Exploiting i–vector posterior covariances for short–duration language recognition

Exploiting i–vector posterior covariances for short–duration language recognition

Sandro Cumani[1], Oldřich Plchot[2], Radek Fér[2]

[1]Politecnico di Torino, Italy
[2]Brno University of Technology, Speech@FIT and IT4I Center of Excellence, Brno, Czech Republic
sandro.cumani@polito.it
{iplchot,ifer}@fit.vutbr.cz

Abstract

Linear models in i–vector space have shown to be an effective solution not only for speaker identification, but also for language recogniton. The i–vector extraction process, however, is affected by several factors, such as noise level, the acoustic content of the utterance and the duration of the spoken segments. These factors influence both the i–vector estimate and its uncertainty, represented by the i–vector posterior covariance matrix. Modeling of i–vector uncertainty with Probabilistic Linear Discriminant Analysis has shown to be effective for short-duration speaker identification. This paper extends the approach to language recognition, analyzing the effects of i–vector covariances on a state–of–the–art Gaussian classifier, and proposes an effective solution for the reduction of the average detection cost (C_{avg}) for short segments.

Index Terms: i–vector, uncertainty, calibration, stacked bottleneck features, language identification

1. Introduction

I–vectors [1] have become a standard approach for speaker identification, and have grown in popularity also for language recognition [2, 3, 4, 5]. An i–vector is a compact representation of a Gaussian Mixture Model (GMM) supervector [6], which captures most of the GMM supervectors variability. It is obtained by a Maximum–A–Posteriori (MAP) estimate of the mean of a posterior distribution [7]. Recent works [8, 9, 10, 11] have shown that, for speaker identification with short utterances, the approximation introduced by performing a point–estimate of an i–vector can adversely impact the accuracy of a speaker recognition system. Indeed, the uncertainty in the i–vector extraction process, represented by the i–vector posterior covariance, conveys useful information that can be exploited by classifiers based on Probabilistic Linear Discriminant Analysis (PLDA). I–vectors have shown to provide very good results also for language recognition. Generative Gaussian models in i–vector space [3, 5] can provide results that are similar or better than those of discriminative classifiers based on Support Vector Machines or Multiclass Logistic Regression [3]. As in speaker recognition, however, these classifiers do not exploit the i–vector uncertainty. The goal of this work is therefore the extension to language recognition of the Full–Posterior–Distribution (FPD) PLDA approach introduced in [8, 9]. In particular, we follow the approach in [5] to show that the Gaussian Backend (GB) model [12], which has been used in [3] for i–vectors classification, can be interpreted as an approximation of PLDA suited for closed–set detection, and that the (closed–set) PLDA scoring becomes equivalent to GB scoring whenever the number of training utterances for each language is sufficiently high. The use of PLDA for language recognition has two advantages. It allows addressing *both* open–set and closed–set language identification tasks, because it allows computing open–set detection likelihood ratios, from which closed–set likelihood–ratios can be recovered [13]. The second advantage is that we can directly apply to language recognition the derivations of the FPD–PLDA approach of [8, 9].

In this work we present the experimental results of a FPD–PLDA system on the 2009 NIST Language Recognition Evaluation (LRE) [14]. Consistently with our findings for speaker identification, the results show that modeling the i–vector uncertainty can be beneficial for short utterances.

The paper is organized as follows. Section 2 briefly describes the i–vector extraction process. Section 3 recalls the GB generative model and its relationship with PLDA. Section 4 shows how i–vector uncertainty can be modeled in the context of PLDA and GB classifiers. Our experimental setup is presented in Section 5, and results are given in Section 6. Conclusions are drawn in Section 7.

2. I–vector model

The i–vector model constrains the GMM supervector representing the characteristics of a given speech segment, to live in a single small-dimensional subspace according to:

$$\mathbf{s} = \mathbf{u} + \mathbf{T}\mathbf{w}\,, \tag{1}$$

where $\mathbf{u}$ is the supervector stacking the means of the Universal Background Model (UBM), composed of C components of dimension F. $\mathbf{T}$ is a low–rank matrix spanning the i–vector supspace, and $\mathbf{w}$ is a realization of a latent variable $\mathbf{W}$, of size M, having a standard normal prior distribution. Given $\mathbf{T}$ and a set of τ feature vectors $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_\tau\}$ the posterior distribution of $\mathbf{W}$ given $\mathcal{X}$ can be computed as:

$$\mathbf{W}|\mathcal{X} \sim \mathcal{N}\left(\boldsymbol{\mu}_{\mathcal{X}}, \boldsymbol{\Gamma}_{\mathcal{X}}^{-1}\right)\,, \tag{2}$$

where

$$\boldsymbol{\Gamma}_{\mathcal{X}} = \mathbf{I} + \sum_{c=1}^{C} N_{\mathcal{X}}^{(c)} \mathbf{T}^{(c)^{\mathrm{T}}} \boldsymbol{\Sigma}^{(c)^{-1}} \mathbf{T}^{(c)}$$

$$\boldsymbol{\mu}_{\mathcal{X}} = \boldsymbol{\Gamma}_{\mathcal{X}}^{-1} \mathbf{T}^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} \mathbf{f}_{\mathcal{X}}\,.$$

In these equations, $N_{\mathcal{X}}^{(c)}$ are the zero–order statistics estimated on the c-th Gaussian component of the UBM for the set of feature vectors in $\mathcal{X}$, $\mathbf{T}^{(c)}$ is the $F \times M$ sub-matrix of $\mathbf{T}$

corresponding to the c–th mixture component such that $\mathbf{T} = \left(\mathbf{T}^{(1)\mathrm{T}}, \ldots, \mathbf{T}^{(C)\mathrm{T}}\right)^{\mathrm{T}}$, and $\mathbf{f}_{\mathcal{X}}$ is the supervector stacking the first–order statistics $\mathbf{f}_{\mathcal{X}}^{(c)}$, centered around the corresponding UBM means:

$$\mathbf{f}_{\mathcal{X}}^{(c)} = \sum_t \left(\gamma_t^{(c)} \mathbf{x}_t\right) - N_{\mathcal{X}}^{(c)} \mathbf{m}^{(c)} \,, \tag{3}$$

$\mathbf{\Sigma}^{(c)}$ is the UBM c–th covariance matrix, $\mathbf{\Sigma}$ is a block diagonal matrix with matrices $\mathbf{\Sigma}^{(c)}$ as its entries, and $\gamma_t^{(c)}$ is the occupation probability of feature vector $\mathbf{x}_t$ for the c-th Gaussian component.

In the i–vector paradigm, an utterance is represented as the MAP point–estimate $\boldsymbol{\mu}_{\mathcal{X}}$ of the i–vector posterior distribution, and the term i–vector usually refers to this point–estimate. In this work, however, we are interested in exploiting the additional information conveyed by the uncertainty in the i–vector extraction process, represented by the i–vector posterior covariance $\mathbf{\Gamma}_{\mathcal{X}}^{-1}$. Thus, we will explicitly refer to $\boldsymbol{\mu}_{\mathcal{X}}$ as the "i–vector point–estimate", to avoid confusion with the i–vector posterior distribution. In order to increase readability, in the following we will also drop the reference to the feature set $\mathcal{X}$ from $\boldsymbol{\mu}_{\mathcal{X}}$ and $\mathbf{\Gamma}_{\mathcal{X}}$.

3. Gaussian models for language recognition

Generative modeling of i–vector point–estimates for language recognition has proven to be an effective alternative to discriminative classifiers based on Logistic Regression or Support Vector Machines. In [3] the authors have proposed a simple linear classifier based on Gaussian distributions which provides accuracies similar to those of linear discriminative approaches. The model assumes that, for each language, the corresponding i–vector point–estimates $\boldsymbol{\mu}_i$ are generated according to:

$$\boldsymbol{\mu}_i = \mathbf{m}_\ell + \boldsymbol{\varepsilon}_i \,, \tag{4}$$

where $\mathbf{m}_\ell$ is a language–dependent mean vector and

$$\boldsymbol{\varepsilon}_i \sim \mathcal{N}\left(\mathbf{0}, \mathbf{\Lambda}^{-1}\right) \tag{5}$$

represents a (language–independent) residual. The model parameters can be easily obtained by Maximum–Likelihood estimation. The class–conditional log–likelihood for $\boldsymbol{\mu}_i$ given language ℓ can be computed as:

$$\log P(\boldsymbol{\mu}_i|\ell) = \frac{1}{2}\log|\mathbf{\Lambda}| - \frac{1}{2}(\boldsymbol{\mu}_i - \mathbf{m}_\ell)^T \mathbf{\Lambda}(\boldsymbol{\mu}_i - \mathbf{m}_\ell) + k \,, \tag{6}$$

where k is a data–independent constant. A drawback of model (4) is that it defines only class–conditional likelihoods [3]. Therefore, it allows computing only closed–set likelihood ratios, and it is not suited for open–set identification tasks. However, as already mentioned in [5], the model (4) can be seen as an approximation of the PLDA model, which is suited for *both* open–set and closed–set tasks. Moreover, addressing the LID taks by means of PLDA allows us to directly introduce i–vector uncertainty in the model using the same approach of [8, 9].

3.1. PLDA and Gaussian Backend

The PLDA model describes the i–vector generation process as:

$$\boldsymbol{\mu}_i = \mathbf{m} + \mathbf{U}\mathbf{y} + \boldsymbol{\varepsilon}_i \,, \tag{7}$$

where $\mathbf{m}$ is a fixed mean vector, $\mathbf{y}$ is a standard normal distributed hidden variable, $\boldsymbol{\varepsilon}_i \sim \mathcal{N}(0, \mathbf{\Lambda}^{-1})$ is a residual term, and $\mathbf{U}$ is a matrix whose columns span the subspace for the hidden variable $\mathbf{y}$. In speaker identification variable $\mathbf{y}$ represents the speaker identity. For language identification we can assume the same model for the i–vector point estimate generation process, with the hidden variable $\mathbf{y}$ representing the language. Given a trained model $\mathcal{M}$, PLDA allows computing the open–set detection likelihood ratios:

$$\begin{aligned} r &= \frac{P\left(\boldsymbol{\mu}, \mathbf{D}_\ell | \mathcal{H}_S, \mathcal{M}\right)}{P\left(\boldsymbol{\mu}, \mathbf{D}_\ell | \mathcal{H}_D, \mathcal{M}\right)} \\ &= \frac{P\left(\boldsymbol{\mu}, |\mathbf{D}_\ell, \mathcal{H}_S\right) P\left(\mathbf{D}_\ell\right)}{P\left(\boldsymbol{\mu}\right) P\left(\mathbf{D}_\ell\right)} \\ &= \frac{\int P\left(\boldsymbol{\mu}|\mathbf{y}\right) P\left(\mathbf{y}|\mathbf{D}_\ell\right) \mathrm{d}\mathbf{y}}{\int P\left(\boldsymbol{\mu}|\mathbf{y}\right) P\left(\mathbf{y}\right) \mathrm{d}\mathbf{y}} \,, \end{aligned} \tag{8}$$

where the conditioning on $\mathcal{M}$ was dropped to ease readability. In (8), $\mathbf{D}_\ell$ denotes the set of training utterances for language ℓ, $\mathcal{H}_S$ and $\mathcal{H}_D$ denote the same–language and different–language hypotheses, respectively. Equations (8) correspond to the familiar likelihood–ratio expressions for speaker identification, where $\boldsymbol{\mu}$ plays the role of test segment and $\mathbf{D}_\ell$ represents the set of enrollment utterances for a target speaker. Indeed, expressions (8) allow addressing open–set language identification tasks with the same approaches used in speaker identification. Closed–set likelihood ratios and class posteriors required for closed–set identification can be directly computed from open–set likelihood–ratios [13].

For closed–set tasks, the PLDA model becomes equivalent to the GB model, whenever the size of $\mathbf{D}_\ell$ is large enough. Indeed, the numerator of (8) can be interpreted as the class–conditional likelihood for an i–vector point estimate:

$$P(\boldsymbol{\mu}|\ell) = P(\boldsymbol{\mu}|\mathbf{D}_\ell) = \int P(\boldsymbol{\mu}|\mathbf{y}) P(\mathbf{y}|\mathbf{D}_\ell) \mathrm{d}\mathbf{y} \,. \tag{9}$$

If the size of $\mathbf{D}_\ell$ is sufficiently large, as it usually happens in language recognition, the posterior distribution for $\mathbf{y}_\ell|\mathbf{D}_\ell$ becomes sharp, and can be replaced by its MAP point estimate

$$\widehat{\mathbf{y}}_\ell = \left(\mathbf{U}^T \mathbf{\Lambda} \mathbf{U} + \frac{\mathbf{I}}{N_\ell}\right)^{-1} \mathbf{U}^T \mathbf{\Lambda} \overline{\mathbf{m}}_\ell \,, \tag{10}$$

where $\overline{\mathbf{m}}_\ell = \frac{1}{N_\ell}\sum_i \left(\boldsymbol{\mu}_{\ell,i} - \mathbf{m}\right)$ and N_ℓ is the number of utterances for language ℓ. Assuming that $\overline{\mathbf{m}}_\ell$ lies in the range space of $\mathbf{U}$, as N_ℓ increases the PLDA term $\mathbf{U}\widehat{\mathbf{y}}_\ell$ converges to:

$$\mathbf{U}\widehat{\mathbf{y}}_\ell \xrightarrow{N_\ell \to \infty} \overline{\mathbf{m}}_\ell \,, \tag{11}$$

and the class–conditional likelihood $P(\boldsymbol{\mu}|\ell)$ has the same expression of (6).

4. Gaussian models and i–vector uncertainty

In the previous section we have shown that the generative models employed for closed–set LID tasks can be interpreted as an approximation of the PLDA model. This allows us to account for i–vector uncertainty following exactly the same approach that has been used for speaker recognition [8, 9]. In particular, the i–vector uncertainty can be taken into account through the modified PLDA model:

$$\boldsymbol{\mu}_i = \mathbf{m} + \mathbf{U}\mathbf{y} + \overline{\boldsymbol{\varepsilon}}_i \,, \tag{12}$$

where the residual term $\boldsymbol{\varepsilon}_i$ in (7) has been replaced by the term $\overline{\boldsymbol{\varepsilon}}_i$, with an utterance–dependent distribution given by:

$$\overline{\boldsymbol{\varepsilon}}_i \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Lambda}_{eq,i}^{-1}\right) ,$$
$$\boldsymbol{\Lambda}_{eq,i}^{-1} = \left(\boldsymbol{\Lambda}^{-1} + \boldsymbol{\Gamma}_i^{-1}\right) , \quad (13)$$

where $\boldsymbol{\Gamma}_i$ is the i–vector posterior precision. This model can again be interpreted as a generative model for i–vector point estimates, where the i–vector posterior covariance appears through an additional linear term in the PLDA formulation [10, 8]. Model parameters can be estimated through Expectation–Maximization following the approach in [10]. For long training utterances, however, i–vector covariances can be safely neglected during training, so that PLDA parameters can be obtained using the standard approach. Long training utterances allow also for efficient scoring strategies, such as the Asymmetric FPD–PLDA scoring [8], which uses point–estimates for enrollment segments. Moreover, if we are interested only in closed–set detection, and training utterances are long, the model can be simplified as:

$$\boldsymbol{\mu}_i = \mathbf{m}_\ell + \overline{\boldsymbol{\varepsilon}}_i . \quad (14)$$

The class–conditional log–likelihoods $\log P(\boldsymbol{\mu}_i|\ell)$ for a test i–vector mean $\boldsymbol{\mu}_i$, with associated i–vector posterior covariance $\boldsymbol{\Gamma}_i^{-1}$, given language ℓ, can be computed as:

$$\log P(\boldsymbol{\mu}_i|\ell) = -\frac{1}{2}(\boldsymbol{\mu}_i - \mathbf{m}_\ell)^T \left(\boldsymbol{\Lambda}^{-1} + \boldsymbol{\Gamma}_i^{-1}\right)^{-1} (\boldsymbol{\mu}_i - \mathbf{m}_\ell) - \frac{1}{2}\log\left|\boldsymbol{\Lambda}^{-1} + \boldsymbol{\Gamma}_i^{-1}\right| + k \quad (15)$$

where k is a data–independent constant.

5. Experimental set–up

5.1. LID training and evaluation corpora

The results of this work are presented for the NIST Language Recognition Evaluation (LRE) 2009 [14]. Model training follows the setup in [15]. In particular, training data comprises utterances from the Callfriend, Fisher English Part 1 and 2, Fisher Levantine Arabic, HKUST Mandarin, Mixer (data from NIST SRE 2004, 2005, 2006, 2008) datasets. The data has been arranged in three sets. The first set, denoted as *full54*, contains all the utterances in the datasets, belonging to 54 languages and corresponding to 79 thousand segments. The second set, denoted as *full23*, is a subset of *full54* set and contains utterances from the 23 target languages from NIST LRE 2009, corresponding to about 51 thousand segments. The third set, denoted as *balanced*, is a subset of *full23* containing at most 500 utterances for every language, corresponding to a total of 9.8 thousand segments.

The UBM was trained using the *balanced* dataset, while the i–vector extractor was trained on the *full54* set. PLDA and GB have been trained on the *full23* set, restricted to utterances of at least 60 seconds(with the exception of Indian English, for which only shorter segments were available). Calibration was trained on a separate development dataset, which comprises data from all previous NIST LRE evaluations, OGI-multilingual, OGI 22 languages, Foreign Accented English, SpeechDat-East, Switch Board and Voice of America radio broadcasts.

5.2. LID system description

The architecture chosen for our LID system is based on the state-of-the-art acoustic i–vector system from [3], with acoustic features based on Stacked Bottle–Neck (SBN) instead of Shifted Delta Cepstra (SDC) coefficients [12]. The choice of stacked bottle–neck features was motivated by the superior results these features achieved with respect to SDCs [16]. A full description of SBN can be found in [17], and is summarized in the next paragraph.

5.2.1. Stacked Bottleneck features

Bottleneck Neural Networks and especially their multilingual variants have become a favorite tool to extract information–rich features from the acoustic signal. This approach has been successfully used for speech recognition [18, 19, 20] and recently also in the field of speaker and language recognition [21, 22, 23].

These networks are characterized by the presence of a low–dimensional intermediate hidden layer, which compresses the information needed to map the network inputs to its outputs. The networks are trained for a specific task, in our case phone state classification. Bottleneck features are the outputs of the low-dimensional bottleneck layer.

In the Stacked Bottleneck approach, a cascade of two such networks is used. The bottleneck outputs of the first network are stacked in time, and used as inputs for a second network. This allows providing the second network with a broader context input.

In our work, the network input features are 24 mel-scale filter bank outputs augmented with fundamental frequency features as described in [17, 16]. The network is trained following a multilingual approach [20], so that the final bottleneck features are able to capture relevant information for more than one language. The training scheme is based on a block–softmax approach [19]. The network was trained on the IARPA Babel Program data[1]. More details about the data can be found in [24].

5.2.2. Estimation of i–vectors and scoring

After feature extraction, voice activity detection (VAD) is performed by the BUT Hungarian phoneme recognizer, dropping all frames that are labeled as silence or noise. The GMM is composed of 2048 full–covariance components. The dimension of i–vectors was set to 400. Before training the PLDA or GB models, i–vector point–estimates have been whitened by means of Within Class Covariance Normalization (WCCN) and length–normalized. The i–vector posterior covariances are normalized accordingly. In particular, the transformed i–vector posteriors means and covariances are computed as:

$$\boldsymbol{\mu} \leftarrow \frac{\mathbf{A}\boldsymbol{\mu}}{\|\mathbf{A}\boldsymbol{\mu}\|} , \qquad \boldsymbol{\Gamma}^{-1} \leftarrow \frac{\mathbf{A}\boldsymbol{\Gamma}^{-1}\mathbf{A}^T}{\|\mathbf{A}\boldsymbol{\mu}\|^2} , \quad (16)$$

where $\mathbf{A}$ is the inverse of the Cholesky decomposition of the training i–vector point–estimates within–class covariance matrix $\mathbf{C}_w = \mathbf{A}^{-1}\mathbf{A}^{-T}$. A rationale for these transformations can be found in [8].

5.2.3. Calibration

A simple linear model with a single scaling factor and a language–dependent bias was used for calibration [13]. The parameters were obtained optimizing the C_{llr} cost [13] on the development set. The L–BFGS algorithm [25] was used to optimize the objective function.

[1] Collected by Appen `http://www.appenbutlerhill.com`

Table 1: Actual and optimal % C_{avg} and normalized C_{llr} for the 3s, 10s and 30s conditions of the NIST LRE 2009 evaluation.

System	3s condition				10s condition				30s condition			
	% C_{avg}	% C^*_{avg}	C_{llr}	C^*_{llr}	% C_{avg}	% C^*_{avg}	C_{llr}	C^*_{llr}	% C_{avg}	% C^*_{avg}	C_{llr}	C^*_{llr}
PLDA	6.43	6.12	0.254	0.246	2.07	1.78	0.104	0.091	1.20	1.11	0.071	0.060
GB	6.44	6.11	0.254	0.246	2.07	1.78	0.104	0.091	1.21	1.11	0.071	0.060
FPD PLDA	5.99	5.68	0.237	0.227	2.03	1.75	0.100	0.087	1.21	1.12	0.071	0.059
FPD GB	6.03	5.73	0.239	0.229	2.05	1.75	0.101	0.087	1.23	1.13	0.071	0.059

6. Experimental results

Results are reported in terms of percent C_{avg} as defined by NIST [14], and in terms of C_{llr} [13], normalized so that a well–calibrated, but useless, recognizer would obtain $C_{llr} = 1$. We also report the "optimal" costs, denoted by C^*_{avg} and C^*_{llr}. Optimal costs should be interpreted as the costs that would be obtained by performing a "cheating" calibration, i.e., by training the calibration directly on the evaluation set.

Table 1 shows the results of the different systems on the 3, 10 and 30s conditions defined by NIST for the 2009 LRE evaluation. The first and second line of the table show the results of a PLDA and of a GB classifier, respectively. As expected, the two systems provide very close results. The third and fourth lines show the results of the PLDA and GB classifiers incorporating the i–vector uncertainty, denoted as FPD–PLDA and FPD–GB, respectively. The covariance of training i–vectors was ignored both in model training and scoring, because training utterances are long. The results show that the introduction of the i–vector uncertainty allows reducing both the actual and optimal costs for short utterances, whereas the accuracy of the standard approaches is retained for longer utterances. As for standard PLDA and GB, also the FPD–PLDA and FPD–GB systems have very close performance.

7. Conclusions

In this work we have proposed an approach that accounts for the uncertainty in the i–vector extraction process in the framework of generative Gaussian models for language recognition. In particular, we have shown that the successful Gaussian linear model for language identification can be interpreted as a particular instance of PLDA. PLDA–based LID allows accounting for i–vector uncertainty using the same techniques employed in speaker recognition. Experimental results show that modeling i–vector uncertainty improves the system accuracy for short segments.

8. Acknowledgements

This work was supported by the DARPA RATS Program under Contract No. HR0011-15-C-0038. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. The work was also supported by Czech Ministry of Interior project No. VG20132015129 "ZAOM" and IT4Innovations Centre of Excellence (CZ.1.05/1.1.00/02.0070). The authors would like to thank František Grézl and Pavel Matějka from the Brno University of Technology for training the neural networks. The authors would also like to thank Daniele Colibro and Claudio Vair from NUANCE for providing a non–official dataset over which the technique was originally validated. Finally, we would like to thank Pietro Laface from Politecnico di Torino for useful discussions. A share of the computational resources for this work was provided by HPC@POLITO (http://www.hpc.polito.it).

9. References

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

[2] N. Dehak, P. A. Torres-Carrasquillo, D. A. Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction," in *Proceedings of Interspeech 2011*, 2011, pp. 857–860.

[3] D. G. Martínez, O. Plchot, L. Burget, O. Glembek, and P. Matějka, "Language recognition in ivectors space," in *Proceedings of Interspeech 2011*, 2011, pp. 861–864.

[4] N. Brummer *et al.*, "Description and analysis of the brno276 system for LRE2011," in *Proceedings of Odyssey: The Speaker and Language Recognition Workshop*, 2012, pp. 216–223.

[5] A. McCree, "Multiclass discriminative training of i–vector language recognition," in *Proceedings of Odyssey: The Speaker and Language Recognition Workshop*, Joensu, Finland, 2014.

[6] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian Mixture Models," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 31–44, 2000.

[7] P. Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms," in *Technical report CRIM-06/08-13*, 2005.

[8] S. Cumani, O. Plchot, and P. Laface, "On the use of i-vector posterior distributions in probabilistic linear discriminant analysis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 846–857, 2014.

[9] S. Cumani, O. Plchot, and P. Laface, "Probabilistic Linear Discriminant Analysis of i–vector posterior distributions," in *Proceedings of ICASSP 2013*, 2013, pp. 7644–7648.

[10] P. Kenny, T. Stafylakis, P. Ouellet, M. Alam, and P. Dumouchel, "PLDA for speaker verification with utterances of arbitrary duration," in *Proceedings of ICASSP 2013*, 2013, pp. 7649–7653.

[11] B. Borgstrom and A. McCree, "Supervector bayesian speaker comparison," in *Proceedings of ICASSP 2013*, 2013, pp. 7693–7697.

[12] P. Torres-Carrasquillo, E. Singer, M. Kohler, R. Greene, D. Reynolds, and J. Deller, "Approaches to language identification using gaussian mixture models and shifted delta cepstral features," in *ICSLP 2002*, Sep. 2002, pp. 89–92.

[13] N. Brümmer and D. van Leuween, "On calibration of language recognition scores," in *Proceedings of Odyssey: The Speaker and Language Recognition Workshop*, 2006.

[14] "The NIST 2009 language recognition evaluation plan," available at http://www.itl.nist.gov/iad/mig/tests/lre/2009/LRE09_EvalPlan_v6.pdf.

[15] Z. Jančík, O. Plchot, N. Brummer, L. Burget, O. Glembek, V. Hubeika, M. Karafiát, P. Matějka, T. Mikolov, A. Strasheim, and J. Černocký, "Data selection and calibration issues in automatic language recognition - investigation with BUT-AGNITIO NIST LRE 2009 system," in *Proceedings of Odyssey: The Speaker and Language Recognition Workshop*, 2010, pp. 215–221.

[16] R. Fér, P. Matějka, F. Grézl, O. Plchot, and J. Černocký, "Multilingual bottleneck features for language recognition," to appear in *Proceedings of Interspeech 2015*.

[17] M. Karafiát, F. Grézl, K. Veselý, M. Hannemann, I. Szőke, and J. Černocký, "BUT 2014 Babel system: Analysis of adaptation in NN based systems," in *Proceedings of Interspeech 2014*, 2014, pp. 3002–3006.

[18] S. Scanzio, P. Laface, L. Fissore, R. Gemello, and F. Mana, "On the use of a multilingual neural network front-end." in *Proceedings of Interspeech 2008*, 2008, pp. 2711–2714.

[19] K. Veselý, M. Karafiát, F. Grézl, M. Janda, and E. Egorova, "The language-independent bottleneck features," in *Proceedings of IEEE 2012 Workshop on Spoken Language Technology*, 2012, pp. 336–341.

[20] F. Grézl, E. Egorova, and M. Karafiát, "Further investigation into multilingual training and adaptation of stacked bottle-neck neural network structure," in *Spoken Language Technology Workshop (SLT)*, South Lake Tahoe, Nevada USA, 2014.

[21] S. Yaman, J. W. Pelecanos, and R. Sarikaya, "Bottleneck features for speaker recognition," in *Proceedings of Odyssey: The Speaker and Language Recognition Workshop*, 2012, pp. 105–108.

[22] P. Matějka *et al.*, "Neural network bottleneck features for language identification," in *Proceedings of Odyssey: The Speaker and Language Recognition Workshop*, Joensu, Finland, 2014.

[23] B. Jiang, Y. Song, S. Wei, J.-H. Liu, I. V. McLouglin, and L.-R. Dai, "Deep bottleneck features for spoken language identification," *PLoS ONE*, 2014.

[24] M. Harper, "The BABEL program and low resource speech technology," in *Proceedings of ASRU 2013*, Dec 2013.

[25] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Program.*, vol. 45, no. 3, pp. 503–528, Dec. 1989.