SPEAR: AN OPEN SOURCE TOOLBOX FOR SPEAKER RECOGNITION BASED ON BOB

Elie Khoury, Laurent El Shafey, Sébastien Marcel

Idiap Research Institute, Martigny, Switzerland

ABSTRACT

In this paper, we introduce *Spear*, an open source and extensible toolbox for state-of-the-art speaker recognition. This toolbox is built on top of Bob, a free signal processing and machine learning library. *Spear* implements a set of complete speaker recognition toolchains, including all the processing stages from the front-end feature extractor to the final steps of decision and evaluation. Several state-of-the-art modeling techniques are included, such as *Gaussian mixture models*, *inter-session variability*, *joint factor analysis* and *total variability* (i-vectors). Furthermore, the toolchains can be easily evaluated on well-known databases such as NIST SRE and MOBIO. As a proof of concept, an experimental comparison of different modeling techniques is conducted on the MOBIO database.

Index Terms— Speaker recognition, open source, Gaussian mixture model, inter-session variability, joint factor analysis, I-Vectors, NIST SRE, MOBIO.

1. INTRODUCTION

Automatic speaker recognition is the use of a machine to recognize a person's identity from the characteristics of his voice. The technology employed in this field has now reached a good level of performance due to the success of new paradigms such as *session variability* modeling [1, 2] and *total variability* (i-vectors) modeling [3]. Furthermore, it benefited from improvements in channel compensation [4, 5, 6] and noise reduction [7, 8] techniques. In addition, such techniques have successfully been used for other speaker-based tasks (e.g., speaker diarization [9]) and other biometric modalities (e.g., face recognition [10]). This explains the increased number of embedded and distributed applications that entered the market recently. As a matter of fact, the NIST speaker recognition evaluation (SRE) series [11] have seen a record in their number of participants in the last editions (58 sites in 2012). The same trend was observed at the speaker recognition evaluation in mobile environments (12 sites in 2013) [12].

A typical speaker recognition toolchain consists of speech activity detection, feature extraction and normalization, background modeling, target speaker enrollment, matching (score computation), score normalization and decision. Such a toolchain is depicted in Figure 1. In order to properly evaluate and compare algorithms, an audio database is usually split into three subsets: background training, development (DEV) and evaluation (EVAL). Both DEV and EVAL subsets are further split into enrollment (DEV.Enroll and EVAL.Enroll) and test (DEV.Test and EVAL.Test). This database partitioning defines an evaluation protocol, that sometimes ensures similar conditions in the three subsets.

Although automatic speaker recognition is investigated since the 1970s, only few related open source tools are available [13] and a good number of them are either outdated or incomplete. Therefore, running experiments with state-of-the-art systems and comparing their results with the ones of any newly proposed approach is often challenging and time consuming. Interestingly, a solution to this problem has recently been proposed for face recognition in [14], with a toolbox that allows fair evaluations of state-of-the-art systems on several publicly available databases. Following the same spirit, we developed *Spear*,[1] an open source toolbox for state-of-the-art speaker recognition.

The contribution of this paper is to present *Spear*, which is, to the best of our knowledge, the first open source and extensible toolbox that provides complete toolchains for state-of-the-art speaker recognition, from the front-end feature extractor to the final steps of decision and evaluation. Experiments can be conducted both on well-known databases such as NIST SRE and MOBIO, and on in-house datasets with user-defined protocols. This constitutes an ideal playground for researchers, since it allows rapid prototyping of novel ideas and testing meta-parameters of existing algorithms.

The remainder of the paper is structured as follows: Section 2 gives a brief review of existing open source tools used by the speaker recognition community. Section 3 presents the different features of *Spear*. Section 4 shows experimental results. Section 5 concludes the paper.

2. PRIOR WORK

Several existing tools are helping researchers to build and evaluate their speaker recognition systems. At the front-end level, one can cite HTK[2] (Hidden Markov Model Toolkit)

[1] `http://pypi.python.org/pypi/bob.spear`

[2] `http://htk.eng.cam.ac.uk/`

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

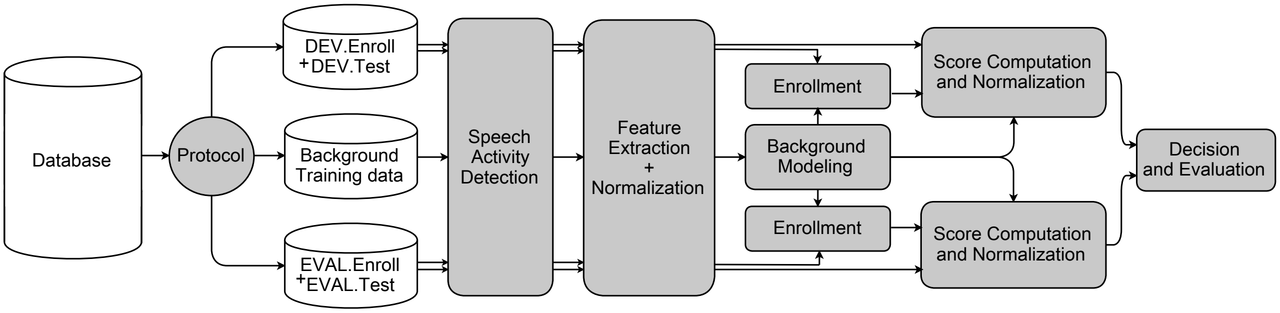

Fig. 1. TYPICAL SPEAKER RECOGNITION TOOLCHAIN. *This figure shows the main stages of a speaker recognition system.*

and SPro[3] (speech signal processing toolkit) for feature extraction, and [15] for voice activity detection. For the modeling and classification, researchers often use the GMM implementation of HTK or Matlab, the JFA Matlab implementation from Brno University of Technology[4] or the *support vector machine* (SVM) implementation in LIBSVM.[5] For score calibration and fusion, decision and evaluation, BOSARIS[6] or Focal[7] toolkits are mainly used. Therefore, to build their systems, researchers are often constrained to deal with several programming and scripting languages, different types and versions of operating systems, as well as various file formats.

To solve this problem, few researchers have worked on open source toolboxes where several speaker recognition modules are connected together. For example, the Munich automatic speaker verification framework (MASV) [16] provides an experimental system that depends on: HTK for feature extraction and modeling, Perl scripts to control enrollment and testing of speaker models, and Matlab scripts for score normalization and evaluation. This system is distributed under GNU General Public License (GPL), but unfortunately, it depends on HTK and Matlab that have license restrictions, which make its use very constrained and not affordable to all researchers. In addition, this system in now outdated (last update dates back to 2004), and thus does not include the latest state-of-the-art modeling techniques such as JFA and i-vectors.

Among all the existing tools, the closest competitor to *Spear* is probably ALIZE [17, 18]. This is an open source platform that provides C++ implementation for energy-based speech activity detection, background modeling including universal background modeling (UBM), subspace (for JFA and i-vectors) modeling and session compensation, target model enrollment, score computation and normalization. Unlike *Spear*, ALIZE does not contain an integrated feature extraction module, but is able to read features extracted with both HTK and SPro. In addition, it does not benefit from user-friendly and Matlab-like scripting languages such as Python, which provides a free solution and an ideal environment for rapid development and testing of new ideas. Finally, ALIZE does not rely on an elaborated file format for storing and managing data (only "raw" and "txt" formats supported), which makes it difficult to handle, view, analyze, and exchange data (e.g. model or score files).

3. SPEAR TOOLBOX

The strength of *Spear* first comes from the use of Bob[8] [19], which is designed to meet the needs of researchers by providing efficient C++ implementations of a large set of machine learning and signal processing algorithms. Bob also provides a researcher-friendly Python environment, which, among others, helps reducing development time. The interaction between Python and C++ environments is facilitated by a thin layer, seamless to the user. In addition, Bob relies on the open and portable HDF5 library and file format for storing and handling data, for which many tools are already available for visualization and analysis purposes. The entire code is well documented and nightly tested against several platforms.[9]

Spear also takes advantages of the experience acquired from *facereclib*[10] [14], an open source face recognition toolbox that aims to compare a variety of state-of-the-art algorithms on several facial image databases. Similarly to *facereclib*, *Spear* includes a set of configurable command line scripts, that allows to run different toolchains (GMM, JFA, ISV, and i-vectors based-systems) on well-known or in-house speaker recognition databases. In addition, experiments can be executed either on a local workstation (single or multiple processes) or on a grid computing infrastructure.[11]

In the subsequent sections, we present the different techniques that are currently available at each processing stage of the toolchain.

[3] `https://gforge.inria.fr/projects/spro`

[4] `http://speech.fit.vutbr.cz/software/joint-factor-analysis-matlab-demo`

[5] `http://www.csie.ntu.edu.tw/\textasciitildecjlin/libsvm/`

[6] `https://sites.google.com/site/bosaristoolkit/`

[7] `https://sites.google.com/site/nikobrummer/focal`

[8] `http://www.idiap.ch/software/bob`

[9] `http://www.idiap.ch/software/bob/buildbot/waterfall`

[10] `http://pypi.python.org/pypi/facereclib`

[11] `http://pypi.python.org/pypi/gridtk`

3.1. Preprocessing

Two techniques for speech activity detection (SAD) are implemented. The first one is a simple unsupervised energy-based SAD where frame-level energy values are computed, normalized and then classified into two classes. The class with the higher mean is considered as speech, and corresponding speech segments are hence kept before being smoothed. The second technique is based on a combination of the energy and its modulation around 4Hz [20]. A simple adaptive thresholding is applied on both features to remove non-speech parts. This technique was shown to work well on data acquired in mobile environment [21]. In addition, the toolbox supports the use of external or manually-labeled SAD.

3.2. Feature extraction

Spear provides an efficient implementation of spectral and cepstral (MFCC and LFCC) features, that can be optionally coupled with their first and second derivatives. *Cepstral Mean and Variance Normalization* technique is also integrated in the toolbox. In addition, it provides interfaces to read HTK and SPro features.

3.3. Modeling, enrollment, and score computation

Gaussian mixture model (GMM) [22], *joint factor analysis* [1], *inter-session variability* (ISV) [2], *total variability* (i-vectors) [3] state-of-the-art modeling techniques are integrated in the toolbox. They rely on efficient C++ implementations available in Bob.

The GMM system includes *universal background model* (UBM) training using the *maximum-likelihood* (ML). Both simple and parallel implementations of the training are supported. The target model enrollment is done using *maximum-a-posteriori* (MAP) adaptation [23]. The matching between a test utterance and a target model is performed using *log-likelihood ratio* (LLR). A linear approximation of the LLR [24] is also supported by this system. In addition, ZT-norm [25] for score normalization can optionally be used.

The other three modeling techniques are built on top of the GMM modeling. Once the UBM training is achieved, zero-, first- and second-order statistics are computed on each speech utterance [1].

For the JFA system, the statistics of the utterances belonging to the training set are used to estimate eigenvoice ($\boldsymbol{V}$) and eigenchannel ($\boldsymbol{U}$) matrices, as well as a matrix ($\boldsymbol{D}$) that models the residual noise. Matching between speaker models and speech utterances is performed using the linear scoring approximation [24].

The ISV system is similar to the JFA system, with the only difference that there is no eigenvoice matrix ($\boldsymbol{V}$) to estimate.

Finally, the i-vectors system involves the computation of a total variability matrix ($\boldsymbol{T}$), which is estimated using the same algorithm adopted to train $\boldsymbol{V}$. Both simple and parallel implementations are available for this purpose. Low-dimensional i-vectors are then extracted from each of the speech utterances. *Whitening* [26], *length normalization* [5], *linear discriminant analysis* (LDA) [27] and *within-class covariance normalization* (WCCN) [6] are included, which aim to map i-vectors into a more adequate space. Matching between a test utterance and a target speaker is done using either a fast scoring based on cosine distance between i-vectors or *probabilistic linear discriminant analysis* (PLDA) [28]. In this toolbox, an efficient and scalable implementation of PLDA is used [29].

3.4. Score Fusion

In recent speaker recognition evaluation campaigns such as NIST SRE [11] and MOBIO [12], it has been shown that fusion substantially boosts the recognition performance. Therefore, a score fusion strategy based on *logistic regression* [30] is also integrated.

3.5. Decision and Evaluation

Evaluation measures such as *Equal error rate* (EER), *half total error rate* (HTER), *minimum decision cost function* (minDCF), detection error trade-off (DET), receiver operating characteristic (ROC), expected performance curve (EPC) [31], *log-likelihood ratio cost* (CLLR) and *minimum log-likelihood ratio cost* (min-CLLR) are implemented in Bob and available in *Spear*. This makes the exercise of tuning the hyperparameters and evaluating the systems very easy and fast-forward.

3.6. Databases

Several databases with well-defined evaluation protocols are supported in *Spear* to improve the reproducibility and comparability of scientific publications.

Previously defined protocols for NIST SRE 2012 [11], MOBIO [21], BANCA [32], TIMIT and Voxforge[12] are included in *Spear*. The development set of NIST SRE 2012 protocol[13] was prepared by I4U partners [33] during their participation to the evaluation. Voxforge is an open source database, for which we created an open source protocol.[14] It is used as a toy example that allows researchers to freely run and test the toolbox.

Finally, experiments on in-house databases with user-defined protocols can be easily conducted.

4. EXPERIMENTS

As a proof of concept, we conducted an experimental comparison between GMM, ISV, i-vectors and their fusion on the the *mobile-0* protocol of the MOBIO database. Details about the database and this protocol can be found in [21].

[12] http://www.voxforge.org

[13] http://pypi.python.org/pypi/spear.nist_sre12/

[14] http://pypi.python.org/pypi/xbob.db.voxforge

Table 1. PERFORMANCE SUMMARY ON **MOBILE-0**. *This table reports the EER (%) on* DEV *and* HTER *(%) on* EVAL, *and min-CLLR on both of them, obtained with the* ***mobile-0*** *protocol.*

	Set	Measure	GMM	ISV	i-vectors	Fusion
Male	DEV	EER	13.41	10.40	11.31	7.31
		min-CLLR	0.4509	0.3708	0.3795	0.2610
	EVAL	HTER	12.12	10.36	11.11	7.89
		min-CLLR	0.4189	0.3621	0.3621	0.2769
Female	DEV	EER	17.94	12.22	12.59	9.21
		min-CLLR	0.5693	0.4214	0.4182	0.3197
	EVAL	HTER	17.68	16.23	17.36	14.65
		min-CLLR	0.5464	0.5156	0.5641	0.4813

In these experiments, GMMs are composed of 512 Gaussian components. For ISV, the rank of the subspace $\boldsymbol{U}$ is set to 50, whereas for i-vectors the rank of $\boldsymbol{T}$ is set to 400.[15] In addition to enforcing reproducible research, satellite packages can also be used to extend the toolbox with new algorithms that can be integrated at any stage of the toolchain.

Table 1 shows the results of the four systems in terms of EER on DEV set, HTER on EVAL set and min-CLLR on both of them, and for both Male and Female speakers. It can be observed that, as reported in the literature, ISV and i-vectors systems are more accurate than the GMM baseline. However, Table 1 shows that ISV is slightly better than i-vectors on this particular database. This can be explained by the lack of data needed to train the different stages of the i-vectors toolchain. The last column of Table 1 also shows that combining the three single systems substantially boosts the system performance: HTER are 7.9% and 14.7% for Male and Female speakers, respectively. Similar trends are shown by the DET curves shown in Figure 2.

5. CONCLUSIONS

In this paper we presented *Spear*, an open source speaker recognition toolbox based on Bob. It provides several toolchains relying on state-of-the-art modeling techniques such as inter-session variability modeling and i-vectors. The combination of Python and C++ offers a researcher-friendly environment for rapid development and testing of novel ideas. *Spear* is extensible and an ongoing community effort; contributions are encouraged and will be integrated.

6. ACKNOWLEDGMENT

The development leading to this toolbox has received funding from the Swiss National Science Foundation (SNSF) under the LOBI project and from the European Community's Seventh Framework Programme (FP7) under grant agreement 284989 (BEAT).

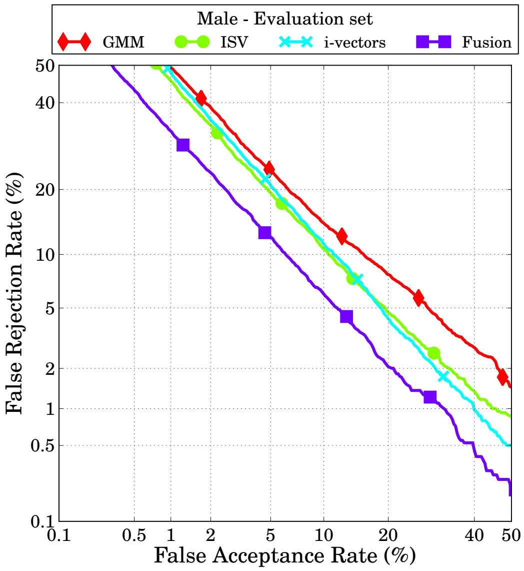

Fig. 2. DET CURVES ON THE EVALUATION SET OF MOBIO DATABASE, PROTOCOL MOBILE-0. *This figure shows the performance of GMM, ISV, i-vectors and their fusion for Male speakers.*

7. REFERENCES

[1] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1435–1447, 2007.

[2] R. Vogt and S. Sridharan, "Explicit modelling of session variability for speaker verification," *Computer Speech & Language*, vol. 22, no. 1, pp. 17–38, 2008.

[3] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 19, pp. 788–798, 2011.

[4] A. Solomonoff, W.M. Campbell, and I. Boardman, "Advances in channel compensation for svm speaker recognition," in *IEEE ICASSP*, 2005, vol. 1, pp. 629–632.

[5] D. Garcia-Romero and C.Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *INTERSPEECH*, 2011, pp. 249–252.

[6] N. Dehak, R. Dehak, P. Kenny, N. Brümmer, P. Ouellet, and P. Dumouchel, "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification," in *INTERSPEECH*, 2009, pp. 1559–1562.

[7] J. Ming, T.J. Hazen, J.R. Glass, and D.A. Reynolds, "Robust speaker recognition in noisy conditions," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1711–1723, 2007.

[15] A script containing the instructions to reproduce these experiments is available in the package.

[8] Y. Lei, L. Burget, and N. Scheffer, "A noise robust i-vector extractor using vector taylor series for speaker recognition," in *IEEE ICASSP*, 2013, pp. 6788–6791.

[9] M. Rouvier, G. Dupuy, P. Gay, E. Khoury, T. Merlin, and S. Meignier, "An open-source state-of-the-art toolbox for broadcast news diarization," in *INTERSPEECH*, 2013.

[10] R. Wallace and M. McLaren, "Total variability modelling for face verification," *IET Biometrics*, vol. 1, no. 4, pp. 188–199, 2012.

[11] C. Greenberg et al., "The 2012 NIST Speaker Recognition Evaluation," in *INTERSPEECH*, 2013.

[12] E. Khoury et al., "The 2013 speaker recognition evaluation in mobile environment," in *IAPR International Conference on Biometrics*, 2013.

[13] H. Li and B. Ma, "Techware: Speaker and spoken language recognition resources [best of the web]," *Signal Processing Magazine, IEEE*, vol. 27, no. 6, pp. 139–142, 2010.

[14] M. Günther, R. Wallace, and S. Marcel, "An open source framework for standardized comparisons of face recognition algorithms," in *ECCV, Workshops and Demonstrations*, 2012, vol. 7585, pp. 547–556.

[15] T. Kinnunen and P. Rajan, "A practical, self-adaptive voice activity detector for speaker verification with noisy telephone and microphone data," in *IEEE ICASSP*, 2013.

[16] U. Türk and F. Schiel, "Speaker verification based on the german veridat database," in *INTERSPEECH*, 2003.

[17] J.-F. Bonastre et al., "ALIZE/SpkDet: a state-of-the-art open source software for speaker recognition," in *Proc. Odyssey: the Speaker and Language Recognition Workshop*, 2008.

[18] A. Larcher et al., "ALIZE 3.0 - Open Source Toolkit for State-of-the-Art Speaker Recognition," in *INTERSPEECH*, 2013.

[19] A. Anjos, L. El Shafey, R. Wallace, M. Günther, C. McCool, and S. Marcel, "Bob: a free signal processing and machine learning toolbox for researchers," in *ACM International Conference on Multimedia*, 2012.

[20] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," in *IEEE ICASSP*, 1997, vol. 2, pp. 1331–1334.

[21] E. Khoury, L. El Shafey, C. McCool, M. Günther, and S. Marcel, "Bi-modal biometric authentication on mobile phones in challenging conditions," *Image and Vision Computing*, 2013.

[22] 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.

[23] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE Trans. on Speech and Audio Processing*, vol. 2, pp. 291–298, 1994.

[24] O. Glembek, L. Burget, N. Dehak, N. Brümmer, and P. Kenny, "Comparison of scoring methods used in speaker recognition with joint factor analysis," in *IEEE ICASSP*, 2009, pp. 4057–4060.

[25] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," *Digital Signal Processing*, vol. 10, no. 1, pp. 42–54, 2000.

[26] L. Burget, O. Plchot, S. Cumani, O. Glembek, P. Matejka, and N. Brümmer, "Discriminatively trained probabilistic linear discriminant analysis for speaker verification," in *IEEE ICASSP*, 2011, pp. 4832–4835.

[27] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 19, pp. 179–188, 1936.

[28] S. J. D. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *IEEE ICCV*, 2007, pp. 1–8.

[29] L. El Shafey, C. McCool, R. Wallace, and S. Marcel, "A scalable formulation of probabilistic linear discriminant analysis," *IEEE Trans. in Pattern Analysis and Machine Intelligence*, 2013.

[30] S. Pigeon, P. Druyts, and P. Verlinde, "Applying logistic regression to the fusion of the NIST'99 1-speaker submissions," *Digital Signal Processing*, vol. 10, no. 1–3, pp. 237–248, 2000.

[31] S. Bengio, J. Marithoz, and M. Keller, "The expected performance curve," in *In Proceedings of the 22nd International Conference on Machine Learning*, 2005, pp. 9–16.

[32] E. Bailly-Bailliere et al., "The BANCA database and evaluation protocol," in *International conference on Audio- and video-based biometric person authentication*, 2003, AVBPA'03, pp. 625–638.

[33] R. Saeidi et al., "I4U submission to NIST SRE 2012: a large-scale collaborative effort for noise-robust speaker verification," in *INTERSPEECH*, 2013.