Discriminating Languages in a Probabilistic Latent Subspace

Aleksandr Sizov[1,2], *Kong Aik Lee*[2], *Tomi Kinnunen*[1]

[1]School of Computing, University of Eastern Finland, Finland

[2]Institute for Infocomm Research, A*STAR, Singapore

sizov@cs.uef.fi, kalee@i2r.a-star.edu.sg, tkinnu@cs.joensuu.fi

Abstract

We explore a method to boost discriminative capabilities of Probabilistic Linear Discriminant Analysis (PLDA) model without losing its generative advantages. To this end, our focus is in a low-dimensional PLDA latent subspace. We optimize the model with respect to MMI (Maximum Mutual Information) and our own objective functions, which is an approximation to the detection cost function. We evaluate the performance on NIST Language Recognition Evaluation 2015. Our model trains faster and performs more accurately in comparison to both generative PLDA and discriminative LDA baselines with 12% and 4% relative improvement in the average detection cost, respectively. The proposed method is applicable for a broad range of closed-set tasks.

1. Introduction

Spoken language recognition is a task to determine the identity of the language spoken in a given speech utterance. It serves to aid general-purpose multilingual speech-based applications, such as spoken language translation [1] and multilingual speech recognition [2]. Spoken Language Recognition Evaluation (LRE) campaigns, regularly conducted by National Institute of Standards and Technology (NIST), are one of the main driving forces advancing language recognition technology. Because of the relatively high recognition accuracy obtained in the basic language detection task [3], the focus has shifted, first, to closely related language pairs in LRE 2011[1] and later to whole clusters of closely related languages and dialects in LRE 2015[2] (see Table 1). We focus on the latter in this study.

Language detection within a language cluster is essentially a closed-set identification task. Discriminative methods are known to be highly effective in such scenarios [4, 5]. Discriminative training takes into account the data from both target and competing classes [6] to optimize the model parameters. Discriminative training criteria, such as maximum mutual information (MMI) and minimum classification error (MCE), have become fundamental tools for speech recognition [7]. Motivated by the success in speech recognition, it was first shown in [8] that the parameters of Gaussian Mixture Model trained with maximum likelihood criterion could be refined using MMI training for language identification. It is generally perceived that discriminative training of generative model increases robustness against the mismatch between the generative model and the real data [9].

Table 1: Language clusters and target languages for NIST LRE'15.

Cluster	Target Languages
Arabic	Egyptian, Iraqi, Levantine, Maghrebi, Modern Standard
Chinese	Cantonese, Mandarin, Min, Wu
English	British, General American, Indian
French	West African, Haitian Creole
Slavic	Russian, Polish
Iberian	Caribbean Spanish, European Spanish, Latin American Spanish, Brazilian Portuguese

In this paper, we review state-of-the-art generative methods, based on the Total Variability (TV) model [10], with the aim to improve their performance with discriminative fine-tuning of each language cluster at a time. The TV approach maps each audio file to a single low-dimensional vector, *i-vector*, that contains speaker, channel, and phonetic variability. This approach has been widely explored recently, especially during LRE 2011 [11], [12], [13]. Usually, i-vectors are modeled by a generative back-end classifier either in the form of a simple Gaussian model [14] or a Probabilistic Linear Discriminant Analysis (PLDA) model [15]. The PLDA model assumes that an i-vector generation process has a certain structure, decomposing into language, channel (environmental effects, an influence of a recording device etc.), and residual noise components. This structure is imposed via component-specific subspaces and their associated latent variables.

Recently, a fairly simple system [16] — consisting of a Gaussian distribution of each language followed by discriminative MMI fine-tuning (or re-training) — yielded promising results on NIST LRE'11 data. In this study, we select this approach as our baseline and analyze how the latent structure of PLDA might be further exploited to improve performance. To validate our hypothesis, we carry out experiments on the latest NIST LRE'15 data. Further, as an alternative to the MMI cost, we propose a new objective function that directly minimizes an approximated version of the primary cost. Like the MMI objective function, it uses the same optimization approach, based on extended Baum-Welch equations [6]. Different from the MMI objective function, however, it takes into account only the misclassified cases. Although the default MMI optimization is reasonable for our application, we expect that our new objective function can lead to a better accuracy and shorter training time.

Figure 1 presents the outline of our system and gives references for the particular Sections.

[1] http://www.nist.gov/itl/iad/mig/upload/LRE11_EvalPlan_releasev1.pdf

[2] http://www.nist.gov/itl/iad/mig/upload/LRE15_EvalPlan_v23.pdf

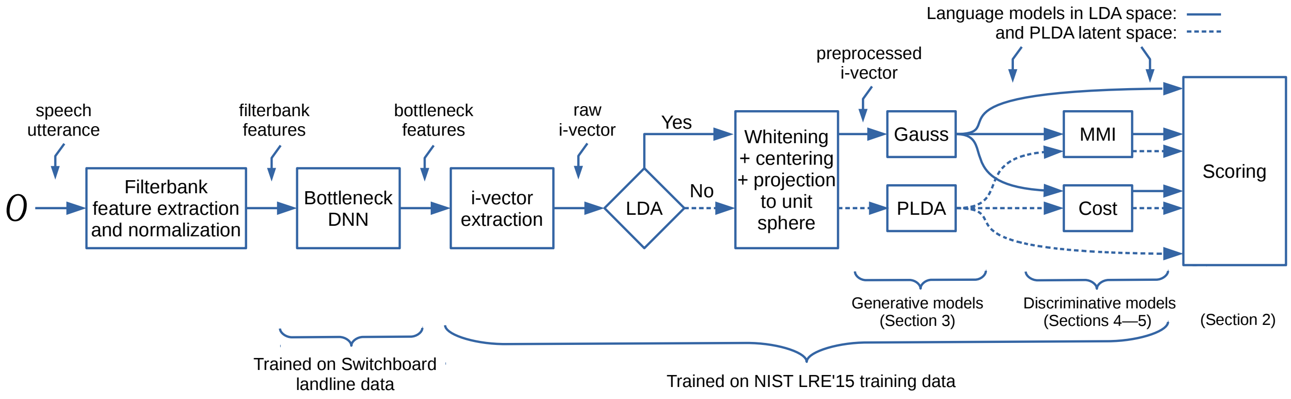

Language models in LDA space:
and PLDA latent space:
speech utterance
filterbank features
bottleneck features
raw i-vector
preprocessed i-vector
O
Filterbank feature extraction and normalization
Bottleneck DNN
i-vector extraction
LDA
Yes
No
Whitening + centering + projection to unit sphere
Gauss
PLDA
MMI
Cost
Scoring
Generative models (Section 3)
Discriminative models (Sections 4—5)
(Section 2)
Trained on Switchboard landline data
Trained on NIST LRE'15 training data

Figure 1: Block diagram of our system.

2. Language detection using i-vectors

Let K be the total number of languages, $N = N_1 + \ldots + N_K$ be the total number of i-vectors, and $\Phi_i = \{\boldsymbol{\varphi}_n\}_{n=1}^{N_i}$ be a collection of all i-vectors of language $\mathcal{L}_i$. Considering a more general case of language detection task, where closely related languages are grouped into clusters, we introduce $\{\mathcal{C}_k\}_{k=1}^{I}$ as a collection of language clusters. Each cluster contains the indices of languages that belong to it. For the specific case of LRE'15, we have $I = 6$ clusters and each cluster consists of a number of languages as shown in Table 1. For $I = 1$, the formulation reduces to the ordinary language detection task.

The primary task of language detection is to minimize the average detection cost, C^k_{avg}, which has the following form for each cluster $\mathcal{C}_k$:

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

where $\text{FRR}(\bullet)$ and $\text{FAR}(\bullet)$ are the false rejection and false acceptance rates, respectively. To compute them we first evaluate log-likelihood ratio (llr) score for each language $\mathcal{L}_i$ ($i \in \mathcal{C}_k$) and each i-vector $\boldsymbol{\varphi}_n$:

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

then we compare them against a threshold of 0 and apply an indicator function $I(\bullet)$ in a following way:

$$\text{FRR}(\mathcal{L}_i) = \frac{1}{N_i}\sum_{\boldsymbol{\varphi}_n\in\Phi_i} I\{\text{llr}(\mathcal{L}_i|\boldsymbol{\varphi}_n) < 0\}, \tag{3}$$

$$\text{FAR}(\mathcal{L}_i,\mathcal{L}_j) = \frac{1}{N_i}\sum_{\boldsymbol{\varphi}_n\in\Phi_i} I\{\text{llr}(\mathcal{L}_j|\boldsymbol{\varphi}_n) \geqslant 0\}. \tag{4}$$

Notice that in (1), the costs for making both types of errors are assumed equal.

3. Generative Gaussian back-end

The most important part is to reliably estimate $p(\boldsymbol{\varphi}|\mathcal{L}.)$. In this Section, we present two generative approaches to address this problem.

3.1. Gaussian modeling with smoothing

For our baseline system, we use a simple yet effective Gaussian classifier [14]: each language $\mathcal{L}_i$ is represented by a sample mean vector

$$\boldsymbol{\mu}_i = \frac{1}{N_i}\sum_{j=1}^{N_i}\boldsymbol{\varphi}_{ij}\,, \tag{5}$$

and a covariance matrix $\boldsymbol{\Sigma}_i$, which is a weighted sum of global sample covariance matrix and language-specific sample covariance matrix:

$$\begin{aligned}\boldsymbol{\Sigma}_i = {} & \frac{\alpha}{N}\sum_{i=1}^{K}\sum_{j=1}^{N_i}(\boldsymbol{\varphi}_{ij}-\boldsymbol{\mu}_i)(\boldsymbol{\varphi}_{ij}-\boldsymbol{\mu}_i)^T \\ & + \frac{1-\alpha}{N_i}\sum_{j=1}^{N_i}(\boldsymbol{\varphi}_{ij}-\boldsymbol{\mu}_i)(\boldsymbol{\varphi}_{ij}-\boldsymbol{\mu}_i)^T\,.\end{aligned} \tag{6}$$

We optimized the model for the weight parameter α with a simple grid search. See results in Table **??**. Given a test i-vector $\boldsymbol{\varphi}_t$, we compute its detection score by (2), where $p(\boldsymbol{\varphi}_t|\mathcal{L}_i)$ is taken as a normal distribution $\mathcal{N}(\boldsymbol{\varphi}_t|\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)$ with mean $\boldsymbol{\mu}_i$ and covariance $\boldsymbol{\Sigma}_i$ as given above.

3.2. Structured Gaussian modeling with Probabilistic LDA

Another popular model that splits apart channel and language variability is the so-called Probabilistic Linear Discriminant Analysis (PLDA) model [15]. We use its modification, *simplified* PLDA [17], where channel and noise variabilities are bound together and modeled by a full covariance matrix $\boldsymbol{\Lambda}^{-1}$ (for more details and comparison of PLDA variants, the reader may refer to [18]). In particular, we assume the following form:

$$\begin{aligned}\boldsymbol{\varphi}_{ij} &= \boldsymbol{\mu} + \mathbf{V}\mathbf{y}_i + \boldsymbol{\varepsilon}_{ij}\,,\\ \mathbf{y}_i &\sim \mathcal{N}(\mathbf{y}_i|\mathbf{0},\mathbf{I})\,,\\ \boldsymbol{\varepsilon}_{ij} &\sim \mathcal{N}\left(\boldsymbol{\varepsilon}_{ij}\middle|\mathbf{0},\boldsymbol{\Lambda}^{-1}\right)\,,\end{aligned}$$

where, $\boldsymbol{\mu}$ is the global mean vector, $\mathbf{V}$ is the factor loading matrix, $\mathbf{y}_i$ is the language latent variable, and $\boldsymbol{\varepsilon}_{ij}$ is the residual noise. The effective number of dimensions for $\mathbf{y}_i$ cannot exceed the total number of languages, K, because of the properties of the expectation-maximization algorithm used to train the model. To utilize as much information about the language variability as possible, we set it to $K = 20$ in this study.

Given the set Φ_i of training i-vectors for the i-th language and a test i-vector $\boldsymbol{\varphi}_t$, the standard procedure to compute PLDA detection scores is to evaluate the log-likelihood ratio between two hypotheses. The null hypothesis dictates that Φ_i and $\boldsymbol{\varphi}_t$ belong to the same language; the alternative hypothesis dictates that Φ_i and $\boldsymbol{\varphi}_t$ belong to different languages:

$$\text{score}(\Phi_i, \boldsymbol{\varphi}_t) = \log \frac{p(\Phi_i, \boldsymbol{\varphi}_t)}{p(\Phi_i)p(\boldsymbol{\varphi}_t)}. \tag{7}$$

It is computationally more efficient [19, 20] to view this as a log ratio between language-specific and default PLDA models:

$$\text{score}(\Phi_i, \boldsymbol{\varphi}_t) = \log \frac{p(\boldsymbol{\varphi}_t|\Phi_i)}{p(\boldsymbol{\varphi}_t)} \,,$$

where

$$\begin{aligned} p(\boldsymbol{\varphi}_t) &= \int \mathcal{N}\left(\boldsymbol{\varphi}_t \middle| \boldsymbol{\mu} + \mathbf{V}\mathbf{y}_i, \boldsymbol{\Lambda}^{-1}\right) \mathcal{N}\left(\mathbf{y}_i|\mathbf{0}, \mathbf{I}\right) \mathbf{dy}_i \\ &= \mathcal{N}\left(\boldsymbol{\varphi}_t \middle| \boldsymbol{\mu}, \mathbf{V}\mathbf{V}^T + \boldsymbol{\Lambda}^{-1}\right) \\ p(\boldsymbol{\varphi}_t|\Phi_i) &= \int \mathcal{N}\left(\boldsymbol{\varphi}_t \middle| \boldsymbol{\mu} + \mathbf{V}\mathbf{y}_i, \boldsymbol{\Lambda}^{-1}\right) \mathcal{N}\left(\mathbf{y}_i|\mathbf{m}_i, \mathbf{G}_i\right) \mathbf{dy}_i \\ &= \mathcal{N}\left(\boldsymbol{\varphi}_t \middle| \boldsymbol{\mu} + \mathbf{V}\mathbf{m}_i, \mathbf{V}\mathbf{G}_i^{-1}\mathbf{V}^T + \boldsymbol{\Lambda}^{-1}\right). \end{aligned} \tag{8}$$

Equation (8) involves two language-specific terms, $\mathbf{m}_i$ and $\mathbf{G}_i^{-1}$. We consider three different approaches to evaluate them.

By-the-book scoring. The first approach, popularly known as *by-the-book* scoring [15], treats all the i-vectors in Φ_i as independent observations, resulting in the following equations

$$\mathbf{m}_i = \mathbf{G}_i^{-1}\mathbf{V}^T\boldsymbol{\Lambda}\sum_{j=1}^{N_i}(\boldsymbol{\varphi}_{ij} - \boldsymbol{\mu}) \,, \tag{9}$$

$$\mathbf{G}_i^{-1} = (\mathbf{I} + N_i\mathbf{V}^T\boldsymbol{\Lambda}\mathbf{V})^{-1} \,. \tag{10}$$

Although this approach gives us a proper posterior estimate, it is undesirable in our case for two reasons. First, its underlying assumption that the i-vectors are conditionally independent of each other given a language label is unreasonable from both speech production and i-vector generation process viewpoints. Second, since for each individual language we might have thousands of i-vectors, it will result in a highly peaked distribution in a latent subspace.

I-vector averaging. The second and the most popular approach involves simply averaging all the i-vectors in Φ_i, so that new N_i effectively becomes 1 and $\{\mathbf{m}_i, \mathbf{G}_i^{-1}\}$ have the following form:

$$\mathbf{m}_i = \mathbf{G}_i^{-1}\mathbf{V}^T\boldsymbol{\Lambda}\left(\frac{1}{N_i}\sum_{j=1}^{N_i}\boldsymbol{\varphi}_{ij} - \boldsymbol{\mu}\right) \,, \tag{11}$$

$$\mathbf{G}_i^{-1} = (\mathbf{I} + \mathbf{V}^T\boldsymbol{\Lambda}\mathbf{V})^{-1} \,. \tag{12}$$

One of the reasons for the success of this method is that the posterior precision matrix $\mathbf{G}_i$ is independent of any class-specific data and, hence, is more robust for classes with small amounts of data. In this study, we have the opposite situation with relatively small amount of classes (20) and large amount of data per class.

Minimum divergence estimation. That is why the third and the last approach we consider is based on so-called minimum divergence (MD) estimation of language-adapted priors [20], allowing us to have more degrees of freedom. To estimate posterior language latent variable distribution $\mathcal{N}\left(\mathbf{y}_i|\mathbf{m}_i, \mathbf{G}_i\right)$ with the MD-approach, we first project each i-vector onto this latent subspace:

$$\boldsymbol{\phi}_{ij} = (\mathbf{V}^T\boldsymbol{\Lambda}\mathbf{V} + \mathbf{I})^{-1}\mathbf{V}^T\boldsymbol{\Lambda}(\boldsymbol{\varphi}_{ij} - \boldsymbol{\mu}) \,. \tag{13}$$

Notice that we use $\boldsymbol{\varphi}_{ij}$ and $\boldsymbol{\phi}_{ij}$ to denote the i-vector in the original total variability space and the latent subspace, respectively. Then we find the parameters $\mathbf{m}_i$ and $\mathbf{G}_i^{-1}$ as

$$\mathbf{m}_i = \frac{1}{N_i}\sum_{j=1}^{N_i}\boldsymbol{\phi}_{ij} \,, \tag{14}$$

$$\mathbf{H}_i = \frac{1}{N_i}\sum_{j=1}^{N_i}(\boldsymbol{\phi}_{ij} - \mathbf{m}_i)(\boldsymbol{\phi}_{ij} - \mathbf{m}_i)^T \tag{15}$$

$$\mathbf{G}_i^{-1} = (\mathbf{I} + \mathbf{V}^T\boldsymbol{\Lambda}\mathbf{V})^{-1} + \mathbf{H}_i \,. \tag{16}$$

The difference from the previous case comes from the matrix $\mathbf{H}_i$, which is an empirical covariance estimate in the latent subspace.

4. MMI fine-tuning

Maximum mutual information (MMI) training [6] aims at increasing the discrimination abilities of a classifier, by maximizing the posterior probability of correct class given training data:

$$\mathcal{Q}_{\text{MMI}} = \sum_{i\in\mathcal{C}}\sum_{\boldsymbol{\varphi}_n\in\Phi_i}\log p(\mathcal{L}_i|\boldsymbol{\varphi}_n) = \sum_{i\in\mathcal{C}}\sum_{\boldsymbol{\varphi}_n\in\Phi_i}\log\frac{p(\boldsymbol{\varphi}_n|\mathcal{L}_i)}{\sum\limits_{j\in\mathcal{C}}p(\boldsymbol{\varphi}_n|\mathcal{L}_j)} \,, \tag{17}$$

where $\mathcal{C}$ is a set of language (or class) indices. Depending on the application, MMI optimization could be performed within or across language clusters. In (17) we assume that all languages have an equal prior probability.

A number of details have to be considered in optimizing the MMI cost depending on the application. Taking NIST LRE'15 specificities into account, we modify the basic MMI algorithm [6, 16] in several ways:

1. Since the primary cost function (1) is applied for each language cluster at a time and it does not penalize for between-cluster errors, we perform MMI fine-tuning separately for each cluster.

2. Since each cluster has a relatively small number of languages (in the range of two to five), we include regularization in the form of a prior distribution. To this end, we consider using either an initial ML estimate of each class or a standard normal distribution. The latter yielded better results and has an interesting interpretation under one of our classifiers, so we will not consider the former in this study.

3. Because of a severe data mismatch and noting that the primary cost function (1) treats all languages within a cluster with equal weights, we apply a *balanced* version of MMI algorithm, where the weight of each i-vector is inversely proportional to the total number of the i-vectors in its class, making it $1/N_{\cdot}$.

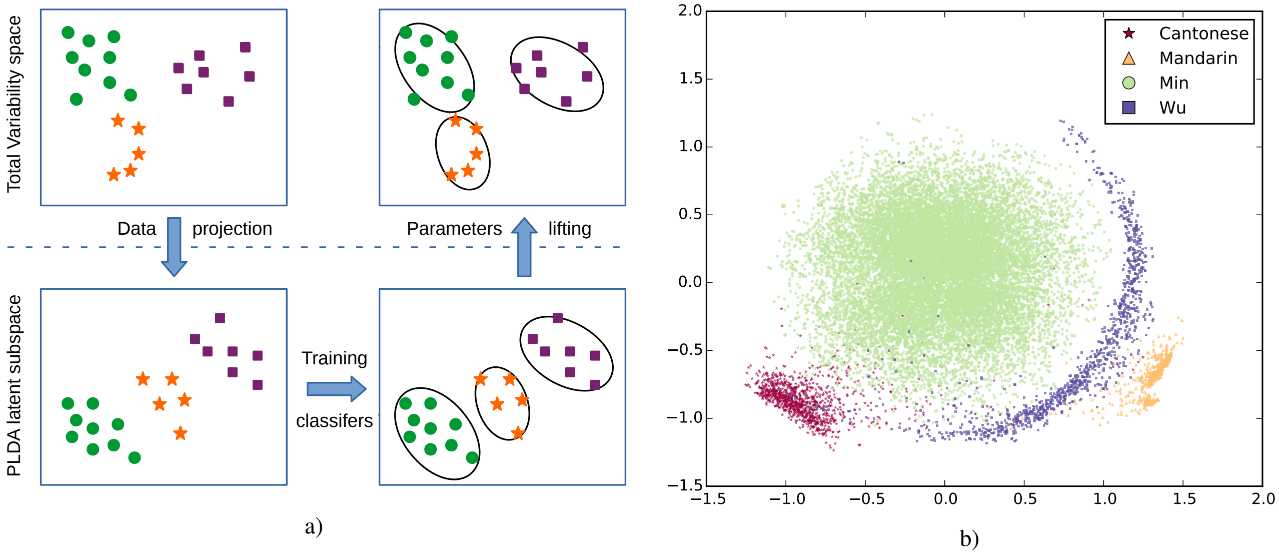


Figure 2: a) A schematic illustration of a discriminative training in a Probabilistic LDA latent subspace. Each point corresponds to an i-vector and each color to a language. b) Two-dimensional projection of Chinese language cluster training set. Each point corresponds to an i-vector in a 20-dimensional Probabilistic LDA latent subspace. We used an accelerated version of t-SNE algorithm [21] to produce the picture. Both pictures are best viewed in color.

4.1. MMI in Total Variability space

In this scenario, we consider Gaussian distribution with parameters from (5)-(6) and apply MMI [16] training on all distributions within a single language cluster at a time. Thus, the balanced objective function $\mathcal{Q}_{\mathrm{MMI}}$ (17) has the following form:

$$\mathcal{Q}^k_{\mathrm{MMI}} = \sum_{i\in\mathcal{C}_k} \frac{1}{N_i} \sum_{\boldsymbol{\varphi}_n\in\Phi_i} \log \frac{\mathcal{N}\left(\boldsymbol{\varphi}_n|\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i\right)}{\sum\limits_{j\in\mathcal{C}_k}\mathcal{N}\left(\boldsymbol{\varphi}_n|\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j\right)}\,. \tag{18}$$

At each iteration, we maximize (18) via the Extended Baum-Welch equations [6] which involve computation of the following sufficient statistics per language:

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

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

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

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. Further, to prevent over-fitting, we regularize our equations with τ data points that has 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.

$$s^0_i \leftarrow s^0_i + \lambda + \tau\,, \tag{19}$$

$$\mathbf{s}^1_i \leftarrow \mathbf{s}^1_i + \lambda\boldsymbol{\mu}_i\,, \tag{20}$$

$$\mathbf{S}^2_i \leftarrow \mathbf{S}^2_i + \lambda(\boldsymbol{\mu}_i\boldsymbol{\mu}_i^T + \boldsymbol{\Sigma}_i) + \tau\mathbf{I}\,. \tag{21}$$

Afterwards, the parameter updates are as follows

$$\boldsymbol{\mu}_i = \frac{\mathbf{s}^1_i}{s^0_i}\,, \tag{22}$$

$$\boldsymbol{\Sigma}_i = \frac{\mathbf{S}^2_i - \boldsymbol{\mu}_i{\mathbf{s}^1_i}^T - \mathbf{s}^1_i\boldsymbol{\mu}_i^T + s^0_i\boldsymbol{\mu}_i\boldsymbol{\mu}_i^T}{s^0_i}\,. \tag{23}$$

4.2. MMI in Probabilistic LDA latent space

Another way to apply MMI fine-tuning is to take the already trained language-specific PLDA distribution $p(\boldsymbol{\varphi}_t|\Phi_i)$ from (8) as a class-conditional distribution $p(\boldsymbol{\varphi}_t|\mathcal{L}_i)$. If we optimize full class mean vectors and class covariance matrices, 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:

- **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 (13).
- **Minimum divergence estimation**: estimate the mean vector $\mathbf{m}_i$ and covariance matrix $\mathbf{G}_i^{-1}$ using (14)–(16).
- **MMI fine tuning**: retrain the mean vectors and covariance matrices to optimize for better separation between classes. In particular, the MMI objective function becomes

$$\mathcal{Q}^k_{\mathrm{MMI}} = \sum_{i\in\mathcal{C}_k}\frac{1}{N_i}\sum_{\boldsymbol{\phi}_n:\,\boldsymbol{\varphi}_n\in\Phi_i}\log\frac{\mathcal{N}\left(\boldsymbol{\phi}_n|\mathbf{m}_i,\mathbf{G}_i^{-1}\right)}{\sum\limits_{j\in\mathcal{C}_k}\mathcal{N}\left(\boldsymbol{\phi}_n|\mathbf{m}_j,\mathbf{G}_j^{-1}\right)}.$$

  Notice that the inputs are now the projected i-vectors $\boldsymbol{\phi}_n$. The update equation have to be changed accordingly.
- **Parameter lifting**: Lift the mean vectors and covariance matrices from the latent space back to the i-vector space.

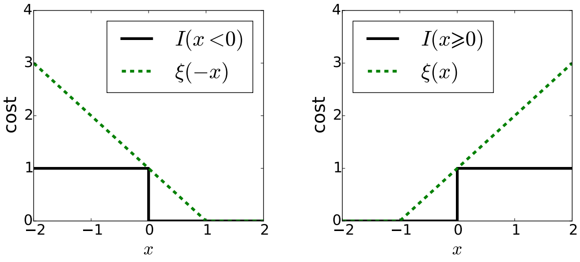


Figure 4: Hinge loss approximation for a) false rejection and b) false acceptance errors as defined in (3) and (4) respectively.

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.

5. Cost fine-tuning

We notice that the posterior probabilities used in MMI objective function (17) have essentially the same structure as log-likelihood ratios (2) used in cluster cost function. This makes it possible to apply MMI-like optimization to *directly* maximize the primary cost function (1). The main difficulty in doing so is that (1) relies on 0-1 classification loss that is discrete and non-differentiable. To address this issue, we approximate 0-1 loss in (3)–(4) with its continuous convex upper-bound in a form of a hinge function [22] (see Fig. 4) as follows:

$$I(x<0) \approx \xi(-x), \tag{24}$$

$$I(x \geqslant 0) \approx \xi(x)\,, \tag{25}$$

where $\xi(x) = \max(x+1, 0)$. Thus, the approximated cost function for cluster $\mathcal{C}_k$ takes the form

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

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

$$\Psi_{ij} = \left\{ \boldsymbol{\varphi}_n \in \Phi_i : \begin{cases} \text{llr}(\mathcal{L}_j|\boldsymbol{\varphi}_n) < 1, & i=j \\ \text{llr}(\mathcal{L}_j|\boldsymbol{\varphi}_n) > -1, & i\neq j \end{cases} \right\}. \tag{26}$$

This leads us to a revised objective function:

$$\mathcal{Q}^k_{\text{cost}} = \sum_{i\in\mathcal{C}_k} \frac{1}{N_i} \left\{ \sum_{\boldsymbol{\varphi}_n\in\Psi_{ii}} \text{llr}(\mathcal{L}_i|\boldsymbol{\varphi}_n) - \frac{1}{|\mathcal{C}_k|-1} \sum_{\substack{j\in\mathcal{C}_k\\ j\neq i}} \sum_{\boldsymbol{\varphi}_n\in\Psi_{ij}} \text{llr}(\mathcal{L}_j|\boldsymbol{\varphi}_n) \right\}. \tag{27}$$

As in the case of MMI, we apply iterative training with the help of Extended Baum-Welch equations [6] to maximize (27). The sufficient statistics of each language are computed as follows:

$$s^0_i = \frac{|\mathcal{C}_k|-1}{N_i} \sum_{\boldsymbol{\varphi}_n\in\Psi_{ii}} 1 + \sum_{j\in\mathcal{C}_k} \frac{1}{N_j} \sum_{\substack{h\in\mathcal{C}_k\\ h\neq j\\ h\neq i}} \sum_{\boldsymbol{\varphi}_n\in\Psi_{jh}} \gamma^{-h}_{ni} - \sum_{\substack{j\in\mathcal{C}_k\\ j\neq i}} \frac{1}{N_j} \left[(|\mathcal{C}_k|-1) \sum_{\boldsymbol{\varphi}_n\in\Psi_{jj}} \gamma^{-j}_{ni} + \sum_{\boldsymbol{\varphi}_n\in\Psi_{ji}} 1 \right], \tag{28}$$

$$\mathbf{s}^1_i = \frac{|\mathcal{C}_k|-1}{N_i} \sum_{\boldsymbol{\varphi}_n\in\Psi_{ii}} \boldsymbol{\varphi}_n + \sum_{j\in\mathcal{C}_k} \frac{1}{N_j} \sum_{\substack{h\in\mathcal{C}_k\\ h\neq j\\ h\neq i}} \sum_{\boldsymbol{\varphi}_n\in\Psi_{jh}} \gamma^{-h}_{ni} \boldsymbol{\varphi}_n - \sum_{\substack{j\in\mathcal{C}_k\\ j\neq i}} \frac{1}{N_j} \left[(|\mathcal{C}_k|-1) \sum_{\boldsymbol{\varphi}_n\in\Psi_{jj}} \gamma^{-j}_{ni} \boldsymbol{\varphi}_n + \sum_{\boldsymbol{\varphi}_n\in\Psi_{ji}} \boldsymbol{\varphi}_n \right], \tag{29}$$

$$\mathbf{S}^2_i = \frac{|\mathcal{C}_k|-1}{N_i} \sum_{\boldsymbol{\varphi}_n\in\Psi_{ii}} \boldsymbol{\varphi}_n \boldsymbol{\varphi}_n^T + \sum_{j\in\mathcal{C}_k} \frac{1}{N_j} \sum_{\substack{h\in\mathcal{C}_k\\ h\neq j\\ h\neq i}} \sum_{\boldsymbol{\varphi}_n\in\Psi_{jh}} \gamma^{-h}_{ni} \boldsymbol{\varphi}_n \boldsymbol{\varphi}_n^T - \sum_{\substack{j\in\mathcal{C}_k\\ j\neq i}} \frac{1}{N_j} \left[(|\mathcal{C}_k|-1) \sum_{\boldsymbol{\varphi}_n\in\Psi_{jj}} \gamma^{-j}_{ni} \boldsymbol{\varphi}_n \boldsymbol{\varphi}_n^T + \sum_{\boldsymbol{\varphi}_n\in\Psi_{ji}} \boldsymbol{\varphi}_n \boldsymbol{\varphi}_n^T \right], \tag{30}$$

where

$$\gamma^{-j}_{ni} = \frac{p(\boldsymbol{\varphi}_n|\mathcal{L}_j)}{\sum\limits_{\substack{h\in\mathcal{C}_k\\ h\neq i}} p(\boldsymbol{\varphi}_n|\mathcal{L}_h)}\,. \tag{31}$$

Afterwards, we smooth and regularize the sufficient statistics (19)–(21) and update the parameters (22)–(23) the same way as in MMI training.

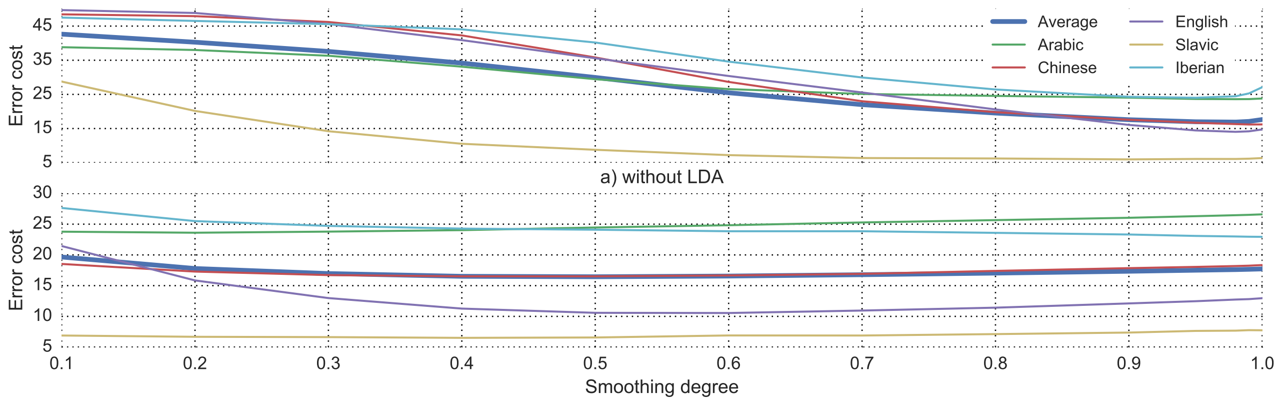
We consider two different representations for $p(\boldsymbol{\varphi}_n|\mathcal{L}_i)$: one that comes from the Gaussian distribution with the parameters from (5)–(6) and the other that comes from the language-specific PLDA model in (8).

6. Experiments

I-vectors for this study were prepared by “Fantastic 4” team [23] 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 [24, 25]. The DNN was trained on the switchboard landline 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 [23] for more details (this set of i-vectors is abbreviated as BNF2 in that paper).

We whitened all i-vectors on the training data and then projected them to the unit sphere [26]. Within Class Covariance Normalization [27] (WCCN) was not found helpful. LDA (Linear Discriminant Analysis) transformation on the raw i-vectors was of some help and we evaluate its results in the experiments. We set LDA subspace to 20 dimensions to make it equal and comparable to the PLDA latent language subspace.

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



a) without LDA

b) with LDA

Figure 3: Evaluation of the $\mathcal{C}^{\bullet}_{\text{avg}}$ — cost error (%) for each language cluster — for Gaussian classifier (from subsection 3.1) with varied degree of smoothing, α, on the tune set. LDA is set to 20 dimensions. The average of $\mathcal{C}^{\bullet}_{\text{avg}}$ for all clusters corresponds to the primary metric of NIST LRE'15 and this paper. We omit the results for the case when $\alpha = 0$, because a few languages do not have sufficient amount of i-vectors to estimate full rank covariance matrices without smoothing.

Table 2: Evaluation of the cost performance (%) for simplified PLDA classifier on the tune set. The results comprise by-the-book (book) (9) – (10), averaged (avg) i-vector (11) – (12), and minimum-divergence (min-div) scoring (14) – (16).

	No LDA			LDA		
	book	avg	min-div	book	avg	min-div
Arabic	23.80	24.39	23.92	26.66	26.90	26.68
Chinese	16.81	17.91	16.81	18.36	20.54	20.12
English	14.33	12.88	12.71	13.81	14.89	14.79
Slavic	6.36	6.52	6.26	7.74	7.38	7.45
Iberian	27.02	26.42	24.18	23.04	24.58	25.17
Average	17.55	17.62	**16.78**	17.92	18.86	18.84

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.

6.1. Baseline generative models

First, we set up our baseline generative classifiers. Figure 3 presents the evaluation of the Gaussian back-end with smoothing, described in Subsection 3.1. LDA prior to other preprocessing techniques improves not only average cost performance, 16.45% versus 16.90%, but increases stability as well: we observe a flat plateau for smoothing coefficient $\alpha \in [0.4, 0.6]$. Further, as discarding LDA leads to considerably slower discriminative stage, we only consider LDA processing for the remaining experiments on the Gaussian classifier. Without LDA, Gaussian classifiers were trained on 600-dimensional i-vectors as opposed to 20 in the case with LDA.

Table 2 presents the evaluation of a simplified PLDA classifier, described in Subsection 3.2. It was trained on all the 20 languages at once. The results indicate that 1) LDA is not only redundant but also detrimental for the PLDA classifier; we will not consider it for the following experiments with PLDA, 2) Addition of empirical covariance matrix to the language-specific PLDA models during minimum-divergence scoring consistently improves accuracy for all the language clusters. This can be seen by comparing the results between avg and min-div. The only difference between these two methods is in the covariance matrix, where the second term in (16) is absent from the former.

Table 3: Evaluation of the cost performance (%) for Gaussian classifier with and without discriminative fine-tuning.

		Baseline	MMI		Cost	
Regularization		no	no	yes	no	yes
Tune set	Arabic	24.46	24.26	23.68	23.78	23.35
	Chinese	16.52	16.88	15.65	17.97	16.14
	English	10.58	10.66	10.29	10.97	10.37
	Slavic	6.57	6.68	6.36	6.75	6.47
	Iberian	24.10	24.08	21.87	24.44	22.09
	Average	16.45	16.51	**15.57**	16.78	15.68
Test set	Arabic	22.78	22.56	21.90	22.20	21.77
	Chinese	16.30	16.65	15.49	17.67	16.17
	English	11.00	11.30	10.55	11.94	11.00
	Slavic	5.63	5.61	5.30	5.40	5.53
	Iberian	24.48	24.46	22.24	24.70	22.10
	Average	16.04	16.11	**15.10**	16.38	15.31

6.2. Discriminative models

For each subsystem, the parameters of discriminative training, namely, τ, λ and the number of iterations, were individually optimized on the tuning set. Table 3 presents a comparison of two discriminative approaches for Gaussian classifier. We observe the following:

1. Mismatch (mainly due to channel) between development and evaluation data is so huge that MMI algorithm without regularization fails to improve over the baseline on both tune and test sets, even if the errors on the training set (not shown here) reduced substantially.
2. Regularization in the form of a standard normal distribution consistently improves over the baseline. The impact is more profound for the MMI cost function.

Table 4: Evaluation of the cost performance (%) for simplified PLDA classifier with and without discriminative fine-tuning. The results comprise averaged (avg) i-vector scoring (11) – (12) and minimum-divergence (min-div) scoring (14) – (16). The average of $\mathcal{C}^{\bullet}_{\text{avg}}$ for all clusters (shaded row) corresponds to the primary metric of NIST LRE'15 and this paper.

		Baseline		MMI				Cost			
		avg	min-div	avg		min-div		avg		min-div	
Regularization		no	no	no	yes	no	yes	no	yes	no	yes
Tune set	Arabic	24.39	23.92	24.31	22.17	23.94	22.12	23.97	22.05	23.75	21.97
	Chinese	17.91	16.81	17.77	14.56	16.83	14.56	17.74	14.80	16.95	14.87
	English	12.88	12.71	12.57	8.86	12.78	10.00	12.72	8.64	12.97	9.69
	Slavic	6.52	6.26	6.43	6.33	6.09	6.19	6.52	6.27	6.17	6.26
	Iberian	26.42	24.18	24.81	22.81	24.08	22.50	24.44	22.73	24.06	22.63
	Average	17.62	16.78	17.18	14.95	16.74	15.07	17.08	**14.90**	16.78	15.08
Test set	Arabic	23.00	22.41	22.84	20.75	22.38	20.69	22.49	20.53	22.14	20.45
	Chinese	17.81	16.78	17.69	14.34	16.79	14.38	17.73	14.64	16.92	14.77
	English	13.24	13.47	13.26	9.44	13.56	10.16	13.54	9.30	13.87	10.06
	Slavic	5.37	5.28	5.33	5.30	5.03	5.12	5.21	5.29	4.85	5.15
	Iberian	27.05	24.72	25.07	23.05	24.50	22.27	24.39	22.88	24.25	22.38
	Average	17.30	16.53	16.84	14.58	16.45	**14.53**	16.67	**14.53**	16.41	14.56

Table 4 extends the results of Table 3 to the case of PLDA classification in a latent subspace (by-the-book scoring is not used for PLDA model because the covariance matrices in latent subspace are too small and non-invertible). We observe the following:

1. Poor performance of discriminative methods without regularization extends to this case as well.
2. Discriminative fine-tuning with regularization effectively eliminates the differences between different PLDA scoring variants. This is a good sign in the sense that the fine-tuning is not very sensitive to the initial point.
3. On the test set, the results of both discriminative approaches are very close to each other regardless of initialization. It took 5 MMI optimization iterations to achieve the optimal performance; direct cost optimization achieved the same total error in just 2 iterations.

Finally, we analyze the duration effects. To this end, we split the test data into 7 duration groups, as specified by NIST. Figure 5 compares the 2 best generative methods and their corresponding discriminative counterparts (as reported in Tables 3 and 4), given the same optimization method, MMI. The lower levels of the bar chart are quite similar for all the systems, the differences being mostly determined by the mid and top levels.

Table 5 further compares both discriminative systems against the Gaussian back-end for each duration category. For short utterances, the gain is minor but increases with increasing test utterance duration, reaching up to 20% for PLDA system with MMI updates in the latent subspace. Since the test data is biased towards short utterances, the net effects of the discriminative systems are not so prominent.

7. Conclusion

We have demonstrated how to effectively apply a discriminative fine-tuning of a PLDA model—for a closed-set language identification task—in a low-dimensional PLDA latent subspace and then lift the parameters back to the Total Variability space. This operation improved the results by 9.4% and 3.8% relative to our best generative and discriminative baselines for all utterances and by 20% and 9% for long utterances, respectively, with the added benefit of accelerated convergence speed. Also, we have developed a new objective function for discriminative training to better match primary cost function of NIST LRE'15. Its top performance on our test set is very close to MMI, which might indicate that we have reached the limits of Gaussianity assumption in this context. On the bright side, fine-tuning with the new cost function is considerably faster than MMI: it takes only 50 iterations instead of 500 in the LDA space and 2 iterations instead of 5 in the PLDA latent space.

8. Acknowledgments

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

9. References

[1] Alex Waibel and Christian Fugen, "Spoken language translation," *IEEE Signal Processing Magazine*, vol. 3, no. 25, pp. 70–79, 2008.

[2] Alex Waibel, Hagen Soltau, Tanja Schultz, Thomas Schaaf, and Florian Metze, "Multilingual speech recognition," in *Verbmobil: Foundations of Speech-to-Speech Translation*, pp. 33–45. Springer, 2000.

[3] Alvin Martin, Craig Greenberg, John Howard, George Doddington, and John Godfrey, "NIST language recognition evaluation – past and future," in *Odyssey: the Speaker and Language Recognition Workshop*, 2014.

[4] Haizhou Li, Bin Ma, and Kong Aik Lee, "Spoken language recognition: from fundamentals to practice," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1136–1159, 2013.

[5] Bin Ma, Rong Tong, and Haizhou Li, "Discriminative vector for spoken language recognition," in *ICASSP*, 2007.

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

[7] Georg Heigold, Hermann Ney, Ralf Schlüter, and Simon Wiesler, "Discriminative training for automatic speech

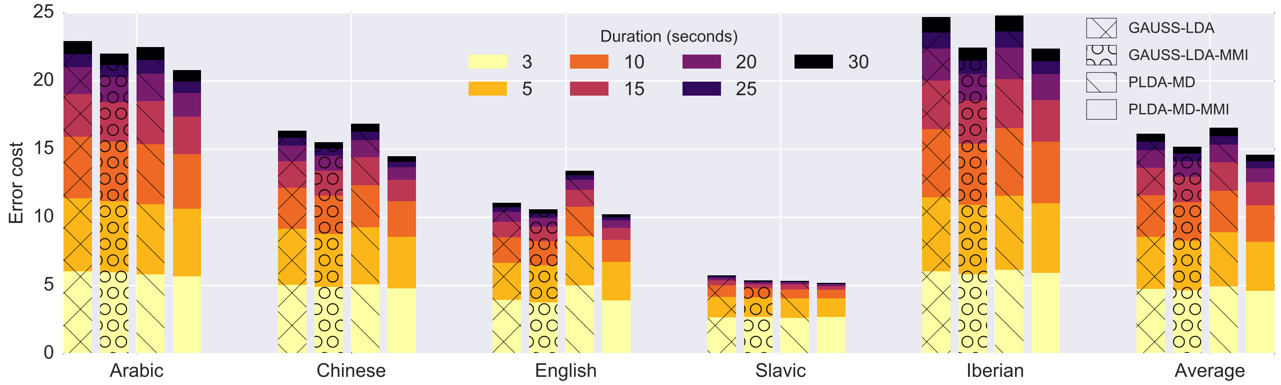


Figure 5: Evaluation of the cost performance (%) on the test set for Gaussian back-end with smoothing and PLDA classifier with minimum-divergence scoring together with their corresponding discriminative counterparts.

recognition: Modeling, criteria, optimization, implementation, and performance," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 58–69, 2012.

[8] Lukáš Burget, Pavel Matějka, and Jan Černocký, "Discriminative training techniques for acoustic language identification," in *ICASSP*, 2006.

[9] Christopher Bishop and Julia Lasserre, "Generative or discriminative? Getting the best of both worlds," *Bayesian statistics*, vol. 8, pp. 3–24, 2007.

[10] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 19, pp. 788–798, 2011.

[11] Elliot Singer, Pedro Torres-Carrasquillo, Douglas Reynolds, Alan McCree, Fred Richardson, Najim Dehak, and Doug Sturim, "The MITLL NIST LRE 2011 language recognition system," in *ICASSP*, 2012.

[12] Niko Brümmer, Sandro Cumani, Ondřej Glembek, Martin Karafiát, Pavel Matějka, et al., "Description and analysis of the Brno276 system for LRE2011," in *Odyssey: the Speaker and Language Recognition Workshop*, 2012.

[13] Najim Dehak, Pedro Torres-Carrasquillo, Douglas Reynolds, and Reda Dehak, "Language recognition via ivectors and dimensionality reduction," in *Interspeech*, 2011.

[14] David Martínez, Oldřich Plchot, Lukáš Burget, Ondřej Glembek, and Pavel Matějka, "Language recognition in iVectors space," in *Interspeech*, 2011.

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

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

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

[18] Aleksandr Sizov, Kong Aik Lee, and Tomi Kinnunen, "Unifying probabilistic linear discriminant analysis variants in biometric authentication," in *S+SSPR*, 2014.

Table 5: Relative improvement (%) of the average cost performance for discriminative systems from Figure 5 with respect to the Gaussian classifier (GAUSS-LDA) on the test set.

Classifier	Duration, sec						
	3	5	10	15	20	25	30
GAUSS	2.6	3.9	7.2	8.9	10.9	12.7	11.7
PLDA	3.2	5.5	12.2	16.3	18.8	20.6	20.0

[19] Patrick Kenny, Themos Stafylakis, Pierre Ouellet, Mohammad Jahangir Alam, and Pierre Dumouchel, "PLDA for speaker verification with utterances of arbitrary duration," in *ICASSP*, 2013.

[20] Liping Chen, Kong Aik Lee, Bin Ma, Wu Guo, Haizhou Li, and Li Rong Dai, "Minimum divergence estimation of speaker prior in multi-session PLDA scoring," in *ICASSP*, 2014.

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

[22] Christopher Bishop, *Pattern Recognition and Machine Learning*, 2006.

[23] Kong Aik Lee, Ville Hautamäki, Anthony Larcher, Wei Rao, et al., "Fantastic 4 system for NIST 2015 language recognition evaluation," Tech. Rep., 2015, `http://arxiv.org/abs/1602.01929`.

[24] Yuning Song, Bo Jiang, YeBo Bao, Shaojun Wei, and Li-Rong Dai, "I-vector representation based on bottleneck features for language identification," *Electronics Letters*, vol. 49, no. 24, pp. 1569–1570, 2013.

[25] Fred Richardson, Douglas Reynolds, and Najim Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Processing Letters*, vol. 22, pp. 1671–1675, 2015.

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

[27] Andrew Hatch, Sachin Kajarekar, and Andreas Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," in *Interspeech*, 2006.