \left(\mathbf{I} + \mathbf{V}^\top\boldsymbol{\Lambda}\mathbf{V}\right)^{-1} + \frac{1}{N_i} \sum_{\boldsymbol{\varphi}_n \in \Phi_i} (\boldsymbol{\phi}_n - \mathbf{m}_i)(\boldsymbol{\phi}_n - \mathbf{m}_i)^\top \,, \tag{16}$$

where

$$\boldsymbol{\phi}_n = \left(\mathbf{I} + \mathbf{V}^\top\boldsymbol{\Lambda}\mathbf{V}\right)^{-1}\mathbf{V}^\top\boldsymbol{\Lambda}(\boldsymbol{\varphi}_n - \boldsymbol{\mu}) \tag{17}$$

is the projection of an i-vector $\boldsymbol{\varphi}_n$ to the latent space. Compared to that in (ii), the covariance estimate $\mathbf{G}_i^{-1}$ in (16) has an additional empirical covariance estimate in the latent subspace.

IV. Discriminative Training for PLDA

In this section, we show that a generatively estimated PLDA model, as presented in Section III, can be re-estimated with discriminative training. In particular, we advocate the use of the conventional MMI criterion but also propose to directly minimize the detection cost in (3). We further show how this is accomplished in the i-vector space or the latent subspace.

A. Discriminative training of generative model

Central to discriminative training is a cost function that takes into account a functional dependence between the input features and the output labels without explicitly modeling input data [26]. In our case, the elementary building block of the cost function is the log-posterior class probability:

$$\log p(\mathcal{L}_i|\boldsymbol{\varphi}_n) = \log \frac{p(\boldsymbol{\varphi}_n|\mathcal{L}_i)}{\sum\limits_{j \in \mathcal{C}} p(\boldsymbol{\varphi}_n|\mathcal{L}_j)} \,, \tag{18}$$

where $\mathcal{C}$ is a set of language (or class) indices. We follow the general methodology from [21] that relies on so-called *weak-sense auxiliary functions* for optimization. It is easy to show

that the following function is a weak-sense auxiliary function for (18):

$$\log p(\boldsymbol{\varphi}_n|\mathcal{L}_i) - \sum_{j\in\mathcal{C}} \frac{q(\boldsymbol{\varphi}_n|\mathcal{L}_j)}{\sum\limits_{h\in\mathcal{C}} q(\boldsymbol{\varphi}_n|\mathcal{L}_h)} \log p(\boldsymbol{\varphi}_n|\mathcal{L}_j)\,, \qquad (19)$$

where $\{q(\boldsymbol{\varphi}_n|\mathcal{L}_\bullet)\}$ is set of Gaussian distributions with parameters from the previous iteration of the training algorithm.

For the case of *maximum mutual information* (MMI) training [21], we aim at maximizing the posterior probability of the correct class given the training data across all classes, as follows:

$$\mathcal{Q}_{\text{MMI}} = \sum_{i\in\mathcal{C}} \sum_{\boldsymbol{\varphi}_n\in\Phi_i} \log p(\mathcal{L}_i|\boldsymbol{\varphi}_n) = \sum_{i\in\mathcal{C}} \sum_{\boldsymbol{\varphi}_n\in\Phi_i} \log \frac{p(\boldsymbol{\varphi}_n|\mathcal{L}_i)}{\sum\limits_{j\in\mathcal{C}} p(\boldsymbol{\varphi}_n|\mathcal{L}_j)}\,. \qquad (20)$$

The default MMI objective (20) assumes that all the languages have an equal prior probability. In our implementation, as detailed below, we found that balancing the classes by the amount of their training data points is beneficial.

We maximize the balanced version of (20) via the *Extended Baum-Welch equations* [21] which amounts to the iterative computation of the following normalized zero-, first-, and second-order sufficient statistics per language, followed by Gaussian parameter updates:

$$s_i^0 = \frac{1}{N_i} \sum_{\boldsymbol{\varphi}_n\in\Phi_i} 1 - \sum_{j\in\mathcal{C}} \frac{1}{N_j} \sum_{\boldsymbol{\varphi}_n\in\Phi_j} p(\mathcal{L}_i|\boldsymbol{\varphi}_n)\,, \qquad (21)$$

$$\mathbf{s}_i^1 = \frac{1}{N_i} \sum_{\boldsymbol{\varphi}_n\in\Phi_i} \boldsymbol{\varphi}_n - \sum_{j\in\mathcal{C}} \frac{1}{N_j} \sum_{\boldsymbol{\varphi}_n\in\Phi_j} p(\mathcal{L}_i|\boldsymbol{\varphi}_n)\boldsymbol{\varphi}_n\,,$$

$$\mathbf{S}_i^2 = \frac{1}{N_i} \sum_{\boldsymbol{\varphi}_n\in\Phi_i} \boldsymbol{\varphi}_n\boldsymbol{\varphi}_n^\top - \sum_{j\in\mathcal{C}} \frac{1}{N_j} \sum_{\boldsymbol{\varphi}_n\in\Phi_j} p(\mathcal{L}_i|\boldsymbol{\varphi}_n)\boldsymbol{\varphi}_n\boldsymbol{\varphi}_n^\top\,.$$

To ensure that the covariance matrices are positive definite, we smooth the sufficient statistics with a positive coefficient λ. Its value also affects the convergence speed of the algorithm. In our experiments, we linearly increase λ after each iteration [21]. Further, to prevent over-fitting, we regularize our equations with τ data points that have zero mean and unit covariance. Notice that in the balanced MMI version — where the weight of each class is 1— coefficient τ corresponds to percentage rather than raw counts.

$$s_i^0 \leftarrow s_i^0 + \lambda + \tau\,, \qquad (22)$$

$$\mathbf{s}_i^1 \leftarrow \mathbf{s}_i^1 + \lambda\boldsymbol{\mu}_i\,, \qquad (23)$$

$$\mathbf{S}_i^2 \leftarrow \mathbf{S}_i^2 + \lambda(\boldsymbol{\mu}_i\boldsymbol{\mu}_i^\top + \boldsymbol{\Sigma}_i) + \tau\mathbf{I}\,. \qquad (24)$$

Afterwards, the parameter updates are as follows

$$\boldsymbol{\mu}_i = \frac{\mathbf{s}_i^1}{s_i^0}\,, \qquad (25)$$

$$\boldsymbol{\Sigma}_i = \frac{\mathbf{S}_i^2 - \boldsymbol{\mu}_i\mathbf{s}_i^{1^\top} - \mathbf{s}_i^1\boldsymbol{\mu}_i^\top + s_i^0\boldsymbol{\mu}_i\boldsymbol{\mu}_i^\top}{s_i^0}\,. \qquad (26)$$

B. Direct optimization of detection cost

We notice that the posterior probabilities used in MMI objective function (20) have essentially the same structure as the log-likelihood ratios (2) used in cluster cost function. This

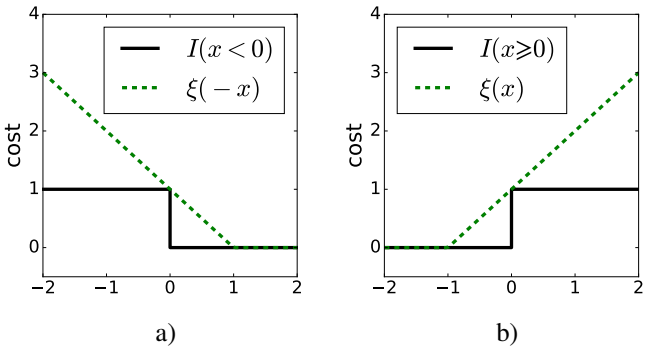

Fig. 1. Hinge loss approximation for a) false rejection and b) false acceptance errors as defined in (4) and (5) respectively.

makes it possible to apply MMI-like optimization to *directly* maximize the primary cost function (3). The main difficulty in doing so is that (3) relies on 0-1 classification loss that is discrete and non-differentiable. To address this issue, we approximate the 0-1 loss in (4)–(5) with its continuous convex upper-bound in a form of a *hinge function* [27] (see Fig. 1), as follows:

$$I(x<0) \approx \xi(-x), \qquad (27)$$

$$I(x\geqslant 0) \approx \xi(x)\,, \qquad (28)$$

where $\xi(x) = \max(x+1, 0)$. Initially, we considered a more general case of $\xi(x) = \max(\alpha x + 1, 0)$ but it had a minimal influence on the performance, so we present a simpler model here.

Thus, the approximated cost function for cluster $\mathcal{C}_k$ takes the form

$$\mathcal{Q}_k^{\text{avg}} = \frac{1}{2|\mathcal{C}_k|} \sum_{i\in\mathcal{C}_k} \frac{1}{N_i} \sum_{\boldsymbol{\varphi}_n\in\Phi_i} \{\xi(-\text{llr}(\mathcal{L}_i|\boldsymbol{\varphi}_n)) + \frac{1}{|\mathcal{C}_k|-1} \sum_{\substack{j\in\mathcal{C}_k \\ j\neq i}} \xi(\text{llr}(\mathcal{L}_j|\boldsymbol{\varphi}_n))\}, \qquad (29)$$

where $\text{llr}(\bullet)$ comes from (2). Let us agglomerate all i-vectors that result in a non-zero $\xi(\bullet)$ values into sets Ψ_{ij}:

$$\Psi_{ij} = \left\{ \boldsymbol{\varphi}_n \in \Phi_i : \begin{cases} \text{llr}(\mathcal{L}_j|\boldsymbol{\varphi}_n) < 1, & i = j \\ \text{llr}(\mathcal{L}_j|\boldsymbol{\varphi}_n) > -1, & i \neq j \end{cases} \right\}. \qquad (30)$$

Because we aim at minimizing the detecion cost but maximizing the objective function, we reverse the signs in (29). We then substitute $x+1$ instead of $\xi(x)$ for all non-zero values and remove all the constant terms. The revised objective function is as follows:

$$\widehat{\mathcal{Q}}_k^{\text{avg}} = \sum_{i\in\mathcal{C}} \frac{1}{N_i} \left\{ \sum_{\boldsymbol{\varphi}_n\in\Psi_{ii}} \text{llr}(\mathcal{L}_i|\boldsymbol{\varphi}_n) - \frac{1}{|\mathcal{C}|-1} \sum_{\substack{j\in\mathcal{C} \\ j\neq i}} \sum_{\boldsymbol{\varphi}_n\in\Psi_{ij}} \text{llr}(\mathcal{L}_j|\boldsymbol{\varphi}_n) \right\}. \qquad (31)$$

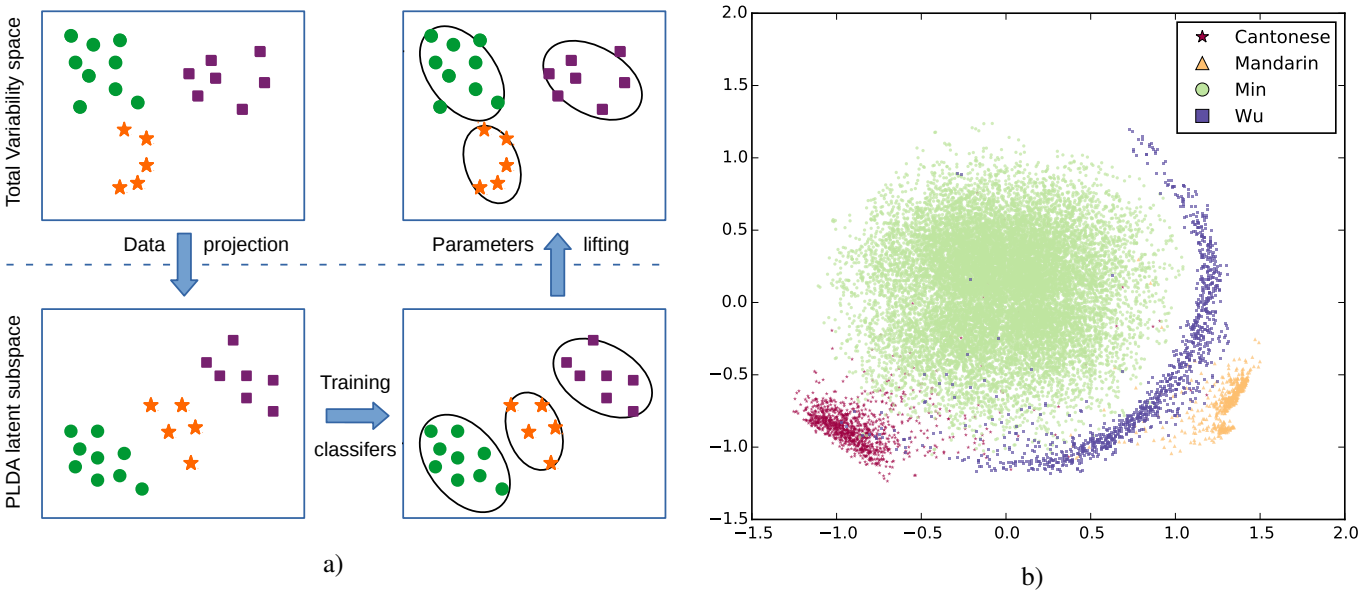

Fig. 2. a) A schematic illustration of a discriminative training in a Probabilistic LDA latent subspace. Each point corresponds to an i-vector and each color to a language. b) Two-dimensional projection of Chinese language cluster training set. Each point corresponds to an i-vector in a 20-dimensional Probabilistic LDA latent subspace. We used an accelerated version of t-SNE algorithm [28] to produce the picture. Both pictures are best viewed in color.

As in the case of MMI, we first compute a weak-sense auxiliary function (19) for each $\mathrm{llr}(\mathcal{L}_\bullet|\boldsymbol{\varphi}_n)$ defined in (2), as follows:

$$\log p(\boldsymbol{\varphi}_n|\mathcal{L}_i) - \sum_{\substack{j\in\mathcal{C}\\ j\neq i}} \frac{q(\boldsymbol{\varphi}_n|\mathcal{L}_j)}{\sum_{\substack{h\in\mathcal{C}\\ h\neq i}} q(\boldsymbol{\varphi}_n|\mathcal{L}_h)} \log p(\boldsymbol{\varphi}_n|\mathcal{L}_j)\,. \quad (32)$$

Here, $\{q(\boldsymbol{\varphi}_n|\mathcal{L}_\bullet)\}$ is a set of Gaussian distributions with fixed parameters that, typically, correspond to the parameters from the previous iteration of the training algorithm. After that, we replace all $\mathrm{llr}(\mathcal{L}_\bullet|\boldsymbol{\varphi}_n)$ terms in (31) with the weak-sense auxiliary functions from (32) and iteratively maximize the resultant sum with the help of Extended Baum-Welch equations [21]. The sufficient statistics of each language are computed as follows:

$$s_i^0 = \frac{|\mathcal{C}|-1}{N_i} \sum_{\boldsymbol{\varphi}_n\in\Psi_{ii}} 1 + \sum_{j\in\mathcal{C}} \frac{1}{N_j} \sum_{\substack{h\in\mathcal{C}\\ h\neq j\\ h\neq i}} \sum_{\boldsymbol{\varphi}_n\in\Psi_{jh}} \gamma_{ni}^{-h}$$
$$- \sum_{\substack{j\in\mathcal{C}\\ j\neq i}} \frac{1}{N_j} \left[(|\mathcal{C}|-1) \sum_{\boldsymbol{\varphi}_n\in\Psi_{jj}} \gamma_{ni}^{-j} + \sum_{\boldsymbol{\varphi}_n\in\Psi_{ji}} 1 \right], \quad (33)$$

$$\mathbf{s}_i^1 = \frac{|\mathcal{C}|-1}{N_i} \sum_{\boldsymbol{\varphi}_n\in\Psi_{ii}} \boldsymbol{\varphi}_n + \sum_{j\in\mathcal{C}} \frac{1}{N_j} \sum_{\substack{h\in\mathcal{C}\\ h\neq j\\ h\neq i}} \sum_{\boldsymbol{\varphi}_n\in\Psi_{jh}} \gamma_{ni}^{-h} \boldsymbol{\varphi}_n$$
$$- \sum_{\substack{j\in\mathcal{C}\\ j\neq i}} \frac{1}{N_j} \left[(|\mathcal{C}|-1) \sum_{\boldsymbol{\varphi}_n\in\Psi_{jj}} \gamma_{ni}^{-j} \boldsymbol{\varphi}_n + \sum_{\boldsymbol{\varphi}_n\in\Psi_{ji}} \boldsymbol{\varphi}_n \right], \quad (34)$$

$$\mathbf{S}_i^2 = \frac{|\mathcal{C}|-1}{N_i} \sum_{\boldsymbol{\varphi}_n\in\Psi_{ii}} \boldsymbol{\varphi}_n\boldsymbol{\varphi}_n^\top + \sum_{j\in\mathcal{C}} \frac{1}{N_j} \sum_{\substack{h\in\mathcal{C}\\ h\neq j\\ h\neq i}} \sum_{\boldsymbol{\varphi}_n\in\Psi_{jh}} \gamma_{ni}^{-h} \boldsymbol{\varphi}_n\boldsymbol{\varphi}_n^\top$$
$$- \sum_{\substack{j\in\mathcal{C}\\ j\neq i}} \frac{1}{N_j} \left[(|\mathcal{C}|-1) \sum_{\boldsymbol{\varphi}_n\in\Psi_{jj}} \gamma_{ni}^{-j} \boldsymbol{\varphi}_n\boldsymbol{\varphi}_n^\top + \sum_{\boldsymbol{\varphi}_n\in\Psi_{ji}} \boldsymbol{\varphi}_n\boldsymbol{\varphi}_n^\top \right], \quad (35)$$

where

$$\gamma_{ni}^{-j} = \frac{q(\boldsymbol{\varphi}_n|\mathcal{L}_j)}{\sum_{\substack{h\in\mathcal{C}\\ h\neq i}} q(\boldsymbol{\varphi}_n|\mathcal{L}_h)}\,. \quad (36)$$

Afterwards, we smooth and regularize the sufficient statistics (22)–(24) and update the parameters (25)–(26) the same way as in MMI training. Computation of sufficient statistics and parameter updates are applied iteratively until either a convergence or a predefined number of iterations is reached.

C. Discriminative training in PLDA latent space

In the above, we apply discriminative training by taking the already trained language-specific distribution $p(\boldsymbol{\varphi}_n|\mathcal{L}_i)$. For the case of PLDA, if we optimize class mean vectors and class covariance matrices in the original i-vector space, this would result in implicit changes for the PLDA parameters for each language which is undesirable. Thus, we propose to perform MMI training in the latent subspace consisting of the following steps:

1) **Probabilistic projection to the latent space**: project an i-vector onto the low-dimensional language subspace by inference of the posterior mean of the latent variable $\mathbf{y}_i$. This projection is given by (17).
2) **Minimum divergence estimation**: estimate the mean vector $\mathbf{m}_i$ and covariance matrix $\mathbf{G}_i^{-1}$ using (15)–(16).
3) **Discriminative training**: retrain the mean vectors and covariance matrices to optimize for better separation between classes using either the MMI or the detection cost.

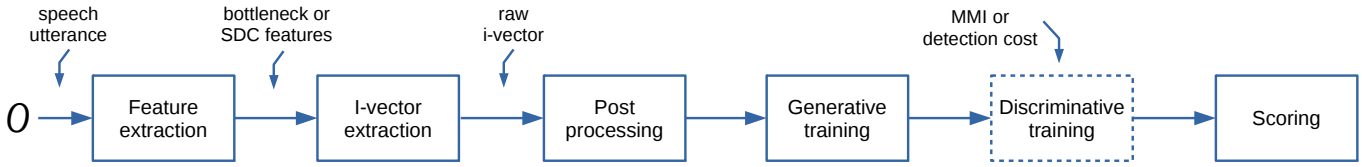

Fig. 3. Block diagram of our language detection system. We use bottleneck features for LRE'15 and shifted delta cepstra features for I2R Dev, while all back-end preprocessing and classifiers for i-vectors are the same.

4) **Parameter lifting**: Lift the mean vectors and covariance matrices from the latent space back to the i-vector space. This is given by (8).

The advantage of performing discriminative training in the latent subspace is that the PLDA structure, as in (6), is preserved. In particular, the global mean $\boldsymbol{\mu}$, factor loading matrix $\mathbf{V}$ and residual covariance $\boldsymbol{\Lambda}^{-1}$ are preserved while the language-specific priors are retrained. Fig. 2 illustrates the 4 steps as described above taking a simplified example in a 2-dimensional vector space. Notice also the projection and lifting procedures are based on the posterior inference and marginalization defined for factor analysis model.

V. Experimental settings

A. Corpora

We conducted experiments on the recent LRE'15 and I2R Dev corpora. The latter was constructed by us in preparation towards LRE'15 submission. LRE'15 focuses on multi-cluster closed-set identification task. We constructed I2R Dev set in a similar manner by reusing samples from previous LREs, including 96, 03, 05, 07, 09 and 11. A summary of both datasets is presented in Table I. Note, that for the reasons explained in Subsection VI-B, the training sets that we present in this table contain *only* utterances longer than 15 seconds for LRE'15 and longer that 3 seconds for I2R Dev. LRE'15 comprises $M = 20$ target languages partitioned into $K = 6$ clusters. The official results of NIST LRE'15 revealed that development and evaluation datasets have a severe mismatch for all language clusters, especially for French — the majority of participants obtained Equal Error Rates close to 50% on this cluster. Therefore, we decided to, firstly, carefully split (without speaker overlap) the evaluation set into two disjoint parts: $1/3$ for tuning and $2/3$ for the final testing. Secondly, we exclude French cluster from scoring. Since some of the LRE'15 languages were absent in previous evaluations, I2R Dev set has a reduced number of languages and clusters as shown in the table.

TABLE I
Summary of LRE'15 and I2R Dev datasets.

Cluster	Languages	Number of i-vectors			
		LRE'15		I2R Dev	
		Train	Test	Train	Test
Arabic	Egyptian	13962	5355	237	240
	Iraqi	5661	5987	100	1224
	Levantine	6599	4516	100	1224
	Maghrebi	5568	5504	100	1215
	Modern Standard	400	1630	100	1281
Chinese	Cantonese	330	14984	730	240
	Mandarin	11446	4013	4345	1536
	Min	646	5683	93	240
	Wu	653	4982	131	240
English	British	47	5319	—	—
	General American	14467	4646	4572	1596
	Indian	862	4631	1463	1728
Slav.	Russain	2175	2039	1080	1803
	Polish	4324	3205	100	1443
Fren.	West African	315	4605	—	—
	Haitian Creole	319	19195	—	—
Iberian	Caribbean Spanish	4310	1541	—	—
	European Spanish	566	3854	—	—
	Latin Amer. Spanish	656	4656	—	—
	Brazilian Portuguese	92	3089	—	—
Duration (sec): $\mu \pm \sigma$		22 ± 7	9 ± 7	20 ± 10	14 ± 11

B. I-vector extraction and preprocessing

Our main set of i-vectors was prepared by "Fantastic 4" team [29], [30] during NIST LRE'15. They are based on 40-dimensional normalized filter bank features with the first and second order derivatives, followed by a Deep Neural Network (DNN) with a bottleneck layer [31], [32]. The DNN was trained on the Switchboard land-line data. It takes 21 stacked frames as an input (2520 units) and has 6 consecutive hidden layers with 1024 units and a prefinal bottleneck layer with 64 units. The output layer has 6111 units corresponding to 6111 senones. The i-vector dimensionality is set to 600. Refer to [29] for more details (this set of i-vectors was abbreviated as BNF2 in that paper). To increase i-vector diversity, we used a simple data augmentation technique for LRE'15, where long original utterances were processed as a single i-vector and also cut into several utterances with smaller durations. For I2R Dev, we used a front-end that represents state-of-the-art for LRE'11. Namely, the i-vectors were extracted using a UBM with 512 mixtures trained on shifted delta cepstra (SDC) features and the $\mathbf{T}$ matrix has a rank of 400. All the above parameter settings were based on empirical observations on development sets which provided a reasonable compromise between computational complexity and the amount of training data.

Fig. 3 presents the pipeline of our language detection system. To make i-vectors more suitable for Gaussian-based models [33], we whitened them (using whitening matrix computed from the training data) and then projected to the unit sphere. *Within class covariance normalization* [34] (WCCN) was not found helpful. *Linear discriminant analysis* (LDA) transformation on the raw i-vectors was of some help and we evaluate its impact in the experiments. We follow the observation in our previous study [8] that LDA projection prior to modeling stage is beneficial for Gaussian classifier

but detrimental for the PLDA classifier. For each corpus, we set the LDA subspace to the number of languages a corpus has to make it comparable with the PLDA latent language subspace model. Results for a tuning set are presented with 10-fold cross-validation on the training set.

VI. Experiments

Unlike the results in our preliminary study [8], where we excluded the utterances with speech duration of less than one second from both training and tuning sets, we conduct a more detailed investigation of this issue in Subsection VI-B. For this reason, the results of our baseline systems slightly differ from those reported in the paper.

A. Baseline generative models

Table II compares different scoring approaches for the simplified PLDA classifier, described in Subsection III-A. It was trained on all the languages at once (20 for LRE'15 and 13 for I2R Dev). The results for both datasets present a consistent ordering for PLDA scoring methods, with minimum-divergence scoring to be the best one. Its comparison with the averaged i-vector scoring tells us that the additional uncertainty term in the covariance matrix (16) is beneficial for language identification.

TABLE II
Evaluation of the cost performance (%) for simplified PLDA classifier on the tune set. The results comprise by-the-book (book) (11) – (12), averaged (avg) i-vector (13) – (14), and minimum-divergence (min-div) scoring (15) – (16).

	LRE'15			I2R Dev		
	book	avg	min-div	book	avg	min-div
Arabic	23.15	23.81	23.53	7.77	7.45	7.33
Chinese	15.66	17.43	16.59	9.03	8.25	8.29
English	15.21	13.49	13.06	11.25	12.84	12.31
Slavic	6.41	6.63	6.64	3.36	3.71	2.96
Iberian	24.20	24.25	23.67	—	—	—
Average	16.93	17.12	**16.70**	7.85	8.06	**7.72**

B. Use of duration information and calibration of the scores

Duration of a speech utterance is an important characteristic that directly affects the degree of uncertainty of the i-vectors [35]. We address this issue in two ways. Firstly, we study the removal of all training utterances shorter than a certain duration. This allows us to concentrate only on i-vectors deemed more reliable. Secondly, following the results in [36], [37], presented at the post-evaluation workshop, we apply a scaling of the log-likelihood scores as follows [38]:

$$\log \hat{p}(\boldsymbol{\varphi}_n|\mathcal{L}_i) = \frac{\alpha t_n}{\beta + t_n} \log p(\boldsymbol{\varphi}_n|\mathcal{L}_i)\,, \qquad (37)$$

where t_n is the speech duration for the n-th utterance and the parameters $\alpha > 0$ and $\beta > 0$ are optimized the usual way using logistic regression on a held-out set.

Another popular method is a discriminative calibration of the scores [39] that also requires optimization of the parameters $\alpha' > 0$, β'_i via logistic regression on a held-out set:

$$\log \hat{p}(\boldsymbol{\varphi}_n|\mathcal{L}_i) = \alpha' \log p(\boldsymbol{\varphi}_n|\mathcal{L}_i) + \beta'_i\,. \qquad (38)$$

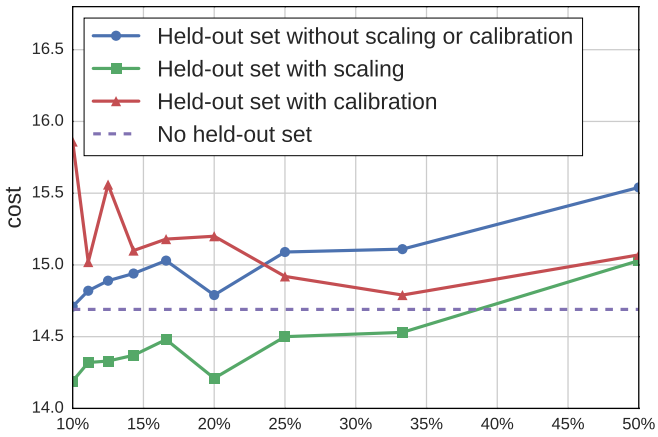

Fig. 4. Evaluation on the LRE'15 tune set of Gaussian classifier, when the training set is split into two, with the held-out set used for duration scaling (37) and calibration (38).

Because a creation of the held-out set splits the training set in two parts, this might lead to a degraded performance. Fig. 4 explores pros and cons of the held-out set, the duration scaling and calibration. It shows that the held-out set equal to 10% of the training set minimally affects the performance (14.71 vs 14.69) while providing sufficient amount of data to train the duration scaling parameters. Calibration is more demanding to the amount of training data and in our experiments has consistently shown worse results compared to the case without held-out set at all. We will exclude calibration from further experiments.

Table III presents the results for both methods and their joint effects for the best PLDA and Gaussian classifiers from the previous subsection. Excluding the short utterances improves the results in all cases and we will use it for our subsequent experiments with the threshold of 15 seconds for LRE'15 and 3 seconds for I2R Dev. A similar finding was reported in [40], where the authors show that not chunking long utterances at all improves the performance for Gaussian classifier. Scaling of the likelihoods works only for the Gaussian classifier and does not benefit neither PLDA nor discriminative classifiers of the next Section. We suspect that it is caused by the necessity of an additional dataset, that reduces the training set. For simple Gaussian classifier at Fig. 4 such effects are minimal, but more advanced classifiers extract more information from the data and, hence, are affected stronger. We will exclude this method from further experiments.

TABLE III
Evaluation on the LRE'15 / I2R Dev tune sets of the best generative classifiers and their corresponding versions with likelihood scaling (see (37)) for the cases when only the utterances longer than t seconds are used for training.

	LDA-Gauss		PLDA	
t	Non-scaled	Scaled	Non-scaled	Scaled
0	15.98 / **7.42**	15.69 / 7.67	16.70 / 7.72	16.99 / 7.74
3	15.74 / 7.44	15.25 / 7.95	16.20 / **7.42**	16.28 / 7.69
10	15.00 / 8.42	14.50 / 8.60	15.11 / 7.58	15.21 / 7.62
15	14.69 / 8.42	**14.19** / 8.60	14.61 / 7.58	**14.58** / 7.62

C. Discriminative models

Optimization. A number of details related to model optimization have to be considered, taking NIST I2R Dev and LRE'15 specificities into account. We modify the basic learning algorithms [21], [10] as follows:

1) To better match the primary cost function (3) — which does not penalize for the between-cluster errors and treats all languages within a cluster equally — we put more emphasis on performing discriminative training separately for each cluster and balance the optimization function by the size of each training class. The latter also helps to compensate for a data imbalance.
2) We consider five variants for discriminative training: three for MMI and two for direct cost optimization. For MMI, we have an optimization done globally (G) with the covariance matrices shared for all languages in a cluster (C), we abbreviate it as G/C, unique for each language (L): (G/L) and the optimization done per cluster with the covariance matrices unique for each language (C/L). Since cost optimization is based on the per cluster scope, we consider just two variants: with the shared covariance matrices (C/C) and without sharing (C/L). For each subsystem, the parameters of discriminative training, namely, τ, λ and the number of iterations, were individually optimized on the tuning set, while the results are presented on an unseen test set.
3) We regularize the model parameters by the standard normal distribution.

D. The role of PLDA subspace

To analyze the effects of PLDA latent subspace, we fix the optimization algorithm to be MMI G/C (we select the best method among MMI for consistency with our preliminary study [8]) and compare the results with classification in LDA space. Fig. 5 presents an extended analysis for LRE'15. We split the evaluation data into 7 duration groups, as specified by NIST, and add the best system of our preliminary study. Table IV presents the corresponding results for I2R Dev.

Both visuals show that while discriminative fine-tuning in LDA space of an already well optimized generative baseline seldom brings noticeable improvements, the same fine-tuning in PLDA latent subspace brings significant improvements. The effects are more prominent for longer test utterances.

TABLE IV
COMPARISON BETWEEN CLASSIFIERS OPERATING IN LDA AND PLDA LATENT SPACES FOR I2R DEV ON **TEST** DATA, GIVEN THE SAME DISCRIMINATIVE OPTIMIZATION METHOD.

	LDA-Gauss		PLDA	
	Generative	MMI	Generative	MMI
Arabic	26.50	25.56	25.49	25.32
Chinese	17.08	17.33	16.48	16.70
English	12.94	12.63	12.65	12.12
Slavic	18.59	19.44	21.11	19.55
Average	18.78	**18.74**	18.93	**18.43**

E. Comparison between MMI and direct cost optimization

Table V shows the performance of the best methods for both MMI and direct cost optimization. We observe the following:

- Performance in PLDA subspace is consistently better than in LDA subspace for all discriminative methods and all language clusters. We observe the relative improvement of 5% for MMI and of 9% for direct cost optimization. While the relative improvement over the baseline in LDA case is 3.7%, it goes up to 10.0% in PLDA case.
- Synergy of PLDA latent subspace and direct cost optimization brings the best results we were able to achieve. It outperforms both MMI optimization in PLDA subspace (column 5) and direct cost optimization in LDA subspace (column 3) for all language clusters except Arabic.

TABLE V
COMPARISON BETWEEN CLASSIFIERS OPERATING IN LDA AND PLDA LATENT SPACES FOR LRE'15 ON **TEST** DATA, FOR THE BEST MMI AND DIRECT COST OPTIMIZERS.

	LDA-Gauss			PLDA		
Discriminative	None	MMI	Cost	None	MMI	Cost
Arabic	20.80	19.91	20.61	19.67	18.75	19.21
Chinese	15.13	15.01	14.92	15.38	13.59	13.15
English	9.80	10.20	9.89	11.70	9.37	8.84
Slavic	4.57	4.46	4.53	4.31	4.25	4.23
Iberian	21.73	19.83	21.79	21.47	20.00	19.81
Average	14.41	**13.88**	14.35	14.50	13.19	**13.05**

VII. CONCLUSION

We have proposed a chain of data transformations that aims at training the classifiers in a discriminative low-dimensional PLDA subspace and propagating the trained parameters back to the original *total variability* space. We have experimented with both generative and discriminative training in that subspace. This approach consistently outperforms training in a standard LDA space for all language clusters. It brings 5% relative improvement for MMI optimization and 9% for direct cost optimization. We have also designed a new objective function, which is an approximation to the primary cost of NIST LRE'15, and have shown how to discriminatively optimize it. The combination of PLDA latent subspace and direct cost optimization led to 9% relative improvement over the best generative system and 10% relative improvement over our best discriminative system reported earlier [8]. We extended our previous study by integrating duration information into the system and parameter tying within language clusters.

REFERENCES

[1] H. Li, B. Ma, and K. A. Lee, "Spoken language recognition: from fundamentals to practice," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1136–1159, 2013.

[2] D. Jurafsky and J. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Prentice Hall, 2000.

[3] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," in *Digital Signal Processing*, 2000.

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

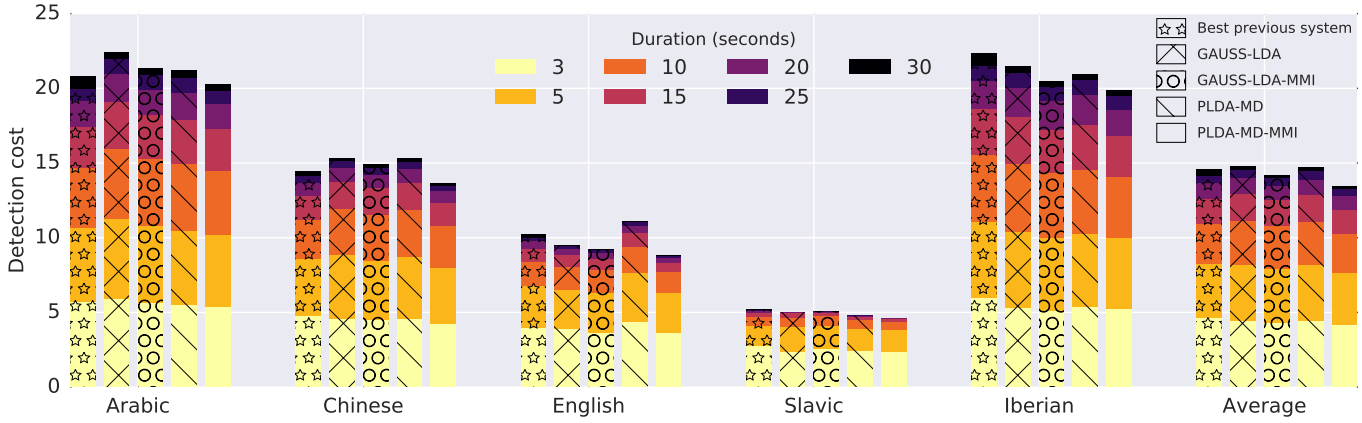

Fig. 5. Evaluation of the detection cost (%) on the LRE'15 test set for 2 best generative classifiers (Gaussian classifier on LDA-projected i-vectors and PLDA classifier with minimum-divergence (MD) estimation of the latent language prior) and their corresponding discriminative versions with respect to 5 language clusters processed separately and all together. "Best previous system" refers to PLDA-MD-MMI system from our previous paper [8].

[5] L. Burget, P. Matějka, and J. Černocký, "Discriminative training techniques for acoustic language identification," in *ICASSP*, 2006.

[6] N. Dehak, P. 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, pp. 788–798, 2011.

[7] S. Prince and J. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *IEEE Int. Conf. on Computer Vision (ICCV)*, 2007.

[8] A. Sizov, K. A. Lee, and T. Kinnunen, "Discriminating languages in a probabilistic latent subspace," in *Odyssey: the Speaker and Language Recognition Workshop*, 2016, pp. 81–88.

[9] D. Martínez, O. Plchot, L. Burget, O. Glembek, and P. Matějka, "Language recognition in iVectors space," in *Interspeech*, 2011.

[10] A. McCree, "Multiclass discriminative training of i-vector language recognition," in *Odyssey: the Speaker and Language Recognition Workshop*, 2014.

[11] N. Brümmer, S. Cumani, O. Glembek, M. Karafiát, P. Matějka, et al., "Description and analysis of the Brno276 system for LRE2011," in *Odyssey: the Speaker and Language Recognition Workshop*, 2012.

[12] E. Singer, P. Torres-Carrasquillo, D. Reynolds, A. McCree, F. Richardson, N. Dehak, and D. Sturim, "The MITLL NIST LRE 2011 language recognition system," in *ICASSP*, 2012.

[13] N. Dehak, P. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via ivectors and dimensionality reduction," in *Interspeech*, 2011.

[14] *The 2011 NIST Language Recognition Evaluation Plan (LRE11)*, http://www.nist.gov/itl/iad/mig/upload/LRE11_EvalPlan_releasev1.pdf.

[15] *The 2015 NIST Language Recognition Evaluation Plan (LRE15)*, http://www.nist.gov/itl/iad/mig/upload/LRE15_EvalPlan_v23.pdf.

[16] H. Li, B. Ma, and C.-H. Lee, "A vector space modeling approach to spoken language identification," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 271–284, 2007.

[17] S. Wright, D. Kanevsky, L. Deng, X. He, G. Heigold, and H. Li, "Optimization algorithms and applications for speech and language processing," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 21, no. 11, pp. 2231–2243, 2013.

[18] G. Heigold, H. Ney, R. Schlüter, and S. Wiesler, "Discriminative training for automatic speech recognition: Modeling, criteria, optimization, implementation, and performance," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 58–69, 2012.

[19] N. Dehak, R. Dehak, J. Glass, D. Reynolds, and P. Kenny, "Cosine similarity scoring without score normalization techniques," in *Odyssey: the Speaker and Language Recognition Workshop*, 2010.

[20] L. Chen, K. A. Lee, B. Ma, W. Guo, H. Li, and L. R. Dai, "Minimum divergence estimation of speaker prior in multi-session PLDA scoring," in *ICASSP*, 2014.

[21] D. Povey, *Discriminative training for large vocabulary speech recognition*, Ph.D. thesis, University of Cambridge, 2005.

[22] D. Zhu, H. Li, B. Ma, and C.-H. Lee, "Optimizing the performance of spoken language recognition with discriminative training," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 16, no. 8, pp. 1642–1653, 2008.

[23] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Odyssey: the Speaker and Language Recognition Workshop*, 2010.

[24] A. Sizov, K. A. Lee, and T. Kinnunen, "Unifying probabilistic linear discriminant analysis variants in biometric authentication," in *S+SSPR*, 2014.

[25] P. Kenny, T. Stafylakis, P. Ouellet, M. J. Alam, and P. Dumouchel, "PLDA for speaker verification with utterances of arbitrary duration," in *ICASSP*, 2013.

[26] S. Theodoridis, *Machine learning: a Bayesian and optimization perspective*, Academic Press, 2015.

[27] C. Bishop, *Pattern Recognition and Machine Learning*, 2006.

[28] L. Van Der Maaten, "Accelerating t-SNE using tree-based algorithms," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3221–3245, 2014.

[29] K. A. Lee, V. Hautamäki, A. Larcher, W. Rao, et al., "Fantastic 4 system for NIST 2015 language recognition evaluation," Tech. Rep., 2015, http://arxiv.org/abs/1602.01929.

[30] K. A. Lee, H. Li, L. Deng, et al., "The 2015 NIST language recognition evaluation: the shared view of I2R, Fantastic4 and SingaMS," in *Interspeech*, 2016.

[31] Y. Song, B. Jiang, Y. Bao, S. Wei, and L.-R. Dai, "I-vector representation based on bottleneck features for language identification," *Electronics Letters*, vol. 49, no. 24, pp. 1569–1570, 2013.

[32] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Processing Letters*, vol. 22, pp. 1671–1675, 2015.

[33] D. Garcia-Romero and C. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Interspeech*, 2011.

[34] A. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," in *Interspeech*, 2006.

[35] T. Stafylakis, P. Kenny, P. Ouellet, J. Perez, M. Kockmann, and P. Dumouchel, "Text-dependent speaker recognition using PLDA with uncertainty propagation," in *Interspeech*, 2013.

[36] P. Torres-Carrasquillo, N. Dehak, E. Godoy, D. Reynolds, F. Richardson, S. Shum, E. Singer, and D. Sturim, "The MITLL NIST LRE 2015 language recognition system," in *Odyssey: the Speaker and Language Recognition Workshop*, 2016.

[37] A. Mccree, G. Sell, and D. Garcia-Romero, "Augmented data training of joint acoustic/phonotactic DNN i-vectors for NIST LRE15," in *Odyssey: the Speaker and Language Recognition Workshop*, 2016.

[38] A. McCree, F. Richardson, E. Singer, and D. Reynolds, "Beyond frame independence: parametric modelling of time duration in speaker and language recognition," in *Interspeech*, 2008, pp. 767–770.

[39] N. Brummer and D. A. Van Leeuwen, "On calibration of language recognition scores," in *Odyssey: the Speaker and Language Recognition Workshop*, 2006, pp. 1–8.

[40] M. Mclaren, D. Castán, and L. Ferrer, "Analyzing the effect of channel mismatch on the SRI language recognition evaluation 2015 system," in *Odyssey: the Speaker and Language Recognition Workshop*, 2016.

Aleksandr Sizov is a Ph.D. student at the Speech and Image Processing Unit at the University of Eastern Finland. He received his Specialist degree in mathematics from the Saint-Petersburg State University, Russia, in 2011, and the M.E degree in computer science from the Saint-Petersburg State University of Information Technologies, Mechanics and Optics, Russia, in 2013. Currently, he is supported by ARAP scholarship from A*STAR, Singapore. His research interests include speaker and language recognition, anti-spoofing and machine learning.

Kong Aik Lee (M05–SM16) received the B.Eng. (first class honors) degree from University Technology Malaysia in 1999, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2006. He is currently a Scientist at the Human Language Technology department, Institute for Infocomm Research (I^2R), A*STAR, Singapore, where he leads the speaker recognition group. He is the recipient of Singapore IES Prestigious Engineering Achievement Award 2013 for his contribution to voice biometrics technology. He serves as an Editorial Board Member/Editor for Elsevier Computer Speech and Language. His current research interests include speaker recognition and characterization, multilingual recognition and identification, speech analysis and processing, machine learning and digital signal processing. He is the leading author of the book Subband Adaptive Filtering: Theory and Implementation (Wiley, 2009).

Associate Professor Tomi Kinnunen received the Ph.D. degree in computer science from the University of Eastern Finland (UEF, formerly Univ. of Joensuu) in 2005. From 2005 to 2007, he was an associate scientist at the Institute for Infocomm Research (I2R) in Singapore. Since 2007, he has been with UEF. In 2010—2012, his research was funded by a post-doctoral grant from Academy of Finland focusing on speaker recognition. He was the PI in a 4-year Academy of Finland project focusing on speaker recognition and a co-PI of another Academy of Finland project focusing on audio-visual spoofing. He chaired *Odyssey 2014: The Speaker and Language Recognition workshop*. He served as an associate editor in *Digital Signal Processing* from 2013 to 2015. He currently serves as an associate editor in *IEEE/ACM Trans. on Audio, Speech and Language Processing* and *Speech Communication*. He is a partner in H2020-funded OCTAVE project focusing on speaker recognition for access control. In 2015–2016 he visited 6 months at National Institute of Informatics (NII), Japan, under a mobility grant from Academy of Finland, with focus on voice conversion, speaker verification and spoofing. He received Docent nomination from Aalto University, Finland, with specialization area in speaker and language recognition. He has authored and co-authored more than 100 peer-reviewed scientific publications in these topics. Since 2017 he is an Associate Professor at UEF.