ALIZE 3.0-Open Source Toolkit for State-of-the-Art Speaker Recognition

Anthony Larcher, Jean-François Bonastre, Benoît Fauve, Kong Aik Lee, Christophe Levy, Haizhou Li, John Mason, Jean-Yves Parfait

ALIZE 3.0 - Open Source Toolkit for State-of-the-Art Speaker Recognition

Anthony Larcher[1]*, Jean-Francois Bonastre*[2]*, Benoit Fauve*[3]*, Kong Aik Lee*[1]*,*
Christophe Lévy[2]*, Haizhou Li*[1]*, John S.D. Mason*[4]*, Jean-Yves Parfait*[5]

[1]Institute for Infocomm Research - A*STAR, Singapore, [2] University of Avignon - LIA, France
[3] ValidSoft Ltd, UK, [4] Swansea University, UK, [5] Multitel, Belgium

alarcher@i2r.a-star.edu.sg

Abstract

ALIZE is an open-source platform for speaker recognition. The ALIZE library implements a low-level statistical engine based on the well-known Gaussian mixture modelling. The toolkit includes a set of high level tools dedicated to speaker recognition based on the latest developments in speaker recognition such as Joint Factor Analysis, Support Vector Machine, i-vector modelling and Probabilistic Linear Discriminant Analysis. Since 2005, the performance of ALIZE has been demonstrated in series of Speaker Recognition Evaluations (SREs) conducted by NIST and has been used by many participants in the last NIST-SRE 2012. This paper presents the latest version of the corpus and performance on the NIST-SRE 2010 extended task.

Index Terms: speaker recognition, open-source platform, i-vector

1. Introduction

As indicated by the number of applications developed recently, speech technologies have now reached a level of performance that makes them attractive for distributed and embedded applications. Following this trend, NIST speaker recognition evaluations (SREs) have seen their number of participants increase significantly since the first edition. These campaigns clearly illustrate the substantial improvements in performance that have been achieved in the last few years. Speaker verification systems have benefited from a number of developments in noise and channel robustness [34, 30, 3] and new paradigms such as Joint Factor Analysis [21] and i-vectors [12]. At the same time, techniques developed in the field of speaker verification have been shown to be useful for other areas [13, 27, 35].

State-of-the-art techniques are now based on intensive use of corpus and computational resources [16, 11]. The continual improvement in performance calls for enormous number of trials to maintain confidence in the results. For instance, the number of trials from the core task of NIST-SRE evaluation has increased from about 24,000 in 2006 to more than 1.88 millions in 2012 (or 88 millions for the extended task) and participation to such an event has become a true engineering challenge. The rapidly growing effort needed to keep up-to-date with state-of-the-art performance has strongly motivated an increasing number of collaborations between sites. However, system development often remains a challenge and large scale implementation is resource consuming. In this context, collaborative open-source software offers a viable solution as it can be used to reduce the individual development effort and offer a baseline system implementation [26].

The ALIZE project has been initiated in 2004 by the University of Avignon LIA within the ELISA consortium [29] with the aim to create an open-source C++ library for speaker recognition. Since then, many research institutes and companies have contributed to the toolkit through research projects or by sharing source code. More recently the development has been supported by the BioSpeak project, part of the EU-funded Eurostar/Eureka program[1]. Based on the ALIZE core library, high level functionalities dedicated to speaker recognition are available through the LIA_RAL package. All the code from the toolkit is distributed through open source software licenses (LPGL) and has been tested on different platform including Windows, Linux and Mac-OS.

Recent developments include Joint Factor Analysis [21], i-vector modelling and Probabilistic Linear Discriminant Analysis [32]. These developments stem mainly from a collaboration between LIA and the Institute for Infocomm Research (I^2R).

This paper presents an overview of the ALIZE toolkit. Section 2 gives a description of the collaborative tools and details about the toolkit implementation. In Section 3 we describe the main functions available in the LIA_RAL package. Section 4 presents the performance of i-vector systems based on ALIZE for the NIST-SRE 2010 extended task. Finally, Section 5 discusses the future evolution of the project.

2. ALIZE: an Open Source Platform

2.1. A Community of Users

A number of tools are available for dissemination, exchange and collaborative work through a web portal[2]. To federate the community, this portal collects and publishes scientific work and industrial realisations related to ALIZE. The users can register to the mailing list that allows them to be informed of the latest developments and to share their experience with the community. A LinkedIn group also provides a way to know about the facilities and the people working in the field of speaker recognition.

Documentation, wiki and tutorials are available on the website to get started with the different parts of the toolkit. The official release of the toolkit can be downloaded from the website and the latest version of the sources are available through a SVN server.

2.2. Source Code

ALIZE software architecture is based on UML modelling and strict code conventions in order to facilitate collaborative development and maintenance of the code. An open-source and cross-platform test suite enables ALIZEs contributors to

[1] http://www.eurekanetwork.org/activities/eurostars
[2] http://alize.univ-avignon.fr

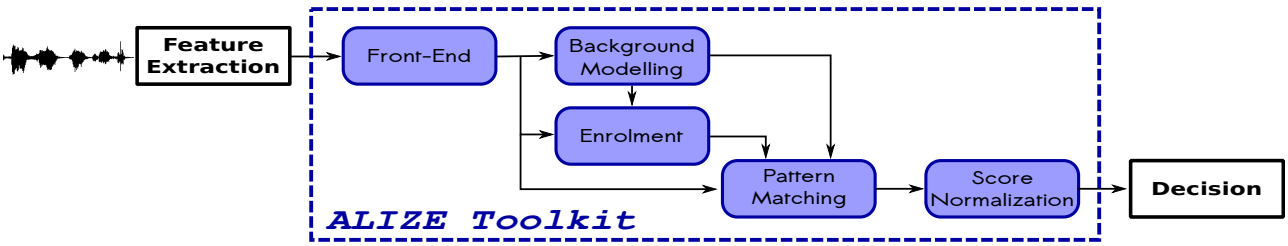

Figure 1: General structure of a speaker verification system.

quickly run regression tests in order to increase the reliability of future releases and to make the code easier to maintain. Test cases include low level unit tests on the core ALIZE and the most important algorithmic classes as well as an integration test level on the high-level executable tools. Doxygen documentation is available on line and can be compiled from the sources.

The platform includes a Visual Studio solution and autotools for compilation under Windows and UNIX-like platforms. A large part of the LIA RAL functions use parallel processing for speed The multi-thread implementation based on the standard POSIX library is fully compatible with the most common platforms. The LIA RAL library can be linked to the well known Lapack[3] library to give high accuracy in matrix manipulations.

All sources are available under LPGL licence that impose minimal restriction on the redistribution of the software.

3. High Level Executables

The goal of this paper is to show the steps to set-up a state-of-the art biometric engine using the different components of ALIZE. For more details, excellent tutorials on speaker recognition can be found in the literature [7, 2, 22].

Figure 1 shows the general architecture of a speaker recognition engine. LIA_RAL high level toolkit provides a number of executables that can be used to achieve the different functions depicted in this diagram. The rest of this section gives an overview of the main functionnalities of the LIA_RAL toolkit with the corresponding executables.

3.1. Front-End

The first stage of a speaker recognition engine consists of a feature extraction module that transforms the raw signal into a sequence of low dimension feature vectors. ALIZE interfaces to features generated by SPRO[4] and HTK[5] and also accepts "raw" format.

Once the acoustic features have been generated, they can be normalised to remove the contribution of slowly varying convolutive noises (mean subtraction) as well as reduced (variance normalization) by using the function `NormFeat`.
Low energy frames, corresponding to silence and noise, can then be discarded with `EnergyDetector`. This executable computes a bi- or tri-Gaussian model of the feature vector or of the log-energy distribution and selects the features belonging to "highest" mean distribution.
Finally, it is possible to smooth a selection of feature vectors. Applied to a single channel recording, `LabelFusion` smooths the selection of frames with a morphological window. When applied to a two-channel recording, `LabelFusion` removes overlapping sections of high energy features.

[3] `http://www.netlib.org/lapack/`
[4] `http://www.irisa.fr/metiss/guig/spro/`
[5] `http://htk.eng.cam.ac.uk/`

3.2. Enrolment

In speaker recognition, the enrolment module generates a statistical model from one or several sequences of features. Although it is possible to generate one model for each recording session, depending on the system's architecture, it is common to consider a single model to represent a speaker. By extension, we refer to this as the speaker model for the remainder of the paper.

State-of-the-art speaker recognition engines are mainly based on three types of speaker models. Although these three models are all related to a Gaussian mixture model (GMM), we distinguish between a first first type of model that explicitly makes use of a GMM and the two other types that represent the speaker or session as a fixed-length vector derived from a GMM. This vector can be a super-vector [8] (concatenation of the mean parameters of the GMM) or a more compact representation known as an i-vector [12]. Each of the three models can be obtained from the corresponding executable of the LIA_RAL toolkit.

Robustness to inter-session variability, that could be due to channel or noise, is one of the main issue in speaker recognition. Therefore, ALIZE includes the most common solutions to this challenge for each type of model described below.

`TrainTarget` generates GMM models given one or more feature sequences. The GMMs can be adapted from a universal background model (UBM) $\mathcal{M}$ by using a maximum a posteriori (MAP) criterion [15, 33] or a maximum likelihood linear regression (MLLR) [25]. Noise and channel robust representations can be obtained by using Factor-Analysis (FA) based approaches. Factor Analysis for speaker recognition has been introduced in [20] and assumes that the variability due to the speaker and channel both lie in distinct low dimension subspaces. Different flavours of FA have been proposed and two are available in ALIZE. The more general one is known as Joint Factor Analysis [21], in which the super-vector $\mathbf{m}_{(s,n)}$ of the session and speaker dependent GMM is a sum of three terms given by Eq.1.

$$\mathbf{m}_{(s,n)} = \mathcal{M} + \mathbf{V}\boldsymbol{y}_{(s)} + \mathbf{U}\boldsymbol{x}_{(s,n)} + \mathbf{D}\boldsymbol{z}_{(s)} \qquad (1)$$

In this formulation, V and U are "factor loaded matrices", D is a diagonal MAP matrix while $\boldsymbol{y}_{(s)}$ and $\boldsymbol{x}_{(s,n)}$ are respectively called speaker and channel factors. $\boldsymbol{y}_{(s)}$, $\boldsymbol{x}_{(s,n)}$ and $\boldsymbol{z}_{(s)}$ are assumed to be independent and have standard normal distributions. A simplified version of this model, often referred to as EigenChannel or Latent Factor Analysis [30] is also available in the `TrainTarget`. In this version, the simplified genera-

tive equation becomes:

$$\mathbf{m}_{(s,n)} = \mathcal{M} + \mathbf{U}\boldsymbol{x}_{(s,n)} + \mathbf{D}\boldsymbol{z}_{(s)} \quad (2)$$

ModeToSv extracts super-vectors from GMMs. A super-vector is the representation of a speaker in a high dimension space that has been popularised by the development of Support Vector Machines (SVM) for speaker recognition [8]. LibSVM library [10] has been integrated into the ALIZE `SVM` executable. Nuisance Attribute Projection (NAP) [34] aims to attenuate the channel variability in the super-vector space by rejecting a subspace that contains most of the channel variability (nuisance effect). The most straightforward approach consists of estimating the within-session co-variance matrix $\boldsymbol{W}$ of a set of speakers and to compute the projection $\hat{\mathbf{m}}_{(s,n)}$ of super-vector $\mathbf{m}_{(s,n)}$ such that:

$$\hat{\mathbf{m}}_{(s,n)} = (I - SS^t)\mathbf{m}_{(s,n)} \quad (3)$$

where S contains the first eigenvectors resulting from the singular value decomposition of $\boldsymbol{W}$.

IvExtractor extracts a low-dimensional i-vector [12] from a sequence of feature vectors. An i-vectors is generated according to

$$\mathbf{m}_{(s,n)} = \mathcal{M} + \mathbf{T}\boldsymbol{w}_{(s,n)} \quad (4)$$

where $\mathbf{T}$ is a low rank rectangular matrix called the Total Variability matrix and the i-vector $\boldsymbol{w}_{(s,n)}$ is the probabilistic projection of the super-vector $\mathbf{m}_{(s,n)}$ onto the Total Variability space, defined by the columns of $\mathbf{T}$.

Many normalization techniques have been proposed to compensate for the session variability into the Total Variability space. The `IvNorm` executable can be used to apply normalizations based on the Eigen Factor Radial (EFR) method [4]. EFR iteratively modifies the distribution of i-vectors such that it becomes standard normal and the i-vectors have a unitary norm. Given a development set $\mathcal{T}$ of i-vectors, of mean $\boldsymbol{\mu}$ and total co-variance matrix $\boldsymbol{\Sigma}$, an i-vectors is modified according to:

$$\boldsymbol{w} \leftarrow \frac{\boldsymbol{\Sigma}^{\frac{1}{2}}(\boldsymbol{w} - \boldsymbol{\mu})}{|\boldsymbol{\Sigma}^{\frac{1}{2}}(\boldsymbol{w} - \boldsymbol{\mu})|} \quad (5)$$

After this transformation has been applied to all i-vectors from the development set $\mathcal{T}$ and from the test data, the mean, $\boldsymbol{\mu}$, and co-variance matrix, Σ, are re-estimated to perform the next iteration. Note that the length-norm proposed in [14] is equivalent to one iteration of the EFR algorithm.

A variation of the EFR, proposed later in [3] as Spherical Nuisance Normalization (sphNorm), is also available in the ALIZE toolkit. For sphNorm, the total co-variance matrix $\boldsymbol{\Sigma}$ is replaced by the within class co-variance matrix of the development set $\mathcal{T}$. After normalization of their norm, all i-vectors lie on a sphere and it is therefore difficult to estimate a relevant within-class co-variance matrix. Spherical Nuisance Normalization is then used to project the i-vectors onto a spherical surface while assuring that there is no principal direction for the session variability.

Other standard techniques such as Within Class Covariance Normalization (WCCN) and Linear Discriminant Analysis (LDA) [12] are also available in ALIZE.

3.3. Pattern Matching

Given a test utterance, $\mathcal{X}$, and a target speaker model, the matching module returns a score that reflects the confidence of the system in $\mathcal{X}$ being spoken by the target speaker. The nature and computation of this score vary depending on the type of speaker model and the different assumptions made. Similarly for the enrolment module, LIA_RAL includes three executables, each dedicated to a specific type of model.

ComputeTest, given a sequence of acoustic features, $\mathcal{X} = \{\boldsymbol{x}_t\}_{t\in T}$ of length T, computes a log-likelihood ratio between the UBM and a speaker dependent GMM. If no channel compensation method is applied, the log-likelihood of utterance $\mathcal{X}$ over a model **s** is computed as the average log-likelihood of features $\boldsymbol{x}_t$ such that:

$$\log P(\mathcal{X}|\boldsymbol{s}) = \sum_{t=1}^{T} \log \sum_{c=1}^{C} \gamma_c \mathcal{N}(\boldsymbol{x}_t, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \quad (6)$$

where C is the number of distribution in $\boldsymbol{s}$ and γ_c, $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$ are the weight, mean and co-variance matrix of the c^{th} distribution respectively.

For the case of Joint Factor Analysis where it is difficult to integrate out the channel effect, `ComputeTest` can compute two approximations of the log-likelihood. The first one is adapted directly from [19] and uses a MAP point estimate for the channel factor and the second is the linear scoring proposed in [17]. A detailed description of both approaches can be found in [17].

SVM returns a score that reflects the distance of a test super-vector to the hyper-plan defined by the classifier. Different kernels such as GLDS [9] or GSL derived from the Kullback-Liebler divergence [8] are available through the LibSVM library.

IvTest is dedicated to i-vector comparison. The i-vector paradigm offers an attractive low-dimensional representation of speech segments that enables standard classification techniques to be applied for speaker recognition. Amongst the most popular scoring methods, four have been implemented in `IvTest`: cosine [12] and Mahalanobis [4] scoring, two-co-variance scoring (2cov) [6] as well as two versions of the Probabilistic Linear Discriminant Analysis (PLDA) scoring [32].

In the remainder of this section, $\boldsymbol{W}$, $\boldsymbol{B}$ and $\boldsymbol{\mu}$ are respectively the within- and between-class co-variance matrices and the mean of a large set of i-vector.

Cosine similarity has been proposed in [12] to compute the similarity between two i-vectors $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$. In the same paper, the authors compensate the session variability through Within Class Co-variance Normalization (WCCN) and Linear Discriminant Analysis (LDA). Considering that $\boldsymbol{\Upsilon}$ is the Cholesky decomposition of the within-class co-variance matrix $\boldsymbol{W}$ calculated over a large data set and that $\boldsymbol{\Lambda}$ is the LDA matrix computed on the same data set, the cosine similarity score is given by:

$$S(\boldsymbol{w}_1, \boldsymbol{w}_2) = \frac{< \boldsymbol{\Upsilon}^t\boldsymbol{\Lambda}^t\boldsymbol{w}_1 | \boldsymbol{\Upsilon}^t\boldsymbol{\Lambda}^t\boldsymbol{w}_2 >}{||\boldsymbol{\Upsilon}^t\boldsymbol{\Lambda}^t\boldsymbol{w}_1||\ ||\boldsymbol{\Upsilon}^t\boldsymbol{\Lambda}^t\boldsymbol{w}_2||} \quad (7)$$

Mahalanobis distance is a generalisation of the Euclidian distance for the case where the data are not following a standard normal distribution. The Mahalanobis score is given by:

$$S(\boldsymbol{w}_1, \boldsymbol{w}_2) = (\boldsymbol{w}_1 - \boldsymbol{w}_2)^t \boldsymbol{W}^{-1} (\boldsymbol{w}_1 - \boldsymbol{w}_2) \quad (8)$$

The two-co-variance model, described in [6], can be seen as a special case of the PLDA. It consists of a simple linear-Gaussian generative model in which an i-vector $\boldsymbol{w}$ can be decomposed as $\boldsymbol{w} = \boldsymbol{y}_s + \boldsymbol{\epsilon}$ where the speaker and noise components $\boldsymbol{y}_s$ and $\boldsymbol{\epsilon}$ follow respectively normal distributions given by $P(\boldsymbol{y}_s) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{B})$ and $P(\epsilon|\boldsymbol{y}_s) = \mathcal{N}(\boldsymbol{y}_s, \boldsymbol{W})$. The resulting score can be expressed as:

$$s = \frac{\int \mathcal{N}(\boldsymbol{w}_1|\boldsymbol{y}, \boldsymbol{W})\, \mathcal{N}(\boldsymbol{w}_2|\boldsymbol{y}, \boldsymbol{W})\, \mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}, \boldsymbol{B})\, d\boldsymbol{y}}{\prod_{i=1,2} \int \mathcal{N}(\boldsymbol{w}_i|\boldsymbol{y}, \boldsymbol{W})\, \mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}, \boldsymbol{B})\, d\boldsymbol{y}} \quad (9)$$

PLDA [32] is one of the most recent addition to the ALIZE toolkit. The generative model of PLDA considers that an i-vector $\boldsymbol{w}$ is a sum of three terms:

$$\boldsymbol{w}_{(s,n)} = \boldsymbol{\mu} + \mathbf{F}\boldsymbol{\phi}_{(s)} + \mathbf{G}\boldsymbol{\psi}_{(s,n)} + \boldsymbol{\epsilon} \quad (10)$$

where $\mathbf{F}$ and $\mathbf{G}$ are low rank speaker and channel "factor loaded matrices". $\boldsymbol{\epsilon}$ is a normally distributed additive noise of full co-variance matrix. ALIZE implementation of the PLDA follows the work of [18]. Two scoring methods, described by Figure 2, have been implemented. The first is based on the native PLDA scoring while the second one is using the mean of the L enrolment i-vectors, $\tilde{\boldsymbol{w}}$. Note that both methods allow multiple enrolment sessions. More details can be found in [23] and in a companion paper [24].

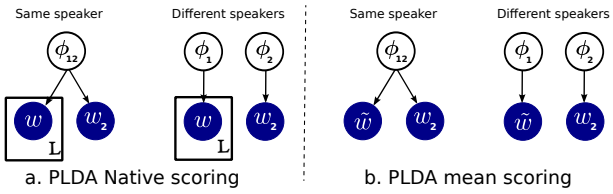

Figure 2: Graphical model of the two PLDA scoring implementations in ALIZE for L enrolment i-vectors.

3.4. Background Modelling

Speaker recognition is a data-driven technology and all approaches implemented in ALIZE rely on a background knowledge learned from a large quantity of development data. Estimation of the knowledge component is computationally intense. Efficient tools have been developed in the toolkit to optimize and simplify the development efforts. The UBM can be trained efficiently by using the `TrainWorld` that uses a random selection of features to speed up the iterative learning process based on the EM algorithm. Meta-parameters of JFA and LFA models can be trained by using `EigenVoice`, `EigenChannel` and `EstimateDmatrix` while the `TotalVariability` has been especially optimised to deal with the computational constraints of learning the Total Variability matrix for i-vector extraction. The implementation follows the work described in [20] with additional minimum divergence described in [5]. Nuisance Attribute Projection matrices can be trained using `CovIntra` and, for i-vector systems, normalization and PLDA meta-parameters can be trained by respectively using `PLDA` and `IvNorm`. PLDA estimation follows the algorithm described in [18].

3.5. Score Normalization

Different combinations of score normalization based on Z- and T-norm are available through `ComputeNorm`, [28, 1].

4. Performance of i-Vector Systems

This section presents the performance of different i-vector systems based on the ALIZE toolkit on the Condition 5 of the NIST-SRE'10 extended task for male speakers [31]. 50 dimension MFCC vectors are used as input features (19 MFCC, 19Δ, 11$\Delta\Delta$ and ΔE). High energy frames are retained and normalized so that the distribution of each cepstral coefficient is 0-mean and 1-variance for a given utterance. A 2048-distribution UBM with diagonal co-variance matrix is trained on 6,687 male sessions from NIST-SRE 04, 05 and 06 telephone and microphone data. The same data augmented with Fisher and Switchboard databases (28,524 sessions) are used to train a Total Variability matrix of rank 500. All meta-parameters required for i-vector normalization and scoring are estimated from 710 speakers from NIST-SRE 04,05 and 06 with a total of 11,177 sessions. Rank of the $\mathbf{F}$ and $\mathbf{G}$ matrices from the PLDA model are set to 400 and 0 respectively. When applied, two iterations of Eigen Factor Radial and 3 iterations of Spherical Nuisance Normalization are performed.

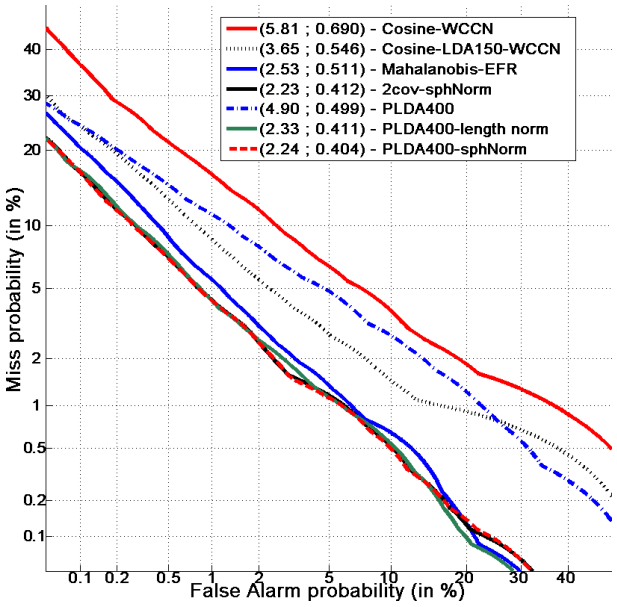

Figure 3: Performance of ALIZE i-vector systems on NIST-SRE10 extended male tel-tel task (condition 5) given in terms of (% EER, minDCF2010).

Figure 3 shows the performance of seven systems using different i-vector normalization and scoring functions. The performance of these systems are consistent with the current state-of-the-art considering that simple acoustic features have been used.

5. Discussion

We have described ALIZE, an open-source speaker recognition toolkit. This toolkit includes most of the standard algorithms recently developed in the field of speaker recognition, including Joint Factor Analysis, i-vector modelling and Probabilistic Linear Discriminant Analysis. The aim of ALIZE collaborative project is to pool development efforts and to make efficient implementation of standard algorithms available for the community. In the future, efforts will be concentrated on the documentation of the toolkit through online help and tutorials.

6. References

[1] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score Normalization for Text-Independent Speaker Verification System," *Digital Signal Processing*, vol. 1, no. 10, pp. 42–54, 2000.

[2] F. Bimbot, J.-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meigner, T. Merlin, J. Ortega-Garcia, D. Petrovska-Delacretaz, and D. A. Reynolds, "A tutorial on text-independent speaker verification," *EURASIP Journal on Applied Signal Processing*, vol. 4, pp. 430–451, April 2004.

[3] P.-M. Bousquet, A. Larcher, D. Matrouf, J.-F. Bonastre, and O. Plchot, "Variance-Spectra based Normalization for I-vector Standard and Probabilistic Linear Discriminant Analysis," in *Odyssey Speaker and Language Recognition Workshop*, 2012.

[4] P.-M. Bousquet, D. Matrouf, and J.-F. Bonastre, "Intersession compensation and scoring methods in the i-vectors space for speaker recognition," in *Annual Conference of the International Speech Communication Association (Interspeech)*, 2011, pp. 485–488.

[5] N. Brümmer. The em algorithm and minimum divergence. Online http://niko.brummer.googlepages. Agnitio Labs Technical Report.

[6] N. Brummer and E. de Villiers, "The speaker partitioning problem," in *Odyssey Speaker and Language Recognition Workshop*, 2010.

[7] J. P. Campbell, "Speaker recognition: a tutorial," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1437–1462, September 1997.

[8] W. M. Campbell, D. E. Sturim, D. A. Reynolds, and A. Solomonoff, "SVM based speaker verification using a GMM supervector kernel and NAP variability compensation," in *Proc. ICASSP*, vol. 1, 2006, pp. 97–100.

[9] W. Campbell, J. Campbell, D. Reynolds, E. Singer, and P. Torres-Carrasquillo, "Support vector machines for speaker and language recognition," in *Computer Speech & Language*, vol. 20. Elsevier, 2006, pp. 210–229.

[10] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 1–27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[11] S. Cumani and P. Laface, " Memory and Computation Trade-Offs for Efficient I-Vector Extraction ," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 5, pp. 934–944, 2013.

[12] N. Dehak, P. A. Torres-Carrasquillo, D. A. Reynolds, and R. Dehak, "Language Recognition via Ivectors and Dimensionality Reduction," in *Annual Conference of the International Speech Communication Association (Interspeech)*, 2011.

[13] C. Fredouille, G. Pouchoulin, A. Ghio, J. Revis, J.-F. Bonastre, A. Giovanni *et al.*, "Back-and-forth methodology for objective voice quality assessment: from/to expert knowledge to/from automatic classification of dysphonia," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, 2009.

[14] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Annual Conference of the International Speech Communication Association (Interspeech)*, 2011, pp. 249–252.

[15] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of Markov chains," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, vol. 2, 1994, pp. 291–298.

[16] O. Glembeck, L. Burget, P. Matejka, M. Karafiat, and P. Kenny, "Simplification and optimization of I-Vector extraction," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, 2011, pp. 4516–4519.

[17] O. Glembek, L. Burget, N. Dehak, N. Brummer, and P. Kenny, "Comparison of Scoring Methods used in Speaker Recognition with Joint Factor Analysis," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, Taipei (Taiwan), 2009.

[18] Y. Jiang, K. A. Lee, Z. Tang, B. Ma, A. Larcher, and H. Li, "PLDA Modeling in I-vector and Supervector Space for Speaker Verification," in *Annual Conference of the International Speech Communication Association (Interspeech)*, 2012.

[19] P. Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms," CRIM, Tech. Rep., 2005.

[20] P. Kenny and P. Dumouchel, "Disentangling speaker and channel effects in speaker verification," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, 2004, pp. 37–40.

[21] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, p. 1435, 2007.

[22] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech Communication*, vol. 52, no. 1, pp. 12–40, 2010.

[23] A. Larcher, K. A. Lee, B. Ma, and H. Li, "Phonetically-Constrained PLDA Modeling for Text-Dependent Speaker Verification with Multiple Short Utterances," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, 2013.

[24] K. A. Lee, A. Larcher, C.-H. You, B. Ma, and H. Li, "Multi-session PLDA Scoring of I-vector for Partially Open-set Speaker Detection," in *Annual Conference of the International Speech Communication Association (Interspeech)*, 2013.

[25] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer Speech and Language*, vol. 9, no. 2, pp. 171–185, April 1995.

[26] H. Li and M. Bin, "TechWare: Speaker and Spoken Language Recognition Resources," *IEEE Signal Processing Magazine*, vol. 27, no. 6, pp. 139–142, 2010.

[27] H. Li, B. Ma, , and K. A. Lee, "Spoken language recognition: from fundamentals to practice," *Proceedings of the IEEE*, 2013.

[28] K.-P. Li and J. E. Porter, "Normalizations and selection of speech segments for speaker recognition scoring," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, vol. 1, New York (USA), April 1998, pp. 595–598.

[29] I. Magrin-Chagnolleau, G. Gravier, , and R. Blouet, "Overview of the 2000-2001 ELISA consortium research activities," in *Odyssey Speaker and Language Recognition Workshop*, 2001.

[30] D. Matrouf, N. Scheffer, B. Fauve, and J.-F. Bonastre, "A straightforward and efficient implementation of the factor analysis model for speaker verification," in *Annual Conference of the International Speech Communication Association (Interspeech)*, 2007.

[31] NIST, "Speaker recognition evaluation plan," http://www.itl.nist.gov/iad/mig/tests/sre/2010/NISTSRE10evalplan.r6.pdf, 2010.

[32] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[33] D. A. Reynolds and R. C. Rose, "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 3, no. 1, pp. 72–83, January 1995.

[34] A. Solomonoff, W. Campbell, and I. Boardman, "Advances in channel compensation for svm speaker recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, vol. 1, 18-23, 2005, pp. 629–632.

[35] C. Vaquero, A. Ortega, and E. Lleida, "Intra-session variability compensation and a hypothesis generation and selection strategy for speaker segmentation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, 2011, pp. 4532–4535.