AN EXTENSIBLE SPEAKER IDENTIFICATION SIDEKIT IN PYTHON

Anthony Larcher[1], *Kong Aik Lee*[2], *Sylvain Meignier*[1]

[1]LIUM - Université du Maine, France
[2]Human Language Technology Department, Institute for Infocomm Research, A*STAR, Singapore
anthony.larcher@univ-lemans.fr

ABSTRACT

SIDEKIT is a new open-source Python toolkit that includes a large panel of state-of-the-art components and allow a rapid prototyping of an end-to-end speaker recognition system. For each step from front-end feature extraction, normalization, speech activity detection, modelling, scoring and visualization, SIDEKIT offers a wide range of standard algorithms and flexible interfaces. The use of a single efficient programming and scripting language (Python in this case), and the limited dependencies, facilitate the deployment for industrial applications and extension to include new algorithms as part of the whole tool-chain provided by SIDEKIT. Performance of SIDEKIT is demonstrated on two standard evaluation tasks, namely the RSR2015 and NIST-SRE 2010.

Index Terms: speaker recognition, toolkit, open-source, python, tutorials

1. INTRODUCTION

Speaker verification is the task of comparing audio recordings to answer the question " Is the same speaker speaking in all the recordings?" [1]. The domain is still an active research area as many problems are not solved; the performance of systems in adverse conditions such as noisy environment, degraded communication channels or short duration of speech samples [2] is still limiting an extensive use of the technology. At the same time, performance in more controlled conditions have reached a point that allows a number of commercial applications.

Over the years, a number of toolkits have been developed that fulfil different purposes; some are dedicated to research and focus on flexibility while others target on efficiency to be compatible with industrial requests. As researchers, we aim at developing new algorithms while keeping close to industrial standards in order to enable quick technology transfer. To achieve these two goals, a speaker recognition toolkit should fulfil a number of requirements:

- easy to understand and modify;
- easy to install and start with;
- allow the development of an end-to-end speaker recognition system;
- minimum dependencies on other tools;
- implement a wide range of standard algorithms;
- enable the use of large data sets an fast computation to obtain state-of-the-art performance;
- manage standard data formats to allow compatibility with existing tools.

Considering the advantages and drawbacks of existing tools, we developed a new toolkit for speaker recognition, SIDEKIT, that aims at fulfilling the above-mentioned requirements while providing an end-to-end solution to integrate a wide choice of state-of-the-art algorithms. Focusing on the easiness of use, we included a complete documentation, examples and tutorials on standard tasks for an easy first-use of the toolkit.

Additionally, the use of an open-source licence would enable a wide diffusion, a quick development and facilitate the technology transfer, if the licence is permissive enough.

This article details our motivations, describes the main functionalities of the Speaker IDEentification toolKIT, SIDEKIT, explains how to start with this new tool and demonstrates the performance of SIDEKIT on two standard tasks.

2. MOTIVATIONS

SIDEKIT aims at providing an end-to-end tool-chain encompassing various state-of-the-art methods, easy to start with and to modify. The content of SIDEKIT has been thought to address the lacks of existing toolkits. Our intention is to keep the architecture simple so as to facilitate the use and the development of new approaches.

2.1. Comparison with other tools

Several good tools are available but don't serve the purpose for one or multiple reasons. ALIZE [3] is an open-source C++ toolkit widely used. It includes recent developments in speaker recognition and its efficient implementation in C++

978-1-4799-9988-0/16/$31.00 ©2016 IEEE

provides fast integration for commercial applications. Modifying the C++ code efficiently requires a deep knowledge of the software architecture and is usually time consuming. Furthermore, ALIZE does not provide feature extraction or visualization tool.

Kaldi [4] is an open-source C++ toolkit dedicated to speech recognition. Due to the recent use of i-vectors for session adaptation [5], an i-vector module has been added into Kaldi that can be used for speaker recognition. Kaldi is evolving quickly thanks to a very dynamic community but the toolkit, for instance the front-end processing, is highly motivated for speech recognition task.

MSR [6] is a Matlab toolbox that includes the entire toolchain to develop an i-vector PLDA system. It includes basic feature extraction and visualization tools but is limited to the i-vector approach. The cost of the Matlab environment limits the use of this tool and integration in a commercial application imposes to rewrite the code in a more standard computer language.

Spear-BOB [7] is one of the most recent toolbox for speaker recognition. The whole chain of recognition is efficiently implemented in C++ and Python including basic feature extraction, GMM modelling, joint factor analysis (JFA), i-vector, back-end and visualization tools. The Python higher layer of Spear makes it easy to set-up a state-of-the-art system but modification of the lower C++ layer could be complex and time consuming.

2.2. Compatibilities with existing tools

In order to benefit from the best of all available tools and to facilitate smooth transitions between them, SIDEKIT is compatible with some of the most popular formats for speaker recognition. SIDEKIT is able to read and write features in both SPRO4[1] and HTK [8] formats, and GMMs in ALIZE [3] and HTK formats. Most of the objects in SIDEKIT can also be saved in the open and portable HDF5 format used in BOSARIS[2].

2.3. Structure of SIDEKIT

SIDEKIT is 100% Python and has been tested on several platforms under Python 2.7 and > 3.4. The toolkit has been developed with minimum dependencies to external modules and to make full use of the most standard Python modules for linear algebra, matrix manipulation, etc.. To maximize readability and flexibility, SIDEKIT is built on a limited number of classes that are listed below.

FeaturesServer offers a simple interface to load and save acoustic features read in SPRO4, HTK format or extracted from audio files (RAW, WAV, SPHERE)

StatServer class used to store and process zero and first order statistics considering different types of observations (acoustic features, i-vectors or super-vectors)

Mixture stores and process Gaussian Mixture Models (GMM)

Bosaris classes SIDEKIT makes use of the main classes of the BOSARIS toolkit to manage files and trial lists, scores matrices and DET plots.

3. WHAT IS IN SIDEKIT?

This section describes the main features included in the toolkit by the time we wrote this article. On-going development that will be included in the toolkit will be discussed in the last section of this article.

3.1. Front-End

SIDEKIT offers a simple interface to extract, extend and normalize filter banks and cepstral coefficients with linear- or Mel-scale filter bank (LFCC and MFCC). Two voice-activity detection algorithms based on energy are available. Additionally, SIDEKIT supports selection of feature frames based on external labels and exports labels in ALIZE format.

Several options are offered for contextualization of acoustic features. In particular, Δ and $\Delta\Delta$ can be computed with a simple two points difference or by using a window filtering as described in [9]. Alternatively, a recently proposed method based on a 2D-DCT followed by Principal Component Analysis dimension reduction is also provided [10].

Standard normalizations are implemented: cepstral mean subtraction (CMS), cepstral mean variance normalization (CMVN) and short term Gaussianization (STG) [11]. RASTA filtering is also included in the toolkit.

The `FeaturesServer` includes standard front-end algorithms organized in sequential function calls that enable easy integration of new methods for the different steps of the process.

3.2. Modelling and classifiers

The core of SIDEKIT is based on GMM-based approaches. The `Mixture` class includes two versions of the *Expectation Maximization* (EM) algorithm with *Maximum Likelihood* criteria to train a Universal Background Model (UBM). One that initializes a single Gaussian and perform iterative splitting based on variance gradient, and a second that randomly initializes a GMM and performs EM algorithms with a constant number of distributions. The mixtures variance can be constrained between a flooring and a ceiling value. Target model can be enrolled using *Maximum a Posteriori* (MAP) adaptation [12]

[1] http://www.irisa.fr/metiss/guig/spro/

[2] https://sites.google.com/site/bosaristoolkit/

On top of the simple GMM modelling, SIDEKIT integrates Factor Analysis based approaches in a single framework. Indeed, both Joint-Factor Analysis (JFA)[13, 14] and Probabilistic Linear Discriminant Analysis (PLDA)[15] were derived from a basic Factor Analyser [16] and therefore share a common decoupled implementation despite exhibiting two major differences. Firstly, JFA considers acoustic frames as observations while PLDA models a single distribution of i-vectors [17] or super-vectors [18]. Secondly, JFA ties latent factors across a temporal sequence of observations and across mixtures while PLDA ties the latent factors across speakers [19]. The Factor Analysis implementation follows [20] with minimum divergence step as described in[21].

SIDEKIT includes a standard i-vector extractor as well as two fast implementations based on the work of [22]. Several normalization algorithms are included: Eigen Factor Radial [23], Spherical Nuisance Normalization [23], LDA, WCCN [17] and various scoring methods: Cosine [24], Mahalanobis, Two-Covariance model [25], as well as partially closed-set PLDA likelihood ratio scoring [26, 27, 28].

We also made binding of Support Vector Machines [29, 1] available by simply compiling the LibSVM toolkit [30] and placing a copy of the library in SIDEKIT's directory. Nuisance Attribute Projection (NAP)[31] commonly used for speaker verification was implemented as well.

Models and classifiers available in SIDEKIT cover the standard development for speaker recognition.

3.3. Evaluation and visualization

Based on the BOSARIS toolkit, SIDEKIT includes tools to compute Equal Error Rate (EER), Decision Cost Function (DCF) and minimum-DCF and plot two types of Detection Error Trade-off (DET) curves: steppy (from the ROC) and ROC Convex-Hull. It is also possible to indicate points on the curve beyond which miss rate and false alarm rates are not reliable. Figures 1 and 2 were generated using SIDEKIT.

4. GETTING STARTED WITH SIDEKIT

4.1. Installation

The SIDEKIt is easily accessible via the Pypi repository and a simple command line[3] that will install all necessary Python modules at once. Another way to get the sources is to clone the SIDEKIT GIT repository [4].

4.2. Ready to run tutorials

Ready to run tutorials on standard databases are made available on the web portal for easy reproducibility and comparison. Figure 1 is obtained by following the tutorial on the RSR2015 database [32] for simple GMM-UBM and GMM-SVM. 13 MFCC plus the log-energy and their Δ and $\Delta\Delta$ are normalized using CMVN after a RASTA filtering to train a 128 distribution UBM. MAP adaptation is performed for each speaker following the protocol proposed in [32].

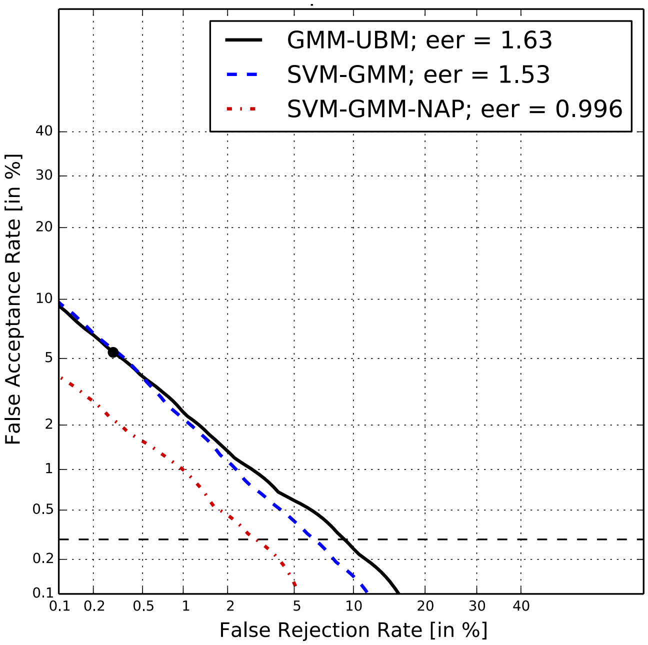

Fig. 1: DET curve on the male impostor-correct condition of RSR2015 protocol Part I.

Figure 2 shows the performance of the standard i-vector system from the on-line NIST-SRE tutorial using different scoring functions on the male part of the extended-core task of NIST-SRE10 [33]. A 512-distribution UBM and a total variability matrix of rank 400 are trained using 13 MFCC plus the log-energy and their Δ and $\Delta\Delta$, normalized using CMVN after a RASTA filtering. Recordings from the Switchboard corpora, NIST-SRE 2004, 2005, 2006 and 2008 are used to train the meta-parameters. Note that the selection of training data is done automatically and might not be optimal but demonstrates the state-of-the-art performance of the toolkit. Details about the configurations are available on the SIDEKIT tutorial web page

4.3. Tools for the community

To support the use of SIDEKIT, a web portal including a complete documentation, links on related tools, tutorials, references on related articles is available at `http://lium.univ-lemans.fr/sidekit/`

A GIT repository is freely accessible for installation and contributions will be welcome. A mailing list is open for developers and users to exchange comments and help [5].

[3] pip install sidekit

[4] `gitclonehttps://antho_l@bitbucket.org/antho_l/sidekit.git`

[5] registration via the SIDEKIT web portal

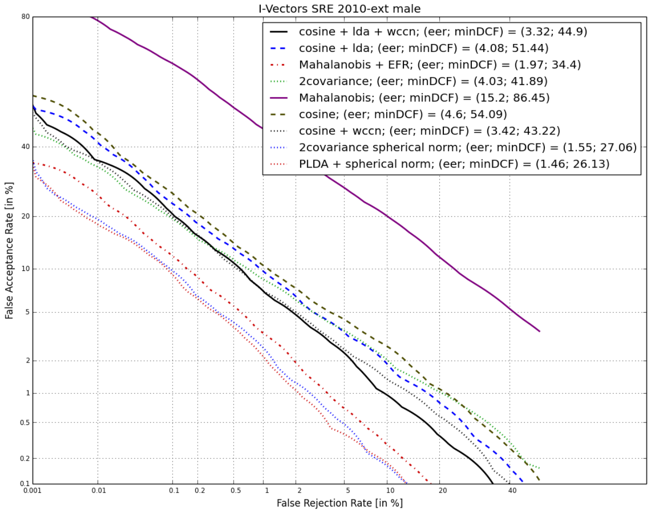

Fig. 2: DET curves of different scoring techniques in the i-vector framework for the condition 5 of the male core-extended task of NIST-SRE2010.

5. DISCUSSION

We have presented SIDEKIT, a new open-source toolkit for speaker recognition. To our knowledge, it is the most comprehensive toolkit available that provides an end-to-end solution for speaker recognition with a variety of ready-to-use state-of-the-art algorithms. We hope that its simple and efficient 100% Python implementation, the tutorials and complete documentation would benefit researchers, students and industry practitioners alike. In the near future, there is plan to include tools for language identification and speaker diarization as well as developing a streaming interface that is the most important limitation of the current version of the toolkit. Currently, developers are working on a bridge with Theano[6] to provide a simple integration of neural networks in the tool-chain [7].

6. ACKNOWLEDGEMENTS

We would like to thank Niko Brümmer and Agnitio for allowing us to port part of the BOSARIS codes to SIDEKIT.

7. REFERENCES

[1] 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.

[2] D. Bansé, G. R. Doddington, D. Garcia-Romero, J. J. Godfrey, C. S. Greenberg, J. Hernández-Cordero, J. M. Howard, A. F. Martin, L. P. Mason, A. McCree, and D. A. Reynolds, "Analysis of the second phase of the 2013–2014 i-vector machine learning challenge," in *Annual Conference of the International Speech Communication Association (Interspeech)*, 2015, pp. 3041–3045.

[3] A. Larcher, J.-F. Bonastre, B. Fauve, K. A. Lee, C. Lévy, H. Li, J. S. Mason, and J.-Y. Parfait, "ALIZE 3.0 - Open Source Toolkit for State-of-the-Art Speaker Recognition," in *Annual Conference of the International Speech Communication Association (Interspeech)*, 2013, pp. 2768–2773.

[4] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

[5] V. Gupta, P. Kenny, P. Ouellet, and T. Stafylakis, "I-vector-based Speaker Adaptation of Deep Neural Networks for French Broadcast Audio Transcription," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, 2014.

[6] S. O. Sadjadi, M. Slaney, and L. Heck, "MSR Identity Toolbox v1.0: A MATLAB Toolbox for Speaker-Recognition Research," *Speech and Language Processing Technical Committee Newsletter*, vol. 1, no. 4, November 2013.

[7] E. Khoury, L. E. Shafey, and S. Marcel, "Spear: An open source toolbox for speaker recognition based on Bob," in *International Conference on Audio, Speech and Signal Processing (ICASSP)*, 2014.

[8] S. Young and S.J.Young, "The HTK Hidden Markov Model Toolkit: Design and Philosophy," *Entropic Cambridge Research Laboratory, Ltd*, vol. 2, pp. 2–44, 1994.

[9] M. McLaren, N. Scheffer, L. Ferrer, and Y. Lei, "Effective use of DCTs for contextualizing features for speaker recognition," in *International Conference on Audio, Speech and Signal Processing (ICASSP)*, 2014, pp. 4027–4031.

[10] M. McLaren and Y. Lei, "Improved speaker recognition using DCT coefficients as features." in *International Conference on Audio, Speech and Signal Processing (ICASSP)*, IEEE, Ed., 2015, pp. 4430–4434.

[11] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Odyssey Speaker and Language Recognition Workshop*, 2001.

[6] `http://deeplearning.net/software/theano/`

[7] by the time this article is published, the language ID, diarization tools and bridge to Theano are already available on-line

[12] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, “Speaker Verification Using Adapted Gaussian Mixture Models,” *Digital Signal Processing*, vol. 10, pp. 19–41, 2000.

[13] 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, pp. 1435–1447, 2007.

[14] 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.

[15] 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.

[16] S. J. Prince, *Computer Vision: Models Learning and Inference*. Cambridge University Press, 2012.

[17] N. Dehak, R. Dehak, J. Glass, D. Reynolds, and P. Kenny, “Cosine similarity scoring without score normalization techniques,” in *Odyssey Speaker and Language Recognition Workshop*. Odyssey, 2010, pp. 1–5.

[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, pp. 1680–1683.

[19] L. P. Chen, K. A. Lee, B. Ma, W. Guo, H. Li, and L. R. Dai, “Local variability modeling for text-independent speaker verification,” in *Odyssey: Speaker and Language Recognition Workshop*, 2014.

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

[22] 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.

[23] 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, pp. 1–8.

[24] N. Dehak, R. Dehak, P. Kenny, N. Brummer, P. Ouellet, and P. Dumouchel, “Support Vector Machines versus Fast Scoring in the Low-Dimensional Total Variability Space for Speaker Verification,” in *Annual Conference of the International Speech Communication Association (Interspeech)*, 2009, pp. 1559–1562.

[25] N. Brümmer and E. de Villiers, “The speaker partitioning problem,” in *Odyssey Speaker and Language Recognition Workshop*, 2010, pp. 1–8.

[26] S. J. Prince, J. Warrell, J. Elder, and F. Felisberti, “Tied factor analysis for face recognition across large pose differences,” *IEEE transactions on Pattern Analysis and Machine intelligence*, vol. 30, no. 6, pp. 970–984, 2008.

[27] P. Kenny, “Bayesian speaker verification with heavy-tailed priors,” in *Odyssey Speaker and Language Recognition Workshop*, 2010.

[28] 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, pp. 3651–3655.

[29] 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 *International Conference on Audio, Speech and Signal Processing (ICASSP)*, vol. 1, 2006, pp. 97–100.

[30] 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.

[31] 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.

[32] A. Larcher, K. A. Lee, B. Ma, and H. Li, “Text-dependent Speaker Verification: Classifiers, Databases and RSR2015,” *Speech Communication*, vol. 60, pp. 56–77, 2014.

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